

Université de Montréal

**Reconnaissance des actions humaines: méthode
basée sur la réduction de dimensionnalité par
MDS spatio-temporelle**

par

Lilia Chorfi Belhadj

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

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

Août, 2015

© Lilia Chorfi Belhadj, 2015.

Université de Montréal
Faculté des arts et des sciences

Ce mémoire intitulé:

**Reconnaissance des actions humaines: méthode
basée sur la réduction de dimensionnalité par
MDS spatio-temporelle**

présenté par:

Lilia Chorfi Belhadj

a été évalué par un jury composé des personnes suivantes:

Aaron Courville
président-rapporteur

Max Mignotte
directeur de recherche

Jean Meunier
membre du jury

Résumé :

L'action humaine dans une séquence vidéo peut être considérée comme un volume spatio-temporel induit par la concaténation de silhouettes dans le temps. Nous présentons une approche spatio-temporelle pour la reconnaissance d'actions humaines qui exploite des caractéristiques globales générées par la technique de réduction de dimensionnalité MDS et un découpage en sous-blocs afin de modéliser la dynamique des actions. L'objectif est de fournir une méthode à la fois simple, peu dispendieuse et robuste permettant la reconnaissance d'actions simples. Le procédé est rapide, ne nécessite aucun alignement de vidéo, et est applicable à de nombreux scénarios. En outre, nous démontrons la robustesse de notre méthode face aux occultations partielles, aux déformations de formes, aux changements d'échelle et d'angles de vue, aux irrégularités dans l'exécution d'une action, et à une faible résolution.

Mots-clés : représentation de l'action, reconnaissance de l'action, analyse spatio-temporelle, positionnement multidimensionnel (MDS).

Abstract :

Human action in a video sequence can be seen as a space-time volume induced by the concatenation of silhouettes in time. We present a space-time approach for human action recognition, which exploits global characteristics generated by the technique of dimensionality reduction MDS and a cube division into sub-blocks to model the dynamics of the actions. The objective is to provide a method that is simple, inexpensive and robust allowing simple action recognition. The process is fast, does not require video alignment, and is applicable in many scenarios. Moreover, we demonstrate the robustness of our method to partial occlusion, deformation of shapes, significant changes in scale and viewpoint, irregularities in the performance of an action, and low-quality video.

Keywords : Action representation, action recognition, space-time analysis, multidimensional scaling (MDS).

TABLE DES MATIÈRES

Resumé	ii
Abstract	ii
Table des figures	v
Liste des tableaux	vii
Liste des algorithmes	ix
1 INTRODUCTION ET ÉTAT DE L'ART	1
1.1 Motivation	2
1.2 Domaines d'application	4
1.2.1 Biométrie comportementale	4
1.2.2 Analyses vidéo basées sur le contenu	4
1.2.3 Surveillance et sécurité	4
1.2.4 Applications et environnements interactifs	5
1.2.5 Animation et synthèse d'images	5
1.3 Aperçu général	5
1.3.1 Flux optique	6
1.3.2 Trajectoire	6
1.3.3 Silhouettes, squelettes et contours	6
1.3.4 Réponses de filtres	7
1.4 Modéliser les "ACTIONS"	7
1.4.1 Les méthodes séquentielles	8
1.4.2 Les méthodes spatio-temporelles	9
1.5 Reconnaître les "ACTIONS"	17
1.5.1 Réduction de dimensionnalité	17
1.5.2 k -ppv	18
1.5.3 Classifieurs discriminants	18
1.6 Bases de données des actions humaines	19
2 ÉTUDE BIBLIOGRAPHIQUE	23
2.1 Introduction	24
2.2 Algorithmes d'apprentissage	24
2.2.1 Les types d'apprentissage	25
2.2.2 Apprentissage supervisé : méthodes de classification	26
2.3 Réduction de la dimensionnalité	34
2.3.1 La sélection de caractéristiques	35
2.3.2 L'extraction de caractéristiques	37
2.4 Détection de mouvements	42
2.4.1 Étapes d'une opération de soustraction de fond	43

2.4.2	Techniques de modélisation du fond de référence	46
3	IMPLEMENTATION ET RÉALISATION	55
3.1	Introduction	56
3.2	<i>Multi-dimensional scaling</i> MDS	56
3.2.1	<i>FastMap</i>	57
3.3	Vue globale du système construit	61
3.3.1	Soustraction de fond et extraction des silhouettes	61
3.3.2	Opérations de prétraitement des images	63
3.3.3	Modélisation des actions par MDS	66
3.3.4	Classification et reconnaissance des actions	69
3.4	Conclusion	70
4	EXPÉRIMENTATIONS ET RÉSULTATS	71
4.1	Introduction	72
4.2	Résultats sur la base WEIZMANN	74
4.3	Résultats sur la base KTH	79
4.4	Conclusion	84
	CONCLUSION	85
	Bibliographie	87

TABLE DES FIGURES

1.1	Exemple d'un volume 3-D (XYT) construit par concaténation.	10
1.2	Représentations spatio-temporelles des actions selon la méthode de Bobick et al. [23].	11
1.3	Représentation de l'action "marcher" par des trajectoires spatio-temporelles des articulations selon la méthode de Sheikh et al. [186].	12
1.4	Représentation des actions selon la méthode de Blank et al. [21] et Gorelick et al. [72].	14
1.5	Exemples de caractéristiques spatio-temporelles 3-D locales.	15
1.6	Représentations des contours d'objets et le volume spatio-temporel 3-D (XYT) correspondant pour l'action "chuter" selon la méthode de Yilmazs et al. [227].	16
3.1	Illustration de la loi des cosinus - projection sur la droite ($O_a O_b$).	59
3.2	Projection sur un hyper-plan H , perpendiculaire à la droite ($O_a O_b$).	59
3.3	Étape de soustraction de l'arrière-plan et extraction de la silhouette.	64
3.4	Étape de modélisation des actions par MDS.	67
3.5	Prototypes des actions étudiées.	68
4.1	Échantillon d'images extraites des séquences vidéos de la base WEIZMANN	72
4.2	Échantillon d'images extraites des séquences vidéos de la base KTH	73
4.3	Matrices de confusion des actions lors de la classification.	75
4.4	Matrices de confusion des actions lors de la classification pour chaque scénario à l'aide de la MDS + 1-ppv.	80
4.5	Matrices de confusion des actions lors de la classification de $\{\mathbf{s1}, \mathbf{s2}, \mathbf{s3}, \mathbf{s4}\}$ à l'aide de la MDS + 1-ppv.	81
4.6	Matrices de confusion des actions lors de la classification de $\{\mathbf{s2}\}$ à l'aide de la MDS + 1-ppv.	81
4.7	Comparaison des résultats de notre méthode avec celle de Schüldt et al. [180].	82

LISTE DES TABLEAUX

4.1	Reconnaissance de l'action " walk " selon différents scénarios	76
4.2	Reconnaissance de l'action " walk " selon différents angles de vue	78
4.3	Tableau comparatif des taux de reconnaissance sur les bases WEIZ- MANN et KTH	83

LISTE DES ALGORITHMES

1	Heuristique pour choisir deux objets éloignés	58
2	<i>FastMap</i>	60
3	Soustraction de l'arrière-plan et prétraitement des images.	65
4	Système de reconnaissance des actions	69

1

INTRODUCTION ET ÉTAT DE L'ART

Sommaire

1.1	Motivation	2
1.2	Domaines d'application	4
1.2.1	Biométrie comportementale	4
1.2.2	Analyses vidéo basées sur le contenu	4
1.2.3	Surveillance et sécurité	4
1.2.4	Applications et environnements interactifs	5
1.2.5	Animation et synthèse d'images	5
1.3	Aperçu général	5
1.3.1	Flux optique	6
1.3.2	Trajectoire	6
1.3.3	Silhouettes, squelettes et contours	6
1.3.4	Réponses de filtres	7
1.4	Modéliser les "ACTIONS"	7
1.4.1	Les méthodes séquentielles	8
1.4.2	Les méthodes spatio-temporelles	9
1.5	Reconnaître les "ACTIONS"	17
1.5.1	Réduction de dimensionnalité	17
1.5.2	k -ppv	18
1.5.3	Classifieurs discriminants	18
1.6	Bases de données des actions humaines	19

1.1 Motivation

La dernière décennie a été témoin d'une rapide prolifération de caméras vidéo de différents types, allant de la plus simple à la plus sophistiquée. Cela a donné lieu à une explosion de contenus vidéo. Plusieurs applications telles que la recherche et l'archivage de vidéos basé sur le contenu, l'extraction d'informations ou encore, le résumé de vidéos exigent la reconnaissance des activités se produisant dans celles-ci. Parmi ces activités, celle sur laquelle a porté notre intérêt est l'activité humaine.

L'analyse des activités humaines dans les vidéos est un domaine dont les aboutissements sont de plus en plus considérables dans des secteurs aussi divers que la surveillance, la sécurité, la santé, le divertissement, etc.

Plusieurs défis à différents niveaux de traitement - robustesse face aux erreurs dans les traitements de bas niveau, représentations invariantes dans les traitements de niveaux intermédiaires et interprétation sémantique des activités humaines dans les traitements de haut niveau - rendent ce problème difficile à résoudre.

Depuis les premiers travaux de **Johansson** [100] en 1973 dans le domaine de la neuroscience, l'analyse de la perception du mouvement humain a fait l'objet de très nombreuses recherches. La technique consistait à placer sur le corps d'un sujet des cibles lumineuses au niveau de chaque articulation (épaule, hanche, genou, pied) et à filmer ensuite, dans l'obscurité complète, la production de différents types de mouvements (danse, locomotion, manipulation d'objets). La tâche consistait à faire identifier la nature des mouvements représentés par l'ensemble des points lumineux. Lorsque les points sont présentés statiquement, l'identification est impossible. En revanche, il suffit de quelques images en mouvement, pour que les sujets reconnaissent très rapidement qu'il s'agit d'un mouvement humain. Ces résultats ont ouvert la voie à la modélisation mathématique de l'action humaine et la reconnaissance automatique, faisant de la reconnaissance des activités humaines dans les vidéos un des champs les plus prometteurs de la vision par ordinateur et la reconnaissance de formes, attirant ainsi, l'attention des chercheurs dans divers domaines : industrie, milieu universitaire, organismes de sécurité, organismes de consommateurs.

Le problème est dès lors, en termes simples, étant donné une séquence d'images représentant une action, est-il possible de concevoir un système capable de reconnaître cette action de manière automatique? Aussi simple que la question semblait être, la solution a été d'autant plus difficile à trouver.

Il existe dans la littérature, diverses études traitant et résumant les approches mises en œuvre au cours de ces vingt dernières années afin de répondre à ce problème.

1.1. Motivation

Aggarwal et al. [6] discutent les trois sous-problèmes importants qui, ensemble, forment un système de reconnaissance d'action complet : extraction de la structure du corps humain à partir d'images, suivi dans toutes les images et la reconnaissance de l'action. **Cedras et al.** [35] présentent une étude sur les approches basées sur les mouvements, par opposition, aux approches fondées sur les structures. Ils soutiennent que le mouvement est un indice plus important pour la reconnaissance de l'action que la structure du corps humain en elle-même. **Gavrila** [69] a présenté une approche basée, principalement, sur le suivi des mains et des humains via des modèles 2-D ou 3-D ainsi qu'une discussion sur les techniques de reconnaissance d'actions. Enfin, plus récemment, **Moenslund et al.** [145] ont présenté un résumé sur les problèmes et les approches concernant la capture de mouvement humain, y compris l'initialisation du modèle humain, le suivi, l'estimation de la pose et la reconnaissance de l'activité.

Il existe différents types d'activités humaines. Selon leur complexité, les activités humaines sont usuellement classées en quatre niveaux : les gestes, les actions, les interactions et enfin les activités. Les gestes font référence à des mouvements élémentaires d'une partie du corps d'un humain, et sont ainsi les composants atomiques décrivant un mouvement significatif ; "étirer un bras" et "élever une jambe" sont de bons exemples de gestes. Les actions sont les exécutions de mouvements d'une seule personne. elles peuvent être composées de plusieurs gestes organisés chronologiquement, comme "marcher" et "courir". Les interactions sont des activités humaines simples impliquant deux personnes et/ou des objets. Par exemple : "combat entre deux sujets" est une interaction entre deux humains et "deux sujets jouant au volley-ball" est une interaction homme-objet impliquant deux humains et un objet. Enfin, les activités sont exercées par des groupes conceptuels composés de personnes et/ou d'objets multiples telles "une partie de hockey".

Dans ce qui suit, nous nous concentrerons exclusivement sur les approches pour la reconnaissance de l'action, les interactions et activités étant au-delà de ce qui est traité dans ce mémoire. De plus nous nous limiterons aux approches pour la reconnaissance et non pas sur les modules de niveau inférieur de détection et de suivi, qui seront, eux, discutés dans le chapitre suivant.

La suite du chapitre est organisée comme ceci. Tout d'abord, nous présenterons quelques domaines d'application de la reconnaissance de mouvements humains. Puis nous donnerons un aperçu des composants généralement utilisés lors de l'extraction des caractéristiques des actions. Par la suite, nous discuterons de certaines méthodes de modélisation et de reconnaissance de l'action constituant une partie de l'état de l'art de la reconnaissance de l'activité humaine. Enfin nous conclurons par un aperçu succinct de l'approche proposée dans ces travaux.

1.2 Domaines d'application

Nous présentons, dans cette section, quelques domaines d'application afin de mettre en valeur l'impact potentiel des systèmes de reconnaissance d'actions basés sur la vision par ordinateur.

1.2.1 Biométrie comportementale

La biométrie implique l'étude des approches et algorithmes pour la reconnaissance humaine basée sur des indices physiques ou comportementaux. Les approches traditionnelles sont basées sur l'empreinte digitale, le visage, ou l'iris et peuvent être classées comme de la biométrie physiologique. Ces méthodes exigent la coopération du sujet pour la collecte des données biométriques. Récemment, "la biométrie comportementale" a gagné en popularité, où la prémisse est, que le comportement est un indice tout aussi utile pour reconnaître des humains que leurs attributs physiques. L'avantage de cette approche est que la coopération du sujet n'est plus nécessaire et qu'elle peut procéder sans interruption ni interférence avec le sujet ou son action. Actuellement, l'exemple le plus prometteur de biométrie comportementale est la démarche humaine [175].

1.2.2 Analyses vidéo basées sur le contenu

Avec la multiplication des sites de partage de vidéos, il est devenu nécessaire de développer des outils d'indexation et de stockage fiables et efficaces afin d'améliorer l'expérience de l'utilisateur. Cela nécessite l'apprentissage de modèles à partir de vidéos brutes et résumer celles-ci selon leur contenu. Cette pratique a gagné un regain d'intérêt avec les progrès des applications de recherche d'image par le contenu [170].

1.2.3 Surveillance et sécurité

Les systèmes de sécurité et de surveillance ont traditionnellement compté sur un réseau de caméras vidéo surveillé par un opérateur humain. Avec la croissance récente du déploiement des caméras, l'efficacité et la précision des opérateurs humains

1.3. Aperçu général

faiblissent. Ainsi, les organismes de sécurité cherchent des solutions basées sur la vision permettant de remplacer ou aider l'opérateur humain. La reconnaissance automatique des anomalies dans le champ de vision d'une caméra est un tel problème qu'il a attiré l'attention de plusieurs chercheurs de vision [203, 236].

1.2.4 Applications et environnements interactifs

Comprendre l'interaction entre un ordinateur et un humain reste l'un des grands défis dans la conception d'interfaces homme-machine. Les repères visuels sont le mode le plus important de la communication non verbale. L'utilisation adéquate de ce mode peut amener à la création d'ordinateurs interagissant de manière efficace avec leur utilisateur. De même, les environnements interactifs tels que les maisons intelligentes [159] réagissant aux gestes de l'utilisateur peuvent bénéficier de méthodes basées sur la vision par ordinateur.

1.2.5 Animation et synthèse d'images

L'industrie de l'animation et du jeu vidéo repose sur la synthèse réaliste de l'humain et de ses mouvements. La synthèse de mouvement trouve une large application dans l'industrie du jeu où l'exigence est de produire une grande variété de mouvements avec quelques compromis sur la qualité. L'industrie du film d'autre part repose traditionnellement davantage sur des animateurs humains pour fournir des animations de haute qualité en terme de réalisme. Toutefois, ces tendances tendent à changer [65]. Grâce à l'amélioration des algorithmes et du matériel, la synthèse de mouvements beaucoup plus réalistes est maintenant possible à partir de l'apprentissage. Une application possible est l'apprentissage dans des environnements de simulation.

1.3 Aperçu général

Un système de reconnaissance de l'action peut être considéré comme un processus partant d'une séquence d'images et arrivant à une interprétation de plus haut niveau en une série d'étapes. Les principales étapes sont les suivantes :

1. Saisie de vidéos ou de séquences d'images en entrée ;
2. Extraction des composants pertinents en bas niveau ;

3. Descriptions d'actions à partir des composants en niveau intermédiaire ;
4. Interprétations sémantiques des actions primitives en haut niveau

Les vidéos, de façon générale, se composent de quantités massives de données brutes sous la forme d'un cube spatio-temporel de variations d'intensité. Néanmoins, la majorité de ces informations n'est pas directement pertinente pour les tâches de compréhension et d'identification de l'activité qui se déroule dans la vidéo. Des facteurs externes tels que la couleur des vêtements, les conditions d'éclairage et les changements de fond ne facilitent pas la tâche de reconnaissance. Nous décrivons, brièvement, quelques composants populaires utilisés dans les systèmes de reconnaissance.

1.3.1 Flux optique

Le flux optique est défini comme le mouvement apparent de pixels individuels sur le plan de l'image. Il constitue une bonne approximation du véritable mouvement physique projeté sur le plan de l'image. La plupart des méthodes pour calculer le flux optique supposent que la couleur/intensité d'un pixel est invariante au déplacement d'une image à l'autre. Le flux optique fournit une description concise des régions en mouvement dans l'image ainsi que la vitesse de celui-ci. Cependant, le calcul de flux est sensible au bruit et aux changements d'éclairage. Parmi les applications du flux optique, l'une des plus courantes est celle de la surveillance automatisée de trafics routiers [88].

1.3.2 Trajectoire

Le suivi de trajectoires des objets en mouvement a, souvent, été utilisé comme caractéristique de déduction de l'activité de celui-ci. La trajectoire en elle-même n'est pas significative car elle est sensible aux translations, aux rotations et aux changements d'échelle. D'un autre côté, certaines caractéristiques dérivées de celle-ci, telles que la vitesse, la vélocité, la courbure spatio-temporelle ou encore le mouvement relatif, peuvent être révélateurs sur la nature du mouvement en plus d'être invariant aux variabilités citées plus haut.

1.3.3 Silhouettes, squelettes et contours

Plusieurs méthodes basées sur une description globale de la silhouette, du contour ou encore du squelette ont été proposées pour la quantification du mouvement. La

1.4. Modéliser les “ACTIONS”

forme de la silhouette humaine joue un rôle très important dans la reconnaissance des actions humaines. Des approches globales telles que celles basées sur les moments [87] considèrent la région de la silhouette entière afin de calculer des descripteurs de forme. Les approches dites de frontière, quant à elles, ne considèrent que le contour de la forme comme caractéristique. Ces méthodes comprennent, mais non exclusivement, les approches à base de chaînes de code [66], les descripteurs de Fourier (par tangente, par représentation complexe), etc. Enfin, les méthodes basées sur la squelettisation, redéfinissent une forme complexe en un ensemble de courbes 1-D centrées, appelé squelette ou axe médian [22].

1.3.4 Réponses de filtres

Il existe plusieurs caractéristiques extraites à partir de réponses de filtres spatio-temporels. Dans leurs travaux, **Zhong et al.** [236] ont traité les séquences vidéo à l'aide d'une Gaussienne spatiale et d'une dérivée Gaussienne sur l'axe temporel. En raison de l'opération de dérivation sur l'axe temporel, le filtre enregistre les hautes fréquences au niveau des régions de mouvement. Ces fréquences sont, par la suite, seuillées afin de générer un masque de mouvement binaire suivi par une agrégation en histogrammes spatiaux. Une telle caractéristique encode le mouvement et son information spatiale de façon compacte d'où son utilité dans la surveillance de zones larges. La théorie de l'espace-d'échelle (Scale-space, en anglais) a, également, été appliquée aux vidéos par plusieurs chercheurs. **Laptev et al.** [120] ont proposé une généralisation du détecteur d'angle de Harris aux séquences vidéo en utilisant un ensemble de filtres à base de dérivées gaussiennes spatio-temporelles. De même, **Dollár et al.** [50] ont extrait des points saillants basés sur des mouvements périodiques distinctifs dans une vidéo donnée en utilisant un noyau Gaussien dans l'espace et des fonctions de Gabor dans le temps. Du fait que ces approches sont basées sur des opérations de convolution simples, elles sont rapides et faciles à mettre en œuvre. De plus, elles sont très utiles dans les cas de vidéos à faible résolution ou de mauvaise qualité dans lesquelles il est difficile d'extraire d'autres caractéristiques telles que le flux optique ou les silhouettes.

1.4 Modéliser les “ACTIONS”

Les approches de reconnaissance des actions humaines considèrent celles-ci comme des instances de classes particulières formées de séquences d'images. Diverses méthodes de modélisation et d'algorithmes d'appariement ont été développées pour permettre

au système de reconnaissance de prendre une décision précise quant à savoir si une séquence d'images appartient à une certaine action ou non. Pour la reconnaissance de vidéos en continu, la plupart des approches ont adopté une technique de fenêtres glissantes afin de classer toutes les sous-séquences possibles.

Les méthodes de modélisation peuvent être divisées en deux catégories : les méthodes spatio-temporelles et les méthodes séquentielles. Les approches spatio-temporelles modélisent une action humaine sous la forme d'un volume 3-D dans une dimension spatio-temporelle ou sous la forme d'un ensemble de caractéristiques extraites à partir du volume. Les volumes résultent d'une concaténation d'image le long de l'axe des temps, puis sont comparés afin de mesurer leurs similarités. D'autre part, les approches séquentielles considèrent une action comme une séquence d'observations particulières. Plus précisément, elles représentent une action humaine comme une séquence de vecteurs de caractéristiques extraites à partir des images et procèdent à la reconnaissance en cherchant la séquence se rapprochant le plus.

1.4.1 Les méthodes séquentielles

Les approches séquentielles procèdent à la reconnaissance des actions humaines en analysant des séquences de caractéristiques. Ces approches considèrent une vidéo d'entrée comme une séquence d'observations (par exemple, vecteurs de caractéristiques), et en déduisent l'action si une séquence particulière caractérisant celle-ci est observée. Les approches séquentielles convertissent, d'abord, une séquence d'images en une séquence de vecteurs de caractéristiques décrivant l'état d'une personne par image. Une fois les vecteurs de caractéristiques extraits, ces approches analysent la séquence en comparant les probabilités entre la séquence et la classe d'action. Si celle-ci est suffisamment grande, le système décide que l'action a eu lieu.

Efros et al. [53] ont présenté une méthode pour reconnaître les actions à distance, où chaque être humain a une taille d'environ 30 pixels de hauteur. Afin de reconnaître les actions dans cette situation où la résolution est si faible, ils ont utilisé des descripteurs de mouvement basés sur l'estimation du flux optique obtenu pour chaque image. Leur système calcule d'abord le volume spatio-temporel de chaque sujet suivi, puis calcule les flux optiques 2-D (XY) pour chaque image en suivant les sujets via une différence temporelle d'images. Ils ont utilisé des canaux de flou cinétique comme descripteurs de mouvements, en convertissant les flux optiques en descripteurs spatio-temporels par image. Une méthode simple de classification par le plus proche voisin est, ensuite, appliquée à une séquence de descripteurs pour procéder à la reconnaissance des actions.

Lublinerman et al. [133] ont proposé une méthodologie qui reconnaît les actions humaines en les modélisant comme des systèmes linéaires invariants dans le temps (*linear-time-invariant*, LTI, en anglais). Leur système convertit une séquence d'images

1.4. Modéliser les “ACTIONS”

en une séquence de silhouettes, en extrayant deux types de représentations de contour : largeur de la silhouette et les descripteurs de Fourier. Une action est représentée comme un système LTI capturant la dynamique des variations des caractéristiques de la silhouette. Les SVM ont été appliqués pour classer une nouvelle entrée convertie en paramètres d’un modèle de LTI.

Veeraraghavan et al. [204] ont décrit une action comme une fonction temporelle décrivant des changements de paramètres. La principale contribution du système de **Veeraraghavan et al.** [204] est dans la modélisation explicite des variations des vitesses inter- et intra-personnelles de l’action lors de l’exécution. Mettant l’accent sur le fait que les humains peuvent être en mesure de changer la vitesse d’exécution de certaines parties de l’action et d’autres non, ils apprennent des caractéristiques non linéaires des variations de vitesses de celle-ci. En clair, leur système apprend la nature des transformations des alignements temporels par action. Ils ont modélisé l’exécution d’une action à l’aide de deux fonctions : (i) une fonction temporelle des changements des caractéristiques et (ii) une fonction spatiale des alignements temporels possibles.

Depuis le début des années 1990, les modèles de Markov cachés HMM ont commencé à trouver une large applicabilité dans les systèmes de vision par ordinateur. **Yamato et al.** [223] sont les premiers à appliquer les HMM standard pour reconnaître les actions. À chaque image, leur système représente une image de la silhouette binaire dans un tableau de mailles. Le nombre de pixels dans chaque maille est considéré comme une caractéristique, extrayant ainsi un vecteur de caractéristiques par image. Ces vecteurs de caractéristiques sont considérés comme une séquence d’observations générée par le modèle d’action. Chaque activité est représentée en construisant un HMM correspondant de façon probabiliste à des séquences particulières de vecteurs de caractéristiques (i.e. mailles). Plus précisément, les paramètres du HMM (probabilités de transition et probabilités d’observation) sont entraînés à l’aide d’un ensemble de données étiquetées, puis utilisés pour la reconnaissance d’une action recherchant la classe d’action associée à la vraisemblance maximale.

1.4.2 Les méthodes spatio-temporelles

Une vidéo est composée d’une séquence d’images 2-D placées dans un ordre chronologique. Par conséquent, une vidéo d’entrée comprenant l’exécution d’une action peut être représentée comme un volume spatio-temporel 3-D (XYT) construit en concaténant des images 2-D (XY) en fonction du temps (T) (**Figure 1.1**).

Les méthodes spatio-temporelles suivent ce principe. En effet, basé sur des vidéos d’entraînement, le système construit un modèle de volume spatio-temporel 3-D (XYT) représentant chaque action. Quand une vidéo non étiquetée est fournie, le système construit le volume spatio-temporel 3-D associé. Ce dernier est, par la suite, comparé à chaque modèle d’action pour mesurer la similarité des formes ou des apparences

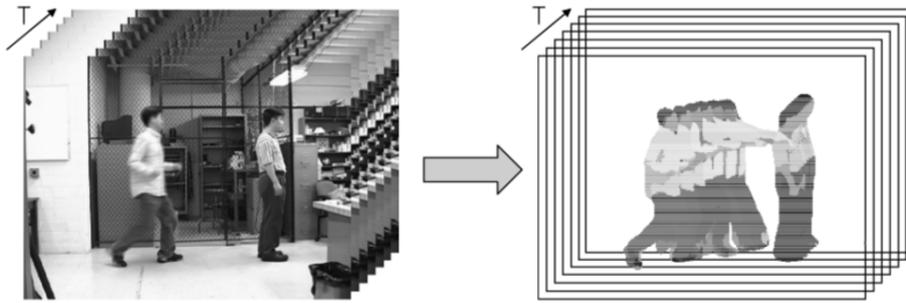


Figure 1.1 – Exemple d'un volume 3-D (XYT) construit par concaténation.

entre les deux volumes, et ainsi déduire l'action correspondante.

En plus de la représentation en volume 3-D brute, il existe plusieurs variations de la représentation spatio-temporelle. Premièrement, certains systèmes représentent une action sous forme de trajectoire dans une dimension spatio-temporelle ou d'autres dimensions. Si le système est capable de suivre des points caractéristiques tels que l'estimation des positions des articulations du sujet, le mouvement peut être représenté de façon plus explicite comme un ensemble de trajectoires de ces points. Dans un autre ordre d'idées, d'autres systèmes représentent une action comme un ensemble de caractéristiques extraites à partir du volume ou de la trajectoire.

Différents types d'algorithmes ont été appliqués afin de procéder à la reconnaissance à partir des modèles spatio-temporels. L'algorithme typique lors de l'utilisation de volumes est l'algorithme de *template-matching* qui construit un modèle représentatif (i.e. un volume) par action à l'aide de données d'entraînement puis reconnaît une nouvelle action en associant celle-ci à un des modèles appris. Les algorithmes des plus proches voisins ont également été largement appliqués, particulièrement avec les représentations sous forme de trajectoires ou de caractéristiques. Enfin, des algorithmes de modélisation statistique ont été développés qui associent les classes d'actions selon différentes distributions de probabilité.

Reconnaissance des actions avec des volumes spatio-temporels. Le cœur de la reconnaissance est la mesure de similarité entre deux volumes. Le système doit être en mesure de calculer le degré de similarité des mouvements humains décrits dans les deux volumes. Afin de calculer les similarités de manière précise, différents types de représentations de volumes spatio-temporels et de méthodes de reconnaissance ont été développés.

Au lieu de la concaténation des images entières le long de l'axe des temps, certaines approches alignent uniquement les silhouettes d'une personne pour suivre explicitement les changements de forme. **Bobick et al.** [23] ont construit un système de reconnaissance de l'action en temps réel en utilisant une approche de *template-matching*. Au lieu de traiter le volume spatio-temporel de chaque action en trois dimensions, ils représentent chaque action avec un modèle composé de deux images en deux dimensions :

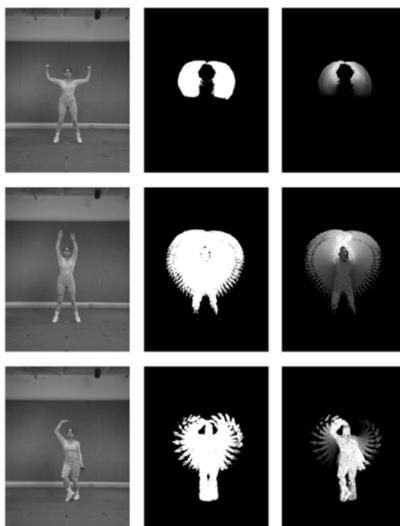


Figure 1.2 – Représentations spatio-temporelles des actions selon la méthode de **Bobick et al.** [23] : Gauche : Image originale. Milieu : Image d’énergie de mouvement binaire (MEI). Droite : Image de l’historique du mouvement à valeurs scalaires (MHI).

une image d’énergie de mouvement binaire (MEI) et une image de l’historique du mouvement à valeurs scalaires (MHI) (**Figure 1.2**). Les deux images sont construites à partir d’une séquence d’images segmentées en deux classes mobile/immobile et représentent les sommes pondérées 2-D (XY) des valeurs du volume spatio-temporel 3-D (XYT) initial. En appliquant une technique de *template-matching* traditionnelle à une paire de (MEI, MHI), leur système est capable de reconnaître des actions simples.

