# Reconnaissance robuste d'activités humaines par vision

Geoffrey Vaquette

**Université Pierre et Marie Curie**

École doctorale **Science Mécaniques, Acoustique, Electronique et Robotique de Paris (ED 391)**

Unité de recherche **Laboratoire Vision et Ingénierie des Contenus**

Thèse présentée par **Geoffrey Vaquette**

Soutenue le **14 février 2018**

En vue de l'obtention du grade de docteur de l'Université Pierre et Marie Curie

Discipline **Informatique**

Titre de la thèse

# Reconnaissance robuste d'activités humaines par vision

**Thèse dirigée par**

| | | |
|---|---|---|
| Catherine Achard | directeur |
| Laurent Lucat | co-encadrant |

**Composition du jury**

| | | |
|---|---|---|
| *Rapporteurs* | Antoine Manzanera | professeur à l'ENSTA-ParisTech |
| | Mounîm A. El Yacoubi | professeur au l'Institut Mines Telecom |
| *Examinateurs* | Mohamed Daoudi | professeur au Telecom Lille |
| | Jean-Luc Zarader | professeur à l'UPMC Paris VI ISIR UMR 7222 |
| *Directeurs de thèse* | Catherine Achard | mcf à l'UPMC |
| | Laurent Lucat | CEA/LIST/DIASI |

L'Université Pierre et Marie Curie n'entend donner aucune approbation ni improbation aux opinions émises dans les thèses : ces opinions devront être considérées comme propres à leurs auteurs.

**Mots clés :** détection d'activités, fusion d'informations, transformée de hough fortement optimisée (doht), jeu de données

**Keywords:** activity detection, information fusion, deeply optimized hough transform (doht), database

Cette thèse a été préparée au

**Laboratoire Vision et Ingénierie des Contenus**
CEA Saclay
Centre d'intégration Nano-INNOV
Avenue de la Vauve
Bâtiment 861
91120 Palaiseau
France

Site `www.kalisteo.eu`


Version intermédiaire en date du 30 janvier 2018

**Reconnaissance robuste d'activités humaines par vision**

**Résumé**

Cette thèse porte sur la segmentation supervisée d'un flux vidéo en fragments correspondant à des activités de la vie quotidienne. En différenciant geste, action et activité, cette thèse s'intéresse aux activités à haut niveau sémantique telles que "Cuisiner" ou "Prendre son repas" par opposition à des actions comme "Découper un aliment".
Pour cela, elle s'appuie sur l'algorithme DOHT (Deeply Optimized Hough Transform), une méthode de l'état de l'art utilisant un paradigme de vote (par transformée de Hough). Dans un premier temps, nous adaptons l'algorithme DOHT pour fusionner les informations en provenance de différents capteurs à trois niveaux différents de l'algorithme. Nous analysons l'effet de ces trois niveaux de fusion et montrons son efficacité par une évaluation sur une base de données composée d'actions de la vie quotidienne. Ensuite, une étude des jeux de données existant est menée. Constatant le manque de vidéos adaptées à la segmentation et classification (détection) d'activités à haut niveau sémantique, une nouvelle base de données est proposée. Enregistrée dans un environnement réaliste et dans des conditions au plus proche de l'application finale, elle contient des vidéos longues et non découpées adaptées à un contexte de détection. Dans un dernier temps, nous proposons une approche hiérarchique à partir d'algorithmes DOHT pour reconnaître les activités à haut niveau sémantique. Cette approche à deux niveau décompose le problème en une détection non-supervisée d'actions pour ensuite détecter les activités désirées.

**Mots clés :** détection d'activités, fusion d'informations, transformée de hough fortement optimisée (doht), jeu de données

**Abstract**

This thesis focuses on supervised activity segmentation from video streams within application context of smart homes. Three semantic levels are defined, namely gesture, action and activity, this thesis focuses mainly on the latter. Based on the Deeply Optimized Hough Transform paridigm, three fusion levels are introduced in order to benefit from various modalities. A review of existing action based datasets is presented and the lack of activity detection oriented database is noticed. Then, a new dataset is introduced. It is composed of unsegmented long time range daily activities and has been recorded in a realistic environment. Finaly, a hierarchical activity detection method is proposed aiming to detect high level activities from unsupervised action detection.

**Keywords:** activity detection, information fusion, deeply optimized hough transform (doht), database

**Laboratoire Vision et Ingénierie des Contenus**
CEA Saclay – Centre d'intégration Nano-INNOV – Avenue de la Vauve – Bâtiment 861 – 91120 Palaiseau – France

# Sommaire

# Liste des tableaux

# Table des figures

Chapitre 1

# Introduction

Les thématiques de recherche autour de l'apprentissage automatique connaissent actuellement un intérêt considérable. Il s'agit d'un domaine passionnant visant à rendre un système informatisé capable de comprendre, d'interpréter et de réagir aux informations en provenance du monde réel. Ces informations peuvent être de nature textuelle, sonore, visuelle, mais aussi biologique, physique, *etc.*

Dans le cas plus spécifique du numérique, les applications sont multiples allant de l'indexation automatique de documents à l'interprétation haut niveau de leur contenu en passant par la proposition de documents similaires, la segmentation d'images, l'authentification, *etc.*

Avec l'omniprésence des systèmes informatisés dans nos vies, ce domaine intéresse non seulement les laboratoires de recherche publique, mais aussi les multinationales qui investissent abondamment pour proposer des applications toujours plus innovantes. En 2015, le géant américain *Facebook* ouvre à Paris un centre de recherche autour de l'intelligence artificielle, s'implantant ainsi mondialement dans la recherche du domaine. Entre 2015 et 2017, *Google* marque la communauté scientifique avec *AlphaGo*, un programme informatique capable de gagner des parties de Go contre les meilleurs joueurs mondiaux. Concevoir un algorithme capable d'égaler l'humain à ce jeu était considéré jusqu'alors comme un problème extrêmement difficile au vu du nombre très élevé de configurations possibles.

La vision par ordinateur est un des domaines de ce thème de recherche et ouvre un champ d'application très varié. Il s'agit de modéliser, de comprendre et d'interpréter automatiquement des informations visuelles en vue de les rendre directement exploitables par un être humain. Le développement de telles méthodes est favorisé par l'abondance de photos, de vidéos et plus généralement de contenu numérique mis en ligne chaque jour sur les réseaux sociaux.

Sur internet, d'après le géant américain *Google*, plusieurs centaines d'heures de vidéos sont mises en ligne chaque minute et plus d'un milliard d'heures de vidéos sont visionnées chaque jour sur *Youtube*. Devant une telle quantité de données, l'en-

treprise doit faire un tri personnalisé pour proposer à ses utilisateurs des contenus qui l'intéressent, afin de l'inciter à rester sur la plateforme. Cette sélection se fait à l'aide d'algorithmes d'apprentissage automatique prenant en compte le profil de l'utilisateur, ses habitudes, les pages internet qu'il consulte, mais aussi les comportements des internautes visionnant des contenus similaires.

Dans le domaine image, pour chacune des 350 millions de photos publiées chaque jour sur *Facebook*, le réseau social génère automatiquement, à partir de l'image elle-même, une description textuelle destinée aux utilisateurs malvoyants. Ces descriptions évoquent d'une part les personnes présentes dans l'image, mais également l'action qui s'y déroule probablement.

Pour l'analyse de vidéos, on peut citer tout d'abord les applications d'indexation automatique de contenus sur des plateformes dédiées (*Youtube*, *Dailymotion*, *Netflix*, *etc.*) afin de faciliter la recherche d'éléments précis. L'objectif est alors d'extraire d'un flux vidéo un maximum d'informations pouvant intéresser un utilisateur et d'associer ces informations à des mots-clefs qu'il utilisera pour sa requête.

Le développement d'algorithmes autour de la vision par ordinateur a aussi fait avancer de manière considérable les applications autour de la voiture autonome. Cinq états des États-unis d'Amérique autorisent d'ailleurs déjà leur circulation, avec un ingénieur à bord. A Singapour, des taxis autonomes sont en circulation dans une partie limitée de la ville. Ces véhicules impliquent de multiples défis d'apprentissage automatique à partir de l'image, que ce soit pour détecter des obstacles, des piétons, reconnaître les panneaux de signalisation, estimer les trajectoires des autres véhicules, *etc.*.

La reconnaissance d'actions et d'activités humaine n'échappe pas à cet engouement et vise une grande variété d'applications.

En plus des applications d'interprétation du contenu citées précédemment, d'autres applications s'orientent vers l'interaction homme-machine. Principalement développées pour les jeux vidéos, ces applications nécessitent l'interprétation des gestes réalisés par la personne devant une caméra. Par exemple, *Microsoft* a mis sur le marché en novembre 2010 la Kinect capable d'extraire par apprentissage statistique les positions des articulations de la personne en vue de proposer une nouvelle façon de jouer. C'est ici les mouvements même de la personne qui sont au cœur de l'expérience de jeu.

Citons également les applications liées à la vidéo protection ou la vidéo surveillance. Malgré une représentation divisée dans l'imaginaire collectif, soulevant des questions éthiques liées au respect des libertés individuelles, le nombre de caméras de surveillance installées dans notre environnement ne cesse de croître. Cela génère une abondance de données impossible à traiter humainement. Une interprétation automatique s'impose, que ce soit en vue de faciliter le travail d'un agent de surveillance ou pour alerter automatiquement sur un comportement dangereux.

Chez les particuliers, l'analyse du comportement humain ouvre la voie des applications orientées vers les maisons intelligentes. Au-delà de la simple domotique, il s'agit de faire réagir l'habitation aux différentes situations qu'elle peut observer. Il peut également s'agir d'avertir des situations dangereuses comme un enfant jouant sans surveillance

près d'un four allumé.

L'analyse du comportement peut aussi s'orienter vers la santé et assister les personnes en perte d'autonomie. Dans le cas des personnes âgées, ces technologies peuvent permettre de prolonger leur autonomie en créant des environnements réagissant à leurs besoins. Elles peuvent également soulager le personnel aidant, en prévenant des situations à risque, en alertant des situations de détresse telles que les chutes ou encore en détectant des signes de comportements anormaux comme une non-alimentation ou une répétition excessive d'une activité. Ces méthodes, utilisant des caméras, sont moins invasives pour les patients concernés que d'autres méthodes nécessitant le port d'un équipement comme des accéléromètres et permet d'éviter le risque de retrait ou d'oubli de l'équipement par le patient.

## Positionnement de la thèse

### Contexte

Cette thèse s'inscrit dans un contexte d'intérêt croissant pour les technologies d'interprétation du comportement humain. Plus spécifiquement, nous nous concentrons sur l'objectif à long terme d'algorithmes autour de l'habitation intelligente. Il s'agit non seulement de reconnaître, mais également de **localiser temporellement** différents comportements humains. Cette notion de détection (ou de localisation temporelle) est un point important dans un contexte où la grande majorité des méthodes existantes s'intéresse à de la **classification** de vidéo plus qu'à de la détection temporelle d'actions ou d'activités humaines. Nous différencions donc dans cette thèse la classification, consistant à reconnaître le comportement présent dans une vidéo complète, de la détection consistant à localiser temporellement les différentes classes recherchées.

Par ailleurs, dans un tel cadre applicatif, nous définissons trois niveaux pour caractériser le comportement humains : *les gestes*, *les actions* et *les activités*.

**Les gestes** d'une durée très courte n'ont pas nécessairement de réalité sémantique. Il peut s'agir, par exemple de *"Lever un bras"* ou encore *"Ouvrir la bouche"*.

**Les actions** composées de plusieurs gestes ont une sémantique bas niveau telle que *"Ranger un objet dans un tiroir"*, *"Découper un aliment"*, *etc.* Bien que leurs durées soient plus longues que celles des gestes, elles restent cependant limitées à quelques secondes. Pour ces actions (comme pour les gestes), l'aspect temporel revêt un caractère particulièrement important : lorsqu'on ouvre un tiroir, le mouvement débute bras tendu et le bras se rapproche ensuite du corps ;

**Les activités** composées de plusieurs actions ont une sémantique de plus haut niveau telle que *"Conduire un véhicule"*, *"Faire ses achats"*, *"Cuisiner"*, *"Prendre son déjeuner"*, *etc.* Les activités sont composées de plusieurs actions qui se succèdent dans le temps dans un ordre plus ou moins précis. Par exemple lorsque l'on prend son repas, on ne boit pas toujours au même moment, certains aliments ont besoin d'être coupés et d'autres non, certaines personnes se resservent, *etc.*

On souhaite dans cette thèse s'intéresser principalement à des classes de la troisième catégorie : **les activités humaines à haut niveau sémantique**. Ce haut niveau sémantique implique une grande variété dans les différentes façons possibles de réaliser ces activités. Il en résulte une variabilité intra-classe élevée, à prendre en compte dans le développement des méthodes de détection.

### Problématiques à considérer

Dans un tel contexte applicatif, plusieurs problématiques sont soulevées. Tout d'abord, dans un environnement réel (en dehors des conditions idéales de laboratoire), l'agencement de la scène ne peut pas être contrôlé. Cela implique une forte probabilité de présence d'occultations dans la scène dues au mobilier positionné dans la pièce par exemple. Les méthodes de détection d'activités doivent donc être robustes à d'éventuelles occultations partielles, voire totales, des personnes réalisant les activités.

Afin de gérer ces inévitables occultations, il est nécessaire d'utiliser plusieurs capteurs de manière à observer la scène sous plusieurs points de vue. Une fusion de tous ces capteurs, qui peuvent être de nature différente, doit donc être prévue. Elle peut être plus ou moins intégrée au sein de l'algorithme. Dans le cadre d'une application réelle, l'indisponibilité temporaire d'un capteur doit aussi être traitée afin de ne pas trop affecter les performances de détection de la méthode.

Comme évoqué précédemment, les méthodes proposées doivent permettre de regrouper sous un même nom des activités observant une variabilité intra-classe élevée : variabilités dans la façon de réaliser l'activité, variabilité dans la vitesse d'exécution, variabilité dans les points de vues ou encore dans la corpulence des personnes, *etc.*

Ces méthodes doivent d'autre part être compatibles avec des activités de durées très différentes selon la classe : dans le cadre de l'habitat intelligent, l'activité *"Prendre son déjeuner"* est bien plus longue que l'activité *"Débarasser la table"* mais bien plus courte que l'activité *"Dormir"*.

Enfin, une autre contrainte importante réside dans les temps de calcul et dans la latence de la méthode. En effet, même si une latence importante (quelques secondes) peut être tolérée pour certaines applications, il est impératif que le système fonctionne en temps réel de manière à pouvoir traiter des flux vidéo de manière continue.

### Contributions

Afin de répondre à ces problématiques, cette thèse propose trois contributions. Tout d'abord, nous proposons l'adaptation d'un algorithme de détection d'actions de la littérature à la prise en compte de données en provenance de différents capteurs ou de différentes modalités (images, squelettes, cartes de profondeur, *etc.*), robuste à une perte temporaire d'un flux d'information.

Ensuite, constatant le manque de données adaptées à la détection d'activités à haut niveau sémantique, nous avons créé un scénario permettant de mettre en œuvre des activités réalistes pour notre contexte applicatif, à forte variabilité intra-classe. Ce nouveau jeu de données, enregistré dans la plateforme de recherche d'habitat intelligent

MobileMii du CEA, représente plusieurs heures de vidéo acquises par plusieurs capteurs. Il met en avant les défis cités précédemment (occultations, différences de durées des activités, *etc.*) et a été mis à la disposition de la communauté.

Pour finir, nous avons mis en place une approche hiérarchique en vue de détecter et reconnaître les activités humaines au sein d'une vidéo. Cette approche repose sur l'algorithme d'apprentissage précédant exploité à deux niveau sémantiques différents : un premier apprentissage semi-supervisé est mis en place pour détecter des actions élémentaires ; puis un second apprentissage supervisé considère ces actions en entrée en vue de reconnaître les activités en elle-même.

Ces travaux ont menés aux publications suivantes : [1]–[3]

## Plan du Manuscrit

Ce manuscrit est structuré selon les contributions apportées. Le chapitre 2 présente un état de l'art des différentes méthodes existantes dans le domaine de l'analyse du comportement. Ensuite, le chapitre 3 introduit l'algorithme sur lequel reposent les travaux de cette thèse et présente les adaptations mises en place pour la fusion d'informations dans ce paradigme de détection. Le chapitre 4 introduit le jeu de données *DAHLIA (DAily Home LIfe Activities)* que nous avons acquis, après avoir passé en revue les différents jeux existants en reconnaissance d'actions. Enfin, le chapitre 5 présente l'approche hiérarchique que nous proposons pour améliorer la détection d'activités humaines.

Chapitre 2

# Les méthodes de reconnaissance d'actions

## 2.1 Introduction

Dans la littérature, la reconnaissance d'actions est un sujet largement exploré et pour lequel une grande variété de méthodes a été proposée. De la classification de gestes simples à la reconnaissance d'activités plus complexes, il s'agit de représenter, d'analyser puis d'interpréter les mouvements humains en vue d'associer une sémantique au comportement humain observé. Dans ce chapitre, nous introduisons plus rigoureusement les problématiques visées et présentons les principales méthodes de reconnaissance d'actions de l'état de l'art.

### 2.1.1 Définition de l'apprentissage par ordinateur

Avant tout, il s'agit de définir ce qu'est l'apprentissage par ordinateur. D'après une définition proposée par Mitchell dans [4] :

**Définition 1.** *Un ordinateur apprend une tâche $T$ avec la performance $P$ à partir d'une expérience $E$ si ses performances pour la tâche $T$, mesurées par $P$, s'accroissent avec l'expérience $E$.*

Dit autrement, il s'agit pour l'ordinateur, à partir de l'expérience, de modifier son comportement vis-a-vis d'une tâche pour résoudre cette dernière de façon plus satisfaisante à l'avenir [5].

Un des principaux avantages des algorithmes d'apprentissage est de permettre la résolution de tâches trop difficiles à coder par un être humain [6]. Il existe un grand nombre de tâches auxquelles les algorithmes d'apprentissage ont été ou peuvent être appliqués telles que les tâches de *classification*, *régression*, *localisation*, *transcription*, *traduction*, *détection d'anomalie*, *débruitage de signaux*, *etc.* [6]. Cette thèse s'intéresse plus particulièrement aux tâches de *détection* et de *classification* que nous décrivons plus en détail dans la section 2.1.3.

### 2.1.2 Apprentissage non-supervisé, supervisé et par renforcement

On peut distinguer trois grandes catégories d'algorithmes d'apprentissage.

**L'apprentissage non-supervisé** désigne les méthodes pour lesquelles on fournit à l'algorithme un ensemble de données dont on souhaite extraire une structure cachée (comme sa densité ou un partitionnement de l'espace à partir de la similarité entre ses exemples). L'algorithme ne travaille qu'à partir des données elles-même, sans information supplémentaire.

**L'apprentissage supervisé** désigne les méthodes qui, à l'inverse, optimisent leur paramètres à partir d'ensembles de données annotées : chaque exemple est manuellement associé à une *cible* ou *étiquette* ou *classe* considérée comme étant véritable. L'objectif d'un tel algorithme est d'apprendre à prédire la cible associée à chaque nouvel exemple.

**L'apprentissage par renforcement** consiste à faire interagir le système avec un environnement qui offrira des *récompenses* en fonction du comportement du système. Le but d'un tel système est d'optimiser ses paramètres (et donc modifier son comportement) afin de maximiser la valeur des récompenses reçues.

La frontière entre ces trois catégories est parfois floue et beaucoup d'algorithmes supervisés contiennent une étape non-supervisée comme, par exemple, une étape de partitionnement (cf section 2.1.3).

Dans cette thèse, nous nous intéressons principalement à des algorithmes de *détection* et *classification* qui entrent dans la catégorie de l'apprentissage supervisé. Nous explicitons le contexte mathématique associé dans la section suivante.

### 2.1.3 Classification et Détection

De façon générale, le but d'un algorithme **d'apprentissage supervisé** est de **prédire** la **classe** $y_i^* \in \mathcal{Y}$ associée à un **échantillon** $\mathbf{x}_i \in \mathcal{X}$. $\mathcal{Y}$ est l'ensemble des classes (ou concepts) que l'on cherche à reconnaître et $\mathcal{X}$ est l'ensemble des données considérées. Par soucis de lisibilité, nous désignerons $\mathbf{x}_i$ par $\mathbf{x}$ en l'absence d'ambiguïté. Il s'agit alors d'estimer la fonction $h$ donnant $\hat{y}$ sachant $\mathbf{x}$. $\hat{y}$ est l'étiquette estimée par l'algorithme et $h$ est communément appelée *hypothèse*.

Ainsi, avec $\hat{y} = h(\mathbf{x})$ la classe estimée par l'algorithme, on définit un coût $L(\hat{y}, y^*) = L(h(\mathbf{x}), y)$ pénalisant l'erreur de classification. Soit $\mathcal{P}(\mathbf{x}, y)$ la densité inconnue décrivant la distribution de probabilité de $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$, l'objectif de la classification supervisée est alors de minimiser le **risque attendu** $l(h)$ [6] : on cherche $h_0$ telle que :

$$h_0 = \min_{h \in \mathcal{H}} l(h) \qquad \text{avec} \qquad l(h) = \int_{\mathcal{X} \times \mathcal{Y}} L(h(\mathbf{x}), y) \mathcal{P}(\mathbf{x}, y) d\mathbf{x} dy \tag{2.1}$$

Cependant, $\mathcal{P}(\mathbf{x}, y)$ étant inconnu, il est impossible d'optimiser directement cette expression. Considérons un sous-ensemble $\mathcal{X}_E \subset \mathcal{X}$ dont on connaît pour chaque $\mathbf{x} \in \mathcal{X}_E$ la classe associée $y^* \in \mathcal{Y}$. L'ensemble $\mathcal{E} = \{(\mathbf{x}_i, y_i^*) \mid i \in [\![1, n]\!]\}$, avec $n = \mathbf{card}(\mathcal{X}_E)$, est

appelé **ensemble d'entraînement**. On définit alors un estimateur du risque attendu, le **risque empirique** :

$$l_{emp}(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(\mathbf{x}_i), y_i^*) \tag{2.2}$$

C'est ce risque qui sera minimisé lors de la phase d'apprentissage en vue d'obtenir un estimateur $h_E$ de la fonction idéale $h_0$.

Notons que lorsque $\mathcal{Y}$ est un ensemble discret, on parle de **classification** (par exemple, $\mathcal{Y} = \{-1, 1\}$ dans le cas d'une classification binaire). Dans le cas contraire, on parle d'un problème de **régression**.

En résumé, concevoir un algorithme de classification consiste à proposer une méthode qui prédit à partir d'une donnée $\mathbf{x}$ une cible $\hat{y}(\mathbf{x})$ que l'on souhaite égale à $y^*$, la cible réellement associée à $\mathbf{x}$. On appelle cette prédiction *classification*.

Dans le contexte présenté précédemment, les algorithmes de classification cherchent à attribuer une classe à une vidéo complète. On suppose alors que la vidéo entière est associée à une classe $y$. Dans un paradigme de détection, l'objectif est de localiser dans une vidéo les instants associés à chaque classe $y \in \mathcal{Y}$ d'actions lorsque ces actions sont présentes.

On cherche donc à partir d'une vidéo $v$, à obtenir un ensemble de segments $\{s_1, \ldots, s_n\}$ ainsi que leur classe. Selon les méthodes mises en place, la segmentation pourra être préalable à la reconnaissance ou les deux étapes pourront être menées de front.

La reconnaissance/détection d'actions ou d'activités peut être décomposée en différentes étapes principales communes à la plupart des méthodes de la littérature. Dans un premier temps, il s'agit de décrire les données brutes (images, cartes de profondeur, signaux acquis), par des descripteurs exploitables par un algorithme de reconnaissance et/ou de détection. Dans un deuxième temps, la classification et/ou la détection est réalisée à partir de ces primitives [7], [8].

## 2.2 L'extraction de primitives

Le but de cette première étape est de détecter et d'extraire des *descripteurs* qui représenteront la vidéo au sein de l'algorithme de reconnaissance d'actions. Puisqu'idéalement les vidéos doivent pouvoir être décrites de la même façon quelles que soient les conditions d'acquisition, ces descripteurs doivent être :

1. invariants face aux rotations, translations et changements d'échelles,
2. invariants face aux transformations affines,
3. invariants face au bruit,
4. robustes aux changements de luminosité,
5. répétitifs.

Selon le capteur utilisé pour acquérir les vidéos, les algorithme peuvent utiliser les images (couleur ou noir et blanc), des images en proche infrarouge, des cartes de

profondeur, ou tout autre signal pouvant contenir une information pertinente pour la reconnaissance d'actions. Nous décrivons dans la suite de cette section l'exploitation d'images, de cartes de profondeur et de positions d'articulations (squelettes) en vue d'une compréhension du comportement humain.

### 2.2.1 Extraction à partir d'images

Parmi les approches image, on peut distinguer les approches globales qui s'intéressent à la personne dans son entité ou les approches locales utilisant par exemple les points d'intérêt.

#### Approches holistiques

La reconnaissance d'actions s'articule autour des mouvements d'une personne dans une vidéo. Une idée relativement intuitive est d'utiliser la silhouette de la personne sur les images capturées par les caméras. Il s'agit de la forme obtenue par projection du corps de la personne sur le plan de l'image. Lorsque la caméra est fixe et que l'arrière plan de la vidéo ne varie pas trop, cette silhouette peut par exemple être acquise à l'aide d'une soustraction de fond sur les images de la vidéo.

En 1992, Yamato, Ohya et Ishii [9] utilisent une silhouette binaire pour reconnaître des gestes de tennis. Celle-ci est décrite en divisant l'image en une grille régulière puis en encodant la proportion de pixels remplis dans chaque cellule. Considérant l'aspect temporel, Bobick et Davis [10] proposent en 2001 de calculer les différences d'une image à l'autre afin de créer, en les accumulant, une image de l'énergie du mouvement (*Motion Energy Image (MEI)*) ainsi qu'une image de l'historique du mouvement (*MHI*), qui conserve une représentation de l'ordonnancement des silhouettes successives. Les *MHI* ont par la suite été étendus par Weinland, Ronfard et Boyer en 2006 dans [11] avec l'introduction d'un volume 3D d'historique du mouvement, créant une représentation de la silhouette indépendante de la vue [12].

En 2013, Chaaraoui, Climent-Pérez et Flórez-Revuelta [12] proposent de réduire la redondance dans la description de la silhouette en encodant son contour plutôt que la forme pleine entièrement. Les auteurs avancent le fait que cette description est moins sensible à un léger changement de vue et moins coûteuse à extraire. A partir de ces contours, les auteurs proposent d'extraire pour chaque action un ensemble de poses clés. Ils obtiennent alors un dictionnaire de poses dont l'agencement temporel décrit chacune des séquences à traiter. L'extraction de ces poses clés est illustrée Figure 2.1.

Remarquons que la description de la silhouette est tributaire des changements de vue, d'une part, mais aussi et surtout d'une occultation partielle de la personne dans l'image. En effet, si une moitié du corps (les jambes par exemple) est cachée par un objet de la scène (table par exemple), alors la projection du corps de la personne sur l'image de la caméra engendrera une forme significativement différente de celle obtenue sans occultation.

Pour s'affranchir de ces limitations, d'autres méthodes préfèrent travailler à partir de points d'intérêts capturés dans l'image.

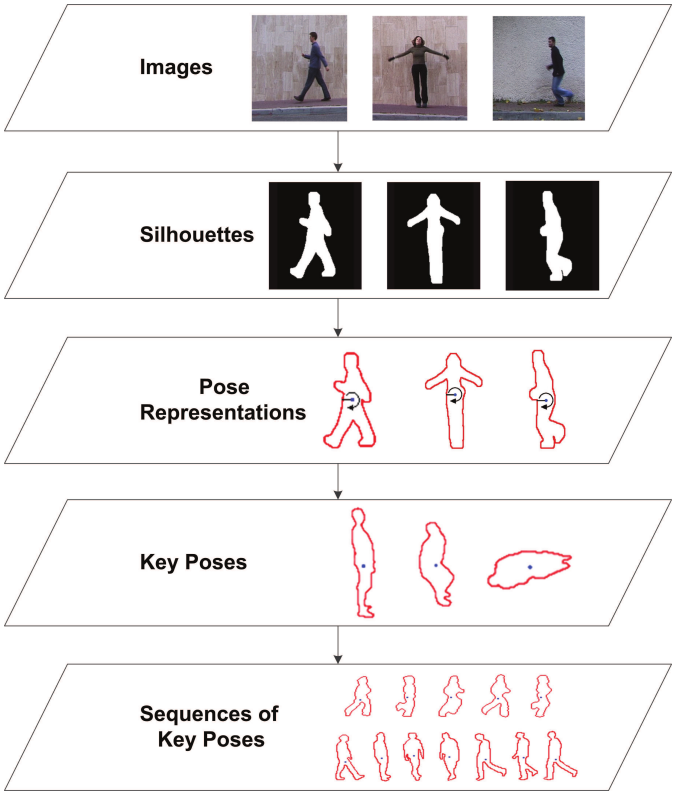



FIGURE 2.1 – Extraction de poses clés issues de la silhouette par [12]

### Approches utilisant les points d'intérêt

Ces approches sont généralement constituées de deux étapes principales que sont la détection et la caractérisation de points d'intérêt. Certaines approches utilisent la trajectoire et ajoutent pour cela une étape de suivi des points au cours du temps.

**Détection des points d'intérêt** Nous présentons ici différentes méthodes d'extraction de points d'intérêts au sein d'une image. La Figure 2.2 illustre les principaux types de points obtenus à partir des méthodes que nous décrivons.

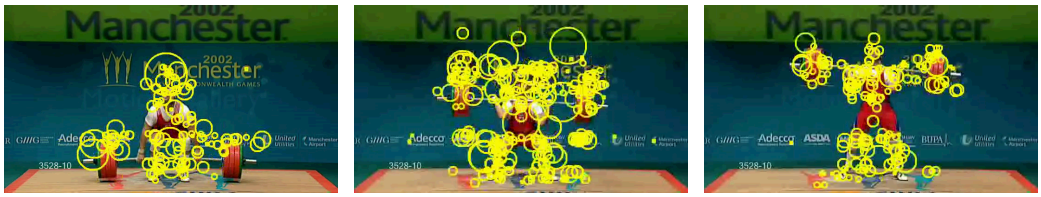
HARRIS et STEPHENS proposent en 1988 un des premiers détecteurs de points d'intérêts : le *détecteurs de coin Harris*. Il s'agit d'une détection à deux dimensions localisant les régions dans lesquelles l'image observe de fortes variations locales d'intensité dans deux directions orthogonales [17]. Ce détecteur a été largement utilisé dans le domaine de la vision par ordinateur pour la reconnaissance d'images [18], [19], la stéréo-vision [20], [21], l'estimation de flux optique [22], *etc.*

En 2003, LAPTEV et LINDEBERG [13], [23] étendent ce détecteur à la vidéo en ajoutant une composante temporelle. Ce détecteur (*3D-Harris* [13], [23]) extrait des régions à fortes variations à la fois dans l'espace (coins de Harris) et dans le temps (changement de direction du point d'intérêt). En prenant en considération une information temporelle

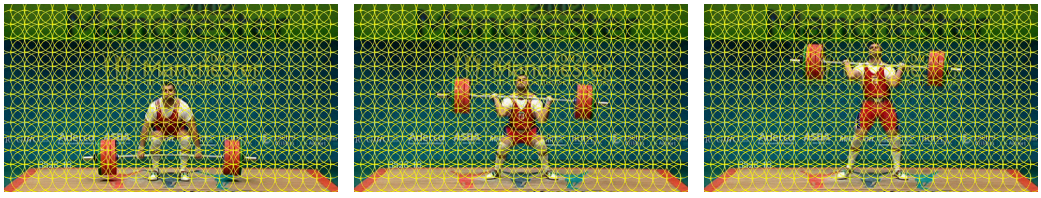
(a) Images simples

(b) Points Harris 3D [13]

(c) Points à partir de filtres de Gabor [14]



(d) Points Hessian 3D [15]



(e) Points denses [16]

FIGURE 2.2 – Comparaison des différentes méthodes d'extraction de points d'intérêt.

locale, les auteurs augmentent ainsi le pouvoir discriminant des primitives extraites.

Pour Dollár, Rabaud, Cottrell et al. [14], il existe des situations dans lesquelles ce type de points (à fortes variations dans les 3 dimensions) est trop rare dans une vidéo pour décrire correctement certains problèmes comme la détection de visages ou l'analyse de comportement de rongeurs. Ils proposent alors en 2005 dans [14] une extraction de points d'intérêts à partir d'un filtre gaussien 2D, appliqué spatialement, et d'une paire de filtres de Gabor en quadrature appliqués temporellement. Cette méthode d'extraction génère un plus grand nombre de points d'intérêts et augmente ainsi la quantité d'information disponible.

Willems, Tuytelaars et Van Gool [15], relèvent en 2008 la dépendance à l'échelle de ce dernier détecteur et proposent un extracteur à partir d'une matrice Hessienne spatio-temporelle afin d'obtenir une extraction encore plus dense de données. Les paramètres d'échelles spatiales et temporelles sont optimisés directement au sein de l'algorithme de reconnaissance d'actions.

Partant du constat qu'un échantillonnage dense améliore les résultats en classification d'images [24], [25] ainsi qu'en reconnaissance d'actions [26], Wang, Kläser, Schmid et al. proposent en 2011 une extraction de trajectoires sur une grille dense [16]. Plus précisément les auteurs proposent de définir, sur chaque image d'une vidéo, une grille régulière de points espacés de $W$ pixels (en pratique un point tous les $W = 5$ pixels) à différentes échelles spatiales (en pratique jusqu'à 8 échelles différentes pour des vidéos à forte résolution).

Garrigues et Manzanera [27] proposent aussi, en 2012, de suivre un nombre important (échantillonnage *semi-denses*) de points d'intérêts au cours du temps. Les auteurs extraient pour cela des points clefs à différentes échelles à l'aide d'un critère sur le contraste local. Les points instables ou probablement non pertinents sont ensuite filtrés lors de la génération des trajectoires.

La description de vidéos à partir de points particuliers a fait ses preuves pour la classification d'action. On constate une évolution vers une extraction de plus en plus dense de points d'intérêts dans chaque image. Pour exploiter l'information temporelle contenue dans une vidéo, ces points peuvent être suivis d'image en image afin de représenter les mouvements qu'ils observent au sein de la vidéo.

**Suivi des points d'intérêt au cours du temps** En suivant ces points au cours du temps, on obtient des trajectoires de points saillants, dont la description sert à des algorithmes de classification. Une des premières méthodes d'extraction de trajectoire est le *KLT tracker* [28], s'appuyant sur une mise en correspondance image à images de points.

Dans leur article proposant une extraction de trajectoires sur une grille dense [16], Wang, Kläser, Schmid et al. suivent chacun de ces points à partir du flux optique. Le flux optique $\omega_t$ est calculé pour chaque image $I_t$, relativement à $I_{t+1}$, à toutes les échelles indépendamment. La position à l'instant $t+1$ d'un point $P_t = (x_t, y_t)$ de l'image $I_t$ est alors estimé à partir de ce flux après filtrage médian [29] :

$$P_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + \mathrm{Med}\left(\omega_t|_{(x_t, y_t)}\right)$$

Notons qu'une fois le flux optique calculé et filtré sur une image, estimer la position d'un point à l'image suivante consiste en une simple somme. En outre, puisque les points d'une zone homogène ne peuvent être suivis, les auteurs proposent de les retirer selon un critère proposé dans [30] dépendant des valeurs propres de la matrice d'autocorrélation du voisinage de ces points. La Figure 2.3 montre un exemple de points extraits par cette méthode. Une trajectoire est alors définie comme la succession de positions d'un point $P_t$ sur $L$ instants temporels : $(P_t, P_{t+1}, \ldots, P_{t+L})$. Pour limiter la dérive d'un point lors du calcul de la trajectoire, les auteurs définissent $L = 15$ instants temporels.

FIGURE 2.3 – Exemple de grille de points denses extraits. Image extraite de [29]

Une fois extraite, la trajectoire en elle-même est décrite par la séquence $(\delta P_t, \ldots, \delta P_{t+L-1})$ avec $\delta P_t = (P_{t+1} - P_t)$, normalisé par la somme des magnitudes des vecteurs déplacements :

$$T = \frac{(\delta P_t, \ldots, \delta P_{t+L-1})}{\sum_{j=t}^{t+L-1} \left\| \delta P_j \right\|}$$

Dans la suite de cette thèse, nous appellerons ce descripteur *forme de trajectoire* ou *TS (Trajectory Shape)* puisque qu'il décrit l'apparence de la trajectoire indépendamment de sa longueur.

