Reconnaissance d'Actions à Partir de Capture de Mouvements

Mathieu Barnachon[1] Saida Bouakaz [1] Boubaker Boufama[2] Erwan Guillou[1]

[1]Université de Lyon, CNRS
Université Lyon 1, LIRIS, UMR5205, F-69622, France
{firstname}.{lastname}@liris.cnrs.fr
[2]School of Computer Science
University of Windsor
Windsor, Ontario
Canada N9B 3P4
boufama@uwindsor.ca

Résumé

Cet article présente une méthode de reconnaissance des actions à partir de données issues de Capture de Mouvements (MoCap).Notre but est de réaliser cette tâche en temps réel, et sans recours à une base d'apprentissage lourde. Notre approche s'efforce de reconnaître une action au cours de son déroulement. Pour cela, nous calculons un histogramme des poses de MoCap pour chaque action. Cet histogramme est construit à partir d'une distance entre les poses, et nous comparons les histogrammes ainsi créés à l'aide de la distance de Bhattacharyya. Grâce à un algorithme de programmation dynamique, ainsi qu'à une construction incrémentale de notre histogramme, nous sommes en mesure de reconnaître des actions à partir d'un flux de capture de mouvements. Nous présentons les résultats d'expérimentations sur des données de synthèse, provenant de la base de MoCap de CMU, complétées par des données réelles, issues du dispositif Kinect. Les résultats obtenus montrent l'efficacité de notre méthode.

1 Introduction

Le besoin de simplifier les interactions entre les utilisateurs et les machines afin de les rendre plus naturelles, plus simples est de plus en plus d'actualité. Des systèmes de captation de mouvements temps-réel plus ou moins efficaces visant aussi bien les professionnels que le grand public sont proposés . L'exemple le plus populaire est le dispositif Kinect de Microsoft [19]. Si ces techniques fournissent des données brutes intéressantes, leur utilisation reste néanmoins restreinte à des cas simples (quelques mouvements dans les jeux par exemple). Pour généraliser leur utilisation, des algorithmes doivent encore être développés afin de fournir des outils pour la reconnaissance et l'interprétation d'actions utilisables dans des applications quotidiennes.

Dans cet article, nous présentons un système capable de reconnaître des actions, à partir de Capture de Mouvement (MoCap), en temps réel, et sans avoir recours à une base d'apprentissage lourde. Chaque action est décrite par un histogramme de poses. Grâce à un algorithme de programmation dynamique, nous sommes en mesure de reconnaître des actions « réelles » apprises à partir de capture de mouvement de « synthèse ».

La suite de cet article est organisé comme suit. Dans la section 2 nous présentons un rapide état de l'art des méthodes de reconnaissance d'actions. La section 3 décrit notre méthode, basée sur des distances d'histogrammes, permettant d'apprendre ainsi que de reconnaître des actions. La section 4 présente nos résultats sur des données synthétiques et réelles, et enfin la section 5 propose une conclusion et donne quelques perspectives à ce travail.

2 État de l'art

Beaucoup de chercheurs se sont intéressés à l'analyse d'activités humaines (*human activity monitoring*) [22, 18, 5, 11]. Certains de ses travaux se sont spécialisés dans domaines spécifiques [16]. Nombre de ces méthodes s'appuient sur des algorithmes d'apprentissage (*machine learning*), qui nécessite un grand nombre de données, ainsi qu'une variance intra-classe importante au sein des actions étudiées.

Fujiyoshi *et al.* [7], ont été parmi les premiers à utiliser la squelétisation de la silhouette pour identifier des actions comme la marche ou la course. Cette solution a l'avantage d'être facile à appliquer, mais n'est utilisable que pour l'identification de mouvements simples. Bobick et Davis [3] ont proposé une méthode utilisant des modèles spatio-temporels pour la reconnaissance des activités humaines. Leur méthode fonctionne en temps réel et utilise une base de données des actions déjà extraites. Bien que leur processus de reconnaissance soit en temps réel, l'ajout de nouvelles actions à la base de données est coûteux en temps et peu souple. Xiong et Liu [20] ont utilisé des Modèles de Markov Cachés sur des silhouettes extraites. Les