Une approche de comparaison des volumes en fonction des patches extraits a été proposée par **Shechtman et al.** [184]. Ces derniers ont estimé des flux de mouvement à partir d’un volume spatio-temporel afin de reconnaître les actions humaines. Ils ont calculé les corrélations hiérarchiques de modèles 3-D, en mesurant la similarité entre un volume vidéo observé et les modèles de volumes construits. À chaque localisation du volume, à savoir, (x, y, t) , ils extraient un petit patch spatio-temporel autour de la localisation. Chaque volume de patches capture le flux du mouvement local, et la corrélation entre un volume de patches modèle et un volume de patches requête au même endroit donne un score de correspondance au système. L’agrégation de ces scores identifie l’action.

Ke et al. [109] ont utilisé des volumes spatio-temporels sur-segmentés pour modéliser les activités humaines. Leur système applique un algorithme *meanshift* hiérarchique pour regrouper les voxels de même couleur, et obtenir plusieurs volumes segmentés. La motivation est de trouver les segments de volume automatiquement et de mesurer leur similarité avec le modèle d’action. La reconnaissance consiste en la recherche d’un sous-ensemble de volumes spatio-temporels sur-segmenté correspondant le mieux à la forme du modèle d’action. Les machines à vecteurs de support SVM ont été appliquées à la reconnaissance des actions humaines, tout en considérant à la fois les formes et les flux des volumes.

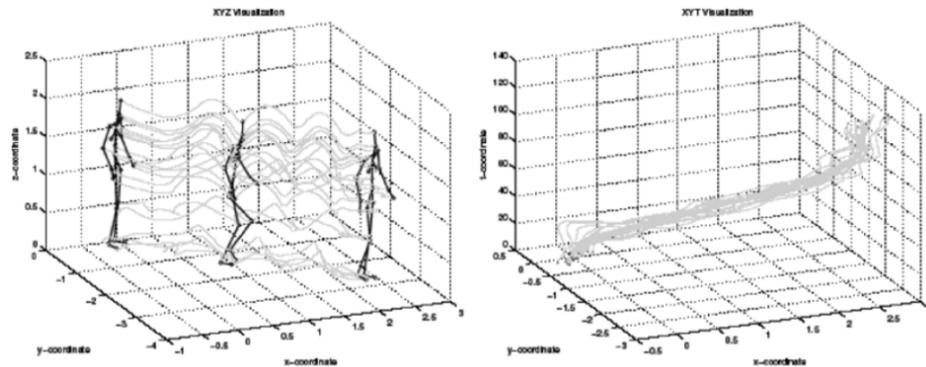


Figure 1.3 – Représentation de l'action “marcher” par des trajectoires spatio-temporelles des articulations selon la méthode de **Sheikh et al.** [186] : Gauche : Trajectoires dans l’espace (XYZ). Droite : Trajectoires dans l’espace (XYT).

Rodriguez et al. [166] ont analysé les volumes spatio-temporels en synthétisant les filtres *maximum average correlation height* (MACH), utilisés pour l’analyse des images et la reconnaissance d’objets afin de résoudre le problème de la reconnaissance de l’action. Autrement dit, ils ont généralisé le filtre MACH 2-D classique pour les volumes 3-D (XYT). Pour chaque classe d’action, un filtre de synthèse correspondant au volume observé est généré. la classification de l’action est effectuée en analysant les réponses des filtres sur les nouvelles observations.

Reconnaissance des actions avec des trajectoires spatio-temporelles. Dans les approches basées sur la trajectoire, une personne est généralement représentée comme un ensemble de points en 2-D (XY) ou en 3-D (XYZ) correspondant aux positions de ses articulations. Lorsqu’un sujet effectue une action, les changements de position de ses articulations sont représentés sous forme de trajectoires spatio-temporelles 3-D (XYT) ou 4-D (XYZT).

Plusieurs approches ont utilisé les trajectoires elles-mêmes (i.e. l’ensemble de points 3-D) pour représenter et reconnaître les actions directement. **Sheikh et al.** [186] ont représenté une action comme un ensemble de trajectoires de treize points d’articulation dans un espace 4-D (XYZT). Ils ont utilisé, par la suite, une projection affine pour obtenir les trajectoires normalisées en (XYT) d’une action dans le but de mesurer l’invariance de similitude entre deux ensembles de trajectoires (**Figure 1.3**).

Campbell et al. [33] ont reconnu les actions humaines en les représentant sous forme de courbes dans des espaces de phases de faible dimension. Basés sur les modèles de corps 3-D (XYT) estimés pour chaque image, ils ont défini l’espace de phase du corps comme un espace où chaque axe représente un paramètre indépendant du corps (par exemple, l’angle de la cheville ou l’angle du genou) ou sa dérivée première. Dans leur espace de phase, une personne statique correspond à un point, et une action correspond à un ensemble de points (par exemple, une courbe). Les auteurs ont projeté la courbe dans l’espace des phases en plusieurs sous-espaces 2-D. le système sélectionne

1.4. Modéliser les “ACTIONS”

automatiquement les k courbes les plus stables parmi toutes les courbes possibles des sous-espaces 2-D pour le processus de reconnaissance. Lors de la présentation d’une nouvelle séquence, celle-ci est convertie en un ensemble de points dans l’espace des phases sans analyse explicite de leur dynamique. Le système vérifie simplement si les points générés se retrouvent sur les courbes (trajectoires dans les sous-espaces) lors de la projection.

Au lieu de conserver des trajectoires brutes pour représenter les actions humaines, **Rao et al.** [164] ont extrait les motifs de courbure significatifs des trajectoires. Leur système extrait les positions des pics des courbes de trajectoire, représentant, ainsi, une action comme un ensemble de pics et les intervalles les séparant. L’apprentissage automatisé des actions humaines est possible dans leur système par la construction incrémentale de plusieurs prototypes représentant les modèles d’action. Ainsi l’ensemble du processus de reconnaissance peut être considéré comme une technique de *template-matching*.

Reconnaissance des actions avec des caractéristiques spatio-temporelles locales.

Les approches présentées dans cette section utilisent des caractéristiques locales extraites de volumes spatio-temporels 3-D pour représenter et reconnaître les actions. La motivation de ces approches réside dans le fait qu’un volume spatio-temporel 3-D est, essentiellement, un objet rigide en 3-D. Cela implique que si un système est en mesure d’extraire les caractéristiques appropriées décrivant les spécificités des volumes 3-D pour chaque action, l’action peut être reconnue par la résolution d’un problème de correspondance d’objets.

Chomat et al. [39] ont suggéré l’idée d’utiliser les descripteurs d’apparences locales pour caractériser une action, permettant ainsi la classification d’actions. Le système consiste à combiner un récepteur de champs d’énergie de mouvement avec des filtres de Gabor afin de capturer des informations de mouvement à partir d’une séquence d’images. Plus précisément, les caractéristiques d’apparence spatio-temporelles locales qui décrivent l’orientation du mouvement sont détectées par image. Des histogrammes multidimensionnels sont construits sur la base des caractéristiques locales détectées, et la probabilité a posteriori d’une action se produisant étant donné les caractéristiques détectées est calculée en appliquant la règle de Bayes. Ce système calcule en premier, la probabilité qu’un mouvement local se produise à chaque emplacement de pixel puis intègre ces probabilités dans la reconnaissance définitive des actions.

Zelnik-Manor et al. [230] ont proposé une approche utilisant les caractéristiques spatio-temporelles locales estimées sur différentes échelles temporelles. Plusieurs échelles temporelles de volumes de vidéos ont été analysées pour gérer les variations de vitesse d’exécution d’une action. Pour chaque point dans un volume 3-D (XYT), leur système estime un gradient normalisé d’intensité locale. similairement à **Chomat et al.** [39], ils estiment un histogramme des caractéristiques de gradient spatio-temporel par vidéo et présentent une mesure de distance basée sur l’histogramme (tout en ignorant les

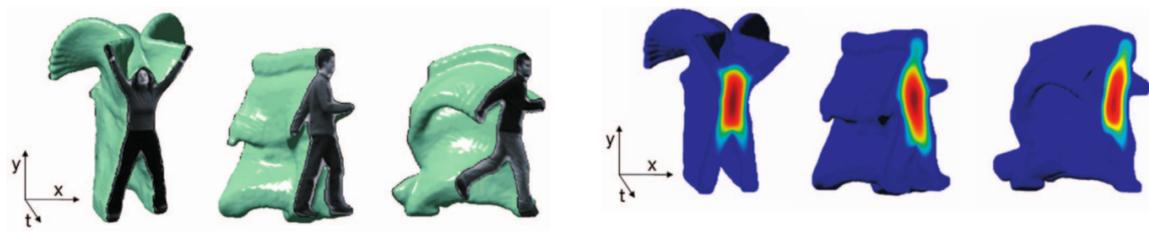


Figure 1.4 – Représentation des actions selon la méthode de **Blank et al.** [21] et **Gorelick et al.** [72] : Gauche : Exemples de volumes spatio-temporels. Droite : Solutions de l'équation de Poisson représentant ces volumes.

positions des caractéristiques extraites). Un algorithme de classification non supervisé est ensuite appliqué à ces histogrammes pour l'apprentissage des actions.

De même, **Blank et al.** [21] et **Gorelick et al.** [72] ont également calculé les caractéristiques locales à chaque image. Au lieu d'utiliser les flux optiques, ils ont calculé les caractéristiques locales basées sur l'apparence des silhouettes segmentées en construisant un volume spatio-temporel dont les valeurs sont les solutions de l'équation de Poisson. Ce modèle s'est révélé être capable d'extraire une grande variété de propriétés locales pertinentes. Leur système extrait des caractéristiques locales spatio-temporelles de saillance et des caractéristiques locales spatio-temporelle d'orientation à l'aide de l'équation. Chaque séquence d'une action est représentée comme un ensemble de caractéristiques globales calculées à partir des caractéristiques locales pondérées. Les auteurs ont appliqué une classification simple du plus proche voisin avec une distance Euclidienne pour la reconnaissance (**Figure 1.4**).

Laptev et al. [122] ont reconnu les actions humaines par l'extraction d'une distribution de points d'intérêt spatio-temporels à partir de vidéos. Ils ont généralisé les détecteurs de caractéristiques locaux de **Harris et al.** [80] couramment utilisés pour la reconnaissance d'objets, dans le but de détecter les points d'intérêt dans un volume spatio-temporel. Ce détecteur de points d'intérêt invariants en échelle recherche des angles spatio-temporels dans un espace 3-D (XYT), qui capture différents types de schémas de mouvements non constants (**Figure 1.5**). Les schémas de mouvement tels que les changements dans la direction de l'objet ou les occultations sont, aussi, détectés. En outre, **Schüldt et al.** [180] réussissent à distinguer de multiples actions en appliquant les SVM aux caractéristiques de **Laptev et al.** [122], illustrant ainsi, leur fiabilité pour la reconnaissance de l'action humaine.

Selon le même principe, **Dollár et al.** [50] ont proposé un nouveau détecteur de caractéristiques spatio-temporelles pour la reconnaissance des actions humaines (et animales). Leur détecteur est spécialement conçu pour extraire les points spatio-temporels avec des mouvements périodiques locaux, obtenant, ainsi, une répartition clairsemée de points d'intérêt à partir d'une vidéo. Une fois ces derniers détectés, le système associe un petit volume 3-D appelé cuboïde à chaque point d'intérêt (**Figure 1.5**).

1.4. Modéliser les “ACTIONS”

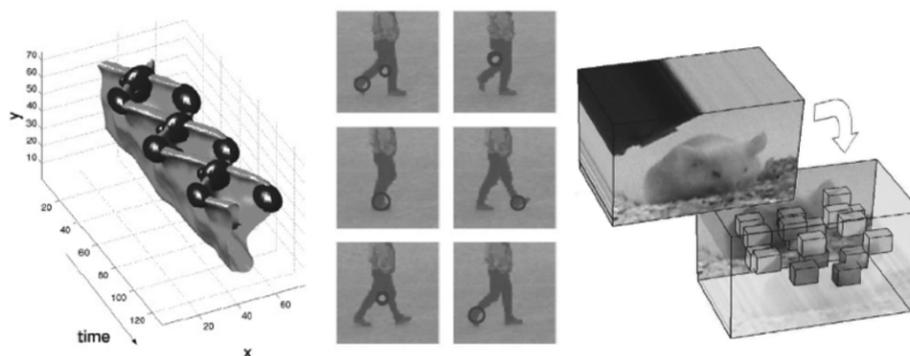


Figure 1.5 – Exemples de caractéristiques spatio-temporelles 3-D locales. Gauche : points d’intérêt extraits par la méthode de **Laptev et al.** [122]. Droite : caractéristiques cuboïdes extraites par la méthode de **Dollár et al.** [50].

Chaque cuboïde capture les valeurs de l’apparence des pixels appartenant au voisinage du point d’intérêt. Ils ont testé diverses transformations sur les cuboïdes afin d’extraire des caractéristiques locales finales. Ainsi, ils ont choisi le vecteur de gradients de luminosité qui montre la meilleure performance. Une bibliothèque de prototypes cuboïdes est construite pour chaque ensemble de données en regroupant les apparences des cuboïdes à l’aide de l’algorithme des k-moyennes. Par conséquent, chaque action est modélisée comme un histogramme de types cubiques détecté dans un volume spatio-temporel 3-D, tout en ignorant leurs emplacements (i.e. paradigme du sac-de-mots). Leur approche a connu plusieurs applications, reconnaissance de visages, comportements de souris, et enfin la reconnaissance des actions humaines. **Niebles et al.** [149] ont présenté une méthode d’apprentissage et de classification non supervisée pour les actions humaines en utilisant l’extracteur de caractéristiques ci-dessus [50]. Leur méthode de reconnaissance est une approche générative, modélisant les classes d’actions comme une collection de caractéristiques d’apparence spatio-temporelle. Une analyse sémantique latente probabiliste PLSA, couramment, utilisée dans le domaine de l’exploration de texte a été appliquée afin de reconnaître les actions statistiquement. Chaque élément dans la scène est classé dans une classe d’action en calculant sa probabilité a posteriori d’être généré par l’action.

Yilmazs et al. [227] ont proposé une modélisation des actions fondée à la fois sur la forme et le mouvement de l’objet. Lorsque l’objet effectue une action en 3D, les points de la frontière extérieure de l’objet sont projetés en forme de contour 2-D en (XY) dans le plan de l’image. La concaténation de ces contours 2-D génère dans le temps un volume spatio-temporel 3-D (XYT) en résolvant le problème de correspondance des points entre les images consécutives (**Figure 1.6**).

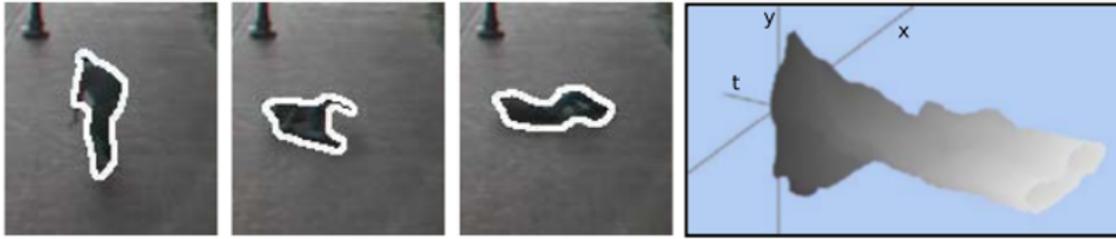


Figure 1.6 – Représentations des contours d’objets et le volume spatio-temporel 3-D (XYT) correspondant pour l’action “chuter” selon la méthode de **Yilmazs et al.** [227].

Les correspondances sont déterminées en utilisant une approche en deux étapes basée sur la théorie des graphes. Ils analysent les volumes en utilisant les propriétés de surface géométriques différentielles afin d’identifier les descripteurs à la fois spatiaux et temporels de l’action. Enfin, à l’aide de ces descripteurs, ils procèdent à la reconnaissance en utilisant la théorie des graphes.

Dans cet ordre d’idées, divers extracteurs de caractéristiques spatio-temporelles ont été développés récemment. **Scovanner et al.** [181] ont conçu une version 3-D des descripteurs SIFT, similaire aux caractéristiques cuboïdes de [50]. **Liu et al.** [131] ont présenté une méthodologie de raffinement des caractéristiques cuboïdes de manière à ne choisir que les caractéristiques importantes et significatives. **Bregonzio et al.** [26] ont proposé un détecteur amélioré pour extraire des caractéristiques cuboïdes, et présenté une méthode de sélection de celles-ci similaire à [131]. **Rapantzikos et al.** [165] ont généralisé les caractéristiques cuboïdes à la couleur aussi bien qu’à l’information de mouvement, contrairement aux méthodes précédentes qui utilisent uniquement les intensités.

Toutefois, ces approches ne modélisent pas la géométrie globale des caractéristiques locales, mais les considèrent comme un sac de caractéristiques. Différentes actions peuvent être composées de caractéristiques spatio-temporelles similaires, mais peuvent différer dans leurs relations géométriques.

l’intégration de la géométrie globale dans la représentation des parties de la vidéo est traitée dans les travaux **Boiman et al.** [24] et **Wong et al.** [214]. Contrairement aux approches suivant le paradigme “sac-de-mots”, ces approches tentent de modéliser la répartition spatio-temporelle des caractéristiques extraites pour une meilleure reconnaissance des actions. **Wong et al.** [214] ont étendu la PLSA en introduisant un modèle de forme implicite PLSA-ISM. Contrairement à la PLSA utilisée par **Niebles et al.** [149], cette dernière capture l’information spatio-temporelle relative des caractéristiques à partir du centre de l’action.

Savarese et al. [176] ont proposé une méthode pour capturer les informations de proximité spatio-temporelle entre les caractéristiques. Pour chaque vidéo d’action, ils ont mesuré les motifs des caractéristiques de co-occurrence dans une région locale 3-D, construisant ainsi des histogrammes appelés corrélogrammes spatio-temporels ST-

1.5. Reconnaître les “ACTIONS”

Correlograms.

Ryoo et al. [173] ont introduit le *spatio-temporal relationship match* STR-match, qui considère explicitement les relations spatiales et temporelles entre les caractéristiques détectées de manière à reconnaître les actions. Leur méthode mesure une similarité structurale entre deux vidéos en calculant les relations spatio-temporelles de paires de caractéristiques locales, permettant la détection et la localisation des actions simples ainsi que des activités complexes.

1.5

Reconnaître les “ACTIONS”

La reconnaissance de l’action humaine est un problème de classification. De ce fait, dans cette section nous traitons des approches qui classent les représentations d’images en actions. nous aborderons l’approche de classification des plus proches voisins, où une séquence observée est comparée à des séquences étiquetées ou encore à des prototypes représentant les actions. Une seconde classe de méthodes est celle des classifieurs discriminants. Ces derniers, quant à eux, apprennent une fonction discriminante entre deux ou plusieurs classes en opérant directement sur les modélisations des actions. La réduction de dimensionnalité étant, souvent, une étape préliminaire commune aux deux types de classification, elle sera discutée en premier.

1.5.1 Réduction de dimensionnalité

La plupart des approches en reconnaissance de l’action impliquent le traitement des données dans des espaces de très grandes dimensions. Par conséquent, ces approches souffrent souvent de la “malédiction de la dimensionnalité”. En effet, l’espace des caractéristiques se disperse de façon exponentielle proportionnellement à la dimension, nécessitant ainsi un plus grand nombre d’échantillons pour construire des modèles de classes conditionnelles efficaces. L’apprentissage de la variété sur lequel résident les données nous permet de déterminer la dimension intrinsèque des données, plutôt que la dimension brute. Celle-ci contient moins de degrés de liberté et permet la conception de modèles efficaces dans des espaces de faible dimension.

Une des façons les plus simples pour réduire la dimensionnalité est l’analyse en composantes principales PCA. Cette dernière a été utilisée par **Masoud et al.** [139] et **Rosales** [167] dans leur approche de reconnaissance. Cependant la PCA suppose les données linéaires, ce qui n’est, généralement, pas le cas. Nécessitant, ainsi, des méthodes qui apprennent la géométrie intrinsèque de la variété à partir d’un grand nombre

d'échantillons. **Chin et al.** [38] ont opté pour l'approche *local linear embedding* LLE. **Wang et al.** [207] ont utilisé, quant à eux, l'approche *locality preserving projections* LPP. Enfin, **Blackburn et al.** [20] ont utilisé l'Isomap. Toutefois, ces méthodes de réduction de dimensionnalité sont des approches non supervisées, pouvant ne pas garantir une bonne discrimination entre les classes. **Poppe et al.** [160] abordent cette question en apprenant les caractéristiques transformées discriminantes entre des paires de classes. **Jia et al.** [99] ont utilisé une approche discriminatoire à la fois dans le sens spatial et temporel.

1.5.2 k -ppv

L'approche des k plus proches voisins (k -ppv) utilise la distance entre la représentation ou le modèle d'une séquence observée et les représentations ou les modèles dans un ensemble d'entraînement. L'étiquette la plus courante parmi les k plus proches séquences d'entraînement est choisie comme la classe représentant la séquence. Une classification basée sur les plus proches voisins peut être appliquée soit au niveau de chaque image de la séquence à tester, ou encore être appliquée à des séquences entières. Dans ce dernier cas, des difficultés liées aux différentes longueurs de séquences peuvent survenir et doivent être résolues, par exemple à l'aide d'un vote majoritaire entre les sous-séquences la composant. **Blank et al.** [21] ont adopté le 1-ppv à l'aide d'une distance Euclidienne entre les caractéristiques globales dans leur approche, de même que **Batra et al.** [14] mais appliqué aux histogrammes de mots de code.

1.5.3 Classifieurs discriminants

Les classifieurs discriminants se concentrent sur la séparation de deux classes ou plus, plutôt que la modélisation de celles-ci. Les machines à vecteurs de support SVM ont largement été utilisées en combinaison avec des représentations locales de longueurs fixes, telles que les histogrammes de mots de code dans [98, 121, 180]. Selon le même principe, les machines à vecteurs de pertinence (*Relevance Vector Machine* en anglais, RVM), une variante probabiliste des SVM, ont été utilisées pour la reconnaissance de l'action dans [155]. Les autres approches largement utilisées sont celles basées sur le *boosting*, soit comme une étape de sélection de caractéristiques discriminatoires ou en tant que classifieur : AdaBoost dans [60, 124, 154] et LPBoost dans [150].

1.6 Bases de données des actions humaines

Ces dernières années, de plus en plus de jeux de données vidéos publics pour la reconnaissance des actions et/ou des activités humaines ont été créés. L'accessibilité à ces bases de vidéos permet d'une part, une économie en temps et en ressources afin que les chercheurs se concentrent principalement au développement de méthodes de reconnaissance. Et d'autre part, elle facilite et favorise la comparaison des différentes approches de reconnaissance afin de fournir un aperçu des capacités de ces dernières.

Cette section se concentre principalement sur les ensembles de données composés d'actions humaines hétérogènes, à savoir, des actions typiques enregistrées à l'aide de caméras à spectre visible et effectuées selon divers scénarios réalistes.

Néanmoins, il existe d'autres bases de vidéos dédiées à la reconnaissance d'actions très spécifiques telles que la détection d'objets abandonnés, la reconnaissance des activités de la vie quotidienne (recognition of activities of daily living, en anglais, ADL), le comportement de la foule, la détection de chute humaine, l'analyse de la marche ou encore la reconnaissance de gestes.

Nous rappelons qu'une action peut être considérée comme une séquence de mouvements primitifs remplissant une simple fonction telles que marcher, sauter, etc. D'autre part, une activité est composée de séquences d'actions dans l'espace et le temps telle qu'une personne préparant un plat ou encore des gens jouant une partie de football. La principale caractéristique d'une activité est la notion d'interaction entre une personne et une ou plusieurs autres personnes ou une interaction entre une personne et une ou plusieurs personnes et des objets de l'environnement. Toutefois, les différences entre actions et activités ne sont pas toujours claires. En effet, le déplacement d'une personne d'un endroit à un autre peut être considéré comme une simple action ou bien comme une activité si celle-ci se déplace en évitant des obstacles, d'où le fait qu'un grand nombre des jeux de données présentés dans cette section ne distinguent pas entre action et activité.

La chronologie d'apparition des différents ensembles de vidéos est étroitement liée aux défis envisagés par la communauté scientifique dans la résolution du problème de reconnaissance automatique des actions et des activités à partir de vidéos.

Ainsi le premier défi était d'analyser une seule action effectuée par un seul humain, d'où l'apparition des ensembles de données *WEIZMANN action as space-time shapes* (2001) [73], *WEIZMANN event-based analysis* (2005) [229] et enfin *KTH recognition of human actions* (2004) [119]. Dans celles-ci les actions sont effectuées de manières très similaires selon un point de vue fixe, dans un contexte statique et des conditions

contrôlées.

Afin de gérer des situations plus complexes dans des environnements réels, de nouvelles bases de données vidéos ont été enregistrées dans des conditions plus réalistes où les conditions d'éclairage ne sont pas contrôlées (extérieur) et les milieux sont complexes et multimodaux telles que la base *Context Aware Vision using Image-based Active Recognition, CAVIAR* (2004) [1], la base *Evaluation du Traitement et de l'Interpretation de Sequences Video, ETISEO* (2005) [2], la base *CASIA action database* (2007) [3], la base *MSR action dataset* (2009) [94], la base *UTexas databases* [4] composée des deux sous-ensembles *UT-interaction dataset* et *UT-tower dataset* (2010), la base *VIRAT video dataset* (2011) [5] et la base *Video Surveillance On-line Repository for Annotation Retrieval, ViSOR* (2005) [153].

D'autres bases de données ont, quant à elles, été recueillies à partir du contenu Web, principalement à partir de la plate-forme Youtube, telles que la base *HOLLYWOOD & HOLLYWOOD-2 : human actions datasets* (2008) [118], la base *UCF datasets* [151] composée de cinq sous-ensembles *UCF aerial action dataset* (2007), *UCF-ARG* (2008), *UCF sports action dataset* (2008), *UCF YouTube action dataset* (2009) et *UCF50* (2010), la base *Olympic sports dataset* (2010) [201] et enfin la base *HMDB51, a large video database for human motion recognition* (2011) [117].

Bien que la majorité de ces bases intègre la notion d'interaction entre humain-humain et humain-objet, d'autres bases de vidéos ont été, spécifiquement, créées pour ce type de problèmes telles que la base *BEHAVE, computer-assisted prescreening of video streams for unusual activities* (2004) [63] et la base *TV human interactions dataset* (2010) [76].

La caractéristique commune aux ensembles de données précédents est qu'ils sont tous destinés à l'analyse et la reconnaissance de mouvements selon un seul point de vue observationnel. Toutefois, ces dernières années, la recherche s'intéresse d'avantage à la compréhension des comportements humains dans des grands espaces publics impliquant plusieurs caméras tels que les aéroports ou les stations de métro et donc à la compréhension des comportements humains selon de multiples points de vue. Dans cette optique, plusieurs ensembles de données ont été créés pour étudier les problèmes liés à ce contexte telles que la base *INRIA Xmas Motion Acquisition Sequences, IXMAS* (2006) [93], la base *i3DPost multi-view dataset* (2009) [152], la base *Multicamera Human Action Video Data, MuHAVi* (2010) [200] et la base *VideoWeb dataset* (2010) [75].

Le travail présenté dans ce mémoire s'inscrit dans le cadre de la reconnaissance des actions humaines, à savoir, mettre en œuvre un système basé sur une approche spatio-temporelle pour la reconnaissance d'actions humaines. L'objectif de notre méthode est d'exploiter l'information globale d'un volume spatio-temporel à l'aide d'un processus d'extraction de caractéristiques globales afin de procéder à la reconnaissance des actions de façon automatique, efficace et particulièrement simple. Pour ce faire, nous

1.6. Bases de données des actions humaines

avons choisi de modéliser nos prototypes d'actions humaines à l'aide de la technique de réduction de dimensionnalité *Multi-Dimensional Scaling* MDS et ainsi visualiser les caractéristiques spatio-temporelles globales que prend la forme de la silhouette d'un sujet dans le temps pour une action donnée. Le choix de la MDS a été motivé, d'une part, par sa capacité à représenter les données en espace de dimension réduite tout en respectant la géométrie globale de l'action dans le temps en considérant les relations spatiales et temporelles entre les silhouettes, et d'autre part, par les bon résultats que celle-ci a obtenu lors de son application dans divers champs de traitement d'images et de vision par ordinateur tels que la reconnaissance faciale [19] et la reconnaissance d'objets [208].

2

ÉTUDE BIBLIOGRAPHIQUE

Sommaire

2.1	Introduction	24
2.2	Algorithmes d'apprentissage	24
2.2.1	Les types d'apprentissage	25
2.2.2	Apprentissage supervisé : méthodes de classification	26
2.3	Réduction de la dimensionnalité	34
2.3.1	La sélection de caractéristiques	35
2.3.2	L'extraction de caractéristiques	37
2.4	Détection de mouvements	42
2.4.1	Étapes d'une opération de soustraction de fond	43
2.4.2	Techniques de modélisation du fond de référence	46

2.1 Introduction

L'analyse des activités humaines, sur la base de séquences vidéos, nécessite différents niveaux de traitements. Les traitements de bas niveaux qui consistent en la détection des zones de mouvements pertinents. Les traitements de niveaux intermédiaires qui comprennent l'extraction d'information visuelle et leur représentation sous une forme concise et qui est la plus invariante possible. Enfin, les traitements de hauts niveaux permettant l'interprétation de ces informations et la reconnaissance de l'activité humaine. Il existe dans la littérature, une multitude de techniques pour mener à bien chacune de ces trois étapes.