En 2013, WANG et SCHMID améliorent ces trajectoires denses [16] en introduisant les iDT [31] (*improved Dense Trajectories*). Dans ces travaux ils estiment le mouvement global de l'image, en vue de rendre l'extraction de trajectoire robuste au mouvement de la caméra. De plus, les auteurs définissent une région d'intérêt à partir d'une détection de personne.

**Caractérisation d'une région spatio-temporelle** Une fois les points d'intérêts localisés au sein de l'image et éventuellement suivis, il est nécessaire de les caractériser afin d'en extraire un maximum d'information exploitable (discriminante et comparable). Plus généralement, il s'agit de capturer l'information spatio-temporelle locale dans

une région de l'image. Lorsque l'on considère des points d'intérêt, cette région spatio-temporelle est définie au voisinage de ces points. On peut également caractériser une région plus vaste comme une partie de l'image, l'image entière, ou une succession temporelle d'images. Par soucis de généralité, considérons une région comme un cuboïde de taille $M \times N \times T$.

Parmi les descripteurs les plus couramment répandus, on trouve les *Histogrammes de Gradients Orientés (HOG)* [32] et les *Histogrammes de Flux optiques (HOF)* [33]. Le descripteur *HOG* décrit l'apparence spatiale locale par la génération d'un histogramme modélisant les proportions de présence des différentes directions de gradient au sein de la région considérée. Un *HOG* à $n$ dimensions accumule les valeurs de gradient pour chacune des $n$ directions réparties uniformément. Un choix répandu de $n$ est $n = 8$ directions de gradient. Klaser, Marszalek et Schmid proposent en 2008 [34] une extension des *Histogrammes de gradients (HoG)* au domaine spatio-temporel : les *HoG3D*. Ils considèrent une dimension supplémentaire du vecteur gradient, correspondant ici à la dimension temporelle. Les gradients du HOG3D sont alors calculés à l'aide de vidéos intégrales.

Structurellement proche du *HOG*, le descripteur *HOF* [33] modélise les mouvements locaux au sein de l'image. Il s'agit d'un histogramme des directions principales du flux optique au sein de la région considérée. Notons que pour ce descripteur, on considère également une direction nulle, représentant les pixels immobiles.

A partir de ce descripteur HOF, Wang, Kläser, Schmid et al. [16], [31] introduisent le descripteur *MBH (Motion Boundary Histograms)* [35]. Ce dernier consiste en un histogramme généré à partir des dérivées spatiales (horizontale et verticale) du flux optique. L'introduction d'une dérivée dans ce descripteur implique que les mouvements localement constants sont ignorés et seuls les variations de flux optiques sont encodées [29]. En codant les changements de flux optiques dans l'image, ce descripteur met en valeur les frontières des mouvements, d'où son nom *Motion Boundary Histograms* (Histogramme des frontières de mouvement). La Figure 2.4 illustre l'information capturée par le gradient, le flux optique et les MBH, respectivement.

Ces descripteurs ont montré leur efficacité pour la reconnaissance d'actions [16]. Ils sont extraits dans un volume spatio-temporel autour des trajectoires extraites. Ce volume, d'une taille spatiale $N \times N$ pixels et s'étendant sur $L$ instants temporels, est divisé en cellules selon une grille spatio-temporelle de dimension $n_\sigma \times n_\sigma \times n_\tau$. Sur chacune de ces cellules sont calculés les trois descripteurs (HOG, HOF et MBH) décrits précédemment. Les auteurs définissent comme paramètres $N = 32$ pixels, $n_\sigma = 2$ pixels et $n_\tau = 3$ instants. Cette extraction est illustrée sur la Figure 2.5.

### 2.2.2 Extraction à partir de cartes de profondeur

Depuis 2010, avec l'émergence des capteurs capables d'extraire des cartes de profondeur à faible coût tels que la Kinect, la Kinectv2, la *Asus Xtion* ou encore la *Camine PrimeSense*, des descripteurs adaptés ont vu le jour.

Li, Zhang et Liu proposent en 2010 de représenter un humain par un ensemble de points extraits des contours des projections des points 3D dans 3 plans orthogonaux

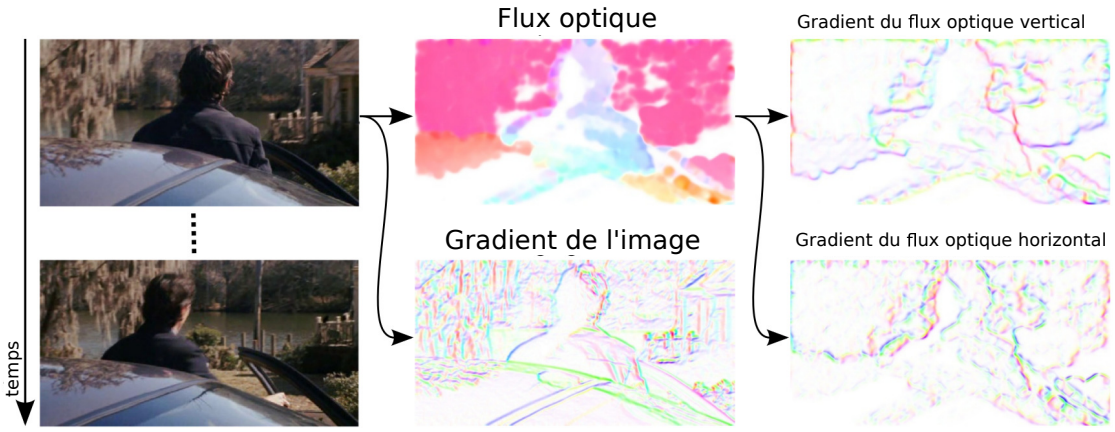



FIGURE 2.4 – Illustration de l'information capturée par les descritpeurs HOG, HOF et MBH. La caméra observe un mouvement de gauche à droite et la personne s'en éloigne. L'orientation des flux et gradients est représentée par la couleur et leur intensité par la saturation. Image extraite de [29].

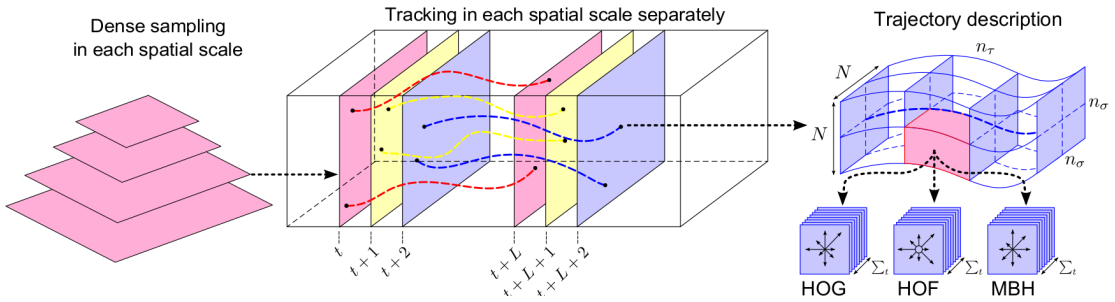



FIGURE 2.5 – Extraction des descripteurs de trajectoires denses, proposée dans [16]. Cette figure est issue de cet article.

(face, côté, dessus) [36]. Cette représentation est inspirée des travaux sur les silhouettes 2D et permet une description parcimonieuse de la posture. Notons qu'une carte de profondeur ne fournit pas une information 3D complète : seule la profondeur des points visibles par le capteur est disponible. Ainsi, les projections dans les plans orthogonaux au plan de la caméra (côté et dessus) contiennent beaucoup moins d'informations que celles dans le plan parallèle (voir Figure 2.6).

XIA et AGGARWAL proposent dans [37] une extraction de points d'intérêt spatio-temporels (STIPs) adaptée à des données de profondeur : les *DSTIPs*. Ils ont la forme de cubes spatio-temporels. Ces cubes sont divisés en sous-blocs pour lesquels sont calculés des histogrammes sur les profondeurs locales. Pour décrire ces points d'intérêts (cubes spatio-temporels), les auteurs introduisent le *Depth Cuboid Similarity Feature (DCSF)* : un vecteur composé des similarités $S$ calculées entre tous les sous-blocs, à l'aide de la distance de Bhattacharyya. La Figure 2.7 illustre l'extraction et la représentation de ces DCSF.

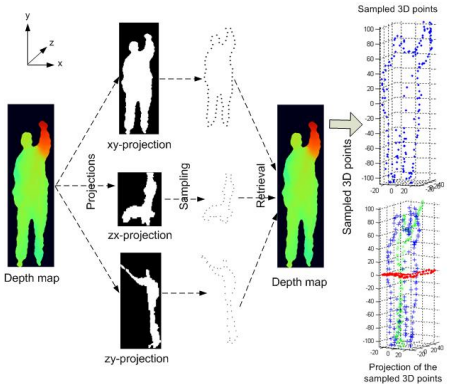



FIGURE 2.6 – Projections des points 3D de la silhouette, figure extraite de [36]

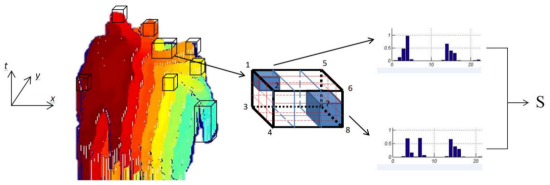



FIGURE 2.7 – Illustration de l'extraction des DCSF à partir d'une succession temporelle de cartes de profondeur. $S$ est le vecteur de similarité calculé à partir de l'ensemble des distances de Batthacharyya entre les sous-blocs. Images extraite de [37].

D'autres méthodes représentent les cartes de profondeur par l'occupation locale de l'espace [38]–[40]. Il s'agit d'une représentation du taux d'occupation d'un cube spatial 3D [38], [39] ou spatio-temporel 4D [40]. Dans [38], [39], les auteurs proposent de diviser le cube spatial en blocs $b_{xyz}$ et de calculer sur chacun d'entre eux l'information locale d'occupation $o_{xyz}$ à partir d'une sigmoïde :

$$o_{xyz} = \delta\left(\sum_{p \in b_{xyz}} I_p\right) \qquad \delta(x) = \frac{1}{1 + e^{-\beta x}} \tag{2.3}$$

avec $I_p = 1$ si de la matière est présente au point $p$, $I_p = 0$ sinon. Ces cubes sont extraits de façon aléatoire [38] ou autour des articulations de la personne [39].

En vue de fournir un descripteur plus riche dans l'espace spatio-temporel, d'autres travaux s'intéressent à la modélisation des surfaces 3D ou 4D [41], [42]. OREIFEJ et LIU proposent en 2013 un descripteur consistant en un histogramme des normales à la surface spatio-temporelle (HoN4D) [41]. Cette surface se définit à partir de l'évolution temporelle de la profondeur de la scène. Le HoN4D se différencie principalement du HOG3D par la prise en compte de l'amplitude du gradient dans la composition de l'histogramme.

SONG, TANG, LIU et al. introduisent en 2014 un descripteur utilisant également une

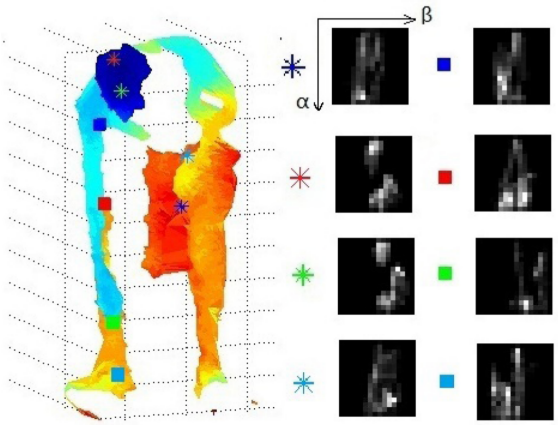



FIGURE 2.8 – Illustration du BSC [42]. $\alpha$ et $\beta$ sont deux composantes d'un vecteur $\overrightarrow{PP_0}$ en coordonnées cylindriques

surface dans l'espace de profondeur : le *Body Surface Context (BSC)* [42]. Il s'agit d'un histogramme à deux dimensions représentant un nuage de points au voisinage d'un point de référence $P_0$. Il est calculé à partir de la répartition des distances $d$ d'un nuage de points $\{P \mid d(P,P_0) < \sqrt{2} \times d_n\}$ avec $d_n$ un paramètre définissant la taille du voisinage considéré. Ce descripteur, illustré Figure 2.8, caractérise le contexte spatial 3D d'un point d'intérêt dans l'espace de profondeur.

Pour modéliser le mouvement au sein des cartes de profondeur et ainsi considérer une composante temporelle, YANG, ZHANG et TIAN introduisent en 2012 des cartes de mouvement de profondeur (*Depth Motion Maps (DMM)* [43]), reprises plus récemment par CHEN, LIU et KEHTARNAVAZ [44]. Après avoir obtenu des cartes binaires par projection des données de profondeur sur 3 plans orthogonaux (face, côté et dessus), une séquence est décrite par la différence image à image de ces cartes, cf Figure 2.9.

Par leur structure, les cartes de profondeur ne contiennent pas, ou très peu, de texture. Pour combler ce manque, certains travaux les combinent avec les informations extraites des images couleur associées [45]–[47]. SONG, LIU et TANG [47] suivent des points d'intérêt dans le domaine image à l'aide d'une méthode KLT pour obtenir des trajectoires 2D. Ils les enrichissent ensuite avec les informations 3D des cartes de profondeur. Elles sont alors décrites par la concaténation des descripteurs BSC [42] de chacun des points de la trajectoire.

### 2.2.3 Extraction de primitives à partir du squelette

Les descripteurs présentés jusqu'ici modélisent une vue globale de la scène, à partir d'images ou d'une modélisation 3D de la scène. L'analyse du comportement humain, et plus spécifiquement des actions ou activités, peut également se faire à partir d'une description des personnes réalisant ces actions ou activités.

L'objectif est de décrire la pose des personnes impliquées dans la scène à chaque

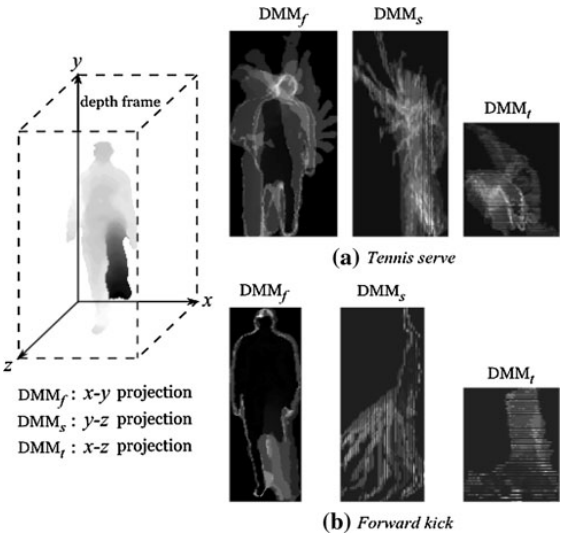



FIGURE 2.9 – Illustration des DMM utilisées dans [44]. Image extraite de cet article.

instant. Pour cela, une estimation des coordonnées (2D ou 3D) des articulations des personnes est faite. L'ensemble des positions des articulations définit alors un squelette qui pourra être utilisé pour la reconnaissance d'actions.

Ces squelettes peuvent être extraits à partir de dispositifs spécialisés (*motion Capture*) fonctionnant avec des caméras infrarouges associées à des marqueurs réfléchissants portés par la personne observée. Ce type de systèmes est très utilisé dans le cinéma d'animation pour donner vie à des personnages créés en images de synthèse. Certains travaux [48], [49] l'ont également utilisé dans le cadre de la reconnaissance de gestes. Le caractère invasif des marqueurs fixés sur le corps des participants rend cependant ce système non approprié pour des applications d'habitation intelligente.

Pour une estimation moins contraignante des poses des personnes présentes dans une scène, différents travaux ont été menés dans le but d'extraire les squelettes à partir des cartes de profondeur [50]–[53]. Ces méthodes, et plus particulièrement l'utilisation des squelettes qu'elles estiment, se sont popularisées avec le développement des capteurs de profondeur à moindre coût comme la Kinect.

A partir des coordonnées de chaque articulation du squelette dans un repère lié au capteur, on définit classiquement une description de ces squelettes qui se veut robuste au point de vue ainsi qu'aux variations inter-personnelles (tailles, physionomie).

Pour s'affranchir des variations liées à la position du capteur dans la scène, les coordonnées des articulations peuvent être exprimées dans un repère lié à la personne. On fixe alors le centre du repère sur un membre particulier et les coordonnées des autres articulations sont définies par rapport à cette nouvelle origine. WANG, WANG et YUILLE centrent ce repère sur la tête de la personne, puis ils normalisent les longueurs par rapport à la taille de la tête [54]. CHAUDHRY, OFLI, KURILLO et al. [55] et WU et

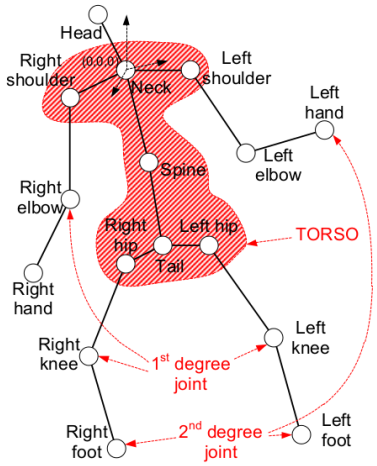



FIGURE 2.10 – Les articulations considérées dans [57]

SHAO [56], centrent le repère par rapport aux hanches. Partant du constat que les données issues des capteurs de profondeur sont habituellement bruitées, RAPTIS, KIROVSKI et HOPPE [57] définissent le repère à partir d'une réduction de dimension (ACP) utilisant les coordonnées des 7 articulations composant le torse (représentées en Figure 2.10).

Une autre méthode pour s'affranchir d'un repère fixé par le capteur est de définir une représentation locale des articulations du squelette. Par exemple à partir de l'ensemble des angles formés aux articulations (angles entre deux membres) [58] ou des positions relatives des articulations deux-à-deux [59], [60].

AMOR, SU et SRIVASTAVA développent dans [61] une suite d'outils géométriques à partir de l'espace de formes de Kendall [62] pour représenter et comparer des poses de squelettes.

On peut ensuite ajouter un aspect temporel à ces descripteurs en considérant la suite temporelle des positions des articulations [63]. YANG et TIAN [59] définissent en 2014 les *EigenJoints* à partir des vecteurs entre les articulations prises deux à deux :

1. au même instant $t$ ($f_{t,t}$),
2. entre deux instants temporels successifs ($f_{t,t-1}$),
3. entre l'instant considéré et un instant initial 0 ($f_{t,0}$),

avec $f_{t,p} = \{x_i^t - x_j^p \mid x_i^t \in X_t; x_j^p \in X_p\}$ et $X_p$ l'ensemble des articulations extraites à l'instant $p$. Ce descripteur décrit à la fois la pose de la personne ainsi que son évolution temporelle.

### 2.2.4 Caractérisation d'un ensemble de descripteurs

La plupart des méthodes présentées précédemment utilisent une description de données locales dont le nombre de primitives extraites varie fortement d'une image à l'autre. Une représentation plus stable doit être trouvée. Celle-ci peut passer par une discrétisation puis une agrégation des données.

### Discrétiser les données

L'idée est de créer un partitionnement de l'espace des descripteurs et d'associer chaque exemple à la partition à laquelle il appartient.

L'approche qui semble la plus intuitive pour obtenir un tel partitionnement est de créer un nombre fixé de groupes à partir des distances entre les exemples. Cette approche géométrique conduit naturellement à l'algorithme dit des *k-moyennes* [64], [65] ou une variante très proches des *k-médiannes* [66]. Ce partitionnement peut aussi être réalisé de manière hiérarchique [67] ou à partir d'une étude spectrale de la similarité (*spectral clustering* [68]).

**L'algorithme des K-moyennes** consiste à partitionner l'espace des descripteurs en K parties. Soit $\{\mathbf{x_1} \dots \mathbf{x_N}\}$ l'ensemble des descripteurs extraits, avec $\mathbf{x}_i \in \mathbb{R}^q$. L'algorithme des *k-moyennes* consiste à trouver $K$ points (appelés *centres*) $\{\mathbf{c}_1, \dots, \mathbf{c}_K\}$ minimisant la somme des distances entre les $\mathbf{x}_i$ et leur plus proche centre respectif $\mathbf{c}_k$.

A cause du caractère NP-difficile de ce problème, le partitionnement est obtenu à l'aide d'un algorithme qui, à partir d'un ensemble de centres initiaux $\mathbf{c}_1, \dots, \mathbf{c_K}$, associe chaque $\mathbf{x}_i$ au centre $\mathbf{c}_k$ le plus proche ; puis redéfinit les centres $\mathbf{c}_k$ comme étant le barycentre des point associés à ces centres. L'algorithme répète ces opérations jusqu'à convergence.

L'algorithme des k-médianes (k-medoids) est très similaire à celui des k-moyennes. Il se différencie à l'étape de mise à jour des centres qui est faite à partir du mot médian de chaque groupe et non de son barycentre.

**Le clustering spectral** partitionne l'espace en utilisant une étude spectrale de l'ensemble des données. Deux approches voisines sont généralement utilisées : le clustering spectral normalisé et le clustering spectral non-normalisé [69]. Ces méthodes présentent l'avantage de ne pas faire de supposition sur la forme des regroupements considérés et génèrent, à partir de l'étude d'une matrice de similarité des regroupements qui peuvent ne pas être globulaires, contrairement à ceux obtenus avec les algorithmes de *k-moyenne* et *k-médianne*. En revanche, la génération de la matrice de similarité pour des ensembles de données relativement larges peut s'avérer trop coûteuse.

Les centres $\mathbf{c}_k$ ainsi obtenus sont souvent appelés mots visuels, par analogie aux mots d'un dictionnaire.

### Assignation des descripteurs

La quantification par partitionnement génère un dictionnaire de $K$ partitions, chacune représentée par un centre $\mathbf{c}_k$, aussi appelé *mot* du dictionnaire. L'objectif de l'étape d'encodage est de représenter à partir de ce dictionnaire chaque descripteur $\mathbf{x_i}$. En d'autres termes, il s'agit de définir un vecteur $\mathbf{s_i}$ représentant $\mathbf{x_i}$ après quantification (par soucis de simplification, on notera ce vecteur $\mathbf{s}$ en l'absence d'ambiguïté).

Pour présenter les différentes méthodes d'encodage, introduisons une fonction $\phi(\mathbf{x}, k)$ utilisée pour déterminer la valeur de la $k^{\text{ème}}$ composante $s_k$ de $\mathbf{s}$ : $s_k = \phi(\mathbf{x}, k)$.

**L'encodage dur** (HC pour *Hard Coding*) consiste à associer chaque descripteur à son plus proche voisin parmi les $K$ mots du dictionnaire. $\mathbf{s}$ a alors toutes ses composantes nulles sauf $s_k$ qui vaut 1 :

$$\phi_{HC}(\mathbf{x}, k) = \begin{cases} 1, & \text{si } k = \arg\min_k \|\mathbf{x} - \mathbf{c}_k\|_2 \\ 0 & \text{sinon} \end{cases} \tag{2.4}$$

Un tel encodage réalise une quantification sévère qui peut engendrer une perte importante d'informations.

**L'encodage doux** (SoC pour *Soft Coding*) fait participer tous les mots du dictionnaire dans la description du point $\mathbf{x}$. Chaque composante $k$ étant pondérée relativement à la distance entre $\mathbf{x}$ et le centre $\mathbf{c}_k$. Par exemple, dans le cas d'un encodage normalisé, on peut utiliser [70] :

$$\phi_{SoC}(\mathbf{x}, k) = \frac{\exp(-\beta\|\mathbf{x} - \mathbf{c}_k\|_2^2)}{\sum_{j=1}^{K} \|\mathbf{x} - \mathbf{c}_j\|_2^2)} \tag{2.5}$$

où $\beta$ contrôle la "sévérité"" de l'encodage.

**L'encodage creux** (SpC pour *Sparse Coding*) réalise un compromis entre le caractère éparse de l'encodage et la perte d'information qu'il génère. Il peut être généré à partir des représentations parcimonieuses [71] :

$$\mathbf{s} = \arg\min_{\mathbf{s}} \left( \|\mathbf{s}\|_1 + \beta \|\mathbf{x} - \sum_k s_k \mathbf{c}_k\|_2^2 \right) \tag{2.6}$$

où la norme 1 assure la parcimonie et $\beta$ est un terme de pondération entre la parcimonie et l'attache aux données.

L'encodage creux peut aussi être réalisé en faisant participer que les $l$ plus proches centres de $\mathbf{x}$, avec $l$ fixé. On a alors :

$$\phi_{SpC}(\mathbf{x}, k) = \begin{cases} \frac{\exp(-\beta\|\mathbf{x} - \mathbf{c}_k\|_2^2)}{\sum_{j=1}^{K} \|\mathbf{x} - \mathbf{c}_j\|_2^2)}, & \text{si } \mathbf{c}_k \in N_l(\mathbf{x}) \\ 0 & \text{sinon} \end{cases} \tag{2.7}$$

avec $N_l(\mathbf{x})$ l'ensemble des $l$ plus proches voisins de $\mathbf{x}$.

Ces différentes méthodes de codage sont illustrées Figure 2.11.

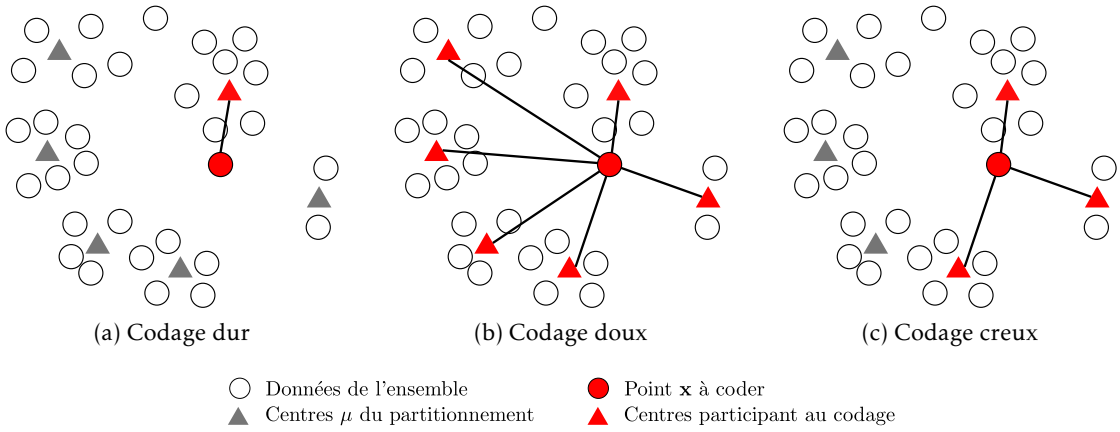



FIGURE 2.11 – Illustration des techniques de codage des descripteurs.

### 2.2.5 Représentation d'un ensemble de descripteurs

Pour représenter les images à un niveau plus élevé que le descripteur lui-même (à l'échelle d'une zone de l'image, d'une image ou d'une succession temporelle d'images par exemple), plusieurs méthodes sont couramment utilisées.

**Les sacs de mots** qui sont assez intuitifs, consistent à accumuler les vecteurs $\mathbf{s_i}$ représentant les points $\mathbf{x_i}$ pour obtenir le descripteur $\mathbf{S}$ de l'ensemble de points. $\mathbf{S}$ est alors de dimension $K$ :

$$\mathbf{S} = \sum_{i=1}^{N} \mathbf{s_i} \tag{2.8}$$

**Le vecteur de descripteurs aggrégés localement(VLAD)** a été proposé en 2010 par JÉGOU, DOUZE, SCHMID et al. qui introduisent un nouveau descripteur inspiré du succès des vecteurs de Fisher [73]. Le VLAD (pour *Vector of Locally Aggregated Descriptor*) décrit à l'aide d'un vecteur de taille fixe un ensemble de descripteurs. Une fois un partitionnement réalisé (par un algorithme des *k-moyennes* par exemple), chaque partition est décrite par une représentation de la distribution des descripteurs qu'elle contient. Une description plus détaillée de ce descripteur se trouve en section 5.1.2.

## 2.3 Les méthodes de reconnaissance d'actions

### 2.3.1 Paradigme globaux

Les méthodes de classification dites *globales* sont des méthodes prenant en compte une vidéo dans son ensemble sans considération conséquente de la dimension temporelle. En d'autres termes, il s'agit d'associer une action à une séquence complète, en négligeant

totalement, ou fortement, l'agencement temporel des descripteurs. Il existe quelques grandes familles de paradigmes de classification, se distinguant d'un point de vue théorique et fonctionnel, dont dérive la grande majorité des méthodes globales. Ces méthodes se différencient alors majoritairement par les descripteurs qu'elles extraient, les paramètres intrinsèques qu'elles considèrent ainsi que la modélisation globale de la séquence. Pour être compatibles avec une telle approche, les données extraites des différentes séquences doivent être exprimées par un vecteur de primitives de taille fixe dont le choix peut être crucial pour les performances.

Nous présentons dans cette section les principaux algorithmes de classification et pour chacun d'eux nous décrivons plus précisément certains travaux de l'état de l'art.

### Classification par plus proche(s) voisin(s)

Un des algorithmes de classification les plus simples consiste à associer à un exemple $\mathbf{x_i}$ la classe de son plus proche voisin dans l'ensemble des exemples de la base d'entraînement. Mathématiquement, cela s'écrit :

$$\hat{y}(\mathbf{x_i}) = y(\arg\min_{\mathbf{x_j}} \|\mathbf{x_i} - \mathbf{x_j}\|) \tag{2.9}$$

D'une façon plus générale, l'algorithme des $K$ plus proches voisins consiste à choisir pour $\mathbf{x_i}$ la classe majoritaire parmi les classes associées à ses $K$ plus proches voisins.

Bien que très simple, cet algorithme a fait ses preuves dans de nombreux travaux [74]–[78]. La notion de plus proches voisins est dépendante de la distance considérée. Par exemple, Batra, Chen et Sukthankar utilisent dans [75] une distance euclidienne alors que Mokhber, Achard et Milgram [79] utilisent une distance de Mahalanobis, plus adaptée à leur descripteur. Plus précisément, ces derniers construisent des volumes spatio-temporels 3D en empilant les régions correspondant aux silhouettes. Ils obtiennent ainsi une représentation volumétrique qu'ils décrivent par un ensemble de caractéristiques globales.

### Classification bayésienne

Implicitement, l'algorithme des $K$ plus proches voisins apprend la probabilité $P$ sur $\mathcal{X} \times \mathcal{Y}$ selon laquelle sont tirés les exemples [80]. Un autre paradigme de classification fait intervenir une estimation explicite de la densité de probabilité $\mathcal{P}(y|\mathbf{x})$ d'avoir la classe y connaissant l'exemple $\mathbf{x}$. Supposons que la donnée $\mathbf{x}$ est représentée par $n$ mots $c_i, i = \{1,\ldots,n\}$ d'un dictionnaire. $c_i$ est un entier naturel ($c_i \in \{1,\ldots,K\}$) identifiant le $i^{\text{ème}}$ centre $\mathbf{c}_i$ du dictionnaire.

A l'aide du théorème de Bayes et de la définition des probabilités conditionnelles, on exprime $\mathcal{P}(y|c_1,\ldots,c_n)$ comme :

$$\mathcal{P}(y|c_1,\ldots,c_n) = \frac{\mathcal{P}(y)\mathcal{P}(c_1,\ldots,c_n|y)}{\mathcal{P}(c_1,\ldots,c_n)} = \frac{\mathcal{P}(y)\mathcal{P}(c_1|y)\mathcal{P}(c_2|y,c_1)\cdots\mathcal{P}(c_n|y,c_1,\ldots,c_{n-1})}{\mathcal{P}(c_1,\ldots,c_n)} \tag{2.10}$$

En faisant certaines hypothèses, dont l'hypothèse d'indépendance des caractéristiques, on a $\mathcal{P}(c_i|y, c_j) = \mathcal{P}(c_i|y)$ quel que soit $i \neq j$ et on peut écrire :

$$\mathcal{P}(y|\mathbf{x}) \propto \mathcal{P}(y) \prod_{c \in M} \mathcal{P}(c|y)^{\mathcal{N}(c,\mathbf{x})} \tag{2.11}$$

Avec $\mathcal{N}(c, \mathbf{x})$ le nombre d'occurrences du mot $c$ dans l'exemple $\mathbf{x}$ et $M$ l'ensemble des mots du dictionnaire. Le premier terme concerne la distribution des classes que l'on peut supposer équiprobables ou estimer à partir de l'ensemble d'entraînement. Les termes $\mathcal{P}(c|y)$ sont à estimer pour chacun des mots à partir de l'ensemble d'entraînement.

En supposant que les exemples sont générés à partir d'une loi multinomiale, un estimateur du maximum de vraisemblance donne :

$$\mathcal{P}(c|y) = \frac{\sum_{x' \in \mathcal{X}_E^y} \mathcal{N}(c, x')}{\sum_{x' \in \mathcal{X}_E} \mathcal{N}(c, x')} \tag{2.12}$$

avec $\mathcal{X}_E^y$ l'ensemble des exemples d'entraînement associés à la classe $y$ et $\mathcal{X}_E$ l'ensemble du jeu d'entrainement.

En considérant chaque classe équiprobable, on a alors :

$$\mathcal{P}(y|x) \propto \mathcal{P}(y) \prod_{c \in M} \left( \frac{\sum_{x' \in \mathcal{X}_E^y} \mathcal{N}(c, x')}{\sum_{x' \in \mathcal{X}_E} \mathcal{N}(c, x')} \right)^{\mathcal{N}(c,x)} \tag{2.13}$$

Notons que si un couple d'observation et d'activité $(c, y)$ n'a jamais été observé dans l'espace d'entraînement, alors $\mathcal{P}(c|y)$ est estimée comme nulle. Or, $\mathcal{P}(y|\mathbf{x})$ étant un produit, on perd alors toute information apportée par les autres primitives. Pour pallier ce problème, Sarkar, Lee et Lee adaptent dans [81] des méthodes classiques de lissage de la recherche d'information (*Information Retrieval*) [82] [83] pour la reconnaissance d'actions.

Une autre catégorie d'approche gérant les descripteurs globaux réside dans les approches discriminatives comme les SVM (Support Vector Machine) définis dans un premier temps pour les cas à deux classes (binaires).

#### SVM, le cas binaire linéaire

Pour prédire l'action se déroulant dans une vidéo, une approche classique [14], [33], [84]–[86] est l'utilisation d'un SVM (*Séparateur à Vaste Marge* ou *Support Vector Machine* ) [87]. Cet algorithme, apprenant une séparation des données en fonction de leur classe peut être, par exemple, appliqué aux histogrammes représentant la vidéo comme dans Laptev, Marszałek, Schmid et al.

Soit un ensemble de données $\mathcal{X}$ dont chaque exemple est associé à une cible $y^* \in \mathcal{Y}$, avec $\mathcal{Y} = \{-1, 1\}$. Un tel ensemble est illustré sur la Figure 2.12 pour des données $\mathbf{x} \in \mathbb{R}^2$.

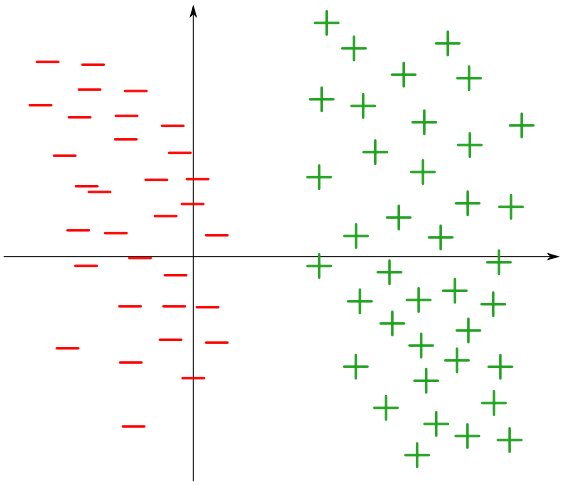


FIGURE 2.12 – Données associées à un label positif ou négatif

La classification binaire consiste à prédire la classe $\hat{y}(\mathbf{x}_i)$ associée à l'exemple $\mathbf{x}_i$. On parle de classification binaire car $\hat{y}(\mathbf{x})$ ne peut prendre que deux valeurs ($-1$ ou $1$).