actions reconnues sont limitées à des comportements humains simples. Ahmad et Lee [1] ont étendu le concept des historiques de mouvement proposé par Davis [6] a l'utilisation de silhouettes. Leur méthode, basée sur l'algorithme SVM, dépend de nombreux paramètres rendant les résultats obtenus sensibles à la qualité des silhouettes extraites et au point de vue de la caméra. Sans extraction de squelette, Ryoo [18] a proposé une méthode utilisant des histogrammes intégraux et un « sac de mots » (*bag of words)* pour la reconnaissance d'action au plus tôt. Il a adapté les caractéristiques 2D à l'action, spatio-temporelle par nature, et est ainsi capable de reconnaître une action à miparcours de son exécution avec une certitude de l'ordre de 50%. L'algorithme a été appliqué sur des activités simples, et gagnerait à être présenté dans le cadre des Interactions Homme Machine. Dans les travaux de Iv et Nevatia [12], les actions sont modélisées par un ensemble de poses clés virtuelles destinées à être utilisées dans le processus de reconnaissance. Cette méthode est toutefois limitée par les calculs coûteux ainsi que par le nombre de poses clés virtuelles disponibles. Allant au-delà des simples Interactions Homme Machine, Okada et Stenger [16] ont présenté des travaux, exploitant les silhouettes, permettant de construire une arborescence hiérarchique de forme. La capture de mouvement est réalisée en même temps que ce mouvement est utilisé dans un environnement virtuel où l'utilisateur peut interagir par le biais de son avatar. En raison des dépendances entre silhouettes, ils utilisent 30 modèles humains pour produire l'ensemble de toutes les configurations reconnaissables. Huang et Trivedi [10] ont présenté le concept d'histogramme cylindrique, basé sur une représentation voxelique, pour effectuer une reconnaissance en utilisant des chaînes de Markov cachées. Une méthode similaire a été proposée par Parameswaran et Chellappa [17] pour traiter des données de capture de mouvements avec marqueurs dans un espace de mouvement invariant. Les auteurs soulignent qu il n'y a pas d'invariance 3D dans l'espace de mouvement, ainsi toutes les actions doivent être décrites indépendamment. Dans [8] , les auteurs proposent une interprétation de mouvement en temps réel à l'aide de données de capture de mouvement simplifiées. Cependant, comme ils utilisent un sous-ensemble des données de capture de mouvement, leur solution exige une grande base de données d'actions similaires pour obtenir des résultats d'interprétation corrects. En outre, une étape de prétraitement complexe et couteuse en temps est nécessaire pour déterminer les actions similaires. Des approches stochastiques ont été mise en œuvre par Yao *et al.* [22] afin de trouver un espace latent discriminant les activités à reconnaître. Dans [21], les auteurs font appel à la capture de mouvement pour enrichir la reconnaissance d'actions complexes, effectuées par un humain. Ils démontrent ainsi la prépondérance de la capture de mouvement pour des actions humaines complexes par rapport aux méthodes basées sur l'apparence.

3 Méthode

À l'instar de Fujiyoshi *et al.* [7], nous avons recours aux squelettes extraits des images de la MoCap. Ainsi, on peut considérer qu'à chaque pose extraite d'une frame à un instant donné correspond un squelette. Dans la suite, pour simplifier la présentation et sans nuire à la généralité, nous considérons les poses de capture de mouvements comme des squelettes hiérarchiques, extraits à chaque intervalle de temps. Une action est une succession de squelettes (poses), ordonnée afin de transcrire le mouvement du corps humain lors de l'exécution de l'action considérée. Ainsi deux actions différentes peuvent comporter un ensemble de poses identiques. Cet ensemble peut être plus ou moins important. Par ailleurs, la même action exécutée par deux personnes différentes ou à des moments différents peut varier en vitesse, en amplitude, *etc.* Cependant, l'action garde la même apparence, ce qui permet de la reconnaître. Pour reconnaître des actions, nous avons besoin de les comparer et par conséquent de comparer deux poses. Pour comparer deux poses, nous utilisons la distance de Hausdorff [9] qu'on note par D. Ainsi on peut introduire la notion de ε-équivalence entre deux poses.