Ces dernières années, les chercheurs ont de plus en plus recouru aux techniques d'apprentissage machine, notamment dans la détection de mouvements et soustraction de fond. De ce fait nous présentons, tout d'abord, ces techniques afin de permettre une meilleure compréhension des autres sections de cette étude bibliographique et d'éviter toute redondance dans les définitions. Puis nous poursuivons avec une description succincte des techniques de réduction de dimensionnalité et enfin nous exposons un bref état de l'art des techniques de détection de mouvements.

2.2 Algorithmes d'apprentissage

Au cours des deux dernières décennies, qu'ils soient sous la forme d'informations mises à disposition par des individus via le web (images, textes, sons, vidéos), de bases de données collectées (clients, mesures diverses, médicales) ou encore sous forme de données générées par des applications (surveillance, sécurité, production), la collecte et le partage d'informations ont pris une telle ampleur que le volume de données stockées sous forme numérique ne cesse de croître en quantité et en variétés. Tant il est vrai que ces données nous fournissent un nombre important d'informations hétérogènes : numériques, catégorielles, courbes, etc, il nous manque souvent la connaissance. Il existe dès lors un très grand intérêt à développer des outils permettant d'exploiter au mieux tous ces stocks d'informations afin d'en extraire un maximum de savoir, menant ainsi à l'émergence de l'apprentissage machine. L'apprentissage machine est un domaine dont l'intérêt majeur est d'extraire de telles connaissances à l'aide d'algorithmes permettant la résolution de tâches complexes. À la différence des algorithmes classiques, les algorithmes d'apprentissage automatique intègrent la notion d'intelligence. Pour

2.2. Algorithmes d'apprentissage

être intelligent, un système évoluant dans un environnement changeant devrait avoir la capacité d'améliorer ses performances à partir de données acquises en cours de fonctionnement et ainsi apprendre de son expérience. Partant de ce fait, les techniques d'apprentissage machine ont été développées de sorte à modéliser l'apprentissage d'un point de vue mathématique afin de générer un modèle optimisant un critère de performance et d'analyser de manière automatique un ensemble limité de données représentatives d'une tâche précise : phase d'entraînement, en vue d'être appliqués sur de nouvelles données : phase de test. Le modèle peut être prédictif, ainsi prévoir des valeurs futures, descriptif pour acquérir des connaissances et détecter des schémas à partir des données, ou les deux.

2.2.1 Les types d'apprentissage

Il existe plusieurs types différents d'apprentissage automatique. Ces derniers se distinguent essentiellement par leurs objectifs, i.e : la nature de ce qui doit être appris. Usuellement, les tâches d'apprentissage sont divisées en deux principales catégories :

Dans les approches prédictives ou supervisées, l'apprentissage correspond au cas où l'objectif de celui-ci est déterminé explicitement via la définition d'une cible à prédire. Dès lors, l'objectif est d'apprendre un modèle qui décrit au mieux la relation entre un attribut \mathbf{x} (entrée) et son label \mathbf{y} (sortie ou cible), étant donné un ensemble de paires d'entrées/sorties $D = \{(x_i, y_i)\}_{i=1}^N$ où D est l'ensemble d'entraînement et N est le nombre d'exemples. Dans le contexte le plus basique, chacune des variables d'entrée de l'ensemble d'entraînement x_i est représentée sous la forme d'un vecteur numérique à m dimensions, néanmoins, le plus souvent la structure de l'entrée est plus complexe pouvant prendre la forme d'une image, d'une chaîne de caractères, ou encore la forme d'un graphe, etc. De même, la variable de sortie ou la réponse peut, en principe, prendre diverses formes. Toutefois, la plupart des méthodes supposent que la sortie y_i soit sous la forme d'une valeur nominale, dès lors la tâche devient un problème de classification et/ou de reconnaissance de forme ou bien une valeur réelle pour un problème de régression.

L'autre catégorie d'approche sont, les approches descriptives ou non supervisées. Ces dernières se distinguent des approches supervisées par le fait qu'elles ne bénéficient pas d'un oracle pour les guider. Elles ne reposent ni sur une fonction d'évaluation, ni sur l'étiquetage d'un échantillon d'exemples. Ainsi l'objectif, dans ce cas, n'est plus de décrire des relations entre entrées et sorties mais d'extraire un schéma de connaissance intéressant à partir des données. En effet l'idée consiste à partitionner un ensemble de données hétérogènes en sous-ensembles de façon à ce que les données relativement similaires soient associées au sein d'ensembles homogènes et vice-versa, les données relativement différentes se retrouvent dans des ensembles distincts [95].

Il existe un troisième type d'apprentissage automatique appelé apprentissage par renforcement, néanmoins ce dernier reste très peu répandu. L'idée générale de celui-ci est l'apprentissage par récompenses et/ou punitions. L'apprentissage par renforcement s'adresse aux entités autonomes et permanentes dans un environnement dont la structure est inconnue par celles-ci. Ces entités apprennent de leurs interactions (erreurs, succès) avec le milieu qui les entoure et ainsi optimiser une certaine fonction de gain. De ce fait, contrairement aux types précédents, l'objectif ici est l'association d'états de l'environnement à une action, i.e : apprendre un comportement [104, 190, 172].

Bien qu'elles puissent trouver application dans des contextes différents, ces trois approches d'apprentissage peuvent aussi être combinées dans un même système. Cela engendre une large quantité de variantes d'un même algorithme de base.

C'est sur un problème de classification qu'a porté notre travail, de ce fait, nous limiterons notre attention, dans ce qui suit, aux méthodes supervisées de classification. Pour approfondir la curiosité sur les autres types d'apprentissage, le lecteur est invité à se référer aux documents cités plus haut.

2.2.2 Apprentissage supervisé : méthodes de classification

La classification est probablement la forme la plus répandue de l'apprentissage automatique, elle a permis la résolution de nombreux problèmes intéressants et souvent difficiles du monde réel. Une grande variété d'applications peut être vue comme des tâches de classification, telle que la prédiction de faillite, l'inspection de produit, le diagnostic médical ou encore la reconnaissance de formes (paroles, écritures manuscrites, visages, actions, etc.).

L'objectif principal de la classification est d'être capable d'étiqueter des données en leur associant une classe à travers un modèle descriptif appris à partir de l'exploration d'exemples déjà classés. De manière plus formelle, chaque exemple est décrit par un nombre de mesures groupées dans un vecteur d'attributs $x \in X^m \subseteq \mathbb{R}^m$ ainsi que par une étiquette $y \in Y = \{1, 2, \dots, K\}$ qui lui est associée. Le processus d'apprentissage établit une transformation h de l'espace d'attributs vers l'espace d'étiquettes $X^m \xrightarrow{h} Y = \{1, 2, \dots, K\}$. Cette transformation est le classifieur dont le rôle est d'associer un label à tout nouvel exemple ne disposant pas d'informations a priori [18].

La conception d'un classifieur sur la base d'un ensemble d'apprentissage repose essentiellement sur trois points [61] :

- Le choix d'une structure de classifieur (arbres de décision, discriminateur linéaire, ..., etc.);
- Le critère d'évaluation des performances du classifieur. Le critère le plus classique est le taux de bonne décision. D'autres facteurs peuvent également être pris en considération comme par exemple, le temps d'exécution, la stabilité,

2.2. Algorithmes d'apprentissage

l'interprétabilité ou encore des contraintes de performance ;

- La sélection du modèle. La construction d'un classifieur s'apparente à un processus de recherche dans un espace de classifieurs. Une fois, la structure et le critère choisis, la construction d'un classifieur devient un problème d'optimisation où, l'algorithme de recherche est utilisé pour trouver un ensemble de paramètres qui optimise ce critère.

Dans la suite, nous présentons les principaux algorithmes de classification supervisée proposés dans la littérature. Il ne s'agit pas de faire une présentation exhaustive de toutes les méthodes mais seulement de préciser les méthodes les plus classiques. Par souci de concision, la description de chaque algorithme est succincte, se concentrant sur les points importants pour la compréhension de leurs principes.

2.2.2.1 k -plus proches voisins

La méthode des k -plus proches voisins k -ppv (*k-nearest neighbor* en anglais, k -NN) [92] se base sur une comparaison directe entre le vecteur caractéristique représentant l'entité à classer et les vecteurs caractéristiques représentant des entités de référence. Le principe est d'assigner à la donnée d'entrée la classe majoritaire parmi ses k plus proches voisins dans l'échantillon d'apprentissage. En effet, étant donné une mesure de distance dans l'espace d'entrée \mathbb{R}^m , la prédiction du modèle sur un exemple de test $x \in T$ où, T est l'ensemble de test, dépend uniquement des k plus proches voisins de x dans l'ensemble d'entraînement D . En notant $i_1(x), \dots, i_k(x)$ les indices des k exemples de D les plus proches de x selon la distance choisie. Pour un problème de classification, la prédiction du modèle est, dès lors, un vote parmi les k voisins :

$$f(x) = \arg \max_y \sum_{j=1}^k \mathbf{1}_{y=y_{i_j(x)}} \quad (2.1)$$

où, en cas d'égalité parmi les votes, le modèle choisit aléatoirement l'une des classes majoritaires.

Il existe de nombreuses variantes de cette méthode, selon la fonction de distance utilisée ou encore selon la pondération des voisins entre eux [183]. Les distances suivantes sont usuellement employées par les classificateurs k -ppv :

Notons par $X_p = (x_{p1}, x_{p2}, \dots, x_{pm})$ le vecteur caractéristique de l'entité p , avec m le

nombre de caractéristiques et par p et q deux entités à comparer.

$$\text{Distance Euclidienne : } D(X_p, X_q) = \sqrt{\sum_{i=1}^m (x_{pi} - x_{qi})^2} \quad (2.2)$$

$$\text{Distance Manhattan : } D(X_p, X_q) = \sum_{i=1}^m (|x_{pi} - x_{qi}|) \quad (2.3)$$

$$\text{Distance Minkowski : } D(X_p, X_q) = \left(\sum_{i=1}^m (x_{pi} - x_{qi})^r\right)^{1/r} \quad (2.4)$$

$$\text{Distance Tchebychev : } D(X_p, X_q) = \max_{i=1}^m (|x_{pi} - x_{qi}|) \quad (2.5)$$

La principale limite de cette approche est d'être coûteuse, notamment à cause de la recherche de voisins dans un échantillon potentiellement grand, ainsi que sa sensibilité au bruit, potentiellement présent dans les données d'apprentissage.

2.2.2.2 Inférence d'arbres de décision

Le formalisme des arbres de décision permet d'attribuer une classe à un nouvel exemple en testant ses caractéristiques séquentiellement. Ces tests sont organisés hiérarchiquement, de sorte à ce que la réponse à un test indique quel est le prochain à effectuer, et ainsi de suite jusqu'à ce que le dernier pointe sur la réponse finale, i.e : la classe. L'apprentissage, ici, consiste à choisir les variables testées à chaque nœud, les seuils de comparaison, la profondeur de l'arbre, ainsi que la fonction de décision associée à chaque feuille. En effet, Dans la phase de construction de ce classifieur, les exemples d'apprentissage sont divisés récursivement par des tests définis sur les caractéristiques pour obtenir des sous-ensembles d'exemples ne contenant que des exemples appartenant tous à une même classe. Cette approche est connue sous le nom d'induction descendante d'arbres de décision. Les algorithmes fondateurs basés sur cette idée sont CART [27] et ID3 [163], ils diffèrent essentiellement par leur façon de choisir la caractéristique de segmentation, à une étape donnée et par le critère d'arrêt. L'avantage des arbres de décision est qu'ils sont souvent concis et compréhensibles. En outre, à la différence de la méthode des k -ppv, la décision est peu coûteuse à prendre une fois l'arbre obtenu. À l'inverse leur utilisation impose une certaine structure de données compatible qui, selon le problème, peut être difficile à obtenir ou trop coûteuse à exploiter. De plus, cette approche est très sensible au problème de sur-apprentissage et peu robuste face aux données manquantes, ce qui constitue les obstacles majeurs à son application sur des cas réels.

2.2. Algorithmes d'apprentissage

2.2.2.3 Machines à vecteurs de support

Les machines à vecteurs de support SVM (*Support Vector Machines* en anglais) sont des méthodes d'apprentissage pour la classification binaire, motivées par les résultats de la théorie de l'apprentissage statistique. Il s'agit d'algorithmes fondés sur la notion de marge. L'idée des algorithmes de SVM est de partager l'espace en deux parties à l'aide d'un hyperplan qui maximise la distance minimale des observations à ce plan (i.e : la marge) dans le but d'obtenir, par la suite, une meilleure généralisation, i.e : la faculté d'un classifieur à prédire correctement les classes de nouvelles observations et non pas seulement les classes des observations d'apprentissage [42, 25, 202, 30].

Dans le cas d'une séparation linéaire de deux classes, il s'agit de trouver l'hyperplan qui sépare les classes tout en maximisant la marge. L'équation d'un hyperplan étant :

$$h(x) = w \cdot x + w_0 \quad (2.6)$$

où w est le vecteur normal de l'hyperplan et w_0 une constante représentant son origine.

Un exemple (x_i, y) est bien classé si et seulement si :

$$y \cdot h(x_i) > 0 \quad (2.7)$$

Une première expression du problème est alors la suivante :

$$\begin{cases} \min(\frac{1}{2} \|w\|^2) \\ \forall i, y_i(w \cdot x_i + w_0) \geq 1 \end{cases} \quad (2.8)$$

où w (le vecteur normal de l'hyperplan) et w_0 sont les paramètres à trouver, x_i et y_i sont les données de l'échantillon d'apprentissage. Tel quel, le problème est difficile (voire impossible) à résoudre lorsque la dimension des données d'entrée est grande. D'où sa reformulation sous une forme duale, qui ne dépend plus de la dimension des données mais de la taille de l'échantillon d'apprentissage et qui est :

— trouver les multiplicateurs de Lagrange α tels que :

$$\begin{cases} \max_{\alpha} \left\{ \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j (x_i, x_j) \right\} \\ \alpha_i \geq 0, i = 1, \dots, N \\ \sum_{i=1}^N \alpha_i y_i = 0 \end{cases} \quad (2.9)$$

où N est la taille de l'échantillon d'apprentissage.

— la solution de l'hyperplan est alors donnée par :

$$h(x) = (w^* \cdot x) + w_0^* = \sum_{i=1}^N \alpha_i^* y_i (x \cdot x_i) + w_0^* \quad (2.10)$$

où les α_i^* sont les solutions de l'équation (3.2) et où w_0^* peut être calculé à partir d'un vecteur de support. Les vecteurs de support sont les seuls à avoir un multiplicateur de Lagrange non nul, ils sont ainsi les seuls à définir l'hyperplan optimal. C'est pourquoi ils sont, parfois, appelés "exemples critiques".

Dans le cas non-linéaire la solution consiste à transformer l'espace de représentation de l'échantillon d'apprentissage en un espace de plus grande dimension dans lequel il existe une séparation linéaire. Mais comment trouver cette transformation non-linéaire Φ ? En pratique cela équivaut souvent à connaître la solution d'avance. C'est ici qu'interviennent les fonctions noyaux. Muni de Φ le problème à résoudre serait de trouver les α tels que :

$$\left\{ \begin{array}{l} \max_{\alpha} \left\{ \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j (\Phi(x_i), \Phi(x_j)) \right\} \\ \alpha_i \geq 0, i = 1, \dots, N \\ \sum_{i=1}^N \alpha_i y_i = 0 \end{array} \right. \quad (2.11)$$

On remarque que Φ n'intervient que dans le produit scalaire $\Phi(x_i), \Phi(x_j)$. Plutôt que de trouver Φ , on peut donc chercher à trouver la fonction k telle que :

$$K(x, x') = \Phi(x) \cdot \Phi(x') \quad (2.12)$$

Cette fonction K est une fonction noyau. Elle permet, lorsqu'elle est bien choisie, d'utiliser des représentations non vectorielles et d'éviter de calculer la représentation des exemples dans le nouvel espace. Plusieurs noyaux sont couramment utilisés et parfois combinés : noyau linéaire, polynomial, gaussien ou laplacien [179].

Enfin, comme cité précédemment, les méthodes de SVM classiques sont des méthodes de classification binaire, le cas multi-classes doit être décomposé en un ensemble de problèmes à deux classes ou traité directement par des méthodes multi-classes [126, 212].

2.2. Algorithmes d'apprentissage

2.2.2.4 Approche Bayésienne

Un classifieur Bayésien [97, 148] est basé sur une approche probabiliste employant la fameuse règle de Bayes :

$$P(\Theta|D) = \frac{P(D|\Theta)P(\Theta)}{P(D)} \quad (2.13)$$

où Θ représente les paramètres d'un modèle et D l'ensemble d'entraînement. $P(D|\Theta)$ est la probabilité d'observer les données D en supposant qu'elles ont été générées par le modèle dont les paramètres sont Θ , aussi appelée vraisemblance. $P(\Theta)$ est la probabilité a priori. Dès lors, $P(\Theta|D)$, appelée probabilité a posteriori, indique la probabilité des paramètres après avoir observé les données.

L'importance pratique de la règle de Bayes tient au fait qu'elle permet de ré-exprimer la probabilité a posteriori, difficile à calculer, en terme de probabilités a priori et conditionnelles plus faciles à obtenir.

L'analyse discriminante se présente comme un cas particulier de l'approche Bayésienne. Dans ce cas, les données d'apprentissage sont modélisées par des distributions Gaussiennes. Sur la base des paramètres estimés, des fonctions discriminantes sont construites permettant, ainsi, de classer tout vecteur de caractéristiques.

2.2.2.5 Algorithmes du Perceptron

a. Perceptron (linéaire)

Le perceptron est un classifieur linéaire proposé initialement par **Rosenblatt** [168], qui peut être brièvement décrit comme suit :

étant donné un exemple $\mathbf{x} = [x_1, x_2, \dots, x_m]^t$ et $\mathbf{w} = [w_1, w_2, \dots, w_m]^t$ un vecteur de poids associés aux attributs, aussi nommé vecteur de prédiction. Le perceptron calcule la somme pondérée des attributs $\sum_{i=1}^m w_i x_i$, puis, cette dernière, est comparée à un seuil θ pour obtenir l'estimation du label \hat{y} de \mathbf{x} par exemple au moyen de la fonction signe comme suit :

$$Z = \mathbf{w} \cdot \mathbf{x} - \theta = \sum_{i=1}^m z_i x_i - \theta \quad (2.14)$$

avec

$$\hat{y} = \text{sign}(Z) = \begin{cases} 0 & \text{si } Z < 0 \\ 1 & \text{sinon} \end{cases} \quad (2.15)$$

L'algorithme est, généralement, exécuté à plusieurs reprises sur un ensemble d'ap-

prentissage jusqu'à ce qu'il trouve un vecteur optimal de prédiction permettant de classer correctement tous les exemples de l'ensemble d'apprentissage. Autrement dit, le vecteur de prédiction est mis à jour selon la formule $\mathbf{w} = \mathbf{w} + y\mathbf{x}$ tant que la prédiction \hat{y} diffère du label vrai y , sinon \mathbf{w} n'est pas modifié. Ce dernier est, par la suite, utilisé pour prédire les labels inconnus de nouveaux exemples.

Les approches basées sur le perceptron ont l'avantage d'offrir une faible complexité de calcul dans les cas où, peu d'attributs sont pertinents. Toutefois, notons que ces dernières sont des méthodes de classification binaire. Pour le problème multi-classes, il faut donc se ramener à un ensemble de problèmes de classification binaire.

b. Réseaux de neurones (non-linéaire)

Le perceptron est, en réalité, un réseau de neurones mono-couche, pouvant uniquement classer les données linéairement séparables. Toutefois, dans la plupart des applications les données sont rarement séparables linéairement. Une solution à cette limitation est proposée par **Fiesler** [62] consistant à étendre l'architecture du perceptron à une architecture multi-couches.

Un réseau de neurones multi-couches se compose d'une série de couches d'unités dites neurones, auxquels sont associés des poids. À la réception de signaux provenant de neurones appartenant à une couche précédente du réseau, un neurone réagit en produisant un signal de sortie, nommée activation, qui sera transmis à d'autres neurones appartenant à la couche suivante du réseau.

Dans les perceptrons multi-couches, les neurones formels se classent en trois catégories :

- Les neurones d'entrée servent à transmettre les données d'entrée (les exemples de l'échantillon d'apprentissage aussi bien que les futurs exemples à classer) ;
- les neurones de sortie sont ceux qui fournissent l'hypothèse d'apprentissage. Chaque neurone de sortie correspond à une classe ;
- Les neurones cachés sont exclusivement connectés à d'autres neurones et non aux entrées/sorties du réseau. Ils effectuent des traitements intermédiaires.

En résumant, un réseau de neurones dépend de trois aspects fondamentaux :

1. Les fonctions d'activation des neurones :

Une hypothèse est, que chaque neurone fournit une contribution aux neurones qui lui sont connectés. Ces contributions s_k sont pondérées, sommées puis complétées par un coefficient de biais θ_k comme suit :

$$s_k = \sum_j w_{jk} \cdot y_j + \theta_k \quad (2.16)$$

où w_{jk} est le poids qui détermine l'effet du neurone j sur le neurone k et y_j représente l'état d'activation (la sortie) du neurone j connecté au neurone k .

une fois s_k calculé, une fonction d'activation g_k détermine l'état d'activation \hat{y}_k du

2.2. Algorithmes d'apprentissage

neurone k donnée par la relation :

$$\hat{y}_k = g_k(s_k) \quad (2.17)$$

Les fonctions d'activation suivantes sont des choix admissibles utilisés le plus fréquemment :

$$\text{Fonction sigmoïde : } g(s) = \frac{1}{1 + e^{-\lambda s}} \quad (2.18)$$

$$\text{Fonction seuil : } g(s) = \text{sign}(s) \quad (2.19)$$

$$\text{Fonction rectified linear unit : } g(s) = \max(0, s) \quad (2.20)$$

$$\text{Fonction logistique : } g(s) = \frac{1}{1 + e^{-s}} \quad (2.21)$$

$$\text{Fonction tangente hyperbolique : } g(s) = \tanh(s) = \frac{e^{2s} - 1}{e^{2s} + 1} \quad (2.22)$$

$$\text{Fonction arc tangente : } g(s) = \frac{2}{\pi} \arctan\left(\frac{\pi s}{2}\right) \quad (2.23)$$

néanmoins, les fonctions sigmoïdes restent les plus populaires grâce à leur propriété :

$$\begin{cases} 0 & \text{si } s \rightarrow -\infty \\ 1 & \text{si } s \rightarrow \infty \end{cases} \quad (2.24)$$

2. L'architecture du réseau (*feed-forward* ou récurrent) :

Il existe, principalement, deux topologies de réseaux de neurones :

- Les réseaux de neurones *feed-forward*, dans lesquels, les neurones ne sont connectés que dans le sens de l'entrée vers la sortie et sans aucune rétroaction. Ce type est relativement simple et est couramment utilisé dans beaucoup de domaines ;
- Les réseaux de neurones récurrents contenant des connexions de rétroaction. En effet les neurones de sortie peuvent par exemple voir leur sortie utilisée comme entrée d'un neurone de la couche précédente ou de la même couche.

3. La détermination des poids des connexions :

Enfin, une fois les deux premiers aspects fixés, la performance du réseau de neurones est alors définie par les valeurs des poids. Ces derniers sont, habituellement, initialisés aléatoirement. Les données de l'échantillon d'apprentissage sont, par la suite, passées séquentiellement au réseau afin d'ajuster les poids à chaque passage dans le sens qui amène la valeur de sortie du réseau au plus proche de celle attendue. L'ajustement des poids est dicté par ce que l'on appelle la règle delta. De manière plus rigoureuse, il s'agit de la méthode de rétro-propagation du gradient (*backpropagation* en anglais) comme suit :

$$\Delta w_{ji} = \eta \delta_j \hat{y}_i \quad (2.25)$$

- ★ η est un nombre positif (nommé le taux d'apprentissage), qui détermine la taille d'un pas dans la recherche descendante du gradient ;
- ★ \hat{y}_i est la sortie calculée du neurone i ;
- ★ pour les neurones de sortie, $\delta_j = \hat{y}_j(1 - \hat{y}_j)(y_j - \hat{y}_j)$ où y_j est la sortie attendue du neurone j ;
- ★ pour les neurones cachés, $\delta_j = \hat{y}_j(1 - \hat{y}_j) \sum_k \delta_k w_{kj}$.

Hormis leur grande lenteur, un des problèmes majeurs de l'algorithme de rétro-propagation du gradient original et ses variantes réside dans la convergence vers des minima locaux. Parmi les solutions pour accélérer le processus d'apprentissage, serait d'estimer les poids initiaux au lieu de les tirer au hasard [221].

Diverses études ont comparé les performances entre les réseaux de neurones et les classifieurs conventionnels [47, 89, 144] , celles-ci illustrent que les réseaux de neurones présentent l'avantage d'être des méthodes auto-adaptatives, s'adaptant aux données sans aucune forme explicite de la distribution pour le modèle sous-jacent. Les réseaux de neurones ont, également, la propriété d'être des modèles non-linéaires permettant plus de flexibilité pour modéliser les relations complexes. Toutefois les inconvénients majeurs des réseaux de neurones résident dans leur lenteur, leur convergence vers des minima locaux, une architecture difficile à paramétrer et aussi la difficulté à interpréter leurs sorties.

Ce paragraphe clôt le tour d'horizon de l'apprentissage automatique supervisée pour une tâche de classification. La prochaine section est principalement dédiée aux problèmes de prétraitement des données et à la détermination d'espaces de représentation efficaces qui constitue une étape importante et préliminaire à la classification.

2.3 Réduction de la dimensionnalité

La classification de données situées dans un espace de grande dimension est un problème délicat qui apparaît dans de nombreuses sciences telles que l'analyse d'images. La taille des données se mesure selon deux dimensions, le nombre de variables et le nombre d'exemples. Ces dernières peuvent prendre des valeurs très élevées et ainsi poser des difficultés lors de l'exploration et l'analyse des données. En effet, dans de nombreux cas, les variables mesurées ne sont pas toutes importantes pour la compréhension du phénomène sous-jacent. Pour cela il est fondamental de pré-traiter ces dernières à l'aide d'outils permettant une meilleure compréhension des connaissances. Un des concepts les plus utilisés, dans le domaine de la reconnaissance de formes, de l'apprentissage

2.3. Réduction de la dimensionnalité

machine et l'exploration des données, est la réduction de dimensionnalité. Le but de la réduction de dimensionnalité est, d'extraire un petit nombre de caractéristiques, de telle sorte que l'information intrinsèque contenue dans les données initiales soit préservée, afin de visualiser les données et d'accélérer leur traitement ultérieur, ou encore (ce qui nous intéresse le plus ici) de faciliter l'apprentissage à partir de celles-ci. Il existe, dans la littérature, deux approches de réduction de la dimensionnalité, à savoir, la sélection de caractéristiques et l'extraction de caractéristiques. La première catégorie est appropriée quand l'acquisition de mesures des formes est coûteuse. Ainsi son objectif principal est de réduire le nombre de mesures requises. Par contre, les techniques d'extraction de caractéristiques utilisent toute l'information contenue dans les formes pour la compresser et produire un vecteur de plus petite dimension.

2.3.1 La sélection de caractéristiques

La sélection de caractéristiques est une technique de recherche permettant de trouver un sous-ensemble optimal de caractéristiques parmi celles de l'ensemble de départ en utilisant des mesures objectives. La notion de pertinence d'un sous-ensemble de caractéristiques dépend des objectifs et des critères d'évaluation du système. Cette approche de sélection vise à réduire le nombre de caractéristiques en supprimant les données non pertinentes, redondantes et bruyantes. Les méthodes d'évaluation d'un sous-ensemble de caractéristiques dans les algorithmes de sélection se divisent en trois types, en fonction de la façon dont ils combinent la recherche des caractéristiques pertinentes avec la construction du modèle de classification : les méthodes de filtrage, les méthodes enveloppées et enfin les méthodes intégrées.

2.3.1.1 Les méthodes de filtrage

Les modèles par Filtrage estiment un indice de pertinence pour chaque caractéristique afin de mesurer la pertinence de cette dernière sur la cible. S'en suit un classement des caractéristiques selon leurs indices de pertinence et une recherche basée sur les rangs des indices ou encore une recherche basée sur un critère d'évaluation statistique. Les modèles par filtrage se distinguent par le fait que, l'indice de pertinence n'est calculé que par rapport à une seule caractéristique sans tenir compte des valeurs des autres caractéristiques. Cette mise en œuvre implique que les filtres supposent l'orthogonalité des caractéristiques ce qui n'est généralement pas le cas dans la pratique. Par conséquent, les filtres omettent toute dépendance (ou indépendance) conditionnelle qui pourrait exister, en résulte une sélection comportant de l'information redondante plutôt que complémentaire. De plus, les filtres ne prennent pas en considération la performance des méthodes de classification qui suivent la sélection [113]. Néanmoins, ces modèles

se révèlent être très efficaces et robustes face au problème de sur-apprentissage [78].

2.3.1.2 Les méthodes enveloppées

Afin que la performance du classifieur choisi soit prise en considération lors de la sélection des caractéristiques pertinentes, **Kohavi et al.** [113] introduisent le concept “enveloppé”. Dans cette configuration, une procédure de recherche dans l’espace des sous-ensembles de caractéristiques possibles est définie, et divers sous-ensembles de caractéristiques sont générés et évalués. L’évaluation d’un sous-ensemble spécifique de caractéristiques est obtenue par l’entraînement et le test d’un modèle de classification spécifique, ce qui rend cette approche adaptée à un algorithme de classification précis. Pour rechercher l’espace de tous les sous-ensembles de caractéristiques, un algorithme de recherche est alors “enveloppé” autour du modèle de classification. Cependant, comme l’espace des sous-ensembles croît de façon exponentielle avec le nombre de caractéristiques, des méthodes de recherche heuristique sont, alors, utilisées pour guider la recherche d’un sous-ensemble optimal. Si les modèles enveloppés sont généralement considérés comme étant meilleurs que les modèles de filtrage en termes de pertinence des caractéristiques choisies, notamment, grâce à l’exploration de l’information mutuelle entre les caractéristiques, ils restent, néanmoins, limités par trois inconvénients majeurs, la complexité et le temps de calcul nécessaire pour la sélection, la dépendance des caractéristiques optimales par rapport au classifieur utilisé. En effet ces dernières ne sont pas forcément valides pour un autre classifieur. Et enfin, ces techniques sont sujettes à de forts risques de sur-apprentissage.