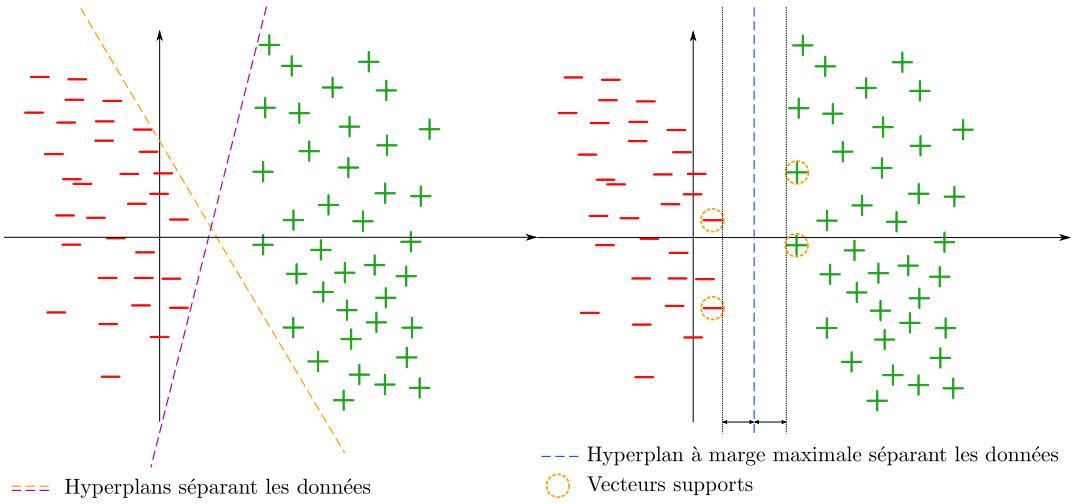
L'ensemble des données est dit *séparable* s'il existe un hyperplan séparant les données selon leur classe. C'est à dire si tous les exemples $\mathbf{x}_+$ associés à la cible $y_+ = 1$ sont situés d'un côté de ce plan et si tous les exemple $\mathbf{x}_-$ associés à $y_- = -1$ sont situés de l'autre côté.

Adoptons, pour un hyperplan $P$, le paramétrage par un couple $(b, \mathbf{w}) \in \mathbb{R} \times \mathbb{U}_D$ avec $\mathbb{U}_D$ l'ensemble des vecteurs unitaires de dimension $D$. Le vecteur $\mathbf{w}$ est normal à $P$. Si l'ensemble est séparable et que l'on connaît un hyperplan séparant les données selon leur classe, classifier un exemple $\mathbf{x}_i$ revient à connaître le signe du produit scalaire $\mathbf{w}^\top \mathbf{x}_i$ +b.

Dans le cas de la Figure 2.12, il existe une infinité de plans séparant les données dont deux d'entre-eux sont représentés sur la Figure 2.13a. Ces deux plans sont des cas limites, très sensibles à une perturbation des données. La solution la plus stable à une perturbation (et également la plus intuitive) est l'hyperplan se trouvant à la plus grande distance possible des deux sous-espaces de données $\mathcal{X}_+$ et $\mathcal{X}_-$ [88]. Cette distance, appelée *marge*, de l'ensemble de point à un plan est définie comme la distance minimale entre le plan et chacun des points de l'ensemble. La distance d'un point $\mathbf{x}$ au plan $P$ se définit comme $y^*(\mathbf{x}) \times (\langle \mathbf{w} | \mathbf{x} \rangle + b)$. Ce sont donc les points les plus proches du plan $P$ qui définissent la marge. Ils sont appelés *vecteurs supports*. Notons que deux ensembles seront séparables s'il existe une marge positive à ces deux ensembles.

Afin de trouver cet hyperplan maximisant la marge, illustré en Figure 2.13b, nous présentons l'algorithme SVM (*Support Vector Machine* ou *Séparateur à Vaste Marge*).

On cherche un vecteur $\mathbf{w}$ tel que la fonction $f$ définie par $f(\mathbf{x}_i) = \mathbf{w}^\top \mathbf{x_i} + b$ sépare les exemples selon leur classe. C'est à dire $f(\mathbf{x}_i) > 0$ si $y(\mathbf{x}_i) = +1$ et $f(\mathbf{x}_i) < 0$ si $f(\mathbf{y}_i) = -1$. Puisque les équations $\mathbf{w}^\top \mathbf{x} + b = 0$ et $C \cdot (\mathbf{w}^\top \mathbf{x} + b)$ désignent le même plan quel que soit $C \in \mathbb{R}$, on peut choisir de normer $\mathbf{w}$ sans changer le résultat. Choisissons une




(a) Deux hyperplans séparant les données

(b) Hyperplan séparant les données et maximisant la marge

FIGURE 2.13 – Séparation de données par un hyperplan

normalisation telle que l'exemple $\mathbf{x}_+ \in \mathcal{X}_+$ le de l'ensemble de points $\mathcal{X}_-$ appartienne au plan $P_+$ d'équation $\mathbf{w}^\top \mathbf{x}_+ + b = 1$ et que l'exemple $x_- \in \mathcal{X}_-$ le plus proche de l'ensemble de points $\mathcal{X}_+$ appartienne à $P_-$ : $\mathbf{w}^\top \mathbf{x}_- + b = -1$. Ces deux plans sont parallèles par construction et à une distance égale du plan recherché $P$. La marge $m$, définie sur la Figure 2.13b, est alors la moitié de la distance entre $P_+$ et $P_-$, à savoir :

$$m = \frac{\mathbf{w}^\top (\mathbf{x}_+ - \mathbf{x}_-)}{2\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|} \tag{2.14}$$

Trouver l'hyper-plan maximisant la marge $m$ s'écrit alors comme le problème d'optimisation :

$$\max_{\mathbf{w}} \frac{1}{\|\mathbf{w}\|}$$

*sous contrainte*

$$\begin{cases} \mathbf{w}^\top \mathbf{x}_i + b \geq 1 \text{ si } y^*(\mathbf{x}_i) = +1 \\ \mathbf{w}^\top \mathbf{x}_i + b \leq 1 \text{ si } y^*(\mathbf{x}_i) = -1 \end{cases} , \forall i \in \{1, \ldots, N\}$$

ou de façon équivalente :

$$\min_{\mathbf{w}} \|\mathbf{w}\|^2 \; s.c. \; y^*(\mathbf{x}_i)\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1, \forall i \in \{1, \ldots, N\}.$$

Dans le cas non-séparable, on relâche les contraintes et on introduit le terme $\xi \geq 0$ :

$$\min_{\mathbf{w},\xi_i \in \mathbb{R}^+} \|\mathbf{w}\|^2 + C \sum_{i=1}^{N} \xi_i,$$
$$y^*(\mathbf{x}_i)\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1 - \xi_i, \forall i \in \{1, \ldots, N\}.$$

Puisque $\xi_i \geq 0$, l'apprentissage revient à optimiser :

$$\min_{\mathbf{w}} \|\mathbf{w}\|^2 + C \sum_{i=1}^{N} \max(0, 1 - y^*(\mathbf{x}_i) f(\mathbf{x}_i)) \tag{2.15}$$

Cortes et Vapnik montrent dans [89] que le vecteur $\mathbf{w}$ peut s'exprimer comme une combinaison linéaire des vecteurs support $\mathbf{w} = \sum_{i=1}^{l} \alpha_i y(\mathbf{x}_i)\mathbf{x}_i$. Ils proposent alors de résoudre le SVM par le problème dual qui peut s'écrire :

$$\arg\max_{\alpha \in \mathbb{R}^N} \left( \sum_{n=1}^{N} \alpha - \sum_{i,j=1}^{N} y^*(\mathbf{x}_i) y^*(\mathbf{x}_j) \alpha_i \alpha_j \phi(\mathbf{x}_i, \mathbf{x}_j) \right)$$
$$\text{sous contrainte} \tag{2.16}$$
$$\sum_{n=1}^{N} y(\mathbf{x}_n)\alpha_n = 0, 0 \leq \alpha_n \leq C,$$

avec $\phi(\mathbf{x}_i, \mathbf{x}_j)$ la fonction noyau $\phi(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i | \mathbf{x}_j \rangle$.

On note au passage que considérer une fonction noyau $\phi$ différente revient à chercher une surface séparatrice non linéaire.

**SVM : Le cas non-linéaire**

Dans les cas, plus répandus, où les données observées ne sont pas séparables linéairement, il est possible qu'elles le soient dans un espace de plus grande dimension judicieusement choisi. Pour l'exemple très simple illustré sur la Figure 2.14, les données ne sont pas initialement séparables linéairement. Par contre, elles le sont dans le nouvel espace obtenu après transformation par la fonction $\psi(\mathbf{x}) = \psi(u, v) = (u, v, \sqrt{u^2 + v^2})$, avec $u$ et $v$ les deux composantes de $\mathbf{x}$. Il s'agit ici d'une séparation circulaire, comme illustré Figure 2.14.

Cette transformation revient à considérer dans l'équation 2.16 le noyau $\phi_{\text{circ}}(\mathbf{x}_i, \mathbf{x}_j) = \langle \psi(\mathbf{x}_i) | \psi(\mathbf{x}_j) \rangle$. Plus généralement, considérer un SVM non linéaire se fait par un choix judicieux du noyau $\phi$ considéré.

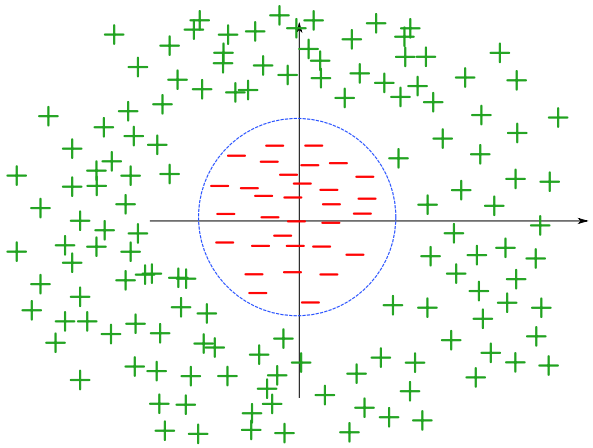


FIGURE 2.14 – Cas de données non séparables linéairement

**SVM : le cas multi-classes**

Le SVM présenté dans la section précédente n'est applicable qu'à un problème de classification binaire ($\mathcal{Y} = \{-1, 1\}$). Nous présentons dans la suite trois des principales adaptations pour gérer le cas à $|\mathcal{Y}| > 2$ classes.

**La méthode 1 contre tous** consiste à considérer pour chaque classe $y_k$ les exemples associés à cette classe $\{\mathbf{x}_i | y^*(\mathbf{x}_i) = y_k\}$ en positifs et tous les exemples associés à une classe différente en négatif. Après optimisation, on se retrouve alors avec $|\mathcal{Y}|$ hyperplans, un pour chaque SVM binaire défini.

Ces SVM étant indépendants, plusieurs d'entre eux peuvent donner une réponse positive pour un même exemple. L'attribution d'une classe à un exemple n'est alors plus triviale. On pourrait comparer les distances aux différents hyperplans pour faire un choix, mais puisque ces plans sont appris indépendamment, il n'y a pas de raison pour que lesdites distances soient comparables [90].

**La méthode 1 contre 1** considère autant de problèmes binaires qu'il y a de paires de classes possibles. Ainsi, un exemple reçoit $\frac{|\mathcal{Y}|(|\mathcal{Y}|-1)}{2}$ réponses binaires différentes correspondant à chaque classe et on lui attribue celle ayant reçu le plus de décisions binaires positives. Cela fait autant de frontières à optimiser en phase d'entraînement et de produits scalaires à calculer en phase de test ; ce qui rend impossible un passage à grande échelle.

Pour des problèmes de taille modeste, cette méthode se montre plus performante que la méthode *1 contre tous* [91]. Cela est compatible avec l'intuition selon laquelle séparer deux classes paraît plus simple que séparer une classe de toutes les autres[80].

**Apprendre une structure** Comme présenté précédemment, résoudre un SVM binaire revient à trouver un hyper-plan séparant les nuages de points, ou encore à trouver $\mathbf{w}$ tel

que $y^*(\mathbf{x}_i)\mathbf{w}^\top\mathbf{x}_i > 0$.

Réécrivons cette inégalité comme $\langle \mathbf{w} | (y^*(\mathbf{x}_i) * \mathbf{x}_i) \rangle > 0$, avec $\langle | \rangle$ désignant le produit scalaire. Trouver $\mathbf{w}$ respectant cette dernière formulation correspond au problème dit du *Perceptron*, introduit en 1958 par Rosenblatt dans [92]. Il consiste à chercher un vecteur $\mathbf{w} \in \mathcal{R}^D$ *représentatif* des nuages de points.

Or, Blum et Dunagan montrent dans [93] que l'algorithme du *perceptron* proposé dans [92] renvoie, si elle existe, une solution approché de :

$$\min_{\mathbf{w}\in\mathcal{R}^D} (\|\mathbf{w}\|_2^2 + C_\infty \sum_{i=1}^{N} max(0, \mathbf{w}^\top\mathbf{x}_i - 1)) \tag{2.17}$$

où les termes $\mathbf{w}^\top\mathbf{x} - 1$ sont contraints d'être positifs [80]. On constate que cette formulation est très similaire à l'équation 2.15 de l'apprentissage d'un SVM structuré. La seule différence étant la disparition du terme $b$.

Le problème du SVM consistant à chercher un hyperplan séparateur est donc similaire au problème de recherche d'un vecteur représentatif (perceptron). Or, ce dernier est généralisable au cas multi-classes en cherchant $n_C = |\mathcal{Y}|$ vecteurs, chacun représentatif des exemples associés à une classe particulière. Pour cela, on impose que le vecteur représentatif d'un nuage de points soit plus proche de ces points que tous les autres vecteurs représentatifs :

$$\min_{\mathbf{w}\in\mathcal{R}^{Y\times D}} \left( \|\mathbf{w}\|_2^2 + C \sum_{n,y} \max(0, (\mathbf{w}_{y^*(\mathbf{x}_n)} - \mathbf{w}_y)\mathbf{x}_n - 1) \right) \tag{2.18}$$

Trouver la classe associé à l'exemple $\mathbf{x}$ revient alors à trouver le vecteur représentatif le plus proche.

#### Méthodes de l'état de l'art

Les SVMs ont été utilisés avec succès pour la reconnaissance d'action, particulièrement par des méthodes utilisant des points d'intérêt. En 2005 Laptev proposent dans [23] de modéliser les vidéos par des sacs de mots extraits à partir de point d'intérêt *Harris 3D*. Ces points sont représentés par des descripteurs *HOG* et *HOF* puis la classification est faite à l'aide d'un SVM à noyau $\chi^2$. La même année, Dollár, Rabaud, Cottrell et al. proposent une méthode similaire [14] à partir de descripteurs extraits sur des cuboïdes autour de points d'intérêts, mais la classification en elle-même est faite dans le paradigme des *K plus proches voisins*.

### 2.3.2 Méthodes séquentielles

Les méthodes de classification globales, présentées dans la section précédente ignorent totalement l'agencement temporel des descripteurs extraits tout au long de la vidéo. Or, une très forte part de l'information est justement contenue dans cet agence-

ment. Nous présentons dans cette section les méthodes les plus utilisées reconnaissant les actions à partir des informations séquentielles.

**Méthodes génératives**

La classification par modèle de Markov cachés (HMM) est une méthode de classification générative. On cherche donc par cette méthode à estimer la distribution jointe de la séquence d'observation $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ et la cible $y$.

Considérons une séquence $v$ de longueur $T$ dont on a pour chaque instant $t$ une observation $o_t$ pouvant prendre ses valeurs dans $O_1, \ldots, O_M$. Un modèle de Markov caché discret associe à cette séquence une succession $(s_1, \ldots, s_T)$ d'états, chacun pouvant prendre sa valeur dans $\{e_1, \ldots, e_N\}$. Ce paradigme suppose que la séquence d'observation a été générée à partir de la séquence d'états cachés avec deux hypothèses importantes :

— Hypothèse de Markov d'ordre 1 : La transition d'un état à l'autre ne dépend que de l'état précédent.

— Une observation $o_t$ n'est conditionnée que par l'état $s_t$.

On décrit alors cette HMM par :

— $A = \{a_{ij} \mid a_{ij} = \mathcal{P}(s_{t+1} = e_i | s_t = e_j)$, les probabilités de transition d'un état à l'autre,

— $B = \{b_j(k) \mid b_j(k) = \mathcal{P}(o_t = O_k | s_t = e_j)$, les probabilités, pour chaque état, de générer chaque observation.

— $\pi = \{\pi_i | \pi_i = \mathcal{P}(s_1 = e_i)\}$, la probabilité de l'état initial.

Ces paramètres sont appris durant la phase d'entraînement.

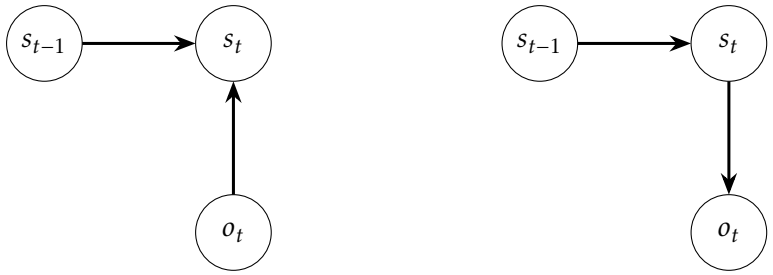
Pour réaliser une reconnaissance d'actions à partir de HMM, Achard, Qu, Mokhber et al. proposent dans [94] d'apprendre $|\mathcal{Y}|$ modèles, un pour chaque action possible. Pour classifier une séquence de l'ensemble de test, les auteurs retiennent l'action associée au modèle ayant le plus probablement généré cette séquence. Dans ces travaux, chaque instant est représenté par un volume 3D constitué des silhouettes extraites autour de cet instant.

Lv et Nevatia utilisent dans [95] un modèle 3D, à partir des angles des articulations, puis apprennent plusieurs HMM, chacun spécialisé sur un sous-ensemble d'articulations.

Il est également possible d'agencer plusieurs HMMs hiérarchiquement [96], [97]. C'est ce qu'appliquent Nguyen, Phung, Venkatesh et al. à la reconnaissance d'actions dans [98]. Dans ces travaux, un HMM bas niveau reconnaît des actions élémentaires à partir de déplacements dans la pièce, puis un HMM plus haut niveau reconnaît des activités à partir de ces actions.

Plus récemment, Wu, Zhang, Sener et al. [99] proposent en 2015 une adaptation de l'Allocation de Dirichlet latente à la reconnaissance d'action. Il s'agit d'un modèle génératif très utilisé dans le domaine de la modélisation de documents à partir du langage. Pour l'utiliser dans le domaine qui nous intéresse, ils ajoutent entre autre une modélisation de la distribution temporelle entre les sujets.

On pourrait citer d'autres modèles génératifs, comme les réseaux bayésiens dynamiques (DBN), utilisés pour la reconnaissance d'actions notamment dans [100] et [101].




(a) HMM (b) MMEM

Figure 2.15 – Graph de dépendances des méthodes HMM et MMEM [102]

Lors de l'apprentissage, les modèles génératifs estiment la probabilité conjointe $\mathcal{P}(\mathbf{x}, y)$. Or, l'objectif d'un algorithme de classification est de prédire la classe $y$, à partir de données $\mathbf{x}$ observées. C'est donc, à cette étape, la probabilité conditionnelle $\mathcal{P}(y|\mathbf{x})$ qui nous intéresse. L'idée des modèles discriminatifs est d'apprendre directement cette probabilité $\mathcal{P}(y|\mathbf{x})$ durant la phase d'apprentissage.

**Méthodes discriminatives**

En 2000, McCallum, Freitag et Pereira [102] introduisent les modèles de Markov d'Entropie Maximale (MMEM) qui peuvent être vus comme le pendant des HMMs pour les méthodes discriminatives. Dans ce modèle, le passage d'un état à l'autre dépend de l'état de l'instant précédent et de l'observation à l'instant considéré. Les probabilités de transitions et de génération d'une observation du modèle HMM sont alors remplacées par une unique probabilité $\mathcal{P}(s_t|s_{t-1}, o_t)$. D'un point de vue du graph des dépendances, la différence entre ces deux méthodes est illustrée sur la Figure 2.15.

Les champs aléatoires conditionnels (CRF pour *Conditionnal Random Fields*) ont été introduits en 2001 par Lafferty, McCallum et Pereira dans [103] en vue de construire un modèle probabiliste qui segmente et étiquette une séquence de données. Les auteurs mettent en avant la capacité de ce modèle à relâcher les hypothèses fortes d'indépendance faites dans les modèles génératifs tels que les HMMs.

Les CRF consistent donc en un champ aléatoire globalement conditionné par les observations. Les performances des CRFs ont été comparées [103]–[105] avec celles de modèles génératifs comme les HMMs. Ils ont notamment été utilisés pour la reconnaissance d'actions par Sminchisescu, Kanaujia et Metaxas dans [104] à partir de données issues de dispositifs *motion capture* et des descripteurs d'images. Les auteurs comparent leur approche utilisant les CRFs en chaine linéaire avec celles utilisant les HMMs et celles utilisant les modèles de Markov d'Entropie Maximale (MEMM) et constatent une amélioration des performances face à ces autres méthodes. Pour capturer la structure intermédiaire des séquences observées, Wang, Quattoni, Morency et al. présentent dans [106], [107] les CRFs cachés (HCRFs), de façon analogue aux états cachés présents dans les HMMs.

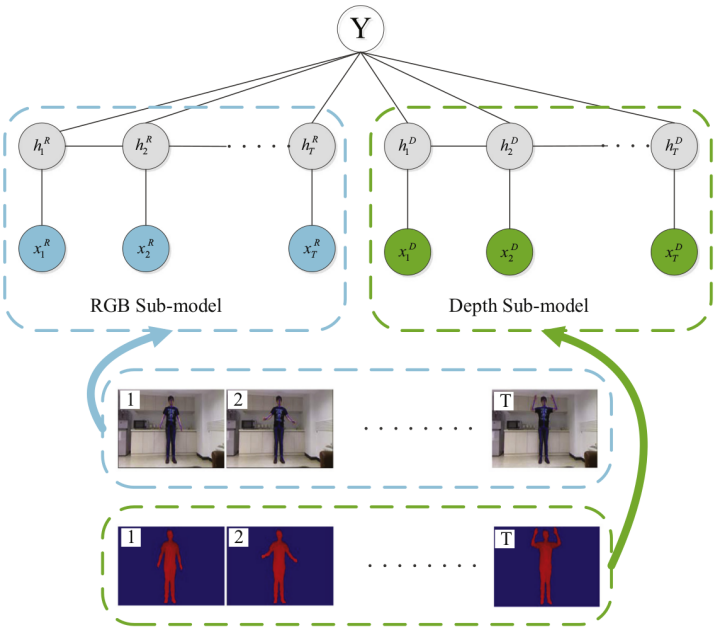



FIGURE 2.16 – Structure des HCRFs couplés (cHCRF), image extraite de [111]

Ces travaux ont été largement employés pour la reconnaissance d'actions [108]–[110] et se sont montrés performants dans ce domaine. Pour exploiter les informations en provenance à la fois des cartes de profondeur et des images, tout en s'adaptant à la spécificité de chacun de ces canaux, LIU, NIE, SU et al. introduisent en 2015 les HCRF couplés (cHCRF), illustrés en Figure 2.16. Il s'agit de deux HCRF, un par canal, couplés à un niveau supérieur. Plus récemment, SELMI et EL-YACOUBI proposent dans [112] une approche à deux couches. Une première couche décrit des segments de vidéos de façon discriminative à l'aide d'un SVM, puis un HCRF prédit l'action à partir de l'agencement temporel de ces segments.

D'autres variantes ont été explorées, comme celle de HAN, WU, LIANG et al. [113] qui proposent d'apprendre les actions à partir de 3 CRFs en cascade ou encore celle [114], [115] utilisant des CRFs factoriels, pour modéliser des interactions plus complexes entres les états. Cette modélisation des liens complexes entre états a notamment permis aux auteurs [115] d'exploiter les interactions entre les personnes et les objets qu'elles utilisent.

### 2.3.3 Méthodes utilisant des votes

Remarquons que le terme de droite de l'équation 2.13, introduite avec la classification bayésienne, peut être vu comme le score d'un vote de chaque mot $m$ appartenant à la donnée $\mathbf{x}$ avec un poids $\left(\frac{\sum_{x' \in \mathcal{X}_E^y} \mathcal{N}(c,x')}{\sum_{x' \in \mathcal{X}_E} \mathcal{N}(c,x')}\right)$ [80]. Cela permet d'introduire les méthodes d'élection pour lesquelles chaque mot vote pour chaque classe avec un poids différent. Le

choix des poids de vote peut être fait à partir du paradigme de classification bayésienne par exemple, comme présenté ici.

Les transformées de Hough se rapprochent également d'un paradigme de classification par élection. Nous présentons ces méthodes dans la section 3.1.

## 2.4 Synthèse

Nous avons présenté dans ce chapitre les algorithmes de classification de gestes, d'actions ou d'activités les plus importants de la littérature. La très grande majorité d'entre-eux discrétisent les données avant l'étape de reconnaissance en elle-même. Les méthodes reconnaissant les actions à partir de *sacs de mots* sont particulièrement répandues [7].

On note cependant une quantité moins importante de méthode de **détection** d'activités. En effet, la grande majorité des méthodes proposées travaillent à partir de vidéos pré-découpées qu'il s'agit de classifier. Il est possible de généraliser ces méthodes à des applications de détection en les appliquant sur une fenêtre glissante que l'on cherche à classifier ; mais une approche dédiée à la détection semble plus appropriée et surtout bien plus rapide car elle évite de paver l'espace temporel.

Les méthodes d'élection, et notamment les méthodes utilisant une transformée de Hough, obéissent à un paradigme adapté à la détection [80]. Le chapitre suivant présente quelques-unes de ces méthodes avant d'introduire une première contribution de cette thèse : une fusion d'information au sein d'un algorithme de détection utilisant un paradigme de transformée de Hough.

Chapitre 3

# Reconnaissance d'actions multi-vues et multi-descripteurs

## Introduction

Cette thèse s'intéresse à la détection (segmentation et reconnaissance) d'actions et d'activités humaines par vision. Cette détection peut être réalisée à partir de diverses modalités telles que des informations visuelles (issues d'images RVB), des informations sur la pose des personnes effectuant les actions (squelettes estimés) ou encore des informations 3D qui peuvent être par exemple éstimées à partir de cartes de profondeur. Ces différentes sources ne portent pas les mêmes informations et sont souvent complémentaires. Par exemple, les informations de contexte spatial présentes au sein de l'image ne portent pas la même information que la pose d'une personne. Considérer à la fois l'une et l'autre de ces modalités paraît très intuitif pour combler les lacunes de chacune.

Ce chapitre a pour objectif l'exploration de la fusion d'informations au sein d'un algorithme de détection. Nous présentons ci-après l'algorithme que nous avons choisi de mettre au cœur de la détection d'actions puis nous proposons trois niveaux de fusion au sein de cet algorithme. Nous évaluons ensuite les performances et l'amélioration apportée par cette combinaison d'informations sur une base de données adaptée avant de conclure par une mesure des temps de calcul associés.

## 3.1 Reconnaissance d'actions par transformée de Hough

Deux solutions existent pour détecter des actions dans des vidéos. La première consiste à paver l'espace temporel et appliquer un algorithme de reconnaissance d'actions sur chaque segment. Cette solution nécessite de considérer tous les instants possibles et toutes les longueurs d'actions possibles, ce qui amène à des temps de calcul ou une latence très importants et est difficilement réalisable pour une application temps réel. L'autre solution consiste à utiliser directement des algorithmes dédiés à la détection.

Parmi ceux-ci, on peut citer les HMM hiérarchiques comme présenté dans le chapitre précédant. Leur inconvénient est qu'ils nécessitent de voir toute la vidéo afin de réaliser l'optimisation et obtenir la segmentation idéale. Nous avons opté pour les méthodes d'élection par transformée de Hough. Cette approche, qui s'appuie sur une accumulation progressive d'indice est en outre assez intuitive. Elle affine ses hyppothèse au cours du temps et avec l'accumulation de descripteurs. Cette section décrit le paradigme de la transformée de Hough appliqué à la reconnaissance et détection d'action (ou activité).

**Origines de la méthode** La transformée de Hough est une technique originalement décrite pour des applications de reconnaissances de formes par Hough dans [116] puis généralisée par Duda et Hart dans [117]. Dans sa forme primaire, il s'agit de détecter les droites présentes dans une image. Une droite de l'image est représentée par son équation en coordonnées polaires :

$$\rho = x\cos\theta + y\sin\theta$$

La détection de droites se fait par une extraction des points de contours dans l'image puis un vote de chacun d'entre-eux pour l'ensemble des droites possibles passant par ce point. En pratique, il s'agit d'un arc de sinusoïde dans le plan $(\rho, \theta)$. Dans cet espace transformé, les courbes représentant les points issus d'une même droite dans l'image se coupent en un même point $(\rho_D, \theta_D)$ paramétrant la droite en question. L'accumulation des votes dans l'espace de Hough permet donc la localisation des droites de l'image. Cette méthode a ensuite été généralisée à la détection de formes paramétrées [118] puis, bien plus tard, à la détection d'objets [118] et d'actions [119].

**Détection d'actions dans le paradigme de Hough** Soit $v \in \mathcal{V}$ une vidéo de l'ensemble $\mathcal{V}$ des vidéos d'un jeu de données. $v$ est représentée par une suite temporelle d'instants représentés par un ensemble de descripteurs locaux $\{\mathbf{x}_i(t)\}$ avec $i \in \{1,\dots,n(t)\}$ et $n(t)$ le nombre de descripteurs issus de l'instant $t$. On veut connaître la classe $a_v(t) \in \mathcal{Y}$ associée à chaque instant $t$. $\mathcal{Y}$ est l'ensemble des classes (actions ou activités) possiblement détectées. Dans cette thèse, pour plus de lisibilité et en l'absence d'ambiguïté, nous désignons $a_v(t)$ par $a$ et $\mathbf{x}_i(t)$ par $\mathbf{x}$.

La détection d'action sur l'exemple $v$ par transformée de Hough peut être décrite en deux étapes principales :

1. **Extraction et quantification** des descripteurs à chaque instant pour l'obtention de mots $c_i(t) \in \mathcal{C}$, avec $\mathcal{C} = \{1,\dots,K\}$ et $K$ le nombre de centres de la quantification (dans le cas d'une quantification à l'aide d'un alogorithme des *k-moyennes*, par exemple). $c_i(t)$ représente le $i^{\text{ème}}$ descripteur $\mathbf{x}_i(t)$ extrait à l'instant $t$ de $v$. Notons que le nombre total de mots extraits à chaque instant peut varier au cours d'une même vidéo. Par soucis de lisibilité et en l'absence d'ambiguïté, nous adoptons la notation simplifié $c \in v$ pour représenter un mot $c_i(t)$ extrait au temps $t$ de la vidéo $v$.

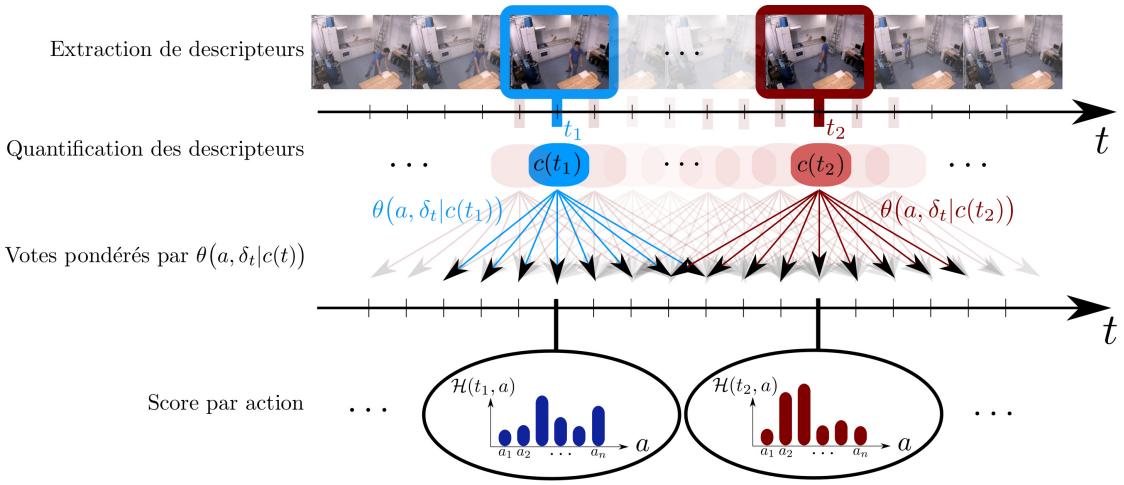



FIGURE 3.1 – Illustration d'un algorithme de détection d'actions par transformée de Hough. Sur cette figure sont représentés les votes engendrés aux temps $t_1$ et $t_2$ ainsi que l'accumulation des votes pour chaque action à ces instants.

2. **Processus de votes** : Chaque mot $c$ vote avec un poids $\theta(a, \delta_t, c)$ pour la présence d'une action $a$ sur un intervalle centré en $t + \delta_t$.

Le score final, appelé *score de Hough* est alors donné par l'équation :

$$\mathcal{H}(t, a) = \sum_{(c,t')} \theta(a, t' - t, c) \tag{3.1}$$

La détection d'une action peut ensuite être faite par seuillage sur ce score ou en recherchant l'action maximisant $\mathcal{H}(t, a)$. Dans ce dernier cas, l'action estimée $\hat{a}(t)$ par un tel algorithme s'écrit :

$$\hat{a}(t) = \arg\max_a \mathcal{H}(t, a) \tag{3.2}$$

Tous les mots de tous les instants votent pour toutes les actions sur une fenêtre temporelle $[-M, M]$, générant un score de vraisemblance de présence des actions $\mathcal{H}(t, a)$ pour chaque instant $t$ de $v$ et action $a \in \mathcal{Y}$. Ce paradigme est illustré Figure 3.1.

L'apprentissage d'un modèle de Hough réside alors principalement dans l'apprentissage des poids $\theta(a, \delta_t, c)$.

LEIBE, LEONARDIS et SCHIELE proposent une estimation relativement intuitive de ces poids dans un *Modèle de Formes Implicites (ISM)* [120]. Ils définissent ce poids comme étant la probabilité, sachant qu'un mot $c$ a été extrait au temps $t$, que l'activité $a$ soit présente à $t + \delta_t$ :

$$\theta_{ISM}(a, \delta_t, c) = \mathcal{P}(a, \delta_t \mid c)$$

Cette dernière quantité est estimée sur l'espace d'entraînement par :

$$\mathcal{P}(a, \delta_t \mid c) \approx \frac{N(a, \delta_t, c)}{N(c)}$$

$N(a, \delta_t, c)$ représente le nombre d'occurrences dans toutes les vidéos de mots $c$ à une distance $\delta_t$ du centre de l'action, limité aux exemples correspondant à l'action $a$. $N(c)$ est le nombre d'occurrences du mot $c$ sur l'ensemble des exemples d'entraînement $\mathcal{X}_E$.

Afin de pondérer l'importance des différents mots extraits, Maji et Malik introduisent, dans MMHT [121] un coefficient $\lambda_c$ pour chaque mot $c$. Ce coefficient donne plus ou moins d'importance au mot $c$ en fonction de son pouvoir discriminant :

$$\theta_{MMHT}(a, \delta_t, c) = \lambda_c \theta_{ISM}(a, \delta_t, c) = \lambda_c \mathcal{P}(a, \delta_t \mid c)$$

Les poids $\lambda_c$ sont appris simultanément au travers d'un processus d'optimisation similaire à un SVM.

Pour Zhang et Chen, dans [122], il s'agit de pondérer chacun des exemples de la base d'entraînement par un poids $\lambda_i$ appris de façon discriminative. Les poids $\theta$ deviennent alors :

$$\theta_{ISK}(a, \delta_t, c) = \sum_i \lambda_i \times \mathcal{P}_i(a, \delta_t \mid c)$$

avec $\mathcal{P}_i$ estimée sur le $i^{\text{ème}}$ exemple $v_i$.

Wohlhart, Schulter, Kostinger et al. pondèrent quant à eux le déplacement temporel. Ils introduisent pour cela un coefficient $\lambda_{\delta_t}$ appris également au travers d'un SVM. On a alors :

$$\theta(a, \delta_t, c) = \lambda_{\delta_t} \times \mathcal{P}(a, \delta_t \mid c)$$

Chacune de ces méthodes optimise le poids des votes relativement au pouvoir discriminant respectivement du mot extrait, de l'exemple considéré ou de l'écart temporel $\delta_t$.