Definition 1. *Deux poses A et B sont dites ε-équivalentes si et seulement si :*

$$A \sim B \quad \Leftrightarrow \quad D(A,B) \leq \varepsilon. \tag{1}$$

Ces équivalences entre deux poses au sens de la distance de Hausdorff [9] permettent de construire les histogrammes représentant les fréquences de poses au cours du temps..

3.1 Histogrammmes

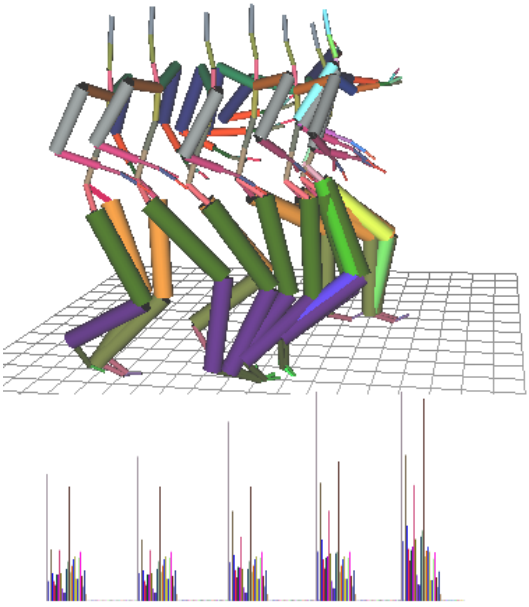


FIGURE 1 – Representation de l'action « Punch » et des histogrammes associés.

Pour effectuer la reconnaissance d'une action nous nous basons sur l'histogramme intégral d'une action au temps t. Notons par A^t, une action au temps t, et $\mathcal{P}$ l'ensemble des poses,

Definition 2. *Un histogramme intégral $\mathcal{IH}$ de A^t est défini par :*

$$\mathcal{IH}\left(A^t, \mathcal{P}\right) \;=\; \bigcup_{j=0}^{t} H^t\left(P^j\right) \qquad (2)$$

Où, $H^t\left(P^j\right)$ est la fréquence cumulée de la posture P^j au cours du temps.

3.2 Comparaison d'histogrammes

La distance entre deux histogrammes peut être évaluée par la métrique de Bhattacharyya [2], rappelée par l'équation 3 :

$$D_H(H_1, H_2) \;=\; \sqrt{1 - \sum_i \frac{\sqrt{H_1(i) \cdot H_2(i)}}{\sqrt{\sum_i H_1(i) \cdot \sum_i H_2(i)}}} \qquad (3)$$

À l'aide de la définition 2, et en utilisant la distance ci-dessus, la reconnaissance d'une action observée O^t à un moment t, peut être évaluée par le score maximal des distances entre l'action O^t et les différentes actions présentes dans le dictionnaire (apprises). Ceci peut être exprimé par :

$$A_0 \;=\; \arg\max_d \left[D_H\left(\mathcal{IH}\left(A_d^t, \mathcal{P}\right), \mathcal{IH}\left(O^t, \mathcal{P}\right)\right)\right] \qquad (4)$$

Où O^t est l'action observée, à reconnaître (en cours de déroulement), $d \in \{1, \cdots, n\}$, $n = |\mathcal{A}|$ est le nombre d'actions apprises. Ainsi A_0 est l'action identifiée.