2.3.1.3 Les méthodes intégrées

Le troisième type de techniques de sélection de caractéristiques, est dit méthode intégrée. Ces méthodes diffèrent des autres méthodes de sélection dans la manière dont la sélection de caractéristiques et l’apprentissage interagissent. En effet, dans ces dernières, les parties, apprentissage et sélection, ne peuvent pas être séparées. La recherche d’un sous-ensemble optimal de caractéristiques est intégrée dans la construction du classifieur, et peut être considérée comme une recherche, dans l’espace combiné des sous-ensembles des caractéristiques et des critères générés au cours du processus d’apprentissage. Tout comme les approches enveloppées, les approches intégrées sont spécifiques à un algorithme d’apprentissage donné, ce qui est leur principal inconvénient. Elles sont, cependant, plus rapides en temps de calcul car elles évitent que le classifieur recommence de zéro pour chaque sous-ensemble de caractéristiques [233].

Les trois types de méthodes de sélection de caractéristiques diffèrent dans leur métho-

2.3. Réduction de la dimensionnalité

dologie et chacun a sa propre force et faiblesse. En ce qui concerne l'efficacité de calcul, les filtres ne nécessitent aucun apprentissage de modèle et sont donc efficaces en terme de temps. Le type enveloppé est généralement le plus complexe et le plus lent parmi les trois, car la procédure de validation croisée sur chaque itération est très coûteuse. Le type intégré peut incorporer des processus d'accélération lors de l'évaluation des sous-ensembles de caractéristiques afin d'éviter les procédures de validation croisée et donc être moins coûteux que le type enveloppé .

2.3.2 L'extraction de caractéristiques

La réduction de la dimensionnalité par extraction de caractéristiques appelée aussi, technique de transformation de données, ne se fait pas par une sélection de certaines caractéristiques, mais par une construction de nouvelles caractéristiques obtenues en combinant les caractéristiques initiales. Dans la dernière décennie, un grand nombre de techniques de réduction de la dimensionnalité a été proposé. Les sections suivantes décrivent brièvement plusieurs approches de base.

2.3.2.1 Analyse en composantes principales

L'analyse en composantes principales PCA (*Principal Component Analysis* en anglais) est une technique statistique linéaire projective non supervisée de réduction de dimensionnalité introduite par **Pearson** [158]. Globalement la PCA consiste en une projection orthogonale des données originales sur un sous-espace linéaire de dimension réduite, engendré par les vecteurs propres sélectionnés, qui maximise la variance des données projetées tout en minimisant l'erreur de reconstruction quadratique moyenne entre un point de l'ensemble original et son projeté [101]. Cette méthode trouve un espace de représentation fidèle aux données lorsque la structure de ces dernières est linéaire, ce qui n'est généralement pas le cas. La formulation de la PCA en terme de produits scalaires [178] permet l'utilisation de l'astuce du noyau [8, 25] , qui établit que tout algorithme formulé avec une fonction noyau peut être reformulé avec une autre fonction noyau, et ainsi, l'application de méthodes linéaires de réduction de dimension lorsque la structure intrinsèque des données n'est pas linéaire. Cette procédure, appelée *Kernel Principal Component Analysis* KPCA [178], revient alors, à effectuer une analyse en composantes principales dans un espace de caractéristiques de haute dimension, construit à l'aide d'une fonction noyau, dans lequel le problème devient linéaire. La PCA et ses variantes ont été appliquées avec succès dans divers champs de reconnaissance de formes tels que la reconnaissance de visages [199, 71, 188, 217].

2.3.2.2 Auto-encodeurs multi-couches

Une autre variante de la PCA non linéaire, basée sur un perceptron multi-couches (MLP) avec une topologie auto-associative, est l'auto-encodeur multicouche [48]. Ces derniers sont des réseaux de neurones possédant un nombre, quelconque, impair de couches cachées, tant que l'on identifie l'une d'entre elles comme contenant la représentation en basse dimension. Le réseau est entraîné à minimiser l'erreur quadratique moyenne entre l'entrée et la sortie du réseau. La partie du réseau menant de l'ensemble d'origine à sa représentation est alors celle qui effectue l'encodage (la réduction de dimensionnalité), et celle menant de la représentation à l'ensemble de données reconstruites effectue le décodage (et n'est utilisée que durant l'entraînement). Toutefois si, lors de l'étape de décodage, les fonctions d'activation sont linéaires, l'auto-encodeur revient à transformer les données de manière similaire à la PCA. D'où l'utilisation, généralement, de fonctions d'activation sigmoïdes. Les auto-encodeurs multicouches possèdent généralement un nombre élevé de connexions. Par conséquent, les approches de rétro-propagation classiques [171] convergent lentement et sont susceptibles de se bloquer dans des minima locaux. Dans [84], cet inconvénient est surmonté par une procédure d'apprentissage qui consiste en trois étapes principales. Tout d'abord, les couches de l'étape d'encodage sont entraînées une par une à l'aide de machines de Boltzmann restreintes RBM. Les RBMs peuvent être entraînés de manière efficace en utilisant une procédure d'apprentissage non supervisée proposée par **Hinton et al.** [83], l'algorithme de la divergence contrastive. D'autre part, les couches de reconstruction du réseau sont formées par l'inverse des couches entraînées de l'encodage. Enfin, l'auto-encodeur est affiné de manière supervisée à l'aide de la technique de rétro-propagation.

2.3.2.3 Positionnement multidimensionnel

Le positionnement multidimensionnel MDS (*Multi-dimensional scaling* en anglais) [196, 143, 43] représente un ensemble de techniques consistant à trouver une représentation dans un espace à faible dimension tout en conservant, au mieux, les distances entre les paires de points de l'espace initial. Ceci revient à optimiser un critère exprimé en terme de fonctions contraintes, une mesure d'erreur entre les distances, généralement Euclidiennes, dans l'espace d'origine et les distances dans l'espace de représentation.

Étant donnée $X = \{x_i\}_{i=1}^N$ l'ensemble des points de l'espace original de dimension M et $Y = \{y_i\}_{i=1}^N$ l'ensemble des points de l'espace réduit de dimension K avec ($K < M$). Les trois principales fonctions contraintes utilisées dans la MDS sont, la fonction contrainte brute ou métrique [196], la fonction contrainte non-métrique [116] et la fonction de coût de Sammon [174].

2.3. Réduction de la dimensionnalité

La première est définie par :

$$\phi(Y) = \sum_{i \neq j} [d(x_i, x_j) - d(y_i, y_j)]^2 \quad (2.26)$$

où $d(x_i, x_j)$ représente la distance entre les points en haute dimension x_i et x_j , et $d(y_i, y_j)$ représente la distance entre les points en faible dimension y_i et y_j .

La MDS métrique utilise les valeurs réelles de l'information de (dis)similarité existant entre les points. Une projection des données basée uniquement sur les valeurs des distances Euclidiennes n'est pas toujours la meilleure représentation des données. Particulièrement si les (dis)similarités sont ordonnées selon un certain ordre (par exemple croissant). Dans ce cas-ci, cet ordre est plus significatif que les valeurs réelles des distances. Dès lors la projection doit maintenir le mieux possible le rang des (dis)similarités dans l'espace réduit. Pour ce faire, **Kruskal** [116] introduit la fonction contrainte non-métrique définie par :

$$\phi(Y) = \sqrt{\frac{\sum_{i \neq j} [d(x_i, x_j) - d(y_i, y_j)]^2}{\sum_{i \neq j} [d(x_i, x_j)]^2}} \quad (2.27)$$

Cette fonction permet de trouver une configuration des points dans un espace de dimension K de telle sorte que les distances entre les points obtenus conservent au mieux la relation monotone de l'information de (dis)similarité correspondante.

La fonction de coût de Sammon est, quant à elle, définie par :

$$\phi(Y) = \frac{1}{\sum_{ij} d(x_i, x_j)} \sum_{i \neq j} \frac{[d(x_i, x_j) - d(y_i, y_j)]^2}{d(x_i, x_j)} \quad (2.28)$$

Bien que son principe est similaire à celui de la fonction contrainte métrique, cette dernière normalise l'erreur de préservation des distances à l'aide des distances calculées dans l'espace d'origine afin de favoriser la préservation des petites distances dans l'espace réduit.

Ces problèmes de minimisation peuvent être résolus par divers procédés tels que la décomposition en valeurs propres, la méthode du gradient conjugué ou encore la méthode de Pseudo-Newton [43].

La popularité de cette technique a conduit au développement de diverses variantes telles que SPE [7], SNE [82] et FastMap [56]. En outre, il existe des variantes non métriques de la MDS, qui visent à préserver les relations ordinales dans les données, au lieu des paires de distances [43].

2.3.2.4 Analyse en Composantes Indépendantes

L'analyse en composantes indépendantes ICA (*Independent Component Analysis* en anglais) a été introduite par **Jutten et al.** [103] dans le contexte de la neurophysiologie. Elle devient populaire lors de son utilisation dans le domaine du traitement du signal pour la séparation de sources aveugles. L'ICA considère les données comme étant générées par un mélange de variables latentes inconnues et bien que, généralement, le nombre de variables latentes est présumé égal à la dimension des données, la méthode a des parallèles avec la réduction de la dimensionnalité. L'ICA recherche des projections de telle sorte que les distributions de probabilité des données le long de ces projections soient statistiquement indépendantes. Contrairement à la PCA, qui ne considère que la matrice de covariance, l'algorithme ICA est capable d'employer des statistiques d'ordre supérieur pouvant contenir des données complémentaires importantes et ainsi être plus performant que la PCA. Il existe une multitude de variantes de l'ICA, elles diffèrent principalement par le type de transformation appris (linéaire ou non linéaire), ainsi que par le choix du critère maximisant l'indépendance des caractéristiques extraites [90].

2.3.2.5 Méthodes de réduction de dimensionnalité locales

De nombreuses techniques de réduction de dimensionnalité, dites locales, ont été introduites ces dernières années. Ces algorithmes se basent sur la notion de voisinage entre les points d'entraînement, généralement calculée à l'aide d'une distance Euclidienne. Le concept de ces dernières est souvent de trouver une représentation des données dans l'espace à faible dimension compatible avec certaines propriétés extraites à partir du voisinage de chaque point de l'espace d'origine :

- *Locally Linear Embedding* LLE [169] préserve la reconstruction de chaque point par une combinaison linéaire de ses voisins ;
- *Laplacian Eigenmap* [15] préserve le voisinage entre les points de l'espace original. En effet cette technique cherche une représentation de telle sorte que deux points voisins dans l'espace original le soient aussi dans l'espace réduit ;
- *Local Tangent Space Alignment* LTSA [232], l'idée de base de cette approche consiste à utiliser l'espace tangent dans le voisinage d'un point de données pour représenter la géométrie locale, puis aligner les espaces tangents locaux afin de construire le système de coordonnées globales en basse dimension en minimisant l'erreur d'alignement de l'apprentissage de coordination globale. Ce problème de minimisation est équivalent à un problème de valeurs propres qui peut être résolu efficacement ;
- *Stochastic Neighbor Embedding* SNE [82] est une approche probabiliste qui fait correspondre les données de grande dimension dans un sous-espace de dimension

2.3. Réduction de la dimensionnalité

réduite de manière à préserver les distances relatives à des voisins proches. Dans SNE, les objets similaires dans l'espace d'origine seront rapprochés dans l'espace réduit, et vice-versa, les objets dissemblables dans l'espace d'origine seront généralement éloignés dans l'espace réduit. Une distribution gaussienne centrée sur un point dans l'espace de grande dimension est utilisée pour définir la distribution de probabilité que ce dernier choisisse d'autres points de données que ses voisins. SNE est plus efficace à maintenir les distances relatives entre chaque deux points de données que la LLE ;

- *Isometric Mapping* Isomap [192] est une technique à la fois locale (car se basant sur les plus proches voisins) et globale (car essayant de conserver toutes les distances, comme dans la MDS). Elle tente de préserver les paires de distances géodésiques entre les points de données. L'approximation de la distance géodésique est divisée en deux cas. Pour les points voisins, la distance Euclidienne dans l'espace initial fournit une bonne approximation de la distance géodésique et des points éloignés. La distance géodésique peut être, aussi, approximée par le plus court chemin sur le graphe reliant entre eux les points voisins. L'Isomap bénéficie des mêmes avantages que la PCA, et la MDS, tels que l'efficacité de calcul et la garantie de la convergence asymptotique, mais avec une plus grande flexibilité afin d'apprendre une large classe de variétés non linéaires ;
- *Maximum Variance Unfolding* MVU [211] cherche une représentation maximisant la variance tout en préservant les distances entre points voisins. Il diffère de l'Isomap par l'optimisation des distances euclidiennes entre les points de données, mais de façon à ce que les distances dans le graphe de voisinage restent inchangées. Le problème d'optimisation résultant peut être résolu efficacement en utilisant la programmation semi-définie.

2.3.2.6 Méthodes supervisées et semi-supervisées

Les méthodes précédentes de réduction de la dimensionnalité sont dites des méthodes non-supervisées et sont applicables à divers problèmes non-supervisés, mais aussi aux problèmes semi-supervisés ou encore supervisés. Cependant lorsque la représentation en basse dimension est destinée à une tâche supervisée, cette dernière n'est pas optimale puisqu'elle ne tient pas compte de l'étiquette des exemples lors de l'apprentissage. D'où le développement de méthodes plus adaptées à ce type de tâches, les techniques (semi-)supervisées.

Parmi ces méthodes, une des plus populaires est l'analyse discriminante linéaire LDA (*Linear discriminant analysis* en anglais) [64, 141] La LDA, destinée à une tâche de classification binaire, cherche une projection linéaire unidimensionnelle séparant au mieux deux classes. Cette dernière a été adaptée par **Bishop** [18] afin de considérer les problèmes multi-classes.

Sur le même principe de projection, l'analyse en composantes canoniques CCA (*Canonical Component Analysis* en anglais) [85] projette linéairement le point de l'espace d'origine de telle manière à maximiser la corrélation avec une projection linéaire de l'étiquette.

Tout comme avec la KPCA, l'astuce du noyau permet la transformation de la LDA et la CCA en algorithmes d'extraction de caractéristiques non linéaires.

Les versions semi-supervisées de réduction de la dimensionnalité, quant à elles, se basent essentiellement sur la combinaison de deux critères, l'un non-supervisé et l'autre supervisé afin d'obtenir une meilleure séparabilité des classes que celle obtenue uniquement par un critère non-supervisé, tout en bénéficiant des exemples non étiquetés lors de l'optimisation de ce dernier [225, 231, 32].

Dans cette section, nous avons traité du domaine de la réduction de dimensionnalité. Dans un premier temps, un aperçu des techniques de sélection de caractéristiques a été présenté. Suivi des techniques de réduction par une transformation de données, cependant, par souci de légèreté, l'aperçu ne couvre que les approches de base et non leurs variantes ou extensions.

2.4

Détection de mouvements

Un mouvement en vision artificielle représente un changement entre différentes images consécutives dans une séquence vidéo. Ce changement est dû soit au déplacement d'un objet dans un plan multidimensionnel ou alors au déplacement du capteur autour de l'objet en question. Détecter un mouvement revient alors à détecter un comportement différent d'une zone de l'image par rapport au comportement principal observé puis à le segmenter afin d'extraire l'objet en mouvement appelé «avant-plan» de l'information statique appelée «arrière-plan».

La méthode la plus intuitive de détection de zones en mouvements est la dérivée temporelle en tout point. Elle consiste à mesurer le changement d'apparence des pixels entre deux images consécutives, soit la différence inter-images [96, 130] ou trois images consécutives, soit la double-différence inter-images [105, 44]. Ainsi elle ne nécessite aucune information préalable concernant l'arrière-plan de la scène.

Autres exemples de méthodes sans modélisation de l'arrière-plan, sont les techniques basées sur le flux optique. Tandis que la dérivée temporelle quantifie la variation de l'aspect de chaque pixel considéré individuellement, le flux optique est un champ de vecteurs à deux dimensions représentant la projection sur le plan image du mouvement réel observé (tridimensionnel). Ce dernier est utilisé afin de segmenter l'image en régions de mouvements homogènes, ainsi, différencier les objets en mouvements des

2.4. Détection de mouvements

objets statiques [13].

Nombreux systèmes de vision par ordinateur, tels que, la vidéo surveillance [37, 195, 182], la capture de mouvements [54], ou encore les applications multimédias [34], ont pour objectifs communs, le suivi et/ou la reconnaissance des objets en mouvement de manière robuste, exigeant ainsi une première étape de détection de l'objet mobile, qui soit fiable et efficace particulièrement lorsque la forme de l'objet a une importance pour la suite des traitements subséquents, conditions généralement, non assurées par les techniques sans modélisation de l'arrière-plan de la scène. En effet, la dérivée temporelle échoue à extraire les pixels des régions mono-couleur et/ou, à l'intensité proche de celle de l'arrière-plan [194]. Quant au flux optique généré, en plus de la complexité calculatoire, il n'est pas toujours correct et ne permet pas de distinguer les objets ayant un mouvement proche de celui du fond. Un autre inconvénient majeur à ce dernier est qu'il n'est pas défini aux bords de l'objet mobile, provoquant ainsi, des erreurs de segmentations [13].

Ces inconvénients majeurs conduisent au développement et la popularité des méthodes basées sur la modélisation et la soustraction de l'arrière-plan. L'idée principale de telles techniques est de générer et maintenir automatiquement une représentation de l'arrière-plan de la scène, puis de trouver des déviations par rapport au modèle pour chaque trame entrante. Les pixels constituant les régions subissant des changements sont marqués pour un traitement ultérieur.

La façon la plus simple de modéliser le fond serait d'acquérir une image représentant la scène dépourvue d'objets mobiles. Cependant, dans certains environnements, l'obtention d'une telle image n'est pas chose facile, particulièrement en environnement extérieur ; Par ailleurs, les variations d'intensité lumineuse, l'introduction et/ou le retrait d'objets en ce dernier, rendent rapidement obsolète un tel modèle. Nécessitant ainsi, un modèle de fond robuste et adaptatif.

Dans cette partie du chapitre, seront présentées les différentes étapes d'une détection de mouvement par soustraction de fond, incluant une vue d'ensemble des différentes méthodes de modélisation de l'arrière-plan. Pour plus de détails sur ces dernières, le lecteur est invité à lire les études associées à chacune d'elles.

2.4.1 Étapes d'une opération de soustraction de fond

Le choix de la technique de modélisation de l'arrière-plan constitue le cœur d'une détection de mouvements par soustraction de fond. En effet, lors de celui-ci, sera déterminé, non seulement, le type de modèle générant l'image de référence mais aussi, l'échelle d'observation utilisée (pixel [189], bloc [58] ou encore *cluster* [17]) qui déterminera la robustesse aux bruits et la précision. Ainsi que la sélection du descripteur employé, qu'il soit spectral (couleur), spatial (contour, texture, stéréo),

temporel (mouvement) ou encore une combinaison de ces derniers. Les propriétés distinctes de ces descripteurs permettent la gestion de différentes situations telles que les changements d'illumination, de mouvement et de structure de fond [127].

Le schéma typique d'une opération de soustraction de fond se compose de trois processus principaux, un processus d'initialisation du modèle de l'arrière-plan suivi d'une répétition de deux processus, la détection de l'avant-plan et la maintenance de l'arrière-plan. Ces processus dépendent entièrement de la technique de modélisation du fond choisie.

2.4.1.1 Initialisation du modèle de l'arrière-plan

L'initialisation du modèle se fait, généralement, à partir d'un ensemble d'images d'entraînement extraites de la séquence vidéo. Le principal défi est d'obtenir un premier modèle de fond lorsque plus de la moitié des images de la séquence contiennent des objets d'avant-plan. Si les algorithmes d'initialisation dépendent, souvent du nombre de modes (uni-modal ou multi-modal) et de la complexité du modèle sélectionné [162], les trois algorithmes, usuellement utilisés, sont : l'algorithme par lot, l'algorithme par incrémentation, utilisant, tous deux, un nombre N connu d'images d'entraînement (consécutives ou non) [146] et l'algorithme progressif, consistant à générer un modèle de fond partiel puis à l'améliorer jusqu'à obtenir un modèle complet [40, 41].

2.4.1.2 Détection de l'avant-plan

Cette étape est une opération de classification étiquetant les pixels comme étant mobiles ou statiques. Si le modèle de l'arrière-plan est une image, une différence en valeur absolue entre ce modèle et l'image courante est effectuée afin d'obtenir une détection de mouvement. Quand il s'agit d'un modèle statistique, on calcule la probabilité que chaque pixel appartienne à l'arrière-plan en testant la valeur observée dans le modèle, l'importance du mouvement observé varie dans le sens opposé à la probabilité calculée.

2.4.1.3 Maintenance de l'arrière-plan

le mécanisme de maintenance de l'arrière-plan détermine la façon dont le modèle de fond s'adapte aux changements critiques susceptibles de survenir au cours du temps. Les points clés de ce mécanisme sont :

- Le système de mise à jour que l'on trouve, dans la littérature, sous trois formes : le système de mise à jour aveugle, le système de mise à jour sélectif et enfin le

2.4. Détection de mouvements

système de mise à jour adaptatif flou [10] ;

La mise à jour aveugle de l'arrière-plan réactualise tous les pixels avec une même règle qui consiste en un filtre à réponse impulsionnelle infinie RII :

$$B_{t+1}(x, y) = (1 - \alpha)B_t(x, y) + \alpha I_t(x, y) \quad (2.29)$$

où α est le taux d'apprentissage. B_t et I_t sont, respectivement, l'image référence de l'arrière-plan et l'image courante au temps t . Le principal inconvénient de ce système est que les valeurs des pixels étiquetés comme étant mobiles sont incluses dans le calcul de la nouvelle image de l'arrière-plan, conduisant ainsi à une représentation erronée du fond.

Afin de contrer ce problème, certains auteurs, utilisent une mise à jour sélective qui consiste à calculer la nouvelle image de fond à l'aide de différents taux d'apprentissage dépendant de la classification précédente des pixels :

$$\begin{cases} B_{t+1}(x, y) = (1 - \alpha)B_t(x, y) + \alpha I_t(x, y) & \text{si } (x, y) \in \text{fond} \\ B_{t+1}(x, y) = (1 - \beta\alpha)B_t(x, y) + \beta\alpha I_t(x, y) & \text{sinon} \end{cases} \quad (2.30)$$

avec $\beta \ll \alpha$ (généralement $\beta = 0$). Ici, l'idée est qu'à la différence du pixel étiqueté comme étant statique, le pixel étiqueté comme étant mobile sera mis à jour plus lentement. Néanmoins une mauvaise classification des pixels entraînera un modèle de fond erroné tout au long de la séquence. Ce problème peut être résolu par un système de mise à jour adaptatif flou qui tient compte de l'incertitude de la classification. Ceci peut être réalisé par la graduation de la règle de mise à jour en utilisant le résultat de l'étape détection de l'avant-plan comme dans [10].

- Le taux d'apprentissage détermine la vitesse d'adaptation aux changements d'éclairage et/ou aux objets dans la scène, mais aussi le temps nécessaire à la survie d'un objet de l'avant-plan en arrêt avant son incorporation dans le modèle de fond. Ainsi, le taux d'apprentissage dépend de plusieurs aspects dont les caractéristiques temporelles diffèrent. Pour différencier le mécanisme d'adaptation et le mécanisme d'incorporation, certains auteurs utilisent un ensemble de compteurs représentant le nombre de fois qu'un pixel est classé comme un pixel d'avant-plan. Lorsque ce nombre est supérieur à un seuil, le pixel est absorbé par le modèle de l'arrière-plan.

- La fréquence de mise à jour. L'objectif est de mettre à jour l'arrière-plan uniquement lorsque cela est nécessaire. La mise à jour peut être effectuée à chaque image, mais aussi uniquement lors de changements importants.

2.4.2 Techniques de modélisation du fond de référence

La recherche en détection de mouvement par soustraction de fond a fait l'objet d'une grande attention ces dernières années. Cette attention concerne plus particulièrement les méthodes de modélisation de l'arrière-plan. En résulte le développement d'une multitude de techniques permettant d'acquérir des modèles robustes traitant à la fois les séquences prises à partir de cameras statiques ou mobiles ainsi que les environnements statiques (uni-modal) ou dynamiques (multimodal).

Nous présentons, ici, un aperçu général des différentes approches existant. Ces modèles peuvent être catégorisés en cinq grandes familles suivant le modèle mathématique utilisé :

2.4.2.1 Modèles basiques

Dans ce cas, l'image de l'arrière-plan est générée à partir d'un nombre N d'images appartenant à la séquence à l'aide d'une moyenne temporelle [125], d'un filtre médian temporel [140] ou encore d'une analyse temporelle d'histogramme [235].

2.4.2.2 Modèles statistiques

À la différence des modèles basiques, les modèles statistiques offrent une meilleure robustesse face aux changements d'illumination et aux arrière-plans dynamiques.

1. Modèles Gaussiens

L'hypothèse sur laquelle sont basés les modèles de mélange Gaussien GMM (*Gaussian Mixture Model* en anglais) est, la possibilité de représenter l'historique des valeurs des pixels par une ou des distributions Gaussiennes. Suivant cette idée, **Wren et al.** [216] proposent un premier modèle utilisant une seule Gaussienne SG (*Single Gaussian* en anglais). **Kim et al.** [110] généralisent la SG en utilisant une Gaussienne générale SGG (*single general Gaussian* en anglais) afin d'atténuer les contraintes d'une Gaussienne stricte. Ce modèle permet d'obtenir de bons résultats pour des scènes d'intérieur où l'arrière-plan est parfaitement statique. Néanmoins, en environnement extérieur, des phénomènes périodiques tels que le mouvement des arbres peuvent le rendre totalement inopérant car la distribution de l'apparence de l'arrière-plan est alors multimodale. Afin de résoudre ce problème, **Stauffer et al.** [189] introduisent le concept de modélisation par mélange Gaussien GMM. Ce dernier a fait l'objet de beaucoup d'études conduisant à l'amélioration de la robustesse face aux situations critiques, telle que l'utilisation

2.4. Détection de mouvements

de mélange Gaussien général MOGG [9]. Cependant, lorsque des changements apparaissent trop rapidement dans le fond, les variances des Gaussiennes, le caractérisant, deviennent trop importantes et toutes les méthodes décrites précédemment échouent. Ainsi, quand la fonction de densité est plus complexe et ne peut être modélisée de manière paramétrique, une approche non-paramétrique capable de manipuler des densités arbitraires est plus adaptée. **Elgammal et al.** [55] proposent une approche dans laquelle des noyaux Gaussiens sont utilisés pour modéliser la densité en chaque pixel à tout instant, connaissant les instants précédents récents, KDE (*Kernel density estimation* en anglais). Ce concept a, lui aussi, connu différentes améliorations. La cohérence spatiale est introduite dans [185] en ajoutant un noyau spatial au noyau temporel de [55]. L'image est représentée comme un champ de Markov qui apporte une dépendance inter-pixels. Il s'agit alors de trouver le maximum a posteriori en minimisant une fonction d'énergie contenant un terme unaire basé sur les distributions de fond et d'objets et un terme binaire traduisant la cohérence spatiale.

2. Modèles à vecteurs de support

La seconde catégorie utilise des modèles statistiques plus sophistiqués basés sur les machines à vecteurs de support. Tout d'abord, **Lin et al.** [129] proposent d'initialiser l'arrière-plan en utilisant une machine à vecteurs de support probabiliste, utilisant comme caractéristiques les valeurs de flux optique et de la différence inter-images. De la même manière, **Wang et al.** [205] utilisent une approche basée sur la régression par les machines à vecteurs supports séparés SVR (*Support Vector regression* en anglais) pour modéliser chaque pixel d'arrière-plan en fonction de l'intensité. Enfin, **Tavakkoli et al.** [191] procèdent à la classification des pixels selon la méthode basée sur la description des données supports SVDD (*Support Vector Data Description* en anglais).

Contrairement aux techniques d'estimation de densité paramétriques et non paramétriques, le modèle d'arrière-plan ne repose pas sur la fonction de probabilité de l'arrière-plan ou de l'avant-plan, mais sur une description analytique de la frontière de décision entre le fond et les classes d'avant-plan. Ainsi, la précision du modèle n'est pas limitée à la précision des fonctions d'estimation de densité de probabilité.

3. Modèles basés sur l'apprentissage de sous-espaces

La troisième catégorie emploie les méthodes d'apprentissage de sous-espaces qui offrent à la fois une réduction de dimensionnalité et une fusion des caractéristiques. L'idée est de considérer les pixels comme des dimensions d'un espace de représentation, et les images successives comme des individus dans cet espace. Les méthodes d'analyse de données reconstructives permettent alors d'inspecter tous les pixels de l'image dans une approche globale pour définir de nouvelles caractéristiques que l'on pourra appliquer en tout point pour y détecter d'éventuels mouvements. **Oliver et al.** [156] proposent une première utilisation de la PCA pour la modélisation de l'arrière-plan de scènes vidéo. Cette dernière est

appliquée sur N images d'apprentissage prises à des instants non consécutifs afin de générer l'image moyenne et la matrice de projection comprenant les p premiers vecteurs propres significatifs de la PCA. De cette manière, la segmentation de l'avant-plan est réalisée en calculant la distance Euclidienne entre l'image d'entrée et l'image reconstruite à partir de sa projection. Toutefois, la version classique de ce modèle comprend différents inconvénients qui ont fait l'objet de diverses améliorations. Notamment, diminuer l'influence des objets mobiles afin de ne pas être absorbés lors de la génération du modèle de fond [107, 220]. Faire face aux exigences temporelles et de robustesse lors de la mise à jour du modèle à l'aide d'un algorithme de PCA incrémental avec une sélection pondérée adaptative des pixels de chaque image [28]. Étendre l'application de ce modèle aux données RGB et RGB+IR à l'aide d'un algorithme de PCA incrémental à deux dimensions [79]. Enfin, améliorer la gestion des changements d'éclairage soudains par un modèle de représentation multimodal. Pour cela **Dong et al.** [51] proposent l'apprentissage de multiples sous-espaces représentant différentes conditions d'éclairage à l'aide d'une PCA locales LPCA. Ainsi à chaque nouvelle image l'algorithme sélectionne le sous-espace partageant les mêmes caractéristiques d'éclairage. Plus récemment, d'autres variantes de modèle d'apprentissage de sous-ensembles ont été introduites afin d'améliorer les points cités précédemment, notamment, la ICA [224], La factorisation en matrices non-négatives INMF pour la réduction de la dimension [29]. L'utilisation d'un tenseur incrémental de rang (R_1, R_2, R_3) afin de considérer l'information spatiale [128]. Ou encore la projection à préservation locale LoPP qui est l'approximation linéaire du *Laplacian Eigenmap* [115].

