Université de Montréal

# Reconnaissance des actions humaines à partir d'une séquence vidéo

par

**Redha Touati**

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en informatique

Janvier, 2014

Université de Montréal

Faculté des études supérieures

Ce mémoire intitulé :

Reconnaissance des actions humaines à partir d'une séquence vidéo

présenté par

Redha Touati

a été évalué par un jury composé des personnes suivantes:

Jean Meunier
(Président-rapporteur)

Max Mignotte
(Directeur de recherche)

Derek Nowrouzezahrai
(Membre)

Mémoire accepté le

*À ma famille,*

*et à mes amis proches...*

# RÉSUMÉ

Le travail mené dans le cadre de ce projet de maîtrise vise à présenter un nouveau système de reconnaissance d'actions humaines à partir d'une séquence d'images vidéo. Le système utilise en entrée une séquence vidéo prise par une caméra statique. Une méthode de segmentation binaire est d'abord effectuée, grâce à un algorithme d'apprentissage, afin de détecter les différentes personnes de l'arrière-plan. Afin de reconnaitre une action, le système exploite ensuite un ensemble de prototypes générés, par une technique de réduction de dimensionnalité MDS, à partir de deux points de vue différents dans la séquence d'images. Cette étape de réduction de dimensionnalité, selon deux points de vue différents, permet de modéliser chaque action de la base d'apprentissage par un ensemble de prototypes (censé être relativement similaire pour chaque classe) représentés dans un espace de faible dimension non linéaire. Les prototypes extraits selon les deux points de vue sont amenés à un classifieur $K$-ppv qui permet de reconnaitre l'action qui se déroule dans la séquence vidéo. Les expérimentations de ce système sur la base d'actions humaines de Wiezmann procurent des résultats assez intéressants comparés à d'autres méthodes plus complexes. Ces expériences montrent d'une part, la sensibilité du système pour chaque point de vue et son efficacité à reconnaitre les différentes actions, avec un taux de reconnaissance variable mais satisfaisant, ainsi que les résultats obtenus par la fusion de ces deux points de vue, qui permet l'obtention de taux de reconnaissance très performant.

**Mots clés: Traitement de la vidéo, l'analyse des activités humaines, reconnaissance des gestes, réduction de dimensionnalité, reconnaissance des formes.**

# ABSTRACT

The work done in this master's thesis, presents a new system for the recognition of human actions from a video sequence. The system uses, as input, a video sequence taken by a static camera. A binary segmentation method of the the video sequence is first achieved, by a learning algorithm, in order to detect and extract the different people from the background. To recognize an action, the system then exploits a set of prototypes generated from an MDS-based dimensionality reduction technique, from two different points of view in the video sequence. This dimensionality reduction technique, according to two different viewpoints, allows us to model each human action of the training base with a set of prototypes (supposed to be similar for each class) represented in a low dimensional non-linear space. The prototypes, extracted according to the two viewpoints, are fed to a $K$-NN classifier which allows us to identify the human action that takes place in the video sequence. The experiments of our model conducted on the Weizmann dataset of human actions provide interesting results compared to the other state-of-the art (and often more complicated) methods. These experiments show first the sensitivity of our model for each viewpoint and its effectiveness to recognize the different actions, with a variable but satisfactory recognition rate and also the results obtained by the fusion of these two points of view, which allows us to achieve a high performance recognition rate.

**Keywords: Video processing, human gait analysis, gesture recognition, reduction of dimensionality, shape recognition.**

# TABLE DES MATIÈRES

# LISTE DES TABLES

# LISTE DES FIGURES

# LISTE DES ALGORITHMES

# ABBREVIATIONS

| | |
|---|---|
| 2D | Deux dimensions |
| 3D | Trois dimensions |
| CCD | Dispositif à transfert de charge |
| RVB | Rouge vert bleu |
| HSV | Teinte saturation valeur |
| MEI | Image d'energie du mouvement |
| MHI | Image d'historique du mouvement |
| MIL | Méthode d'apprentissage multi-instances |
| K-ppv | K plus proches voisins |
| SVM | Séparateurs à vaste marge |
| PDF | Fonction de densité de probabilité |
| ACP | Analyse en composante principale |
| MDS | Multidimensional scaling |
| LLE | Locally Linear Embedding |
| SIFT | Speeded Up Robust Features |
| SURF | Speeded Up Robust Features |
| ROC | Receiver operating characteristic |
| SVH | Système visuel humain |
| PDF | Probability Density Function (fonction de densité de probabilité) |
| MLP | Perceptron multicouches |

# REMERCIEMENTS

Je ne saurais, réellement, trouver les expressions éloquentes que mérite mon directeur Max Mignotte, afin de le remercier pour sa sympathie, ses encouragements, son aide, son dévouement pour le travail et sa totale présence. J'aimerais également le remercier pour ces diverses discussions. Je tiens à remercier Said Benameur pour ses directives et sa bienveillance. Je tiens à remercier tous les enseignants et les responsables de notre département. Mes remerciements vont aussi aux membres de jury. Enfin, je remercie toute ma famille et surtout ma mère pour son encouragement et son soutien tout au long de mes études.

Chapitre 1

# INTRODUCTION GÉNÉRALE

## *1.1 Introduction*

De nos jours, l'information extraite dans le domaine de la vision artificielle provient de diverses sources visuelles et sensorielles telles que la parole, la voix, une vidéo, une image. Les données vidéo permettent d'enregistrer les différents évènements réalisés par l'être humain à l'aide d'un capteur d'acquisition, tel que le capteur CCD.[1] Vu l'expansion de ces données enregistrées dans la vie quotidienne, le besoin de comprendre automatiquement ce qui se passe dans une vidéo est devenu une nécessité, d'où l'importance des systèmes de vision par ordinateur. Ces systèmes de vision exigent deux processus fondamentaux, le processus de bas niveau qui utilise des techniques ou des algorithmes de prétraitements, permettant l'extraction d'un ensemble de caractéristiques pertinentes comme la couleur, les différentes régions homogènes existantes dans l'image, la texture, le mouvement, etc. dont le but est d'obtenir une représentation plus concise et facilement analysable résumant le contenu de la donnée vidéo et fournissant une quantité d'informations sur la scène. Dans un deuxième temps, l'exploitation de ces attributs par un processus de haut niveau plus complexe permet d'analyser, et de reconnaitre l'activité qui se déroule dans la séquence vidéo.

Plusieurs applications de reconnaissance ont été développées dans le passé dans le domaine de la vidéo basée sur des systèmes de vision. Parmi ces applications on trouve les applications de reconnaissance de l'action humaine [4] [10] [14] [34] [35] [11] dans le domaine médical, le sport ou la vidéosurveillance pour en nommer

[1] http://legacy.jyi.org/volumes/volume3/issue1/features/peterson.html

quelques-unes. Le but de ces applications est d'observer les personnes et d'identifier automatiquement ce qu'elles font comme actions dans la séquence vidéo. L'action [23, 69], par exemple, est définie comme une activité périodique exécutée par une seule personne dans un intervalle de temps (cf. Fig. 1.1). Différents types d'actions peuvent être identifiées dans une séquence vidéo comme marcher, sauter, courir, s'assoir, etc. Ces actions sont produites par le mouvement des différentes parties du corps humain. La difficulté de n'importe quel système automatique de reconnaissance des actions humaines [3] provient principalement de la modélisation d'actions produites par la variabilité de la taille de chaque personne et sa manière d'exécuter l'action. Une autre difficulté provient de l'extraction des caractéristiques pertinentes qui vont représenter les parties du corps humain et leurs positions dans le temps ou encore de la diversité des conditions des prises de vues. En général, un système de reconnaissance des actions humaines est constitué de trois étapes:

- Étape de prétraitement
- Étape d'extraction de caractéristiques
- Étape de classification

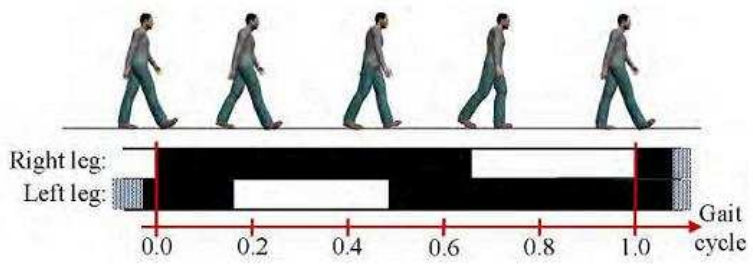



**Figure 1.1.** *Un cycle d'action pour l'activité de la marche humaine [23]*

## 1.2 État de l'art

Le problème de la reconnaissance des actions humaines a attiré l'attention de plusieurs chercheurs et les avantages et les inconvénients des différentes approches proposées ont été discutées au cours de ces dernières années. L'objectif ultime est de trouver une méthode de reconnaissance des actions humaines à la fois efficace, rapide et simple à implémenter. Dans ce domaine, les différentes approches existantes peuvent se classer en plusieurs catégories en fonction de la façon dont on représente ou modélise l'action [4].

Blank *et al.* [10] ont focalisé leur approche sur l'extraction d'une série de points qui représente l'orientation d'un pixel par rapport à son voisinage afin de décrire la forme de l'action. L'extraction de cette série de points est basée sur la résolution de l'équation de poisson pour chaque pixel du volume spatio-temporel. La reconnaissance de l'action s'effectue par une mesure de similarité entre les vecteurs de caractéristiques extraits à l'aide de l'algorithme $K$-ppv. Schuldt *et al.* [14] ont basé leur algorithme sur l'extraction des motifs de mouvements représentés par des caractéristiques invariantes à l'échelle. Cette extraction est faite par le détecteur d'Haris 3D. Pour classer une action, un classifieur SVM est utilisé. Jhuang *et al.* [34] suggèrent la construction d'un modèle multicouche basé sur l'extraction de motifs spatio-temporels par l'utilisation des filtres de Gabor. Un classifieur SVM est utilisé pour reconnaitre les différentes actions. Tseng *et al.* [35] ont basé leur technique sur la construction d'un graphe d'actions spatio-temporelles qui relie les silhouettes d'une même action, de dimension réduite dans l'espace des silhouettes. La reconnaissance est faite par le classifieur $K$-ppv. Bobik et Davis [11] ont proposé, quand à eux, une méthode basée sur la construction d'un modèle d'apparence. Le système de reconnaissance utilise la distance de Mahalanobis afin de mesurer la similarité entre les descripteurs des moments de Hue 2D. Ces descripteurs sont fournis par les images d'énergie du mouvement (MEI), et celles de l'historique du mouvement (MHI).

Saad *et al.* [4] se sont basés sur une méthode d'apprentissage multi-instances (MIL) qui emploie des caractéristiques cinématiques extraites à partir du flux optique pour classer les différentes actions.

Dans ce mémoire, nous présentons un système automatique de reconnaissance des actions humaines à la fois simple et efficace basé sur la génération d'une série de prototypes, selon les différents points de vue du cube de données de la séquence d'images, et la fusion de ceux-ci. Ce mémoire est organisé comme suit; après avoir fait l'état de l'art dans le chapitre I, le chapitre II présente une introduction à la vidéo avec des prétraitements utiles au traitement de la vidéo. Les chapitres III, IV, V sont axés sur les différentes étapes nécessaires à un système de reconnaissance des actions humaines. Nous présenterons aussi, dans ce chapitre, les différentes approches existantes dans la littérature pour les trois étapes du système; *i.e.*,, la détection d'objet, la réduction de dimensionnalité et la classification. Dans le chapitre VI, nous expliquons les différentes étapes de la méthode proposée avec l'expérimentation de la méthode. Ensuite on compare l'efficacité de notre méthode avec d'autres méthodes. Enfin nous terminons ce mémoire par une conclusion générale.

Chapitre 2

# CONCEPTS ÉLÉMENTAIRES DE LA VIDÉO

## *2.1 Introduction*

Le signal vidéo [2,3] est le signal qui permet de transporter une séquence d'images de la source à un dispositif d'affichage sous forme électrique. Il existe deux modes de vidéo, la vidéo analogique et la vidéo numérique. Selon la façon dont les signaux sont traités on peut distinguer les deux modes:

1. La vidéo analogique[2] décrit le signal analogique comme un signal électrique dont l'intensité varie dans le temps de façon continue. La qualité du signal final dans ce mode est plus faible car le bruit rajouté au signal lors son traitement altère sa qualité.

2. La vidéo numérique est un signal[4] qui porte une information représentée par une suite de valeurs minimales ou maximales correspondant respectivement au 0 et au 1. L'un des facteurs qui avantage le signal numérique par rapport au signal analogique est la facilité de distinguer l'information émise du bruit.

## *2.2 Les paramètres clés d'une vidéo*

Le stockage et la diffusion d'une vidéo exigent un espace volumineux et un taux de transfert plus élevé. Le contrôle de qualité, et de taille d'une séquence vidéo est déterminé par deux paramètres clés[3], le nombre d'images par seconde et la résolution.

[2] www.docam.ca/en/component/content/article/307-221-mode-analogique-support-video.html

[3] Le groupe Adobe Dynamic Media: Initiation à la vidéo numérique, juin 2000

[4] wiki.univ-paris5.fr/wiki/

Trouver le compromis entre ces paramètres et les limitations imposées par la technologie permet d'obtenir une qualité de vidéo optimale.

### *2.2.1 Le nombre d'images par seconde*

Le système visuel humain (SVH) est un système qui joue le rôle de percevoir, et d'interpréter les images du monde réel. La sensibilité du système SVH à la variation rapide d'une succession d'images permet à l'oeil de percevoir un phénomène d'animation[3]. Pour créer ce phénomène dans la bande vidéo, un nombre d'images par seconde est exigé, en général 25 ou 30 images par seconde.

### *2.2.2 La résolution*

Ce terme désigne que la quantité de l'information est limitée dans l'image. Autrement, c'est le nombre de pixels qui peuvent être affichés par un dispositif d'affichage[3].

## *2.3 Formation optique de la vidéo numérique*

### *2.3.1 Les espaces de couleurs*

La nécessité de représenter l'image de différentes manières dans un espace de couleur a donné naissance aux nombreux espaces de couleurs avec des propriétés différentes. Une grande variété de ces espaces a été appliquée fréquemment dans plusieurs applications du domaine de la vision par ordinateur, comme par exemple le problème de la détection de la peau [28]. Les espaces les plus populaires sont:

### *2.3.2 L'espace de couleur RVB*

RVB est un espace qui définit la couleur à partir trois couleurs primaires rouge, vert, bleu. L'ajout maximal de ses trois composantes donne une couleur blanche et leur absence donne une couleur noire. Les trois composantes RVB sont très dépendantes

les unes les autres (cf.Fig.2.1)

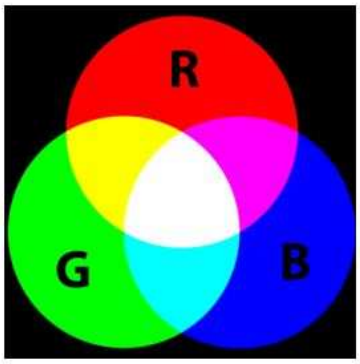



**Figure 2.1.** *L'espace de couleur RVB* [5]

### *2.3.3 L'espace de couleur HSV*

C'est un espace colorimétrique[5,6], défini en fonction de ses trois composantes hue, saturation et value. La composante hue code la teinte suivant l'angle qui lui correspond sur le cercle des couleurs. La saturation est l'intensité de la couleur qui varie entre 0 et 100 pourcents. La brillance de la couleur est donnée par la valeur de value (la composante value). La brillance correspond au noir pour une valeur de value égale à 0 (cf. Fig.2.2).

### *2.3.4 L'espace de couleur Yuv*

Le signal dans cet espace[7] est un signal codé en RVB. L'information de la luminance Y est le produit de la somme pondérée de trois composantes rouge, vert et bleu. L'information de chrominance formée par u et v est obtenue en soustrayant l'information de la luminance Y de deux composantes rouge, et vert.

[5] en.wikipedia.org/wiki/Colorspace

[6] elle.epfl.ch/net/gimp/BOOK/fr/Grokking-the-GIMP-v1.0/images/img150.png

[7] en.wikipedia.org/wiki/YUV

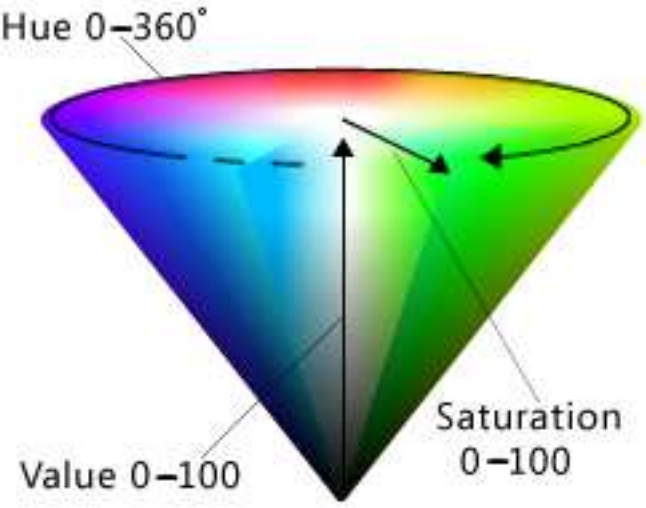



**Figure 2.2.** *L'espace de couleur HSV* [6]

## *2.4 Processus de formation de la vidéo numérique*

D'une manière générale[3], le signal vidéo est capté par une caméra vidéo dotée par un dispositif à transfert de charge (CCD). À l'aide de l'optique de la caméra, un signal lumineux est acquis. Ce dernier frappe une surface photosensible qui libère des charges électriques formant un signal électrique d'intensité variable. La couleur est formée par un filtre de Bayer constitué d'un ensemble de photosites intégrés dans le capteur CCD (cf. Fig. 2.3)[8]. Le signal électrique est numérisé par une opération d'échantillonnage suivie d'une opération de quantification afin de ressortir un signal numérique codé dans un format binaire compréhensible par la machine. Ensuite le signal numérisé est reconverti en mode analogique qui permettra de visualiser l'information par un écran d'affichage.

[8] http://commons.wikimedia.org/wiki/File:CCBayerFilter.png

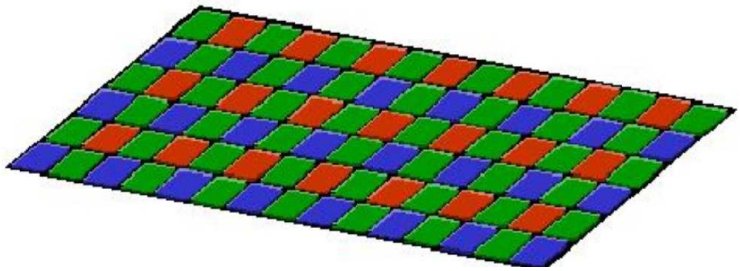


**Figure 2.3.** *Filtre de Bayer* [8]

## 2.5 Prétraitements de la vidéo

L'image acquise par un tel capteur est fortement altérée par différents sources de bruit. La dégradation diffère d'un environnement d'acquisition à l'autre. La restauration est une étape préliminaire qui vise à restaurer ou à améliorer l'image altérée afin d'obtenir une image proche de l'image réelle, et d'éviter tous les effets indésirables qui affectent négativement les différents algorithmes de vision. Pour une meilleure restauration, il est important de déterminer le type de dégradation, par exemple une image dégradée par un flou[9] qui détruit les détails de l'image. Ce bruit est dû à une surface qui diffuse la lumière, ou parfois causé par un objet en mouvement. Un autre type de dégradation peut être causé par l'objectif d'un capteur, comme la distorsion géométrique[10] qui rend les lignes droites courbes. Le bruit impulsionnel est aussi une autre source de bruit dû à la numérisation du signal [11]. Ce bruit prend deux valeurs très proches de 0 ou 255 (cf.Fig.2.4)[9,10,11]. Afin de réduire les effets d'un tel bruit des prétraitements sont envisagés, citons:

[9] www.tsi.telecom-paristech.fr/pages/enseignement/ressources/beti/svd/index.htm

[10] fr.wikipedia.org/wiki/Distorsionoptique

[11] http://www.anirudh.net/courses/cse585/project1/lennaspnoise10.html

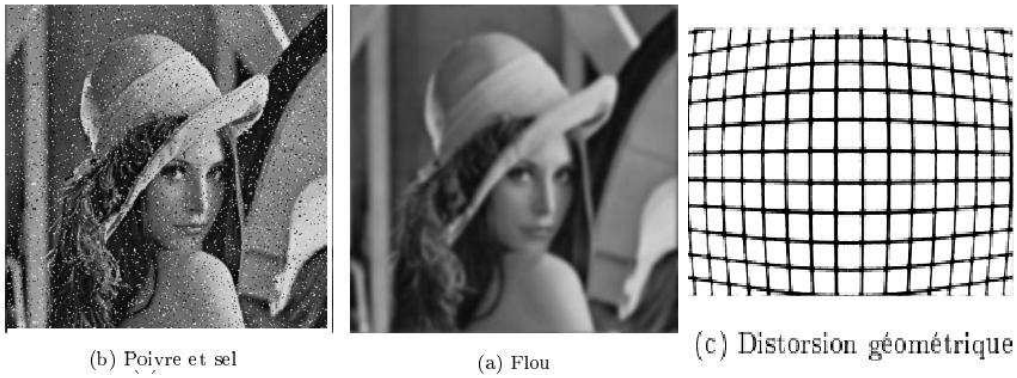



**Figure 2.4.** *Type de bruit ou de dégradation*

### *2.5.1 Filtrage passe haut*

Le filtrage passe haut consiste à accentuer les contours et les détails de l'image. Il est utile pour une image bruitée par une dégradation du type flou. Ce type de filtrage préserve les détails de l'image, mais amplifie le bruit (cf. Fig.2.5)[12].

### *2.5.2 Filtrage passe bas*

Contrairement à un filtrage passe haut, le filtrage passe bas vise à diminuer le bruit uniforme, nous atténue les détails de l'image (cf. Fig.2.5)[12].

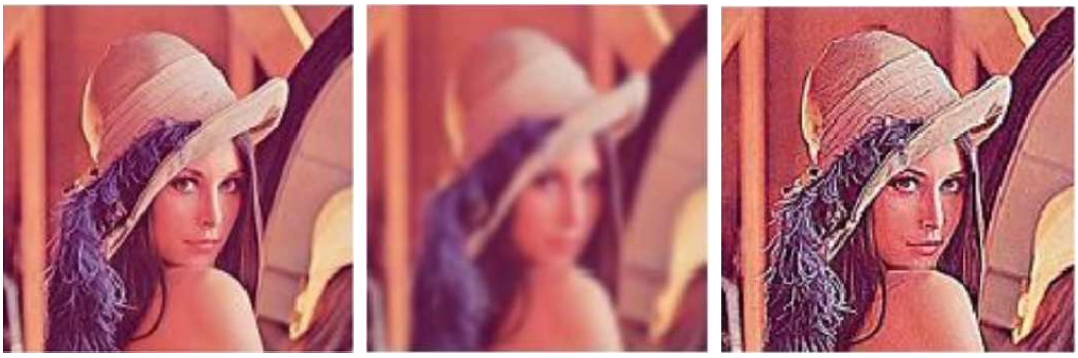


**Figure 2.5.** *Filtrage (Image originale, filtrée passe-bas et passe-haut)*

[12] www.cosy.sbg.ac.at/ pmeerw/Watermarking/lena.html

Les deux techniques de filtrage sont basées sur une opération de convolution. Cette opération est faite entre une image et un masque de convolution.

## 2.6 Compression de la vidéo numérique

Enregistrer, traiter, distribuer une vidéo numérique dans sa taille originale est couteux en terme de matériel et de temps. Pour cette raison, les applications de la vidéo numérique font souvent appel à des techniques de compression sans ou avec perte (dans ce dernier cas, reposant sur l'élimination des très hautes fréquences de l'image correspondant aux détails très fins de l'image [3]).