La transformée de Hough fortement optimisée (*DOHT* pour Deeply Optimized Hough Transform), introduite par Chan-Hon-Tong, Achard et Lucat dans [63], définit une méthode dans laquelle la fonction $\theta_{DOHT}$ est optimisée relativement aux pouvoirs discriminants des actions $a$, des déplacements temporels $\delta_t$ et des mots $c$ conjointement. Avec $a^*_v(t)$ (ou $a^*$ en l'absence d'ambiguïté) l'action effectivement présente à l'instant $t$ de

l'exemple $v$, les auteurs proposent l'optimisation suivante pour les poids $\theta$ :

$$\min_{\theta \geq 0, \xi \geq 0} \Big( L_{reg}(\theta) + C \times L_{data}(\xi) \Big)$$
$$\text{sous constraintes :}$$
$$\begin{cases} \forall v \in \mathcal{V}, t', a \neq a_v^*(t) : \\ \quad \sum_{(c,t') \in v} \begin{pmatrix} \theta(a_v^*(t), t - t', c) \\ -\theta(a, t - t', c) \end{pmatrix} + \xi(t') \geq 1 \\ \forall a, c, \delta_{t1}, \delta_{t2} : \\ \quad \begin{cases} \delta_{t1} \leq \delta_{t2} \leq 0 \Rightarrow \theta(a, \delta_{t1}, c) \leq \theta(a, \delta_{t2}, c) \\ 0 \leq \delta_{t1} \leq \delta_{t2} \Rightarrow \theta(a, \delta_{t1}, c) \geq \theta(a, \delta_{t2}, c) \end{cases} \end{cases} \tag{3.3}$$

La première contrainte impose qu'un mot vote de façon plus importante pour l'action effectivement présente à l'instant $t$ que pour les autres actions. La seconde force une cohérence temporelle en imposant une décroissance des votes avec la croissance du déplacement temporel $\delta_t$. Il s'agit de faire voter un mot plus fortement pour les instants proches de son instant d'extraction afin de maintenir une cohérence temporelle au sein des votes. Les fonctions $L_{reg}$ et $L_{data}$ sont respectivement des fonctions de *régularisation* et d'*attachement au données*. Le paramètre $C$ pondère l'importance relative de ces deux fonctions.

**Reformulation du DOHT** Lorsque la taille des vidéos à segmenter augmente, la contrainte de décroissance temporelle des votes engendre rapidement un nombre de contraintes trop important pour une résolution dans un temps raisonnable. Pour pallier ce problème, les auteurs proposent une version simplifiée de la formulation DOHT sous la forme d'une décomposition pyramidale.

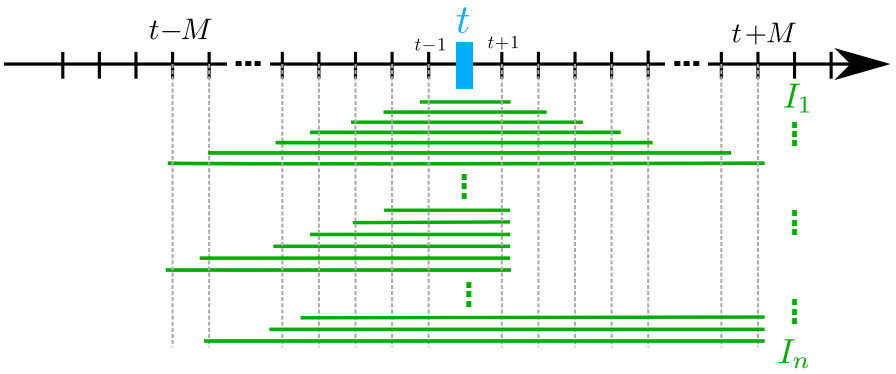
Pour cela, avec $M$ le déplacement temporel maximal considéré, on définit $\mathcal{J}^{\text{complet}}$ l'ensemble des intervalles $I$ contenant 0 et inclus dans $[-M, M]$. Ces intervalles sont représentés sur la Figure 3.2a.

Les auteurs [63] ont montré que le problème d'optimisation des poids est alors équivalent à trouver une fonction $w() > 0$ satisfaisant :

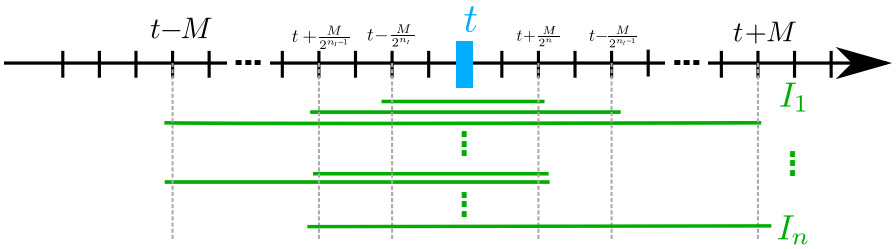
$$\theta(a, \delta_t, c) = \sum_{I \in \mathcal{J}^{\text{complet}}} w(a, I, c) \times \chi_I(\delta_t) \tag{3.4}$$

avec $\chi_I$ une fonction caractéristique définie par

$$\forall t, I : \ \chi_I(t) = \begin{cases} 1 & t \in I \\ 0 & t \notin I \end{cases}$$




(a) Ensemble complet des intervalles $\mathcal{J}^{complet}$ définis au temps $t$ :




(b) Sous-ensemble considéré dans la version du DOHT approximé $\mathcal{J}^{partiel}$

FIGURE 3.2 – Représentation des intervalles dans la formulation du DOHT

Le problème définit en 3.3 s'exprime alors sous la forme suivante :

$$\min_{w,\xi}\left[\|w\|_2^2 + C \times \left(\sum_{v,t,a} g\left(\sum_{(c,t')\in v, I\in\mathcal{J}^{complet}} (w(a^*(t),I,c)\chi_I(t-t') - w(a,I,c)\chi_I(t-t')) - 1\right)\right)\right] \tag{3.5}$$

$g$ est la fonction telle que $g(u) = \max(0,u)$.

Pour mettre en avant l'équivalence avec un SVM, introduisons la matrice $Q^t$ de taille $\left|\mathcal{J}^{complet}\right| \times K$ dont chaque élément $Q^t_{I,c}$ est le nombre de mot $c$ extraits sur l'intervalle $I$ translaté de $t$. Formellement, $Q^t_{I,c} = |\{(c,t') \in v | t - t' \in I\}|$. On appelle cette matrice *carte de présence*.

Pour des raisons mathématiques, afin de faire apparaître un produit scalaire, on introduit la fonction $\phi(I,c)$ qui associe à chaque couple $(I,c)$ un unique entier appartenant à $\left\{1,\ldots,\left|\mathcal{J}^{complet}\right| \cdot K\right\}$. Cette fonction permet de définir $q^t$ le vecteur de taille $\left|\mathcal{J}^{complet}\right| \cdot K$ vérifiant $q^t_{\phi(I,c)} = Q^t_{I,c}$.

Soit $W^a$ le vecteur tel que ${W^a}_{\phi(I,c)} = w(a,I,c)$. On a alors

$$\sum_{(c,t)\in v}\sum_{I\in\mathcal{J}^{\text{full}}} w(a,I,c)\chi_I(t-t') = \langle W^a | Q^t \rangle \tag{3.6}$$

avec $\langle|\rangle$ le produit scalaire canonique. Cela implique :

$$\sum_{(c,t')\in\mathcal{V},I\in\mathcal{J}} ((w(a^*(t),I,c) - w(a,I,c))\chi_I(t-t')) = \langle (W^{a^*(t')} - W^a) | Q^{t'} \rangle \tag{3.7}$$

En fusionnant cette dernière équation avec l'équation 3.5, on en déduit que résoudre l'équation 3.3 est équivalent à résoudre :

$$\min_{w,\xi}\left(\|w\|_2^2 + C \times \left(\sum_{v,t',a} \max\left(0, \langle (W_{a^*_n(t')} - W_a), Q^{t'}\rangle - 1\right)\right)\right), \tag{3.8}$$

Cette nouvelle formulation ressemble fortement à l'équation 2.18 du SVM structurel. Ainsi, les auteurs montrent que l'optimisation des poids de vote $\theta(a,\delta_t,c)$ peut se faire au travers de l'apprentissage d'un SVM sur les cartes de présences $Q^t$.

Cet apprentissage est illustré sur la Figure 3.3.

Dans [124], les auteurs mettent en avant le fait que l'apprentissage peut être accéléré en remplaçant l'espace complet des intervalles $\mathcal{J}^{\text{complet}}$ par un sous-ensemble $\mathcal{J}^{\text{partiel}}$ d'intervalles. Cela permet de réduire le nombre de contraintes lors de l'apprentissage. Cet ensemble est défini de telle sorte que la granulité augmente en approchant de 0. Plus formellement, ces intervalles sont définis comme : $\mathcal{J}^{\text{partiel}} = \left\{[-2^{-\alpha}M, 2^{-\beta}M]\right\}$ avec $\alpha,\beta \in \{0,\ldots,n_I,+\infty\}$. Le terme $+\infty$ est ajouté afin d'inclure la borne $\delta_t = 0$. Les bornes sont alors d'autant plus nombreuses que l'on s'approche de 0. Ces intervalles sont

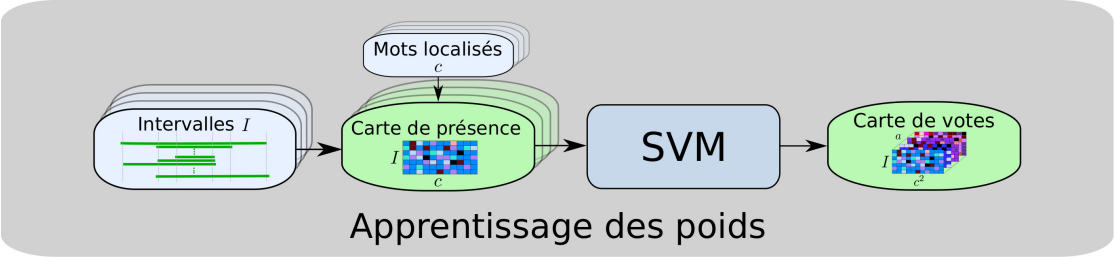



FIGURE 3.3 – Apprentissage des poids au sein du DOHT

représentés sur la Figure 3.2b. Ce DOHT, appliqué avec de tels intervalles, est appelé *DOHT approximé*.

Les performances obtenues par le *DOHT approximé* sont très proches de celles du *DOHT complet* et l'apprentissage est moins coûteux en ressources [63]. Nous utilisons cette approche dans la suite de cette thèse.

Notons qu'en combinant les équations 3.1, 3.4 et 3.6, on obtient que le score de Hough peut être directement obtenu à partir des poids $W^a$ et des cartes présence $Q^t$ :

$$\mathcal{H}(t,a) = \sum_{(c,t')} \sum_{I \in \mathcal{J}^{\text{complet}}} w(a,I,c) \times \chi_I(\delta_t) = \langle W^a | Q^t \rangle . \tag{3.9}$$

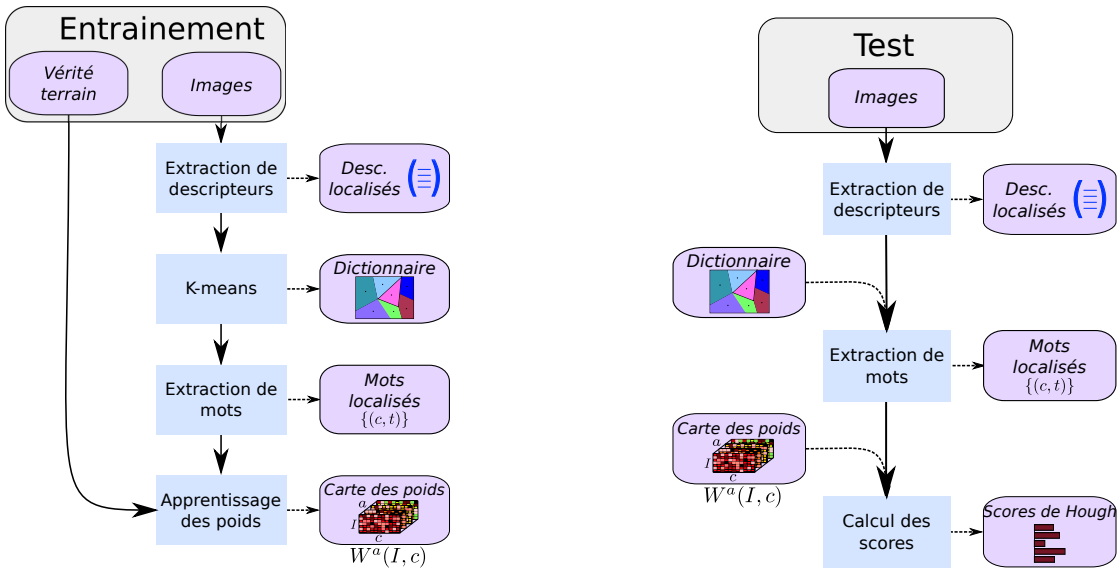
La Figure 3.4 illustre l'algorithme du DOHT en phase d'entraînement ainsi qu'en phase de test. En sortie de l'algorithme, on obtient un score de Hough $\mathcal{H}(t,a)$ pour chaque classe $a$ à chaque instant $t$. L'association de l'instant $t$ à une classe est alors obtenue à l'aide de l'équation 3.2

## 3.2 Paradigme de fusion d'informations au sein du paradigme de Hough

Dans le papier original [124], l'algorithme utilisant le DOHT pour la reconnaissance d'actions humaines a été appliqué uniquement sur des données représentant la forme du squelette 3D (coordonnées des articulations). Ces données sont dépendantes du type de capteur utilisé et ne sont pas toujours disponibles de façon fiable. Par ailleurs, il est rare de pouvoir couvrir entièrement une zone à observer à l'aide d'un unique capteur et le nombre de sources d'information requis dépend de l'application ainsi que de la forme de cette zone.

Dans la suite, nous nous intéressons à la problématique de la reconnaissance d'actions à partir de sources multiples d'information sous deux angles :

— Nous étudions l'utilisation, au sein de l'algorithme DOHT, de données de natures différentes,




(a) Phase d'entrainement (b) Phase de test

FIGURE 3.4 – Illustration de l'algorithme DOHT

— Nous nous intéressons à la prise en compte d'un nombre inconnu et variable de capteurs. Ce deuxième cas se justifie notamment en cas d'indisponibilité temporaire d'une source d'information.

Dans [7], PENG, WANG, WANG et al. mettent en avant trois stratégies de fusion d'informations au sein d'un paradigme de type *Sac de mots*. La première se fait au niveau des descripteurs, la seconde au niveau de la représentation et la dernière au niveau des scores. Nous proposons d'adapter cette formulation au paradigme de la transformée de Hough, et plus particulièrement du DOHT, pour la segmentation et reconnaissance d'actions.

### 3.2.1 Fusion niveau extraction

Le premier niveau de fusion que nous proposons se fait au niveau de l'extraction des descripteurs. Nous l'appelons *fusion niveau descripteurs* ou *fusion niveau extraction* dans la suite de cette thèse.

Il s'agit du niveau le plus bas (sémantiquement) que nous proposons et décrivons ici. Cette combinaison consiste en la concaténation des vecteurs décrivant les caractéristiques extraites au cours de la vidéo $v$. Rappelons que l'algorithme de reconnaissance d'actions commence par l'extraction de descripteurs locaux. Ceux-ci sont obtenus par une première étape de détection de points d'intérêt (ou de trajectoires) et une caractérisation de ceux-ci (section 2.2). Plusieurs caractérisations des mêmes points peuvent être envisagées, provenant dans certains cas de plusieurs modalités (image, profondeur, mouvement, *etc.*). On arrive ainsi à un ensemble de descripteurs $\{\mathbf{x}_{i,l}(t)\}$ avec $i \in \{1,\dots,n(t)\}$

où $n(t)$ est le nombre de descripteurs issus à l'instant $t$ et $l \in \{1,\dots,L\}$ les différentes caractérisations retenues. Un descripteur combinant ces informations est défini comme

$$\mathbf{x}_i^{\text{fusion}} = \begin{bmatrix} \mathbf{x}_{i,1}(t) \\ \vdots \\ \mathbf{x}_{i,L}(t) \end{bmatrix}. \tag{3.10}$$

La quantification (à partir d'un *k-means* par exemple) ainsi que la suite de l'algorithme DOHT sont ensuite appliqués à ce nouveau descripteur. En d'autres termes, cette fusion est équivalente à la création d'un nouveau descripteur contenant les informations de plusieurs modalités différentes. Il se veut plus riche en information que chacun des descripteurs fusionnés pris indépendamment. Cette fusion d'informations est illustrée sur la Figure 3.5.

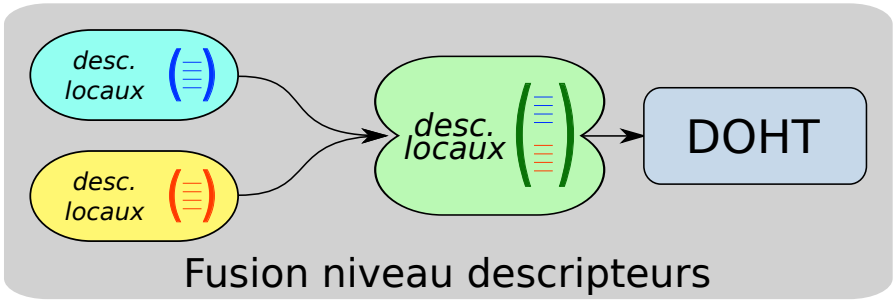



Figure 3.5 – Fusion d'informations niveau descripteurs

Par exemple, si on applique cette fusion au moment de l'extraction des trajectoires denses [16] de Wang, Kläser, Schmid et al., les informations à fusionner sont les différentes caractéristiques définies autour de ces trajectoires, à savoir HOG [32], HOF [33], Forme de trajectoire et MBH [125] (cf Figure 2.5 et plus généralement la section 2.2.1). Plus particulièrement, lorsque l'on fusionne la forme de la trajectoire [16] avec un descripteur HOG, on obtient alors un vecteur décrivant à la fois l'aspect temporel et l'apparence spatiale autour du point suivi.

Notons qu'un choix judicieux des informations à fusionner est essentiel dans ce paradigme. En effet, la concaténation d'informations extraites autour d'une même trajectoire semble pertinente. Par contre, une combinaison d'informations en provenance de différents capteurs est très délicate. sans mise en correspondance.Prenons l'exemple d'une extraction de points d'intérêts sur deux caméras filmant une même scène. La concaténation de deux descripteurs extraits à partir de points physiquement différents pris aléatoirement engendrerait un vecteur contenant une information difficilement interprétable.

Ainsi, ce niveau de fusion n'est applicable à une combinaison de capteurs qu'à la condition contraignante d'effectuer un pré-traitement pour la mise en correspondance des informations propres à chacun d'eux. Nous limiterons donc ce niveau de fusion à la fusion de descripteurs locaux, provenant d'un même capteur.

### 3.2.2 Fusion niveau votes

Le deuxième type de fusion d'informations que nous proposons se situe au cœur de l'algorithme DOHT : au niveau de l'apprentissage des poids des votes. L'idée est de conserver un descripteur propre à chaque modalité que l'on veut fusionner, mais d'apprendre les poids de vote $\omega(a, \delta_t, c)$ de façon globale en considérant toutes les modalités.

Ainsi, pour chaque modalité, des points (ou trajectoires) sont extraits et caractérisés indépendamment. Ceci amène à un ensemble de descripteurs $\{\mathbf{x}_{i,l}(t)\}$ avec $i \in \{1,\dots,n(t)\}$ où $n(t)$ est le nombre de descripteurs issus à l'instant $t$ qui varie en fonction de la modalité $l \in \{1,\dots,L\}$. L'algorithme des k-moyennes appliqué sur chacune des modalités amène à des mots $\{c_i^l(t)\}$ avec $i \in \{1,\dots,n(t)\}$. Ces mots sont accumulés par modalité dans les intervalles $I \in \mathcal{J}^{\text{partiel}}$ tels que définis section 3.1 et amènent donc à une carte de présence par modalité $Q_{I,c}^{t,l}$. Une nouvelle carte de présence, accumulant toutes les cartes de présence des modalités est construite :

$$Q_{I,c}^{t\ \text{fusion}} = [Q_{I,c}^{t,1},\dots,Q_{I,c}^{t,L}], \tag{3.11}$$

La carte de votes finale sera apprise à partir de cette nouvelle carte au travers du SVM (équations 3.8).

Notons que puisque l'optimisation des poids est faite, entre autres, relativement aux mots $c^l$, l'algorithme apprend à accorder plus d'importance aux mots les plus discriminants et donc aux modalités les plus adaptées à la détection d'actions dans ce contexte. La Figure 3.6 illustre la combinaison de descripteurs à ce niveau.

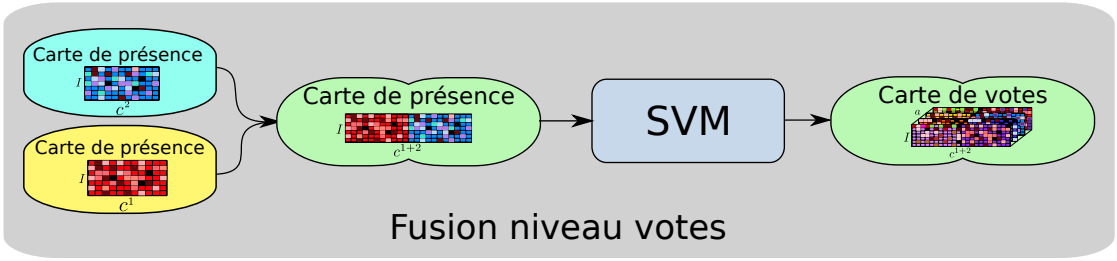



Figure 3.6 – Fusion d'informations niveau votes.

Par ailleurs, contrairement à la fusion niveau descripteurs, ce niveau de fusion admet une robustesse vis-à-vis d'une perte d'information.

En effet, un manque d'information de la part d'un capteur se traduit par un appauvrissement de la carte de présences $Q^{t,l}$de la modalité $l$ correspondante. Les composantes $Q_{I,c}^{t,l}, \forall I$ sont impactées, mais les descripteurs disponibles génèrent tout de même un score $\mathcal{H}(t,a)$ au travers de l'équation 3.9. La diminution du nombre de descripteurs générant un vote engendre des scores inférieurs pour toutes les actions. L'ordre des scores est peu impacté, le choix de l'action présentant le plus grand score reste alors une bonne estimation de l'activité présente.

### 3.2.3 Fusion niveau scores

Le dernier niveau de fusion que nous proposons se fait en aval de l'algorithme DOHT. Il s'agit ici d'apprendre à classifier les activités à partir des scores générés par l'algorithme DOHT appliqué à chacun des descripteurs indépendamment.

Chaque modalité $\{l \in \{1,2,\ldots,L\}\}$ à fusionner est donnée en entrée d'un algorithme DOHT et génère une carte de score $\mathcal{H}_l(t,a)$ propre pour tous les instants de la vidéo. On génère ensuite un vecteur $\mathcal{H}_{\text{fusion}}(t,a)$ qui viendra en entrée d'un SVM multi-classes que l'on nommera *SVM de fusion* :

$$\mathcal{H}_{\text{fusion}}(t,a) = \begin{bmatrix} \mathcal{H}_1(t,a) \\ \vdots \\ \mathcal{H}_L(t,a) \end{bmatrix}$$

A ce niveau de fusion, les poids des DOHT étant appris indépendamment, l'optimisation est faite à un niveau local (et non global comme dans la fusion niveau votes). C'est le SVM de fusion qui pondère ensuite chacune des composantes en fonction de son pouvoir discriminant vis-à-vis de chacune des actions.

Si ce paradigme ne présente, par sa nature, pas de robustesse vis-à-vis d'une perte de donnée (car le vecteur en entrée du SVM serait fortement modifié), il peut cependant être utilisé pour combiner des informations de sources totalement différentes. Ce niveau de fusion est illustré sur la Figure 3.7.

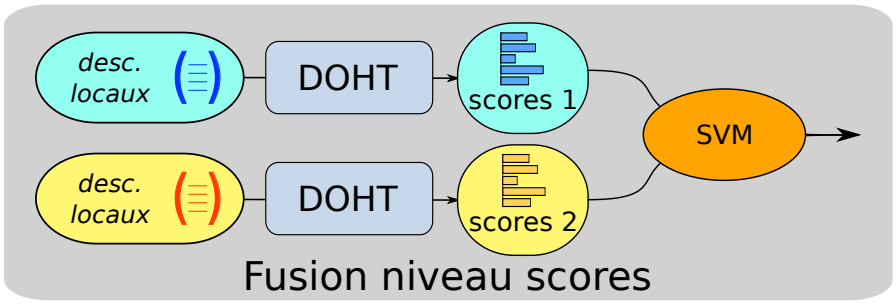



FIGURE 3.7 – Fusion d'informations niveau carte de présence

## 3.3 Evaluation sur le jeu de données TUM

### 3.3.1 Présentation des données

Afin de mettre en avant l'apport de la fusion de descripteurs sur les performances de l'algorithme, nous évaluons nos trois paradigmes de fusion sur la base de données *TUM Kitchen dataset* [126]. Cette base de données a été choisie afin de pouvoir comparer les performances avec celles obtenues dans la version originale du DOHT [63].

Ce jeu de données est composé de 19 vidéos dans lesquelles les personnes dressent

une table soit de façon naturelle, soit de façon plus artificielle selon les exemples. L'étiquetage de ces vidéos est effectué pour chaque image parmi les classes *"Se déplacer en portant un objet"*, *"Atteindre un objet"*, *"Prendre un objet"*, *"Poser un objet"*, *"Relâcher un objet"*,*"Ouvrir une porte"*, *"Fermer une porte"*, *"Ouvrir un tiroir"* et *"Fermer un tiroir"*.

Les vidéos ont été capturées à l'aide de 4 caméras disposées de façon homogène autour de la scène. Elles durent entre 1 et 2 minutes et ont été enregistrées sans interruption entre les actions. Enfin, l'ordre des actions varie en fonction des vidéos. En plus des images, les auteurs fournissent les poses des participants pour chaque image des vidéos et des informations en provenance de capteurs magnétiques et de puces RFID sont fournies. Celles-ci représentent l'état (fermé ou ouvert) des portes de placard et des tiroirs. L'emplacement des caméras est illustré sur la Figure 3.8 et un exemple sur chacune des vues est représenté sur la Figure 3.9.

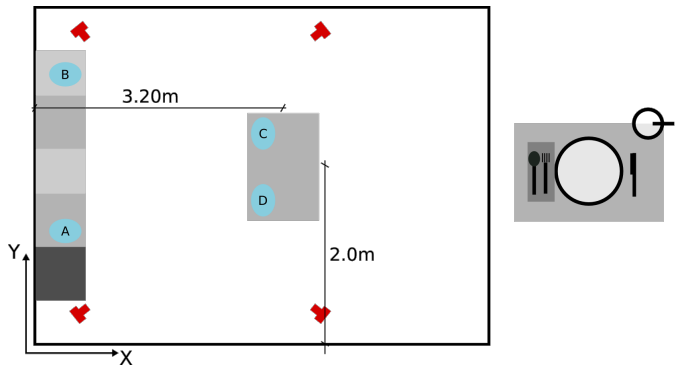



FIGURE 3.8 – Positionnement des caméras pour le jeu de données TUM. En rouge les caméras présentes autour de la scène et en bleu les capteurs RFID. Image issue de [126]

Les participants se déplacent entre le plan de travail et la table afin d'y déposer divers objets tels qu'une assiette, des couverts et un plateau. Durant ces actions, les sujets sont alternativement visibles de face par une (ou deux) des quatre caméras. De fait, lorsqu'un objet est attrapé, un tiroir ouvert, ou que plus généralement une action est faite, la personne occulte avec son corps l'action du point de vue des caméras placées derrière elle. Ainsi, la caméra la plus adaptée à la détection d'actions n'est pas la même tout au long d'une vidéo. Par exemple, sur la Figure 3.9, l'action est visible correctement sur les vues 0 et 1 et est occultée sur les vues 2 et 3.

La vérité terrain de ce jeu de données est fournie indépendamment pour chacune des deux mains. Pour une comparaison juste à l'état de l'art et dans un souci de cohérence, nous suivons [127] et [124] en ne considérant que les étiquettes associées à la main gauche.

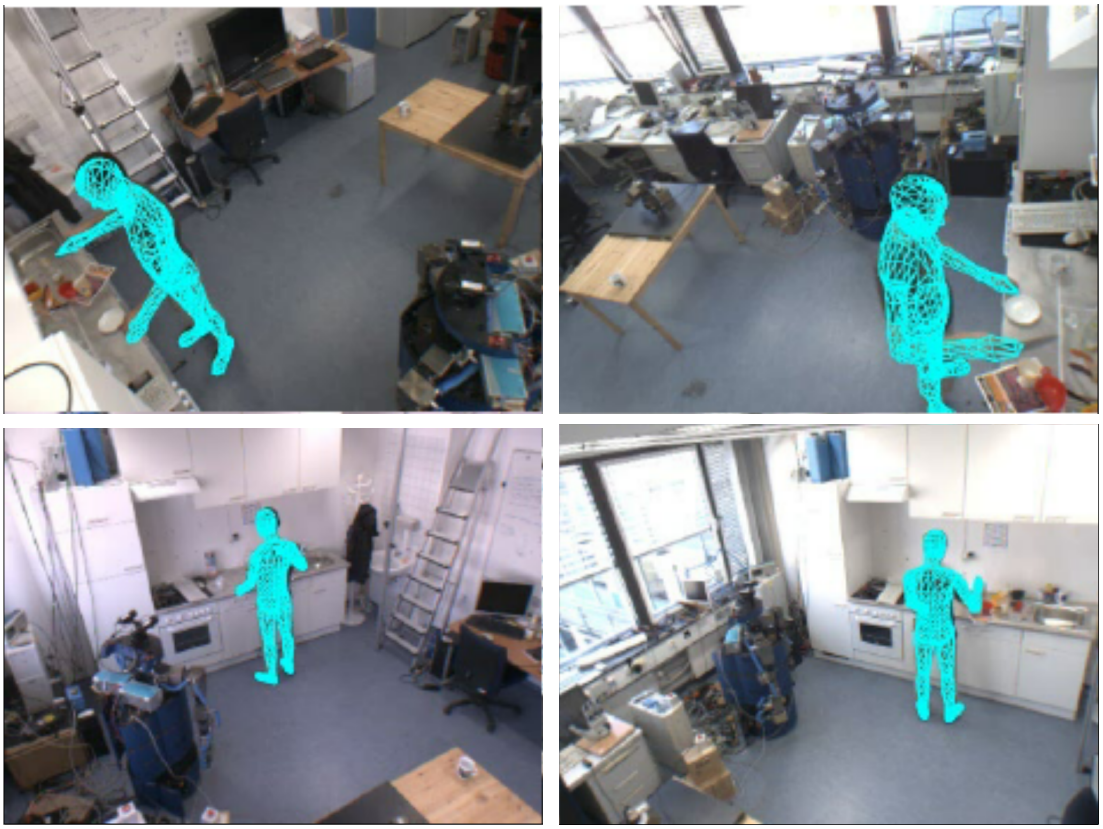


FIGURE 3.9 – Images du jeu de données TUM

### 3.3.2 Résultats obtenus

**Mono-descripteur**

Tout d'abord et afin de pouvoir évaluer l'apport de la fusion de descripteurs sur le taux de reconnaissance, nous évaluons les performances du DOHT sur chaque descripteur pris indépendamment. Pour l'extraction de descripteurs issus des images, nous utilisons l'algorithme proposé par WANG, KLÄSER, SCHMID et al. dans [16] décrit dans la section 2.2.1, en conservant les paramètres originaux. Nous considérons donc tour à tour :

1. la forme de la trajectoire extraite (TS) sur une fenêtre temporelle de 15 images,
2. les histogrammes de gradients (HOG) autour de ces trajectoires avec 8 composantes pour 8 directions de gradient uniformément réparties,
3. les histogrammes de flux optique (HOF) autour de ces trajectoires avec 9 composantes pour 8 directions de flux optique plus une composante pour les points immobiles.

Les paramètres de l'algorithme en lui même ont été conservés tels que dans la

publication originale [63], à savoir la demi-longueur intervalles de $M = 50$ et un nombre de frontière $n_i = 4$. Nous avons alors évalué les performances de l'algorithme pour différents nombre $K$ de clusters et différentes valeurs $C$ du SVM d'apprentissage. Les paramètres retenus pour les résultats présentés dans cette section sont $K = 3000$ et $C = 2$. L'influence du paramètre $C$ est décrite plus précisément dans la suite.

Les performances sont exprimées en pourcentages de bonnes détections et sont définies comme étant le ratio entre le nombre d'images correctement reconnues et le nombre d'images total de l'ensemble de test. Nous choisissons cette métrique car elle est utilisée dans l'état de l'art pour ce jeu de données et elle nous permettra donc de comparer ces résultats à ceux des travaux précédents.

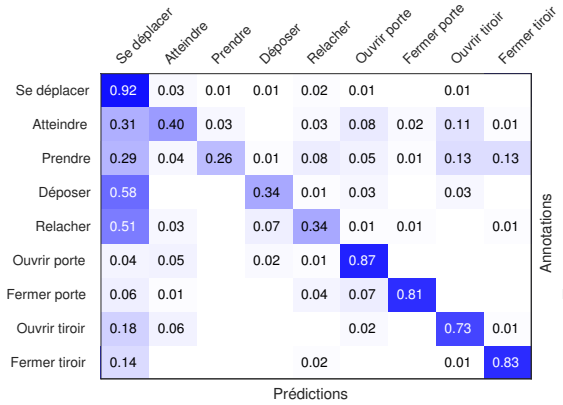
| • | TS | HoG | HOF | Pose |
|---|---|---|---|---|
| Vue 0 | 75.0 | **81.8** | 79.6 | 81.5 |
| Vue 1 | 72.6 | **81.3** | 76.5 | |
| Vue 2 | 70.5 | **80.5** | 74.7 | |
| Vue 3 | 73.5 | **77** | 74.5 | |

Tableau 3.1 – Taux de reconnaissance sur le jeu de données TUM [126] pour chaque descripteur pris indépendamment.

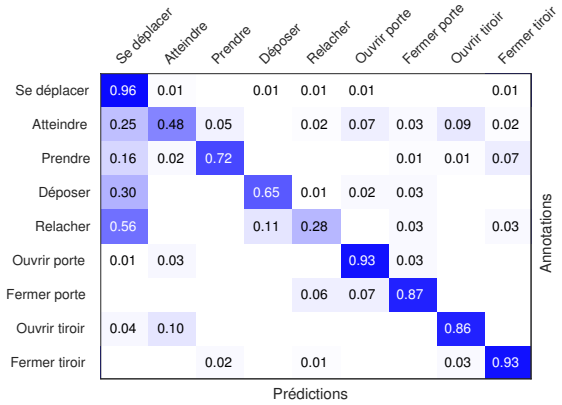
Sur le Tableau 3.1, présentant les résultats obtenus pour chaque modalité indépendamment, on constate que la caractérisation par HOG montre des meilleurs résultats que pour les autres descripteurs, sur toutes les vues. Ces résultats semblent indiquer que, dans le cas de ce jeu de données, les informations de contexte visuel (HOG) sont plus informatives que les informations de mouvement (TS et HOF). Une explication possible à ce phénomène est qu'un point d'intérêt extrait au niveau de la main aura un mouvement similaire pendant les actions *"Se déplacer"* et *"Atteindre un objet"* par exemple. Les trajectoires de ce point ne seront alors pas discriminantes pour ces actions et cela rend la différenciation de ces classes plus difficile. À l'inverse, puisque le contexte spatial autour de ce même point sera très différent en présence d'un objet ou en l'absence de ce même objet, un descripteur prenant en compte ce contexte pourra aider à différencier correctement les actions.

Les résultats pour cet exemple précis semblent corroborer cette interprétation. En effet l'action *"Prendre un objet"* n'est reconnue que dans 26% des cas lorsque le descripteur TS est considéré, elle est alors confondue plus de 10% des fois avec *"Ouvrir un tiroir"*, *"Fermer un tiroir"* et *"Se déplacer"*. Pour le descripteur HOG, cette classe est bien reconnue dans 72% des images et n'est confondue, plus de 10% des fois, qu'avec la classe *"Se déplacer"*. On observe un phénomène similaire avec la classe *"Déposer"*. Les matrices de confusion de ces deux descripteurs pour les vues 0 et 2 sont disponibles sur la Figure 3.10.