3.3 Comparaison d'histogrammes incrémentale

Pour pouvoir reconnaître une action à mesure qu'elle se déroule, au lieu de calculer un score *a posteriori*, il serait judicieux de comparer des histogrammes à chaque pas de temps. L'avantage de l'histogramme intégral est qu'il donne la possibilité d'avoir un histogramme, pour chacune des actions apprises, à un intervalle quelconque. Cela nous permet de calculer un score incrémental pour chacune des actions possibles, à chaque pas de temps. Ainsi, il nous est possible de reconnaître les actions de façon "précoce".
Calculer les scores de similarité pour chacune des actions, à chaque pas de temps par leurs histogrammes est coûteux. Le processus est en $\mathcal{O}(n^2 \cdot t^2)$. En effet, à chaque étape, il faut comparer l'histogramme calculé avec tous les histogrammes partiels des actions apprises. Notons à ce niveau que les différentes évaluations sont indépendantes les unes des autres. Il est dès lors possible de paralléliser le processus comme le montre l'algorithme 1. cela étant, il ne s'agit que d'une amélioration technique de la solution. Cependant, nous pouvons remarquer qu'il n'est pas judicieux de comparer les histogrammes à tous les intervalles. En effet, l'histogramme construit à $t+1$ est égal à l'histogramme construit à t sauf pour une des composantes, dont la valeur est incrémentée de 1. Comme nous utilisons des comparaisons statistiques (distance de Bhattacharyya), il est préférable d'avoir des différences plus importantes entre les séries à comparer. Cela nous permettrait de supprimer les actions très peu probables en cours d'évaluation.

Algorithm 1 Reconnaissance parallèle d'actions

```
t ← 0 ;
repeat
  P ← MoCap(t)
  UpdateHistogram(A, P, t) ;
  for all (en Parallèle) A' ∈ ActionSet do
    s = Compare(H_A(t), H_A'(t)) ;
    if s ≤ ε then
      L'action A' a été reconnue.
      t ← 0 ;
    else
      t ← t + 1 ;
    end if
  end for
until Fin du flux
```

3.4 Comparaison dynamique d'histogrammes

Considérons des sous parties d'un histogramme incrémental entre t et $t+\delta t$, l'histogramme incrémental s'écrit : $\mathcal{IH}(H, [t, t+\delta t])$. Dès lors, nous cherchons des correspondances entre les sous-histogrammes incrémentaux du modèle (issus d'actions apprises) et les décompositions de l'histogramme incrémental de l'action en cours de reconnaissance. Nous cherchons le maximum de vraisemblance de l'observation O^t, représentée par ses sous-histogrammes, et les sous-histogrammes de l'ensemble d'actions ($\mathcal{A}$).

$$P\left(O^t\right) \;=\; \arg\max_{d \in \mathcal{A}} \sum_{\Delta t} D\left(H_d^{[\Delta t]}, H^{[\Delta t]}\right) \qquad (5)$$

Bien que le paradigme de la programmation dynamique nous permette de calculer automatiquement le Δd optimal, il est préférable de limiter l'espace de recherche. Pour cela, nous considérons qu'une modification significative entre deux histogrammes peut avoir lieu dès lors qu'au moins 1% des composantes ont changé. En considérant une distribution uniforme des poses, et connaissant le nombre de poses (2442 poses uniques dans [14]), nous limitons l'intervalle de comparaison entre 20 et 50 poses successives, voir figure 2.

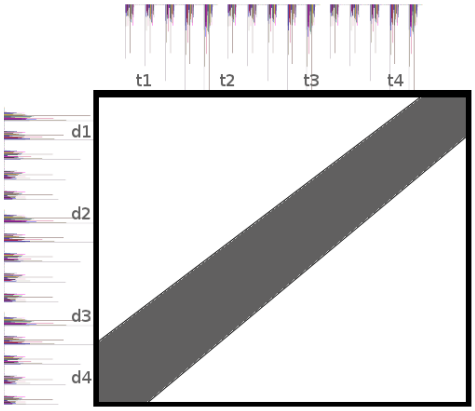
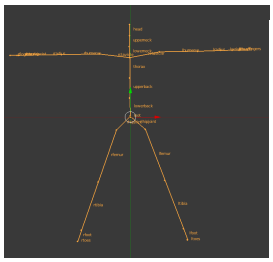
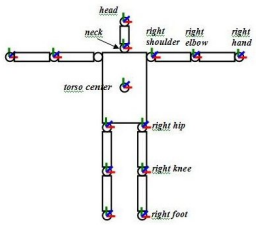


FIGURE 2 – Détermination dynamique de la longueur des sous-histogrammes utilisés pour reconnaître les actions, avec la limite de la zone de recherche en gris.