## 2.7 Conclusion

On a présenté brièvement dans ce chapitre des prétraitements utiles au traitement de la vidéo. On présente dans les chapitres suivants les différentes étapes d'un système de reconnaissance des actions humaines.

Chapitre 3

# LA DÉTECTION D'OBJET MOBILE

## *3.1 Introduction*

Isoler les objets mobiles de l'arrière-plan d'une séquence vidéo est une technique de segmentation nécessaire et souvent très utilisée en vision artificielle. La plupart des algorithmes de vision considèrent cette étape comme une étape de prétraitement nécessaire qui a pour but de réduire l'espace de recherche, et d'améliorer la performance en terme de coût calculatoire d'une application (possiblement de reconnaissance ou de tracking ou de détection) de plus haut niveau d'abstraction [33]. Cette étape est nécessaire, plus précisément, pour le suivi d'objet [65], la reconnaissance d'actions humaines [10, 11, 35] [50], la vidéosurveillance [29], la détection de chute [74]. On trouve un ensemble de méthodes de détection d'objet basées sur la technique de soustraction de fond. La détection d'objet par cette technique se fait par une opération de soustraction de deux images, l'image courante, et l'image qui représente la ou les parties statiques de la scène (cf. Fig. 3.1)[13]. L'un des problèmes majeurs de cette technique est la manière dont on peut obtenir automatiquement un arrière-plan de la scène statique qui soit le plus robuste aux changements de l'éclairage, aux ombres, et au bruit présent dans la séquence vidéo. Il existe différents algorithmes dans la littérature qui ont été conçus pour modéliser tout ce qui est statique, d'éliminer les ombres, et de récompenser l'évolution de l'arrière-plan. La performance de ces algorithmes, pour estimer un arrière-plan plus robuste, est variable d'un algorithme à l'autre [2, 29, 45, 50, 65].

[13] http://www.wisdom.weizmann.ac.il/ vision/SpaceTimeActions.html

**Figure 3.1.** *Technique de soustraction de fond*

## 3.2 Les techniques de modélisation de l'arrière-plan

### 3.2.1 Le filtre médian

Le filtre de médian temporel a été appliquée, par exemple, dans les travaux de [35] afin de construire une image d'arrière-plan. Le modèle de l'arrière-plan est estimé pour chaque pixel $p_t(x, y)$ de l'image $I$ à un instant $t$ à l'aide d'un filtre médian (temporel) de taille $N$. La valeur du pixel de cet arrière-plan est donnée par l'équation suivante:

$$p_t(x, y) = \text{MÉDIAN}\Big(p_{t-1}(x, y), ..., p_{t-N}(x, y)\Big) \tag{3.1}$$

La soustraction entre le modèle d'arrière-plan (préalablement estimé) et l'image originale permette de détecter et segmenter les objets en mouvement dans la scène. L'avantage de cette technique est qu'elle est simple et rapide. L'un des inconvénients de cette approche est le nombre d'images $N$ nécessaires que l'on doit à tout instant stocker dans une mémoire-tampon.

### 3.2.2 Modélisation de l'arrière plan par l'ACP

Cette technique a été décrite dans les travaux de Nuria *et al.* [50]. Dans ces travaux, les auteurs cherchent à construire un modèle d'arrière-plan qui permet de décrire la variation de l'apparence dans un espace représentatif multidimensionnel plus réduit grâce à une technique de réduction de dimensionnalité. Les auteurs

suggèrent d'appliquer l'analyse en composante principale (ACP) sur $N$ images de la séquence vidéo afin d'extraire une base de vecteurs propres (*Eigenbackgrounds*). Ensuite de conserver les deux premiers vecteurs propres qui expliquent le mieux la variance de ces $N$ frames, et qui permettent de réduire la dimension de l'espace de représentation. Une fois le modèle construit, chaque nouvelle image $I$ est projetée dans l'espace de représentation afin de modéliser les parties statiques de la scène. Les objets sont détectés par le calcul de la différence entre l'image d'entrée $I$ et l'image $I'$. L'image $I'$ est l'image $I$ reconstruite à partir de sa projection. Les étapes principales de l'approche communément appelée *eigenbackgrounds* sont les suivantes:

- On arrange les $N$ images sous la forme d'une matrice-colonne $A$
- On trouve la matrice de variance covariance $C = AA^t$
- La matrice $C$ est diagonalisée afin d'obtenir une base de vecteurs propres $\phi$ et des valeurs propres $\lambda$
- Seulement les deux premiers vecteurs propres sont retenus
- On projette chaque nouvelle image $I$ dans cet espace de dimensions deux, ensuite on reconstruit l'image
- On fait la différence entre l'image d'entrée et l'image reconstruite pour détecter l'objet

Notons que cette approche est moins robuste si l'arrière-plan est évolutif.

### *3.2.3 Méthode de distribution gaussienne simple*

Vue que la scène n'est pas statique, McKenna *et al.* [45] ont proposé une méthode adaptative au changement de l'environnement. La méthode vise à compenser l'évolution de l'arrière-plan à l'aide d'une mise à jour qui se fait par les pixels statiques de l'image

suivante par rapport à l'image précédente de chaque image de la séquence. Cette mise à jour permet de capturer le changement entre deux images successives. En tenant compte d'un bruit gaussien avec une variance $\sigma_{bruit}$, dû à l'appareil d'acquisition et le changement d'éclairage, une unique gaussienne paramétrée par deux paramètres $\sigma_t$ et de moyenne $\mu_t$, est utilisée afin de modéliser les valeurs de chaque pixel $p_t(x, y)$ dans le temps:

$$N(p_t; \mu_t, \Sigma_t) = \frac{1}{(2\pi)^{\frac{d}{2}} \mid \Sigma \mid^{\frac{1}{2}}} \exp\Big\{-\frac{1}{2}(p_t - \mu_t)^T \Sigma_t^{-1}(p_t - \mu_t)\Big\} \tag{3.2}$$

où $\Sigma = I\sigma$, $I$ est la matrice identité. La mise à jour de ces paramètres se fait par les relations suivantes:

$$u_{t+1} = \alpha\, u_t + (1 - \alpha)\, p_{t+1} \tag{3.3}$$

$$\sigma_{t+1} = \alpha \left(\sigma_t + (\mu_{t+1} - u_{t+1})^2\right) + (1 - \alpha)(p_{t+1} - u_{t+1})^2 \tag{3.4}$$

avec $\alpha$, un paramètre qui contrôle le taux d'adaptation et $p_{t+1}$ est la valeur d'un pixel. Les pixels sont classés en se basant sur le critère suivant; Si $|p_t(x, y) - u_t| \geq 3$ $\max(\sigma_{bruit}, \sigma_t)$ alors $p_t(x, y)$ est un pixel de mouvement sinon le pixel est classé comme un pixel d'arrière-plan. La modélisation d'un pixel par une seule gaussienne est adaptative seulement au changement de la luminosité, et ne prend pas en compte le changement de la scène, par exemple le cas où des petits objets en mouvements apparaissent dans le temps.

### *3.2.4 La méthode de mixture gaussienne*

Dans une application de suivi d'objet en temps réel, Stauffer *et al.* [65] ont proposé une méthode adaptative basée sur un modèle défini par un mélange de gaussiennes offrant la possibilité de mettre à jour le modèle sans garder en mémoire un grand nombre d'images de la séquence dans une mémoire-tampon [33]. L'approche est constituée de deux étapes: la première étape cherche à modéliser les différentes valeurs de chaque pixel par plusieurs gaussiennes afin de tenir compte, à la fois, du bruit

gaussien, le changement graduel de la luminosité et l'évolution de l'arrière-plan. La deuxième étape de cette méthode vise à décider de la classe de chaque pixel de la séquence d'images, soit comme étant un pixel de l'arrière-plan, ou comme un pixel de mouvement. En considérant que les observations correspondant à un pixel qui varie dans le temps sont considérées comme un processus $X_t$, le processus de segmentation est le suivant:

- Le processus $X$ est initialisé par les valeurs des pixels récentes: $X_t$=$\{x_1, ..., x_t\}$ avec $x_i = [r_i, v_i, b_i]$.

- Chaque pixel de l'image de référence est modélisé par un mélange de $k$ densités de probabilités. La probabilité d'appartenance d'un pixel à l'arrière-plan est donnée par:

$$P(X_t) = \sum_{i=1}^{k} w_{i,t}\, \eta(x_t; \mu_{i,t}, \Sigma_{i,t}) \tag{3.5}$$

  où,

  $k$: représente le nombre de gaussiennes utilisées dans le mélange (de 3 à 5)

  $w_{i,t}$: est un poids associé à chaque gaussienne représentant la proportion des données utilisées dans le calcul de la gaussienne à un instant $t$

  $\eta$: est une fonction gaussienne multidimensionnelle définie par un vecteur de moyenne $\mu_t$ et une matrice de covariance $\Sigma_t$:

$$\eta(p_t; \Sigma_t, \mu_t) = \frac{1}{(2\pi)^{\frac{d}{2}} \mid \Sigma \mid^{\frac{1}{2}}} \exp\Big\{-\frac{1}{2}(p_t - \mu_t)^T \Sigma_t^{-1} (p_t - \mu_t)\Big\} \tag{3.6}$$

  où $d$ représente la dimension de l'espace et $\Sigma = I\sigma$, $I$ la matrice identité. Pour des raisons de calcul, Stauffer *et al.* supposent que les trois composantes rouges, vertes, bleues sont indépendantes.

- Les gaussiennes d'arrière-plan correspondent aux gaussiennes de grande persistance et de faible variance car les objets statiques produisent une faible variance et une répétition forte des données pour les distributions de l'arrière-plan.

- Le modèle est mis à jour au fur et à mesure en vérifiant que chaque nouvelle observation $x_t$ corresponde aux gaussiennes de l'arrière-plan.

- Si oui, la mise à jour des paramètres du modèle se fait par les équations suivantes:

$$w_{i,t} = (1-\alpha)\, w_{i,t-1} + \alpha\, v(w_i, x_t) \tag{3.7}$$

$$\mu_{i,t} = (1-\rho)\, \mu_{i,t-1} + \rho x_t \tag{3.8}$$

$$\sigma_{i,t} = (1-\rho)\, \sigma_{i,t-1} + \rho\, (x_t - \mu_{i,t})^T (x_t - \mu_{i,t}) \tag{3.9}$$

  avec $\rho = \alpha\, \eta(x_t; \mu_{i,t}, \Sigma_{i,t})$, $\alpha$ est le taux d'adaptation (de 0 à 1) et v($w_i$,$x_t$)=1 si $\eta(\Sigma_t, \mu_t)$ est la gaussienne correspondante à $x_t$.

- Sinon le pixel est considéré comme un pixel de mouvement.

L'un des inconvénients de cette approche est l'initialisation du paramètre $k$ qui représente le nombre de gaussiennes. Cette initialisation est faite d'une façon manuelle, et demeure constante pour tous les pixels au fur et à mesure de l'acquisition.

### *3.2.5 Modèle non paramétrique*

Modéliser l'arrière-plan par un modèle paramétrique [33, 65] nécessite d'estimer au préalable les paramètres de ce modèle (et ensuite de les réactualiser). Une autre stratégie consiste à modéliser l'arrière-plan par un modèle non paramétrique à base d'un noyau (*kernel*). Dans cette approche non paramétrique Elgammal *et al.* [2] ont proposé de représenter le modèle de l'arrière-plan par une fonction de densité de probabilité (PDF) définie à l'aide d'un noyau modélisant l'historique récent des différentes valeurs d'un pixel. Cette PDF, qui modélise non paramétriquement la probabilité d'observer la valeur d'un pixel au temps $t$, est définie par l'équation suivante:

$$Pr(X_t) = \frac{1}{N} \sum_{i=1}^{N} \phi(x_t - x_i) \tag{3.10}$$

où $\phi$ représente un noyau.

En utilisant un noyau gaussien qui suit la loi normal N(0,$\Sigma$), avec une moyenne nulle pour tenir compte du bruit gaussien qui est centré à 0, et un paramètre $\Sigma$ qui définit la largeur de la gaussienne, la fonction de densité est égale:

$$Pr(X_t) = \sum_{i=1}^{N} \frac{1}{(2\pi)^{\frac{d}{2}} \mid \Sigma \mid^{\frac{1}{2}}} \exp\Big\{-\frac{1}{2}(x_t - x_i)^T \Sigma_t^{-1}(x_t - x_i)\Big\} \tag{3.11}$$

Pour chaque pixel, la densité de probabilité est estimée Pr($X_t$). Le pixel $x_t$ est considéré comme un pixel d'arrière-plan s'il vérifie le critère suivant: Pr($X_t$) $\geq T$, avec $T$ un seuil à fixer manuellement, et $d$ représente la dimension de l'espace. Cette approche est robuste si l'arrière-plan est dynamique, *i.e.*, robuste à la présence de petits mouvements (feuillage d'un arbre). L'astuce du noyau rend l'algorithme coûteux en matière de temps de calcul.

### *3.3 Binarisation et post-traitement des images binaires*

Binariser une image après une opération de segmentation permet de coder chaque pixel de l'image par un seul bit soit 0 ou 1. Cela signifie que les pixels du fond apparaissent en noir et les autres pixels qui représentent le bruit et les objets apparaissent en blanc. Cette représentation est largement utilisée par plusieurs systèmes de vision, par exemple un système de reconnaissance des chiffres [19]. La rapidité d'exécution et l'espace de stockage de ces images binaires avantagent ce type de codage. Différents posts-traitements[14] peuvent être envisagés sur ce type d'image afin de corriger les résultats obtenus par une opération de soustraction de fond en basant sur des opérations fondées sur la morphologie mathématique (cf. Fig. 3.2).

[14] www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic4.htm

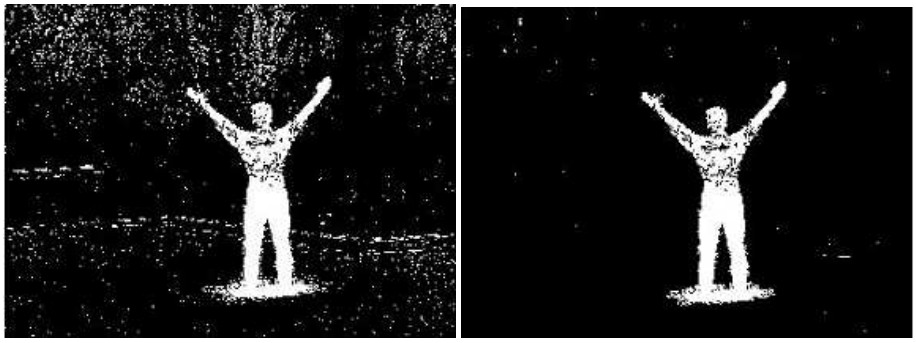


**Figure 3.2.** *Correction du résultat après une opération de soustraction de fond [2].*

## *3.4 Conclusion*

Dans ce chapitre, on a représenté différentes techniques de soustraction de fond qui jouent un rôle-clé dans un système de reconnaissance d'action humaine. Une fois la personne détachée de l'arrière-plan, la taille du vecteur qui représente le contenu de la personne dans chaque image de la séquence est égale à $L \times H$ pixels (*i.e.*, longueur $\times$ largeur de l'image), ce qui rend ensuite son traitement coûteux. Pour cette raison, l'étape qui suit, dans un système de reconnaissance d'actions, vise souvent à réduire la dimensionnalité de ces données. Le chapitre suivant présentera quelques techniques permettant de le faire.

Chapitre 4

# RÉDUCTION DE DIMENSIONNALITÉ

## *4.1 Introduction*

Le terme fléau de la dimensionnalité ou malédiction de la dimensionnalité a été introduit par Richard E. Bellman en 1961 [55]. Ce terme désigne la difficulté à concevoir des modèles (probabilistes) de classification efficace, alors que le nombre de combinaisons possibles croit de façon exponentielle avec le nombre de variables ou la dimension des données. La fouille de données et l'apprentissage machine sont des domaines qui exploitent une grande quantité de données afin d'extraire un savoir, ou interpréter une information selon l'objectif. Plus on dispose d'une grande quantité de données, plus on conserve des informations qui rendent le système performant. L'exploitation brute de ces données en haute dimension par certains algorithmes de vision peut dégrader leurs performances (par exemple, dans un problème de classification, l'algorithme des $K$-plus proches voisins ne fonctionne pas bien si la dimension de l'espace est grande). La corrélation des données et l'information non pertinente offrent la possibilité de réduire la dimension de l'espace. La réduction de dimensionnalité est l'une des techniques utilisées pour traiter le problème du fléau de la dimensionnalité.

## *4.2 Pourquoi la réduction de dimensionnalité*

En plus d'éliminer l'information non utile, le bruit, et l'information redondante, la réduction de dimensionnalité [20, 55] permet aussi une meilleure représentation de l'information (afin, par exemple, de représenter les données par un graphique qui facilite l'analyse et l'interprétation des données). Finalement, la réduction de dimen-

sionnalité permet d'extraire un vecteur de caractéristiques de dimension réduite afin de simplifier le problème de classification, et de compresser les données afin de réduire l'espace de stockage.

## 4.3 Quelques applications de la réduction de dimensionnalité

La réduction de dimensionnalité[15] est utile dans plusieurs disciplines comme la reconnaissance des formes, la théorie de l'information, le fouille de données, l'apprentissage machine. Nous citons quelques applications qui emploient cette technique:

### *4.3.1 La reconnaissance du visage*

La reconnaissance du visage est un système qui permet d'identifier les personnes. On trouve ce système dans les applications biométriques, et dans les applications de sécurité. Dans les travaux présentés en [51], une méthode de reconnaissance du visage est basée sur l'analyse en composante principale (PCA). La PCA permet de décrire la variabilité entre les images de la base. Cette technique cherche à représenter une base d'images constituée de 48 visages avec une taille $L \times H$ pixels sous la forme d'une matrice de variance-covariance. Ensuite, elle consiste à construire un sous-espace de visages (*Eigenface*) à partir des 48 vecteurs propres les plus significatifs aux images de départ après la décomposition. Pour reconnaitre un nouveau visage, le calcul de la distance euclidienne se fait entre le visage d'entrée et les 48 vecteurs propres existants dans le nouvel espace (cf. Fig. 4.1).

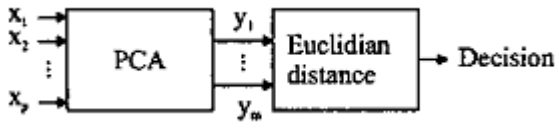



**Figure 4.1.** *Système de reconnaissance du visage [51]*

[15] en.wikipedia.org/wiki/Curseofdimensionality

### 4.3.2 Recherche d'image par le contenu

L'utilisation d'une requête sous forme d'image est aujourd'hui de plus en plus utilisée pour la recherche de similarité dans les grandes bases de données multimédias ou sur le World web wide. Suivant ce principe, le système Chromatik[16] est un système de recherche d'image par le contenu qui utilise à la fois les mots-clés mais aussi l'information visuelle pour effectuer sa recherche. Il existe trois façons de faire une recherche dans une base de données: soit une recherche par mots-clés, soit une recherche par esquisse, soit une recherche par image. La recherche par mot-clé est inefficace si le mot n'exprime pas le sens de l'image. La recherche par image utilise des caractéristiques représentatives de l'image comme la couleur, la texture, la forme. Caractériser une image par un descripteur en haute dimension affecte le temps de réponse du système de recherche. Afin d'améliorer le système et l'efficacité d'accès aux images désirée, Beatty *et al.* [7] suggèrent de décrire l'apparence de l'image par un vecteur de caractéristiques de dimension 60. Ensuite une version de PCA non linéaire permet d'extraire un vecteur de caractéristiques de dimension inférieure (de dimension 60 à 6). Différentes techniques de réduction de dimensionnalité sont discutées par Beatty *et al.* afin d'étudier l'efficacité de chaque technique sur le résultat du système d'indexation. Daniel *et al.* [66] utilisent une autre technique qui est basée sur la sélection, et pas sur l'extraction de caractéristiques. L'amélioration de l'efficacité du système d'indexation est faite à l'aide des caractéristiques sélectionnées [66].

### 4.3.3 La compression d'image

Compresser une image revient à réduire l'information redondante sans trop affecter la qualité de l'image. Dans ce contexte, Vilas *et al.* [68] utilisent une technique de réduction de dimensionnalité (PCA) pour coder les informations pertinentes de

[16] chromatik.labs.exalead.com/home

l'image.

## 4.4 Les techniques de réduction de la dimensionnalité

Un processus d'extraction de caractéristiques est un processus qui consiste à décrire une image, une séquence, ou un objet de l'image par plusieurs attributs dans un espace multidimensionnel. Ce processus d'extraction fournit la donnée d'entrée à un système d'apprentissage ou de reconnaissance. Par exemple, dans le cas d'une image, il existe deux manières pour la représenter sous la forme d'un vecteur d'attributs: soit par des caractéristiques locales, soit par des caractéristiques globales. Les caractéristiques locales sont extraites à partir d'une région d'intérêt identifiée auparavant en utilisant des descripteurs possédants des propriétés d'invariance, par exemple le descripteur SURF [6] ou SIFT [36]. L'avantage de ses caractéristiques locales par rapport aux caractéristiques globales est qu'elles sont plus saillantes, et généralement invariantes aux différentes transformations géométriques. L'extraction des caractéristiques significatives permet de rendre le traitement ultérieur plus robuste. Une façon d'extraire des caractéristiques pertinentes consiste à utiliser une technique de réduction de dimensionnalité directement sur l'ensemble des pixels de l'image, ce qui permettra ensuite de caractériser cette image par un vecteur de faible dimension. La réduction est faite par la transformation des caractéristiques d'un espace de grandes dimensions à un autre espace de dimension inférieure en respectant certains critères. Les techniques de réduction de la dimensionnalité sont divisées en deux familles [55]:

1. La sélection de caractéristiques

2. L'extraction de caractéristiques

La différence entre les deux familles réside dans la manière dont on peut trouver la nouvelle représentation de données à partir de la représentation de départ.

### 4.4.1 Formulation du problème de la réduction de dimensionnalité

On considère une séquence d'images comme un ensemble de points représentés dans un espace multidimensionnel de dimension $d \times n$ ($d$ étant le nombre de pixels de l'image), l'objectif est de passer d'un espace de dimension $d$ à un sous-espace de dimension $k$ à l'aide d'une fonction de coût $\phi$ [20]:

$$min \sum_{i,j=1}^{n} (\delta_{ij}^{(d)} - \delta_{ij}^{(k)})^2,$$

avec $\delta_{ij}^{(d)} = \|x_i - x_j\|$ et $\delta_{ij}^{(k)} = \|y_i - y_j\|$

La fonction de coût qui donne une meilleure représentation est celle qui préserve les propriétés de l'espace de départ après la transformation de tous les points. En d'autres termes, les points les plus proches dans l'espace du départ doivent demeurer plus proches dans le nouvel espace.

### 4.4.2 L'extraction de caractéristiques

L'extraction de caractéristiques est un processus qui consiste à combiner toutes les caractéristiques par une fonction linéaire, ou non linéaire afin de ressortir des nouveaux caractéristiques projetés dans un nouvel espace multidimensionnel de dimension réduite, tout en essayant de garder uniquement l'information pertinente (cf. Fig. 4.2 [17]. Dans l'exemple artificiel swist roll [59], on trouve que les résultats obtenus par les techniques d'extraction linéaire comme PCA classique ne sont pas meilleurs car les données dans l'espace de grande dimension reposent sur une variété courbée. Les techniques non linéaires permettent de résoudre ce problème.