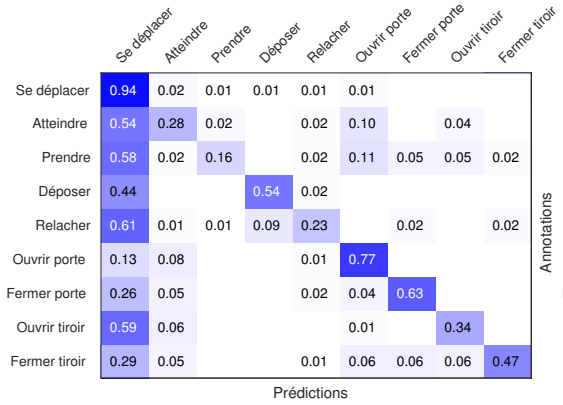
Sur le Tableau 3.1, on note également des différences de performances selon la vue considérée. Cela est dû aux occultations non-homogènes au travers des vues et plus particulièrement aux occultations de la main gauche de la personne. Comme on peut le voir sur la Figure 3.9, la position de la caméra numéro 3 est propice aux occultations



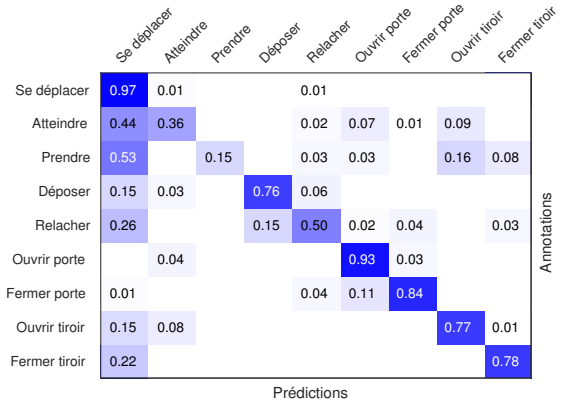

(a) Descripteur de Trajectoire (TS) seul sur la vue 0




(b) Descripteur HOG seul sur la vue 0




(c) Descripteur de Trajectoire (TS) seul sur la vue 2




(d) Descripteur HOG seul sur la vue 2

FIGURE 3.10 – Matrices de confusion sur le jeu de données *TUM* [126] en évaluation mono-descripteur sur les vues 0 et 2

de cette main lorsque les participants sont devant le plan de travail. Or, les classes sont définies par rapport à la main gauche et une plus grande variété d'actions a lieu justement devant le plan de travail (*"Ouvrir/Fermer un tiroir"*, *"Ouvrir/Fermer un placard"*, *"Saisir et Atteindre un objet"* contre *"Déposer un objet"* et *"Relâcher un objet"* du côté de la table). Ainsi, les occultations fréquentes dû à cette configuration font que la vue 3 ne peut pas extraire de descripteurs pertinents dans la majorité des cas.

### Fusion de descripteurs

Nous avons ensuite évalué chacun des paradigmes de fusion de descripteurs présentés pour différentes combinaisons de descripteurs. Le Tableau 3.2 résume les résultats obtenus avec les 3 niveaux de fusion décrits. Notons que pour la fusion bas niveau, nous avons réalisé différentes évaluations et $K = 3000$ est, là aussi, la configuration ayant

donné les meilleurs résultats.

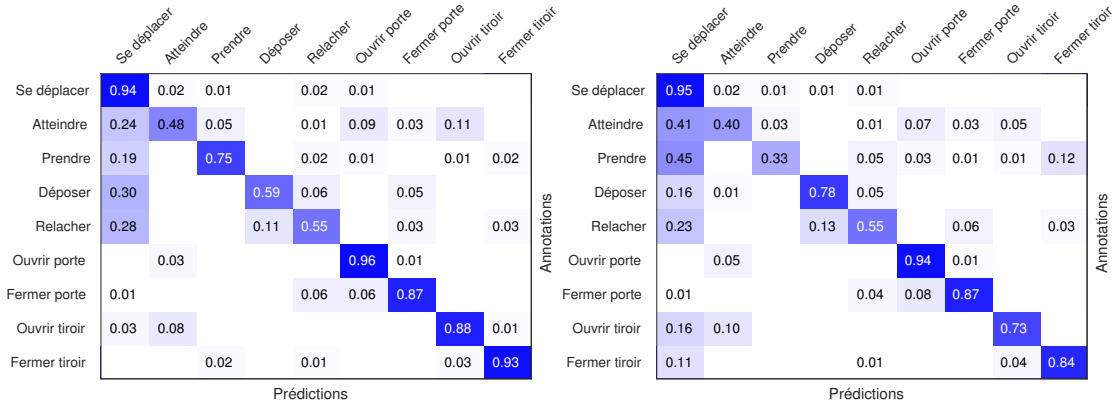
| | | TS+HOG | TS+HOF | HOG+HOF | TS+HOG+HOF |
|---|---|---|---|---|---|
| Vue 0 | Niveau desc. | **82.5** | 78.6 | 80.5 | 80.6 |
| | Niveau votes | 80.3 | 79.3 | **81.9** | 81.2 |
| | Niveau scores | 79.2 | 78.9 | 81.4 | 80.0 |
| Vue 1 | Niveau desc. | **81.7** | 76.5 | 78.2 | 78.0 |
| | Niveau votes | 80.1 | 76.6 | 80.4 | 80.0 |
| | Niveau scores | 78.2 | 79.4 | 79.4 | 78.6 |
| Vue 2 | Niveau desc. | **80.7** | 74.1 | 79.7 | 78.3 |
| | Niveau votes | 78.1 | 73.4 | 80.0 | 77.6 |
| | Niveau scores | 78.1 | 73.9 | 78.5 | 77.3 |
| Vue 3 | Niveau desc. | **77.9** | 74.5 | **77.5** | 77.9 |
| | Niveau votes | 77 | 75.0 | 77.2 | 77.2 |
| | Niveau scores | 76.9 | 75.4 | 76.4 | 76.8 |

Tableau 3.2 – Taux de détection sur le jeu de données TUM [126] avec fusion des descripteurs visuels. Les valeurs en bleu sont celles pour lesquelles les résultats sont meilleurs qu'en mono-descripteurs

Ces résultats indiquent que les performances sont améliorées notamment lorsque l'on combine le descripteur de trajectoire *TS* avec le descripteur de contexte visuel *HOG*. Cela peut s'expliquer par le fait que le premier décrit le mouvement du point au cours de la trajectoire (il est riche en informations temporelles), et le second décrit le contexte visuel, complémentaire dans la description d'un point d'intérêt. On retrouve une légère amélioration lorsque l'on combine *HOF* (évolution temporelle locale) et *HOG*, ce qui corrobore ce résultat tout en laissant supposer que l'évolution temporelle à plus long terme est plus complémentaire avec le descripteur *HOG*. Lorsque l'on combine uniquement les informations de forme de trajectoire avec le descripteur *HOF*, tous deux extraits à partir du flux optique, les résultats sont légèrement moindres, car ces deux modalités proviennent du flux optique et sont donc moins complémentaires. La Figure 3.11 montre les matrices de confusion pour les descripteurs fusionnés *TS+HOG* sur les vues 0 et 2. On constate une amélioration pour la quasi-totalité des classes, amélioration plus nette sur la vue 2. Ces matrices sont à comparer avec celles de la Figure 3.10.

Une tendance claire indiquée par la première colonne du Tableau 3.2 est que les meilleurs taux sont obtenus pour le plus bas niveau de fusion. Ainsi, le choix d'un descripteur contenant des informations complémentaires est bénéfique pour un algorithme de type Hough pour la détection d'action.

Concernant la fusion niveau scores, les résultats présentés jusqu'ici ont été obtenus avec le paramètre $C = 2$ du SVM (équation 3.8). Cette valeur a été déterminée après une succession de tests réalisés avec différentes valeurs de $C$. La Figure 3.12 montre l'évolution, pour chacun des descripteurs, des performances lorsque ce paramètre évolue.




(a) Fusion de descripteurs (TS+HOG) sur la vue 0. (b) Fusion de descripteurs (TS+HOG) sur la vue 2.

FIGURE 3.11 – Matrices de confusion sur le jeu de données *TUM* [126] en évaluation multi-descripteurs sur les vues 0 et 2

Pour la totalité des descripteurs, on constate un plateau à partir de la valeur $C = 2$. Cette insensibilité à l'hyper paramètre $C$ est une autre force de l'algorithme DOHT, puisqu'il ne nécessite pas un réglage fin de ce paramètre réglant l'importance de l'attache aux données.

Afin d'éviter un sur-apprentissage et de conserver les meilleurs résultats possibles, nous privilégions une valeur de $C$ faible donnant de bons résultats. C'est pourquoi nous fixons $C = 2$ dans la suite des expériences.

**Fusion de vues** Nous avons ensuite évalué l'apport d'une combinaison d'informations provenant de vues différentes. Pour cela, nous gardons le nouveau descripteur *TS+HOG* issu de la fusion des caractéristiques TS et HOG, puisqu'il a donné les meilleurs résultats lors de la fusion mono-visuelle.

Comme précisé précédemment, il n'y a que très peu de sens à fusionner les informations au niveau descripteur dans ce contexte car les points extraits des différentes vues ne sont pas appariés et n'ont, de plus, pas nécessairement de correspondant dans les autres vues. Le Tableau 3.3 montre les résultats obtenus lors de la fusion de deux vues et de quatre vues aux niveaux des votes et des scores.

Les résultats observés lors de la fusion d'informations en provenance de deux vues montrent une différence de complémentarité entre les configurations possibles de fusion de vues. Les couples de caméras, tels que la vue 0 et la vue 2, qui correspondent à des caméras symétriquement disposées d'un côté et de l'autre de la salle, produisent une information complémentaire permettant à l'algorithme une amélioration des scores, en particulier pour les actions fortement occultées sur chaque vue telles que *"Atteindre"*, *"Déposer"*, *"Prendre"* et *"relâcher"*. Ce constat est fait à partir de la matrice de confusion représentée en Figure 3.13

Là encore, la fusion d'informations au plus bas niveau améliore les performances de

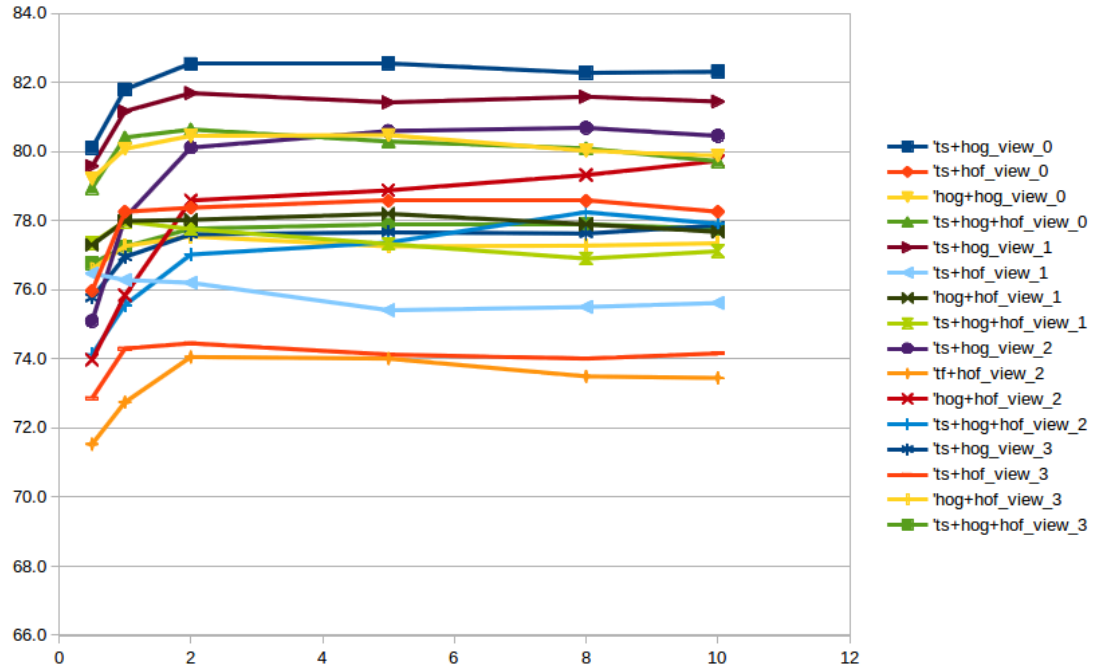



FIGURE 3.12 – Influence du paramètre C (équation 3.8) sur les résultats obtenus par fusion de descripteurs

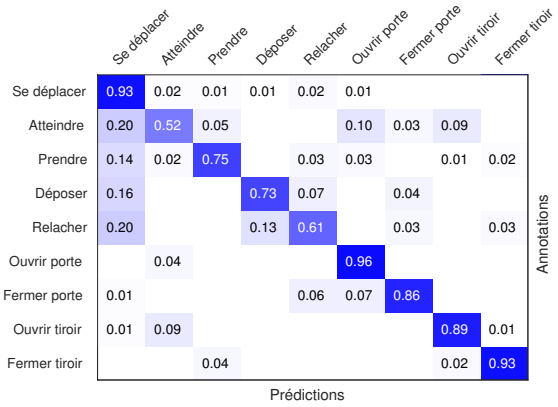



FIGURE 3.13 – Matrice de confusion après fusion au niveau votes des vues 0 et 2 sur le jeu de données *TUM* [126]

| Vues | Niveau carte de présence | Niveau scores |
|---|---|---|
| **Fusion de 2 vues** | | |
| (0 + 1) | 83.1 (+0.6) | **83.0** (+0.5) |
| (1 + 2) | 82.1 (+0.4) | 82.1 (+0.4) |
| (0 + 2) | **83.9** (+1.4) | 82.5 (+0.0) |
| (1 + 3) | 81.7 (+0) | 81.5 (-0.2) |
| (0 + 3) | 83.4 (+0.9) | 82.0 (-0.5) |
| (2 + 3) | 80.6 (-0.1) | 79.5 (-1.2) |
| **Fusion de toutes les vues** | | |
| all | 83.1 (+0.6) | 83.2 (+0.7) |

Tableau 3.3 – Fusion d'informations au niveau des cartes de présence. Entre parenthèse sont donnés les apports des différentes configurations comparées au descripteur TS+HOG seul

façon plus importante. C'est lorsque les poids des votes sont appris de manière globale au sein du SVM interne au DOHT que l'amélioration est la plus significative, avec un gain de 1.4 points pour la fusion des vues 1 et 2.

Lorsque l'on fusionne l'intégralité des 4 vues pour l'apprentissage des activités, on constate une amélioration moins forte que dans le cas de la fusion à 2 vues. Cette différence peut notamment s'expliquer par le grand nombre de paramètres du SVM à optimiser avec un vecteur en entrée (carte de présence) de très grande taille, présentant une redondance d'informations non négligeable. Cette sensibilité à la redondance de données est corroborée par les résultats obtenus lors de la fusion des vues 2 et 3. En effet, les occultations et mouvements visibles sur ces deux vues sont similaires et augmentent le bruit en entrée du SVM en limitant sa capacité d'apprentissage.

**Fusion de modalités** Ensuite, nous avons évalué l'apport d'une fusion de modalités. Dans l'optique de profiter du maximum d'informations nous fusionnons les informations apportées par des extracteurs très différents, à savoir des descripteurs visuels extraits autour de trajectoires denses au sein de l'image avec des descripteurs de poses générés à partir des squelettes estimés de la personne. Cela permet, par exemple, de tirer profil des informations sur le mouvement humain associées au contexte visuel en provenance du domaine image.

Pour décrire les squelettes des personnes présentes dans les vidéos, nous utilisons une description des trajectoires de chaque articulation dans un repère lié à la personne et normalisé afin d'être robuste à des différences de corpulence. Plus précisément, à partir des coordonnées de chaque articulation dans un repère lié à la caméra, nous définissons un nouveau repère lié à la personne et normalisé selon la distance entre les deux épaules. Le premier axe de ce repère est défini par le vecteur $\overrightarrow{u}$ liant le cou au centre des hanches. Le deuxième vecteur est défini comme le vecteur entre les deux épaules projeté dans un plan ortHOGonal à $\overrightarrow{u}$. Le dernier axe est défini de manière

| TS+HOG (4 vues) | Squelette | TS+HOG (4vues) + Squelette |
|---|---|---|
| 83.1 | 81.5 | 85.0 |

Tableau 3.4 – Comparaison des scores lors de la fusion des modalités visuelle et description de poses

à obtenir un repère orthonormé direct. Cette représentation est inspirée des travaux proposés par Raptis, Kirovski et Hoppe [57].

Une fois les coordonnées de chaque articulation exprimées dans ce nouveau repère, nous décrivons l'évolution d'une pose par la succession des positions des articulations sur un segment temporel de longueur $2\tau$. La trajectoire d'une articulation $j$ extraite à l'instant $t$ est décrite par la suite des positions sur l'intervalle $[t-\tau; t+\tau]$.Le paramètre $\tau$ a été déterminé expérimentalement [124] et nous conservons $\tau = 8$, soit des trajectoires d'articulation d'environ 1 seconde. Pour l'étape de quantification, nous conservons le paramètre $K = 16$ déterminé expérimentalement.

Là encore, combiner ces descripteurs de nature totalement différente ne peut se faire qu'au niveau des votes et des scores. Le taux de bonne classification obtenu dans ce cas est de 85 %, surpassant les performances obtenues précédemment, cf Tableau 3.4.

**Robustesse à une perte d'information** Un des avantages majeurs de la fusion d'informations au niveau des cartes de présence est la robustesse possible face à une absence de données lors de la phase de test. Rappelons également que cette robustesse n'existe pas pour les deux autres niveaux de fusion présentés puisqu'une absence de données conduirait à un descripteur corrompu en entrée du DOHT ou du SVM haut niveau respectivement. Cette étude se justifie par le fait que dans un contexte applicatif, la robustesse à une perte locale d'information peut être critique dans le cas d'une indisponibilité temporaire d'un capteur.

Nous évaluons cette robustesse dans deux contextes différents : la fusion multi-caméra d'une part et multi-modale d'autre part. Dans le premier cas, nous gardons l'apprentissage réalisé à partir de la totalité des vues et supprimons les informations en provenance de chacune d'entre-elle tour à tour. Dans le second cas, l'algorithme apprend avec une fusion d'informations en provenance des 4 vues ainsi que du squelette puis chacune des modalités est ignorée tour à tour.

Le Tableau 3.5 présente les résultats obtenus dans ce paradigme. Ces résultats mettent en exergue plusieurs phénomènes.

Premièrement, dans les deux cas (avec et sans information de pose à l'apprentissage), la chute des résultats en l'absence d'une des sources d'information est limitée. On a en effet une perte moyenne de 2,75 points pour le premier cas et de 3,38 points lorsque le squelette est présent lors de l'apprentissage. Ces différences de performances sont, relativement aux performances initiales, de moins de 4% pour les deux configurations.

D'autre part, ces résultats mettent en lumière la différence d'importance donnée à chacun des descripteurs. On constate en effet que les différences de performances varient avec la modalité ignorée ainsi qu'avec les modalités considérées lors de l'apprentissage.

| Test \ Apprentissage | 4 vues | 4 vues + squelette |
|---|---|---|
| Sans suppression | 83.1 | 85.0 |
| vue 0 supprimée | 77.7 | 83.1 |
| vue 1 supprimée | 80.2 | 83.3 |
| vue 2 supprimée | 82.2 | 84.5 |
| vue 3 supprimée | 81.3 | 85.0 |
| squelette supprimé | X | 72.2 |

Tableau 3.5 – Taux de bonne détection (%) sur le jeu de données TUM lors d'une perte d'information après une fusion au niveau des cartes de présence. Les données de chacune des modalités sont ignorées une à une dans tous l'espace de test.

Lorsque l'on fusionne les informations visuelles (descripteur TS+HOG) en provenance des 4 vues, la chute plus forte pour la perte des descripteurs de la caméra 0 que pour les autres nous indique qu'une plus grande importance est donnée aux informations en provenance de cette vue. Cela s'explique par l'optimisation globale des poids de vote au sein de l'algorithme (par le SVM Multi-classe) et est cohérent avec les résultats observés sur les vues indépendamment et exposés dans le Tableau 3.2.

Suivant ce raisonnement, quand les poses sont disponibles à l'apprentissage, le DOHT semble leur accorder une bien plus grande importance qu'aux descripteurs visuels, confirmant l'apport fort qu'apporte l'estimation de poses dans la reconnaissance d'actions.

### 3.3.3 Temps de calcul et latence de détection

L'algorithme proposé, associé à une fusion d'informations, montre des performances de détection supérieures à l'algorithme initial (squelette uniquement) sur le jeu de données *TUM Kitchen* [126], comme montré dans la section précédente. Dans un contexte applicatif tel que celui visé par cette thèse, il est important que cette détection soit faite en temps réel afin de pouvoir apporter une réponse au moment opportun ou de lever une alerte à temps en cas d'un comportement anormal. Cette section étudie les performances de la méthode en termes de temps de détection et de latence de l'algorithme DOHT.

**Latence de détection** Dans un premier temps, considérons la latence due au paradigme de votes temporels. Au sein du DOHT, la carte de présences $Q^t$ est construite à partir des mots extraits sur l'ensemble $\mathcal{J}^{\text{partial}}$ des intervalles de vote. Elle ne peut donc pas être entièrement calculée avant l'instant $t+M$ terminant le plus grand intervalle (cf Figure 3.2b). Ainsi, le score de Hough $\mathcal{H}(t,a)$ ne peut être calculé qu'après l'extraction des descripteurs de l'instant $t+M$ avec $M$ la demi-taille maximale des intervalles considérés. Hors temps de calculs, l'algorithme présente donc une latence de $M$ instants avant de pouvoir attribuer à un instant temporel une des classes.

Cependant, de la même façon que l'absence d'information en provenance d'un

capteur ne rendait pas l'algorithme DOHT inefficace (cf. section précédente), une information temporelle partielle génère aussi, au travers du DOHT, un score interprétable. Ainsi, il est possible d'estimer, avant la collecte totale des votes, la localisation d'une classe. En d'autres termes, puisque l'optimisation est faite pour chaque déplacement temporel $\delta_t = t' - t$ et que d'après l'équation 3.1, $\mathcal{H}(t,a)$ est la somme cumulative sur $t' \in [-M, M]$ des poids $\theta(a, t'-t, c)$ on peut définir un score de Hough partiel $\mathcal{H}^{[-M,T]}(t,a)$ calculé à partir des mots $c$ extrait sur $[t-M, t+T]$ comme

$$\mathcal{H}^{[-M,T]}(t,a) = \sum_{a \in \mathcal{Y}, t'-t \in [-M,T]} \theta(a, t'-t, c). \tag{3.12}$$

Nous étudions ici les performances de détection lorsque, une fois les poids appris sur une fenêtre de vote complète $[-M, M]$, on estime la détection à partir du score partiel $\mathcal{H}^{[-M,T]}(t,a)$.

La Figure 3.14 représente les résultats obtenus en fonction de $T$ avec plusieurs valeurs de $M$ à l'apprentissage. Notons que

— $T = M$ correspond au *DOHT approximé* complet,

— $T = 0$ consiste à ne considérer que les instants ayant eu lieu entre $t - M$ et l'instant d'extraction $t$,

— $T < 0$ consiste à ne considérer que des informations en provenance d'instants temporels antérieur à l'image classifiée. Ce dernier cas revient à prédire l'action située en $t$ avant que cet instant n'ait eu lieu.

Tout d'abord, étudions le cas $T = M$, qui correspond au dernier point de chacune des courbes. On constate que, parmi les configurations testées, celle avec $M = 20$ présente les meilleurs résultats. Cela correspond à une taille de fenêtre de vote égale à 40 images, soit 1,6 s. Pour expliquer ce résultat, observons que la très grande majorité des actions a une durée inférieure ou égale à 40 images, comme le montre la Figure 3.15 représentant la répartition des durées des actions dans ce jeu de données.

Les performances sont légèrement moins bonnes lorsque $M$ augmente. Cela est dû en grande partie à l'approximation faite pour le calcul de la contrainte de décroissance temporelle des votes. En effet, dans sa version approximée, le DOHT considère un sous-ensemble d'intervalles $\mathcal{J}^{\text{partiel}} = \{[-2^{-\alpha}M, 2^{-\beta}M]\}$ avec $\alpha, \beta \in \{0, \ldots, n_I, +\infty\}$. Lorsque $M$ augmente, la taille des intervalles augmente et la granularité de la méthode diminue donc. Cette estimation plus grossière des poids de votes mène à une baisse de précision lors de la détection d'actions. Cependant, relevons que cette baisse de performances est limitée (quelques points seulement pour des performances avoisinant les 80%). Ainsi, il n'apparaît pas capital d'optimiser le paramètre $M$ de façon précise sur ce jeu de données.

Ensuite, la Figure 3.14 montre l'évolution du pourcentage de bonnes détections lorsque l'on fait varier la latence $T$ dans un paradigme de DOHT partiel. On observe une forte augmentation des scores autour de $T = 0$ (donc autour de l'instant à classifier) puis un plateau au dessus de $T = 20$ dans toutes les configurations. Cela est la conséquence de deux effets. Tout d'abord, la décroissance des poids de vote avec la croissance des déplacements temporels $\delta_t$ donne plus d'importance aux poids proches de l'instant

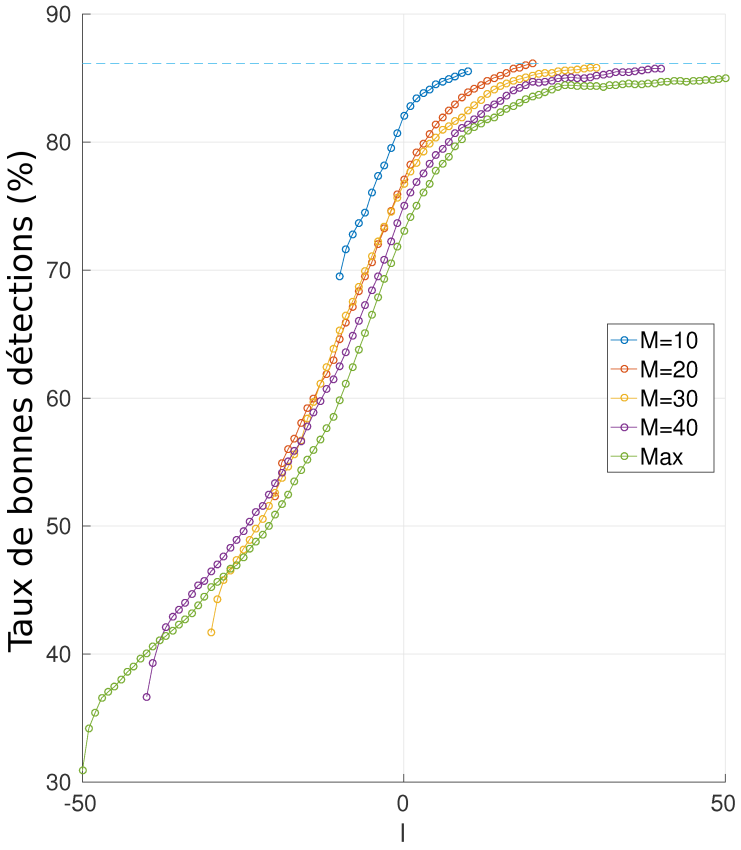



FIGURE 3.14 – Performances de l'algorithme DOHT en fonction de la latence

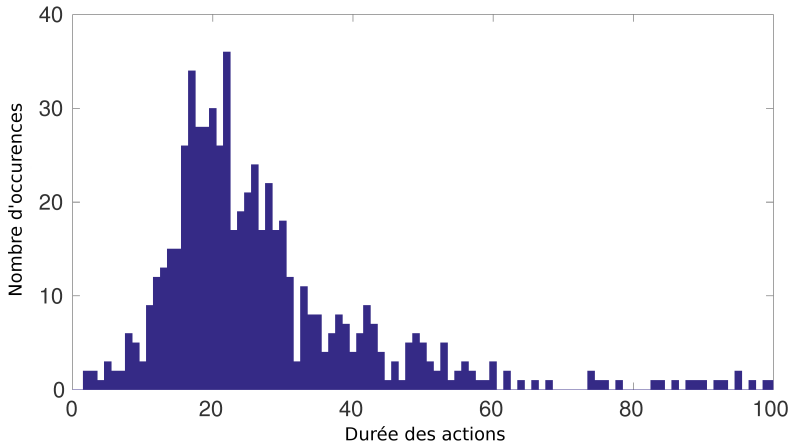



FIGURE 3.15 – Tailles des actions dans le jeu de données TUM [126]

d'extraction et rend donc ces instants plus discriminants. D'autre part, à cause de la granularité hétérogène des intervalles de vote dans le paradigme du DOHT approximé (voir Figure 3.2b page 40), le nombre de limites d'intervalles est plus important autour de $\delta_t = 0$. Ainsi, lorsque l'on s'approche de l'instant d'extraction, le pouvoir discriminant de chaque mot augmente d'autant plus vite qu'il y a de nouvelles limites d'intervalles.

Le taux de reconnaissance haut pour le cas $T = 0$ (au delà de 73% de bonnes détections pour l'ensemble des configurations et supérieure à 80% lorsque $M = 10$) permet une estimation relativement fiable dès l'instant d'extraction, estimation qui s'affinera lors des instants suivants. Le plateau observé pour $T > 20$ s'explique par le fait que la longueur maximale de la quasi-totalité des actions est de 40 images (Figure 3.15). Les mots extraits à des instants temporels situés au delà de $t + 20$ correspondent alors à une action différente et ne sont donc pas discriminant pour l'action située au temps $t$.

**Temps de calcul** A présent, intéressons-nous au temps de calcul de l'algorithme DOHT de l'extraction des descripteurs jusqu'à l'étiquetage d'un instant temporel d'une vidéo. Les auteurs de [124] montrent que la complexité de l'algorithme DOHT est en $\mathcal{O}(M, A)$ pour chaque instant $t$ des vidéos de l'espace de test lors du calcul des votes sur une fenêtre de taille $2M$ pour chaque action $a \in \mathcal{Y}$, avec $A = \text{card}(\mathcal{Y})$. Nous étudions les temps de calcul relativement aux 4 étapes de l'algorithme, à savoir

1. l'extraction de descripteurs,
2. la quantification des descripteurs,
3. la génération des cartes de présence,
4. le calcul des scores de Hough.

Pour les descripteurs $TS + HOG$ extraits à partir des trajectoires denses de WANG, KLÄSER, SCHMID et al., nous utilisons le code des auteurs associé à la publication [16]

et disponible en ligne. L'extraction sur une grille dense de ces trajectoires pourrait être optimisée (notamment par une parallélisation des calculs).

La Figure 3.16 illustre les temps mesurés en fonction de $M$. Ces temps ont été mesurés sur une machine équipée d'un processeur "Intel(R) Xeon(R) CPU E5-2687W 0 @ 3.10GHz" en utilisant 16 coeurs lors de la génération des cartes de présence. Comme attendu, la génération des cartes de votes varie linéairement avec $M$. Pour $M = 20$, le cas donnant les meilleurs résultats, la détection d'activité prend 0,127 s, dont seulement 0,029 s pour la génération des votes. Cela correspond à un traitement de 8 images par seconde, sans parallélisation des tâches. Ces résultats confirment la compatibilité de l'algorithme avec une détection des actions en temps réel.

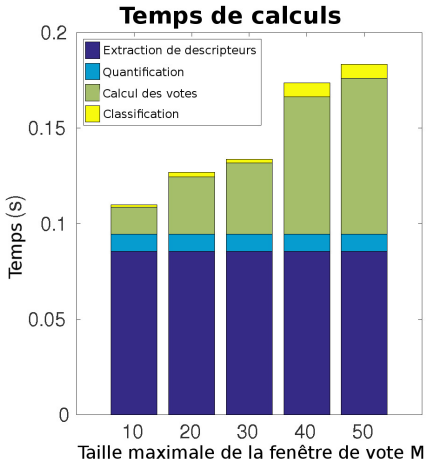



FIGURE 3.16 – Temps de calcul associés au DOHT en fusionnant les informations squelette aux trajectoires denses. Ces temps de calcul sont affichés en fonction de la demi-taille des fenêtres de vote $M$.

### 3.3.4 Comparaison avec les méthodes de l'état de l'art

Le Tableau 3.6 compare les résultats obtenus avec ceux précédemment présentés dans la littérature sur le jeu de donnée *TUM Kitchen Dataset* [126].

La première partie du tableau présente les résultats évalués avec la vérité terrain fournie par le papier original [126]. La seconde prend en compte les modifications proposées dans [127] et divisant la classe *"Se déplacer en portant en objet"* en deux classes : *"Marcher"* et *"Rester sur place"*.

Nos résultats dépassent ceux décrits dans l'état de l'art, montrant que l'ajout d'informations visuelles au sein de l'algorithme DOHT améliore les performances de détection. Ce constat est en phase avec les observations et conclusions de YAO, GALL, FANELLI et al. dans [127]. Dans cet article, les auteurs concluent que le cas idéal pour de la reconnaissance d'action semble être celui dans lequel plusieurs modalités d'informations sont utilisées.

| Methode | Taux de reconnaissance (%) |
|---|---|
| **Labels originaux** | |
| DOHT [63] | 81.5 |
| DOHT (Squelette à 27 articulations) [63] | 83.0 |
| ours (HOG+HOF, Une seule vue, M=50) | 82.5 |
| ours (Toutes vues, M=20) | 84.6 |
| ours (Toutes vues + Squelette, M=20) | **86.1** |
| **labels modifiés par [127]** | |
| Tous descripteurs + HF [127] | 81.5 |
| ours (Toutes vues + Squelette) | 81.6 |

Tableau 3.6 – Comparaison des résultats obtenus avec l'état de l'art. Les résultats présentés pour les méthodes de l'état de l'art sont extraits des papiers correspondants.

Pour comparer nos résultats à ceux de [127] avec la version modifiée des labels du DOHT, nous gardons les mêmes paramètres, sans optimisation. Les performances observées pour l'algorithme DOHT sont du même ordre de grandeur que ceux de [127].

## Conclusion

Dans ce chapitre, nous avons proposé et évalué une fusion d'information au sein d'un algorithme de détection d'actions par transformée de Hough. Plus spécifiquement, cette fusion a été réalisée sur l'algorithme de Transformée de Hough Fortement Optimisée (DOHT), proposé par Chan-Hon-Tong, Achard et Lucat dans [63].

Nous avons présenté trois paradigmes de fusion, à différentes étapes de cet algorithme : au niveau descripteur, au niveau de l'apprentissage des votes puis en aval de la génération des scores. Le deuxième présente l'avantage d'être robuste à une perte temporaire d'informations.

L'évaluation de ces paradigmes a été réalisée sur le jeu de données *TUM Kitchen* [126] qui présente l'avantage d'être multi-vues et propice à l'évaluation d'un algorithme de segmentation d'actions. Nous avons constaté que les performances de l'algorithme, en terme de taux de bonnes détections, étaient plus fortement améliorées lorsque les différentes sources sont fusionnées au plus bas niveau possible. Nous avons également montré et quantifié la robustesse de la fusion d'informations au sein du DOHT à une perte d'information. Ces résultats ont été comparés avec les performances publiées dans l'état de l'art.

Pour finir, nous avons montré que le DOHT permet une estimation prématurée des actions avec une confiance relativement élevée. Nous avons ensuite quantifié les temps de calculs nécessaires à chacune des étapes de l'algorithme et démontré la compatibilité de ce paradigme avec des applications ayant des contraintes temps réel.

Ces travaux ont fait l'objet de deux publications : La première dans *International Conference on Computer Vision and Applications (VISAPP)* [2], puis dans *Journal of Real-Time Image Processing* [1].

Chapitre 4

# Acquisition d'un jeu de données pour la détection d'activités

Dans l'ensemble des domaines de l'apprentissage automatisé, les données sont essentielles pour le développement et l'évaluation des méthodes d'apprentissage. C'est à partir de celles-ci que les programmes informatiques ajustent leurs paramètres afin de réaliser la tâche pour laquelle ils ont été conçus. Dans le cas de la vision par ordinateur en général et notamment de la reconnaissance d'activité humaine, les données peuvent être composés de différentes modalités (images RGB, carte de profondeur, coordonnées d'articulations (squelette), carte de disparité, ...) organisées on non en série temporelle (vidéo). Ces bases doivent contenir suffisamment d'information pour couvrir la variabilité des réalisations des classes d'activités. Cette quantité minimale d'information conditionnera la taille et les spécifications des bases de données. Aux prémices de l'analyse automatiques des mouvements humains, les bases de données contenaient des gestes très simples, avec une variabilité inter-classes élevée et une variabilité intra-classes faible, permettant une distinction relativement simple entre deux activités. Puis, avec la complexité et les performances croissantes des algorithmes, les jeux de données se sont complexifiés et les actions contenues se sont diversifiées.