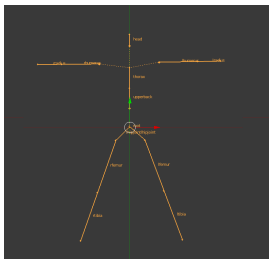
3.5 Flux de poses



(a) Squelette issu de [14, 13]



(b) Squelette issu de [19]



(c) Notre adaptation

FIGURE 3 – Représentation des squelettes utilisés, à l'apprentissage 3a et à la reconnaissance depuis un système temps-réel de MoCap 3b, ainsi que le tronc commun utilisé dans notre processus 3c.

Afin de procéder à la reconnaissance d'actions issues d'un flux de MoCap, il est souvent nécessaire d'adapter les squelettes. En effet, les systèmes d'extraction de poses temps-réel, tel [19], utilisent des squelettes moins détaillés que les systèmes de MoCap hors ligne, tel [4]. Nous utilisons les jeux de données issues des bases [14] et [13], à l'apprentissage. L'adaptation entre ces squelettes, l'un composé de 51 articulations, voir figure 3a et celui extrait par la MoCap live de [19], voir figure 3b, est réalisé *a priori*, voir figure 3c. Cette adaptation est une réduction des données non extraites dans les systèmes temps-réel, comme les extrémités des pieds et des mains, la multiplication des articulations « virtuelles » dans le torse pour simuler la flexibilité de la colonne vertébrale, *etc.* De plus, les articulations sont représentées dans un repère centré sur la personne, *i.e.* normalisé par rapport à la hauteur du personnage et exprimé dans le repère ayant la racine comme origine, nous ne sommes que peu dépendant des précisions apportées par les articulations entre le torse et les bras.

Nom	Appr.	Test
Walk	02_01	02_02 ;03_01 ;05_01 ;07_01 ;08_01
Run	02_03	09_01 ;17_01 ;35_22 ;77_10 ;141_02
Punch	143_23	02_05 ;111_19 ;113_13
Boxing	13_1 7	13_18 ;14_01 ;14_02 ;14_03 ;14_13 ;80_10
Jump	13_32	13_39 ;13_40 ;16_01 ;16_03 ;118_02
Shake hands	18_01	18_02 ;19_01 ;19_02 ;79_06 ;141_23 ;80_73
Laugh	13_14	13_15 ;13_16 ;14_17 ;14_18 ;14_19
Drink	13_09	14_04 ;14_37 ;23_13 ;79_38 ;79_40
Eat	79_12	79_15 ;79_42 ;80_11 ;80_33

TABLE 1 – Jeux de données créer à partir des actions de CMU.

4 Résultats

4.1 Jeux de données

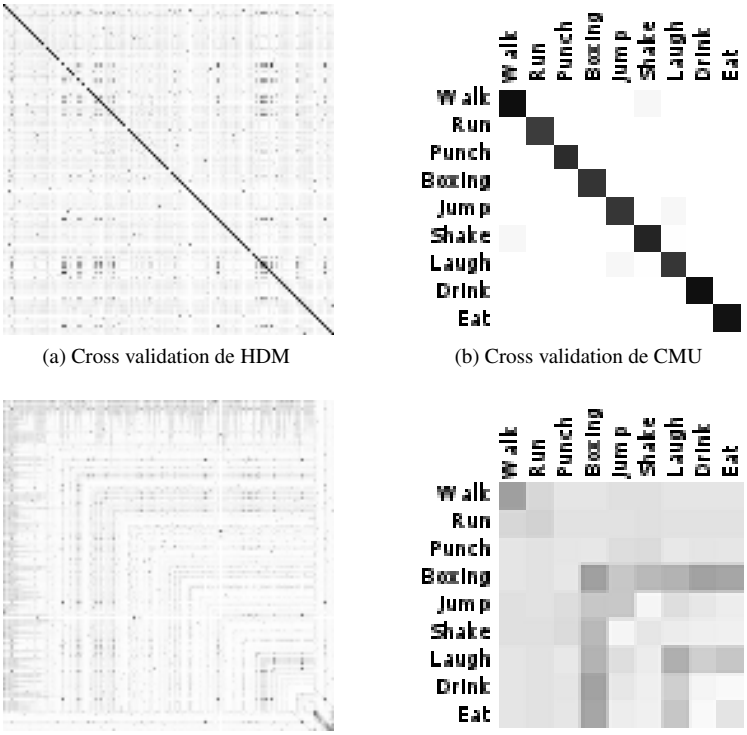
Afin d'évaluer la validité de notre solution, nous avons testé notre méthode sur des données issues de la base de capture de mouvements de l'université Carnegie Mellon (CMU) [13], ainsi que sur une segmentation en actions « élémentaires » faites par [15] à partir de la base de capture de mouvements de l'universität Bonn(HDM) [14].
La base HDM a été découpé en une partie apprentissage, composée de 130 classes : une action choisie aléatoirement parmi chacune des 130 classes d'actions, faites par 5 acteurs différents. Les 2207 actions restantes étant utilisées pour le jeu de test. Nous avons aussi créé un jeu de données, séparé entre apprentissage et test à partir des actions de CMU. Le détail se trouve dans la table 1.