[17] S. Guérif: Réduction de dimension en Apprentissage Numérique Non Supervisé, thèse, 2006

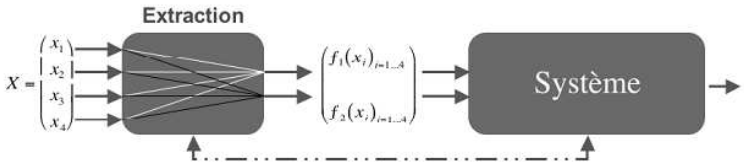



**Figure 4.2.** *L'extraction de caractéristiques*

### *4.4.3 Échelonnage Multidimensionnel*

L'échelonnage multidimensionnel (ou MDS, *Multidimensional Scaling*) est une technique de réduction de dimensionnalité qui a pour but de construire une représentation en faible dimension à partir d'un ensemble de points qui se trouve en haute dimension, tout en essayant de préserver les paires de distances entre les points dans cette nouvelle représentation [12, 18, 24]. De ce fait, la représentation en faible dimension peut être représentée par une fonction de coût, définie comme étant une mesure d'erreur (avec une distance euclidienne) entre les paires dans l'espace de grande dimension et l'espace de faible dimension [70]. L'algorithme classique MDS [15] est basé sur la construction de la matrice de Gram. La matrice de Gram est obtenue par le produit scalaire entre les paires de vecteurs de distances, suivie par une opération de double centrage de la matrice. La coordonnée réduite de chaque point s'obtient par la décomposition spectrale de la matrice de Gram en vecteurs et en valeurs propres. Cette décomposition permet de mesurer la contribution de chaque dimension au produit scalaire. L'un des inconvénients de cet algorithme est sa complexité algorithmique lorsqu'on dispose d'une grande base de données. D'autres variantes de l'algorithme MDS se trouvent dans [53], et peuvent être appliquées sur une grande base de données.

### 4.4.4 Isomap

L'isomap est une variante non linéaire qui a été proposée par Tenenbaum *et al.* [64, 67]. Cette technique a été appliquée dans plusieurs problèmes, par exemple, le problème de la reconnaissance de chiffres manuscrits. Dans ce problème [67], 1000 chiffres manuscrits du chiffre 2 sont extraits de la base de données MINIST. Chaque chiffre est représenté par un vecteur de taille $64 \times 64$ pixels. L'algorithme isomap [67] cherche à projeter ces images dans un espace de caractéristiques significatives de deux dimensions comme illustré dans la figure (4.3), tout en essayant de préserver les distances géodésiques.

Dans un problème de classification du visage [72], on trouve une version améliorée de l'algorithme isomap. Dans cette version, la projection du vecteur de distance géodésique est faite par l'analyse discriminante linéaire qui a l'avantage de maximiser les distances entre les centres des classes, ce qui permet de différencier entre l'algorithme original et la version améliorée. L'algorithme de l'isomap [67] [64] est basé sur l'algorithme classique MDS qui vise à préserver les distances géodésiques entre les paires de points afin d'obtenir une représentation non linéaire des données dans un espace Euclidien de dimension faible. La distance géodésique est estimée entre chaque paire de points par la somme des distances entre les points intermédiaires qui relient chaque paire en suivant le chemin le plus court dans un graphe de voisinage à l'aide de l'algorithme de Djikstra. Le graphe de voisinage est constitué à partir d'un ensemble de points répartis dans l'espace original dont chaque point est connecté à ses $k$ plus proches voisins. Le choix d'une mauvaise valeur de voisinage peut créer des connexions erronées dans le graphe. Ces connexions peuvent affecter la performance de l'algorithme.

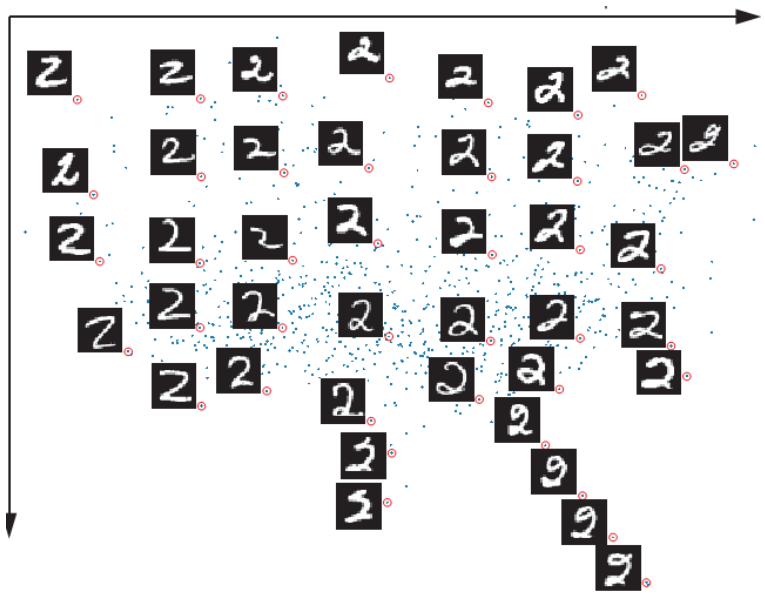


**Figure 4.3.** *Réprésentation des images de taille* $64 \times 64$ *en 2D par l'algorithme isomap [67]*

### *4.4.5 Locally Linear Embedding*

LLE est un algorithme qui tient compte du voisinage des points pour décrire les propriétés locales de chaque point de l'ensemble de données. Dans [30], les auteurs proposent l'algorithme LLE pour sélectionner les visages les plus représentatifs à partir d'une séquence de visages et les projeter dans un espace à deux dimensions afin de construire un modèle d'apparence représentatif qui facilite la reconnaissance d'un nouveau visage (cf. Fig. 4.4). Le principe de cet algorithme [59, 61] est de décrire les propriétés locales de chaque point $x_i$ par une combinaison linéaire $W_i$ qui reflète les propriétés locales de ses $k$ plus proches voisins $x_{ij}$. Trouver ces combinaisons linéaires revient à minimiser l'erreur quadratique (Eq. (4.1)) sous la contrainte (Eq. (4.2)). Ces combinaisons permettent de construire une matrice avec des poids $W_i$ les

plus invariants à la translation, la rotation, et la mise à l'échelle.

$$\varepsilon(W) = \sum_i \Big( x_i - \sum_j W_{ij} x_j \Big)^2 \tag{4.1}$$

$$\sum_{j \in k(i)} W_{ij} = 1 \tag{4.2}$$

La réduction de dimensionnalité se fait par la diagonalisation de cette matrice de poids en retenant les vecteurs qui correspondent aux petites valeurs propres non nuls.

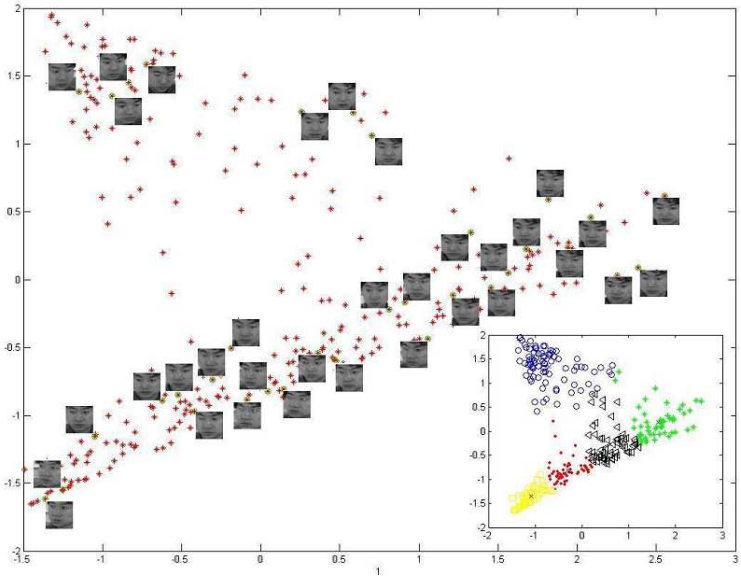


**Figure 4.4.** *Projection des images selectionnées en 2D par LLE [30]*

### *4.4.6 ACP à noyau*

L'analyse en composante principale à noyau [9] est une combinaison d'une transformation non linéaire avec l'algorithme classique ACP. Cette méthode est basée sur l'astuce de noyau qui permet de rendre l'algorithme classique ACP non linéaire. Dans une application de reconnaissance des paroles [40], un noyau gaussien a été appliquée afin d'extraire les relations non linéaires des vecteurs d'entrées. L'idée de cet algorithme [9] est de générer un espace de caractéristiques de haute dimension $F$ par la projection implicite des données dans le nouvel espace $F$ à l'aide d'un noyau $k$,

ensuite l'algorithme classique ACP s'opère dans l'espace de caractéristiques $F$ afin de calculer les composantes principales qui maximisent la variance de l'ensemble des données. Étant donné $C$ la matrice de covariance calculée par l'ACP à noyau dans le nouvel espace $F$, centrée, le but de l'ACP à noyau est d'extraire les valeurs et les vecteurs propres de la matrice $C$. Afin de réduire la dimension de l'espace, seules les valeurs et les vecteurs propres les plus élevés sont retenus. La taille de la matrice de noyau représente un inconvénient majeur de cette technique, où on trouve que la taille de cette matrice est égale au nombre d'observations au carré.

### *4.4.7 La sélection de caractéristiques*

Le processus de sélection des attributs est un processus qui vise à représenter un sous-espace de dimension inférieure par la sélection d'un certain nombre de caractéristiques de l'espace de départ selon un certain critère de performance, et sans combiner les caractéristiques de l'espace de départ par une fonction linéaire, ou non linéaire (cf. Fig.4.5)[17]. Une meilleure méthode de sélection est celle qui cherche à former un sous-ensemble de caractéristiques pertinentes dans un temps optimal. Former cet ensemble revient à évaluer toutes les combinaisons possibles de ses caractéristiques à l'aide d'une fonction d'évaluation qui mesure la capacité d'une variable, ou d'un ensemble de variables. Ce processus de sélection est constitué de trois éléments [13, 42]:

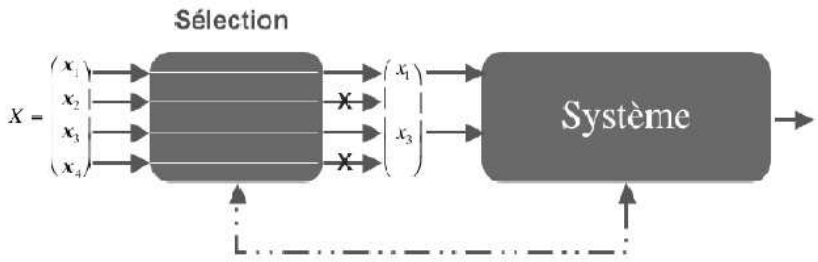



**Figure 4.5.** *Principe de sélection de variables*

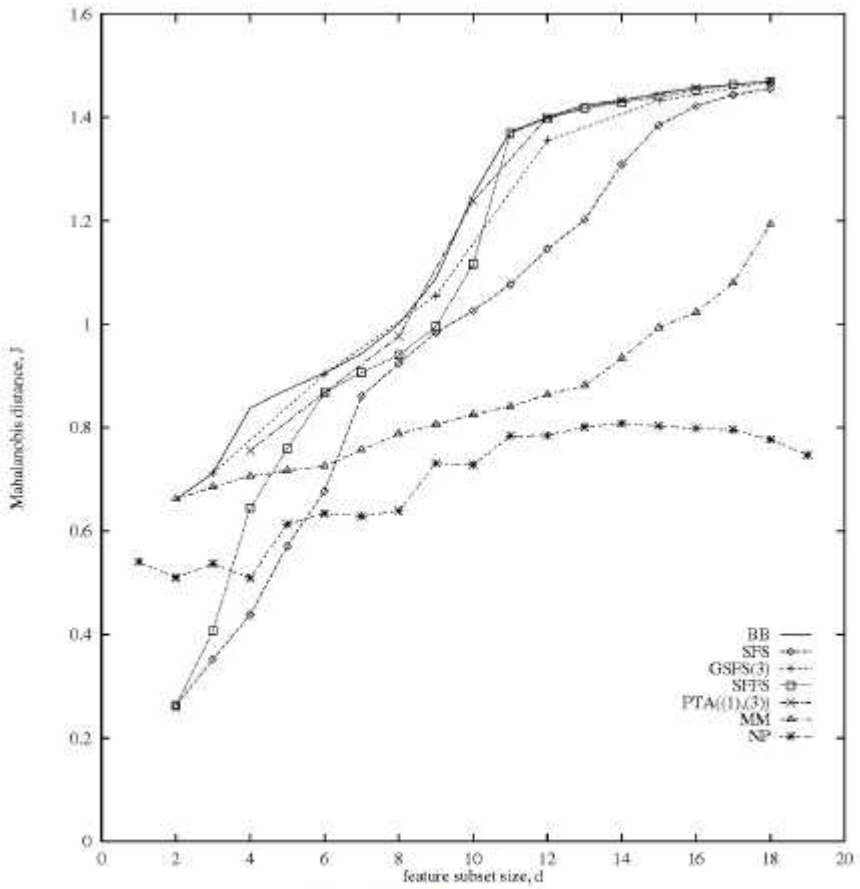
A) Un critère d'évaluation

Afin de déterminer un sous-ensemble de variables pertinentes, il est important de définir un critère d'évaluation qui qualifie l'utilité d'une variable, ou d'un groupe de variables. Dash et Liu [17] regroupent les critères d'évaluation en cinq catégories

- Critère de distance
- Critère d'information
- Critère d'indépendance
- Critère de consistance
- Critère de précision

B) Une procédure de recherche

Optimiser le temps de recherche d'un sous-ensemble optimal dépend de la procédure de recherche utilisée. On trouve dans la littérature différentes méthodes de recherche, par exemple, la méthode Branch et Bound qui a été proposée par Narendra et Fukunaga [47]. D'autres stratégies emploient des procédures séquentielles qui n'évaluent pas toutes les combinaisons possibles de l'ensemble de variables, comme la stratégie ascendante, la stratégie descendante, la stratégie bidirectionnelle [5, 56]. La figure (figure ci-dessous) dans l'article de [5] illustre la performance et le temps d'exécution de différentes stratégies de sélection.




Executation Time

Fig. 2. Performance and execution times of selected algorithms on synthetic 2-class Gaussian data set.

C) Un critère d'arrêt

Un critère d'arrêt permet d'arrêter le processus de sélection de variables quand un certain critère est satisfait. La définition d'un tel critère d'arrêt peut être lié à la mesure de pertinence, ou à la procédure de recherche. En général, le critère d'arrêt est déterminé par la combinaison de la mesure de pertinence et la procédure de recherche [17]. Selon les travaux de [13, 42], les méthodes de sélection sont réparties en trois types:

### *4.4.8 Les méthodes par filtre*

Les méthodes par filtre emploient un processus de prétraitement qui permet d'évaluer la pertinence d'un ensemble de variables à l'aide des propriétés statistiques de l'ensemble afin d'exclure les variables non pertinentes indépendamment de l'algorithme qui va les utiliser (cf. Fig. 4.6). Mark *et al.* [43] proposent, une heuristique de sélection basée sur une mesure de corrélation dont le but est d'extraire un sous-ensemble de variables utiles permettant d'améliorer la performance de l'algorithme de classification. Liu *et al.* [73] proposent une autre heuristique de sélection FCBF qui a l'avantage d'avoir une complexité linéaire par rapport à l'heuristique de [43]. Cette heuristique est basée sur la corrélation de prédominance. Ces méthodes ignorent les interactions entre les variables et le système.

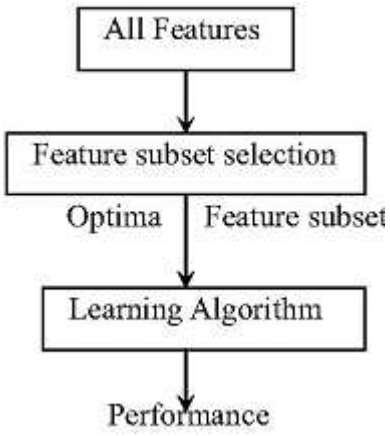



**Figure 4.6.** *Approche filtre [57]*

### *4.4.9 Les méthodes envellopées (warpper)*

La sélection par ces méthodes est basée sur un critère de précision. Le principe de cette approche est de tester toutes les combinaisons de sous-ensembles à travers un classifieur qui est considéré comme une fonction d'évaluation (cf. Fig. 4.7). Dans un problème de reconnaissance des chiffres manuscrits [49], les auteurs utilisent des

algorithmes génétiques pour réduire le nombre de sous ensembles à générer et qui permettent de trouver une solution optimale au problème étudié.

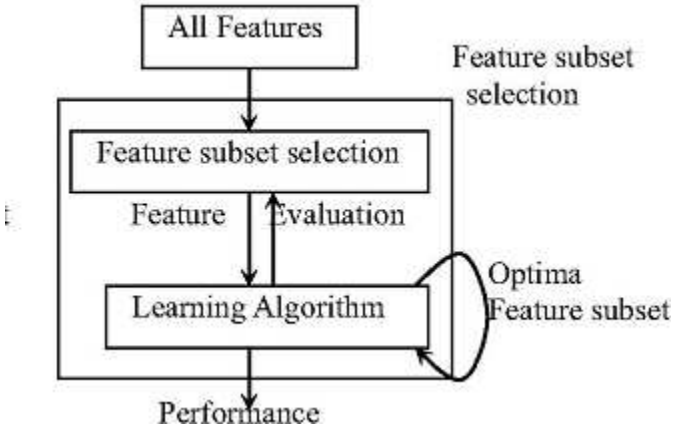



**Figure 4.7.** *Approche enveloppée [57]*

### *4.4.10 Les méthodes intégrées*

La sélection d'un groupe de variables s'effectue à l'aide d'un algorithme d'apprentissage tout en essayant de conserver le sous-ensemble qui optimise le mieux le critère d'apprentissage. Dans l'article [16], plusieurs techniques sont évoquées. Un tableau comparatif illustre les avantages, les inconvénients, et les différentes fonctions d'évaluations utilisées pour chaque approche dans [42] (Table 4.1).

| Modèle | | Avantages | Inconvénients | Exemples |
|---|---|---|---|---|
| Filtre | Univariée | Rapide, scalable | Ignore les interactions entre les variables | T-test, distance Euclidienne |
| | | Indépendant du classifieur | Ignore les interactions entre les variables et le classifieur | Gain d'informations |
| | Multivariée | Interacation entre les caractéristiques | lentes que les technique univariées | Fonction de corrélation (SCF) |
| | | Indépendant du classifieur | Moins scalable que les techniques univariées | Filtre de Markov (MBF) |
| | | Complexité de calcul meilleure | Ignore les interactions entre les variables et le classifieur | |
| Enveloppée | Déterministe | Simple | Risque de sur-apprentissage | Plus L Minus R |
| | | Interactions entre les variables et le classifieur | Recherche gloutonne | Sequential forward selection (SFS) |
| | | Interacation entre les caractéristiques | Interactions entre les variables et le classifieurs | Sequential backward elimination (SBE) |
| | | | | Algorithme de recherche en faisceau |
| | Aléatoire | Interactions entre les variables et le classifieur | Calcul coûteux | Recuit simulé |
| | | Interacation entre les caractéristiques | Risque de sur-apprentissage | Les algorithmes génétiques |
| | | | Interactions entre les variables et le classifieur | Algorithme à estimation de distribution |
| Intégrée | | Complexité de calcul meilleure que les méthodes enveloppées | Interactions entre les variables et le classifieur | Les arbres de décision |
| | | Interactions entre les variables et le classifieur | | Classifieur de Bayes Naïf |
| | | Interacation entre les caractéristiques | | |

**Table 4.1. Tableau comparatif.**

## 4.5 Conclusion

On a étudié dans ce chapitre les différentes techniques de réduction de dimensionnalité, de sélection de variables et d'extraction de caractéristiques. Cette famille de techniques permettant d'extraire plusieurs informations pertinentes à partir d'une base d'apprentissage dont le but est de faciliter les différentes tâches de l'apprentissage machine (comme une tâche de classification ou de régression). On présente dans le chapitre suivant quelques techniques de classification utiles dans les systèmes de vision par ordinateur en particulier dans les systèmes de reconnaissance.

Chapitre 5

# LES TECHNIQUES DE CLASSIFICATION

## *5.1 Introduction*

L'apprentissage machine est l'ensemble des techniques permettant de doter la machine de systèmes automatisés capables de simuler le comportement intelligent de l'être humain. Ces techniques permettent de traiter divers problèmes liés à la vision par ordinateur. L'objectif de l'apprentissage automatique est de concevoir un modèle à partir d'un nombre d'exemples important afin de résoudre un problème spécifique. Étant donné que la base d'apprentissage est constituée d'un nombre d'exemples fini généré par un processus, le modèle conçu doit être plus robuste lors de la présentation de nouveaux cas qui ne se trouvent pas explicitement dans la base d'apprentissage. Un tel problème de reconnaissance peut être vu comme un problème de classification. La classification est l'une des méthodes d'apprentissage automatique qui consiste à définir une fonction dont le but est d'associer une étiquette à un objet représenté par un ensemble de caractéristiques. En général, les méthodes de classification sont divisées en deux familles:

- Méthodes supervisées
- Méthodes non supervisées

## *5.2 Les méthodes de classification supervisées*

Dans ces méthodes, l'ensemble d'apprentissage est constitué d'un ensemble de couples entrées-sorties dans lequel l'entrée représente le vecteur de caractéristique et la sortie représente l'étiquette correspondant à l'entrée. Le problème de classification supervisé

cherche à identifier la classe d'appartenance d'une nouvelle entrée qui n'appartient pas à l'ensemble d'apprentissage, tout en essayant d'apprendre une fonction de classification à partir d'un ensemble d'entrainement. La fonction de classification apprise permet d'associer une classe à une nouvelle entrée à l'aide de certaines caractéristiques qui décrivent l'entrée.

### *5.2.1 K plus proches voisins*

Le $K$-ppv est un modèle qui appartient à la famille des modèles non paramétriques. L'algorithme $K$ plus proches voisins est un algorithme qui se base sur le concept de proximité. L'apprentissage par cet algorithme est considéré comme un apprentissage paresseux car il consiste seulement à stocker l'ensemble d'entrainement, cela veut dire, qu'il ne comporte pas d'étape d'apprentissage qui permette d'apprendre un modèle à partir d'un ensemble d'échantillons. Cet algorithme définit une fonction de distance entre les vecteurs de caractéristiques pour faire la classification d'une nouvelle entrée. Pour classer une nouvelle entrée, l'algorithme procède en identifiant un ensemble de $K$ plus proches voisins pour chaque classe. Cet ensemble est obtenu en faisant une comparaison entre la nouvelle entrée et chaque exemple de l'ensemble d'entrainement à l'aide d'une mesure de similarité. Une fois que l'ensemble de $K$ plus proches voisins est trouvé, l'algorithme cherche la classe qui a le plus de représentants dans cet ensemble afin d'associer une étiquette à la nouvelle entrée. La performance du classifieur $K$-ppv est dépendante de la valeur de $K$ et de la fonction de distance utilisée. L'avantage de cette méthode est qu'elle construit un nouveau modèle pour chaque nouvelle d'entrée. D'autres avantages sont aussi que cette méthode est simple et robuste au bruit. Cette méthode est sensible si le vecteur de grandes dimensions contient un grand nombre de caractéristiques non pertinentes. Le classifieur $K$-ppv a été appliquée avec succès dans différentes applications de reconnaissance telles que la reconnaissance des actions humaines [10, 35], la reconnaissance d'objets [46], la reconnaissance des postures [58], la détection de chute [21], etc.

### 5.2.2 Machines à vecteurs de support (SVM)

La machine à support vectoriel représente une famille d'algorithmes d'apprentissage qui s'inspire de la théorie statistique de l'apprentissage de Vapnik[18]. Le SVM est un classifieur binaire basé sur l'astuce de noyau. L'apprentissage par ce classifieur consiste à apprendre une fonction discriminante linéaire $f$ à partir d'un jeu d'entrée constituée d'un certain nombre de couples entrées-sorties afin de pouvoir produire une sortie $y$ étant donnée une nouvelle mesure. Ce type de classifieur tente de trouver une frontière de décision permettant de séparer linéairement les exemples de la première classe des exemples de la deuxième classe dans l'ensemble d'apprentissage. Le principe de ces méthodes est de trouver le meilleur hyper plan qui maximise la marge entre les exemples d'apprentissage et l'hyper plan (cf. Fig. 5.1)[18]. La classification d'une nouvelle mesure se fait par la fonction apprise. L'avantage principal de ces méthodes est qu'elles puissent être appliquées dans le cas où les classes ne sont pas linéairement séparables. Dans ce cas, les méthodes des SVM tentent de trouver des frontières de décision non linéaire. Le classifieur SVM emploie des fonctions de noyau (polynomial, gaussien) permettant de projeter les caractéristiques initiales dans un nouvel espace à grande dimension. Cette projection vise à rendre les données linéairement séparables dans le nouvel espace. Ensuite le classifieur SVM cherche à trouver un séparateur linéaire dans le nouvel espace qui devient un séparateur non linéaire dans l'espace originale. Les méthodes SVM ont été appliquées avec succès dans les problèmes de catégorisation des textes [41], de reconnaissances des actions humaines [14, 34], de reconnaissance d'objets [54], etc.