En résumé, les modèles d'apprentissage de sous-espace utilisés dans la modélisation de fond surclassent les modèles Gaussiens et à vecteurs de support dans la gestion des changements d'illumination, qui sont, quant à eux, grandement optimisés pour les milieux multimodaux. De manière générale, Les modèles statistiques sont les modèles les plus utilisés en raison d'un bon compromis entre la performance et la complexité calculatoire.

2.4.2.3 Modèles basés sur les méthodes de partitionnement

Les modèles basés sur les méthodes de partitionnement supposent que chaque pixel de l'image peut être temporairement représenté par des *clusters*. Les approches de partitionnement sont constituées de l'algorithme des K-moyennes [31], *Codebook* [111] ou encore méthodes de *clustering* séquentielles [218].

1. Algorithme des K-moyennes

Butler et al. [31] proposent un algorithme qui attribue un groupe de *clusters* à chaque pixel dans l'image. L'initialisation du fond est réalisée hors ligne. Les *clusters* sont ordonnés selon leur vraisemblance à modéliser le fond. Chaque pixel

2.4. Détection de mouvements

entrant est associé au groupe de *clusters* correspondant, ce dernier déterminera l'appartenance du pixel à l'arrière-plan ou non. **Xiuman et al.** [219] améliorent la robustesse du modèle en insérant un algorithme génétique dans l'algorithme des K-moyennes.

2. Modèles basés sur les dictionnaires

Une autre approche optimisée pour les milieux multimodaux est la méthode dite de dictionnaire (*codebook* en anglais) par **Kim et al.** [111]. Sur la base d'une séquence d'apprentissage, le procédé associe à chaque pixel d'arrière-plan une série de valeurs de couleurs clés appelées mots de code (*codewords* en anglais) stockées dans un dictionnaire. Ces mots de code décrivent la couleur qu'un pixel est susceptible de prendre sur une certaine période de temps. La détection consiste à tester la différence entre l'image actuelle et le modèle de fond en terme de couleur et luminosité. Si un pixel d'entrée vérifie -1) la distorsion de couleurs de certains des mots de code est inférieur au seuil de détection, et -2) la luminosité se situe dans la plage de luminosité de ce mot de code, celui-ci est classé comme arrière-plan. Sinon, il est classé comme avant-plan. Afin d'accentuer la robustesse face aux changements d'éclairage, le modèle de représentation des couleurs clés en forme de cylindre et le modèle de représentation de la luminosité en forme de cône sont fusionnés afin d'obtenir un seul modèle hybride cylindre-cône [52], ou encore représenté en forme sphérique [86]. Enfin d'autres modifications ont été appliquées afin d'atteindre les exigences temps réel, telles que l'approche hiérarchique [77] [104] ou encore l'approche multi-échelles [228].

3. Modèles de *clustering* séquentielles

L'approche proposée par **Xiao et al.** [218] est basée sur l'hypothèse que le fond n'appartient pas aux régions apparaissant durant de courts laps de temps lors de la séquence. Premièrement, les intensités des pixels sont classées à l'aide d'un modèle de partitionnement en ligne puis, sont calculées les valeurs des centres de chaque partition ainsi que leurs probabilités d'apparition. Enfin une ou plusieurs partitions dont les probabilités d'apparition sont supérieures à un certain seuil représenteront le modèle de fond. Une amélioration proposée par les mêmes auteurs consiste à contrôler les déviations rapides des partitions à l'aide d'un second seuil et fusionner les partitions très proches. Pour résoudre le problème de déviation des partitions sans l'utilisation d'une procédure de marge ou plusieurs seuils. **Benalia et al.** [16] suggèrent un algorithme consistant en -1) la sauvegarde de la première valeur de la partition à sa création dans un autre centre de partition puis, -2) en la comparaison de la valeur actuelle de la partition à sa valeur précédente après chaque mise à jour pour le contrôle de la déviation. Si cette dernière est importante une nouvelle partition sera créée à partir de l'ancienne et les valeurs de poids de l'actuelle. Afin d'optimiser l'utilisation de la mémoire, les partitions ne subissant aucun changement sont supprimées selon l'hypothèse que les partitions représentant le fond sont mises à jour fréquemment.

2.4.2.4 Modèles basés sur les réseaux de neurones

Dans ce cas, les modèles de fond sont représentés au moyen des poids d'un réseau de neurones appris sur N images ne contenant pas d'objets mobiles. Les principales approches sont :

1. Réseau de neurones de régression généralisé

Culibrk et al. [45, 46] proposent l'utilisation d'une architecture en forme de réseau de neurones pour former un classifieur bayésien non supervisé pour la modélisation de l'arrière-plan et la détection de l'avant-plan. Les poids permettent la modélisation du fond et leur mise à jour reflète les statistiques de changements de l'arrière-plan. Ce modèle de classifieurs est particulièrement optimisé pour la segmentation de séquences en environnement extérieur comprenant des mouvements répétitifs et des changements de luminosité.

2. Réseau de neurones multivalué

Luque et al. [135] suggèrent une méthode de détection basée sur l'utilisation d'un réseau de neurones discret multivalué. Ce dernier vient combler les lacunes de l'algorithme GMM. Parmi les avantages de celui-ci, nous citerons le parallélisme du calcul de la solution ainsi que la capacité à représenter les classes en format qualitatif, arrière-plan, avant-plan et ombre.

3. Réseau de neurones compétitif

Dans d'autres travaux, **Luque et al.** [134] proposent l'utilisation d'un réseau de neurones compétitif basé sur un voisinage adaptatif pour modéliser le fond. Ils améliorent ce dernier en optant pour un réseau de neurones compétitif dipolaire qui permet de classer les pixels comme étant statiques ou mobiles. La représentation dipolaire est conçue pour traiter le problème de classification à un faible coût de calculs.

4. Réseau de neurones auto-organisateur

Maddalena et al. [136] adoptent, quant à eux, un réseau de neurones auto-organisateur pour l'apprentissage des séquences dans l'espace de couleurs HSV. Cet algorithme nommé *Self-Organizing Background Subtraction* SOBS détecte les objets mobiles en utilisant une méthode de carte auto-organisatrice SOM (*Self-Organizing Map*, en anglais [114]), représentant les motifs mobiles et statiques, afin de rendre la structure du réseau de neurones beaucoup plus simple et l'apprentissage plus efficace. Ils introduisent, dans un souci d'amélioration, la notion de cohérence spatiale lors de la maintenance de l'arrière-plan pour une meilleure détection [137]. cette architecture présente, cependant, certaines limites dues en premier : à sa structure fixe en terme de nombre et d'arrangement des neurones, Celle-ci doit être définie à l'avance, et en second : à l'absence de représentation hiérarchique entre les entrées. Pour cela, **Palomo et al.** [157] suggèrent un réseau de neurones hiérarchique croissant. Ce dernier a une structure

2.4. Détection de mouvements

hiérarchique divisée en couches. Chaque couche est composée de différents réseaux de neurones auto-organiseurs simples avec des structures adaptatives qui sont déterminées au cours de l'apprentissage non supervisé selon les données en entrée. En résulte de meilleures détections d'objets mobiles et ce même lors de changements d'éclairage importants.

2.4.2.5 Modèles d'estimation

Dans ces modèles l'arrière-plan est estimé à l'aide d'un filtre. Chaque pixel de l'image courante déviant significativement de la valeur prédite est considéré comme appartenant à l'avant-plan.

1. Filtre de Wiener :

Toyama et al. [198] proposent une méthode à trois niveaux sémantiques (local, semi-local et global) l'algorithme Wallflower. La segmentation au niveau local est effectuée à l'aide d'un filtrage prédictif de Wiener. Ce dernier permet de construire une valeur estimée de la valeur que l'on devrait observer à un instant t , à partir d'un échantillon de N mesures bruitées. Les auteurs utilisent un échantillon de cinquante valeurs pour calculer trente coefficients de prédiction. Le principal avantage du filtre de Wiener est qu'il réduit l'incertitude sur la valeur d'un pixel en tenant compte de la façon dont celui ci varie dans le temps. Néanmoins, des erreurs se produisent lors de la corruption de l'historique par un objet mobile. Pour cela les auteurs maintiennent non seulement l'historique des valeurs prédites pour chaque pixel mais aussi l'historique des valeurs réelles et de là, pour chaque nouveau pixel, ils calculent deux prédictions l'une basée sur l'historique réel et l'autre basée sur l'historique prédit. Si l'une des deux valeurs appartient à l'intervalle de tolérance, le pixel est dit d'arrière-plan.

2. Filtre de Kalman :

Usuellement, le filtrage prédictif est réalisé à l'aide d'un filtre de Kalman. La méthode suppose que la meilleure information que l'on puisse avoir sur l'état d'un système est obtenue par le calcul d'une estimation qui fait explicitement mention du bruit enregistré lors de l'observation. De nombreuses variantes ont été proposées pour la modélisation du fond, elles se distinguent essentiellement par le vecteur d'état utilisé dans la description du système. Le schéma le plus populaire est celui de **Karmann et al.** [106]. Dans leur algorithme, l'état du système correspond à l'image de fond à l'instant t et les mesures à l'image entrante à l'instant t . Ainsi la méthode suppose que l'évolution des intensités des pixels de l'arrière-plan peut être décrite par un système dynamique de dimension finie. D'autres méthodes utilisent comme caractéristique la texture [222] ou encore une décision basée sur une région locale plutôt que sur un pixel [67]. Quant à **Wang et al.** [209], ils proposent l'utilisation d'une extension du filtre de Kalman pour les systèmes non linéaires afin de répondre, de façon plus robuste, aux

milieux dynamiques. Néanmoins, la masse de calcul à réaliser de ces approches est difficilement compatible avec des contraintes de temps réel, inconvénient résolu par **Fan et al.** [57] à l'aide d'un filtre de Kalman auto-adaptatif.

3. Filtre de Chebyshev :

Le choix de **Chang et al.** [36] s'est porté sur le filtre de Chebyshev pour la modélisation de l'arrière-plan. L'idée consiste à mettre à jour le fond progressivement jusqu'à atteindre une estimation correcte de celui-ci et ce au bout de mille deux cent cinquante images (environ quarante secondes de temps d'initialisation). Les changements soudains peuvent être détectés et intégrés si l'écart entre le fond estimé et l'image courante persiste durant plusieurs images. Le principal avantage de cette approche réside dans le fait qu'elle soit relativement peu coûteuse en terme de calculs.

Nous avons présenté les méthodes dites traditionnelles, celles-ci font référence aux premiers modèles utilisés dans le domaine. Elle se caractérisent par une mise en œuvre relativement facile et une capacité de traitement en temps réel, ce qui contribue à leur popularité. Cependant elles sont souvent optimisées pour des contraintes spécifiques et les améliorations tendant à généraliser les environnements couverts semblent atteindre leurs limites. Ces dernières années sont apparues les méthodes dites récentes, elles sont plus sophistiquées permettant ainsi une adaptation plus facile à différents environnements. Néanmoins leurs exigences calculatoires complexes ne favorisent pas des applications en temps réel et donc leur vulgarisation. Parmi ces méthodes nous citerons comme exemples, l'utilisation de distributions de Student au lieu de distributions Gaussiennes [147] ou encore un modèle de **Dirichlet** [81] dans les modèles basés sur les densités mélanges. L'algorithme *Visual Background Extractor* ViBe [12] dans les modèles non-paramétriques. D'autres proposent la combinaison des modèles de mélange de Gaussiennes GMM ainsi que les modèles non-paramétriques KDE afin d'obtenir des modèles hybrides et ainsi d'approcher la distribution de couleur du fond de référence [49]. Dans la continuité des modèles basés sur l'apprentissage de sous-espace, certains auteurs introduisent les méthodes discriminatives [59] ou encore une combinaison des deux types de méthodes, reconstructives à l'aide d'une PCA et discriminatives à l'aide de la LDA afin de générer un modèle de fond plus robuste [138].

Une autre approche consiste à séparer le fond et l'avant-plan dans des domaines différents. Pour cela, différentes méthodes de transformation peuvent être utilisées comme la Transformée de Fourier rapide FFT [215], la Transformée en Cosinus Discrète [161], la Transformée de Walsh [193], la Transformée en ondelettes [68] ou encore la Transformée de Hadamard [11].

Il existe une multitude d'autres méthodes de modélisation de l'arrière-plan dites récentes dont, certaines, ont été expérimentées dans des cas réels. Ces approches génèrent des modèles de fond très robustes et leur optimisation ne concerne plus

2.4. Détection de mouvements

uniquement le changement d'illumination ou les scènes dynamiques mais les deux à la fois, en résulte une utilisation plus générale de ces derniers et non plus selon le thème de l'étude faite. Cependant comme mentionné plus haut leur utilisation est souvent restreinte aux applications non temps réel.

3

IMPLEMENTATION ET RÉALISATION

Sommaire

3.1	Introduction	56
3.2	<i>Multi-dimensional scaling</i> MDS	56
3.2.1	<i>FastMap</i>	57
3.3	Vue globale du système construit	61
3.3.1	Soustraction de fond et extraction des silhouettes	61
3.3.2	Opérations de prétraitement des images	63
3.3.3	Modélisation des actions par MDS	66
3.3.4	Classification et reconnaissance des actions	69
3.4	Conclusion	70

3.1 Introduction

L'analyse et la reconnaissance du mouvement humain sur la base de vidéos acquises à partir d'une ou plusieurs caméras, impliquent l'extraction d'informations visuelles pertinentes, la représentation de ces informations sous une forme adéquate et enfin, l'interprétation de celles-ci.

Le mouvement humain est caractérisé par une séquence de postures spécifiques du corps. Le principe est, qu'à partir de ces postures clefs, une identification puisse se faire de manière automatique afin de reconnaître différentes actions humaines (marcher, courir, sauter,..., etc.).

Nous allons donc en un premier temps mettre en œuvre une série de traitements qui seront appliqués à la vidéo en entrée afin d'optimiser les tâches subséquentes. Puis dans un second temps extraire des modèles représentant les mouvements à partir des séquences d'images afin de les soumettre au processus de reconnaissance et ainsi obtenir le mouvement décrit par celles-ci.

3.2 *Multi-dimensional scaling* MDS

Le positionnement multidimensionnel permet la découverte de la structure spatiale sous-jacente dans un ensemble de données à partir de l'information de (dis)similarité existant entre ces dernières. Il existe différentes variantes. L'approche suivie dans ces travaux est celle permettant l'extraction de caractéristiques à partir des données brutes en entrée et non pas à partir de l'information de distance entre celles-ci. Elle se résume comme tel :

Entrée - N vecteurs de n -dimensions.

Sortie - N vecteurs de k -dimensions ($k \ll n$), de telle sorte que les distances entre les vecteurs soient maintenues aussi bien que possible.

L'algorithme relie chaque objet (vecteur) à un point dans l'espace de dimension k , de sorte à minimiser la fonction contrainte non-métrique [116] :

$$stress = \sqrt{\frac{\sum_{i \neq j} (\hat{d}_{ij} - d_{ij})^2}{\sum_{i \neq j} d_{ij}^2}} \quad (3.1)$$

3.2. Multi-dimensional scaling MDS

où d_{ij} représente la mesure de dissimilarité entre deux objets O_i et O_j dans l'espace original et \hat{d}_{ij} est la mesure de dissimilarité entre les images de projection P_i et P_j respectivement de O_i et O_j . La fonction contrainte reflète ainsi l'erreur relative en moyenne entre les distances originales et celles dans l'espace à k -dimensions.

Pour atteindre son objectif, la MDS commence par une hypothèse puis l'améliore de manière itérative, jusqu'à convergence. Dans sa version la plus simple, l'algorithme fonctionne à peu près comme suit : Il attribue chaque élément à un point de l'espace à k -dimensions (en utilisant, par exemple, une heuristique, ou de manière aléatoire). Puis, il examine chaque point en calculant les distances par rapport aux $(N - 1)$ autres points afin de déplacer celui-ci de sorte à réduire au minimum l'écart entre les distances réelles (dans l'espace original) et les distances estimées dans l'espace à k -dimensions. Techniquement, la MDS emploie la méthode de "la plus forte pente" pour mettre à jour les positions des points dans l'espace à k -dimensions. Intuitivement, l'algorithme traite chaque paire de distance comme un "ressort" entre deux points. Dès lors, il essaie de réarranger les positions des points dans l'espace à k -dimensions de sorte à minimiser la "raideur" de celui-ci.

Cependant, pour certaines applications, la MDS dans sa forme basique souffre d'un inconvénient majeur. En effet celle-ci nécessite un temps de calcul équivalent à $O(N^2)$, où N est le nombre d'éléments. Ainsi, elle est peu pratique pour de grands ensembles de données, tel notre cas. Pour cela, nous choisissons l'utilisation de l'algorithme **FastMap** proposé par **Faloutsos et Lin** [56], une variante de la MDS qui détermine une coordonnée à la fois en examinant un nombre constant de rangées de la matrice de distances.

3.2.1 *FastMap*

L'idée principale de cet algorithme est de considérer les objets comme étant des points dans un espace inconnu de n dimensions (avec $n \gg k$), puis d'essayer de projeter ces points sur k axes mutuellement orthogonaux. Le défi consiste à calculer ces projections à partir de la matrice des distances, calculées préalablement, dans l'espace original uniquement.

Dans la suite, dans un souci de clarté, un objet sera traité comme étant un point dans un espace n dimensions, (avec n inconnue).

Le cœur de la méthode proposée est de projeter des objets sur une "droite" soigneusement sélectionnée. Pour ce faire, l'algorithme choisit deux objets O_a et O_b dits objets pivots, puis envisage la "droite" traversant ces deux pivots dans l'espace à n dimensions qui représentera l'axe de projection. Le choix des deux pivots O_a et O_b est contraint par le fait de trouver une droite sur laquelle les projections soient les plus éloignées possibles les unes des autres, ce qui implique de choisir les pivots de sorte

3. IMPLEMENTATION ET RÉALISATION

que leur distance soit maximisée.

Ce processus requière, toutefois, le calcul de $O(N^2)$ distances. Pour remédier à cet inconvénient, les auteurs proposent l'algorithme heuristique linéaire suivant afin de maintenir une complexité linéaire $O(N)$:

Algorithme 1: Heuristique pour choisir deux objets éloignés

Entrées: ensemble des objets O

Sorties: O_a, O_b

Algorithme *Choix-Pivots* ($O, dist()$)

début

```
-Choisir arbitrairement un objet, et le déclarer comme le second objet pivot  $O_b$ ;  
- $O_a \leftarrow$  Objet qui est le plus éloigné de  $O_b$  selon la fonction distance  $dist()$   
  utilisée;  
- $O_b \leftarrow$  Objet qui est le plus éloigné de  $O_a$  selon la fonction distance  $dist()$   
  utilisée;  
/* Les étapes 2 et 3 peuvent être répétées un nombre constant de fois, tout  
  en maintenant la linéarité de l'heuristique */  
-Déclarer  $O_a$  et  $O_b$  comme la paire d'objets souhaitée.
```

fin

Les projections des objets de l'ensemble de données sur cet axe de projection sont calculées à l'aide de la loi des cosinus comme tel :

— Dans le cas où $k = 1$:

Théorème 1 (loi des cosinus) : étant donné un triangle $O_a O_i O_b$, la loi des cosinus énonce :

$$d_{b,i}^2 = d_{a,i}^2 + d_{a,b}^2 - 2x_i d_{a,b} \quad (3.2)$$

où d_{ij} représente la distance $D(O_i, O_j)$ pour $i, j = 1, \dots, N$.

Preuve : à partir du théorème de Pythagore, la solution de x_i , première coordonnée de l'objet O_i , dans l'équation (3.2) est donnée par :

$$x_i = \frac{d_{a,i}^2 + d_{a,b}^2 - d_{b,i}^2}{2d_{a,b}} \quad (3.3)$$

Grâce à l'équation (3.3), il est possible de projeter les objets en points sur une droite tout en conservant une partie de l'information de distance : Par exemple, si O_i est proche du pivot O_a , x_i sera petit (**Figure 3.1**).

— Dans le cas où $k \geq 2$:

La généralisation à des espaces de projection de dimension $k \geq 1$ se fait en considérant, un hyper-plan H de dimensions $(n - 1)$, perpendiculaire à la droite

3. IMPLEMENTATION ET RÉALISATION

La complexité de l'algorithme *FastMap* est d'environ $O(Nk)$:

Algorithme 2: *FastMap*

```
début
  Variables globales:
  X[ ] une matrice de taille  $N \times k$ ;
  /* À la fin de l'algorithme, la  $i$ -ème ligne correspondra à l'image du  $i$ -ème
     objet. */
  Entier Col# = 0;
  /* Pointe vers la colonne de la matrice X[] en cours de mise à jour. */
  PA[ ] une matrice des pivots de taille  $2 \times k$ ;
  /* Stocke les idts des objets pivots - une paire par appel récursif. */
  Entrées: ensemble des objets  $O$ ,  $k$ 
  Algorithme FastMap ( $k$ ,  $D()$ ,  $O$ )
  début
    si ( $k \leq 0$ ) alors
      | retourne;
    sinon
      | Col# ++;
    fin
    /* Choisir les pivots  $O_a$  et  $O_b$ . */
     $O_a, O_b \leftarrow \text{Choix-Pivots}(O, D());$ 
    /* Sauvegarder les identifiants des objets pivots */
    PA[1,Col#] =  $a$ ;
    PA[2,Col#] =  $b$ ;
    si  $D(O_a, O_b) = 0$  alors
      | pour  $i \leq N$  faire X[ $i$ ,Col#] = 0;
    fin
    /* Car toutes les distances inter-objets sont égales à zéro */
    pour  $i \leq N$  faire
      | Calculer  $x_i$ , coordonnée de l'objet  $O_i$  projeté sur la droite  $(O_a O_b)$  selon
      | l'équation (3.3); X[ $i$ ,Col#] =  $x_i$ ;
    fin
    /* considérer les projections des objets sur une hyper-plan  $H$ 
       perpendiculaire à la droite  $(O_a O_b)$ ; la fonction distance  $D'()$  entre
       deux projections selon l'équation (3.4). */
    Appeler FastMap ( $k - 1$ ,  $D'()$ ,  $O$ );
  fin
fin
```

3.3 Vue globale du système construit

Le procédé de reconnaissance mis en place dans notre étude est constitué de trois principales étapes séquentielles. La phase une consiste en le traitement des vidéos afin d'en extraire les informations utiles à la suite du processus. La phase deux consiste en la modélisation de ces informations sous une forme pertinente. Enfin, la phase trois consiste en la classification des modèles et la reconnaissance du mouvement représenté dans la vidéo.

3.3.1 Soustraction de fond et extraction des silhouettes

L'objectif de cette étape est d'identifier la silhouette de la personne en mouvement en séparant celle-ci de l'arrière plan de la scène. Cette étape joue un rôle très important dans l'évaluation de notre système car le résultat de celle-ci va influencer sur toutes les étapes suivantes.

Dans cette première partie, nous présentons la méthode de soustraction d'arrière plan utilisée. Cette dernière comporte trois étapes importantes : initialisation, extraction de l'objet et la mise à jour du modèle de fond. En résulte une binarisation des images produites par l'étape d'échantillonnage. Les traitements suivants étant en noir et blanc, ce choix s'est avéré être le plus approprié.

1. Initialisation :

La première étape consiste à modéliser l'arrière-plan en utilisant N images ($N \approx 30$) extraites à partir d'une séquence vidéo représentant la scène dépourvue de tout objet mobile quand celle-ci est fournie ou bien à partir de la séquence du mouvement elle même.

Une moyenne d'intensité est donc calculée à partir de ces images pour chaque pixel et pour chacun des canaux RGB. Ces moyennes seront la valeur des pixels de l'image de l'arrière plan (**Figure 3.3b**).

S'en suit le calcul des écarts-types pour chaque pixel pour les trois canaux RGB. Ceux ci représenteront les seuils de détection lors de la phase d'extraction du mouvement. Cette opération nécessite habituellement le stockage des N premières images. Or, une équation modifiée permet de contourner cette contrainte de façon incrémentale et ainsi réduire la consommation d'espace mémoire.

Pour ce faire, deux accumulateurs sont utilisés, soit $S_c(x, y)$ une matrice pour stocker la somme des intensités de chaque pixel et pour chacun des trois canaux

durant les N premières images et $SC_c(x, y)$ une matrice pour stocker la somme des carrés de ces derniers, que l'on définit comme suit :

$$S_c(x, y) = \sum_{i=0}^N I_{i,c}(x, y) \quad (3.7)$$

et

$$SC_c(x, y) = \sum_{i=0}^N [I_{i,c}(x, y)]^2 \quad (3.8)$$

Où I_i est la i ème image d'initialisation, N le nombre d'images utilisées, c le canal sélectionné et (x, y) les coordonnées du pixel traité.

La moyenne d'intensité d'un pixel pour un canal donné se résume, dès lors, par l'équation suivante :

$$\mu_c(x, y) = \frac{1}{N} S_c(x, y) \quad (3.9)$$

et l'écart-type par l'équation suivante :

$$\sigma_c = \sqrt{\left(\frac{SC_c(x, y)}{N}\right) - \mu_c(x, y)^2} \quad (3.10)$$

2. Extraction de l'avant-plan

Afin d'extraire l'objet d'intérêt dans une image, le modèle de l'arrière-plan doit lui être soustrait.

Chaque pixel dont la différence, en valeur absolue, avec la moyenne est supérieure à un certain multiple α de l'écart-type ($> \alpha\sigma$) sera classé comme un pixel en mouvement. En pratique, ce paramètre se situe dans l'intervalle $[1.0, 3.0]$ et dépend du niveau d'exclusion désiré. Un masque binaire de l'objet peut alors être généré pour chaque canal à l'aide de :

$$m_c(x, y) = \begin{cases} 1 & \text{si } |I_c(x, y) - \mu_c(x, y)| > \alpha\sigma_c(x, y) \\ 0 & \text{sinon} \end{cases} \quad (3.11)$$

où $m_c(x, y)$ représente la présence ou l'absence de mouvement au niveau du pixel (x, y) pour le canal c . I_c est l'image d'entrée à analyser.

Par la suite les masques des trois canaux sont combinés à l'aide d'un opérateur « OU » logique. Autrement dit, si un mouvement est détecté pour un pixel dans un seul canal, cela sera suffisant pour en modifier l'état. L'équation suivante représente cette combinaison produisant ainsi le masque M de l'objet pour l'image testée (**Figure 3.3c**) :

$$M(x, y) = m_R(x, y) \cup m_G(x, y) \cup m_B(x, y) \quad (3.12)$$

3. Mise à jour du modèle

Au cours de la période d'acquisition, certaines régions de la scène peuvent subir

3.3. Vue globale du système construit

des modifications d'éclairage ou de décor, ce qui rend la mise à jour du modèle statique (l'arrière-plan) primordiale. Afin de procéder à la mise à jour de l'image référence de l'arrière-plan, le complément \bar{M} du masque M généré dans l'étape précédente est calculé. Puis l'image du fond, qui rappelons-le est la moyenne des pixels de l'arrière-plan, est modifiée selon l'équation suivante :

$$\mu'_c(x, y) = (1 - \eta)\mu_c(x, y) + \eta I_c(x, y)\bar{M}(x, y) \quad (3.13)$$

où μ' représente un pixel de l'arrière-plan moyen mis à jour, I l'image courante, \bar{M} le complément du masque M et η le taux d'apprentissage. Ce dernier représente le pourcentage du nombre de pixels de l'arrière-plan à modifier. En pratique, ce taux d'apprentissage peut prendre des valeurs comprises entre l'intervalle $[0.05, 0.25]$. Plus la valeur de ce paramètre est élevée, plus les changements s'intégreront rapidement. Cela revient alors à oublier rapidement le modèle construit lors de la phase d'initialisation. Il est alors conseillé d'utiliser des valeurs relativement faibles (par exemple 0.05).

Dans le cas de notre étude, différents tests manuels ont été effectués afin de déterminer les valeurs des paramètres α et η . Nous avons fixé les valeurs de ces derniers à 2.5 et 0.05, respectivement.

Les changements dans les séquences traitées dans ces travaux sont relativement minimes, dès lors, la mise à jour de l'arrière-plan n'est effectuée que toutes les quatre images. Néanmoins, dans de rares cas, des changements de luminosité soudains mais de très courte période peuvent survenir, produisant ainsi du bruit qui sera corrigé par les processus de prétraitements à suivre.

3.3.2 Opérations de prétraitement des images

Après l'opération de soustraction de fond, les images résultantes sont souvent bruitées. De plus certains paramètres importants à la suite de notre étude doivent être normalisés afin d'optimiser le processus de reconnaissance. Ces paramètres sont la taille et la position de la silhouette de l'objet dans l'image.

— Filtrage et opération morphologique

Afin de raffiner les images et diminuer le bruit potentiel dans celles-ci, une première correction est appliquée à l'aide d'une opération morphologique d'ouverture, i.e : une opération d'érosion suivie d'une opération de dilatation à l'aide d'un noyau de taille 3×3 afin d'éliminer les pixels isolés. S'en suit une étape de filtrage à l'aide d'un filtre médian de taille 3×3 permettant ainsi de lisser et de mieux définir les contours de l'objet (**Figure 3.3c**).