4.2 Classification d'actions

En premier lieu, nous présentons la validation de la méthode en utilisant le jeu d'apprentissage comme jeu de reconnaissance, voir figures 4a et 4b. Nous obtenons un score de 100% sur les deux bases HDM et CMU. Dans ce cas, nous avons fixé $\varepsilon = 1,0$, et $\Delta t = 20$.
L'évaluation sur la base de test HDM, avec l'intégralité de chacune des actions, donne un score de $67,85\%$ et de 86.67% sur le jeu de test de CMU.
La figure 4 montre les matrices de confusion pour chacune des approches.

4.3 Reconnaissance précoce d'actions

Nous avons évalué l'efficacité de notre méthode quant à la reconnaissance d'actions précocement. La figure 5 montre l'évolution de la reconnaissance pour chaque pas de temps (en ne tenant pas compte de l'optimisation dynamique du



(a) Cross validation de HDM

(b) Cross validation de CMU

(c) Test de HDM

(d) Test de CMU

FIGURE 4 – Matrice de Confusion des jeux de données.

paramètre temporel). Le jeu d'apprentissage offre un score augmentant rapidement dès lors que le temps augmente. Notons toutefois que même lorsque peu d'action sont vues, les résultats sont proches de 80%. Ceci s'explique par le fait que la base HDM a été faite afin de réaliser des transitions fluides entre actions, donc avec des poses initiales importantes.

Lorsque l'on évalue notre solution sur l'ensemble de la base, le score augmente avec le nombre de poses extraites. Il est dès lors possible de donner une réponse quant à l'action en cours de reconnaissance très rapidement afin de laisser la puissance de calcul à l'application hôte (jeux vidéo, applications spécifiques, *etc.*).

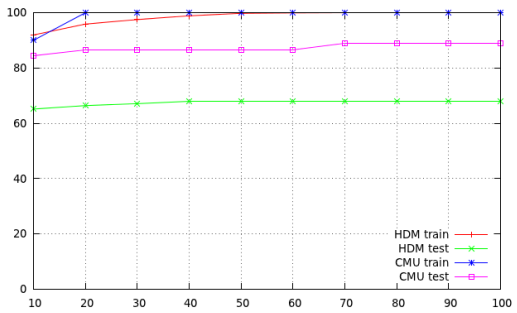


FIGURE 5 – Évolution de la performance de notre solution en fonction du temps, sur les bases HDM et CMU, en apprentissage (cross validation) et tests.

Jeux de données	Précision (moitié)	Précision (toute)
HDM (unique)	66,41%	67,86%
HDM (fusion)	95,74%	96,55%
CMU	84,45%	86,67%

TABLE 2 – Reconnaissance sur les bases HDM et CMU, à la moitié de l'action, et sur une action complète.

Un démonstrateur réel, contrôlé par un Kinect a montré l'efficacité de cette approche dans des cas d'utilisations réelles (Interface Homme Machine).