[18] en.wikipedia.org/wiki/Supportvectormachine

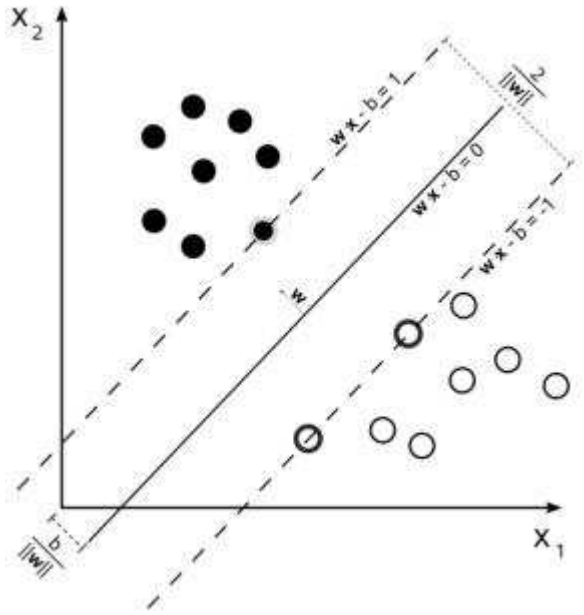



**Figure 5.1.** *Frontière de décision linéaire d'un classifieur SVM. Les échantillons qui se trouvent sur la marge s'appellent les vecteurs de support*

### *5.2.3 Classifieur bayésien*

L'apprentissage par cette méthode issue de la théorie bayésienne, consiste à entrainer un estimateur de densité de probabilité sur chaque classe afin de concevoir un modèle probabiliste (en modélisant, par exemple, les probabilités conditionnelles de chaque classe par des gaussiennes multidimensionnelles). Dans ce cas, le problème de l'apprentissage revient à apprendre les deux paramètres de la gaussienne qui sont le vecteur de moyenne et la matrice de covariance à l'aide d'une base d'apprentissage. Pour une nouvelle entrée, le modèle construit permet d'estimer la probabilité de vraisemblance de chaque classe. La classification d'une nouvelle observation est basée sur l'estimation des probabilités *a posteriori* des classes en retenant la classe pour laquelle la probabilité *a posteriori* est maximale. La probabilité *a posteriori* d'une classe est estimé à partir la connaissance de la probabilité *a priori* et la probabilité

de vraisemblance par la règle de Bayes. Dans [37], les auteurs ont montré que la performance de la version simple du classifieur bayésien est similaire à la performance de certains algorithmes d'apprentissage comme les arbres de décision et les réseaux de neurones. L'inconvénient de cette méthode est qu'elle exige la connaissance de plusieurs informations *a priori*. Les classifieurs bayésiens et leurs extensions ont été appliquées avec succès dans les problèmes de catégorisation des textes [44], et dans les applications médicales [31].

### *5.2.4 Réseaux de neurones*

Un réseau de neurones artificiel est un modèle de calcul issu des modèles biologiques. Ce modèle permet de simuler le comportement du cerveau humain. Les réseaux de neurones sont caractérisés par leur capacité d'apprentissage. En général, la structure d'un réseau de neurones est composée d'une succession de couches cachées. La couche d'entrée est reliée à la couche de sortie du réseau à travers les couches cachées selon une architecture défini, comme l'architecture du perceptron, le perceptron multicouche. Chaque couche cachée est constituée d'un certain nombre de neurones reliés à la couche précédente. La couche suivante reçoit en entrée les sorties de la couche précédente du réseau. Le vecteur d'entrée pour chaque couche est pondéré par un poids. À l'aide d'une fonction d'activation, le réseau procède à calculer ses poids afin de produire une sortie (cf. Fig. 5.2)[19].

[19] fr.wikipedia.org/wiki/Reseaudeneuronesartificiels

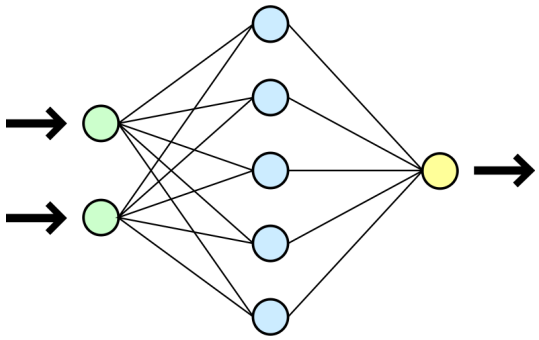


**Figure 5.2.** *Exemple d'un réseau de neurones* [19]

### *5.2.5 Les réseaux à apprentissage supervisé*

Dans ce modèle, le but est d'entrainer un réseau de neurones qui cherche à converger vers une sortie précise. Étant donné en entrée un vecteur de caractéristiques avec une cible correspondant à ce vecteur, l'apprentissage consiste à mettre à jour les poids du réseau en faisant une comparaison entre la sortie désirée et la sortie que le réseau a produites. Le réseau s'adapte jusqu'à ce que la sortie corresponde à la cible.

### *5.2.6 Les perceptrons*

Les perceptrons se divisent en deux classes selon la manière dont les neurones sont interconnectés dans le réseau.

#### *5.2.6.1 Les perceptrons monocouches*

Le perceptron monocouche est un réseau de neurones qui s'inspire du système visuel. Ce type de réseau ne possède que deux couches, une couche d'entrée et une couche de sortie. La couche d'entrée est connectée à la couche de sortie sans aucune couche intermédiaire (cf. Fig. 5.3)[20]. Le réseau tente de trouver une frontière de décision linéaire. L'entrainement de ce réseau se fait par l'initialisation de chaque poids de

[20] www.csulb.edu/ cwallis/artificialn/History.htm

liaison entre l'entrée $i$ et le neurone de sortie $j$ par des valeurs aléatoires, puis un vecteur d'entrée et un vecteur de sortie désiré sont présentés au réseau afin de permettre au réseau de produire un résultat. La mise à jour des poids des liaisons se répète jusqu'à ce que le réseau produise une sortie correspondante à la sortie attendue. Ce réseau est appliqué seulement si les données sont linéairement séparables.

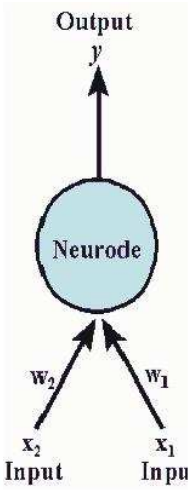



**Figure 5.3.** *Perceptron monocouche*[20]

#### *5.2.6.2 Perceptron multicouches*

Contrairement au perceptron monocouche, ce type de réseau possède des couches intermédiaires qui font la liaison entre la couche d'entrée et la couche de sortie (cf. Fig. 5.4)[21]. L'entrainement de ce réseau se fait par l'algorithme de rétropropagation de l'erreur du gradient qui consiste à minimiser l'erreur de classification dont le but est de trouver les meilleurs poids de liaisons. Ce réseau tente de trouver un séparateur non linéaire dans l'espace de caractéristiques. La mise à jour des poids des liaisons permette de différencier les perceptrons multicouches et monocouches. Dans le perceptron multicouche la mise à jour de ces poids se fait d'une façon récursive en partant

[21] en.wikipedia.org/wiki/Feedforwardneuralnetwork

de la couche de sortie vers la couche d'entrée en modifiant tous les poids de liaisons de chaque couche. Ensuite en faisant la somme pondérée de ces poids. Ce type de réseau offre la possibilité de traiter les cas où les données ne sont pas linéairement séparables. L'un des inconvénients de ce réseau est sa taille. Dans l'article [39], un MLP a été appliqué pour une application de reconnaissance d'actions humaines multivue. Dans [8], l'auteur suggère un MLP pour détecter la face humaine à partir d'une image.

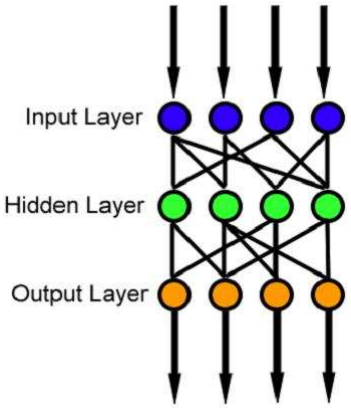



**Figure 5.4.** *Perceptron multicouches*[21]

### *5.2.7 Les arbres de décision*

Les arbres de décision sont des méthodes appartenant à la famille des modèles non paramétriques. Un arbre de décision est constitué d'un ensemble de noeuds et des feuilles dans lequel les noeuds de décision représentent les caractéristiques et les feuilles correspondent aux étiquettes. L'apprentissage d'un arbre revient à utiliser des algorithmes comme Cart [25] qui essaient de minimiser l'erreur de classification afin de construire un arbre de décision plus robuste. L'arbre construit permet de fournir un ensemble de règles de décision. La classification d'un nouvel exemple se fait par le parcours d'un chemin dans l'arbre en évaluant l'exemple au niveau de chaque noeud jusqu'à ce qu'on atteigne une feuille dans l'arbre. Un arbre a l'avantage

qu'il n'exige aucune connaissance *a priori* sur la distribution des données. L'un des inconvénients d'un arbre de décision est leur instabilité, cela signifie que la qualité de prédiction est affectée par le changement d'une telle caractéristique dans l'arbre. Dans [60], l'auteur suggère d'entrainer un arbre de décision pour un problème de catégorisation de texte (cf. Fig. 5.5).

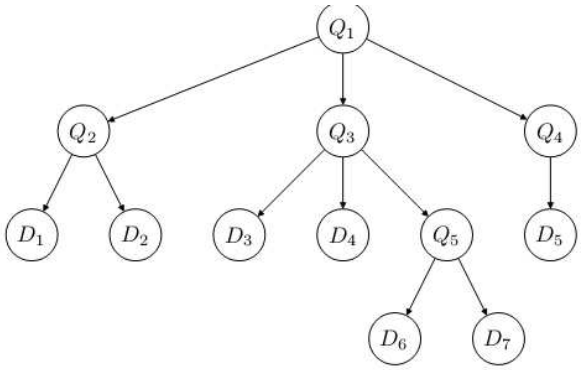



**Figure 5.5.** *Exemple d'un arbre de décision [60]*

## *5.3 Les méthodes de classification non supervisées*

Les méthodes de classification non supervisée sont des méthodes[22] qui cherchent à identifier, ou à partitionner un ensemble de données à un certain nombre de classes distinctes à partir d'un fichier de description, tout en essayant d'optimiser un critère qui vise à regrouper les données les plus homogènes dans chaque classe.

### *5.3.1 K-moyennes*

C'est une méthode itérative qui a été proposée par MacQueen en 1967[22]. Le principe de l'algorithme k-means est le suivant:

- On définit un nombre k de nuages *a priori*

[22] home.deib.polimi.it/matteucc/Clustering/tutorialhtml/index.html

- Chaque nuage est initialisé par un centre. Le centre est tiré d'une façon aléatoire de l'espace de l'individu
- On alloue chaque individu à un nuage $i$ en basant sur une mesure de similarité.
- En faisant la moyenne des éléments dans chaque nuage $i$ afin de produire les nouveaux centres
- On ré-itére jusqu'à ce qu'aucun individu ne change de nuage

Un défaut de l'agorithne k-means est le choix des conditions initiales qui peuvent affecter les résultats de classement, ça signifie que la partition d'un groupe d'individus dépend largement des centres initiaux et du nombre de nuages.

### *5.3.2 Fuzzy c-means*

Une variante de l'algorithme k-means[22]. La classification floue consiste à associer à chaque cluster un coefficient $u_{ij}$ qui représente le degré d'appartenance d'un point $x_i$ au cluster $c_i$. Le problème revient à minimiser le critère intra classe (5.1). À chaque itération, la mise à jour des coefficients d'appartenance et les centres de gravité de clusters se fait respectivement par les équations (5.2) et (5.3).

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{c} u_{ij}^m \left\|x_i - c_j\right\|^2 \tag{5.1}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \frac{\|x_i - c_j\|^{\frac{2}{m-1}}}{\|x_i - c_k\|}} \tag{5.2}$$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m x_i}{\sum_{i=1}^{N} u_{ij}^m} \tag{5.3}$$

avec $m$ représente un hyper paramètre qui règle le degré de flou de sous ensemble produit. Cet algorithme a été appliqué avec succès dans le problème de segmentation de l'image [71].

### *5.3.3 Classification hiérarchique*

Les méthodes de classification hiérarchique [52] sont des méthodes itératives fondées sur des mesures de similarité. Ces méthodes visent à construire des regroupements en classes homogènes d'un ensemble d'individus. On cite par exemple la méthode de classification ascendante qui consiste à calculer une matrice de similarité entre chaque paire d'objets. Ensuite à chaque itération, un nouveau cluster est formé par la fusion de deux clusters les plus proches en basant sur la matrice trouvée. La matrice de similarité est mise à jour par le calcul de la ressemblance entre le nouveau cluster et les clusters existants. La mise à jour se répète jusqu'à la fusion de deux derniers clusters. La qualité de clustering par cette méthode dépend largement de la métrique utilisée. Une application de la classification hiérarchique est la recherche d'image [38] (cf. Fig. 5.6)[22].

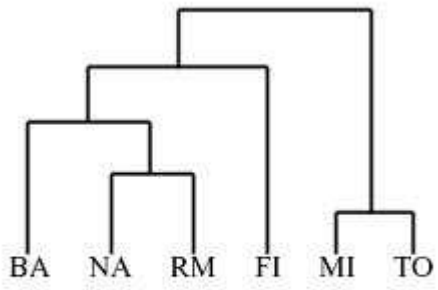



**Figure 5.6.** *Processus de fusion par une méthode hiérarchique*

## *5.4 Évaluation de la performance d'un classifieur supervisé*

Apprendre un classifieur revient à entrainer un modèle sur un ensemble d'apprentissage qui minimise le taux d'erreur, étant donné que le taux d'erreur est le nombre d'exemples mal classé. Le but est d'entrainer un classifieur qui fait moins d'erreurs si on lui présente de nouveaux cas qu'on n'a pas regardés pendant la phase d'entrainement.

Dans ce cas, on parle de l'erreur de généralisation. Il existe plusieurs techniques permettant d'évaluer la performance d'un classifieur:

### *5.4.1 La validation croisée*

Dans ce type, les méthodes de classification sont évaluées sur une base de test. La procédure d'évaluation leave-one-out consiste à diviser la base d'exemples de taille $n$ en $n$ sous-bases [55]. Ensuite à entrainer le modèle sur $n-1$ sous-bases, puis le tester sur la base restante. Le processus se répète un certain nombre de fois. On distingue d'autres techniques de la validation croisée:

- Méthode holdout: la procédure d'évaluation consiste à séparer toutes les données dont on dispose en deux ensembles, un ensemble d'entrainement et un ensemble de test. Ensuite à entrainer le modèle sur l'ensemble d'entraiment, puis l'évaluer sur l'ensemble de test.

- K-fold-cross-validation: la procédure d'évaluation consiste à diviser k fois l'ensemble de données de taille $n$ en $k$ sous-bases. Ensuite à entrainer le modèle sur k-1 sous-bases, puis le valider sur la base $k$. La procédure se répète k fois. On note que cette procédure est appelé leave-one-out dans le cas où $k = n$.

### *5.4.2 Les courbes ROC*

Une courbe ROC est un outil graphique qui vise à représenter, et à comparer la performance d'un classifieur par rapport à un autre en fonction du risque associé à chaque classe [32] (cf. Fig. 5.7). La courbe ROC est inspirée de l'analyse de la matrice de confusion. Cette matrice permet de fournir deux quantités, la sensibilité et la spécificité. L'idée de la courbe ROC est de faire varier un seuil et, pour chaque cas, calculer la spécificité et la sensibilité que l'on reporte dans un graphique où l'inverse de la spécificité se place en abscisse et la sensibilité se trouve en ordonnée, et étant donné les deux mesures:

| | positif | négatif |
|---|---|---|
| positif | TP | FP |
| négatif | FN | TN |

**Table 5.1. Matrice de confusion**

- la sensibilité=TP/TP+FN
- la spécificité=TN/TN+FP

où TP, FP, TN, FN représentent respectivement le nombre de positifs correctement reconnus, le nombre de négatifs détectés par erreur, le nombre de négatif correctement reconnus, le nombre de positifs détectés par erreur.

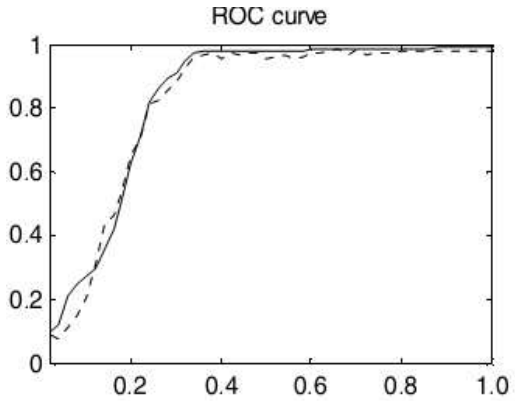



**Figure 5.7.** *Courbe ROC montrant la comparaison de la performance d'un classifieur SVM par rapport à un réseau de neurones [32]*

## *5.5 Conclusion*

Dans ce chapitre, nous avons présenté quelques techniques de classification utilisées dans le domaine de la vision par ordinateur. Nous avons aussi donné un aperçu sur les techniques de mesure de leurs performances. La classification non supervisée se

différencie de la classification supervisée par l'absence de la base d'apprentissage, ce qui rend les méthodes de classification non supervisée moins adaptées pour un tel système de reconnaissance d'actions humaines. On focalise donc notre projet sur des méthodes de classification supervisé.

Chapitre 6

# IMPLÉMENTATION ET RÉALISATIONS

Dans ce chapitre, on présente la méthode que nous avons adopté pour reconnaitre une action à partir d'une séquence vidéo.

## *6.1 Description de la base d'actions*

La base d'actions humaines de Weizmann[13] est une base constituée d'un ensemble de séquences vidéo prises par une caméra statique avec un nombre d'images par seconde égale à 50 fps. Dans cette base, on trouve au total 90 séquences vidéo. La longueur de la séquence diffère d'une séquence à une autre avec une résolution de $180 \times 144$ pixels. La base regroupe 10 classes qui représentent les différentes actions humaines réalisées, à savoir: RUN, WALK, SKIP, JACK, JUMP, PJUMP, SIDE, WAVE-TWO-HANDS, WAVE-ONE-HAND, BEND. Dans chaque classe, une action périodique est réalisée par 9 personnes de taille et de genre différents, et avec une manière et une vitesse d'exécutions différentes (cf. Fig. 6.1)[13]. Pour certaines classes, l'action est effectuée dans deux directions, de la gauche vers la droite et vice-versa. L'avantage de cette base est qu'elle comporte plusieurs classes par rapport à d'autres bases d'actions comme la base proposée en [62].

## *6.2 Méthode*

Notre système de reconnaissance est basé sur trois étapes:

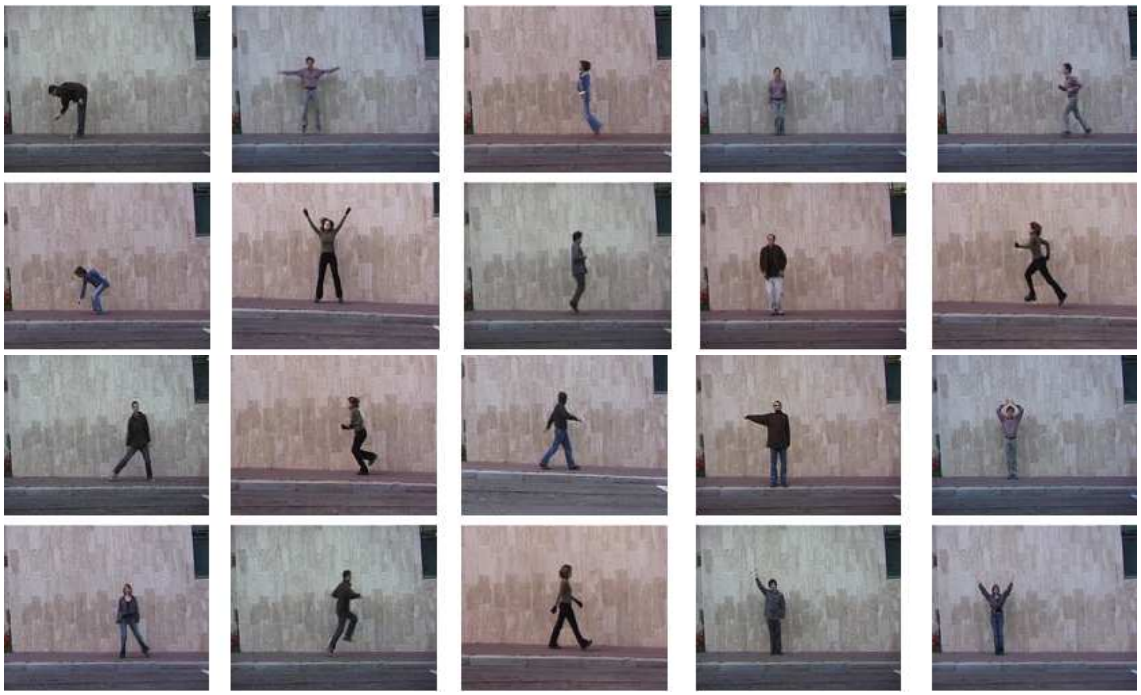


**Figure 6.1. La base d'actions humaines de Weizmann.**

### *6.2.1 Extraction de la silhouette*

La première étape de notre système vise à modéliser la personne par une boîte englobante afin de réduire l'espace spatial de recherche. On envisage les prétraitements suivants:

### *6.2.2 Soustraction du fond*

Étant donné que l'on dispose de l'arrière-plan dans la base, la technique la plus simple qui nous permet de détecter une personne consiste à appliquer une opération de soustraction entre l'arrière-plan et une image à l'instant $t$, suivie d'une opération de seuillage afin de faire apparaitre les pixels de l'arrière-plan en noir et les autres en blanc. L'objet extrait est donné par l'équation suivante:

$$|\text{Fond} - \text{frame}(t)| > T \tag{6.1}$$

Où Fond, Frame($t$) et $T$ représentent respectivement l'arrière-plan, l'image à un instant $t$, et le seuil.

### 6.2.3 Un filtrage

Souvent l'image obtenue après une opération de soustraction de fond est bruitée. Pour diminuer l'effet de ce bruit, nous avons appliqué le filtre median de taille $3 \times 3$ sur l'image où nous avons constaté que les résultats obtenus avec cette dimension sont suffisants.

### 6.2.4 Une opération morphologique

Une fois que le bruit est diminué dans l'image, on applique une opération d'érosion suivie d'une opération de dilatation morphologique avec un masque de dimension $3 \times 3$ pour les deux opérations afin d'éliminer les pixels isolés et de raffiner les résultats.