— Translation et changement d'échelle

Nous considérons qu'en raison des différents points de vue de prise de l'action ou

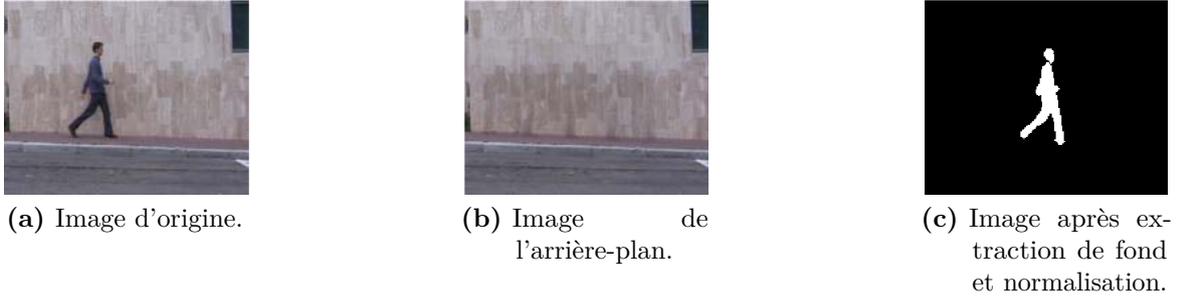


Figure 3.3 – Étape de soustraction de l'arrière-plan et extraction de la silhouette.

encore, les différences de mouvements effectués par une personne de grande taille et une personne de petite taille, la vitesse de translation globale du mouvement dans le monde réel est moins informative pour la reconnaissance de l'action que la forme et la position des membres par rapport au torse de la personne. De plus, afin d'optimiser l'extraction des caractéristiques et ainsi la reconnaissance du mouvement, le vecteur des images doit être invariant au déplacement dans le temps ainsi qu'aux différentes tailles des personnes. Pour cela nous normalisons ces deux paramètres comme suit.

Afin de normaliser la taille des silhouettes, nous avons défini un processus de changement d'échelle préservant le rapport d'aspect spatial de la posture décrite par l'image. Pour cela, nous avons, tout d'abord, défini une hauteur de posture arbitraire selon la base de vidéos étudiée. Puis, à partir de cette hauteur nous déduisons la nouvelle largeur de la silhouette, pour cela nous avons défini un rapport qui garantit la proportionnalité de la silhouette comme suit :

$$L' = \frac{L \times H'}{H} \quad (3.14)$$

où L et L' sont, respectivement, la largeur initiale et la largeur redéfinie et H et H' sont, respectivement, la hauteur initiale et la hauteur redéfinie des silhouettes.

Une fois, toutes les silhouettes normalisées selon une échelle uniforme, une translation des centres de gravité de ces dernières est effectuée afin de les aligner selon un même point de référence qui, dans notre cas, correspond au centre de l'image. Pour cela, nous calculons, en premier, les coordonnées du centre de gravité de la silhouette, puis nous déterminons les nouvelles coordonnées d'origines de la silhouette de la manière qui suit :

$$\begin{pmatrix} x_s' \\ y_s' \end{pmatrix} = \begin{pmatrix} C_{rx} \\ C_{ry} \end{pmatrix} - \begin{pmatrix} C_{gx} - x_s \\ C_{gy} - y_s \end{pmatrix} \quad (3.15)$$

où (x_s, y_s) et (x_s', y_s') sont, respectivement, les coordonnées initiales et les nouvelles coordonnées des pixels de la silhouette. (C_{rx}, C_{ry}) représentent les

3.3. Vue globale du système construit

coordonnées du centre de l'image et enfin (C_{gx}, C_{gy}) représentent les coordonnées du centre de gravité de l'objet (**Figure 3.3c**).

Algorithme 3: Soustraction de l'arrière-plan et prétraitement des images.

Entrées: vidéo

Sorties: séquence d'images binaires normalisées

début

Const : $N = 30, \eta = 0.05, \alpha = 2.5;$

Variables globales:

$S_c[]$ une matrice de taille $Width \times Lenght \times 3;$

$SC_c[]$ une matrice de taille $Width \times Lenght \times 3;$

$\mu_c[]$ une matrice de taille $Width \times Lenght \times 3;$

$\sigma_c[]$ une matrice de taille $Width \times Lenght \times 3;$

 /* Width et Lenght correspondent à la taille de l'image. */

 /* Initialisation. */

 Lire la vidéo;

tant que *Nombre images* < N **faire**

pour chaque *Canal* **faire**

 Calculer $S_c[]$ selon l'équation (3.7);

 Calculer $SC_c[]$ selon l'équation (3.8);

fin

fin

pour chaque *Canal* **faire**

 Calculer $\mu_c[]$ selon l'équation (3.9);

 Calculer $\sigma_c[]$ selon l'équation (3.10);

fin

 /* Extraction de la silhouette. */

 Lire la vidéo;

tant que \neg *fin vidéo* **faire**

pour chaque *Canal* **faire**

 Générer le masque de chaque canal m_c selon l'équation (3.11);

fin

 Générer le masque M de l'image selon l'équation (3.12);

 /* Opérations de prétraitement des images. */

 Appliquer opération morphologique d'ouverture;

 Normaliser la taille de la silhouette selon l'équation (3.13);

 Translater la silhouette au centre de l'image selon l'équation (3.14);

 /* Mise à jour de l'arrière-plan. */

si *Nombre images* $\equiv 0 \pmod{4}$ **alors**

 Générer \bar{M} complément de M ;

pour chaque *Canal* **faire**

 Mettre à jour $\mu_c[]$ selon l'équation (3.15);

fin

fin

fin

fin

Cette étape clôt la première phase de notre processus de reconnaissance. En résultat, pour chaque vidéo, une séquence d'images représentant des silhouettes binaires normalisées. Ces séquences de silhouettes seront présentées, par la suite, au processus de modélisation des actions définies par celles-ci.

3.3.3 Modélisation des actions par MDS

Notre approche est basée sur la considération des actions comme des formes tridimensionnelles induites par des silhouettes dans le temps. De manière similaire aux approches de **Yilmaz et al.** [227] et de **Gorelick et al.** [72], ce volume spatio-temporel résulte de la concaténation des silhouettes 2-D dans le temps afin de contenir à la fois, les informations spatiales sur la pose du sujet à tout moment (emplacement et orientation des membres du sujet) ainsi que les informations dynamiques du mouvement global (enchaînement des membres par rapport au torse du sujet).

Cependant, à l'inverse de l'approche de **Yilmaz et al.** [227] qui analyse le volume spatio-temporel brut en utilisant les propriétés de surface géométriques différentielles ou encore de l'approche de **Gorelick et al.** [72] qui extrait les caractéristiques du volume spatio-temporel, l'extraction de caractéristiques, dans notre approche, se fait à l'aide d'une réduction de dimensionnalité. En effet chaque volume spatio-temporel est réduit dans un espace à une dimension suivant l'axe temporel afin d'obtenir une représentation de l'action, correspondant à une image 2-D en niveaux de gris, soulignant le mouvement global.

Le choix de réduire les cubes spatio-temporels à une seule dimension a été motivé, en premier, par le fait d'obtenir des modèles représentant le mieux possible la dynamique du mouvement tout en minimisant la quantité de données redondantes et non pertinentes. Et, en second, par les résultats obtenus par [197] lors de la quantification du taux de perte d'information après l'application de la MDS sur le même jeux de données WEIZMANN utilisé lors de nos tests et qui est en moyenne inférieur à 20%.

La construction de notre base de modèles d'apprentissage s'effectue, pour chaque séquence d'images binarisées, comme suit.

1. Afin de traiter à la fois les actions périodiques et non périodiques, ainsi que pour compenser les différences de longueurs des séquences, nous utilisons une fenêtre glissante dans le temps dont la taille a fait l'objet de divers tests lors de l'expérimentation, et a été fixée à dix images avec cinq images de déplacement consécutif et ce jusqu'à la fin de la séquence (**Figure 3.4a**). Cette fenêtre permet de générer les cubes spatio-temporels, se composant chacun de dix images consécutives avec cinq images de chevauchement entre deux cubes spatio-temporels consécutifs. Un autre avantage de la fenêtre glissante est l'obtention de

3.3. Vue globale du système construit

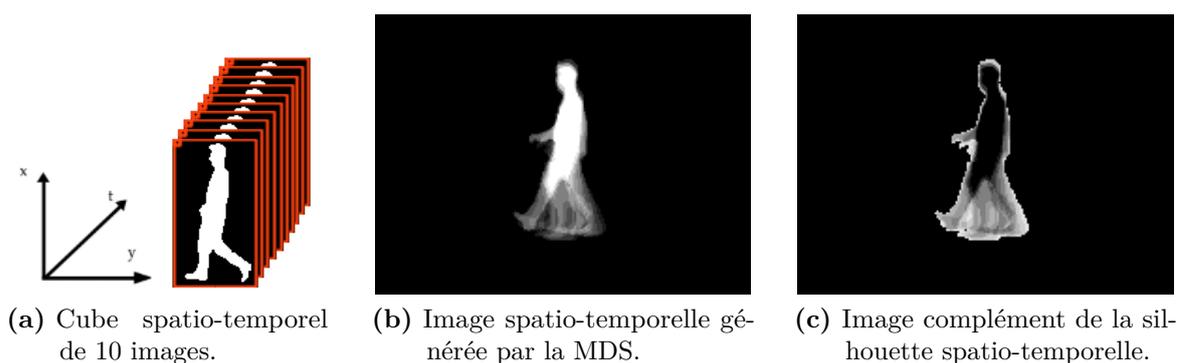


Figure 3.4 – Étape de modélisation des actions par MDS.

cubes spatio-temporels robustes et plus précis lors de la classification de longues séquences vidéos dans des scénarios réalistes. En résulte pour chaque action, une série de cubes spatio-temporels de même taille (hauteur et largeur des images de la séquence) et de profondeur de dix images.

2. Une fois que la séquence de l'action est décomposée en série de cubes spatio-temporels nous effectuons une réduction de dimensionnalité temporelle à l'aide de l'algorithme *Fastmap* décrit plus haut comme suit. Pour chaque pixel du cube spatio-temporel, l'algorithme construit un vecteur colonne de dimension N contenant les différentes valeurs d'intensité de celui-ci, où N correspond à la profondeur du cube spatio-temporel (temps). Afin d'extraire la structure spatiale sous-jacente, l'algorithme considère chaque vecteur de l'ensemble de données comme étant l'objet à projeter dans l'espace de dimension $k = 1$, où k représente la nouvelle dimension. Pour ce faire et suivant le concept de l'algorithme *Fastmap* décrit précédemment, ce dernier choisit une paire d'objets pivots correspondant à deux vecteurs de l'ensemble de données puis procède à la projection des autres objets (vecteurs) selon l'axe représenté par ces deux pivots en respectant le plus possible les distances Euclidiennes entre ces derniers. Cette opération est répétée M fois soit le nombre total de pixels d'une image de la séquence. De ce fait chaque vecteur de l'ensemble initial correspond à un point de l'espace réduit. Ces points représenteront les valeurs des intensités de pixels du cube spatio-temporel réduit que l'on nommera par la suite image spatio-temporelle (**Figure 3.4b**).
3. Nous remarquons que dans les images spatio-temporelles résultant de la réduction de la dimensionnalité, une distribution plus dense est obtenue au niveau du torse du sujet. En effet, étant donnée la propriété de la MDS à conserver les distances Euclidiennes des objets lors de la projection, la grande majorité des points de projections se trouvent au niveau de la partie subissant le moins de changements dans le temps qui en l'occurrence est le torse et ainsi la distribution des points au niveau des parties mobiles est plus dispersée. Dès lors, dans l'image spatio-temporelle induite, les valeurs de pixels les plus élevées correspondent au torse et les valeurs les plus faibles de pixels correspondent aux membres

3. IMPLEMENTATION ET RÉALISATION

mobiles. Cependant, puisque les régions du torse du sujet sont alignées, ces dernières sont par conséquent moins informatives sur la dynamique du mouvement comparées aux régions spatio-temporelles des membres mobiles. Pour cela, afin d'identifier les formes spatio-temporelles saillantes décrivant le mouvement, le complément de chaque silhouette spatio-temporelle est calculé, obtenant ainsi la forme représentant la dynamique du mouvement dans chaque image (**Figure 3.4c**).

À la suite de cette étape, chaque séquence d'action sera modélisée par une série d'images spatio-temporelles qui représenteront notre base d'apprentissage pour la reconnaissance des actions.

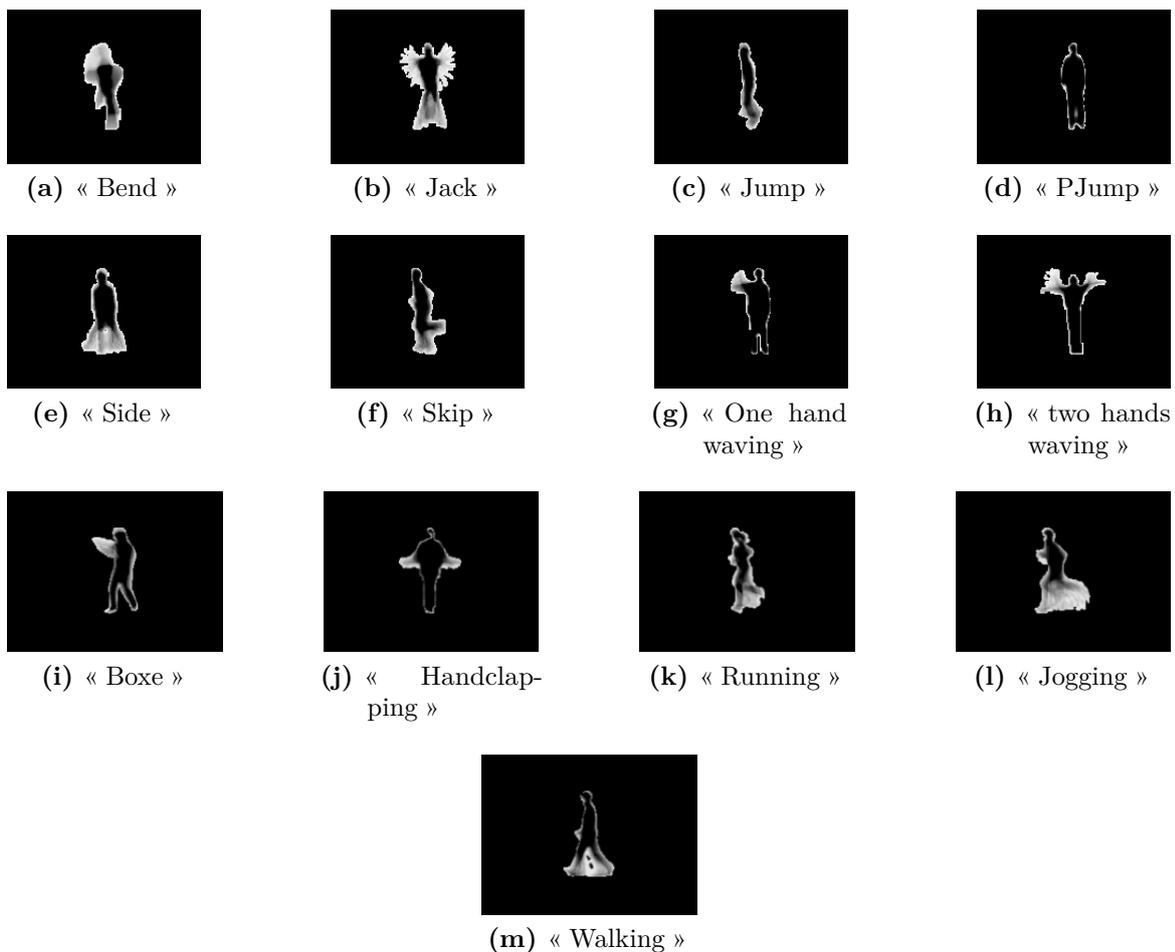


Figure 3.5 – Prototypes des actions étudiées.

3.3.4 Classification et reconnaissance des actions

Afin de procéder à la reconnaissance des mouvements, pour chaque séquence vidéo, nous effectuons une procédure de validation croisée *leave-one-out*, à savoir, une séquence entière (toutes ses images spatio-temporelles) est retirée de la base d'apprentissage tandis que les autres séquences d'actions de la même personne sont maintenues dans celle-ci.

Ainsi pour qu'une séquence vidéo soit correctement classée, celle-ci doit présenter une grande similarité avec une séquence d'une autre personne effectuant la même action.

Pour ce faire, chaque image spatio-temporelle de la séquence éliminée est comparée à toutes les images spatio-temporelles dans la base d'apprentissage à l'aide de la procédure du plus proche voisin basée sur une distance euclidienne entre les caractéristiques globales afin de générer un vecteur de scores indiquant la classe associée à chacune des images spatio-temporelles. Ce vecteur de scores est par la suite soumis à un vote pour designer le label majoritaire et ainsi attribuer une classe à l'action testée.

L'algorithme suivant résume les différentes étapes du processus de reconnaissance de l'action :

Algorithme 4: Système de reconnaissance des actions

Entrées: Vidéo de l'action

Sorties: Classe de l'action

début

 Soustraction de fond **Algorithme - 3** ;

pour chaque 10 images de la séquence **faire**

 | Calculer la MDS et générer l'image spatio-temporelle **Algorithme - 2** ;

fin

pour chaque Image spatio-temporelle **faire**

 | Trouver le plus proche voisin parmi les classes de la base d'apprentissage ;

fin

 Choisir la classe majoritaire parmi les classes attribuées aux images spatio-temporelles ;

fin

3.4

Conclusion

Nous avons présenté tout au long de ce chapitre le cheminement suivi pour la construction de notre système de reconnaissance de mouvements humains. Ce processus se divise en trois étapes importantes. Le prétraitement des vidéos, la construction de volumes spatio-temporels permettant l'extraction des caractéristiques et la modélisation des prototypes d'actions nécessaire à la reconnaissance de celles-ci. Enfin la classification et la reconnaissance des mouvements décrits par les vidéos. Dans le chapitre qui suit nous présenterons les expérimentations faites, ainsi que les résultats relatifs à ces dernières.

4

EXPÉRIMENTATIONS ET RÉSULTATS

Sommaire

4.1	Introduction	72
4.2	Résultats sur la base WEIZMANN	74
4.3	Résultats sur la base KTH	79
4.4	Conclusion	84

4.1 Introduction

Dans le présent chapitre, nous présentons les expérimentations menées afin d'évaluer les performances de notre modèle ainsi qu'une discussion des résultats obtenus en comparaison avec d'autres méthodes constituant l'état de l'art de la reconnaissance des actions humaines.

Nous testons l'efficacité de notre système sur la base de **WEIZMANN** et la base de **KTH**, deux ensembles de données publiques à usage académique qui forment à elles deux la référence actuelle dans l'état de l'art de la reconnaissance des actions humaines. Ces deux ensembles de données sont constitués de vidéos à faible résolution acquises à l'aide de caméras statiques, représentant des actions simples telles que marcher, courir, etc, effectuées par différents sujets, dans divers environnements contrôlés.

Ensemble de données WEIZMANN : L'ensemble de données **WEIZMANN** introduit par **Blank et al.** [21] et étendu par **Gorelick et al.** [72] se compose de 10 catégories d'actions simples qui sont : "run", "walk", "skip", "jumping-jack" (jack), "jump-forward-on-two-legs" (jump), "jump-in-place-on-two-legs" (pjump), "gallop-sideways" (side), "wave-two-hands" (wave2), "wave-one-hand" (wave1) et "bend". Chacune de ces actions est effectuée par 9 personnes, totalisant ainsi 93 séquences vidéo d'une durée de trois secondes et d'une résolution de $180 * 144$ et 25 fps (**Figure 4.1**). En plus de cet ensemble de données, les auteurs fournissent pour certaines séquences, les séquences de l'arrière-plan statique exempt de tout mouvement ainsi que deux autres ensembles distincts pour l'évaluation de la robustesse. Un ensemble de données représentant l'action "walk" sous différents angles de prise de vues et un autre ensemble de données représentant l'action "walk" avec des occultations ou encore avec différents vêtements.

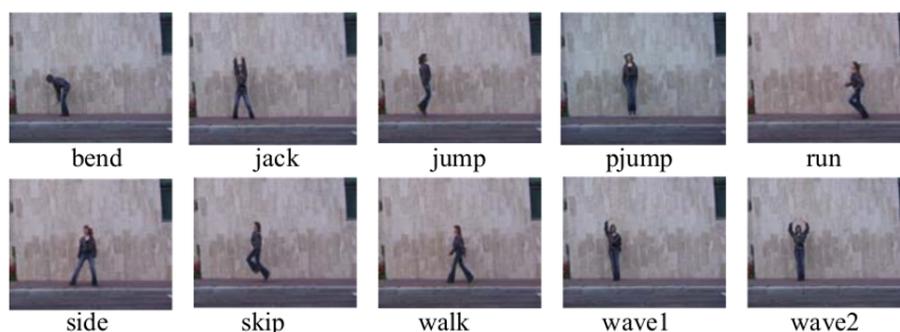


Figure 4.1 – Échantillon d'images extraites des séquences vidéos de la base **WEIZMANN**.

4.1. Introduction

Ensemble de données KTH : L'ensemble de données **KTH** introduit par **Schüldt et al.** [180] se compose, quant à lui, de 6 catégories d'actions simples qui sont : "walking", "jogging", "running", "boxing", "hand waving" et enfin "hand clapping". Chacune de ces actions est effectuée par 25 personnes et selon 4 scénarios différents : extérieur *s1*, extérieur avec changements d'échelle *s2*, extérieur avec changements de vêtements *s3* et finalement intérieur *s4*, totalisant ainsi 2387 séquences vidéo d'une résolution de $160 * 120$ et 25 fps (**Figure 4.2**). À la différence de l'ensemble de données **WEIZMANN**, cet ensemble présente de considérables variations dans les performances des actions et les durées d'exécution de celles-ci entre les sujets. De plus, l'extraction des silhouettes lors de l'étape de soustraction de fond n'est pas aisée. En effet, outre le scénario en extérieur avec changements d'échelle, les conditions d'acquisition des actions ne sont pas stables. Autrement dit, les vidéos sont capturées non seulement selon divers horizons et à des échelles légèrement différentes mais aussi avec des changements d'éclairage et des mouvements de caméras. De plus les auteurs ne fournissent pas de modèles d'arrière-plan. Un autre défi supplémentaire à cet ensemble de données est dû à l'existence de deux actions très similaires qui sont : "jogging" et "running".

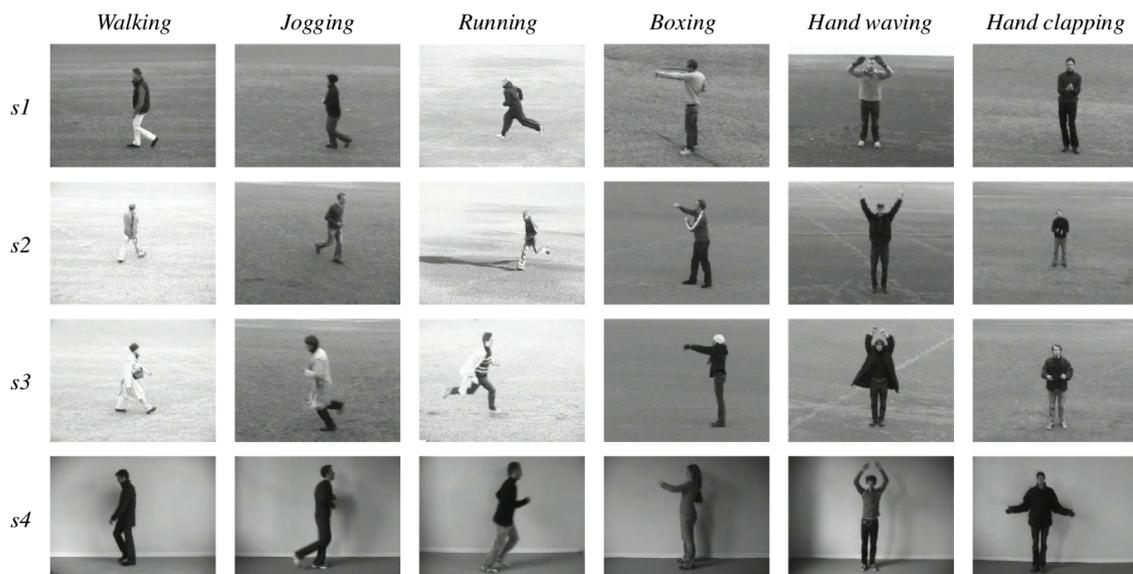


Figure 4.2 – Échantillon d'images extraites des séquences vidéos de la base **KTH**.

Ces deux ensembles de données sont conçus pour évaluer la capacité de classification des systèmes sur des actions simples. Chaque séquence vidéo des deux jeux de données représente l'exécution d'une seule action effectuée par un seul sujet. Autrement dit, les caractéristiques liées au mouvement entier extrait de chaque vidéo correspondent à une seule action. Dès lors, l'objectif est d'identifier l'action de la vidéo tout en sachant que celle-ci appartient à un nombre limité de classes d'actions connues. En outre, toutes les actions dans les deux bases, à l'exception de l'action "bend" dans la base **WEIZMANN**, sont des actions périodiques, ce qui rend ces bases particulièrement

adaptées à l'évaluation des méthodes basées sur l'exploitation de caractéristiques spatio-temporelles.

Dans la suite du chapitre, nous exposerons, en premier lieu, les résultats obtenus sur les bases **WEIZMANN** et **KTH** selon les critères d'évaluation des méthodes originales pour lesquelles ces bases ont été construites. Puis nous discuterons de nos résultats par rapport aux différentes méthodes évaluées sur ces ensembles.

4.2

Résultats sur la base WEIZMANN

La configuration et l'optimisation de notre approche s'est faite, en premier lieu, sur l'ensemble de données **WEIZMANN**. La base **WEIZMANN**, rappelons-le, est une base d'actions simples dont l'environnement est relativement stable. De ce fait lors de la soustraction de fond, les silhouettes extraites sont quasi parfaites, ne nécessitant pas de nettoyage particulier, ce qui, dans notre approche, est un grand avantage dans la mesure où elle est basée sur la pose décrite par chaque silhouette dans le temps et plus précisément sur les positions des membres mobiles par rapport au torse dans le temps.

Afin de générer les cubes spatio-temporels et ainsi les images spatio-temporelles, nous utilisons une fenêtre glissante dans le temps comme cela est expliqué dans le chapitre précédent. Pour ce faire, nous avons testé différentes tailles de fenêtres. Nous avons dans un premier temps pris une fenêtre de la taille de toute la séquence vidéo, produisant ainsi une seule image spatio-temporelle pour chaque action de la base. L'algorithme reconnaît 89 actions sur 93 et donc un taux de reconnaissance de 95.69%. Afin d'optimiser celui-ci, nous avons décrétementé la valeur de la taille de la fenêtre de 10 images à chaque test jusqu'à fixer celle-ci à une valeur de 10 images consécutives avec un chevauchement de 5 images entre deux cubes spatio-temporels consécutifs. Cette taille de fenêtre permet une bonne précision lors de la réduction des cubes spatio-temporels. En effet, l'algorithme reconnaît les 93 actions de la base (taux de reconnaissance de 100%).

Nous comparons nos résultats à ceux de **Gorelick et al.** [72] dont la méthode utilise les caractéristiques locales spatio-temporelles de saillance et les caractéristiques locales spatio-temporelles d'orientation extraites à partir des solutions de l'équation de Poisson appliquée sur un volume spatio-temporel construit à l'aide d'une fenêtre glissante de taille 8 images avec un chevauchement de 4 images entre deux cubes spatio-temporels consécutifs. Leur méthode est une généralisation de l'approche de [74] aux objets 3-D. Cette approche consiste à attribuer à chaque point interne de la silhouette en 2-D une valeur reflétant le temps moyen nécessaire pour que celui-ci atteigne le contour de la silhouette. Ce problème est résolu à l'aide de l'équation de Poisson. Le champ scalaire

4.2. Résultats sur la base WEIZMANN

résultant prend en compte de nombreux points sur le contour de la silhouette et ainsi extrait une grande variété de propriétés globales de la silhouette.

Dans les deux approches la validation croisée *leave-one-out* est adoptée afin d'estimer les erreurs de classification.

Gorelick et al. [72] rapportent un taux de reconnaissance de 97.83% sur 90 séquences vidéo (aucune précision n'est faite sur les trois séquences vidéo supplémentaires).

La figure (**Figure 4.3**) montre respectivement la matrice de confusion obtenue par notre approche (**Figure 4.3a**) et celle obtenue par l'approche de **Gorelick et al.** [72] (**Figure 4.3b**) à l'aide de l'algorithme des k -ppv avec $k = 1$. Il est à noter que les taux de reconnaissance résultant de notre méthode ne diffèrent pas avec l'augmentation du nombre des plus proches voisins ($k = 3, k = 5, k = 7$). Cela démontre le caractère discriminatif de notre approche.

	Bend	Jack	Jump	Pjump	Run	Side	Skip	Walk	Wave1	Wave2
Bend	100	0	0	0	0	0	0	0	0	0
Jack	0	100	0	0	0	0	0	0	0	0
Jump	0	0	100	0	0	0	0	0	0	0
Pjump	0	0	0	100	0	0	0	0	0	0
Run	0	0	0	0	100	0	0	0	0	0
Side	0	0	0	0	0	100	0	0	0	0
Skip	0	0	0	0	0	0	100	0	0	0
Walk	0	0	0	0	0	0	0	100	0	0
Wave1	0	0	0	0	0	0	0	0	100	0
Wave2	0	0	0	0	0	0	0	0	0	100

(a) Notre méthode

	Bend	Jack	Jump	Pjump	Run	Side	Skip	Walk	Wave1	Wave2
Bend	99.1	0	0	0	0	0	0	0	0	0.9
Jack	0	100	0	0	0	0	0	0	0	0
Jump	0	0	89.2	0	0	0	10.8	0	0	0
Pjump	0	0	0	100	0	0	0	0	0	0
Run	0	0	0	0	98	0	2	0	0	0
Side	0	0	0	0	0	100	0	0	0	0
Skip	0	0	0	0	2.9	0	97.1	0	0	0
Walk	0	0	0	0	0	0	0	100	0	0
Wave1	0	0.9	0	0.9	0	0	0	0	94.8	3.5
Wave2	0	0.9	0	0	0	0	0	0	1.9	97.2

(b) Méthode de **Gorelick et al.** [72]

Figure 4.3 – Matrices de confusion des actions lors de la classification.