5 Conclusion

Dans cet article, nous avons introduit une nouvelle méthode de reconnaissance d'actions, en temps réel, capable de fournir une identification au fur et à mesure du déroulement de l'action. Nous avons appliqué une solution statistique simple, qui s'appuie sur des estimations de distance d'histogrammes intégraux. Comme nous l'avons montré à travers les différents résultats obtenus sur une grande variété d'actions, cette méthode a prouvé son efficacité et sa capacité à traiter des actions relativement complexes. Nous avons également identifié quelques limites, dont certaines ambigüités d'actions (manger et boire, par exemple), une solution possible serait d'introduire des primitives de plus haut niveau. Actuellement, nous nous appuyons sur une base apprise *a priori*. Nous travaillons sur une méthode incrémentale qui permettrait d'enrichir cette base à la vo-

lée, en déterminant de nouvelles actions reconnaissables au cours du temps (découvert au fil de la reconnaissance).

Références

[1] Mohiuddin Ahmad and Seong-Whan Lee. Variable silhouette energy image representations for recognizing human actions. *Image and Vision Computing*, 28(5) :814 – 824, 2010.

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

[3] Aaron F. Bobick and James W. Davis. The recognition of human movement using temporal templates. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(3) :257–267, 2001.

[4] Vicon Company. New products. *IEEE Computer Graphics and Applications*, 24 :107–108, 2004.

[5] N.P. Cuntoor, B. Yegnanarayana, and R. Chellappa. Activity modeling using event probability sequences. *IEEE Trans. Image Processing*, 17(4) :594–607, April 2008.

[6] J. Davis. Recognizing Movement using Motion Histograms. Technical report, MIT Media Lab Perceptual Computing Group, MIT, 1999.

[7] Hironobu Fujiyoshi and Alan J. Lipton. Real-time human motion analysis by image skeletonization. *Applications of Computer Vision, IEEE Workshop on*, 0 :15, 1998.

[8] Lei Han, Xinxiao Wu, Wei Liang, Guangming Hou, and Yunde Jia. Discriminative human action recognition in the learned hierarchical manifold space. *Image and Vision Computing*, pages 836 – 849, 2010.

[9] Felix Hausdorff. *Grundzüge der Mengeniehre*. Von Veit, Leipzig, 1914.

[10] Kohsia S. Huang and Mohan M. Trivedi. 3d shape context based gesture analysis integrated with tracking using omni video array. In *CVPR '05 : Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, page 80, Washington, DC, USA, 2005. IEEE Computer Society.

[11] Yuri A. Ivanov and Aaron F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22 :852–872, 2000.

[12] Fengjun Lv and Ramakant Nevatia. Single view human action recognition using key pose matching and viterbi path searching. In *CVPR*, 2007.

[13] MoCap CMU. The data used in this project was obtained from mocap.cs.cmu.edu. the database was created with funding from nsf eia-0196217. *http ://mocap.cs.cmu.edu/*.

[14] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber. Documentation mocap database hdm05. Technical Report CG-2007-2, Universität Bonn, June 2007.

[15] Meinard Müller, Andreas Baak, and Hans-Peter Seidel. Efficient and robust annotation of motion capture data. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 17–26, 2009.

[16] Ryuzo Okada and Björn Stenger. A single camera motion capture system for human-computer interaction. *IEICE - Trans. Inf. Syst.*, pages 1855–1862, 2008.

[17] Vasu Parameswaran and Rama Chellappa. View invariants for human action recognition. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2 :613, 2003.

[18] Michael Ryoo. Human activity prediction : Early recognition of ongoing activities from streaming videos. In *International Conference on Computer Vision*, 2007.

[19] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from a single depth image. In *CVPR*, 2011.

[20] Jing Xiong and ZhiJing Liu. Human motion recognition based on hidden markov models. In *Advances in Computation and Intelligence*, pages 464–471, 2007.

[21] Angela Yao, Juergen Gall, Gabriele Fanelli, and Luc Van Gool. Does human action recognition benefit from pose estimation ? In *Proceedings of the British Machine Vision Conference*, pages 67.1–67.11, 2011.

[22] Angela Yao, Juergen Gall, Luc Van Gool, and Raquel Urtasun. Learning probabilistic non-linear latent variable models for tracking complex activities. In *Neural Information Processing Systems (NIPS)*, 2011.