### 6.2.5 Translation

Le but de cette étape est de rendre le vecteur de l'image invariant au déplacement en faisant le centrage de toutes les personnes sur le même centre de gravité. Pour cela, on utilise une transformation géométrique permettant de translater les personnes détectées au centre de la silhouette. La translation est donnée par l'équation ci-dessous:

$$\begin{pmatrix} x_{pixel\ transalte} \\ y_{pixel\ translate} \end{pmatrix} = \begin{pmatrix} c_x \\ c_y \end{pmatrix} - \begin{pmatrix} x_i \\ y_j \end{pmatrix}, \tag{6.2}$$

avec $\begin{pmatrix} c_x \\ c_y \end{pmatrix}$ représentant le centre de l'image, et $\begin{pmatrix} x_i \\ y_j \end{pmatrix}$ représentant la position d'un pixel à un instant $t$ de la séquence.

On donne ci-dessous un pseudo-code qui résume les différents prétraitements envisagés sur une séquence vidéo:

**Entrées:**

$S$: Cycle d'action représenté par un certain nombre de trames (Frames)

$Fond$: Image de l'arrière plan

$T$: Un seuil

**Sorties:**

$buff$: Cycle d'action representé par une personne centrée sur le même centre de gravité

**foreach** *frame t de la séquence S* **do**

  **if** $|Fond - frame(t)| > T$ **then**

    $buff$=1

  **else**

    $buff$=0

  **end**

  - Appliquer une opération de filtrage suivie d'une opération d'érosion, dilatation sur $buff$
  - Trouver le centre de l'image: $\begin{pmatrix} c_x \\ c_y \end{pmatrix} = \begin{pmatrix} \sum x_i \\ \sum y_j \end{pmatrix}$
  - Translater la personne au centre de l'image: $\begin{pmatrix} x_{pixel\ transalte} \\ y_{pixel\ translate} \end{pmatrix} = \begin{pmatrix} c_x \\ c_y \end{pmatrix} - \begin{pmatrix} x_i \\ y_j \end{pmatrix}$

**end**

**Algorithme 1**: Prétraitement

La figure ci-dessous illustre un cycle d'action dans le cas où la personne marche (séquence WALK):

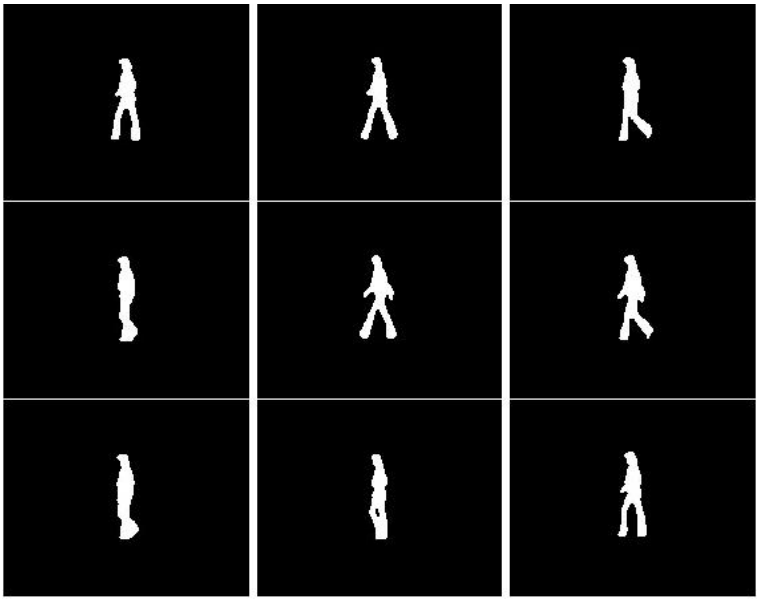


**Figure 6.2.** *Centrage sur le même centre de gravité.*

### *6.2.6 Extraction de caractéristiques*

La deuxième étape de notre méthode vise à appliquer une technique de réduction de dimensionnalité sur notre séquence d'images binarisée prétraitée afin de construire une base d'apprentissage constituée de différents prototypes, générés à partir de deux points de vue, avec le minimum de dimension possible, et sur laquelle on va entrainer un classifieur (cf. Fig. 6.3). Nous avons choisi la technique de réduction de dimensionnalité MDS. Cette technique permet de modéliser une action humaine spécifique par un prototype discriminant représenté dans un espace de faible dimension. Discriminant ici, signifiera que l'on espère obtenir des prototypes proches si les actions sont similaires (*i.e.*, appartenant à une même classe d'action humaine) (cf. Fig. 6.6, 6.7, 6.8, 6.9). Dans notre application, un prototype est un ensemble de points, dans un espace de faible dimension, dans lequel chaque point modélise une image de la

séquence. Dans la base d'actions de Weizmann, l'action est périodique. Nous avons choisi une seule période pour chaque cycle d'action de notre base d'apprentissage au lieu de plusieurs cycles considérant le fait que la technique MDS générera aussi un prototype périodique (à un bruit près) (cf. Fig. 6.10).

Pour construire notre base d'apprentissage de prototypes de faible dimension, on procède comme suit:

1. Chaque cycle d'action de la base d'apprentissage est représenté par une séquence d'images binarisée et prétraitée (cf. Algorithme 1) (dans laquelle chaque pixel ne peut prendre que deux valeurs; 1 ou 0 si celui-ci a été précédemment étiqueté mobile ou immobile) de même hauteur $H$ et largeur $L$ et possédant le même nombre d'images $N_i$ (dans notre application, nous avons pris 13 trames) pour toutes les séquences de la base d'apprentissage.

2. On applique ensuite l'algorithme de réduction de dimensionnalité MDS sur chacune des séquences d'images binarisée selon deux points de vue différents (cf. Fig. 6.3);

   (a) Le premier point de vue réduit ce cube d'images temporellement, *i.e.*, suivant l'axe du temps, en considérant chaque image binarisée comme un vecteur colonne de dimension $N$ ($N$ étant le nombre de pixels dans l'image). Chaque image de la séquence est réduite en dimension 3 par la technique MDS ce qui nous permettra de représenter chaque prototype, associé à chaque séquence d'images, selon ce point de vue, par une trajectoire de $N_i$ points (*i.e.*, une courbe décrite par la succession des $N_i$ différents points ordonnés et associés à chaque image de la séquence) dans un espace tridimensionnel.

   (b) Le deuxième point de vue réduit ce cube d'images latéralement, suivant l'axe des lignes (ou des $y$), en considérant la série des $H$ images

latéralement créées (dans ce cube de données) de longueur $N_i$ et de largeur $L$. Chacune de ces images est réduite en dimension 3 par la même technique MDS ce qui nous permettra aussi de représenter chaque prototype, associé à chaque séquence d'images, selon ce point de vue, par une trajectoire de $l$ points dans un espace tridimensionnel.

La réduction de dimensionnalité selon ces deux points de vue différents permettra de représenter un cycle d'action donnée en éliminant la redondance d'information et le bruit de deux façons différentes et complémentaires. Une technique de fusion que l'on décrira plus tard permettra de fusionner ces deux prototypes efficacement afin de rendre notre système de reconnaissance plus performant. Les tests ont montré que la réduction de dimensionnalité selon le troisième point de vue (latéralement selon l'axe des colonnes, ou des $x$) n'apportait aucun gain en matière de taux de bonne classification (cf. Section 6.6).

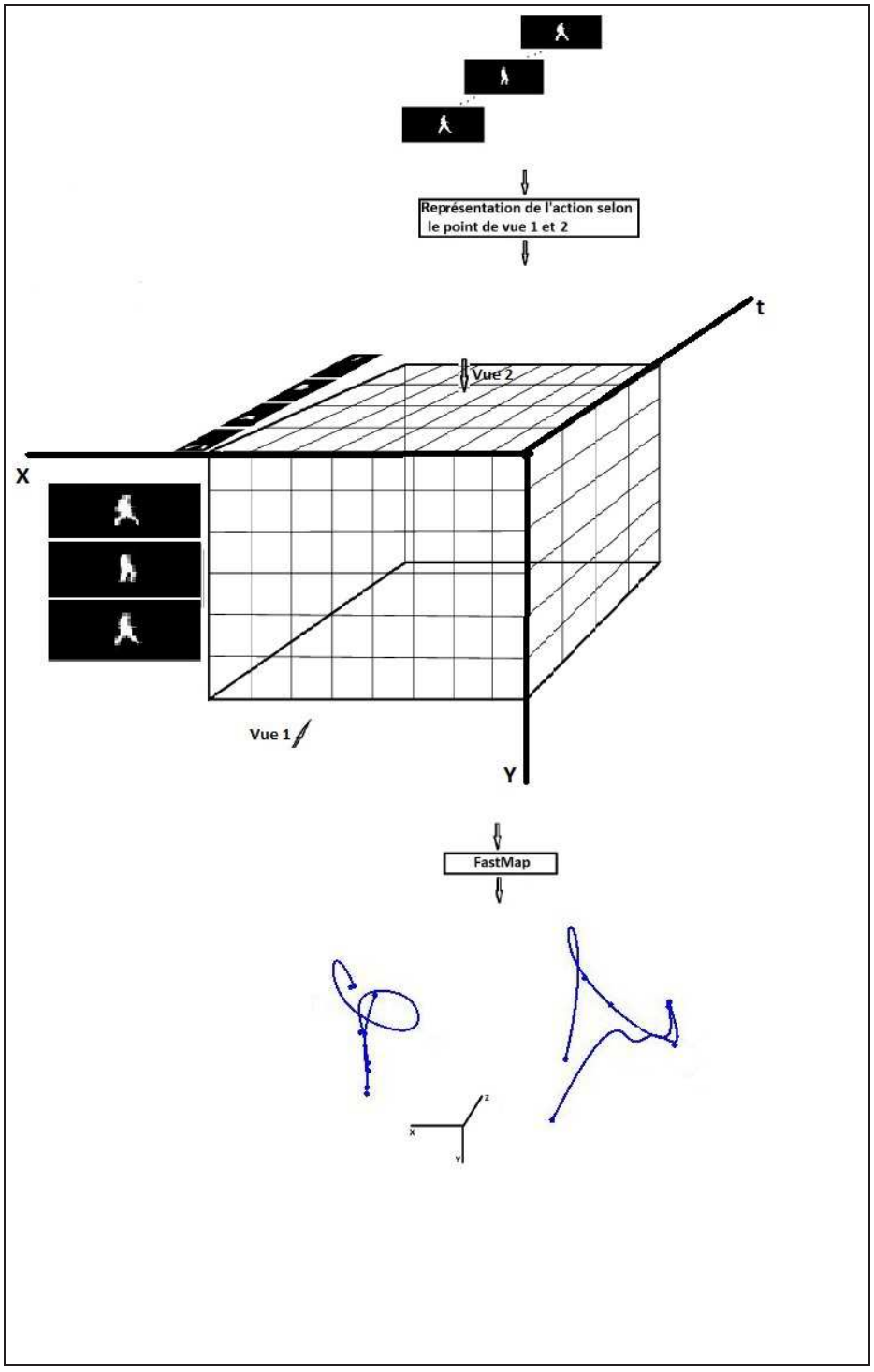



**Figure 6.3.** *Modélisation par réduction de dimensionnalité non linéaire d'un cycle d'action selon deux points de vue différents*

### 6.2.7 Le FastMap

#### 6.2.7.1 Introduction

La réduction de dimensionnalité MDS est réalisé dans notre application par l'algorithme FastMap de Faloutsos et Lin [22] qui est efficace, rapide et très bien adapté pour les réductions de dimensionnalité importantes comme notre application l'exige (nécessitant pour les premiers et deuxièmes points de vue respectivement, une réduction d'une dimension $N$ et $H$, le nombre de pixels dans l'image et la longueur de l'image à la dimension réduite $k = 3$). La seule restriction est de considérer une mesure de distance (de dissimilarité entre objets) qui obéit à l'inégalité triangulaire, ce qui est notre cas puisque l'on considérera la distance Euclidienne.

Dans le cas d'une réduction réduite de dimension $k$, l'idée de base de cet algorithme est de considérer que les $N_p$ objets de l'ensemble de donnée sont des points dans un espace de dimension $n$ ($n >> k$) inconnu et d'essayer de projeter ces points successivement sur $k$ axes de coordonnées mutuellement orthogonales en n'utilisant comme donnée d'entrée que certaines distances de (dis)similarités entre pair d'objets.

#### 6.2.7.2 Principe

Le coeur de cet algorithme est de projeter les différents objets (supposés de dimension initiale $n$) sur un axe de projection, soigneusement sélectionné. À cette fin, l'algorithme FastMap va choisir deux objets $O_a$ et $O_b$, que l'on appelle *pivots* (et que l'on apprendra plus tard à choisir automatiquement) et considérer l'axe de projection passant à travers ces deux *pivots* (de dimension $n$). La projection des objets de l'ensemble de données sur cet axe de projection est calculé à partir du théorème de Pythagore [22] (cf. Fig. 6.4):

$$x_i = \frac{D^2(O_a, O_i) + D^2(O_a, O_b) - D^2(O_b, O_i)}{2\, D^2(O_a, O_b)} \tag{6.3}$$

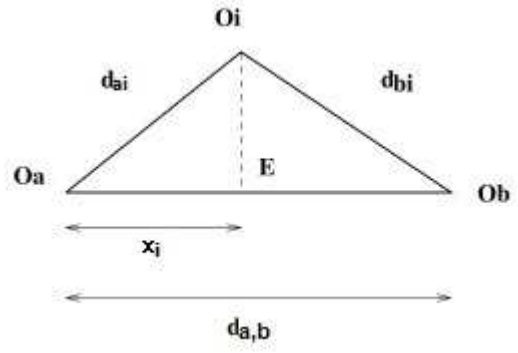



**Figure 6.4.** *Illustration du théorème de Pythagore pour la projection sur l'axe de coordonné $O_aO_b$. Ici $d_{ai} = D(O_a, O_i)$ représente la distance entre l'objet $O_a$ et l'objet $O_i$.*

Dans laquelle $D(O_a, O_b) = d_{a,b}$ représente la distance Euclidienne entre l'objet $0_a$ et l'objet $O_b$. Remarquons que dans le calcul de cette projection selon un axe, on n'utilise que les distances entre certaines paires d'objets et que ce calcul, pour tous les objets $O_i$ $(i = 1, \ldots, N_p)$ se fait en complexité linéaire. Remarquons aussi qu'après cette projection en dimension $k = 1$, on a préservé l'information de distance entre paires d'objets. En effet, si $O_i$ est proche du pivot $O_a$, $x_i$ sera petit et ainsi on a résolu le problème du MDS pour $k = 1$.

Cette méthode est généralisé dans le cas $k = 2$ puis pour n'importe quelle $k$ par l'idée de base du FastMap qui est de considérer que les objets sont des points dans un espace de dimension $n$. Pour cela, on considère un hyperplan $H$ de dimension $(n-1)$ qui est perpendiculaire à l'axe de projection $(O_aO_b)$ et sur lequel on projette nos objets. Soit $O'_i$ la projection de $O_i$ (pour $i = 1, \ldots, N_p$). Cette stratégie nous permet de réduire la dimension des objets de $n$ à $(n-1)$ et en utilisant le théorème de Pythagore une fois de plus, les auteurs [22] montrent que:

$$D^2(O'_i, O'_j) \;=\; D^2(O_i.O_j) - (x_i - x_j)^2 \qquad i, j = 1, \ldots, N_p \tag{6.4}$$

qui permet de recalculer la (dis)similarité entre paires d'objets en dimension $(n-1)$. La capacité à réaliser le calcul de ces (dis)similarités entre paires d'objets en

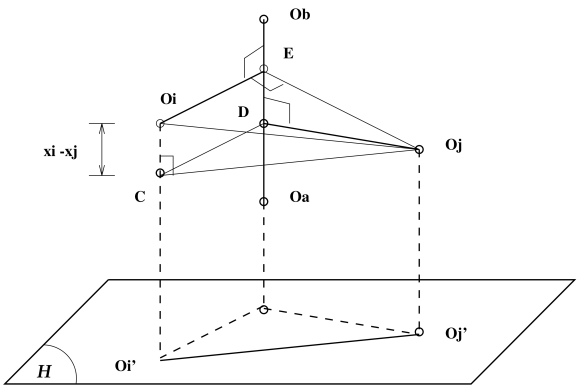



**Figure 6.5.** *Projection sur un hyper-plan $H$, perpendiculaire à la ligne $O_aO_b$ de la figure précédente.*

dimension $(n-1)$ nous permet de projeter les données sur un second axe de coordonné, contenu dans cet hyperplan $H$ et donc orthogonal (par construction) au premier axe de coordonné donné par la droite $(O_aO_b)$. Cela permet ainsi de résoudre le problème MDS pour une dimension réduite $k = 2$. Plus important encore, en appliquant les mêmes étapes récursivement $k$ fois, on est donc capable de résoudre le problème pour n'importe quelle dimension réduite.

#### *6.2.7.3 Estimation des pivots*

Il nous reste maintenant à expliciter la procédure heuristique trouvée par Faloutsos et Lin [22] pour trouver les *pivots* $O_a$ et $O_b$. L'objectif est de trouver un axe ou ligne de projection la plus grande possible (reflétant ainsi l'axe de dispersion maximale des objets). À cette fin, nous avons besoin de choisir $O_a$ et $O_b$ tel que la distance $D(O_a, O_b)$ soit maximale. Cela demanderait une complexité de $O(N_p^2)$. Pour rester dans une complexité linéaire, les auteurs proposent un algorithme heuristique qui

consiste simplement à 1-) choisir arbitrairement un objet dans la base de données et le déclarer comme étant le deuxième *pivot* $O_b$. 2-) Trouver l'objet $O_a$ le plus loin de $O_b$ (selon la distance Euclidienne) puis 3-) trouver l'objet le plus loin de $O_a$ et remplacer $O_b$ par lui. Les étapes 2-) et 3-) peuvent éventuellement être répétées un petit nombre de fois pour améliorer l'estimation mais toutes ces étapes restent en complexité $O(N_p)$ à une constante près.

#### *6.2.7.4 Pseudo-algorithme*

Finalement, la réduction de dimensionnalité d'une séquence d'images par le FastMap se fait donc par l'algorithme récursif suivant:

**Entrées:**
$k$: nombre de dimensions du nouvel espace réduit

*data*: ensemble d'objets $\mathcal{O} = \{O_1,...,O_{N_p}\}$ de taille $N_p$

**Sorties:**
$X_{N_p \times k}$: Matrice de sortie des données réduites

**Init.:** $d \leftarrow 0$

ALGORITHM FASTMAP$(k, D(), \mathcal{O})$

- **if** $k \leq 0$ **then** return $X$
- $d \leftarrow d + 1$
- Trouver les *pivots* $O_a$ et $O_b$

**foreach** *objets i de* $\mathcal{O}$ **do**

- Projection de $O_i$ sur l'axe de coordonné $(O_a, O_b)$

  Calcul $x_i$ en utilisant l'Eq. (6.3) : $X[i, d] = x_i$

**end**

**foreach** *objets i de* $\mathcal{O}$ **do**

- Projection de $O_i$ sur l'hyperplan perpendiculair à

  $(O_a, O_b)$ en utilisant l'Eq. (6.4) $\Rightarrow D'()$

**end**

call FASTMAP$(k - 1, D'(), \mathcal{O})$

**Algorithme 2**: FastMap

#### *6.2.7.5 Calcul de la perte d'information*

Toute réduction de dimensionnalité s'accompagne d'une perte inévitable d'information. Dans le cas du MDS, celle-ci peut être quantifiée à partir de la mesure de corrélation entre les différentes distances Euclidiennes de chaque pair de données non réduites

(appelons $X$ ce vecteur) et les distances Euclidiennes de ces paires de données dans l'espace réduit (soit $Y$ ce vecteur). La corrélation $\rho$ peut ainsi s'estimer par la relation:

$$\rho_{X,Y} = \text{corr}(X,Y) = \frac{\text{cov}(X,Y)}{\sigma_X \, \sigma_Y} = \frac{X^t Y/|X| - \bar{X}\bar{Y}}{\sigma_X \, \sigma_Y} \tag{6.5}$$

dans laquelle $X^t$, $|X|$, $\bar{X}$ et $\sigma_X$ représentent respectivement la transpose, la cardinalité, la moyenne et la l'écart type de $X$. Ce facteur de corrélation (de Pearson) va concrètement quantifier le degré de dépendance linéaire entre les variables $X$ et $Y$ et ainsi quantifier si la technique MDS réussit effectivement à conserver les distances entre paires d'objets dans l'espace réduit. Une corrélation idéale de $\rho = 1$ indique une corrélation ou relation linéaire (positive) parfaite entre les données non réduites et les données réduites (et donc aucune perte d'information) et une corrélation de $\rho = 0$ une perte totale d'information. Une corrélation de $\rho = 0.80$, par exemple, indiquera que la technique MDS utilisée ne réussit à conserver, que seulement 80% des paires d'objets, une distance identique entre paires d'objets dans l'espace non réduit et l'espace réduit. Dans notre application, les taux de corrélation obtenus pour les différentes classes (WALK, RUN , SKIP, JACK, JUMP, PJUMP, SIDE, WAVE-TWO-HANDS, WAVE-ONE-HAND, BEND) et selon chaque point de vue (vue 1, vue 2, vue 3), montrent la proportion de l'information conservée lors de la projection d'un cycle d'action sur les 2 axes considérés, sont les suivants:

| Classe | Vue 1 (%) | Vue 2 (%) | Vue 3 (%) |
|---|---|---|---|
| walk | 83 | 69 | 54 |
| run | 79 | 68 | 56 |
| skip | 83 | 66 | 49 |
| side | 79 | 68 | 57 |
| jack | 80 | 68 | 60 |
| wave1 | 88 | 64 | 56 |
| wave2 | 91 | 63 | 56 |
| jump | 81 | 67 | 51 |
| pjump | 89 | 70 | 58 |
| bend | 80 | 68 | 52 |

**Table 6.1.** *Taux de corrélation obtenus pour les différentes classes (*WALK, RUN ,SKIP, JACK, JUMP, PJUMP, SIDE, WAVE- TWO-HANDS, WAVE-ONE-HAND, BEND*) selon chaque point de vue.*

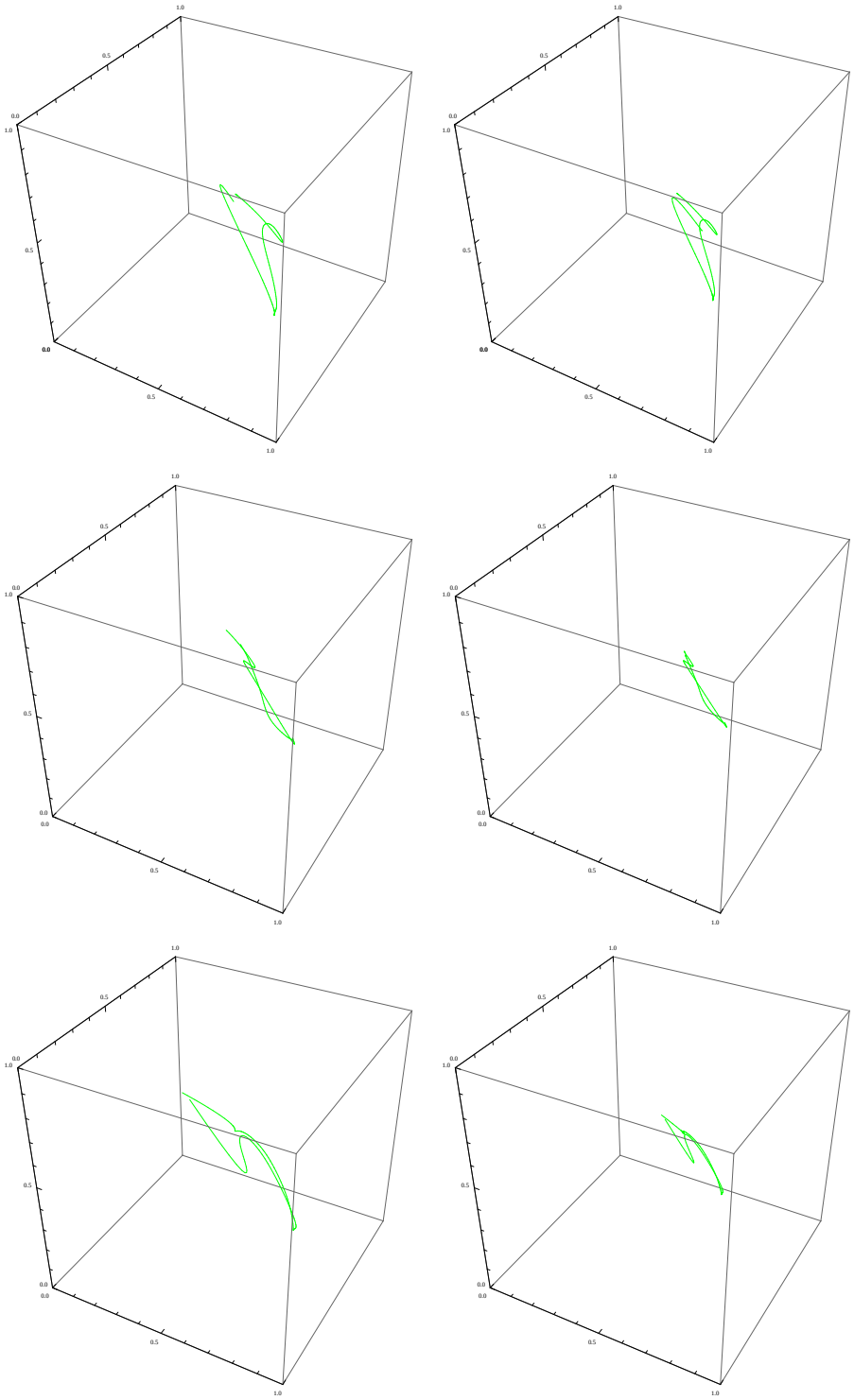


**Figure 6.6.** *Deux prototypes donnés par deux actions similaires selon le point de vue 1 pour les actions:* BEND, WAVE1, WAVE2.