Cependant, le niveau sémantique des classes analysées dans le domaine est resté relativement bas, se limitant aux actions, et peu de jeu de données sont adaptés à des méthodes de localisation temporelle d'activités. Cette thèse s'intéresse justement à la **détection** de classes de **haut niveau sémantique**, et il est nécessaire d'avoir des données adaptées pour évaluer les méthodes de détection de telles classes. C'est pourquoi nous proposons un nouveau jeu de données : la base *DAHLIA (DAily Home LIfe Activity)*.

Ce chapitre présente les jeux de données existants et met en exergue leurs limites vis-à-vis de l'objectif visé. Après avoir montré la nécessité d'un nouveau jeu de données à plus haut niveau sémantique, nous présentons les conditions d'acquisition ainsi que les caractéristiques du jeu de données DAHLIA. Nous présentons également les protocoles d'évaluation retenus et des premières évaluations d'algorithmes de la littérature afin de permettre une comparaison future des méthodes s'évaluant sur notre base de données.

## 4.1 Jeux de données existants

De multiples jeux de données ont vu le jour pour permettre l'analyse du comportement humain. Cette section présente les principales contributions dans ce domaine en commençant par les jeux de données mono-canaux (vidéo RGB) puis en présentant plus largement ceux contenant plusieurs modalités.

### 4.1.1 Les Jeux de données mono-canaux

**Acquis en conditions de laboratoire**

Une des premières bases de données utilisées en reconnaissance d'action est celle capturée par Laptev et Lindeberg, de l'Institut Royal de Technologie en suède, et publiée dans [13] : le jeu de données **KTH**. Cette base est destinée à la reconnaissance d'actions simples : *Marcher*,*Faire un Jogging*, *Courir*, *Faire des mouvements de boxe*, *Faire des signes de la main*, *Taper dans ses mains*. Notons que ces classes peuvent être de longues durée, mais constituées d'une succession quasi-périodique d'un ou deux gestes élémentaires. Ces 6 classes sont réalisées devant un fond uni par 25 personnes dans 4 scénarii différents : en intérieur, en extérieur, en extérieur avec variation d'échelle et en extérieur avec des vêtements différents. En dehors du scénario avec variation d'échelle, il n'existe que très peu de mouvement de caméra au sein des vidéos. Elle comporte un total de $25 \times 6 \times 4 = 600$ vidéos de résolution $160 \times 120$ pixels. La Figure 4.1a illustre quelques exemples de cette base.

Plus tard, en 2005, Blank, Gorelick, Shechtman et al. de l'institut des sciences Weizmann en Israel proposent dans [74], [128] le **Weizmann human action dataset** (Figure 4.1b) composé de 10 actions similaires à celles de *KTH* : *"Marcher"*, *"Courir"*, *"Sauter"*, *"Faire des pas chassés"*, *"Faire des signes d'une main"*, *"Faire des signes de deux mains"*, *"Sauter"*, *"Sauter sur place"*, *"Faire des jumping jacks"*. Les actions qu'il contient sont d'une durée moyenne de 4 s et ont été réalisées par 9 acteurs différents. Bien que la variation intra-classe soit légèrement plus forte que *KTH*, le point de vue fixe ainsi que l'arrière plan statique en font une base de données relativement simple.

Messing, Pal et Kautz de l'Université de Rochester publient en 2009 un jeu de données également composé d'actions simples telles que *"Marcher"*, *"Taper dans les mains"*, *"Courir"*, *etc.* : le jeu de données **Activities of Daily Living (ADL) benchmark [129]** (Figure 4.1c). Les vidéos qu'il contient, d'une durée de 10 à 60 secondes, ont été enregistrées devant un arrière-plan fixe. Là encore, peu de participants (5) réalisent les 10 actions possibles.

En plus d'actions classiques telles que *"S'asseoir"*, *"marcher"*, *etc.*, Hwang, Kim et Lee [130], capturent aussi dans le jeu de données **Korea University Gesture (KUG)** des actions "anormales". Il s'agit de différents types de chutes : vers l'avant, en arrière, à partir d'une chaise, *etc.*. A ces classes s'ajoutent également des gestes très codés et généraux signifiant par exemple *"oui"* ou *"non"* ou dessinant des chiffres du bout des doigts. Ce jeu de données a la particularité de ne pas être en libre accès.

Des bases contenant des gestes plus spécialisés ont été publiées respectivement en

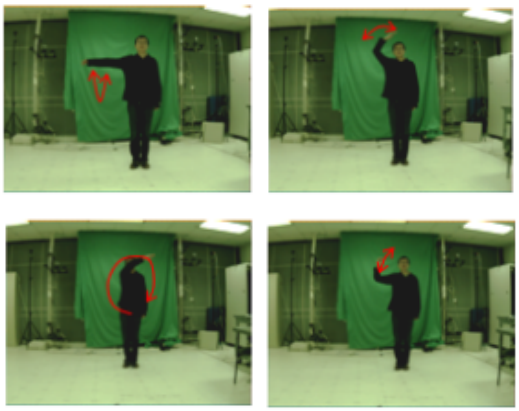
2009 et 2011 : le **Keck Gesture Dataset** [77] (Figure 4.1d) et le jeu de données **NATOPS** [131]. Ces deux jeux de données ont été acquis devant un fond uni et contiennent des gestes visuels servant pour communiquer dans les domaines de l'armée et de l'aviation. Une des particularités du NATOPS est que certains gestes ne peuvent être reconnus qu'avec une prise en compte à la fois du corps mais aussi plus précisément de la position des mains durant la réalisation du geste.

Ces jeux de données, ainsi que d'autres comme [132], [133], ont ouvert le champs de la reconnaissance de gestes et d'actions en fournissant à la communauté des vidéos d'actions très simples. La Figure 4.1 illustre quelques uns des jeux présentés jusqu'ici.

(a) KTH [13]

(b) Weizmann [74], [128]

(c) ADL [129]



(d) Keck Gestures [77]

FIGURE 4.1 – Jeux de données mono-cannaux en laboratoire

En 2006, WEINLAND, RONFARD et BOYER, de l'Institut National de Recherche en Informatique et en Automatique (INRIA) en France, publient le jeu de données **IXMAS**

**(Inria Xmas Motion Acquisition Sequences)IXMAS [11]** capturé à l'aide de 5 caméras placées autour du participant. Dans ce jeu de données, 14 actions sont définies parmi lesquelles *"Regarder sa montre"*, *"Se gratter la tête"*, *"Ramasser un objet"*, *"Croiser les bras"*, *"S'asseoir"*, *etc.* L'arrière-plan est également fixe et est éclairé de façon identique au travers des différents exemples. Les auteurs fournissent, en plus des vidéos, les silhouettes et représentations volumétriques qu'ils ont extraites à l'aide d'une soustraction de fond classique basée sur un modèle gaussien de chaque pixel. Ici encore, les vidéos sont d'une durée moyenne de moins de 5 s.

Rohrbach, Regneri, Andriluka et al., de l'institut Max Planck en Allemagne, publient en 2012 le **MPII Cooking Activity Dataset** [134] dans lequel les participants préparent différents plats (comme une salade ou des sandwichs) à l'aide de divers ingrédients. L'enregistrement a été réalisé sans interruption et comprend 65 actions telles que *"Couper en dés"*, *"Couper en rondelles"*, *"Verser dans un bol"*, *"Nettoyer"*, *etc.* Les auteurs fournissent, en plus des vidéos capturées, les descripteurs de type *trajectoires denses* extraits à partir du code fourni par [16] ainsi que des annotations de squelettes.

Ce jeu a par la suite été étendu dans [134] puis une deuxième version a été proposée en 2016 dans [135]. Cette dernière version reprend les exemples des deux premières, corrige certaines annotations, fournit des vidéos supplémentaires et définit plus explicitement les ensembles d'entraînement, de validation et de test.

Les jeux de données présentés précédemment ont été acquis dans des conditions de laboratoire, avec des gestes simples, un arrière-plan uni et fixe. Ils ont été utilisés pour le développement des premières méthodes de reconnaissance d'actions telles que [128], [136]–[138].

### Agrégation de données internet

En parallèle, d'autres jeux de données composés de vidéos provenant d'internet sont également apparus [33], [137], [139]–[142]. La diversité des conditions d'acquisitions et des contextes dans lesquelles ces vidéos ont été enregistrées leur procure une variabilité intra-classes et inter-classes respectivement bien plus élevée et bien plus faible.

En 2008, Laptev, Marszałek, Schmid et al. de l'Inria de Grenoble recueillent et proposent dans [33] un ensemble de vidéos annotées provenant de 32 films différents (12 pour l'ensemble d'entraînement et 20 pour l'ensemble de test) : le jeu de données **HOLLYWOOD**. L'annotation de l'espace d'entraînement a été réalisée manuellement pour 219 exemples et automatiquement, à partir des dialogues, pour les 233 autres. Les 211 vidéos de l'espace de test ont été manuellement annotées. En 2009, Marszalek, Laptev et Schmid proposent une extension, **HOLLYWOOD2** [139], se voulant devenir une référence pour l'évaluation des algorithmes de reconnaissance d'actions. Au total, Hollywood2 comporte 3669 vidéos extraites à partir de 69 films différents. Les annotations sont faites selon 12 actions dont *"Répondre au téléphone"*, *"Se battre"*, *"Manger"*, *"S'asseoir"*, *etc.* La Figure 4.2 illustre ce jeu de données.

A la même période, Rodriguez, Ahmed et Shah de l'Université de Floride Centrale

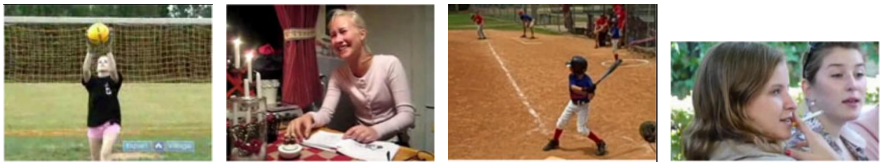


FIGURE 4.2 – Images du jeu de données HOLLYWOOD [33]

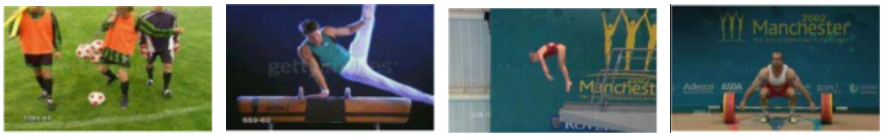


FIGURE 4.3 – Images du jeu de données UCF [137]

proposent le jeu de données **UCF sports action dataset [137]** (Figure 4.3) construit à partir de 150 vidéos extraites de retransmissions sportives, notamment sur les chaînes *BBC* et *ESPN*. Il contient 10 actions, à savoir *"Plonger"*, *"Jouer au Golf"*, *"Lever de poids"*, *"Frapper au pied"*, *"Courir"*, *"Monter à cheval"*, *"Faire du skakeboard"*, *"Marcher"*, *"Cheval d'arçons et Barres hautes"*.

En 2009, LIU, LUO et SHAH proposent le **UCF YouTube Action Dataset** [140], [141] puis son extension **UCF50** [142] composée de 50 actions dans 6676 vidéos téléchargées depuis *Youtube*. Ce jeu de données inclut des vidéos avec des mouvements de caméras variés, des arrières-plans complexes, des variations d'illumination de la scène ainsi que des points de vues très différents, ce qui rend la reconnaissance d'actions plus difficile.

En 2011, KUEHNE et SERRE de l'Institut de technologies de Karlsruhe en Allemagne, proposent un jeu de données composé de 51 catégories d'actions différentes : le **Human Motion DataBase (HMDB51)** [143] (Figure 4.4). Il contient environ 7000 exemples extraits d'internet. Les 51 actions sont regroupées dans 5 types différents à savoir

1. Actions faciales comme *"Sourire"*, *"Rire"*, *"Mâcher"*, *"Parler"* ;
2. Actions faciales avec un objet comme *"Fumer"*, *"Manger"*, *"Boire"* ;
3. Actions impliquant le corps comme *"taper dans les mains"*, *"grimper"*, *"monter des escaliers"*, *"plonger"*, *"tomber"*, *"courir"*, *"faire des pompes"*, *"s'asseoir"*, *"se lever"*, *etc.*
4. Actions impliquant le corps et un objet comme *"se brosser les cheveux"*, *"faire du golf"*, *"tirer au pistolet"*, *"monter à cheval"*, *"frapper une balle"*, *etc.*
5. Actions impliquant une interaction humaine comme *"prendre dans les bras"*, *"frapper quelqu'un"*, *"embrasser"*, *"pousser"*, *etc.*

Pour chaque classe, ils définissent un espace d'entraînement et un espace de test représentant respectivement 70% et 30% des exemples de cette action.

Les Figures 4.2 à 4.4 montrent la diversité d'arrière plan bien plus grande pour ces vidéos que pour celles de la Figure 4.1.

FIGURE 4.4 – Images du jeu de données HMDB [143]

Avec l'émergence des méthodes basées sur les réseaux de neurones profonds nécessitant un nombre d'exemples bien plus grand durant la phase d'apprentissage, des jeux de données à grande échelle ont été proposés [144], [145]. Le jeu de données Sports-1M [145], publié en 2014 par KARPATHY, TODERICI, SHETTY et al. contient un million de vidéos extraites de *Youtube*. Les auteurs identifient 487 classes autour du sport, chacune d'entre-elle associée à entre 1000 et 3000 vidéos. Le jeu de données **ActivityNet** [144] contient plus de 200 classes différentes sur près de 850h de vidéos. Depuis la publication de l'article présentant **ActivityNet**, un challenge a été organisé chaque année pour comparer les méthodes de l'état de l'art sur ce jeu de données. Une particularité de ces articles est qu'ils définissent une taxonomie sur les classes proposées. Cette taxonomie est inspirée de celle utilisée par le gouvernement américain pour classer les activités : *American Time Use Survey (ATUS)*. Dans cette version du jeu de données, les auteurs ont retenu sept catégories principales : *Soins personnels*, *Travail*, *Soutien et aide*, *Nourriture et Boisson*, *Entretien de la maison*, *Activités sociales et Loisirs* et *Sports*.

Les jeux de données introduits jusqu'ici ne sont composés que de vidéos courtes. Par leur nature, ils se limitent à des données en 2 dimensions et ignorent alors une part importante de l'information contenue dans la scène réelle, comme les positions relatives en 3 dimensions des membres de la personne. D'autres bases de données, comme [146] considèrent des actions également simples, mais capturées à l'aide de capteurs actifs et de marqueurs posés sur les participants (cf. Figure 4.5). Cependant, ce type de dispositifs n'est pas envisageable dans des applications orientées smart-home car trop invasif et non réaliste dans un contexte quotidien.

Nous présentons dans la suite des jeux de données considérant d'autres canaux que les simples vidéos RVB.

### 4.1.2 Les jeux de données multi-canaux

Pour tirer profit des informations contenues dans les données 3D et avec l'émergence des capteurs actifs tels que la Kinect, un grand nombre de jeux de données ont vu le jour [147]. Ce type de capteurs fournit les coordonnées 3D des articulations des personnes et/ou les cartes de profondeur acquises par la caméra. Cette section décrit les jeux de données les plus notables tirant profit de cette technologie.

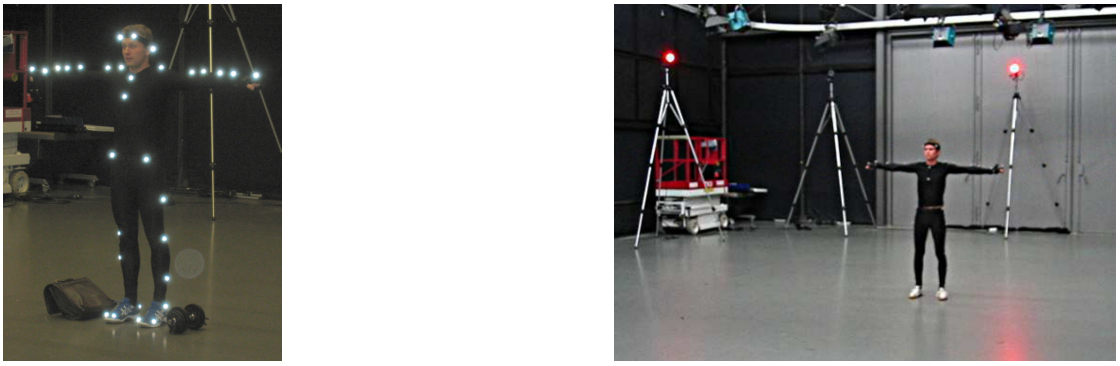


FIGURE 4.5 – Illustration du système d'acquisition *motion capture* utilisé par [146]. Images tirées de cet article.

En 2009, TENORTH, BANDOUCH et BEETZ de l'Université Technique de Munich, proposent un jeu de donées comprenant les vidéos de 4 caméras placées autour d'une cuisine ainsi que les coordonnées squelettes des 4 acteurs annotées manuellement. Il s'agit du **TUM Kitchen Dataset [126]**. Chacune des vidéos inclut une succession d'actions parmi *"Ouvrir/Fermer un tiroir"*, *"Ouvrir/Fermer un placard"*, *"Prendre un objet"*, *"Se dépacer"*, *"Poser un objet"* et est annotée temporellement. Il a été conçu pour des applications de détection et extraction de squelette puis a également été utilisé pour la détection d'actions [2], [119], [124].

Un des premiers jeux de données capturés à l'aide d'une Kinecta été enregitré en 2010 par LI, ZHANG et LIU de l'Université de Wollongong : le **MSR Action3D Dataset [36]**. Il contient 20 actions telles que *"Taper dans les mains"*, *"Faire un signe d'une main"*, *"Faire un signe des deux mains"*, *etc.* Les auteurs ont traité les cartes de profondeur après acquisition afin de retirer l'arrière plan : seule la carte de profondeur des personnes est disponible, sur un fond sans information. Ce jeu est comparable à *KTH* et *Weizmann* dans la complexité des actions effectuées et s'en différencie par le type de données fournies. En effet, seules les cartes de profondeur sont téléchargeables.

Des actions un peu plus complexes sont proposées par WANG, LIU, WU et al., de l'Université Northwestern, qui publient en 2012 un jeu de données composé de 16 actions quotidiennes comme *Lire un livre*, *Boire*, *Manger*, *S'asseoir*, *etc.* : le **MSR DailyActivity3D Dataset [39]**. Ces courtes actions sont réalisées devant un arrière plan fixe et les auteurs fournissent les vidéos, cartes de profondeur ainsi que les coordonnées des articulations, extraites par la Kinect.

En 2012, CHENG, QIN, YE et al. de l'Université de l'Accadémie des sciences chinoise publient le jeu de données **ACT4$^2$** [148] qui comprend le même type d'actions que [39]. Dans l'optique de développer des méthodes pour des applications de maisons intelligentes, ils considèrent en plus les actions *"Chuter"* et *"Trébucher"*. *"Chuter"* concerne les chutes pour des raisons médicales et *"Trébucher"* concerne les chutes ayant pour cause la présence d'un objet. Ce jeu contient les flux en provenance de 4 capteurs Kinectdisposés uniformément autour de la scène. 24 participants réalisent les 14 actions plusieurs fois

pour un total de 6844 exemples.

La même année, Zhang, Liu, Metsis et al., de l'Université de Texas, publient un jeu de données en vue de détecter les chutes à l'intérieur d'un appartement **Falling detection dataset** [149]. Il a été enregistré à l'aide de deux Kinectplacées dans deux coins d'une pièce afin de couvrir l'ensemble de la scène. Seules les cartes de profondeur sont fournies. Il comporte la classe *"Chutes"*, mais aussi des actions pouvant y ressembler telles que *"Ramasser une pièce au sol"*, *"S'allonger sur le lit"*, *"Ouvrir un tiroir au sol"* ou *"Faire ses lacets"*. Au total, 61 exemples sont des actions autres que des chutes et 26 sont de vraies chutes. Notons que pour chacun des 6 acteurs, il n'existe qu'un exemple dans lequel se succèdent les actions.

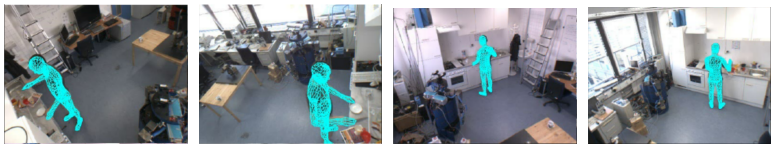
En 2013, Ni, Wang et Moulin [150] du centre des sciences digitales avancées de Singapour proposent un jeu de données ayant un plus haut niveau sémantique : le **RGBD-HuDaAct**. Parmi les 12 actions proposées se trouvent par exemple les classes *"Téléphoner"*, *"Boire de l'eau"*, *"Mettre une veste"*, *...* ). Ces actions sont réalisées par 30 personnes différentes devant un arrière-plan fixe et impliquent pour une partie d'entre-elles l'utilisation d'un objet. Cela permet l'exploitation du contexte dans lequel une action est réalisée.

Pour diversifier ces contextes, Sung, Ponce, Selman et al. de l'Université de Cornell proposent en 2011 le **CAD60** [151], un jeu de données capturé dans 5 lieux différents (une cuisine, une salle de bain, un salon, un bureau et une chambre). En plus des 12 actions *"Se rincer la bouche"*, *"Se brosser les dents"*, *"Mettre des lentilles de contact"*, *"Parler au téléphone"*, *"Boire de l'eau"*, *"Ouvrir une boite de médicaments"*, *"Couper un aliment"*, *"Mélanger des aliments"*, *"Parler sur le canapé"*, *"Se relaxer"*, *"Ecrire"*, *"Travailler à l'ordinateur"* réalisées par les quatre participants, d'autres actions aléatoires ont été ajoutées afin d'augmenter la difficulté de l'espace de test.

Plus tard, en 2013, Koppula, Gupta et Saxena de la même université plublient le **CAD120** [152], une extension du *CAD-60*. Dans cette nouvelle version, l'annotation des vidéos a été faite à deux niveaux : chaque vidéo comporte une action telle que *"Prendre un médicament"*, *"Empiler des objets"*, *"Faire réchauffer de la nourriture"*,*etc.* et est annotée temporellement selon le geste effectué. Les gestes considérés sont *"Atteindre"*, *"Boire"*, *"Ouvrir"*, *"Fermer"*, *"Déplacer"*, *"Placer"*, *"Manger"*, *"Boire"*, *"Verser"*, *"Nettoyer"* ainsi que la classe *"Nul"*. En plus de ces annotations, les auteurs fournissent une ou plusieurs caractéristiques pour chaque objet visible. Un objet peut ainsi être *"Atteignable"*, *"Déplaçable"*, *"Buvable"*, *"Ouvrable"*, *etc.*

Wei, Zheng, Zhao et al., de l'Université Xi'an Jiatong et de l'Université de Californie, publient en 2013 le **Concurrent Action Dataset** [153]. Il présente la particularité de présenter une simultanéïté de plusieurs actions. Différentes actions sont donc présentes dans chaque vidéo ce qui rend ce jeu de données adapté à des applications de type détection temporelle et/ou spatiale d'actions.

Amiri, Pourazad, Nasiopoulos et al., de l'Université de Colombie Britanique au Canada proposent le jeu de données **DMLSmartActions** [154] adapté à la détection d'actions. Ce jeu comporte 12 actions réalisées par 18 personnes différentes et a été



(a) TUM [126]

(b) MSR [36]

(c) CAD [151], [152]

FIGURE 4.6 – Jeux de données multi-canaux

enregistré à l'aide de 2 caméras HD ainsi qu'une Kinect placées autour de la scène. Les actions sont courtes (*"Ramasser quelque chose"* ou *"S'asseoir"* par exemple) et sont réalisées sans interruption.

WEI, ZHAO, ZHENG et al. des Universités Jiao-tong de Shanghai et de Californie à Los Angeles, publient en 2013 le **Multiview 3D Event Dataset** [155]. Il contient les flux de données en provenance de trois Kinect synchronisées, capturés sans interruption. Le jeu a ensuite été segmenté et annoté manuellement et contient un total de 3815 vidéos. Les actions incluses sont courtes telles que *"Boire dans une tasse"*, *"Téléphoner"*, *"Lire un livre"*, *"Utiliser une souris d'ordinateur"*, *"Taper au clavier"*, *"Verser de l'eau"*, *etc.* Dans cet article les auteurs ont donc segmenté les données, acquises de manière continue, et ne présentent des résultats de segmentation uniquement sur 10 vidéos qu'ils conservent non segmentées.

GEMEREN, TAN, POPPE et al. de l'Université d'Utrecht ont mis en ligne en 2014 le jeu de données **ShakeFive** [156] dédié à la reconnaissance d'interactions contenant deux interactions : *"Se serrer la main"*, *"Se taper dans la main"*. Chaque image des 100 vidéos est annotée selon si une personne est présente et est associé à une des 5 classes : *"Immobile"*, *"Rapprochement"*, *"Serrer la main"*, *"Se taper dans la main"*, *"Se séparer"*. VAN GEMEREN, POPPE et VELTKAMP publient en 2016 une nouvelle version : *ShakeFive2* [157] contenant 8 interactions possibles. La première version a été capturée avec une Kinect et la seconde avec une Kinectv2. Les auteurs mettent à disposition les vidéos ainsi que les données squelettes en provenance des capteurs.

Wolf, Lombardi, Mille et al. du Centre National de la Recherche Scientifique (CNRS) publient quant à eux, en 2014, un jeu de données à partir d'une caméra mobile : le **LIRIS human activities dataset** [158]. La caméra est fixée à un robot téléguidé. Leur jeu de données présente une variabilité interclasse faible et chacun des exemples peut être associé à un des trois types d'interactions suivants : *humain-humain*, *humain-objet* et *humain-humain-objet*. A titre d'exemple, il contient des actions visuellement très similaires telles que *Entrer dans une salle*, *Dévérouiller la porte puis entrer* et *"Tenter d'entrer sans y parvenir"*. Pour finir, de la même façon que dans [153], plusieurs actions peuvent avoir lieu à un même instant temporel permettant d'évaluer des algorithmes de détection spatiale d'actions.

Xu, Liu, Nie et al. de l'Université de Tianjin publient, en 2015, le **M$^2$I dataset** [159] composé d'actions impliquant également des interactions humain-humain et humain-objet. Il reste cependant plus simple que [158] car les acquisitions ont été capturées à l'aide de deux caméras immobiles, avec un arrière plan fixe. Ce jeu de données contient 1760 exemples d'actions telles que *"Service de tennis"*, *"Téléphoner"*, *"Lancer un ballon de basket"*, *"Jouer au football"*, *"Se serrer la main"*, *"Se battre"*, ..., réalisées par 22 personnes (20 groupes de deux pour les interactions entre humains).

En 2015, Hu, Zheng, Lai et al. de l'Université Sun Yat-sen publient un jeu d'actions impliquant une interaction avec un objet : le **SYSU 3D Human-Object Interaction Dataset** [45]. Les objets présents dans les vidéos peuvent être *un téléphone*, *une chaise*, *un sac*, *un portefeuille*, *un balai espagnol* et/ou *un balai*. Dans ce jeu de données, 40 participants réalisent les 12 actions *"Boire"*, *"Verser"*, *"Porter un sac à dos"*, *"Jouer sur son téléphone"*, *"Téléphoner"*, *"Remplir un sac à dos"*, *"s'asseoir sur une chaise"*, *"Déplacer une chaise"*, *"Sortir son portefeuille"*, *"Passer la serpière"* et *"Passer le balai"* pour un total de 480 vidéos.

En 2015, après la sortie de la deuxième version de la Kinect capturant des vidéos en HD et ayant une bien meilleure précision dans l'extraction de la carte de profondeur et dans la détection des articulations, Wu, Zhang, Sener et al. des Universités de Cornell et Stanford conçoivent, le **Watch-n-Patch Dataset** [99]. Il a pour but de permettre une détection et reconnaissance d'actions basées sur leurs co-occurences et leurs agencements temporels. Ils proposent donc un jeu de données dans lequel les exemples sont composés d'une succession d'actions (entre 2 et 7 par vidéo) telles que *"Prendre un objet"*, *"Verser"*, *"Boire"*, *"Chauffer au micro-ondes"*, ... Il existe plusieurs combinaisons possibles et certaines actions apparaissent régulièrement ensembles telles que *"Prendre un aliment dans le réfrigirateur"* et *"Ranger un aliment dans le réfrigirateur"*. Dans 222 des 458 vidéos, une action est volontairement omise pour évaluer la capacité des algorithmes à détecter un tel oubli. Il a été enregistré dans 13 lieux différents (8 bureaux et 5 cuisines) ce qui engendre une grande variabilité intra-classe et une certaine variété dans les localisations spatiales des actions. Les auteurs fournissent les flux vidéos, cartes de profondeur ainsi que les coordonnées squelettes extraites par une Kinectv2.

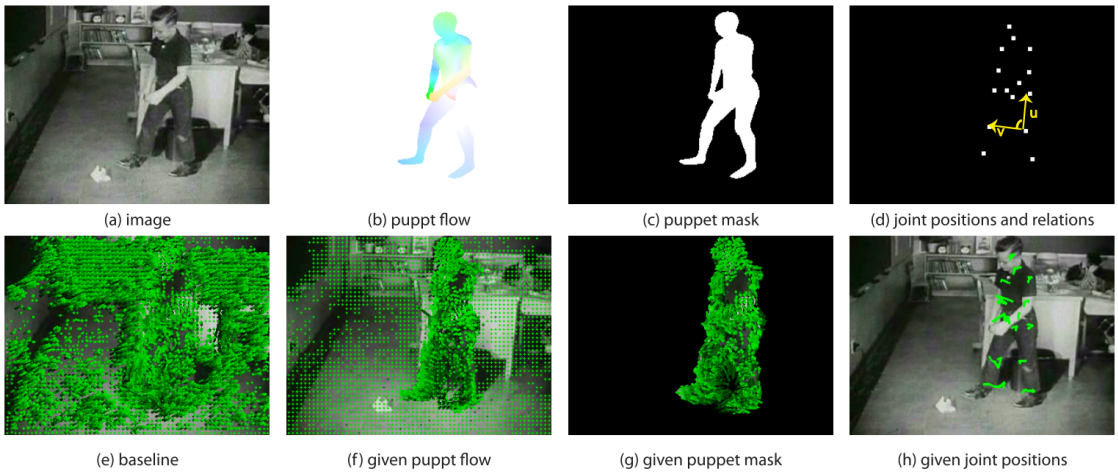



FIGURE 4.7 – Illustration de l'enrichissement de HMBD [143] par [160]

**Des jeux de données à grande échelle** Pour un volume de données plus important, [160] étendent en 2013 le jeu de données *HMBD* [143] dans **JHMBD** [160]. Les auteurs ont extrait 928 exemples et ont ajouté les annotations de squelettes à partir d'un modèle 2D articulé. Il en résulte un jeu de données contenant 21 actions (parmi les 51 de HMBD) pour lesquelles sont fournis la pose de la personne, sa silhouette (masque binaire), les flux optiques sur son corps ainsi qu'un angle de vue grossier.

En 2016, SHAHROUDY, LIU, NG et al. de l'Université technologique de Singapoure proposent dans [161] un jeu de données à grande échelle : le **NTU RGB+D Dataset**. Ce jeu de données consite en 56 880 exemples d'actions courtes (5 s) réalisées par 40 participants et capturé par une Kinectv2 dont la position varie selon les vidéos, engendrant 80 points de vues différents. Parmi les 60 classes présentes, 40 sont des actions du quotidien telles que *"Boire"*, *"Manger"*, *"Lire"*, *etc.*, 9 sont liées au domaine de la santé (*Renifler, Tomber, etc.*) et 11 impliquent une interaction (*"Frapper"*, *"Prendre dans les bras"*, *etc.*) Les auteurs mettent en ligne la totalité des flux capturés : vidéos RGB et infrarouge, cartes de profondeur et données squelettes.

### 4.1.3 Discussion et comparaison

La section précédente a présenté une liste (non-exhaustive) des jeux de données les plus utilisés en reconnaissance et détection d'actions. Afin de comparer efficacement ces différents jeux de données, nous évaluons plusieurs caractéristiques telles les modes d'acquisitions, la complexité de leur arrière-plan et des mouvements impliqués dans les actions représentées. Inspiré par la catégorisation proposée par ZHANG, LI, OGUNBONA et al. dans [147], ces critères sont définis comme suit :

**Mode d'acquisitions :**

**Mode 1** Le jeu de données a été acquis *action par action* et chacun des exemples n'en contient qu'une seule.

**Mode 2** Le jeu de données a été acquis *action par action*, chacun des exemples n'en contient qu'une seule et une *annotation temporelle des gestes* est fournie.

**Mode 3** Le jeu de données a été acquis comme une *succession d'actions* dans un ordre *prédéterminé et fixé pour toutes les vidéos*. Les annotation temporelles sont fournies.

**Mode 4** Le jeu de données a été acquis comme une *succession d'actions* dans un ordre *aléatoire et variant d'une vidéo à l'autre*. Les annotations temporelles sont fournies.

**Niveau de complexité pour l'arrière plan :**

**Faible :** L'arrière plan est *rangé et peu encombré*. Il est stable il n'y a *pas d'occultation* des sujets pendant la vidéo.

**Moyenne :** L'arrière plan est fixe et *encombré*. Il peut y avoir certaines *occultations partielle* des sujets.

**Forte :** L'arrière plan *varie selon les sujets et est encombré*. Des occultations peuvent perturber la visualisation de l'action.

Nous comparons également la complexité cinématique ainsi que le niveau sémantique (gestes, actions, ou activités). Le tableau Tableau 4.1 regroupe les différents jeux de données présentés et en résume les principales caratéristiques.

On constate que la majorité des jeux de données proposés dans la littérature sont adaptés à de la classification d'actions pré-segmentés et non à de la localisation temporelle. Or, les travaux qu'intéressent cette thèse sont orientés vers des applications qui nécessitent une détection en ligne des activités effectuées par une personne. Il serait possible, en contaténant plusieurs vidéos, de transformer un jeu de données de classification en un jeu de données compatible avec de la détection. Cependant, cela génèrerait des artefacts (notamment aux transitions) qui pourraient être exploités par les algorithmes de segmentation pour retrouver les coupures entre actions. De plus, une telle transformation des données n'est pas réaliste et les méthodes évaluées pourraient ne pas se généraliser à des vidéos capturées en conditions réelles.

Par ailleurs, les jeux de données issues de la littérature adaptés à de la segmentation (acquis en *mode 4*) sont composés de classes à faible niveau sémantique (Actions). Or, nos travaux ont pour objectif la détection et reconnaissance temporelle d'activités à haut niveau sémantique telle que *"Prendre un déjeuner"*, *"Se reposer"*, *etc.*, présentant une forte variabilité intra-classes.