Nous démontrons aussi la robustesse de notre système face à des irrégularités dans les performances des actions en testant celle-ci, avec la même configuration, sur deux jeux de séquences vidéos fournis par **Gorelick et al.** [72] Le premier ensemble consiste en dix séquences représentant l'action "walk" selon différents scénarios. Ces dernières sont testées sur la base d'entraînement originale **WEIZMANN**. Le **tableau 4.1** résume les résultats obtenus pour chaque scenario et montre ainsi que notre méthode

4. EXPÉRIMENTATIONS ET RÉSULTATS

n'est relativement pas sensible aux occultations partielles et aux déformations du mouvement.

	action "walk" avec	Notre méthode avec MDS	Gorelick et <i>al.</i> [72]	Notre méthode avec PCA
	dog	Walk	Walk	Side
	bag	Walk	Walk	Walk
	skirt	Walk	Walk	Walk
	nofeet	Walk	Walk	Bend
	pole	Walk	Walk	Walk
	moonwalk	Walk	Walk	Walk
	limp	Walk	Walk	Side
	kneesup	Walk	Walk	Pjump
	briefcase	Walk	Walk	Walk
	normwalk	Walk	Walk	Walk

Table 4.1 – Reconnaissance de l'action "walk" selon différents scénarios

4.2. Résultats sur la base WEIZMANN

En outre, nous démontrons la fiabilité de notre approche face aux changements d'angles de vue à l'aide du deuxième jeu de test qui représente l'action **"walk"** capturée selon des angles de vue variant de 0° à 81° par rapport au plan de l'image avec un pas de 9° à chaque séquence.

Nous remarquons à travers le **tableau 4.2**, que notre approche arrive à bien reconnaître l'action **"walk"** pour les angles de vue allant de 0° à 63° . Cependant aux angles de vue 72° et 81° , le système classe l'action comme étant l'action **"side"** ce qui demeure néanmoins cohérent dans la mesure où le mouvement **"side"** est très proche du mouvement **"walk"**.

En vue de justifier l'utilisation de la réduction de dimensionnalité par MDS, nous implémentons notre approche en substituant l'algorithme de la MDS par un autre algorithme de réduction de dimensionnalité standard, la PCA. Si le taux de reconnaissance des actions est optimal (100%) lors des tests sur la base **WEIZMANN** originale, nous remarquons que lors des tests de robustesse, la PCA performe moins bien que la MDS. En effet l'algorithme échoue à classer quatre séquences dans le premier jeu de données de robustesse et trois séquences dans les deuxième jeux de robustesse. De plus, les actions mal classées dans ces deux jeux sont très différentes de l'action **"walk"** (**tableaux 4.1 et 4.2**). Cela peut s'expliquer principalement par la propriété de la MDS à conserver le mieux possible les distances entre les vecteurs temporels lors de la projection. Ainsi celle-ci exploite mieux les données afin d'extraire la dynamique du mouvement dans le temps et non pas, uniquement, les formes des silhouettes.

4. EXPÉRIMENTATIONS ET RÉSULTATS

	action "walk" avec	Notre méthode avec MDS	Gorelick et <i>al.</i> [72]	Notre méthode avec PCA
	0°	Walk	Walk	Walk
	9°	Walk	Walk	Walk
	18°	Walk	Walk	Walk
	27°	Walk	Walk	Walk
	36°	Walk	Walk	Walk
	45°	Walk	Walk	Walk
	54°	Walk	Walk	Walk
	63°	Walk	Walk	Bend
	72°	Side	Walk	Bend
	81°	Side	Walk	Bend

Table 4.2 – Reconnaissance de l'action "walk" selon différents angles de vue

4.3 Résultats sur la base KTH

Tel que précisé précédemment, la base **WEIZMANN** est une base simple dont les difficultés ne permettent pas réellement de démontrer la robustesse de notre approche. De ce fait, afin de mieux illustrer la fiabilité et la robustesse de notre système, nous testons celui-ci sur la base de données à grande échelle **KTH** de **Schüldt et al.** [180]. En effet cet ensemble de données constitue un plus grand défi en raison des divers scénarios ainsi que les multiples variations de performance, d'échelle, d'angle de vue et enfin d'éclairage. En outre, comme décrit précédemment, les auteurs de cette base ne fournissent pas de fond de référence, en résulte une étape de soustraction de fond assez ardue. Les silhouettes extraites lors de celle-ci souffrent d'imperfections telles que l'absence de certaines parties du corps similaires à l'arrière-plan (jambe, bras). Ainsi nous démontrons dans la suite la fiabilité de notre méthode face à diverses occultations. Pour cela nous maintenons la même configuration que lors des expérimentations sur la base **WEIZMANN**, à savoir, une fenêtre glissante de taille dix images avec cinq images de chevauchement pour générer les cubes spatio-temporels ainsi que la procédure de validation croisée *leave-one-out*.

Nous comparons les résultats obtenus avec ceux de la méthode de **Schüldt et al.** [180] dont l'approche consiste à appliquer les SVM aux caractéristiques spatio-temporelles locales générées par la méthode de **Laptev et al.** [121].

Afin d'analyser l'influence des différents scénarios, nous entraînons et testons notre système sur chacun des scénarios s_1 , s_2 , s_3 et s_4 individuellement. La **figure 4.4** montre les matrices de confusion ainsi que les taux de reconnaissance obtenus à l'aide d'un 1-ppv. Nous constatons que les taux de reconnaissance sont satisfaisants et que le système parvient à identifier la majorité des actions.

4. EXPÉRIMENTATIONS ET RÉSULTATS

		Boxing	Handclapping	Handwaving	Jogging	Running	Walking	
Boxing	100	0	0	0	0	0	0	
Handclapping	1	99	0	0	0	0	0	
Handwaving	0	0	100	0	0	0	0	
Jogging	0	0	0	92	3	5	0	
Running	0	0	0	16	82	2	0	
Walking	0	0	0	0	0	0	100	
		(a) $s1 \approx 95.5\%$						
		Boxing	Handclapping	Handwaving	Jogging	Running	Walking	
Boxing	100	0	0	0	0	0	0	
Handclapping	0	99	1	0	0	0	0	
Handwaving	0	0	99	1	0	0	0	
Jogging	0	0	0	84	3	13	0	
Running	0	0	0	27.83	71.1	1.07	0	
Walking	0	0	0	1	0	99	0	
		(b) $s2 \approx 92.01\%$						
		Boxing	Handclapping	Handwaving	Jogging	Running	Walking	
Boxing	100	0	0	0	0	0	0	
Handclapping	0	97.91	2.09	0	0	0	0	
Handwaving	0	0	100	0	0	0	0	
Jogging	0	1	0	76	11	12	0	
Running	1	1	0	30	67	1	0	
Walking	3	0	0	1	0	96	0	
		(c) $s3 \approx 89.48\%$						
		Boxing	Handclapping	Handwaving	Jogging	Running	Walking	
Boxing	100	0	0	0	0	0	0	
Handclapping	2	98	0	0	0	0	0	
Handwaving	0	0	100	0	0	0	0	
Jogging	0	0	0	81	6	13	0	
Running	0	1	0	25	73	1	0	
Walking	0	0	0	1	0	98	0	
		(d) $s4 \approx 91.66\%$						

Figure 4.4 – Matrices de confusion des actions lors de la classification pour chaque scénario à l'aide de la MDS + 1-ppv.

La **figure 4.5** montre les matrices de confusion ainsi que les taux de reconnaissance obtenus suivant les procédures d'entraînement proposées par **Schüldt et al.** [180] qui consistent à entraîner tout l'ensemble de données **KTH** (les quatre scénarios en même temps) sur les sous-ensembles suivant : $\{s1\}$, $\{s1, s4\}$, $\{s1, s3, s4\}$ et enfin $\{s1, s2, s3, s4\}$.

4.3. Résultats sur la base KTH

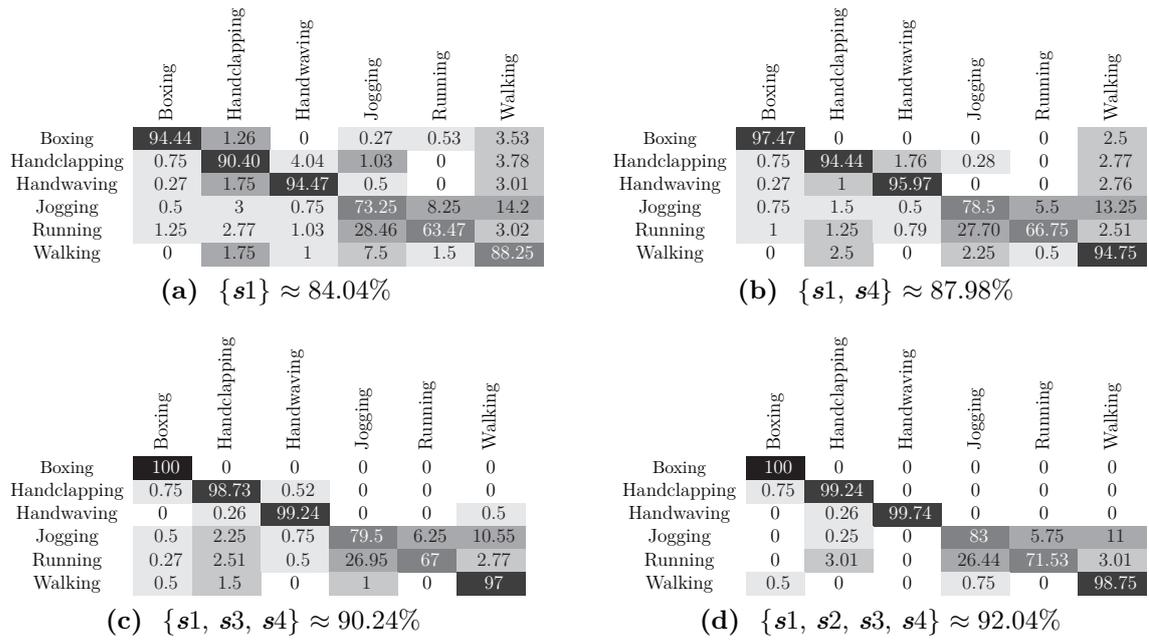


Figure 4.5 – Matrices de confusion des actions lors de la classification de $\{s1, s2, s3, s4\}$ à l'aide de la MDS + 1-ppv.

Étant donné que le scénario $s2$ avec changements d'échelle est celui qui présente le plus de difficultés. Nous exposons, aussi les résultats générés par le test du sous-ensemble $\{s2\}$ sur les sous-ensembles précédents **figure 4.6**.

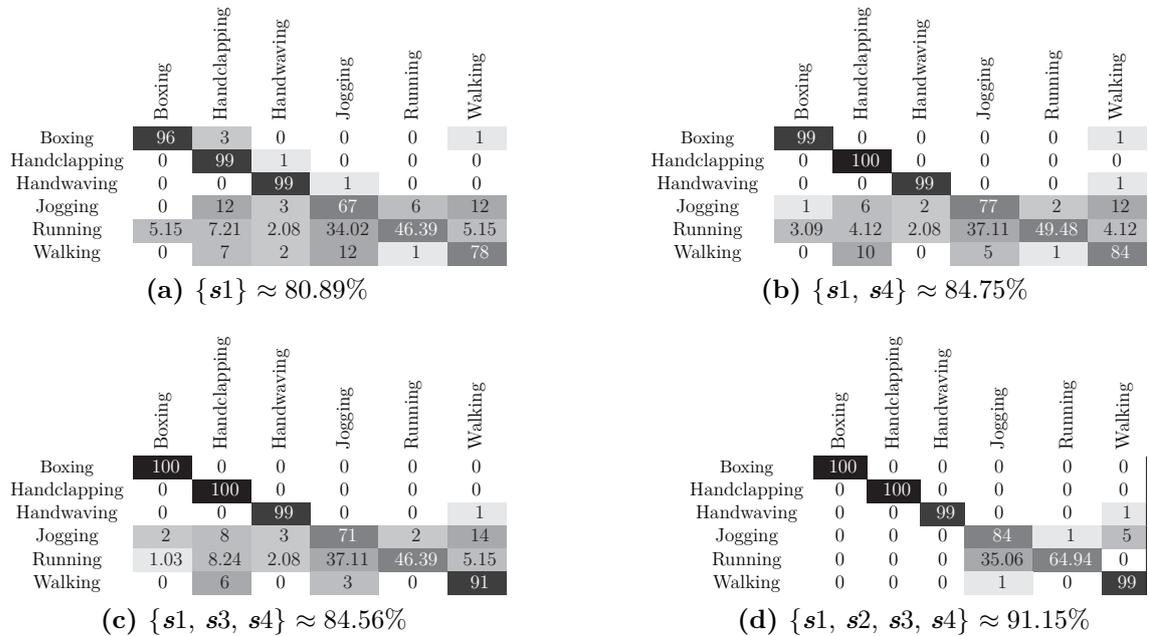


Figure 4.6 – Matrices de confusion des actions lors de la classification de $\{s2\}$ à l'aide de la MDS + 1-ppv.

4. EXPÉRIMENTATIONS ET RÉSULTATS

Dans la **figure 4.7**, est exposée une comparaison entre les résultats obtenus par notre méthode et ceux obtenus par la méthode de **Schüldt et al.** [180]. de manière générale, notre approche a de meilleurs taux de reconnaissance que celle de **Schüldt et al.** [180] et ce dans tous les scénarios.

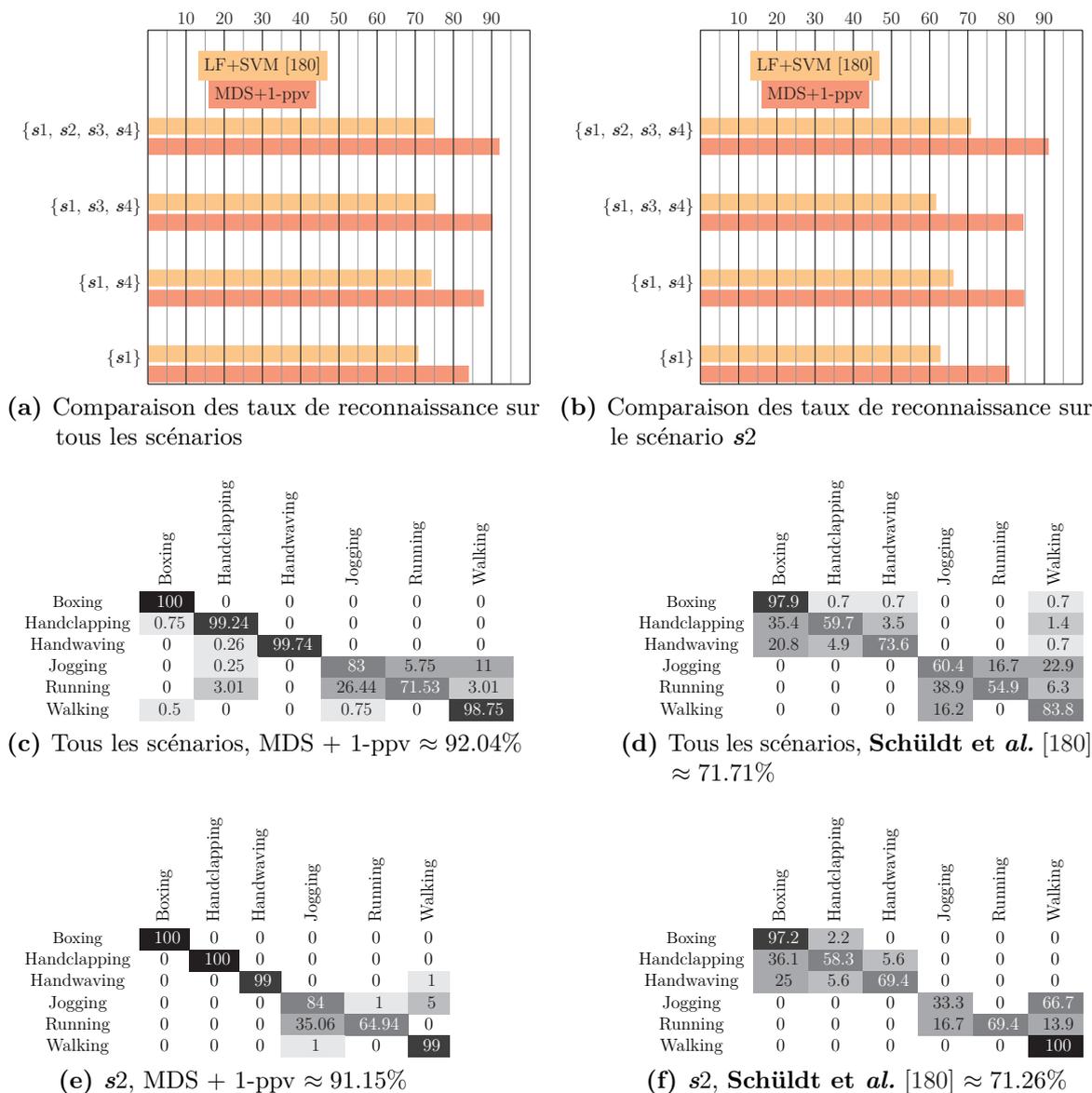


Figure 4.7 – Comparaison des résultats de notre méthode avec celle de **Schüldt et al.** [180].

4.3. Résultats sur la base KTH

La confusion entre "walking" et "jogging" tout comme la confusion entre "jogging" et "running" peut partiellement s'expliquer par la forte similarité entre ces classes d'actions (la façon de courir peut être interprétée différemment selon le sujet). Cependant à partir des matrices de confusion, on distingue clairement que notre approche réussit mieux à différencier ces actions que l'approche de **Schüldt et al.** [180].

De manière globale, nous constatons que les confusions surviennent pour des classes d'actions assez similaires dont la mécanique de mouvement est très semblable.

De même que lors des expérimentations sur la base **WEIZMANN**, nous avons substitué la MDS par la PCA. Dès lors le système reconnaît 1891 séquences d'actions sur la totalité des 2387 séquences d'actions soient un taux de reconnaissance de 79.22%.

Nous résumons à titre comparatif les résultats de diverses méthodes de l'état de l'art sur les deux ensembles de données étudiés dans le **tableau 4.3** et ce malgré le fait que les approches, les méthodes d'évaluation et l'utilisation des deux ensembles de données diffèrent d'un article à l'autre pour une comparaison directe.

Méthode	WEIZMANN		KTH	
	MDS :100%	PCA :100%	MDS :92.04%	PCA :79.22%
Schüldt et al. [180]			71.71%	
Blank et al. [21]	99.6%			
Dollár et al. [50]			81.2%	
Ke et al. [108]			63%	
Niebles et al. [149]	72.8%		81.5%	
Ikizler et al. [91]	100%		89.4%	
Jhuang et al. [98]	98.8%		91.7%	
Laptev et al. [120]			91.8%	
Meng et al. [142]			80.3%	
Nowozin et al. [150]	84.7%			
Scovanner et al. [181]			82.6%	
Wang et al. [206]	100%			
Wang et al. [210]			92.4%	
Wong et al. [213]			81%	
Fathi et al. [60]	100%		90.5%	
Gorelick et al. [72]	97.83%			
Gilbert et al. [70]			89.9%	
Junejo et al. [102]	95.3%			
Klaser et al. [112]	84.3%		91.4%	
Liu et al. [132]	89.3%		94.2%	
Schindler et al. [177]	100%		92.7%	
Zhang et al. [234]	92.89%		91.33%	

Table 4.3 – Tableau comparatif des taux de reconnaissance sur les bases WEIZMANN et KTH

Nous constatons que les performances de notre approche sont relativement compa-

rables aux meilleures approches en vue de la simplicité de la procédure d'extraction des caractéristiques ainsi que la simplicité du classifieur choisi. De plus notre système bénéficie de plusieurs avantages :1) il est facile à comprendre et à implémenter, 2) il se base uniquement sur les formes des silhouettes dans le temps, ainsi il ne nécessite ni alignement temporel de vidéos au préalable ni de suivi 2-D ou 3-D explicite, 3) il se soustrait aux difficultés de suivi temporel de caractéristiques, de calcul de flux optique et de l'extraction de caractéristiques basées sur le gradient ou l'intensité des pixels et par conséquent à leurs complexités et faiblesses, 4) notre système est robuste face aux séquences vidéo à très faible résolution où certaines méthodes, notamment celles basées sur les intensités des pixels, rencontrent d'éventuelles difficultés et enfin 5) notre approche est très rapide. En effet, le temps de calcul pour générer le modèle d'une action est de 5 millisecondes incluant la soustraction de fond et le prétraitement des images.

4.4

Conclusion

Nous avons illustré dans ce chapitre les résultats obtenus par notre approche de reconnaissance d'actions qui a le mérite d'être simple. Ainsi telles que les expérimentations le démontrent, la méthode est fiable et robuste face aux changements d'échelles, d'environnement, aux occultations partielles et enfin aux déformations des actions. En outre, bien que notre approche ne soit pas totalement invariante aux changements d'angles de vue, elle reste néanmoins assez robuste lors des tests de robustesse de l'ensemble **WEIZMANN** et les tests sur scénario *s2* de l'ensemble **KTH**.

CONCLUSION GÉNÉRALE

Avec le progrès scientifique et technologique, la recherche en vision par ordinateur s'est orientée vers la compréhension de scène, comportant tout type d'objet et plus particulièrement vers l'analyse de scène comportant des humains, de ce fait, nous nous sommes intéressés à la reconnaissance du mouvement humain.

Nous avons construit un système basé sur une approche spatio-temporelle pour la reconnaissance d'actions humaines. La nature périodique des actions simples nous a incité à exploiter l'information globale d'un volume spatio-temporel à l'aide d'un processus d'extraction de caractéristiques globales afin de procéder à la reconnaissance des actions de façon automatique, efficace et particulièrement simple.

Pour ce faire, nous avons choisi de modéliser nos prototypes d'actions humaines à l'aide de la technique de réduction de dimensionnalité Multi-Dimensional Scaling MDS et ainsi visualiser les caractéristiques spatio-temporelles globales que prend la forme de la silhouette d'un sujet dans le temps pour une action donnée. Le choix de la MDS a été motivé, principalement, par sa capacité à représenter les données en espace de dimension réduite tout en respectant la géométrie globale de l'action dans le temps en considérant les relations spatiales et temporelles entre les silhouettes.

Les résultats obtenus lors de nos tests sont très bons et compétitifs par rapport aux différentes approches de l'état de l'art actuel en vue de la simplicité du processus de modélisation des actions, de l'utilisation d'un algorithme de classification non paramétrique basé sur les k-ppv et une distance Euclidienne. De plus, notre approche est robuste face aux changements d'échelles, d'environnement, aux occlusions partielles, aux déformations des actions et enfin aux changements d'angles de vue.

Enfin nous souhaitons terminer en évoquant des améliorations pouvant être apportées à cette étude :

Étendre la reconnaissance du comportement aux activités et interactions diverses en envisageant l'utilisation de classifieurs plus performants tels que les SVM [123], les réseaux de neurones à convolution [226] ou encore [187].

Nous envisageons, aussi, de généraliser notre approche à la reconnaissance de modèles en trois dimensions en reconstruisant les silhouettes extraites à partir de plusieurs vues puis, en appliquant le même procédé de réduction et de classification sur les bases de données IXMAS [93] et MuHAVi [200].

BIBLIOGRAPHIE

- [1] “Caviar : Context aware vision using image-based active recognition,” <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/caviar.htm>, 2002–2005.
- [2] “Etiseo : Evaluation du traitement et de l’interpretation de sequences video,” <http://www-sop.inria.fr/orion/ETISEO/index.htm>, 2005.
- [3] “Casia action database,” <http://www.cbsr.ia.ac.cn/english/Gait>
- [4] “Utexas databases,” <http://cvrc.ece.utexas.edu/SDHA2010/>, 2010.
- [5] “Kitware, virat video dataset,” <http://www.viratdata.org/>, 2011.
- [6] J. K. Aggarwal and M. S. Ryoo, “Human activity analysis : A review,” *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, p. 16, 2011.
- [7] D. K. Agrafiotis, “Stochastic proximity embedding,” *Journal of computational chemistry*, vol. 24, no. 10, pp. 1215–1221, 2003.
- [8] A. Aizerman, E. M. Braverman, and L. Rozoner, “Theoretical foundations of the potential function method in pattern recognition learning,” *Automation and remote control*, vol. 25, pp. 821–837, 1964.
- [9] M. S. Allili, N. Bouguila, and D. Ziou, “A robust video foreground segmentation by using generalized gaussian mixture modeling,” in *Computer and Robot Vision, 2007. CRV’07. Fourth Canadian Conference on*. IEEE, 2007, pp. 503–509.
- [10] F. E. Baf, T. Bouwmans, and B. Vachon, “A fuzzy approach for background subtraction,” in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. IEEE, 2008, pp. 2648–2651.
- [11] D. Baltieri, R. Vezzani, and R. Cucchiara, “Fast background initialization with recursive hadamard transform,” in *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*. IEEE, 2010, pp. 165–171.
- [12] O. Barnich and M. Van Droogenbroeck, “Vibe : a powerful random technique to estimate the background in video sequences,” in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 945–948.
- [13] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, “Performance of optical flow techniques,” *International journal of computer vision*, vol. 12, no. 1, pp. 43–77, 1994.
- [14] D. Batra, T. Chen, and R. Sukthankar, “Space-time shapelets for action recognition,” in *Motion and video Computing, 2008. WMVC 2008. IEEE Workshop on*. IEEE, 2008, pp. 1–6.
- [15] M. Belkin and P. Niyogi, “Using manifold structure for partially labeled classification,” in *Advances in neural information processing systems*, 2002, pp. 929–936.

-
- [16] M. Benalia and S. Ait-Aoudia, “An improved basic sequential clustering algorithm for background construction and motion detection,” in *Image Analysis and Recognition*. Springer, 2012, pp. 216–223.
- [17] H. Bhaskar, L. Mihaylova, and A. Achim, “Video foreground detection based on symmetric alpha-stable mixture models,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 20, no. 8, pp. 1133–1138, 2010.
- [18] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.
- [19] S. Biswas, K. W. Bowyer, and P. J. Flynn, “Multidimensional scaling for matching low-resolution face images,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 10, pp. 2019–2030, 2012.
- [20] J. Blackburn and E. Ribeiro, “Human motion recognition using isomap and dynamic time warping,” in *Human Motion—Understanding, Modeling, Capture and Animation*. Springer, 2007, pp. 285–298.
- [21] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, “Actions as space-time shapes,” in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 1395–1402.
- [22] H. Blum and R. N. Nagel, “Shape description using weighted symmetric axis features,” *Pattern recognition*, vol. 10, no. 3, pp. 167–180, 1978.
- [23] A. F. Bobick and J. W. Davis, “The recognition of human movement using temporal templates,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 3, pp. 257–267, 2001.
- [24] O. Boiman and M. Irani, “Detecting irregularities in images and in video,” *International Journal of Computer Vision*, vol. 74, no. 1, pp. 17–31, 2007.
- [25] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, 1992, pp. 144–152.
- [26] M. Bregonzio, S. Gong, and T. Xiang, “Recognising action as clouds of space-time interest points,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1948–1955.
- [27] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.
- [28] S. Brutzer, B. Höferlin, and G. Heidemann, “Evaluation of background subtraction techniques for video surveillance,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1937–1944.
- [29] S. S. Bucak, B. Günsel, and O. Gursoy, “Incremental non-negative matrix factorization for dynamic background modelling,” in *PRIS*, 2007, pp. 107–116.
- [30] C. J. Burges, “A tutorial on support vector machines for pattern recognition,” *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

- [31] D. E. Butler, V. M. Bove Jr, and S. Sridharan, “Real-time adaptive foreground/background segmentation,” *EURASIP journal on applied signal processing*, vol. 2005, pp. 2292–2304, 2005.
- [32] D. Cai, X. He, and J. Han, “Semi-supervised discriminant analysis,” in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–7.
- [33] L. W. Campbell and A. E. Bobick, “Recognition of human body motion using phase space constraints,” in *Computer Vision, 1995. Proceedings., Fifth International Conference on*. IEEE, 1995, pp. 624–630.
- [34] J. Carranza, C. Theobalt, M. A. Magnor, and H.-P. Seidel, “Free-viewpoint video of human actors,” *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3, pp. 569–577, 2003.
- [35] C. Cedras and M. Shah, “Motion-based recognition a survey,” *Image and Vision Computing*, vol. 13, no. 2, pp. 129–155, 1995.
- [36] R. Chang, T. Gandhi, and M. M. Trivedi, “Vision modules for a multi-sensory bridge monitoring approach,” in *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*. IEEE, 2004, pp. 971–976.
- [37] S.-C. S. Cheung and C. Kamath, “Robust background subtraction with foreground validation for urban traffic video,” *Eurasip Journal on applied signal processing*, vol. 2005, pp. 2330–2340, 2005.
- [38] T.-J. Chin, L. Wang, K. Schindler, and D. Suter, “Extrapolating learned manifolds for human activity recognition,” in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 1. IEEE, 2007, pp. I–381.
- [39] O. Chomat and J. L. Crowley, “Probabilistic recognition of activity using local appearance,” in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2. IEEE, 1999.
- [40] Y.-C. Chung, J.-M. Wang, and S.-W. Chen, “Progressive background images generation,” in *Proc. of 15th IPPR Conf. on Computer Vision, Graphics and Image Processing*, 2002, pp. 858–865.
- [41] R. V. Colque and G. Cámara-Chávez, “Progressive background image generation of surveillance traffic videos based on a temporal histogram ruled by a reward/penalty function,” in *Graphics, Patterns and Images (Sibgrapi), 2011 24th SIBGRAPI Conference on*. IEEE, 2011, pp. 297–304.
- [42] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [43] T. F. Cox and M. A. Cox, *Multidimensional scaling*. CRC Press, 2000.
- [44] R. Cucchiara and M. Piccardi, “Vehicle detection under day and night illumination.” in *IIA/SOCO*, 1999.
- [45] D. Culibrk, O. Marques, D. Socek, H. Kalva, and B. Furht, “A neural network approach to bayesian background modeling for video object segmentation.” in *VISAPP (1)*, 2006, pp. 474–479.