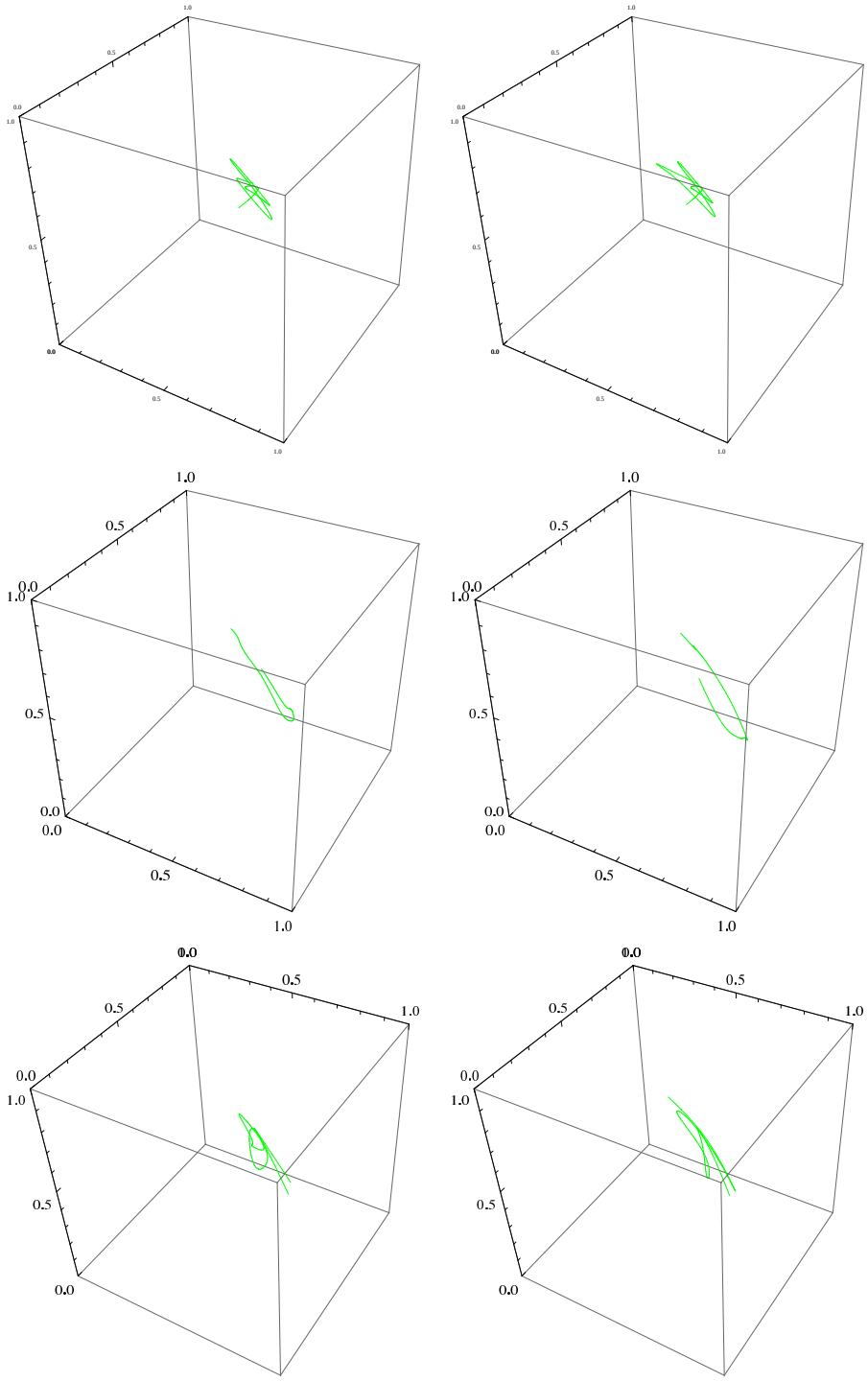


**Figure 6.7.** *Deux prototypes donnés par deux actions similaires selon le point de vue 1 pour les actions:* SKIP, SIDE, RUN.

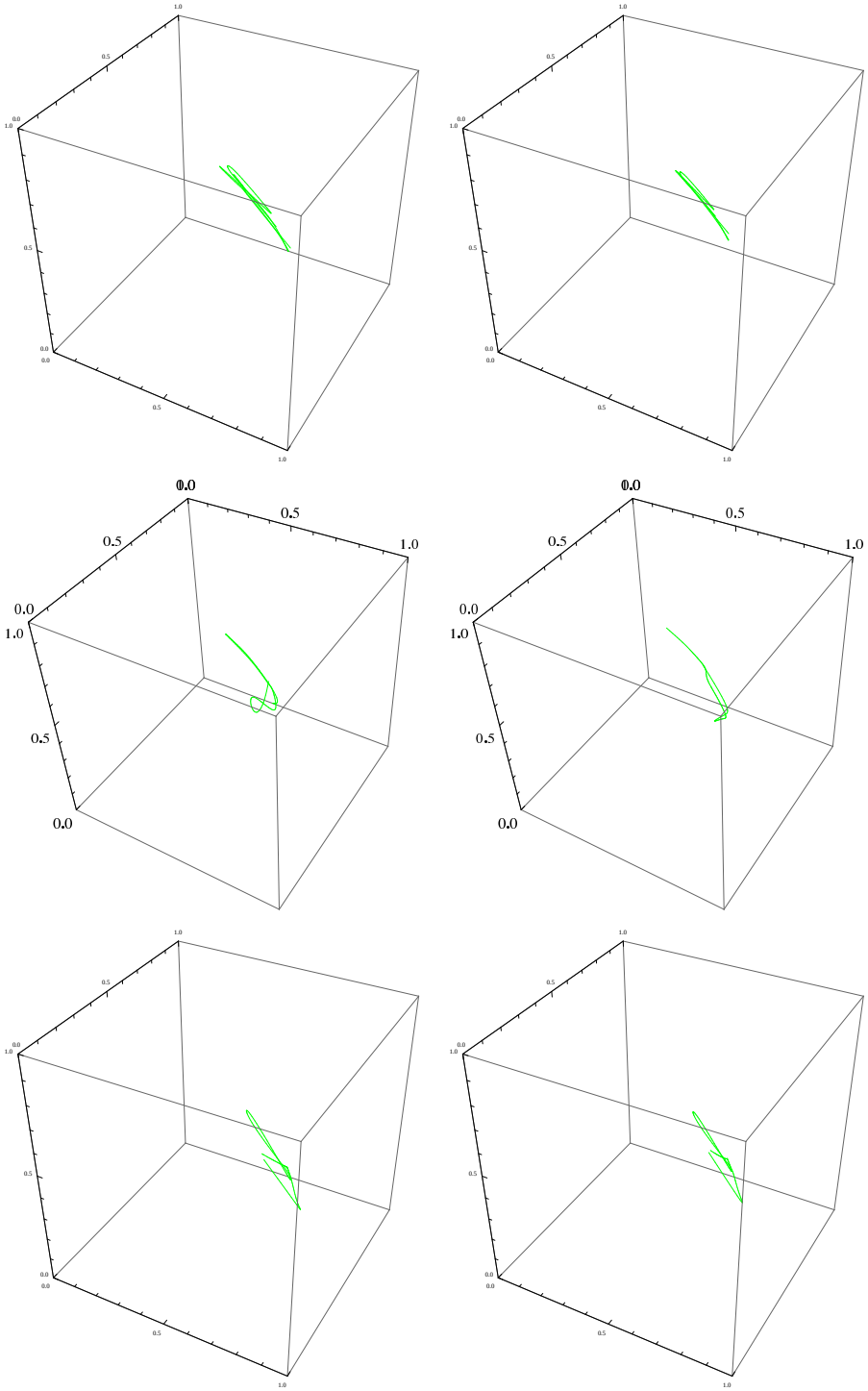


**Figure 6.8.** *Deux prototypes donnés par deux actions similaires selon le point de vue 1 pour les actions:* WALK, JUMP, PJUMP.

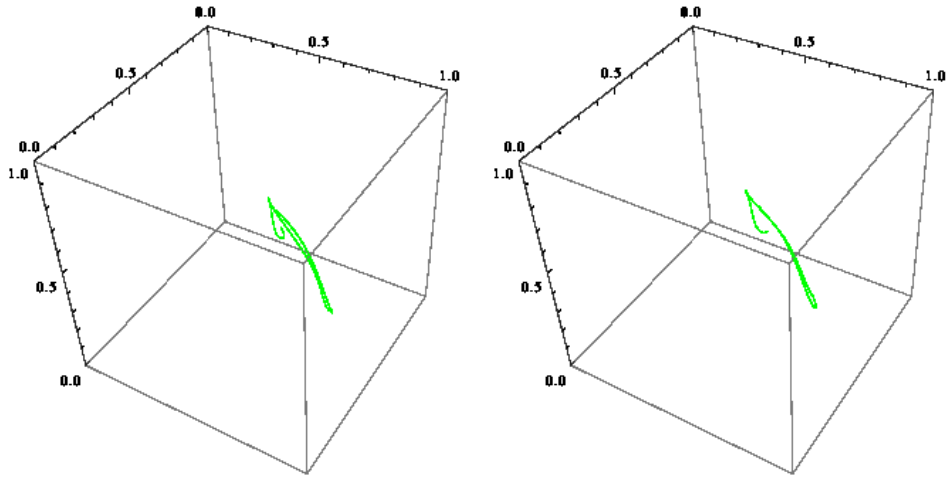



**Figure 6.9.** *Deux prototypes donnés par deux actions similaires selon le point de vue 1 pour l'action:* JACK.

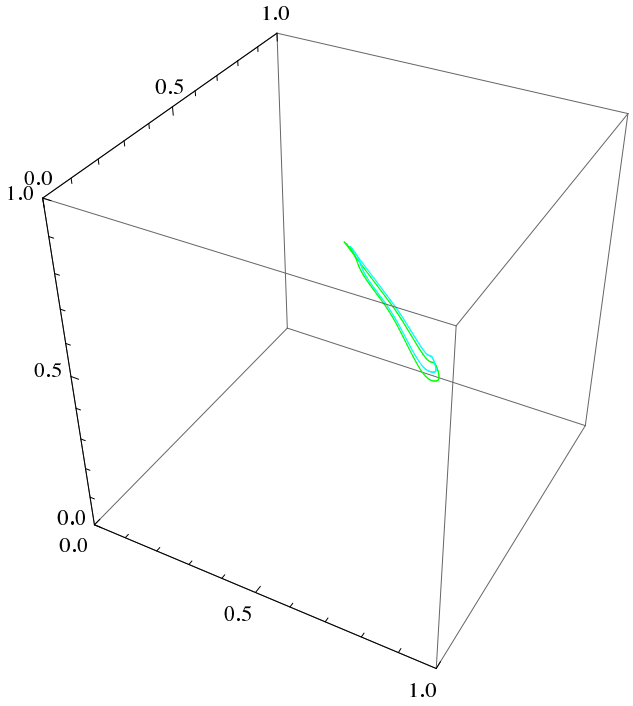



**Figure 6.10.** *Prototype de la classe* SIDE *modélisant deux cycles (ou périodes) d'action.*

### *6.2.8 Classifieur*

La dernière étape de notre système vise à reconnaitre une action (parmi les $m$ existantes) à partir d'une séquence vidéo en se basant sur les prototypes générés dans l'étape précédente. Ces derniers semblent, en effet, visuellement semblables dans chacune des classes, et différents d'une classe à une autre. Pour sa simplicité, nous avons choisi le classifieur $K$-ppv, basé sur une simple distance Euclidienne entre un prototype inconnu et un prototype de la base d'apprentissage, afin de classer une nouvelle entrée. L'apprentissage dans ce cas, consiste à stocker les différents prototypes pour chacune des séquences de notre base d'apprentissage. Rappelons que dans notre application, le premier et le second prototype est respectivement une série de points ordonnés exprimés dans un espace réduit de dimension 3D. Pour le premier prototype, chaque $i-th$ point du prototype de test (à classer) est utilisé comme entré à un $k$-ppv utilisant une distance Euclidienne 3D entre chaque $i-th$ point de chaque protoype de la base d'entrainement. Pour le premier prototype, le résultat de classification de ces $k$-ppv, nous permet de générer un vecteur de score indiquant la classe et le nombre de plus proches voisins du protoype (de la base d'entrainement) possédant le plus grand nombre de plus proche voisins. Ce vecteur de score sera exploité ensuite dans notre procédure de fusion.

## *6.3 La fusion de plusieurs points de vue*

Afin de rendre notre système de reconnaissance plus performant, nous proposons et présentons dans cettes section, une technique simple permettant de fusionner le résultat de classification obtenu par différents prototypes issus d'une même séquence d'action humaine extrait de la base d'apprentissage. Dans le cadre de cette étude, nous avons testé la fiabilité de cette procédure lorsque ces prototypes sont donnés par une réduction de dimensionnalité selon deux points de vue du cube de la séquence d'image vidéo tels que décrit en section 6.2.6 (*i.e.*, selon, l'axe temporel et l'axe

des lignes (cf. Fig. 6.3). La réduction de dimensionnalité selon ces deux de vue différents permettra de représenter un cycle d'action donnée en éliminant la redondance d'information et le bruit de deux façons différentes et complémentaires.

La classification des deux prototypes résultant de ces deux points de vue différents à l'aide de l'algorithme des $K$-ppv permettra d'obtenir deux vecteurs de scores résumant le nombre de plus proches voisins appartenant à chacune des classes (cf. Section précédente). Ces deux vecteurs seront ensuite simplement additionnés et la classe majoritaire de ce vecteur sera aussi la classe qui sera affectée au prototype inconnu. Dans le cas d'une égalité de classe possible, le système lui donnera la classe inconnue et supposera, plus tard, dans le calcul du taux de classification, qu'il s'agit simplement d'une erreur.

## *6.4 Résultats des expérimentations*

Afin d'évaluer et de vérifier l'efficacité de notre système de reconnaissance, nous avons testé notre méthode dans les mêmes conditions communes à celles des articles de référence [1, 3, 26, 27, 35, 48, 63], c'est-à-dire en employant d'une part la base d'actions de Weizamnn et aussi la procédure d'évaluation de validation croisée (*leave-one-out*, cf. section 5.4.1) pour estimer l'erreur de classification. On rappelle que cette procédure consiste à enlever un exemple de la base d'apprentissage, et d'entrainer un classifieur sur les exemples restants. La procédure se répète plusieurs fois. Cela signifie qu'on obtient à chaque itération un nouveau modèle. La matrice de confusion donnée par cette procédure de classification, et dans laquelle chaque colonne de la matrice représente le nombre d'occurrences d'une classe estimée, tandis que chaque ligne représente le nombre d'occurrences d'une classe réelle (le nombre d'actions bien classé correspondent donc aux éléments situés sur la diagonale), permet de mesurer la qualité et la fiabilité du système de reconnaissance. Les matrices de confusion résumées respectivement dans les Tables 6.2, 6.3 et 6.4 pour les trois points de vue

1, 2 et 3 illustrent les différents résultats obtenus avec une valeur de $K$ égale à 1. Pour une valeur de $K = 1$ utilisée dans l'étape de classification (1-ppv), la fusion des points de vue permet d'obtenir la matrice de confusion résumée dans la Table 6.5 que l'on peut comparer avec la matrice de confusion obtenue par les méthodes récentes proposées dans [1, 26, 27, 48, 63] (cf. Tables 6.6, 6.7, 6.8, 6.9 et 6.10). Pour un nombre de $K$ plus proches voisins utilisés dans l'étape de classification égale respectivement à 2, 3, et à l'aide de la base fusionnée, les matrices de confusion obtenues sont résumée dans les Tables 6.11 et Tables 6.12.

| | walk | run | skip | jack | jump | p-jump | side | wave1 | wave2 | bend |
|---|---|---|---|---|---|---|---|---|---|---|
| walk | 89 | | 11 | | | | | | | |
| run | 11 | 67 | 22 | | | | | | | |
| skip | 11 | 22 | 45 | 11 | 11 | | | | | |
| jack | | | | 67 | | | | | 33 | |
| jmup | | | | | 78 | | | 11 | 11 | |
| p-jump | | | | 11 | | 78 | | 11 | | |
| side | | | | | | | 89 | 11 | | |
| wave1 | | | | | | | | 89 | 11 | |
| wave2 | | | | | | | | 33 | 67 | |
| bend | | | | | | | | | 11 | 89 |

**Table 6.2. Matrice de confusion obtenue par notre méthode selon le point de vue** 1**.**

| | walk | run | skip | jack | jump | pjump | side | wave1 | wave2 | bend |
|---|---|---|---|---|---|---|---|---|---|---|
| walk | 78 | 11 | | | | | 11 | | | |
| run | | 89 | 11 | | | | | | | |
| skip | | 22 | 67 | | 11 | | | | | |
| jack | | 11 | | 45 | 11 | | | 11 | 22 | |
| jump | | | 33 | | 34 | 22 | 11 | | | |
| pjump | | | 11 | | 11 | 78 | | | | |
| side | 11 | | | | | | 89 | | | |
| wave1 | | | | | | 22 | | 66 | 22 | |
| wave2 | | | 11 | | | | | 11 | 78 | |
| bend | | | | | | | | | 11 | 89 |

**Table 6.3. Matrice de confusion obtenue par notre méthode selon le point de vue 2.**

| | walk | run | skip | jack | jump | pjump | side | wave1 | wave2 | bend |
|---|---|---|---|---|---|---|---|---|---|---|
| walk | 67 | 22 | | 11 | | | | | | |
| run | | 22 | 11 | 34 | 22 | | | | 11 | |
| skip | | | 34 | | 33 | | 22 | | 11 | |
| jack | 34 | 22 | | 22 | | 11 | 11 | | | |
| jump | | | 22 | | 34 | 11 | 22 | | 11 | |
| pjump | | | | | | 34 | 11 | 22 | 22 | 11 |
| side | | | 11 | | 11 | | 78 | | | |
| wave1 | | | | | 11 | 22 | | 34 | 33 | |
| wave2 | | 11 | 11 | | | 22 | | 56 | 0 | |
| bend | | | | | | 11 | | | 11 | 78 |

**Table 6.4. Matrice de confusion obtenue par notre méthode selon le point de vue 3.**

| | walk | run | skip | jack | jump | pjump | side | wave1 | wave2 | bend |
|---|---|---|---|---|---|---|---|---|---|---|
| walk | 100 | | | | | | | | | |
| run | | 100 | | | | | | | | |
| skip | | 11 | 89 | | | | | | | |
| jack | | | | 78 | | | | | 22 | |
| jump | | | | | 78 | | | 11 | 11 | |
| pjump | | | 11 | | | 89 | | | | |
| side | | | | | | | 100 | | | |
| wave1 | | | | | | | | 100 | | |
| wave2 | | | | | | | | 11 | 89 | |
| bend | | | | | | | | | | 100 |

**Table 6.5. Matrice de confusion obtenue par notre méthode après fusion des deux prototypes pour $K = 1$.**

| | walk | run | skip | jack | jump | p-jump | side | wave1 | wave2 | bend |
|---|---|---|---|---|---|---|---|---|---|---|
| walk | 89 | 11 | | | | | | | | |
| run | | 80 | 10 | | 10 | | | | | |
| skip | | 30 | 50 | | 20 | | | | | |
| jack | | | | 100 | | | | | | |
| jmup | | 11 | 11 | | 67 | | 11 | | | |
| p-jump | | | | | | 100 | | | | |
| side | | | | | | | 100 | | | |
| wave1 | | | | | | | | 78 | 22 | |
| wave2 | | | | | | | | 22 | 78 | |
| bend | | | | | | | | | | 100 |

**Table 6.6. Matrice de confusion obtenue par Scovanner *et al.* [63].**

| | walk | run | skip | jack | jump | p-jump | side | wave1 | wave2 | bend |
|---|---|---|---|---|---|---|---|---|---|---|
| walk | 100 | | | | | | | | | |
| run | 7 | 76 | 13 | | | | 4 | | | |
| skip | | 16 | 64 | | 20 | | | | | |
| jack | | | | 100 | | | | | | |
| jmup | | 3 | | | 89 | | 5 | | | 3 |
| p-jump | | | | 2 | | 98 | | | | |
| side | 2 | | | | | 19 | 79 | | | |
| wave1 | | | | | | 20 | | 80 | | |
| wave2 | | | | | | | | 2 | 98 | |
| bend | | | | | | | | | | 100 |

**Table 6.7. Matrice de confusion obtenue par Kui *et al.* [35].**

| | walk | run | skip | jack | jump | pjump | side | wave1 | wave2 | bend |
|---|---|---|---|---|---|---|---|---|---|---|
| walk | 80 | 10 | | | 10 | | | | | |
| run | | 100 | | | | | | | | |
| skip | | | | | | | | | | |
| jack | | | | 100 | | | | | | |
| jump | | | 22 | | 78 | | | | | |
| pjump | | | | | | 100 | | | | |
| side | 11 | | | | | | 89 | | | |
| wave1 | | | | | | | | 100 | | |
| wave2 | | | | | | | | | 100 | |
| bend | | | | | | | | | | 100 |

**Table 6.8. Matrice de confusion obtenue par Grundmann *et al.* [27].**

| | walk | run | skip | jack | jump | pjump | side | wave1 | wave2 | bend |
|---|---|---|---|---|---|---|---|---|---|---|
| walk | 100 | | | | | | | | | |
| run | | 98 | 2 | | | | | | | |
| skip | | 2.9 | 97.1 | | | | | | | |
| jack | | | | 100 | | | | | | |
| jump | | | 10.8 | | 89.2 | | | | | |
| pjump | | | | | | 100 | | | | |
| side | | | | | | | 100 | | | |
| wave1 | | | | 0.9 | | 0.9 | | 94.8 | 3.5 | |
| wave2 | | | | 0.9 | | | | 1.9 | 97.2 | |
| bend | | | | | | | | | 0.9 | 99.1 |

**Table 6.9. Matrice de confusion obtenue par Gorelick *et al.* [26].**

| | walk | run | skip | jack | jump | pjump | side | wave1 | wave2 | bend |
|---|---|---|---|---|---|---|---|---|---|---|
| walk | 100 | | | | | | | | | |
| run | | 100 | | | | | | | | |
| skip | | | 100 | | | | | | | |
| jack | | | | 1000 | | | | | | |
| jump | | | | | 94 | | | | | |
| pjump | | | | | | 100 | | | | |
| side | | | | | | | 99 | | | |
| wave1 | | | | | | | | 100 | | |
| wave2 | | | | | | | | | 100 | |
| bend | | | | | | | | | | 100 |