Face à l'absence, à notre connaissance, de jeux de données respectant notre cahier des charges, nous avons acquis et mis à disposition de la Communauté un nouveau jeu de données correspondant aux besoins évoqués. Nous présentons dans la suite de ce chapitre le *DAily Home Life Activity (DAHLIA) Dataset* [163]. Après une définition des spécifications, nous explicitons les caractéristiques des données acquises et fournissons des premières évaluations sur les vidéos acquises.

| Jeu de données | Année | Nbr Sujets | Nbr d'actions | Durée moy. d'une action | Modalités | Nb Vues | Mode d'acquisition | Complexité de l'Arrière plan | Complexité cinématique | Niveau sémantique |
|---|---|---|---|---|---|---|---|---|---|---|
| KTH [13] | 2003 | 25 | 6 | 4 s | C | 1 | Mode 1 | Faible | Faible | Gestes |
| Weizmann [74] | 2005 | 9 | 10 | 4 s | C [1] | 1 | Mode 1 | Faible | Faible | Actions |
| IXMAS [11] | 2006 | 11 | 13 | 3 s | C | 1 | Mode 1 | Faible | Faible | Actions |
| ADL [129] | 2009 | 5 | 10 | 30 s | C | 1 | Mode 1 | Moyenne | Moyenne | Actions |
| HOLLYWOOD2 [139] | 2009 | - | 12 | 19 s | C | 1 | Mode 1 | Forte | Forte | Actions |
| TUM [126] | 2009 | 4 | 9 | 2 s | C,S | 4 | Mode 4 | Faible | Moyenne | Gestes |
| MSR-Action 3D [36] | 2010 | 10 | 20 | 3 s | D,S | 1 | Mode 1 | Faible | Faible | Actions |
| RGBD-HuDaAct [150] | 2011 | 30 | 12 | - | C,D | 1 | Mode 1 | Moyenne | Moyenne | Actions |
| CAD-60 [151] | 2011 | 4 | 12 | 45 s | C,D,S | - | Mode 1 | Moyenne | Moyenne | Actions |
| ACT4 Dataset [148] | 2012 | 24 | 4 | 15 s | C,D,S | 4 | Mode 1 | Faible | Faible | Actions |
| Falling Detection [149] | 2012 | 6 | 8 | - | D | 2 | Mode 4 | Forte | Faible | Actions |
| MPII Cook. Act. [162] | 2012 | 12 | 65 | 6 s | C,S | 1 | Mode 4 | Moyenne | Moyenne | Actions |
| MSRDaily-Activity3D [39] | 2012 | 10 | 16 | - | C,D,S | 1 | Mode 1 | Moyenne | Forte | Actions |
| CAD-120 [152] | 2013 | 4 | 10 | 17 s | C,D,S | 1 | Mode 2 | Forte | Forte | Actions |
| Concurrent Action [153] | 2013 | - | 12 | 10 s | S | 1 | Mode 4 | No Bkgd | Moyenne | Actions |
| UCF50 [140], [141] | 2013 | - | 50 | - | C | 1 | Mode 1 | Forte | Forte | Actions |
| DMLSmartActions [154] | 2013 | 18 | 12 | 5 s | C,D | 3 | Mode 4 | Moyenne | Moyenne | Actions |
| Multiview 3D Event [155] | 2013 | 8 | 8 | 10 s | C,D,S | 3 | Mode 3 | Forte | Moyenne | Actions |
| ShakeFive [156] | 2014 | 37 | 5 | 9 s | C,S | 1 | Mode 1 | Faible | Faible | (Inter) Actions |
| LIRIS [158] | 2014 | 21 | 10 | 5 s | C,S,G | 1 | Mode 1 | Forte | Forte | Actions |
| $M^2I$ [159] | 2015 | 22 | 22 | 5 s | C,D,S,M | 2 | Mode 1 | Moyenne | Moyenne | Actions |
| SYSU [45] | 2015 | 40 | 12 | 5 s | C,D,S | 1 | Mode 1 | Moyenne | Moyenne | Actions |
| Watch-n-Patch [99] | 2015 | 7 | 21 | 6 s | C,D,S | 1 | Mode 4 | Forte | Moyenne | Actions |
| ActivityNet [144] | 2015 | - | 201 | - | C | - | Mode 1 | Forte | Forte | Actions |
| ShakeFive2 [157] | 2016 | - | 8 | 10 s | C,S | 1 | Mode 1 | Faible | Faible | (Inter) Actions |
| DAHLIA (proposé) | 2016 | **45** | 7 | **6 min** | C,D,S | **3** | **Mode 4** | Forte | **Très forte** | **Activities** |

Tableau 4.1 – Résumé des jeux de données

[1] et silhouette

## 4.2 Cahier des charges et méthode d'acquisition

Face à l'absence de jeux de données satisfaisants pour les applications que nous visons, nous avons fait l'acquisition d'un nouveau jeu de données : la base DAily Home LIfe Activity *DAHLIA*. Il a été conçu pour évaluer des méthodes de détection des activités de la vie quotidienne et a pour but de permettre, par exemple, le développement d'applications de type *Maison intelligente*.

Nous avons défini 7 classes d'activités de la vie courante contenant des mouvements variés, à savoir

1. Faire la cuisine,
2. Prendre son déjeuner,
3. Faire la vaisselle,
4. Dresser la table,
5. Débarasser la table,
6. Faire le ménage et
7. Travailler

Ces 7 classes ont un niveau sémantique très haut et contiennent une variabilité intra-classe forte. Par exemple, si l'activité *"Prendre son déjeuner"* contient des gestes similaires d'une personne à l'autre (Boire, Découper un aliment, Mener la main à la bouche, *etc.*), ces gestes peuvent être réalisés dans des ordres différents et pendant une durée variable selon les personnes. De même, il y au sein de l'activité *"Faire la vaisselle"* de fortes variations dans l'exécution des actions qu'elle implique en fonction des habitudes propres à chaque participant.

Pour être adapté à la segmentation temporelle d'activités humaines, notre jeu de donnée doit contenir une succession d'activités, sans interruption et dans un ordre varié. Nous voulions également, pour être au plus proche de conditions réelles, que les activités se déroulent dans un ordre naturel et qu'elles se succèdent de façon fluide. Pour cela nous avons défini huit scénarios différents, impliquant chacune des activités dans des ordres différents mais toujours logiques (l'activité *"Débarasser la table"* ne peut, par exemple, pas se trouver juste avant *"Prendre son déjeuner"*). Chaque participant tirait donc au sort un scénario avant de commencer l'acquisition.

Par ailleurs, afin de procurer au jeu de données la plus grande variabilité intra-classe possible, il a été décidé de donner une liberté très forte sur la façon dont chacun des participants effectuait l'activité. Pour cela, les consignes de chacune d'entre-elles étaient très simple :

**Mettre la table :** Avant de manger, dressez la table en récupérant les différents ustensiles dans les placards (ordre de mise en place libre) de manière à prendre votre repas sur la chaise haute, sans oublier d'éléments.

**Préparer un repas :** Sortez les aliments des placards et du frigo pour préparer une salade avec les aliments à disposition. Les aliments seront découpés sur une planche à découper et mélangés dans un saladier. Il n'y a aucune obligation à

utiliser tous les aliments. Vous aurez aussi à préparer une vinaigrette que vous pourrez ajouter à votre salade.

**Prendre son déjeuner :** Savourez votre repas sur la chaise haute, en vous désaltérant librement. Vous n'êtes pas autorisés à vous lever durant cette phase.

**Débarasser :** Débarassez entièrement la table et déposez la vaisselle utilisée dans l'évier. Une poubelle est à votre disposition pour les déchêts. Ranger les éléments comme le sel et le poivre dans les placards appropriés.

**Faire la vaisselle :** Après avoir installé la bassine d'eau [2] sur l'îlot, nettoyez la vaisselle. Vous avez un égouttoir à disposition. Essuyez la vaisselle et ranger-la.

**Faire le ménage :** Nettoyez, à l'aide d'une lingette, les plaques et les plans de travail. Passez le balai. Ces actions peuvent être faites dans un ordre au choix. Au besoin, videz et changez le sac poubelle. Remettez les éléments utilisés à leur place.

**Travailler :** A l'aide des livres fournis, remplissez le questionnaire. *Afin de rendre cette activité la plus réaliste possible, un questionnaire imposant une recherche d'information dans différents supports était soumis aux participants. Il nécessitait une lecture de certains paragraphe et/ou la recherche d'information sur une image.*

Il a donc été demandé aux participants de réaliser les activités naturellement. Ces consignes étaient lues avant le début de l'acquisition, puis expliquées à l'oral. Pendant l'acquisition, un panneau rappelant l'ordre à suivre était affiché. Aucune contrainte de temps n'a été imposée aux participants, cela a permis de conserver les variations de durée d'exécution des différentes activités. Finalement, pour ajouter de la variabilité dans le jeu de données, les lieux de réalisation de certaines activités varient également.

## 4.3 Acquisition des données

### 4.3.1 La plateforme MobileMii

Le jeu de données que nous proposons a été acquis dans la plateforme MobileMii. Il s'agit d'une plateforme expérimentale crée par le CEA LIST en partenariat avec l'Institut Mines-télécom en vue de développer et d'évaluer des services d'intelligence ambiante. Pour les applications de type habitation intelligente (*smart-home*), la plateforme possède un véritable appartement équipé de capteurs multiples (notamment audio et vidéos). Equipé comme le serait un logement habité, cet appartement permet d'évaluer des algorithmes et des technologies dans un environnement très réaliste. Il s'agit également d'un espace de démonstration pour les produits développés par les entités partenaires.

### 4.3.2 Description des données acquises

Le jeu de données *DAHLIA* a été capturé à l'aide de trois Kinectv2 disposées autour de la cuisine. Elles ont été placées de façon à minimiser les configurations dans lesquelles une occultation serait présente sur les trois vues simultanément. La Figure 4.8 illustre

2. La plateforme d'acquisition n'est pas équipé d'eau courante.

approximativement la position des différentes caméras et des principaux éléments de la cuisine.

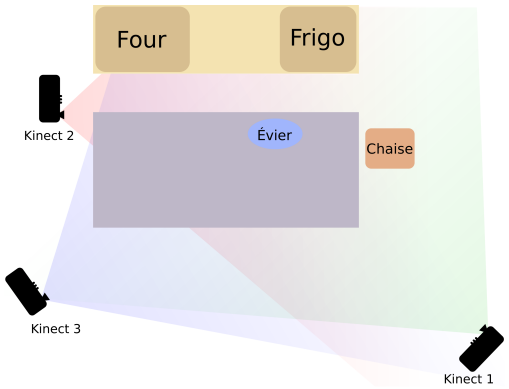



FIGURE 4.8 – Localisation des 3 caméras pour le jeu de donnée DAHLIA.

Pour permettre une compatibilité avec un large spectre d'algorithmes, nous avons capturé et mettons à dispositions tous les flux extraits à partir des 3 kinects, à savoir :

**les vidéos Haute définition** ($1920 \times 1080$ pixels), compressées en H.264 à 2Mbits/s,

**les cartes de profondeur** d'une résolution de $512 \times 424$ pixels. Pour chaque pixel, la distance au capteur (profondeur) est encodée sur 16 bits. Ces cartes ont été traitées à l'aide d'un filtre passe bas composé de 3 filtres médians (sur les deux dimensions spatiales et la dimension temporelle),

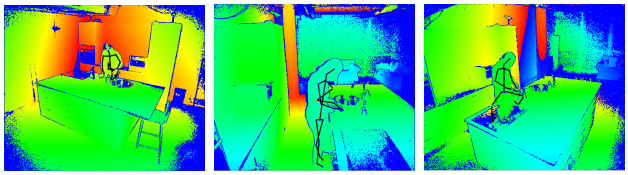
**les squelettes** extraits par l'algorithme fourni avec la Kinectv2. Ces données contiennent les coordonnées estimées de chaque articulation ainsi qu'un indice de confiance associé. Celui-ci peut prendre trois valeurs selon si l'articulation est considérée par la Kinectv2 comme visible (2), invisible mais estimée (1) ou non-visible (0). Les coordonnées des squelettes sont exprimées en mètres dans un repère 3D attaché à la caméra. Les articulations correspondants aux membres inférieurs sont fréquemment associées à un niveau de confiance faible, car souvent occultées par l'îlot central.

**les masques binaires** (silhouettes) de la personne fournis par le capteur et servant à l'estimation des squelettes.

La Figure 4.9 présente un des exemples de la base à un instant temporel.

Les acquisitions ont été realisées avec la participation de 44 personnes (29 hommes et 15 femmes) agées de 23 ans à 61 ans. Les séquences durent en moyenne 39 min, la plus courte durant 24 minutes et la plus longue 64 minutes. Cette haute variabilité est un autre atout de cette base de données, car la durée d'une même activité varie fortement. Par ailleurs, certaines activités sont par nature plus longues que d'autres, par exemple *Prendre son déjeuner* représente en moyenne 23% des vidéos, bien plus que *Débarasser la table* qui n'en représente que 5%. La Figure 4.10 illustre la répartition moyenne des classes au sein du jeu de données.

(a) Images extraites de la vidéo HD sur chacune des vues



(b) Cartes de profondeur et squellettes sur chacune des vues

(c) Masques binaires (silhouettes) sur chacune des vues

FIGURE 4.9 – Extrait de la base DAHLIA

La base de données *DAily Home Life Activity* est un des jeux existants avec le plus grand nombre de sujets. Cela lui octroie une variation intra classe relativement forte. Les 7 activités définies sont d'un **haut niveau sémantique** à l'inverse de ces prédécesseurs. Bien qu'il soit un des jeux de données avec le moins de classes différentes, remarquons que ces activités peuvent être divisées en un grand nombre d'actions ce qui engendre une très **haute complexité cinématique**. Les activités durent en moyenne **6 minutes**, une durée bien plus importante que les autres jeux de données. Pour finir, les activités sont effectuées dans des ordres variés et se succèdent de façon **ininterrompue** (Mode 4).

Le Tableau 4.1 permet une comparaison du jeu de donnée DAHLIA avec les principaux jeux précédemment publiés et présentées au début de ce chapitre.

Le jeu de données *DAily Home Life Activity* est en ligne sur le site http ://www-mobilemii.cea.fr et est téléchargeable gratuitement en plusieurs fichiers d'une taille totale de 280Go.

## 4.4 Protocoles d'évaluation et métriques retenues

### 4.4.1 Protocoles d'évaluations

Chaque image des 51 vidéos est associée à une activité, ou à la classe neutre lorsqu'aucune activité n'est réalisée. Afin de permettre la comparaison de différentes méthodes

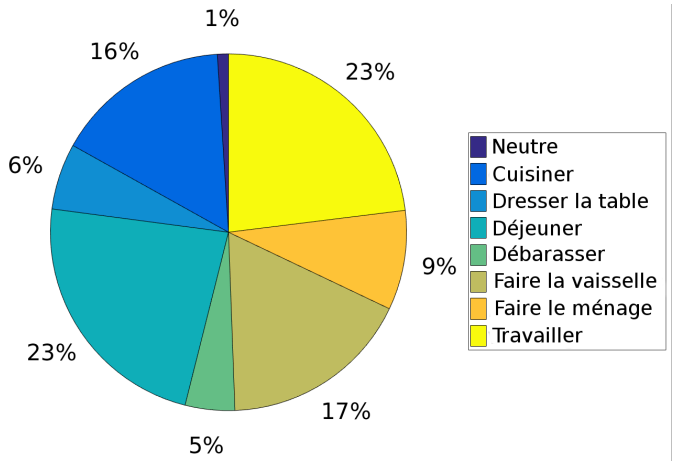



FIGURE 4.10 – Proportion des classes.

sur cette base de données, nous définissons deux protocoles d'évaluation :

**Evaluation cross-subject :** Deux groupes de personnes ($A$ et $B$) ont été définis. Chacun d'eux représente environ la moitié des sujets. L'évaluation se fait alors dans les deux configurations : Entraînement sur $A$, test sur $B$ et Entraînement sur $B$, test sur $A$. Le score final est alors calculé comme la moyenne des scores sur ces deux configurations. Notons que dans cette configuration, toutes les vues peuvent être utilisées.

**Évaluation cross-subjet et cross-views :** Pour compenser le manque de variations d'environnement (une seule cuisine), nous définissons un protocole mélangeant les différentes vues. Dans celui-ci, l'apprentissage se fait sur un des groupes ($A$ ou $B$), sur une seule vue. On évalue alors la sortie de l'algorithme sur l'autre groupe et sur une autre vue. Par exemple, une fois entrainé sur le couple (vue 1, groupe A), l'algorithme sera testé sur (vue 2, groupe B) puis sur (vue 3, groupe B). Les performances finales sont évaluées comme la moyenne sur l'ensemble des 12 configurations possibles.

Ces deux protocoles mettent en valeur deux aspects différents. Le premier estime la capacité d'un algorithme à combiner les informations en provenance de différentes vues alors que le deuxième évalue la capacité d'un algorithme à généraliser les informations apprises à partir d'une seule vue.

### 4.4.2 Métriques retenues

Pour attribuer un score et classer les méthodes de reconnaissance d'actions, il existe une multitude de métriques utilisées dans les travaux de l'état de l'art [158], [162], [164]–[168]. Le choix d'une métrique ou d'une autre peut être crucial car elles n'évaluent pas toutes la même caractéristique d'une réponse d'un algorithme et sont souvent

complémentaires. Pour que des comparaisons justes puissent être faites sur notre jeu de données, nous décrivons ici les métriques retenues.

Pour chaque classe $c$, définissons :

***TP***$^c$ le nombre de vrais positifs, à savoir le nombre d'exemples détectés comme appartenant à la classe $c$ et appartenant effectivement à la classe $c$ selon les annotations.

***FP***$^c$ le nombre de faux positifs, à savoir le nombre d'exemples détectés comme appartenant à la classe $c$ mais n'appartenant pas à la classe $c$ selon les annotations.

***TN***$^c$ le nombre de vrais négatifs, à savoir le nombre d'exemples détectés comme n'appartenant pas à la classe $c$ et n'appartenant effectivement pas à la classe $c$ selon les annotations.

***FN***$^c$ le nombre de faux négatifs, à savoir le nombre d'exemples détectés comme n'appartenant pas à la classe $c$ mais appartenant en fait à la classe $c$ selon les annotations.

A partir de ces 4 mesures, nous introduisons les différentes métriques utilisées :

**Frame-wise Accuracy**

Cette métrique représente le ratio du nombre d'exemples correctement classifié divisé par le nombre total $N_c$ d'exemples appartenant effectivement à cette classe dans l'ensemble du jeu de données.

$$\mathcal{FA} = \frac{\sum_{c\in\mathcal{C}} TP^c}{\sum_{c\in\mathcal{C}} N_c} \tag{4.1}$$

Cette métrique est sensible à la distribution des classes dans le jeu de données, mais donne une mesure intuitive de la capacité d'un algorithme à segmenter les vidéos.

**Mesure $F_1$**

Cette métrique est définie à partir de la *Précision* $\mathcal{P}^c$ et du *Rappel* $\mathcal{R}^c$ de chaque classe $c$. Il s'agit de la moyenne harmonique de ces deux grandeurs pondérées de façon égale :

$$\mathcal{P}^c = \frac{TP^c}{TP^c + FP^c} \qquad \mathcal{R}^c = \frac{TP^c}{TP^c + FN^c} \tag{4.2}$$

$$\textit{F-Score} = \frac{2}{|\mathcal{C}|} \sum_{c\in\mathcal{C}} \times \frac{\mathcal{P}^c \times \mathcal{R}^c}{\mathcal{P}^c + \mathcal{R}^c} \tag{4.3}$$

Elle présente l'avantage de donner une importance égale à la précision et au rappel, deux grandeurs significatives de la capacité d'un algorithme à différencier des classes. Rappelons que le rappel représente le ratio du nombre d'exemples trouvés pour une classe par le nombre d'exemples appartenant à cette classe dans l'ensemble des exemples. La précision représente quant à elle le nombre d'exemples correctement attribués à une classe au regard du nombre total d'exemples attribués à cette classe par l'algorithme.

**Intersection sur Union IoU**

Cette métrique, relativement connue, a notamment été utilisée pour évaluer les segmentations dans le challenge PVOC [169]. Elle est définie comme le ratio de l'intersection sur l'union des prédictions avec les cibles de la vérité terrain :

$$\text{IoU} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{TP^c}{TP^c + FP^c + FN^c} \tag{4.4}$$

L'évaluation de ces trois métriques est faite de façon indépendante et permet une évaluation relativement juste de différents algorithmes. Cela évite notamment l'optimisation d'une méthode pour une seule métrique donnée dans le seul but d'afficher des scores supérieurs aux autres méthodes.

## 4.5 Evaluation

Afin de fournir une base de résultats avec laquelle les futurs travaux en reconnaissance d'activité pourront être comparés, nous avons évalué les résultats de trois méthodes de l'état de l'art. Ces méthodes ont été choisies pour leur compatibilité directe avec une application de localisation et reconnaissance simultanée d'activité. Par ailleurs, afin de proposer une évaluation juste de ces méthodes, nous avons considéré uniquement des méthodes dont le code source était mis en ligne par les auteurs.

Ainsi, nous proposons dans cette section des premières évaluations, sur le jeu de données DAHLIA des méthodes *Deeply Optimized Hough Transform (DOHT)* [63], *Efficient Linear Search (ELS)* [170] et d'une recherche *Max-graph* [171].

### 4.5.1 Deeply Optimized Hough Transform (DOHT)

Cet algorithme est présenté dans la section 3. Les évaluations ont été faites à partir des données *squelette* ainsi que des *Trajectoires Denses* [16]. Considérant le fait que la longueur des activités recherchées est bien plus grande que celle du jeu de données TUM [126] sur lequel s'évaluait le papier original, nous avons augmenté la taille de la fenêtre des votes : $M = 1000$ images.

**Evaluation à partir des données squelette**

Les coordonnées des articulations ont été normalisées de la même façon que présenté dans la section 3.3.2. Après normalisation, la représentation du squelette est alors indépendante de la vue considérée et elle ignore la localisation spatiale du squelette dans l'appartement. Cette dernière considération est importante pour éviter au maximum d'exploiter le biais apporté par l'unicité du lieu filmé dans ce jeu de données.

Nous considérons les articulations correspondant aux membres *Tête*, *Epaules*, *Coudes*, *Poignets*, *Mains*, *Doigts*, *Hanches*, *Genoux* et *Colonne Vertébrale*. Pour chaque instant de la vidéo, nous conservons uniquement les articulations associées à un indice de confiance

haut (*Visible et traqué*) par la Kinectv2. Par ailleurs, pour les instants où les *épaules* ont un indice de confiance faible, nous retirons le squelette complet, car c'est sur ces articulations qu'est basée la normalisation que nous utilisons.

Dans la configuration *Cross-subject*, nous utilisons les informations provenant des trois vues pour compenser les éventuelles occultations qui pourraient survenir. Pour cela, nous suivons la méthode présentée en section 3.3.2 et l'appliquons aux squelettes issus de chacune des trois vues. Ainsi, lorsqu'une articulation est occultée dans une des vues, l'algorithme peut quand même recevoir une information pertinente en provenance d'un autre capteur.

Le Tableau 4.2 présente les résultats obtenus à l'aide du squelette sur le jeu de données DAHLIA.

| | **Squelette** | | |
|---|---|---|---|
| | $\mathcal{FA}_1$ | F-Score | IoU |
| **Vue 1** | 0.60 | 0.58 | 0.42 |
| **Vue 2** | 0.63 | 0.60 | 0.44 |
| **Vue 3** | 0.73 | 0.71 | 0.56 |
| **Multi-vues** | 0.77 | 0.75 | 0.60 |
| **Cross-Vues** | 0.34 | 0.31 | 0.19 |

Tableau 4.2 – Résultats obtenus à partir du DOHT avec des descripteurs basés squelette sur le jeu de données DAHLIA.

Comme attendu, la fusion d'informations en provenance de plusieurs vue améliore nettement les résultats puisque le manque d'information sur une vue causé par une occultation peut être compensé par les autres vues.

Les performances obtenues dans le cadre de l'évaluation *Cross-Vues* sont bien en dessous des résultats obtenus en ne considérant qu'une seule vue. Ils montrent la difficulté du jeu de données lorsque l'on ne connaît pas l'angle de vue considéré, et ce, malgré une normalisation des squelettes.

**Evaluation à partir des trajectoires denses**

A partir de l'algorithme fourni par [16], nous avons extrait des trajectoires sur une grille dense de points. Ces trajectoires exploitent les informations contenues dans les vidéos RVB de la base de données. Nous considérons ici les descripteurs de trajectoire (TS), HoG ainsi que les descripteurs *TS+HOG* présenté en section 3.3.2

Le Tableau 4.3 présente les résultats obtenus à l'aide de ces descripteurs.

Ces résultats sont supérieurs à ceux obtenus avec les descripteurs basés squelette. Plus précisément, le descripteur HoG est celui associé aux meilleurs résultats obtenus avec cette méthode. Cela confirme l'importance du contexte spatial visuel qu'apporte ces descripteurs.

Pour analyser plus précisément ces résultats, le Tableau 4.4 présente les résultats obtenus classe par classe avec le descripteur HoG. Remarquons que les classes les mieux

| | Trajectories | | | HOG | | | Traj+HOG | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\mathcal{FA}_1$ | F-Score | IoU | $\mathcal{FA}_1$ | F-Score | IoU | $\mathcal{FA}_1$ | F-Score | IoU |
| **View 1** | 0.74 | 0.73 | 0.58 | 0.80 | 0.77 | 0.64 | 0.73 | 0.73 | 0.59 |
| **View 2** | 0.78 | 0.76 | 0.62 | 0.81 | 0.79 | 0.66 | 0.79 | 0.78 | 0.64 |
| **View 3** | 0.76 | 0.74 | 0.59 | 0.80 | 0.77 | 0.65 | 0.77 | 0.76 | 0.62 |
| **Multiviews** | 0.81 | 0.80 | 0.67 | 0.85 | 0.82 | 0.71 | 0.82 | 0.80 | 0.68 |

Tableau 4.3 – Résultats obtenus à partir du DOHT avec des descripteurs basés images sur le jeu de données DAHLIA.

reconnues sont *Travailler* et *Prendre son déjeuner*.

| | HoG Multi-vues | |
|---|---|---|
| | F-Score | IoU |
| **Cuisiner** | 0.75 | 0.60 |
| **Dresser la table** | 0.69 | 0.53 |
| **Prendre son Déjeuner** | 0.91 | 0.84 |
| **Débarasser la table** | 0.75 | 0.59 |
| **Faire la vaisselle** | 0.87 | 0.77 |
| **Faire le Ménage** | 0.86 | 0.75 |
| **Travailler** | 0.92 | 0.86 |

Tableau 4.4 – Performances classe par classe obtenues à partir d'un descripteur HoG dans un paradigme multi-vues.

### 4.5.2 Online Efficient Linear Search (ELS) [170]

Meshry, Hussein et Torki proposent dans [170] en 2016 une méthode de détection d'actions basée sur des séquences de coordonnées 3D de squelettes. Un dictionnaire est généré à partir des descripteurs extraits. Un poids $w_c^a$ est appris pour chaque mot $c$ et chaque action $a$ au travers d'un SVM. Chaque instant temporel est alors associé à un score $f(i)$ correspondant à la somme des poids des descripteurs extraits à cet instant. La localisation d'une action consiste ensuite en la détection de l'intervalle $\hat{I}_a$, dont la somme des scores $f(t)$ associés aux instants $i \in \hat{I}_a$ qu'il contient est maximale.

$$s_a = \arg\max_{i \subseteq \hat{I}_a} f(s) \qquad f(i) = \sum_{j=1}^{n} w_{c_j}^a \tag{4.5}$$

Cette détection est de complexité linéaire grâce à l'algorithme de Kanade [172]. L'action $a$ est considérée comme détectée si ce score est supérieur à un seuil $\theta_a$ appris pendant la phase d'entraînement.

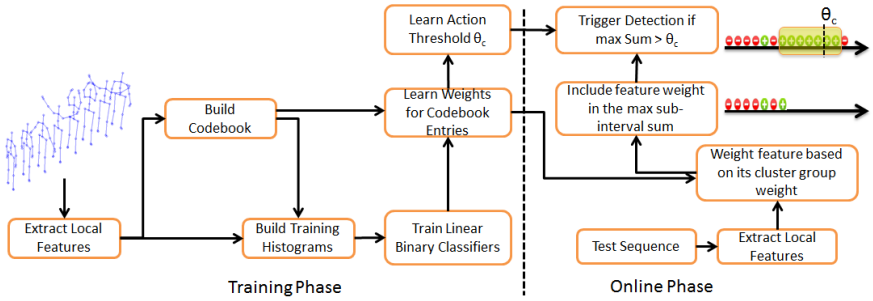



FIGURE 4.11 – Illustration de la méthode ELS [170]. Image extraite de l'article original.

| | Squelette | | |
|---|---|---|---|
| | $\mathcal{FA}_1$ | F-Score | IoU |
| **Vue 1** | 0.18 | 0.18 | 0.11 |
| **Vue 2** | 0.27 | 0.26 | 0.16 |
| **Vue 3** | 0.52 | 0.55 | 0.39 |
| **Cross-Vues** | 0.31 | 0.32 | 0.21 |

TABLEAU 4.5 – Resultats de la méthode ELS [170] sur le jeu de données DAHLIA

Dans nos tests, pour une évaluation équitable avec le DOHT, nous utilisons la version *en ligne* qu'ils proposent. Dans cette version, la recherche d'un intervalle maximal est faite au fur et à mesure de l'avancée de la vidéo. Lorsque le score de l'intervalle considéré dépasse $\theta_a$, l'action est considérée détectée. On connait alors le début de l'action (c'est le début de l'intervalle courant). Cet intervalle est considéré terminé lorsqu'une succession de $N_a$ instants à score négatif est détectée. En pratique, les auteurs fixent le paramètre $N_a = 1$. La Figure 4.11 illustre leur méthode.

Pour ces premiers résultats, nous conservons les caractéristiques définies dans le papier original [170], il s'agit d'une concaténation pondérée des angles inter-articulation [173] $\theta$, de leur dérivées $\delta\theta$ ainsi que d'une adaptation du descripteur *Moving Pose* [174] $P$ et de ses dérivées première et seconde $\delta P$ et $\delta^2 P$. Le descripteur final est alors de la forme $[P, \alpha\delta P, \beta\delta^2 P, \psi\Theta, \psi\delta\Theta]$, avec $\alpha, \beta$ et $\psi$ trois poids, paramètres de la méthode. Sur DAHLIA, nous avons évalué plusieurs jeux de paramètres et présentons ici celui donnant les meilleurs résultats : $\alpha = 0.1, \beta = 0.1, \psi = 0.1$. Les autres paramètres sont gardés tels que dans le papier original. Le Tableau 4.5 présente les résultats obtenus avec cette méthode.

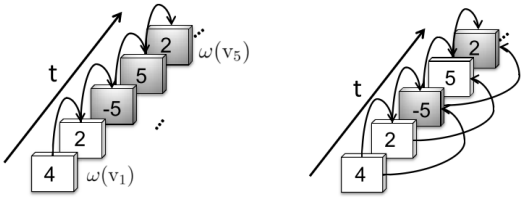
Remarquons que les résultats varient fortement d'une vue à l'autre. La vue sur laquelle les prédictions sont les moins erronnées est la vue 3, vue la moins propice aux auto-occultations par sa position dans la scène.

Les performances en cross-vues sont similaires à celle obtenues avec le DOHT.

### 4.5.3 Recherche Max-Subgraph

Chen et Grauman présentent dans [171] un détecteur d'actions nommé *T-jump-subgraph*. Un graphe est construit pour chaque action à détecter, avec un nœud pour chaque instant temporel. À partir des descripteurs extraits à chaque instant, un score est associé à chacun des nœuds. Il s'agit alors de trouver le graphe de score maximal pour chaque action.

Pour générer ces graphes, les auteurs proposent deux stratégies : l'une n'autorisant un lien qu'entre deux nœuds successifs, l'autre permettant le *saut* d'un instant temporel. Cette deuxième méthode se veut plus robuste au bruit pouvant interrompre un graphe de manière intempestive. La Figure 4.12 illustre ces deux stratégies.




(a) Neighbors only (b) Neighbors + "Jump"

Figure 4.12 – Illustration des deux stratégies proposées par [171]. Chaque rectangle est un nœud et les chiffres représentent les poids associés à ces nœuds.

Le Tableau 4.6 résume les résultats obtenus avec cette méthode. Notons que puisque plusieurs activités peuvent être détectées à chaque instant, le calcul du $\mathcal{FA}_1$ ne peut être fait.

| | F-Score | IoU |
|---|---|---|
| **Vue 1** | 0.25 | 0.15 |
| **Vue 2** | 0.18 | 0.10 |
| **Vue 3** | 0.44 | 0.31 |

Tableau 4.6 – Résultats avec la méthode Max-subgraph Search [171] sur le jeu de données DAHLIA

## 4.6 Conclusion sur le jeu de données DAHLIA

Nous avons, dans ce chapitre, effectué un bilan des différents jeux de données précédemment publiés dans le domaine de l'analyse du comportement humain. Après avoir mis en avant le manque de jeux de données adaptés à la détection d'activités, nous avons présenté un nouvel ensemble de vidéos.

Cette base de données, appelée *DAHLIA*, est adaptée à la détection et reconnaissance d'activités humaines à haut niveau sémantique. En effet, elle est composée de 51 séquences longues non pré-découpées contenant des actions de la vie quotidienne

réalisées par 44 personnes différentes. Les trois Kinectv2 utilisées pour son acquisition nous ont permis de mettre à disposition des flux de données de 4 types différents, sous trois angles de vues. Ceci rend la méthode compatible avec un large spectre de méthodes et permet la fusion d'informations provenant de différents points de vues et types de données.

Nous avons comparé cette nouvelle base aux jeux précédemment décrits dans le domaine de la reconnaissance d'actions. Nous avons démontré sa supériorité en terme de durée et avons mis en avant le haut niveau sémantique des activités complexes qu'elle contient ; celles-ci pouvant être décomposées en de multiples sous-actions. Un soin particulier a été apporté à l'élaboration d'un protocole favorisant la fluidité et l'aspect naturel des activités impliquées.

Enfin, nous avons défini deux protocoles d'évaluation ainsi que des métriques adaptées afin de permettre une comparaison juste des prochains travaux qui se testeront sur ce jeu. A cet effet, nous avons présenté des premiers résultats sur des méthodes de détection d'actions de l'état de l'art mettant leur code à disposition. Les résultats faibles des trois méthodes testées, notamment sur le protocole cross-view, montre le défi que représente l'analyse d'activités dans un environnement réaliste.

Les travaux présentés dans ce chapitre ont fait l'objet d'une publication dans la douzième conférence internationale *Automatic Face and Gesture Recognition (FG 2017)* [3].

Chapitre 5

# Détection hiérarchique d'activités humaines

## Introduction

L'écart important qui existe entre le niveau sémantique ainsi que la dimension temporelle des gestes comparés aux activités suscite l'intuition selon laquelle la détection d'activités à haut niveau sémantique à partir de descripteurs bas niveau classiques (comme les trajectoires de points d'intérêts) est trop complexe. Par exemple, déterminer, pour l'activité « mange », à chaque instant, la probabilité que la personne commence à lever le bras semble difficile et inappropriée. Par contre, en introduisant des actions élémentaires du type « la personne lève puis baisse le bras », il semble plus simple :

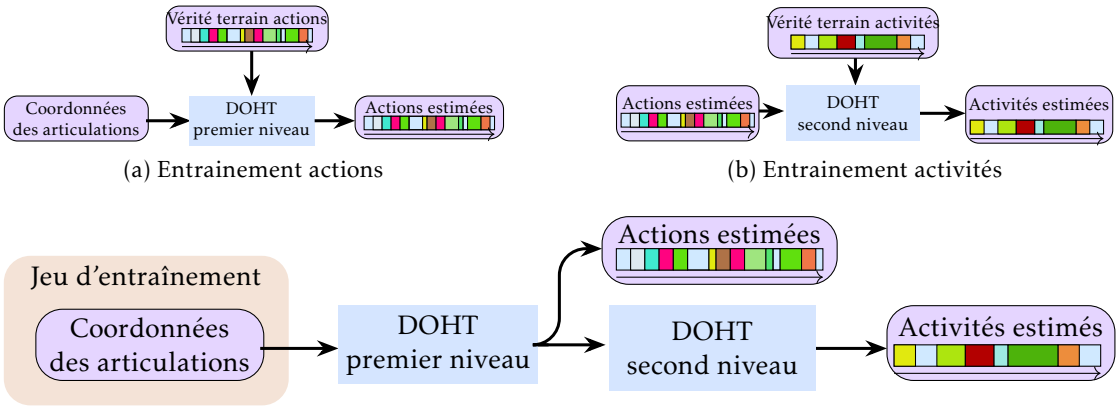
— de détecter ces actions élémentaires en utilisant des gestes du type « commence à lever le bras ». La probabilité de ce geste à chaque instant de l'action élémentaire a alors un sens ;

— de détecter des activités comme « manger » à partir de ces actions élémentaires.

Cette intuition a suscité le paradigme que nous proposons. Il s'agit d'une méthode à deux niveaux dont la hiérarchie est similaire aux différentes définitions données pour les gestes, les actions et les activités. L'idée est donc d'apprendre les activités à partir des actions élémentaires et d'apprendre ces actions à partir de gestes simples.

Nous proposons donc un apprentissage d'activité à deux niveaux. Le premier prend en entrée des descripteurs bas niveau (pose du squelette par exemple) pour prédire des actions élémentaires. Le second prend en entrée ces actions élémentaires puis prédit les activités haut niveau. Ces deux phases sont illustrées sur la Figure 5.1. Ayant constaté l'efficacité de l'algorithme DOHT pour la reconnaissance d'actions et d'activités à partir des gestes extraits, nous avons décidé de mettre en place une architecture composée de deux algorithmes DOHT.