-
- [46] —, “Neural network approach to background modeling for video object segmentation,” *Neural Networks, IEEE Transactions on*, vol. 18, no. 6, pp. 1614–1627, 2007.
- [47] S. P. Curram and J. Mingers, “Neural networks, decision tree induction and discriminant analysis : An empirical comparison,” *Journal of the Operational Research Society*, pp. 440–450, 1994.
- [48] K. Diamantras and S. Kung, “Principal component neural networks,” 1996.
- [49] J. Ding, M. Li, K. Huang, and T. Tan, “Modeling complex scenes for accurate moving objects segmentation,” in *Computer Vision–ACCV 2010*. Springer, 2011, pp. 82–94.
- [50] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, “Behavior recognition via sparse spatio-temporal features,” in *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*. IEEE, 2005, pp. 65–72.
- [51] Y. Dong and G. N. DeSouza, “Adaptive learning of multi-subspace for foreground detection under illumination changes,” *Computer Vision and Image Understanding*, vol. 115, no. 1, pp. 31–49, 2011.
- [52] A. Doshi and M. Trivedi, “" hybrid cone-cylinder" codebook model for foreground detection with shadow and highlight suppression,” in *Video and Signal Based Surveillance, 2006. AVSS’06. IEEE International Conference on*. IEEE, 2006, pp. 19–19.
- [53] A. Efros, A. C. Berg, G. Mori, J. Malik *et al.*, “Recognizing action at a distance,” in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 726–733.
- [54] F. El Baf, T. Bouwmans, and B. Vachon, “Comparison of background subtraction methods for a multimedia learning space.” in *SIGMAP*, 2007, pp. 153–158.
- [55] A. Elgammal, D. Harwood, and L. Davis, “Non-parametric model for background subtraction,” in *Computer Vision—ECCV 2000*. Springer, 2000, pp. 751–767.
- [56] C. Faloutsos and K.-I. Lin, *FastMap : A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets*. ACM, 1995, vol. 24, no. 2.
- [57] D. Fan, M. Cao, and C. Lv, “An updating method of self-adaptive background for moving objects detection in video,” in *Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on*. IEEE, 2008, pp. 1497–1501.
- [58] X. Fang, W. Xiong, B. Hu, and L. Wang, “A moving object detection algorithm based on color information,” in *Journal of Physics : Conference Series*, vol. 48, no. 1. IOP Publishing, 2006, p. 384.
- [59] D. Farcas, C. Marghes, and T. Bouwmans, “Background subtraction via incremental maximum margin criterion : a discriminative subspace approach,” *Machine Vision and Applications*, vol. 23, no. 6, pp. 1083–1101, 2012.

Bibliographie

- [60] A. Fathi and G. Mori, "Action recognition by learning mid-level motion features," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.
- [61] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI magazine*, vol. 17, no. 3, p. 37, 1996.
- [62] E. Fiesler, "Neural network topologies," *The Handbook of Neural Computation, E. Fiesler and R. Beale (Editors-in-Chief), Oxford University Press and IOP Publishing*, 1996.
- [63] R. Fisher, "Behave : computer-assisted prescreening of video streams for unusual activities," <http://homepages.inf.ed.ac.uk/rbf/BEHAVE/>, 2004.
- [64] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.
- [65] D. A. Forsyth, O. Arikian, and L. Ikemoto, *Computational Studies of Human Motion : Tracking and Motion Synthesis*. Now Publishers Inc, 2006.
- [66] H. Freeman, "On the encoding of arbitrary geometric configurations," *Electronic Computers, IRE Transactions on*, no. 2, pp. 260–268, 1961.
- [67] D.-s. Gao, J. Zhou, and L.-p. Xin, "A novel algorithm of adaptive background estimation," in *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 2. IEEE, 2001, pp. 395–398.
- [68] T. Gao, Z.-g. Liu, W.-c. Gao, and J. Zhang, "A robust technique for background subtraction in traffic video," in *Advances in Neuro-Information Processing*. Springer, 2009, pp. 736–744.
- [69] D. M. Gavrila, "The visual analysis of human movement : A survey," *Computer vision and image understanding*, vol. 73, no. 1, pp. 82–98, 1999.
- [70] A. Gilbert, J. Illingworth, and R. Bowden, "Scale invariant action recognition using compound features mined from dense spatio-temporal corners," in *Computer Vision–ECCV 2008*. Springer, 2008, pp. 222–233.
- [71] S. Gong, S. McKenna, and J. J. Collins, "An investigation into face pose distributions," in *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*. IEEE, 1996, pp. 265–270.
- [72] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 12, pp. 2247–2253, 2007.
- [73] —, "Actions as space-time shapes, transactions on pattern analysis and machine intelligence 29 (12)," <http://www.wisdom.weizmann.ac.il/vision/SpaceTimeActions.html>, 2007.
- [74] L. Gorelick, M. Galun, E. Sharon, R. Basri, and A. Brandt, "Shape representation and classification using the poisson equation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 12, pp. 1991–2005, 2006.

-
- [75] V. C. Group, “Videoweb dataset,” <http://www.ee.ucr.edu/~amitrc/vwdata.php>, 2010.
- [76] V. G. Group, “Tv human interactions dataset,” <http://www.robots.ox.ac.uk/vgg/data/tvhumaninteractions/index.html>, 2010.
- [77] G. Guerra-Filho, “Optical motion capture : Theory and implementation.” *RITA*, vol. 12, no. 2, pp. 61–90, 2005.
- [78] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.
- [79] B. Han and R. Jain, “Real-time subspace-based background modeling using multi-channel data,” in *Advances in Visual Computing*. Springer, 2007, pp. 162–172.
- [80] C. Harris and M. Stephens, “A combined corner and edge detector.” in *Alvey vision conference*, vol. 15. Citeseer, 1988, p. 50.
- [81] Y. He, D. Wang, and M. Zhu, “Background subtraction based on nonparametric bayesian estimation,” in *3rd International Conference on Digital Image Processing*. International Society for Optics and Photonics, 2011, pp. 80 090G–80 090G.
- [82] G. E. Hinton and S. T. Roweis, “Stochastic neighbor embedding,” in *Advances in neural information processing systems*, 2002, pp. 833–840.
- [83] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [84] H. Hoffmann, “Kernel pca for novelty detection,” *Pattern Recognition*, vol. 40, no. 3, pp. 863–874, 2007.
- [85] H. Hotelling, “Relations between two sets of variates,” *Biometrika*, pp. 321–377, 1936.
- [86] H. Hu, L. Xu, and H. Zhao, “A spherical codebook in yuv color space for moving object detection,” *Sensor Letters*, vol. 10, no. 1-2, pp. 177–189, 2012.
- [87] M.-K. Hu, “Visual pattern recognition by moment invariants,” *Information Theory, IRE Transactions on*, vol. 8, no. 2, pp. 179–187, 1962.
- [88] T. Huang, D. Koller, J. Malik, G. Ogasawara, B. Rao, S. J. Russell, and J. Weber, “Automatic symbolic traffic scene analysis using belief networks,” in *AAAI*, vol. 94, 1994, pp. 966–972.
- [89] W. Y. Huang and R. P. Lippmann, “Comparisons between neural net and conventional classifiers,” in *Proc. IEEE First International Conference on Neural Networks, San Diego, California*, 1987.
- [90] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent component analysis*. John Wiley & Sons, 2004, vol. 46.
- [91] N. Ikizler and P. Duygulu, “Human action recognition using distribution of oriented rectangular patches,” in *Human Motion–Understanding, Modeling, Capture and Animation*. Springer, 2007, pp. 271–284.

- [92] P. Indyk and R. Motwani, "Approximate nearest neighbors : towards removing the curse of dimensionality," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. ACM, 1998, pp. 604–613.
- [93] INRIA, "Ixmas : Inria xmas motion acquisition sequences," <http://4drepository.inrialpes.fr/public/viewgroup/6>, 2006.
- [94] Y. W. J. Yuan, Z. Liu, "Msr action dataset," <http://users.eecs.northwestern.edu/~jyu410/indexfiles/actiondetection.html>, 2009.
- [95] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering : a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.
- [96] R. Jain and H.-H. Nagel, "On the analysis of accumulative difference pictures from image sequences of real world scenes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 2, pp. 206–214, 1979.
- [97] F. V. Jensen, "Bayesian networks and decision graphs. statistics for engineering and information science," *Springer*, vol. 32, p. 34, 2001.
- [98] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A biologically inspired system for action recognition," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. Ieee, 2007, pp. 1–8.
- [99] K. Jia and D.-Y. Yeung, "Human action recognition using local spatio-temporal discriminant embedding," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.
- [100] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Attention, Perception, & Psychophysics*, vol. 14, no. 2, pp. 201–211, 1973.
- [101] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.
- [102] I. N. Junejo, E. Dexter, I. Laptev, and P. Pérez, *Cross-view action recognition from temporal self-similarities*. Springer, 2008.
- [103] C. Jutten and J. Herault, "Blind separation of sources, part i : An adaptive algorithm based on neuromimetic architecture," *Signal processing*, vol. 24, no. 1, pp. 1–10, 1991.
- [104] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning : A survey," *Journal of artificial intelligence research*, pp. 237–285, 1996.
- [105] Y. Kameda and M. Minoh, "A human motion estimation method using 3-successive video frames," in *International conference on virtual systems and multimedia*, 1996, pp. 135–140.
- [106] K. Karmann and A. Brandt, "Moving object recognition using and adaptive background memory, 2, 289-307," *Time-Varying Image Processing and Moving Object Recognition, Cappellini V.(Ed)*, 1990.
- [107] S. Kawabata, S. Hiura, and K. Sato, "Real-time detection of anomalous objects in dynamic scene," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 3. IEEE, 2006, pp. 1171–1174.

-
- [108] Y. Ke, R. Sukthankar, and M. Hebert, "Efficient visual event detection using volumetric features," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1. IEEE, 2005, pp. 166–173.
- [109] Y. Kel, R. Sukthankar, and M. Hebert, "Spatio-temporal shape and flow correlation for action recognition," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.
- [110] H. Kim, R. Sakamoto, I. Kitahara, T. Toriyama, and K. Kogure, "Robust foreground extraction technique using gaussian family model and multiple thresholds," in *Computer Vision-ACCV 2007*. Springer, 2007, pp. 758–768.
- [111] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Background modeling and subtraction by codebook construction," in *Image Processing, 2004. ICIP'04. 2004 International Conference on*, vol. 5. IEEE, 2004, pp. 3061–3064.
- [112] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *BMVC 2008-19th British Machine Vision Conference*. British Machine Vision Association, 2008, pp. 275–1.
- [113] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence*, vol. 97, no. 1, pp. 273–324, 1997.
- [114] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.
- [115] M. G. Krishna, V. M. Aradhya, M. Ravishankar, and D. R. Babu, "Lopp : locality preserving projections for moving object detection," *Procedia Technology*, vol. 4, pp. 624–628, 2012.
- [116] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.
- [117] S. lab, "Hmdb51, a large video database for human motion recognition," <http://serre-lab.clps.brown.edu/resources/HMDB/index.htm>, 2011.
- [118] I. Laptev, "Hollywood & hollywood-2 : human actions datasets," <http://www.irisa.fr/vista/Equipe/People/Laptev/download.html>, 2008.
- [119] I. Laptev and B. Caputo, "Kth recognition of human actions," <http://www.nada.kth.se/cvap/actions/>, 2004.
- [120] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.
- [121] I. Laptev, B. Caputo, C. Schüldt, and T. Lindeberg, "Local velocity-adapted motion events for spatio-temporal recognition," *Computer Vision and Image Understanding*, vol. 108, no. 3, pp. 207–229, 2007.
- [122] I. Laptev and T. Lindeberg, "Velocity adaptation of space-time interest points," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 1. IEEE, 2004, pp. 52–56.

Bibliographie

- [123] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.
- [124] I. Laptev and P. Pérez, “Retrieving actions in movies,” in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.
- [125] B. Lee and M. Hedley, “Background estimation for video surveillance,” *IVCNZ02*, pp. 315–320, 2002.
- [126] Y. Lee, Y. Lin, and G. Wahba, “Multicategory support vector machines : Theory and application to the classification of microarray data and satellite radiance data,” *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 67–81, 2004.
- [127] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, “Statistical modeling of complex backgrounds for foreground object detection,” *Image Processing, IEEE Transactions on*, vol. 13, no. 11, pp. 1459–1472, 2004.
- [128] X. Li, W. Hu, Z. Zhang, and X. Zhang, “Robust foreground segmentation based on two effective background models,” in *Proceedings of the 1st ACM international conference on Multimedia information retrieval*. ACM, 2008, pp. 223–228.
- [129] H.-H. Lin, T.-L. Liu, and J.-H. Chuang, “A probabilistic svm approach for background scene initialization,” in *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 3. IEEE, 2002, pp. 893–896.
- [130] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, “Moving target classification and tracking from real-time video,” in *Applications of Computer Vision, 1998. WACV’98. Proceedings., Fourth IEEE Workshop on*. IEEE, 1998, pp. 8–14.
- [131] J. Liu, J. Luo, and M. Shah, “Recognizing realistic actions from videos “in the wild”,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1996–2003.
- [132] J. Liu and M. Shah, “Learning human actions via information maximization,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.
- [133] R. Lublinerman, N. Özay, D. Zarpalas, and O. Camps, “Activity recognition from silhouettes using linear systems and model (in) validation techniques,” in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 1. IEEE, 2006, pp. 347–350.
- [134] R. M. Luque, E. Domínguez, E. J. Palomo, and J. Muñoz, “A neural network approach for video object segmentation in traffic surveillance,” in *Image Analysis and Recognition*. Springer, 2008, pp. 151–158.
- [135] R. M. Luque, D. Lopez-Rodriguez, E. Mérida-Casermeiro, and E. J. Palomo, “Video object segmentation with multivalued neural networks,” in *Hybrid Intelligent Systems, 2008. HIS’08. Eighth International Conference on*. IEEE, 2008, pp. 613–618.

-
- [136] L. Maddalena and A. Petrosino, “A self-organizing approach to background subtraction for visual surveillance applications,” *Image Processing, IEEE Transactions on*, vol. 17, no. 7, pp. 1168–1177, 2008.
- [137] ———, “The sobs algorithm : what are the limits ?” in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012, pp. 21–26.
- [138] C. Marghes, T. Bouwmans, and R. Vasiu, “Background modeling and foreground detection via a reconstructive and discriminative subspace learning approach,” in *International Conference on Image Processing, Computer Vision, and Pattern Recognition, IPCV*, 2012.
- [139] O. Masoud and N. Papanikolopoulos, “A method for human action recognition,” *Image and Vision Computing*, vol. 21, no. 8, pp. 729–743, 2003.
- [140] N. J. McFarlane and C. P. Schofield, “Segmentation and tracking of piglets in images,” *Machine vision and applications*, vol. 8, no. 3, pp. 187–193, 1995.
- [141] G. McLachlan, *Discriminant analysis and statistical pattern recognition*. John Wiley & Sons, 2004, vol. 544.
- [142] H. Meng, N. Pears, and C. Bailey, “A human action recognition system for embedded computer vision application,” in *Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on*. IEEE, 2007, pp. 1–6.
- [143] S. J. Messick and R. P. Abelson, “The additive constant problem in multidimensional scaling,” *Psychometrika*, vol. 21, no. 1, pp. 1–15, 1956.
- [144] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, “Machine learning, neural and statistical classification,” 1994.
- [145] T. B. Moeslund, A. Hilton, and V. Krüger, “A survey of advances in vision-based human motion capture and analysis,” *Computer vision and image understanding*, vol. 104, no. 2, pp. 90–126, 2006.
- [146] M. Molinier, T. Häme, and H. Ahola, “3d-connected components analysis for traffic monitoring in image sequences acquired from a helicopter,” in *Image Analysis*. Springer, 2005, pp. 141–150.
- [147] D. Mukherjee and Q. JonathanWu, “Real-timevideosegmentation using student’s mixture model,” *Procedia Computer Science*, vol. 10, pp. 153–160, 2012.
- [148] R. E. Neapolitan *et al.*, *Learning bayesian networks*. Prentice Hall Upper Saddle River, 2004, vol. 38.
- [149] J. C. Niebles, H. Wang, and L. Fei-Fei, “Unsupervised learning of human action categories using spatial-temporal words,” *International journal of computer vision*, vol. 79, no. 3, pp. 299–318, 2008.
- [150] S. Nowozin, G. Bakir, and K. Tsuda, “Discriminative subsequence mining for action classification,” in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

Bibliographie

- [151] U. of Central Florida, “Ucf datasets,” <http://www.cs.ucf.edu>, 2008.
- [152] U. of Surrey and CERTH-ITI, “i3dpost multi-view dataset,” <http://kahlan.eps.surrey.ac.uk/i3dpostaction/>, 2009.
- [153] T. I. L. of the University of Modena and R. Emilia, “Visor : Video surveillance online repository for annotation retrieval,” <http://www.openvisor.org/index.asp>, 2005.
- [154] T. Ogata, W. Christmas, J. Kittler, and S. Ishikawa, “Improving human activity detection by combining multi-dimensional motion descriptors with boosting,” in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 1. IEEE, 2006, pp. 295–298.
- [155] A. Oikonomopoulos, I. Patras, and M. Pantic, “Spatiotemporal salient points for visual recognition of human actions,” *Systems, Man, and Cybernetics, Part B : Cybernetics, IEEE Transactions on*, vol. 36, no. 3, pp. 710–719, 2005.
- [156] N. M. Oliver, B. Rosario, and A. P. Pentland, “A bayesian computer vision system for modeling human interactions,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 831–843, 2000.
- [157] E. J. Palomo, E. Domínguez, R. M. Luque, and J. Muñoz, “Image hierarchical segmentation based on a ghsom,” in *Neural Information Processing*. Springer, 2009, pp. 743–750.
- [158] K. Pearson, “Principal components analysis,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 6, no. 2, p. 559, 1901.
- [159] A. Pentland, “Smart rooms, smart clothes,” in *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, vol. 2. IEEE, 1998, pp. 949–953.
- [160] R. Poppe and M. Poel, “Discriminative human action recognition using pairwise csp classifiers,” in *Automatic Face & Gesture Recognition, 2008. FG’08. 8th IEEE International Conference on*. IEEE, 2008, pp. 1–6.
- [161] F. Porikli and C. Wren, “Change detection by frequency decomposition : Waveback,” in *Proc. of Workshop on Image Analysis for Multimedia Interactive Services*, 2005.
- [162] B. Qin, J. Wang, J. Gao, T. Pang, and F. Su, “A traffic video background extraction algorithm based on image content sensitivity,” in *Advances in Swarm Intelligence*. Springer, 2010, pp. 603–610.
- [163] J. R. Quinlan, “Induction of decision trees,” *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.
- [164] C. Rao and M. Shah, “View-invariance in action recognition,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 2. IEEE, 2001, pp. II–316.
- [165] K. Rapantzikos, Y. Avrithis, and S. Kollias, “Dense saliency-based spatiotemporal feature points for action recognition,” in *Computer Vision and Pattern*

-
- Recognition, 2009. CVPR 2009. IEEE Conference on.* IEEE, 2009, pp. 1454–1461.
- [166] M. D. Rodriguez, J. Ahmed, and M. Shah, “Action mach a spatio-temporal maximum average correlation height filter for action recognition,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on.* IEEE, 2008, pp. 1–8.
- [167] R. Rosales, “Recognition of human action using moment-based features,” BU CS TR, Tech. Rep., 1998.
- [168] F. Rosenblatt, “The perceptron : a probabilistic model for information storage and organization in the brain.” *Psychological review*, vol. 65, no. 6, p. 386, 1958.
- [169] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [170] Y. Rui, T. S. Huang, and S.-F. Chang, “Image retrieval : Current techniques, promising directions, and open issues,” *Journal of visual communication and image representation*, vol. 10, no. 1, pp. 39–62, 1999.
- [171] D. E. Rumelhart, J. L. McClelland, P. R. Group *et al.*, “Parallel distributed processing, vols 1 and 2,” *Cambridge, MA : The MIT Press*, 1986.
- [172] S. Russell and P. Norvig, “Artificial intelligence : a modern approach,” 1995.
- [173] M. S. Ryoo and J. K. Aggarwal, “Spatio-temporal relationship match : Video structure comparison for recognition of complex human activities,” in *Computer vision, 2009 ieee 12th international conference on.* IEEE, 2009, pp. 1593–1600.
- [174] J. W. Sammon, “A nonlinear mapping for data structure analysis,” *IEEE Transactions on computers*, no. 5, pp. 401–409, 1969.
- [175] S. Sarkar, P. J. Phillips, Z. Liu, I. R. Vega, P. Grother, and K. W. Bowyer, “The humanid gait challenge problem : Data sets, performance, and analysis,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 2, pp. 162–177, 2005.
- [176] S. Savarese, A. DelPozo, J. C. Niebles, and L. Fei-Fei, “Spatial-temporal correlatons for unsupervised action classification,” in *Motion and video Computing, 2008. WMVC 2008. IEEE Workshop on.* IEEE, 2008, pp. 1–8.
- [177] K. Schindler and L. Van Gool, “Action snippets : How many frames does human action recognition require?” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on.* IEEE, 2008, pp. 1–8.
- [178] B. Schölkopf, C. Burges, and V. Vapnik, “Incorporating invariances in support vector learning machines,” in *Artificial Neural Networks—ICANN 96.* Springer, 1996, pp. 47–52.
- [179] B. Schölkopf and A. J. Smola, *Learning with kernels : Support vector machines, regularization, optimization, and beyond.* MIT press, 2002.

Bibliographie

- [180] C. Schüldt, I. Laptev, and B. Caputo, “Recognizing human actions : a local svm approach,” in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 3. IEEE, 2004, pp. 32–36.
- [181] P. Scovanner, S. Ali, and M. Shah, “A 3-dimensional sift descriptor and its application to action recognition,” in *Proceedings of the 15th international conference on Multimedia*. ACM, 2007, pp. 357–360.
- [182] A. W. Senior, Y. Tian, and M. Lu, “Interactive motion analysis for video surveillance and long term scene monitoring,” in *Computer Vision–ACCV 2010 Workshops*. Springer, 2011, pp. 164–174.
- [183] G. Shakhnarovich, P. Indyk, and T. Darrell, *Nearest-neighbor methods in learning and vision : theory and practice*, 2006.
- [184] E. Shechtman and M. Irani, “Space-time behavior based correlation,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 405–412.
- [185] Y. Sheikh and M. Shah, “Bayesian object detection in dynamic scenes,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 74–79.
- [186] Y. Sheikh, M. Sheikh, and M. Shah, “Exploring the space of a human action,” in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1. IEEE, 2005, pp. 144–149.
- [187] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in *Advances in Neural Information Processing Systems*, 2014, pp. 568–576.
- [188] S. Srinivasan and K. L. Boyer, “Head pose estimation using view based eigenspaces,” in *null*. IEEE, 2002, p. 40302.
- [189] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2. IEEE, 1999.
- [190] R. S. Sutton and A. G. Barto, *Reinforcement learning : An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.
- [191] A. Tavakkoli, M. Nicolescu, and G. Bebis, “A novelty detection approach for foreground region detection in videos with quasi-stationary backgrounds,” in *Advances in Visual Computing*. Springer, 2006, pp. 40–49.
- [192] J. B. Tenenbaum, V. De Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
- [193] H. Tezuka and T. Nishitani, “A precise and stable foreground segmentation using fine-to-coarse approach in transform domain,” in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. IEEE, 2008, pp. 2732–2735.

-
- [194] Y.-L. Tian and A. Hampapur, “Robust salient motion detection with complex background for real-time video surveillance,” in *Application of Computer Vision, 2005. WACV/MOTIONS’05 Volume 1. Seventh IEEE Workshops on*, vol. 2. IEEE, 2005, pp. 30–35.
- [195] Y. Tian, A. Senior, and M. Lu, “Robust and efficient foreground analysis in complex surveillance videos,” *Machine Vision and Applications*, vol. 23, no. 5, pp. 967–983, 2012.
- [196] W. S. Torgerson, “Multidimensional scaling : I. theory and method,” *Psychometrika*, vol. 17, no. 4, pp. 401–419, 1952.
- [197] R. Touati and M. Mignotte, “Mds-based multi-axial dimensionality reduction model for human action recognition,” in *Computer and Robot Vision (CRV), 2014 Canadian Conference on*. IEEE, 2014, pp. 262–267.
- [198] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, “Wallflower : Principles and practice of background maintenance,” in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 1. IEEE, 1999, pp. 255–261.
- [199] M. Turk and A. Pentland, “Eigenfaces for recognition,” *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
- [200] K. University, “Muhavi : Multicamera human action video data,” <http://dipersec.king.ac.uk/MuHAVi-MAS/>, 2010.
- [201] S. University, “Olympic sports dataset,” <http://vision.stanford.edu/Datasets/OlympicSports/>, 2010.
- [202] V. N. Vapnik and V. Vapnik, *Statistical learning theory*. Wiley New York, 1998, vol. 1.
- [203] N. Vaswani, A. K. Roy-Chowdhury, and R. Chellappa, “" shape activity" : a continuous-state hmm for moving/deforming shapes with application to abnormal activity detection,” *Image Processing, IEEE Transactions on*, vol. 14, no. 10, pp. 1603–1616, 2005.
- [204] A. Veeraraghavan, R. Chellappa, and A. K. Roy-Chowdhury, “The function space of an activity,” in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1. IEEE, 2006, pp. 959–968.
- [205] J. Wang, G. Bebis, and R. Miller, “Robust video-based surveillance by integrating target detection with tracking,” in *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW’06. Conference on*. IEEE, 2006, pp. 137–137.
- [206] L. Wang and D. Suter, “Recognizing human activities from silhouettes : Motion subspace and factorial discriminative graphical model,” in *Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on*. IEEE, 2007, pp. 1–8.
- [207] —, “Visual learning and recognition of sequential data manifolds with applications to human movement analysis,” *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 153–172, 2008.

- [208] Q. Wang and K. L. Boyer, "Feature learning by multidimensional scaling and its applications in object recognition," in *Graphics, Patterns and Images (SIBGRAPI), 2013 26th SIBGRAPI-Conference on*. IEEE, 2013, pp. 8–15.
- [209] T. Wang, G. Chen, and H. Zhou, "A novel background modeling approach for accurate and real-time motion segmentation," in *Signal Processing, 2006 8th International Conference on*, vol. 2. IEEE, 2006.
- [210] Y. Wang, P. Sabzmejdani, and G. Mori, "Semi-latent dirichlet allocation : A hierarchical model for human action recognition," in *Human Motion–Understanding, Modeling, Capture and Animation*. Springer, 2007, pp. 240–254.
- [211] K. Q. Weinberger, F. Sha, and L. K. Saul, "Learning a kernel matrix for nonlinear dimensionality reduction," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 106.
- [212] J. Weston and C. Watkins, "Multi-class support vector machines," Citeseer, Tech. Rep., 1998.
- [213] S.-F. Wong and R. Cipolla, "Extracting spatiotemporal interest points using global information," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.
- [214] S.-F. Wong, T.-K. Kim, and R. Cipolla, "Learning motion categories using both semantic and structural information," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–6.
- [215] C. R. Wren and F. Porikli, "Waviz : Spectral similarity for object detection," in *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2005, pp. 55–61.
- [216] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder : Real-time tracking of the human body," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 780–785, 1997.
- [217] J. Wu and M. M. Trivedi, "A two-stage head pose estimation framework and evaluation," *Pattern Recognition*, vol. 41, no. 3, pp. 1138–1158, 2008.
- [218] M. Xiao, C. Han, and X. Kang, "A background reconstruction for dynamic scenes," in *Information Fusion, 2006 9th International Conference on*. IEEE, 2006, pp. 1–7.
- [219] D. Xiuman, S. Guoxia, and Y. Tao, "Moving target detection based on genetic k-means algorithm," in *Communication Technology (ICCT), 2011 IEEE 13th International Conference on*. IEEE, 2011, pp. 819–822.
- [220] Z. Xu, I. Y.-H. Gu, and P. Shi, "Recursive error-compensated dynamic eigen-background learning and adaptive background subtraction in video," *Optical Engineering*, vol. 47, no. 5, pp. 057 001–057 001, 2008.
- [221] J. Y. Yam and T. W. Chow, "Feedforward networks training speed enhancement by optimal initialization of the synaptic coefficients," *Neural Networks, IEEE Transactions on*, vol. 12, no. 2, pp. 430–434, 2001.

-
- [222] A. Yamamoto and Y. Iwai, “Real-time object detection with adaptive background model and margined sign correlation,” in *Computer Vision–ACCV 2009*. Springer, 2010, pp. 65–74.
- [223] J. Yamato, J. Ohya, and K. Ishii, “Recognizing human action in time-sequential images using hidden markov model,” in *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR’92., 1992 IEEE Computer Society Conference on*. IEEE, 1992, pp. 379–385.
- [224] M. Yamazaki, G. Xu, and Y.-W. Chen, “Detection of moving objects by independent component analysis,” in *Computer Vision–ACCV 2006*. Springer, 2006, pp. 467–478.
- [225] X. Yang, H. Fu, H. Zha, and J. Barlow, “Semi-supervised nonlinear dimensionality reduction,” in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 1065–1072.
- [226] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, “Describing videos by exploiting temporal structure,” *stat*, vol. 1050, p. 25, 2015.
- [227] A. Yilmaz and M. Shah, “Actions sketch : A novel action representation,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 984–989.
- [228] A. Zaharescu and M. Jamieson, “Multi-scale multi-feature codebook-based background subtraction,” in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1753–1760.
- [229] L. Zelnik-Manor and M. Irani, “Event-based video analysis,,” Jerusalem, Israel, Israel, Tech. Rep., 2001.
- [230] —, “Event-based analysis of video,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 2. IEEE, 2001, pp. II–123.
- [231] D. Zhang, Z.-H. Zhou, and S. Chen, “Semi-supervised dimensionality reduction.” in *SDM*. SIAM, 2007, pp. 629–634.
- [232] Z.-y. Zhang and H.-y. Zha, “Principal manifolds and nonlinear dimensionality reduction via tangent space alignment,” *Journal of Shanghai University (English Edition)*, vol. 8, no. 4, pp. 406–424, 2004.
- [233] Z. Zhang, “Mining relational data from text : From strictly supervised to weakly supervised learning,” *Information Systems*, vol. 33, no. 3, pp. 300–314, 2008.
- [234] Z. Zhang, Y. Hu, S. Chan, and L.-T. Chia, “Motion context : A new representation for human action recognition,” *Computer Vision–ECCV 2008*, pp. 817–829, 2008.
- [235] J. Zheng, Y. Wang, N. Nihan, and M. Hallenbeck, “Extracting roadway background image : Mode-based approach,” *Transportation Research Record : Journal of the Transportation Research Board*, no. 1944, pp. 82–88, 2006.
- [236] H. Zhong, J. Shi, and M. Visontai, “Detecting unusual activity in video,” in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2. IEEE, 2004, pp. II–819.