**Table 6.10. Matrice de confusion obtenue par Fathi *et al.* [1].**

| | walk | run | skip | jack | jump | pjump | side | wave1 | wave2 | bend |
|---|---|---|---|---|---|---|---|---|---|---|
| walk | 89 | | 11 | | | | | | | |
| run | 22 | 56 | 22 | | | | | | | |
| skip | 22 | 22 | 23 | | 22 | 11 | | | | |
| jack | | | | 66 | | | | | 34 | |
| jump | | | | | 78 | | | | 22 | |
| pjump | | | | | 11 | 89 | | | | |
| side | | | 11 | | | | 89 | | | |
| wave1 | | | | | | | | 89 | 11 | |
| wave2 | | | | | | | | 11 | 89 | |
| bend | | | | | | | | | | 100 |

**Table 6.11. Matrice de confusion obtenue par notre méthode après fusion des deux prototypes pour $K = 2$.**

| | walk | run | skip | jack | jump | pjump | side | wave1 | wave2 | bend |
|---|---|---|---|---|---|---|---|---|---|---|
| walk | 100 | | | | | | | | | |
| run | 11 | 56 | 22 | | 11 | | | | | |
| skip | 22 | 22 | 23 | | 11 | 11 | | | 11 | |
| jack | | | | 67 | | | 11 | | 22 | |
| jump | | | | | 78 | | | | 22 | |
| pjump | | | | | 11 | 89 | | | | |
| side | 11 | | | | | | 89 | | | |
| wave1 | | | | | | | | 100 | | |
| wave2 | | | | | | | | | 100 | |
| bend | | | | | | | | | | 100 |

**Table 6.12. Matrice de confusion obtenue par notre méthode après fusion des deux prototypes pour $K = 3$.**

Finalement, le tableau 6.13 montre la précision obtenue par chaque expérience, étant donné que la précision est déterminée par le rapport du nombre d'actions correctement reconnues sur le nombre total d'actions de la base. La performance de notre système de reconnaissance en fonction du paramètre $K$ est illustré par la Figure 6.11. Finalement, le tableau comparatif 6.14 illustre la précision de notre système de reconnaissance et la compare à d'autres méthodes récemment publiées [1, 3, 26, 27, 35, 48, 63].

| | Point de vue 1 | Point de vue 2 | Fusion | | |
|---|---|---|---|---|---|
| Nombre de k-ppv | k=1 | k=1 | k=1 | k=2 | k=3 |
| Taux de reconnaissance | 75.8 | 71.3 | **92.3** | 76.8 | 80.2 |

**Table 6.13. Un tableau résumant nos taux de reconnaissance pour les différentes stratégies envisagées dans le cadre de cette étude.**

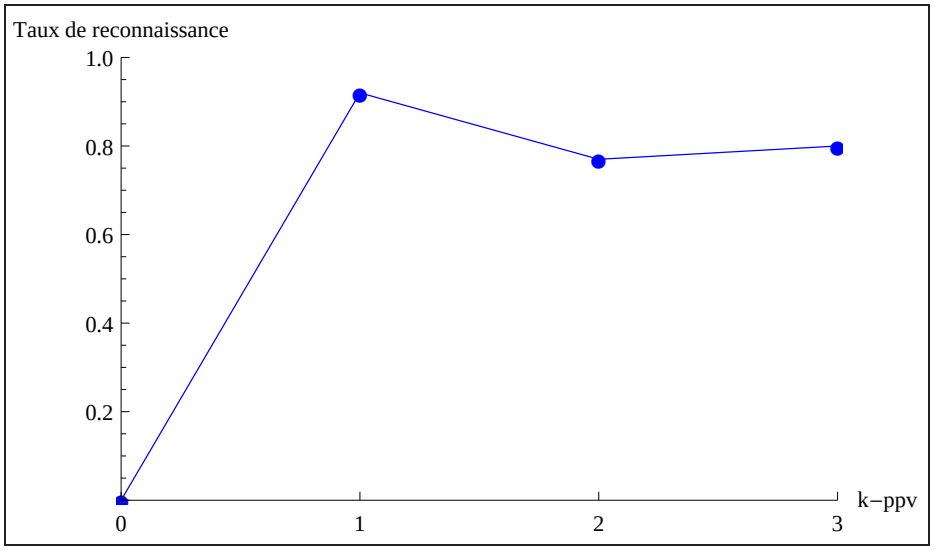



**Figure 6.11.** *Évolution du taux de reconnaissance en fonction du paramètre $K$.*

| Méthode | Précision |
|---|---|
| Notre méthode | **92.3%** |
| Fathi et *et al.* [2008] | 99.9% |
| Gorelick *et al.* [2007] | 97.8% |
| Grundmann *et al.* [2008] | 94.6% |
| Jia et *et al.* [2008] | 90.9% |
| Klaser *et al.* [2008] | 84.3% |
| Scovanneret *et al.* [2007] | 82.6% |
| Niebles *et al.* [2007] | 72.8% |

**Table 6.14. Un tableau comparant notre taux de reconnaissance avec d'autres méthodes récemment publiées [1, 3, 26, 27, 35, 48, 63].**

## *6.5 Remarques*

D'autres expériences et stratégies ont été effectuées sur la base d'actions de Weizmann qui nous ont permis d'élaborer les remarques intéressantes suivantes:

- Les résultats obtenus pour les deux points de vue 1 et 2 avec une valeur de $k = 2, 3$ sont aussi bons.

- Modéliser les actions de la base selon le point de vue 2 où en se basant sur plusieurs couches composées par une série d'images dans la direction latérale du cube permet d'obtenir des résultats satisfaisant pour ce point de vue.

- Modéliser les actions de la base par des prototypes générés selon le troisième point de vue (cf. Fig. 6.3) ne permet pas d'obtenir de très bons résultats de classement et la fusion en considérant ces trois prototypes dégradent nos résultats de classification. Le taux de reconnaissance obtenu selon le point de

vue 3 est moins bon, égale à 40.3 pourcent. Dans notre application, on pense que la réduction selon le point de vue 3 nous permet pas de distinguer l'information de mouvement du bras et de la jambe de la droite vers la gauche.

- Modéliser les actions de la base par des prototypes résultant de deux points de vue différents (vue 1 et vue 2) en se basant uniquement sur l'information de mouvements (différence de paires de trames) permet d'obtenir des taux de reconnaissance moins bon.

- L'ajout de prototypes basés sur l'information de mouvements (différence de paires de trames) ne permet pas d'améliorer les résultats. De même, vraisemblablement à cause de ce fait; une distance entre deux prototypes basée sur la différence des positions de ces points et la différence de deux points successifs (*i.e.*, vecteurs directeurs de chaque couple de points successifs) ne permet pas d'améliorer les résultats de classification.

- Translater l'ensemble des prototypes résultant de nos deux points de vue différents, pour que ceux-ci débutent tous initialement à partir du point d'origine $(0, 0, 0)$ de notre espace tridimensionnel permet d'obtenir sensiblement les mêmes résultats de classification et ceci pour les différentes expériences réalisées.

- Modéliser chaque classe par un modèle représentatif résultant de la moyenne des différents prototypes permet d'obtenir des résultats bons mais pas aussi bon que les résultats obtenus sans cette modélisation.

## *6.6 Discussion*

L'analyse de la matrice de confusion nous permet de constater qu'on peut reconnaitre tous les types d'actions existant dans la base. Le taux de reconnaissance de chaque classe est variable d'une classe à l'autre avec un taux de reconnaissance performant

pour notre stratégie. On trouve que l'erreur de classement dans chaque classe varie selon le degré de ressemblance avec les autres classes. Cela signifie que les actions mal classées par notre système se trouvent en général et très logiquement dans les classes les plus proches de la classe de test (c'est le cas par exemple de la classe WAVE1 ou WAVE2 qui mécaniquement et visuellement reste très similaire à la classe d'action JUMP ou encore de la classe SKIP et RUN). Dans notre application, on constate aussi que les résultats obtenus selon chaque point de vue (vue 1 et vue 2) sont bons et semblent effectivement complémentaires. On constate que les résultats obtenus selon le point de vue 3 sont moins bons.

## *6.7 Conclusion*

Dans ce chapitre, nous avons présenté un système original de reconnaissance d'actions humaines basé sur la définition de deux prototypes générés par une technique de réduction de dimensionnalité selon deux points de vue du cube des données associé à une séquence d'images à classer. Une présentation détaillée des différentes étapes de notre système a été faite. Notre méthode a ensuite été testé sur la célèbre base de Weizmann, suivie d'une comparaison avec d'autres approches existant dans la littérature [1, 3, 26, 27, 35, 48, 63]. Notre méthode est à la fois simple et très efficace avec un taux de reconnaissance performant par rapport à d'autres techniques qui utilisent des caractéristiques spatio-temporelles et des méthodes de classification plus complexes pour classer les actions [1, 3, 26, 27, 35, 48, 63]. À notre connaissance, cette étude est la première qui exploite conjointement, avec une réduction de dimensionnalité classique pour une séquence d'images, un prototype généré aussi selon une direction du cube des données (d'une séquence d'images) qui soit autre que temporelle.

Chapitre 7

# CONCLUSION GÉNÉRALE ET PERSPECTIVES

Dans ce mémoire de maîtrise, nous avons présenté notre contribution à la recherche concernant l'étude d'un système de reconnaissance automatique d'actions humaines par vision par ordinateur. La plupart des études publiées sur ce sujet exploitent une représentation ou des caractéristiques spatio-temporelle de l'action, c'est-à-dire utilisent des caractéristiques calculées spatialement et/ou temporellement sur l'ensemble des images de la séquence lues temporellement mais aucune étude, à notre connaissance, exploite conjointement cette représentation classique avec un prototype généré aussi selon une direction du cube des données (d'une séquence d'images) qui soit autre que temporelle. La similitude entre les actions dans chacune des classes et la redondance inhérente de l'information nous a incité à choisir un prototype basé sur une technique de réduction de dimensionnalité non linéaire et un algorithme de classification non paramétrique basée sur les $K$-ppv. Le système de reconnaissance d'actions humaines proposé, basées sur la fusion de classification de deux prototypes générés à partir de deux points de vue différents, nous a permis l'obtention de taux de reconnaissance performant de 92.3% tout en utilisant un classifieur très simple ($K$-ppv). Les deux prototypes générés par notre stratégie, dans le cadre d'une reconnaissance d'actions humaines sont très complémentaires comme nous l'indiquent les taux de classification obtenus pour les deux points de vue, respectivement de 75.8% et 71.3%. La fusion de ces deux points de vue est aussi une originalité de notre recherche.

L'évaluation de notre modèle porte sur une base d'actions constituée de 10 classes ou actions différentes et réalisées successivement par 9 personnes. Toutes les actions ont été utilisées pour calculer le taux de reconnaissance en employant la technique

de validation croisée *leave-one-out*. Le système proposé permet de reconnaitre correctement un nombre important de la base fournie ainsi les erreurs produites par le système. Les résultats de classement sont intéressants avec un taux de reconnaissance variable mais performant selon le nombre de $K$ plus proches voisins utilisés ($K$ varie de 1 à 3 pour les différents tests effectués).

L'avantage de notre système se manifeste par sa simplicité et son efficacité pour donner une meilleure représentation de l'action. Fondée sur les résultats obtenus en fonction du paramètre $K$, une piste est envisagée pour améliorer l'efficacité de ce système. D'une part, on peut envisager une autre technique de classification plus complexe en employant un classifieur plus performant comme le SVM ou les réseaux de neurones visant à apprendre une fonction linéaire, ou non linéaire afin de mieux détecter et discriminer entre les différentes classes.

# RÉFÉRENCES


[1] A.Fathi et G.Mori. Action recognition by learning mid-level motion features. Dans *Proceedings of CVPR'08*. IEEE Computer Society, 2008.

[2] M.E. Ahmed, H. David, et L.S. Davis. Non-parametric model for background subtraction. Dans *Proceedings of the 6th European Conference on Computer Vision, ECCV'00*, volume 2, pages 751–767, London, UK, 2000. Springer-Verlag.

[3] A.Kläser, M. Marszalek, et C. Schmid. A spatio-temporal descriptor based on 3D-gradients. Dans *Proceedings of BMVC'08*, pages 1–10, 2008.

[4] S. Ali et M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(2):288 – 303, 2010.

[5] J. Anil et Z. Douglas. Feature selection: Evaluation, application, and small sample performance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(2):153–158, Février 1997.

[6] H. Bay, T. Tuytelaars, et L. Van Gool. SURF: Speeded up robust features. Dans *Proceedings of the 2006 International Conference on Computer Vision, ECCV'06*, pages 404–417, 2006.

[7] M. Beatty et B.S. Manjunat. Dimensionality reduction using multi-dimensional scaling for content-based retrieval. *IEEE Proceedings of the International Conference on Image Processing, ICIP'97*, 2:835 –838, Octobre 1997.

[8] S. Ben-yacoub, B. Fasel, et J. Lüttin. Fast face detection using MLP and FFT. Dans *Proccedings of the Second International Conference on Audio and Video-based Biometric Person Authentication, AVBPA'99*, pages 31–36, 1999.

[9] S. Bernhard, S. Alexander, et M. Klaus-Robert. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10(5):1299–1319, juillet 1998.

[10] M. Blanc, L. Gorelick, E. Shechtman, M. Irania, et R. Basri. Actions as space-time shapes. *Proceedings of the Tenth IEEE International Conference on Computer Vision, ICCV'05*, 2:1395 – 1402, 2005.

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

[12] I. Borg et P.J.F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005.

[13] M.L. Carlos, B. Lluís, et N. Àngela. Feature selection algorithms: A survey and experimental evaluation. Dans *Proceedings of the 2002 IEEE International Conference on Data Mining, ICDM '02*, pages 306–314, Washington, DC, USA, 2002. IEEE Computer Society.

[14] S. Christian, L. Ivan, et C. Barbara. Recognizing human actions: A local svm approach. Dans *Proceedings of the 17th International Conference on Pattern Recognition, ICPR'04*, volume 3, pages 32–36, Washington, DC, USA, 2004. IEEE Computer Society.

[15] C. Chun-houh, H. Wolfgang, et U. Antony. *Handbook of Data Visualization (Springer Handbooks of Computational Statistics)*. Springer-Verlag TELOS, Santa Clara, CA, USA, 1 édition, 2008.

[16] Y.L. Cun, J.S. Denker, et S.A. Solla. *Advances In Neural Information Processing Systems 2*, chapitre Optimal Brain Damage, pages 598–605. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[17] M. Dash et H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1:131–156, 1997.

[18] L.J.P. Van der Maaten, E.O. Postma, et H.J. Van den Herik. Dimensionality reduction: A comparative review. Rapport technique, MICC, Maastricht University, P.O. Box 616, 6200 MD Maastricht, Netherlands, Février 2008.

[19] O. Due et A.K. Jain. Goal-directed evaluation of binarization methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(12):1191–1201, Décembre 1995.

[20] D. Engel, L. Hüttenberger, et B. Hamann. A survey of dimension reduction methods for high-dimensional data analysis and visualization. Dans *VLUDS*, pages 135–149, 2011.

[21] S.Z. Erdogan, T.T. Bilgin, et J. Cho. Fall detection by using k-nearest neighbor algorithm on WSN data. Dans *IEEE, Globecom Workshops on Advances in Communications and Networks*, pages 2054 – 2058, 2010.

[22] C. Faloutsos et K. Lin. Fastmap: a fast algorithm for indexing, data-mining and visualization. Dans *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD'95*, pages 163–174, May 1995. source code of FastMap available on his webpage.

[23] P. Fihl et T.B. Moeslund. Recognizing human gait types. Dans *Robot Vision*, pages 183–208. InTech, 2010.

[24] F.Imola. A survey of dimension reduction techniques, 2002.

[25] Y. Freund. The alternating decision tree learning algorithm. Dans *Proceedings of the Sixteenth International Conference in Machine Learning, ML'99*, pages 124–133. Morgan Kaufmann, 1999.

[26] L. Gorelick, M. Blank, E. Shechtman, M. Irani, et R. Basri. Actions as space-time shapes. *IEEE Trans. on Pattern Anal. Mach. Intell.*, 29(12):2247–2253, Décembre 2007.

[27] M. Grundmann, F. Meier, et I. Essa. 3d shape context and distance transform for action recognition. Dans *Proceedings of ICPR'08*, 2008.

[28] Y. Guoliang, L. Huan, Z. Li, et C. Yue. Research on a skin color detection algorithm based on self-adaptive skin color model. Dans *Proceedings of the 2010 International Conference on Communications and Intelligence Information Security, ICCIIS'10*, pages 266–270, Washington, DC, USA, 2010. IEEE Computer Society.

[29] D. Gutchess, E. Cohen-solal, D. Lyons, et A.K. Jain. A background model initialization algorithm for video surveillance. Dans *IEEE Proccedings of the Conference ICCV'2001*, pages 733–740, 2001.

[30] A. Hadid et M. Pietikäinen. Selecting modelsffrom videos for appearance-based face recognition. Dans *Procceedings ICPR'04*, volume 1, pages 304–308, 2004.

[31] A.M. Hani, H.A. Nugroho, et H .Nugroho. Gaussian bayes classifier for medical diagnosis and grading: Application to diabetic retinopathy. *Proccedings of the Conference on Biomedical Engineering and Sciences, EMBS'10*, pages 52 – 56, Novembre 2010.

[32] W.X. Hui, S. Ping, C. Li, et W. Ye. A ROC curve method for performance evaluation of support vector machine with optimization strategy. Dans *Proceedings*

*of the International Forum on Computer Science-Technology and Applications, IFCSTA'09*, volume 2, pages 117–120, Washington, DC, USA, 2009. IEEE Computer Society.

[33] L. Jeisung et P. Mignon. An adaptive background subtraction method based on kernel density estimation. *Sensors*, 12(9):12279–12300, 2012.

[34] H. Jhuang, T. Serre, L. Wolf, et T. Poggio. A biologically inspired system for action recognition. Dans *IEEE Proceedings of the 11th International Conference on Computer Vision, ICCV'07*, pages 1–8, Rio de Janeiro, Brazil, Octobre 2007. IEEE.

[35] K. Jia et D.Y. Yeung. Human action recognition using local spatio-temporal discriminant embedding. *IEEE Proceedings of the Conference on Computer Vision and Pattern Recognition, CVPR'08*, pages 1–8, 2008.

[36] L. Jinxia et Q. Yuehong. Application of SIFT feature extraction algorithm on the image registration. *Proceedings of the 10th International Conference on Electronic Measurement and Instruments, ICEMI'11*, Juin 2011.

[37] S.B. Kotsiantis. Supervised machine learning: A review of classification techniques. informatica 31:249–268, 2007.

[38] S. Krishnamachari et M.A. Mottaleb. Image browsing using hierarchical clustering. Dans *Proceedings ISCC'99*, pages 301–307, 1999.

[39] W. Kusakunniran, Q. Wu, J. Zhang, et H. Li. Multi-view gait recognition based on motion regression using multilayer perceptron. Dans *Proceedings of the 2010 20th International Conference on Pattern Recognition, ICPR'10*, pages 2186–2189, Washington, DC, USA, 2010. IEEE Computer Society.

[40] J.T. Kwok, B. Mak, et S. Ho. Eigenvoice speaker adaptation via composite kernel PCA. Dans *NIPS*, 2003.

[41] J.T.Y. Kwok. Automated text categorization using support vector machine. Dans *Proceedings of the International Conference on Neural Information Processing, ICONIP'98*, pages 347–351, 1998.

[42] L. Ladha et T. Deepa. Feature selection methods and algorithms. *International Journal on Computer Science and Engineering, IJCSE'11*, 3(5):1787 – 1797, 2011.

[43] H.A. Mark et S.A. Lloyd. Feature subset selection: A correlation based filter approach. Dans *1997 International Conference on Neural Information Processing and Intelligent Information Systems*, pages 855–858. Springer, 1997.

[44] A. McCallum et K. Nigam. A comparison of event models for naive bayes text classification, 1998.

[45] S.J. Mckenna, S. Jabri, Z. Duric, A. Rosenfeld, et H. Wechsler. Tracking groups of people. *Computer Vision and Image Understanding*, 80:42–56, 2000.

[46] R. Muralidharan et C. Chandrasekar. Object recognition using SVM-KNN based on geometric moment invariant. *International Journal of Computer Trends and Technology*, 2011.

[47] P.M. Narendra et K. Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Trans. Comput.*, 26(9):917–922, Septembre 1977.

[48] J.C. Niebles et L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. Dans *Proceedings of IEEE International Conference in Computer Vision and Pattern Recognition, CVPR'07.*, 2007.

[49] L.S. Oliveira, N. Benahmed, R. Sabourin, F. Bortolozzi, et C.Y. Suen. Feature subset selection using genetic algorithms for handwritten digit recognition, 2001.

[50] N.M. Oliver, B. Rosario, et A. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Trans. Pattern Anal. and Mach. intell.*, 22(8):831–843, 2000.

[51] M. Oravec et J. PavloviEov. Face recognition methods based on principal component analysis and feedfonvard neural networks. *IEEE International Joint Conference on Neural Networks*, 4, Décembre 2004.

[52] P. Perner et A. Imiya, editeurs. *Machine Learning and Data Mining in Pattern Recognition, 4th International Conference, MLDM'05*, volume 3587 de *Lecture Notes in Computer Science*, Leipzig, Germany, July 2005. Springer.

[53] Platt et C. John. Fastmap, metricmap, and landmark MDS are all nystrom algorithms. Dans *Artificial Intelligence and Statistics, AISTATS'05)*, 2005.

[54] M. Pontil et A. Verri. Support vector machines for 3D object recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(6):637–646, Juin 1998.

[55] K.L. Priddy et P.E. Keller. *Artificial Neural Networks: An Introduction (Chapitre 4)*. The international society for optical engineering, Bellingham, Washigton, USA, 2005.

[56] P. Pudil, F.J. Ferri, J. Novovicova, et J. Kittler. Floating search methods for feature selection with nonmonotonic criterion functions. Dans *Proceedings of the Twelveth International Conference on Pattern Recognition, IAPR'94*, pages 279–283, 1994.

[57] Z. Qin, G. Chun-hua, et L. Jia-jun. Features selection for intrusion detection systems based on support vector machines. Dans *IEEE Conference on Consumer Communications and Networking Conference, CCNC'06*, pages 1 – 5. IEEE Press, 2006.

[58] A. Ranjan. Using a KNN and MOG based algorithm for static hand posture recognition.

[59] T.S. Roweis et L.k. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, Décembre 2000.

[60] Y. Sakakibara, K. Misue, et T. Koshiba. Text classification and keyword extraction by learning decision trees. Dans *Proccedings of the Ninth Conference on Artificial Intelligence for Applications, CAIA'93*, page 466, Orlando, FL, 1993.

[61] L. K. Saul et S. T. Roweis. An introduction to locally linear embedding. Rapport technique, AT&T Labs, 2000.

[62] C. Schuldt, L. Ivan, et C. Barbara. Recognizing human actions: A local SVM approach. Dans *Proceedings of the 17th International Conference on Pattern Recognition, ICPR'04*, volume 2 de *ICPR '04*, pages 32–36, Washington, DC, USA, 2004. IEEE Computer Society.

[63] P. Scovanner, A. Saad, et S. Mubarak. A 3-dimensional SIFT descriptor and its application to action recognition. Dans *Proceedings of the 15th international conference on Multimedia, MULTIMEDIA'07*, pages 357–360, New York, NY, USA, 2007. ACM.

[64] V.D. Silva et J.B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. Dans S. Becker, S. Thrun, et K. Obermayer, editeurs, *NIPS*, pages 705–712. MIT Press, 2002.

[65] C. Stauffer et W.E.L. Grimson. Adaptive background mixture models for real-time. Dans *IEEE Proceedings of the Conference on Computer Vision and Pattern Recognition, CVPR'99*, Ft. Collins, CO, USA, Juin 1999. IEEE Computer Society.

[66] D. L. Swets et J. J. Weng. Efficient content-based image retrieval using automatic feature selection. Dans *IEEE International Symposium of Computer Vision*, pages 85–90, 1995.

[67] J.B. Tenenbaum, V. Silva, et J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, Décembre 2000.

[68] G. Vilas, S. Vijander, et K. Mahendra. Image compression using pca and improved technique with mlp neural network. Dans *Proceedings of the 2010 International Conference on Advances in Recent Technologies in Communication and Computing, ARTCOM '10*, pages 106–110, Washington, DC, USA, 2010. IEEE Computer Society.

[69] P. Shen-Pei Wang. 3D body-joints based humain action recognition. Dans *Pattern Recognition and Machine Vision*, pages 121–132. river, 2010.

[70] P. Wu, B.S. Manjunath, et H.D. Shin. Dimensionality reduction for image retrieval. *IEEE Proceedings of the International Conference on Image Processing, ICIP'00*, 3, 2000.

[71] C. Xianyi et G. Xiangpu. An image segmentation of fuzzy c-means clustering based on the combination of improved ant colony algorithm and genetic algorithm. Dans *Proceedings of the 2008 International Workshop on Education Technology and Training & 2008 International Workshop on Geoscience and Remote*

*Sensing, ETTANDGRS'08*, volume 2, pages 804–808, Washington, DC, USA, 2008. IEEE Computer Society.

[72] M. Hsuan Yang. Face recognition using extended isomap. Dans *Procceedings ICIP'02*, pages 117–120, 2002.

[73] L. Yu et H. Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. Dans *Proceedings of the Twentieth International Conference on Machine Learning*, pages 856–863. AAAI Press, 2003.

[74] M. Yu, A. Rhuma, S. Naqvi, L. Wang, et J. Chambers. Posture recognition based fall detection system for monitoring an elderly person in a smart home environment. *IEEE Trans. Inf. Technol. Biomed.*, 2012.

# ANNEXE

## MDS-Based Multi-Axial Dimensionality Reduction Model For Human Action Recognition

Redha Touati and Max Mignotte
*Département d'Informatique et de Recherche Opérationnelle (DIRO)*
*Université de Montréal, Faculté des Arts et des Sciences,*
*Montréal H3C 3J7 QC, Canada*
*Email: touatire@iro.umontreal.ca*

***Abstract*—In this paper, we present an original and efficient method of human action recognition in a video sequence. The proposed model is based on the generation and fusion of a set of prototypes generated from different view-points of the data cube of the video sequence. More precisely, each prototype is generated by using a multidimensional scaling (MDS) based nonlinear dimensionality reduction technique both along the temporal axis but also along the spatial axis (row and column) of the binary video sequence of 2D silhouettes. This strategy aims at modeling each human action in a low dimensional space, as a trajectory of points or a specific curve, for each viewpoint of the video cube in a complementary way. A simple $K$-NN classifier is then used to classify the prototype, for a given viewpoint, associated with each action to be recognized and then the fusion of the classification results for each viewpoint allow us to significantly improve the recognition rate performance. The experiments of our approach have been conducted on the publicly available Weizmann data-set and show the sensitivity of the proposed recognition system to each individual viewpoint and the efficiency of our multi-viewpoint based fusion approach compared to the best existing state-of-the- art human action recognition methods recently proposed in the literature.**

***Keywords*-Human action recognition; Multi-axial reduction of dimensionality; Gesture recognition; Multidimensional scaling; FastMap; Weizmann data-set; $K$-Nearest Neighbor.**

### I. Introduction

The proliferation of video content on the web or in everyday life makes human action recognition, one of the key prerequisites for video analysis and understanding with many important computer vision applications, such as video surveillance, indexing and browsing, human-computer interfacing, recognition of gesture and analysis of sport events, etc. [1], [2], [3], [4], [5], [6].

The goal of any unsupervised human recognition system is to be able to automatically recognize low-level actions such as running, walking, hand clapping, etc. from an input video sequence and the main difficulty of this human motion categorization [7] lies in representing the different types of human motion with effective models both taking into account the intra-class variations in appearance and size of different individuals, and between-class variations, *i.e.,* in different action types with similar body shapes.

Various approaches for human action recognition have been already proposed in the literature, and a way to classify them into several categories may be considered depending on the type of (*e.g.,* static, dynamic or spatial-temporal) features extracted from the spatial temporal information of the video sequence and intended to model the human action *via* the local or global description of the (spatial) human body information and its (temporal) motion information [1].

Some approaches rely on local features to represent the motion patterns and to capture local events in video. In this way, Schuldt *et al.* [3] have used the spatial Harris 3D detector. Building on the success of the histogram of gradient (HOG) based descriptor for static images, an extension of the SIFT descriptor to 3D was proposed in [9] as a new local spatial-temporal descriptor for video sequences, which was also further generalized in [7] for a quantization without singularities based on regular polyhedrons. Jhuang *et al.* [4] model each action class with a multilayer model based on the set of spatial-temporal features extracted by the Gabor filters. Niebles *et al.* [8] have proposed a hierarchical model that can be characterized as a constellation of bags-of-features and that is able to combine both spatial and spatial-temporal features.

Mid-level motion features constructed from low-level optical flow features can be also used as in [10].

Global temporal approaches rely on global features computed on the whole time span of the action. In this context, some authors have proposed to regard each human action as 3D shape induced by the set of spatial silhouettes and propose to extract a set of local and global spatial-temporal features from this space-time shape with the generalization of the Poisson equation [2], [11] or with a (3D) distance transform [12]. Tseng *et al.* [5] have suggested to construct, in a reduced dimensional space, a spatial and temporal action graph which connects the different (dynamic) shape variation of human silhouettes of a same human action. Saad *et al.* [1] have proposed to use a set of spatial-temporal kinematic features that intend to capture the representative dynamics of the optical flow of the video sequence in the form of its dominant kinematic modes. Each video is then embedded into a kinematic-mode-based feature space and

the coordinates of the video in that space are then used for classification. Bobik and Davis [6] have exploited a temporal image template for stored instances of views of known actions where the value at each point is a function of the motion properties at the corresponding spatial location in an image sequence.

The proposed method first relies on the exploitation of a reliable, compact and discriminative representation of the data cube containing the sequence of binarized silhouettes. To this end, a set of (at most three) prototypes is thus generated by using an MDS-based dimensionality reduction technique with respect first to the time axis but also through the spatial (row and column) axis of the binary video sequence of 2D silhouettes. This strategy aims at modeling each human action in a low dimensional space, as an ordered set (or trajectory) of a few points (or a specific curve), for each viewpoint of the video cube in a very complementary way and thus with a minimal loss of reliable information. A $K$-nearest neighbor classifier will be used to classify an action on each view point and a simple and intuitive fusion technique which allows us to achieve a recognition rate performance close to the best state-of-the-art methods.

## II. Method

### A. Description

Our method is based on three stages:

- Preprocessing
- Prototype extraction
- Classification and Fusion

### B. Preprocessing

The first step of this preprocessing consists in obtaining the binary video sequence of 2D silhouettes (indicating only the body position) for each human action. To this end, we have subtracted the median background from each image of the sequence and have then used a simple thresholding technique. Once the body silhouette extraction is achieved, an additional step of filtering by a classical $3 \times 3$ median filter is then used to remove some misclassified pixels inside and outside the binarized body silhouettes. The last step consists in centering the gravity center of each silhouette inside a rectangular fixed size bounding box ($N_l \times N_w$) with a translation vector (cf. Fig. 1). Finally we consider only the first $N_i = 13$ frames of each sequence (with a step size variable according the class), which corresponds approximately, for the Weizmann data-set, to the number of frames typically occurring during a periodic cycle of human action.

### C. Prototype Extraction

This stage consists in building a set of (at most three) prototypes or, more precisely, a set of reliable, compact and discriminative representations of the data cube containing the sequence of binarized silhouettes. To attain this goal, this set of prototypes are herein generated by an MDS-based non linear dimensionality reduction technique [14] from different view-points of the video cube containing the sequence of binarized silhouettes, namely

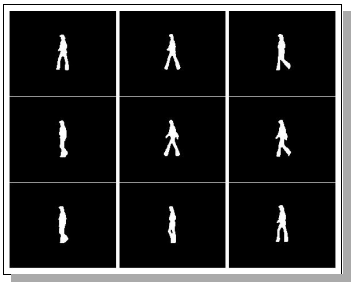


Figure 1: Example of video sequence from the Weizmann data-set after the preprocessing step to extract and center the body silhouettes.

- The first viewpoint aims at reducing the dimensionality of the image cube along the temporal axis of the video sequence. To this end, every $N_i$ silhouette image frames in the $N_l \times N_w$ dimensions is converted into a $N$-dimensional ($N = N_l \times N_w$) vector in a raster scan manner and reduced to 3 dimensions by the FastMap technique [15] (which is a fast alternative to the MDS algorithm with a linear complexity). This strategy aims at modeling each human action in a low 3D dimensional space, as an ordered set of $N_i$ points or a specific curve (possibly periodic if $N_i$ is greater, in terms of number of images that the considered human action cycle, cf. Fig. 3).
- The second viewpoint aims at reducing the dimensionality of the image cube through the spatial (line or column) axis of the set of 2D binarized silhouettes (cf. Fig. 2). For example, if we consider the axis of lines, this amounts to considering the set of $N_l$ images which are laterally created in this data cube. These images are then converted into a $N_i \times N_w$-dimensional vector in a raster scan manner and reduced to 3 dimensions by the FastMap, thus generating, for this viewpoint, a trajectory of $N_l$ ordered points in a 3D dimensional space.

This multi-axial non-linear dimensionality reduction strategy has several advantages. It allows us to obtain a set of compact representations, retaining the most significant information in each human action (by removing redundancy in the data) in two different and complementary ways (with

a minimal loss of reliable information) while preventing, to some extend, the classification model from over-fitting in the training phase. In our application, this will allows us to generate a compact and discriminative (set of) prototype(s) which will be similar and consistent between the same action of two different persons.

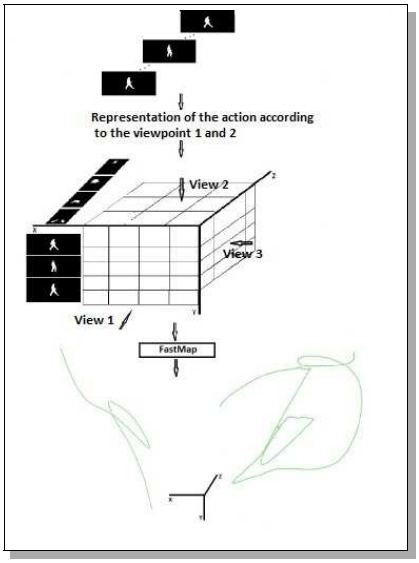



Figure 2: Set of two prototypes or 3D curves generated by a MDS-based dimensionality reduction according two viewpoints or through two axis (namely; the temporal axis and the line axis) on a sequence of binarized silhouettes.

We can evaluate the efficiency of the FastMap technique in its ability to reduce the dimensionality reduction of a sequence of binarized silhouettes in two different and complementary ways when this is achieved according to different axis. To this end, we can easily compute the correlation metric [13] which is simply the correlation of the Euclidean distance between each pairwise vectors in the high dimensional space (let $X$ be this vector) and their corresponding (pairwise) Euclidean distances in the low (3D) dimensional space (let $Y$ be this vector). The correlation $\rho$ can be estimated by the following equation:

$$\rho_{X,Y} = \text{corr}(X,Y) = \frac{\text{cov}(X,Y)}{\sigma_X \, \sigma_Y} = \frac{X^t Y/|X| - \bar{X}\bar{Y}}{\sigma_X \, \sigma_Y} \quad (1)$$

where $X^t$, $|X|$, $\bar{X}$ and $\sigma_X$ respectively represent the transpose, cardinality, mean, and standard deviation of $X$. This

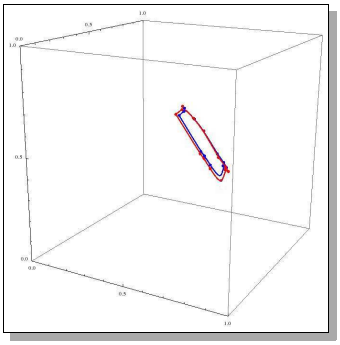


Figure 3: Periodic prototype or 3D curve Modeling two human action cycles of the SIDE class (normalized between $[0, 1]$ for the visualization).

correlation factor (Pearson) will specifically quantify the degree of linear dependence between the variables $X$ and $Y$ and quantify how the FastMap technique is able to give a low dimensional mapping in which each object is placed such that the between-object distances (in the original high dimensional space) are preserved as well as possible [14]. A perfect correlation $\rho = 1$ indicates a perfect relationship between original data and reduced data and a correlation of $\rho = 0$ indicates a total loss of information. The following table shows the mean correlation coefficient obtained on the Weizmann data-set for each viewpoint:

| Class | Viewpoint 1 (%) | Viewpoint 2 (%) |
|---|---|---|
| walk | 83 | 69 |
| run | 79 | 68 |
| skip | 83 | 66 |
| side | 79 | 68 |
| jack | 80 | 68 |
| wave1 | 88 | 64 |
| wave2 | 91 | 63 |
| jump | 81 | 67 |
| pjump | 89 | 70 |
| bend | 80 | 68 |

Table I: Mean correlation rate in percentage for the MDS-based dimensionality reduction with the FastMap technique according to two viewpoints; namely the temporal axis (viewpoint 1) and the line axis (viewpoint 2) for each human action class.

Table I shows us first that the MDS procedure is able to preserve a large quantity of structural information of the original data set and that the main efficient way to reduce the dimensionality of the information contained in the video

cube consists in doing this for each image of the video cube commonly generated along the temporal axis direction (viewpoint 1) compared to the line-axis direction (viewpoint 2). Nevertheless, we will see, in the following section, that the second viewpoint generates a second prototype which is very complementary to the first one, in terms of classification accuracy.

### *D. Classification and Fusion*

Let us recall that, in our application, the first and second prototype is respectively a set of $N_i$ and $N_l$ ordered points in a (reduced) 3D dimensional space.

For the first prototype, each $i$-th point ($1 \leq i \leq N_i$) of a test prototype (to be classified) is used to feed a non-parametric $K$-Nearest Neighbor (KNN) classifier using a 3D Euclidean distance between each $i$-th point (and thus trained with the set of $i$-th points of each prototype belonging to the training set). For the first prototype, the result of these $N_i$ KNN classification results allows us to generate a score vector both indicating the class and the sum (over the K-nearest neighbors) of the number of nearest points of the prototype (of the training set) with the most nearest points.

Our fusion procedure then consists simply in adding the two score vectors generated by the first and second viewpoint and to classify a test prototype by the majority class. If there is no majority class, our recognition system will produce a classification error.

## III. Experimental Results

To evaluate the efficiency of our human action recognition system we validate our approach on the famous Weizmann data-set [11]. This data-set contains $10$ action classes performed by $9$ different human subjects. The actions include bending (bend), jumping jack (jack), jumping-forward-on-two-legs (jump), jumping-in-place-on-two-legs (pjump), running (run), galloping-sideways (side), skipping (skip), walking (walk), waving-one-hand (wave1) and waving-two-hands (wave2). There are totally $93$ video sequence ($180 \times 144$, $25$ fps) since some types of actions are performed twice by some individuals. In order to validate our procedure, we replicate the scenario proposed in [8], [9], [7], [5], [10], [12], [11]. More precisely, for every video sequence, we perform a leave-one-out procedure, *i.e.*, we remove the entire sequence from the database while other actions of the same individual remain. Each video cube of the removed sequence is then compared to all the remaining video cube examples in the database and is classified, in our application, with our KNN-based fusion procedure.

The confusion matrix for each viewpoint is shown in Tables II and III and illustrate the different classification results obtained with $K = 1$ for each class. The confusion matrix given by our fusion procedure combining these two viewpoints is shown in Table IV (in our application, $K = 1$ allows us to obtain the best classification accuracy). Finally, the Table V shows us the recognition rate obtained respectively for each viewpoint and for the fusion procedure combining these two viewpoints. In our application, the third viewpoint according to the column axis does not allow us to improve the recognition rate.

| | walk | run | skip | jack | jump | pjump | side | wave1 | wave2 | bend |
|---|---|---|---|---|---|---|---|---|---|---|
| walk | 0.89 | | 0.11 | | | | | | | |
| run | 0.11 | 0.67 | 0.22 | | | | | | | |
| skip | 0.11 | 0.22 | 0.45 | 0.11 | 0.11 | | | | | |
| jack | | | | 0.67 | | | | | 0.33 | |
| jmup | | | | | 0.78 | | | 0.11 | 0.11 | |
| pjump | | | | 0.11 | | 0.78 | | 0.11 | | |
| side | | | | | | | 0.89 | 0.11 | | |
| wave1 | | | | | | | | 0.89 | 0.11 | |
| wave2 | | | | | | | | 0.33 | 0.67 | |
| bend | | | | | | | | | 0.11 | 0.89 |

Table II: Confusion matrix associated to viewpoint 1.

| | walk | run | skip | jack | jump | pjump | side | wave1 | wave2 | bend |
|---|---|---|---|---|---|---|---|---|---|---|
| walk | 0.78 | 0.11 | | | | | 0.11 | | | |
| run | | 0.89 | 0.11 | | | | | | | |
| skip | | 0.22 | 0.67 | | 0.11 | | | | | |
| jack | | 0.11 | | 0.45 | 0.11 | | | 0.11 | 0.22 | |
| jump | | | 0.33 | | 0.34 | 0.22 | 0.11 | | | |
| pjump | | | 0.11 | | 0.11 | 0.78 | | | | |
| side | 0.11 | | | | | | 0.89 | | | |
| wave1 | | | | | | 0.22 | | 0.66 | 0.22 | |
| wave2 | | | 0.11 | | | | | 0.11 | 0.78 | |
| bend | | | | | | | | | 0.11 | 0.89 |

Table III: Confusion matrix associated to viewpoint 2.

| | walk | run | skip | jack | jump | pjump | side | wave1 | wave2 | bend |
|---|---|---|---|---|---|---|---|---|---|---|
| walk | 1.0 | | | | | | | | | |
| run | | 1.0 | | | | | | | | |
| skip | | 0.11 | 0.89 | | | | | | | |
| jack | | | | 0.78 | | | | | 0.22 | |
| jump | | | | | 0.78 | | | 0.11 | 0.11 | |
| pjump | | | 0.11 | | | 0.89 | | | | |
| side | | | | | | | 1.0 | | | |
| wave1 | | | | | | | | 1.0 | | |
| wave2 | | | | | | | | 0.11 | 0.89 | |
| bend | | | | | | | | | | 1.0 |

Table IV: Confusion matrix associated to the fusion of the two viewpoints.

| | View point 1 | View point 2 | Fusion |
|---|---|---|---|
| Number of K-NN | K=1 | K=1 | K=1 |
| Recognition rate | 75.8 | 71.3 | **92.3** |

Table V: Table show the recognition rate of each view point and the fusion.

## IV. Discussion

It can be observed (see V) that we can recognize all types of actions existing in the data-set with very good performance results on most of the actions except some of them (e.g., JACK and JUMP). We can also notice that some misclassified action classes, given by our system, are generally and very logically in the action classes which are physically the closest to the test class; this is the case for instance of the class WAVE1 or WAVE2 which are mechanically and visually similar to the action class JUMP or class SKIP

and RUN. These action classes are very similar between them in the way the subjects move and bounce across the video sequence. We can also note that the classification results obtained by each individual viewpoint seem complementary. We have compared the accuracy of our approach with other state-of-the-art (recently published) methods using the *leave-one-out* procedure and the Weizmann data-set [8], [9], [7], [5], [10], [12], [11] in the Table VI.

| Method | Accuracy |
|---|---|
| **Our method** | **92.3%** |
| Fathi *et al.* | 99.9% |
| Gorelick *et al.* | 97.8% |
| Grundmann *et al.* | 94.6% |
| Jia *et al.* | 90.9% |
| Klaser *et al.* | 84.3% |
| Scovanner *et al.* | 82.6% |
| Niebles *et al.* | 72.8% |

Table VI: Comparison with other state-of-the-art methods [8], [9], [7], [5], [10], [12], [11].

## V. Conclusion

In this paper, we have presented an original and simple human action recognition system based on a set of (two) compact and discriminative prototype models which is similar and consistent between the same action of two different persons. In our application, these two prototype models are generated from an MDS-based multi-axial non-linear dimensionality reduction strategy which has several advantages. It allows us to retain the most significant information in each human action in two different and complementary ways and also give a better representation of the action in low dimension, while preventing, to some extend, the prototype model-based classification scheme from over-fitting in the training phase. This set of two prototypes contains rich and descriptive information about the action performed and this is clearly demonstrated by the success of the relatively simple classification scheme used in our application (KNN classification and Euclidean distance). Experimental results demonstrate that our method can accurately recognize human actions and outperforms some, more complex, state-of-the-art recognition methods on a publicly available action data-set. Finally, it is also worth mentioning that our recognition performance can also be easily improved by using a more sophisticated and powerful classification strategy such as the SVM classifier or a deep neural network.

## References


[1] S. Ali and M. Shah, *Human action recognition in videos using kinematic features and multiple instance learning*, IEEE Trans. Pattern Anal. Mach. Intell., 32(2):288-303, 2010

[2] M. Blank and L. Gorelick and E. Shechtman and M. Irani and R. Basri, *Actions as space-time shapes*, Proceedings of the Tenth IEEE International Conference on Computer Vision, ICCV'05, pp. 1395–1402, 2005.

[3] C. Schuldt and I. Laptev and B. Caputo, *Recognizing human actions: A local SVM approach*, Proceedings of the 17th International Conference on Pattern Recognition, IEEE Computer Society, ICPR'04 Vol.3, pp. 32–36, 2004

[4] H. Jhuang and T. Serre and L. Wolf and T. Poggio, *A biologically inspired system for action recognition*, IEEE Proceedings of the 11th International Conference on Computer Vision, ICCV'07 pp. 1–8

[5] K. Jia and D.Y. Yeung, *Human action recognition using local spatio-temporal discriminant embedding*, Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, CVPR'08

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

[7] A. Klaser and M. Marszalek and C. Schmid, *A spatio-temporal descriptor based on 3D-gradients*, Proceedings of the 19th British Machine Vision Conference, BMVC'08, pp. 1–10, 2008.

[8] J.C. Niebles and L. Fei-Fei, *A hierarchical model of shape and appearance for human action classification*, Proceedings of IEEE International Conference in Computer Vision and Pattern Recognition, CVPR'07

[9] P. Scovanner and S. Ali and M. Shah, *A 3-dimensional SIFT descriptor and its application to action recognition*, Proceedings of the 15th international conference on Multimedia, MM'07 pp. 357–360, 2007.

[10] A. Fathi and G. Mori, *Action recognition by learning mid-level motion features*, Proceedings of IEEE International Conference in Computer Vision and Pattern Recognition CVPR'08

[11] L. Gorelick and M. Blank and E. Shechtman and M. Irani and R. Basri, *Actions as space-time shapes*, IEEE Trans. on Pattern Anal. Mach. Intell., 2007 29(12):2247–2253, 2007

[12] M. Grundmann and F. Meier and I. Essa, *3D shape context and distance transform for action recognition*, Proceedings of IEEE International Conference in Pattern Recognition, ICPR'08 pp. 1–4, 2008

[13] P. Jacobson and M.R. Gupta, *Design goals and solutions for display of hyperspectral images*, IEEE Trans. Geosci. Remote Sens., 2005

[14] F.T. Cox and M.A.A. Cox, *Multidimensional Scaling*, Chapman and Hall/CRC, 2000

[15] G. Faloutsos and K. Lin, *FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets*, Proceedings of 1995 ACM SIGMOD, pp. 163-174, 1995