La première étape consiste donc en une détection d'actions élémentaires à partir des vidéos. Pour apprendre ces actions élémentaires, l'algorithme DOHT nécessite




(a) Entrainement actions

(b) Entrainement activités

(c) Evaluation

FIGURE 5.1 – Détection hiérarchique d'activités

une base étiquetée en actions élémentaires. Pour éviter une annotation manuelle des flux vidéo en actions élémentaires et afin de rendre possible l'estimation d'actions les plus discriminantes possibles, nous proposons d'apprendre les actions élémentaires intermédiaires de façon semi-supervisée. L'extraction et l'apprentissage de ces actions seront donc réalisés uniquement à partir des étiquettes d'activité, sans vérité terrain plus bas niveau.

Ce chapitre décrit dans un premier temps la génération d'étiquettes d'actions de façon semi-supervisée puis s'intéresse ensuite à la détection d'activité.

## 5.1 Génération semi-supervisée d'actions élémentaires

Le processus de génération des actions élémentaires est illustré Figure 5.2. Il se divise en plusieurs étapes :

1. Segmentation des flux vidéo en segments (section 5.1.1),
2. Description des segments (section 5.1.2),
3. Apprentissage de métrique (section 5.1.2).
4. Génération des annotations en actions élémentaires des flux vidéo (section 5.1.3).

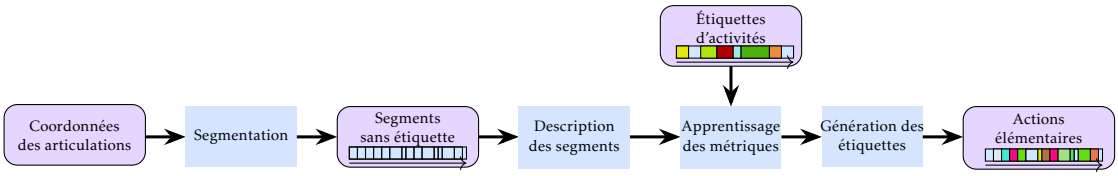



FIGURE 5.2 – Génération semi-supervisée d'actions élémentaires. Seules les activités sont connues, les actions élémentaires sont issues d'une segmentation non-supervisée suivi d'un apprentissage de métrique supervisée par les activités puis d'un regroupement type *k-moyennes*

### 5.1.1 Segmentation non-supervisée des flux vidéo.

En premier lieu, on cherche à générer des segments temporels qui seront ensuite étiquetés en actions élémentaires. Cette segmentation se fait en l'absence d'annotations et est donc non-supervisée. Pour cela, nous adaptons la méthode de KRÜGER, VÖGELE, WILLIG et al. présentée dans [175]. Elle a été originalement pensée pour segmenter une vidéo en actions élémentaires avant d'en extraire des gestes répétés et est donc parfaitement adaptée à notre problème.

L'algorithme présenté dans [175] se décompose en 4 étapes :

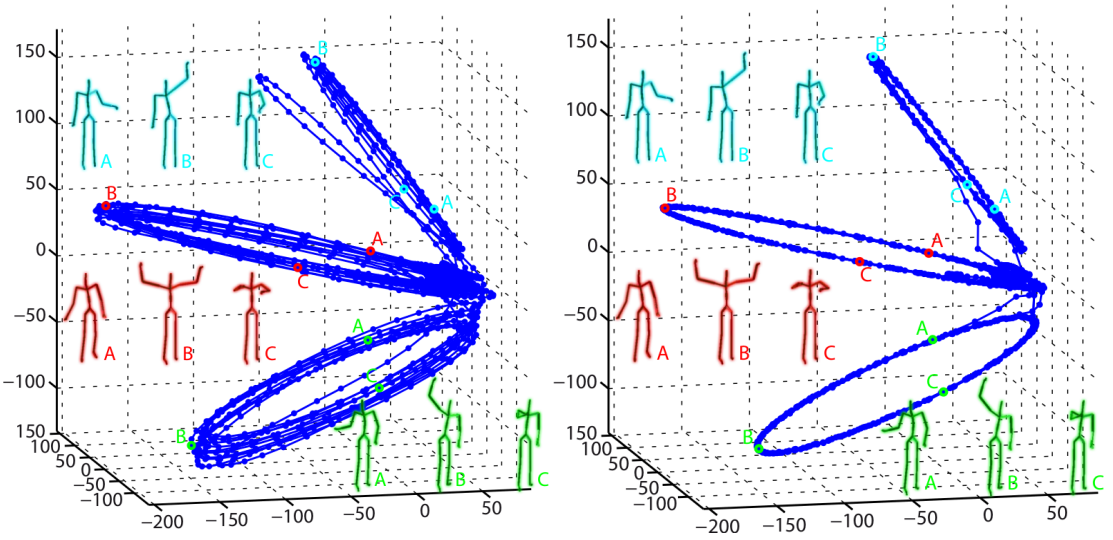
1. **Pré-traitement des données pour faciliter la recherche de similarités,**
2. **Segmentation du flux vidéo en actions élémentaires,**
3. Sous-division en gestes élémentaires,
4. Regroupement des gestes élémentaires.

Dans le contexte de cette thèse, nous nous concentrons sur les deux premières étapes afin d'obtenir des segments correspondant à des activités non étiquetées qui serviront de base à la recherche d'actions élémentaires. Nous décrivons ces deux étapes dans la suite de cette section.

#### Pré-traitement des données

Partant du constat que deux actions élémentaires similaires (sémantiquement ou visuellement) peuvent générer des descripteurs relativement différents, y compris dans un espace dédié, les auteurs de [175] proposent un pré-traitement des données pour les rendre plus facilement exploitables.

L'objectif est de réduire les différences qui peuvent apparaître dans l'espace $\mathcal{X}$ des descripteurs entre deux réalisations d'une même action élémentaire. Il s'agit d'aligner topologiquement les descripteurs des exemples d'une même classe. En pratique, on cherche à rassembler, à l'intérieur d'un même flux vidéo, les points qui correspondent à une même action élémentaire ; et ceci tout en conservant une plus grande distance entre les actions élémentaires sémantiquement différentes ou présentant des fortes différences dans leur réalisation.



(a) Avant traitement (b) Après traitement

FIGURE 5.3 – Illustration de l'étape de préparation des données. A gauche une représentation des données brutes dans l'espace des descripteurs, à droite les mêmes données après traitement. Figure extraite de [175]

Rappelons que cette transformation se fait de façon non-supervisée et donc sans connaitre les actions élémentaires constituant les flux vidéo. Pour compenser ce manque d'étiquette, on pré-suppose que les trajectoires de deux actions élémentaires différentes sont plus éloignées dans $\mathcal{X}$ que deux réalisations d'une même action. Le rassemblement des descripteurs se fait alors à partir des plus proches voisins.

L'effet de ce pré-traitement est illustré sur la Figure 5.3 et la méthode de transformation est décrite ci-après.

Soit $\mathbf{x}_i \in \mathcal{X}$, le descripteur décrivant la $i^{\text{ème}}$ image d'un flux vidéo $v \in \mathcal{V}$ de $N$ instants temporels ; avec $i = \{1,\dots,N\}$ et $\mathcal{X}$, un espace de dimension $D$. On s'intéresse à la trajectoire décrite par la suite des $\mathbf{x}_i$ dans l'espace $\mathcal{X}$ des descripteurs lorsque $i$ croît de 1 à $N$. Pour chaque $\mathbf{x}_i$ l'objectif est d'obtenir un nouveau vecteur $\hat{\mathbf{x}}_i$ représentant $\mathbf{x}_i$ mais étant plus proche des descripteurs de la même étape d'un autre exemple de cette action élémentaire. Pour cela, les auteurs de [175] proposent une méthode décrite par les quatre étapes suivantes :

1. Pour chaque $\mathbf{x}_i$, trouver les *k plus proches voisins* dans $\{\mathbf{x}_j \mid j \in \{1,\dots,N\} \setminus i\}$,
2. Projeter ces plus proches voisins dans un sous-espace estimé à partir de la direction de la trajectoire de $\mathbf{x}_i$.
3. Estimer la densité de probabilité des descripteurs par noyau et en déduire la nouvelle valeur filtrée de $\mathbf{x}_i$ : $\hat{\mathbf{x}}_i$.

Ces données prétraitées vont maintenant être utilisées pour segmenter chaque flux vidéo

en segments non étiquetés.

**Segmentation non supervisée**

La segmentation proposée par , Krüger, Vögele, Willig et al. [175] est composée d'une étape d'aggrégation temporelle des descripteurs, puis de la construction d'une matrice creuse de similarité.

**Aggrégation temporelle des descripteurs** Chaque instant $i \in \{1, \ldots, N\}$ du flux vidéo $v$, est représenté par un descripteur $\hat{\mathbf{x}}_i$. Ces descripteurs décrivent la pose de la personne à un instant précis sans prise en compte de l'évolution temporelle. Pour enrichir cette représentation et ajouter une dimension temporelle, les descripteurs sont concaténés temporellement pendant $l$ images de manière à obtenir un nouveau descripteur :

$$\mathbf{X}_i = \begin{bmatrix} \hat{\mathbf{x}}_{i+1} \\ \vdots \\ \hat{\mathbf{x}}_{i+l} \end{bmatrix}.$$

**Génération d'une matrice creuse de similarité** La segmentation repose ensuite sur l'étude de la similarité des descripteurs au cours du temps. Pour chaque instant $i$, on cherche à trouver les plus proches voisins de $\mathbf{X}_i$ compris dans un rayon $r$ autour de ce point. Ce rayon $r$ est défini à partir d'un hyper-paramètre $R$, appelé rayon généralisé, dépendant de la dimension $D$ d'un descripteur et du nombre de décalages temporels considérés $l$ :

$$r = R\sqrt{l \cdot D}.$$

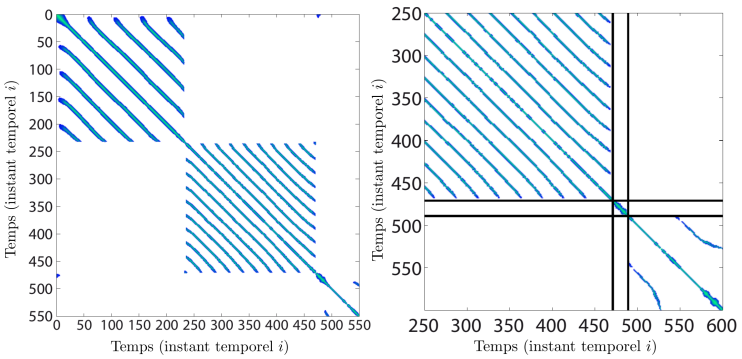
Les plus proches voisins $\mathbf{X}_j$ peuvent être représentés sous la forme d'une matrice creuse $\mathcal{M}$ dont chaque élément $\mathcal{M}_{i,j}$ vaut $d_{i,j}$ (distance entre $\mathbf{X}_i$ et $\mathbf{X}_j$) si $\mathbf{X}_j$ est un des plus proches voisins de $\mathbf{X}_i$ et 0 sinon. Cette matrice est appelée *Matrice creuse d'auto similarité* ou *SSSM* pour *Sparse Self Similarity Matrix*. Notons qu'il s'agit rigoureusement d'une matrice contenant des distances et non des similiratés, mais nous conservons dans la suite le nom *SSSM* pour rester cohérent avec l'article original [175].

La Figure 5.4a montre un exemple de SSSM pour deux actions : *"Marcher"* et *"Courir"*. On constate que ces deux actions sont chacune caractérisées par un bloc au sein de la matrice. La découpe obtenue après application de l'algorithme de segmentation est représentée sur la Figure 5.4b.

**Application à notre cas d'usage**

L'algorithme présenté par Krüger, Vögele, Willig et al. [175] a été conçu pour la détection d'activités cycliques. Ce caractère cyclique est à l'origine des diagonales observées en Figure 5.4a.

Cette thèse s'intéresse à des activités quotidiennes haut-niveau du type *"Prendre son repas"*, *"Débarasser"*, *"Faire la vaisselle"*, *etc.* De telles activités, par leur nature




(a) Matrice SSSM. (b) Segmentation résultante

FIGURE 5.4 – Illustration de l'algorithme de segmentation en actions élémentaires. Figures extraite de [175]

contiennent certains gestes répétés pouvant être assimilés à des cycles. Par exemple, l'activité *"Prendre son repas"* contient de multiples occurrences de l'action élémentaire *"attraper un aliment, le manger et descendre sa fourchette"*. C'est ce type d'actions élémentaires que nous cherchons à segmenter de façon non-supervisée. Notons en outre que les actions segmentées de façon non supervisée peuvent ne pas avoir de vérité sémantique. Puisque nous ne nous intéressons pas aux actions en elles-même mais bien aux activités plus haut niveau, ce dernier point est compatible avec notre méthode.

Notons également que dans l'article original [175], les auteurs s'évaluent sur des données contenant des actions cycliques, mais également sur un jeu de données contenant des actions plus variées comme le jeu de données *MSR Online Action* [176]. Ce jeu contient par exemple des actions des classes *"Boire"*, *"Décrocher le téléphone"*, *"Ecrire un SMS"*, *etc.* très proches des actions que nous recherchons.

Appliquée au jeu de données DAHLIA, la segmentation à l'aide de cet algorithme donne des résultats tels que celui représenté sur la Figure 5.5.

Cette segmentation découpe chacun des flux vidéos du jeu d'entraînement indépendamment et génère un nombre $N_S$ de segments non étiquetés ne permettant pas de déterminer les actions élémentaires qui se répètent. Les segments doivent donc maintenant être étiquetés en *actions élémentaires* et avant cela, être représentés par un descripteur.

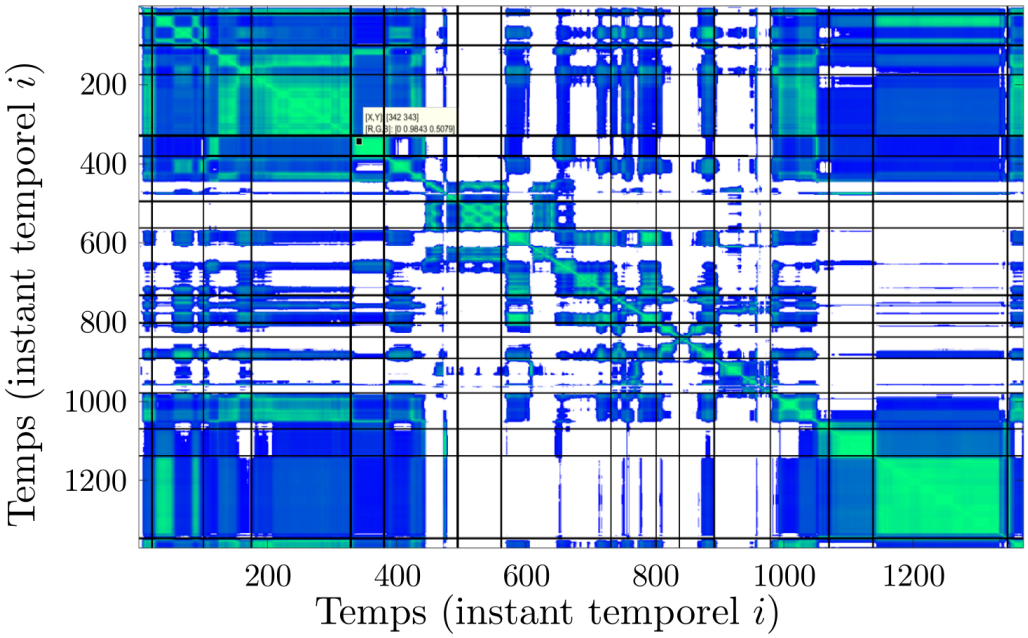



FIGURE 5.5 – Exemple de matrice SSSM obtenue sur DAHLIA

### 5.1.2 Description des segments non étiquetés

En vue d'associer à chacun de ces segments une étiquette d'action élémentaire discriminante pour les activités que nous désirons détecter, les segments doivent être décrits de manière à pouvoir être comparés. La taille de ce descripteur doit être constante quel que soit la longueur du segment, tout en conservant un maximum d'information pour permettre une comparaison la plus fiable possible.

Fortement inspiré des travaux de CARBONERA LUVIZON, TABIA et PICARD dans [60], nous représentons les segments à l'aide d'une description utilisant les VLAD (*Vector of Locally Aggregated Descriptors*) [72]. Cette section décrit l'adaptation de ces travaux pour la représentation des segments obtenus à l'étape précédante.

#### Les descripteurs utilisés

Après extraction des positions (coordonnées) des articulations au sein d'une vidéo, nous décrivons les poses de chaque image à l'aide du descripteur de *positions relatives* décrit dans [60] .

Soit $\mathbf{z}_i^j$ le vecteur représentant la position de l'articulation $j$ au $i^{ème}$ instant du flux vidéo $v \in \mathcal{V}$. La position relative de l'articulation $j$ par rapport à l'articulation $k$ se définit comme :

$$\mathbf{p}_i^{j,k} = \mathbf{z}_i^j - \mathbf{z}_i^k. \tag{5.1}$$

Dans [60] , ces positions relatives sont regroupées en sous-groupes d'articulations pour décrire le squelette. Cela facilite l'étape de quantification (un nombre de centres

plus petit lors de l'étape des *k-moyennes* par exemple).

Suivant cette idée, nous définissons quatre sous-groupes d'articulations, que nous appellons *membres* dans la suite de cette thèse, décrit dans le Tableau 5.1 suivant.

| Descripteur Associé | Membre | Articulations considérées |
|---|---|---|
| $\mathbf{f}^1$ | bras gauche | *Epaule, coude et poignet gauches* |
| $\mathbf{f}^2$ | bras droit | *Epaule, coude et poignet droits* |
| $\mathbf{f}^3$ | jambe gauche | *Hanche, genou et cheville gauche* |
| $\mathbf{f}^4$ | jambe droite | *Hanche, genou et cheville droits* |

TABLEAU 5.1 – Membres utilisés pour la génération d'étiquettes non supervisée

La position relative de chacune de ces articulations est calculée par rapport au centre des épaules. Ces quatre membres ainsi que l'articulation de référence (centre des épaules) sont représentés sur la Figure 5.6

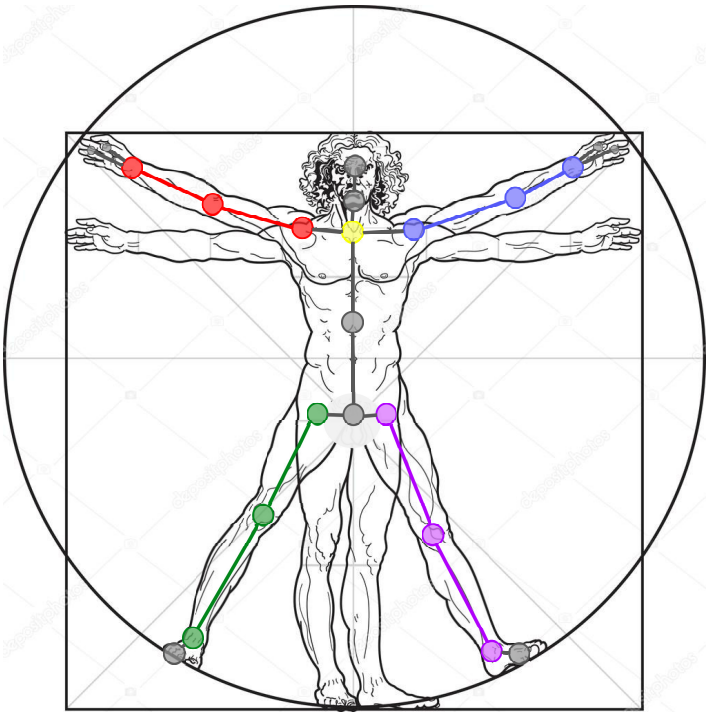


FIGURE 5.6 – Articulations considérées pour définir les membres représentés de couleurs différentes. L'origine considérée pour le calcul des positions relatives est représentée en jaune

Chacun de ces $M$ membres $m \in 1,\ldots,M$ génère alors un vecteur descripteur $\mathbf{f}_i^m$ qui est la concaténation des vecteurs $\mathbf{p}_i^{j,k}$ des articulations $j$ du membre, avec $k$ désignant l'articulation de référence (centre des épaules).

**Aggrégation de descritpeurs**

Pour chaque instant $i$ de chaque segment $v_S$, on a maintenant un ensemble de descripteurs correspondant à chacun des membres considérés. Ainsi, le segment $v_S$ de longueur $\tau$ est décrit par $M$ séquences $[\mathbf{f}_1^m, \ldots, \mathbf{f}_\tau^m]$, avec $m \in 1, \ldots, M$.

On cherche à décrire tous les segments issus de tous les flux vidéo dans un espace commun afin de les comparer et de définir les actions élémentaires. Pour cela, les auteurs [60] proposent une description à l'aide de descripteurs VLAD [72]

Tout d'abord, pour chaque membre, un algorithme des k-moyennes est appliqué sur tous les $\mathbf{f}_i^m$ de tous les segments $v_s$. Chacun de ces $k$ groupes est représenté par son centre $\mu_k$.

Soit $\mathcal{S}_k^m$, l'ensemble des descripteurs du membre $m$ de $v_s$ associé au centre $k$ :

$$S_k^m = \left\{ \mathbf{f}_i^m \in v_s \mid k = \arg\min_{k'} \left\| \mathbf{f}_i^m - \mu_{k'}^m \right\| \right\}, \tag{5.2}$$

On définit alors $\chi^{m,k}$ comme

$$\chi^{m,k} = \sum_{\mathbf{f}_i^m \in S_k^m} \left( \mathbf{f}_i^m - \mu_k^m \right). \tag{5.3}$$

Le segment $v_s$ est représenté par la concaténation des vecteurs $\chi^{m,k}$ pour les différents membres et les différents centres, vecteur de dimension $k \times M \times D$ où D est la dimension des descripteurs $\mathbf{f}_i^m$.

En réalisant cette caractérisation $c$ fois ($c$ initialisations différentes des k-moyennes), la caractérisation finale est de taille $c \times k \times M \times D$. Chaque segment $v_s$ est maintenant représenté par un descritpeur $\mathbf{F}_s$ de taille fixe, indépendamment sa longueur. Ce processus est illustré sur la Figure 5.7.

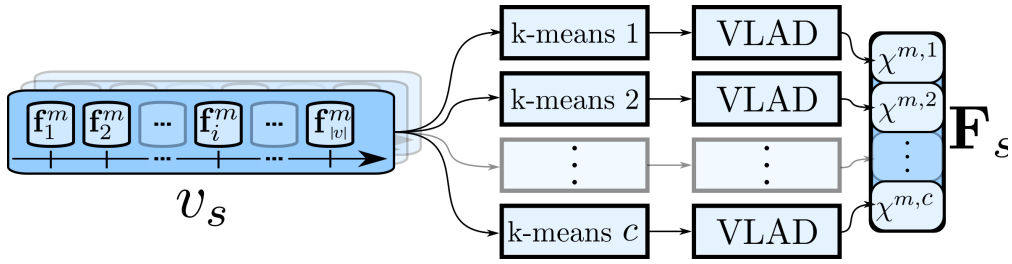



Figure 5.7 – Description des segments d'actions élémentaires. En entrée, un segment $v_s$, en sortie un descripteur $\mathbf{F}_s$ de taille constante au travers des segments $v_s$

En vue d'associer à ces segments des étiquettes d'actions élémentaires discriminantes vis-à-vis des activités haut niveau à reconnaître, une transformation rapprochant les segments d'une même activité et séparant les segments associés à des activités différentes est ensuite mise en place. Cette étape à pour but de faciliter une classification non-supervisé, qui pourrait être réalisée à partir d'un algorithme des *k-moyennes*, en vue de

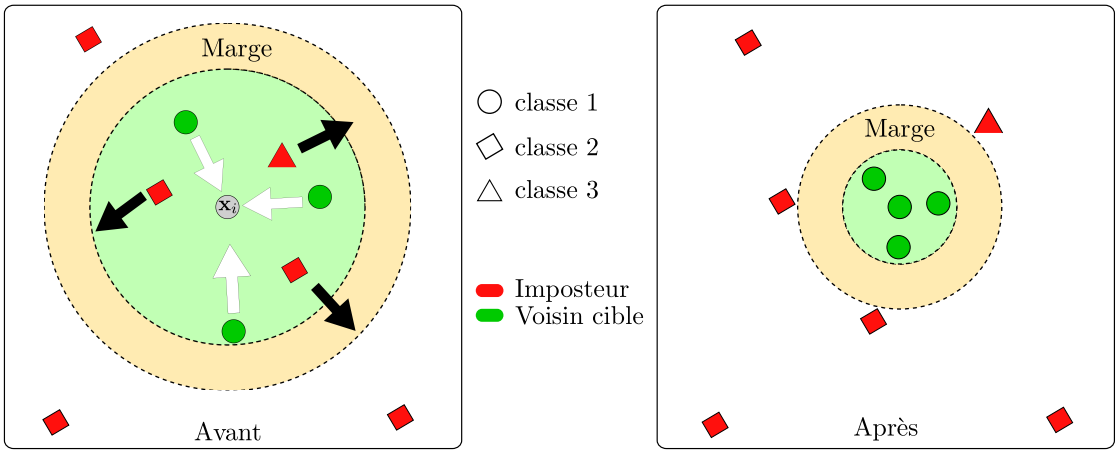



FIGURE 5.8 – Illustration de la méthode LMNN [177]

générer les étiquettes d'actions élémentaires désirées.

### Apprentissage de métrique

Ce problème correspond à un problème d'apprentissage de métrique. Nous proposons de le résoudre en utilisant l'algorithme des plus proches voisins à large marge (*LMNN pour Large Margin Nearest Neighbor*) proposé par WEINBERGER et SAUL dans [177] et [178]. Ces travaux ont été présentés pour la première fois en 2006 [178], puis ont été étendus en 2008 [179]. Ils sont largement inspirés des travaux de GOLDBERGER, HINTON, ROWEIS et al. [180] et CHOPRA, HADSELL et LECUN [181].

L'algorithme LMNN *pour Large Margin Nearest Neighbor*, illustré en Figure 5.8 a été conçu pour améliorer une classification utilisant l'algorithme des *k-moyennes* à partir de deux intuitions simples :

1. chaque exemple $\mathbf{x}_i$ doit être associé à la même étiquette $a_i$ que ses $k$ plus proches voisins,
2. les exemples ne partageant pas la même étiquette doivent être éloignés les uns des autres.

L'objectif du LMNN est donc d'apprendre une *distance* avec laquelle les exemples d'une même classe sont proches et les exemples de classes différentes sont éloignés.

Rappelons qu'une fonction $d : \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R}^+$ est une distance si et seulement si, $\forall(\mathbf{x},\mathbf{y},\mathbf{z}) \in \mathcal{Z}^3$

1. $d(\mathbf{x},\mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}$ (identité des indiscernables),
2. $d(\mathbf{x},\mathbf{y}) = d(\mathbf{y},\mathbf{x})$ (symétrie),
3. $d(\mathbf{x},\mathbf{z}) \leq d(\mathbf{x},\mathbf{y}) + d(\mathbf{y},\mathbf{z})$ (inégalité triangulaire),
4. $d(\mathbf{x},\mathbf{z}) \geq 0$ (positivité).

La quatrième condition étant impliquée par les trois autres. Notons que si la première condition n'est pas remplie, mais que les trois suivantes le sont, $d$ désigne alors une pseudo-métrique.

On peut obtenir une famille de distances $\mathcal{D}_{\mathbf{L}}$ à partir de la distance euclidienne $\|\cdot\|_2^2$ et d'une transformation linéaire $\mathbf{L}$ :

$$\mathcal{D}_L(\mathbf{x},\mathbf{y}) = \|\mathbf{L}(\mathbf{x}-\mathbf{y})\|_2^2. \tag{5.4}$$

Si $\mathbf{L}$ est de rang plein, la condition d'*identité des indiscernables* est respectée et $\mathcal{D}_{\mathbf{L}}$ désigne une distance ; autrement, $\mathcal{D}_{\mathbf{L}}$ désigne une pseudométrique. On cherche à apprendre une matrice $\mathbf{L}$ telle que la distance $\mathcal{D}_{\mathbf{L}}$ remplisse nos objectifs. Pour cela, les auteurs de [177] proposent de déterminer $\mathbf{L}$ par l'optimisation d'une fonction de coup à deux termes $\epsilon_{\text{pull}}$ et $\epsilon_{\text{push}}$. Le premier pénalise les larges distances entre deux exemples d'une même classe, le second pénalise les faibles distances entre deux exemples de classes différentes. Pour définir plus précisément les notions de *larges* et *faibles* distances, on introduit les notions de *voisins cibles* et d'*imposteurs*.

**Voisins cibles** Pour chaque exemple $\mathbf{x}_i$, les *voisins cibles* sont les points que l'on cherche à rapprocher de $\mathbf{x}_i$. Ces *cibles* sont déterminées *a priori* et n'évoluent pas durant l'algorithme. Sans information *a priori* sur les données, comme c'est notre cas, une façon simple de définir un ensemble de *voisins cibles* est de considérer les $k$ plus proches voisins appartenant à la même classe $a_i$. Pour indiquer qu'un exemple $\mathbf{x}_j$ est un *voisin cible* de $\mathbf{x}_i$, on utilise la notation $i \rightsquigarrow j$. Notons que $i \rightsquigarrow j$ n'implique pas $j \rightsquigarrow i$.

Les voisins cibles permettent d'introduire le premier terme de la fonction de coût proposée dans [177] :

$$\epsilon_{\text{pull}} = \sum_{i,j\rightsquigarrow i} \|\mathbf{L}(\mathbf{x}_i-\mathbf{x}_j)\|^2. \tag{5.5}$$

Une descente de gradient selon $\mathbf{L}$ sur ce terme "attire" les voisins cibles vers $\mathbf{x}_i$.

**Imposteurs** On définit pour chaque $\mathbf{x}_i$, ses *imposteurs* comme étant les exemples $\mathbf{x}_k$ plus proches de $\mathbf{x}_i$ qu'un *voisin cible* $\mathbf{x}_j$. Ce sont les exemples $\mathbf{x}_k$ associés à une classe $a_k$ différente de $a_i$ tels que $d_{\mathbf{L}}(\mathbf{x}_i,\mathbf{x}_k) \leq d_{\mathbf{L}}(\mathbf{x}_i,\mathbf{x}_j) \forall j \rightsquigarrow i$. Les auteurs proposent un critère plus sévère consistant à maintenir une marge entre les voisins cibles et les imposteurs.

Formellement, un imposteur d'un exemple $\mathbf{x}_i$ vis à vis du voisin cible $\mathbf{x}_j$, tous deux associés à un label $a_i$, est un exemple $\mathbf{x}_k$ tel que $a_k \neq a_i$ et

$$\|\mathbf{L}(\mathbf{x}_i-\mathbf{x}_k)\|^2 \leq \|\mathbf{L}(\mathbf{x}_i-\mathbf{x}_j)\|^2 + 1. \tag{5.6}$$

L'éloignement des imposteurs définit le deuxième terme de la fonction de coût :

$$\epsilon_{\text{push}}(\mathbf{L}) = \sum_{i,j\rightsquigarrow i} \sum_k (1-a_{ik}) g\left(1 + \|\mathbf{L}(\mathbf{x}_i-\mathbf{x}_j)\|^2 - \|\mathbf{L}(\mathbf{x}_i-\mathbf{x}_k)\|^2\right), \tag{5.7}$$

avec $g(\mathbf{x}) = \max(\mathbf{x}, 0)$ et $a_{i,k} = \begin{cases} 1 & \text{si } a_i = a_k, \\ 0 & \text{sinon} \end{cases}$.

Finalement, la fonction de coût du LMNN s'écrit :

$$\epsilon(\mathbf{L}) = (1-\nu)\epsilon_{\text{pull}}(\mathbf{L}) + \nu\epsilon_{\text{push}}(\mathbf{L}), \tag{5.8}$$

avec $\nu$ un paramètre de pondération.

Dans notre cas, l'objectif est d'augmenter le pouvoir discriminant des actions élémentaires vis-à-vis des activités. Pour cela, on applique un LMNN sur les descripteurs $\mathbf{F_s}$ des segments avec les étiquettes associées aux activités. On cherche ainsi à rapprocher les segments issus d'une même classe en vue de générer des étiquettes d'actions discriminantes vis-à-vis des activités.

### 5.1.3 Génération des annotations des flux vidéo

Rappelons que chaque flux vidéo a été segmenté. Chaque segment $v_s$ a ensuite été représenté par un descripteur $\mathbf{F}_s$ de taille fixe. Puis une métrique $\mathbf{L}$ a été appliquée pour obtenir un nouvel espace de représentation dans lequel les segments appartenant à une même activité ont été rapporchés et ceux appartenant à des activités différentes éloignés.

Appelons $\mathbf{F}'_s$ le nouveau descripteur associé à chacun de ces segments. Un algorithme des *k-moyennes* est ensuite appliqué sur ces vecteurs $\mathbf{F}'_s$ de manière à générer $k$ centres représentant les $k$ actions élémentaires recherchées. Chacun des segments $v_s$ est affecté à l'action élémentaire dont il est le plus proche. Tous les flux vidéos sont alors représentés comme une séquence temporelle de segments de taille variable étiquetés par l'action élémentaire qu'ils représentent.

Ces séquences étiquetées sont utilisées par le DOHT premier niveau pour apprendre à reconnaître automatiquement les actions élémentaires à partir d'un flux vidéo (Figure 5.1). Une fois les actions élémentaires reconnues, elle serviront à alimenter le DOHT $2^{nd}$ niveau pour reconnaître les activités.

## 5.2 DOHT hiérarchique

Les flux vidéo possèdent maintenant une double annotation : une en actions élémentaires et une en activités. Un premier algorithme DOHT est appris pour reconnaître les actions élémentaires. Il prend en entrée des descripteurs bas-niveaux, utilise les annotations d'actions élémentaires et apprend les poids $W_1$ liés aux votes pour reconnaître ces actions.

Le deuxième algorithme DOHT est appris pour reconnaître les activités. Il prend en entrée les actions élémentaires, utilise les annotations d'activités pour apprendre des poids $W_2$ qui serviront à leur reconnaissance.

Ces deux étapes, indépendantes sont illustrées Figure 5.1.

## 5.3 Résultats sur le jeu de données DAHLIA

Ce nouvel algorithme est évalué sur le jeu de données DAHLIA [3], présenté en section 4.3.2. Les squelettes utilisés sont normalisés de la même façon qu'avec le DOHT classique : ils sont exprimés dans un repère lié aux épaules et normalisés par la distance entre les épaules.

### 5.3.1 Découpe non-supervisée en segments

La première étape de l'apprentissage hiérarchique est la découpe non supervisée des flux vidéo en segments qui seront utilisés pour la définition des actions élémentaires. Rappelons que cette étape est réalisée uniquement en apprentissage pour définir quelles sont les actions élémentaires utiles pour décrire les activités. Elle ne sera pas réalisée lors de la détection puisqu'elle nécessite la connaissance des vidéos complètes pour filtrer les données ainsi que pour générer la segmentation. Cette dernière contrainte est incompatible avec une extraction *en ligne* des actions élémentaires.

Pour obtenir des segments temporels identiques quelle que soit la vue et pour gérer les nombreuses vues sans descripteurs (qui rendraient impossible la segmentation), nous fusionnons les squelettes des trois vues en amont de la segmentation. Pour cela, un nouveau squelette est estimé à partir des trois squelettes initiaux issus des kinects : pour chaque articulation, une nouvelle position est calculée comme la moyenne des coordonnées de cette articulation sur les trois vues, pondérée par l'indice de confiance donné par le capteur Kinect.

Pour cette découpe en segments, et parce que les jambes ne sont pas visibles sur deux des vues dans la majorité des cas, nous choisissons de conserver les articulations liées aux membres : *Tête*, *Epaules*, *Coudes*, *Poignets*, *Mains* et *Colonne Vertébrale*. Les descripteurs de ces articulations sont concaténés sur 16 instants temporels, comme pour le DOHT classique.

Cette segmentation non-supervisée est paramétrée par :

— le nombre de plus proches voisins à considérer pour l'estimation de la densité de probabilité utilisée lors du pré-traitement des données (section 5.1.1) ;

— le rayon de recherche généralisé $R$ utilisé pour construire la matrice creuse de similarité (section 5.1.1).

Pour le premier, nous fixons arbitrairement un nombre de plus proches voisins à $k = 1024$.

Intuitivement le rayon de recherche $R$ a une grande influence sur les découpes en segments et son choix est donc plus délicat. En effet, un rayon de recherche trop faible serait trop sévère et très peu de voisins seraient extraits pour la création de la matrice SSSM. Cette matrice serait alors trop éparse, et rendrait inefficace la segmentation. A l'inverse, un rayon de recherche trop grand engendrerait une matrice SSSM sans zones "vides" et la recherche des blocs d'actions (Figure 5.4) génèrerait des segments de durée très importante.

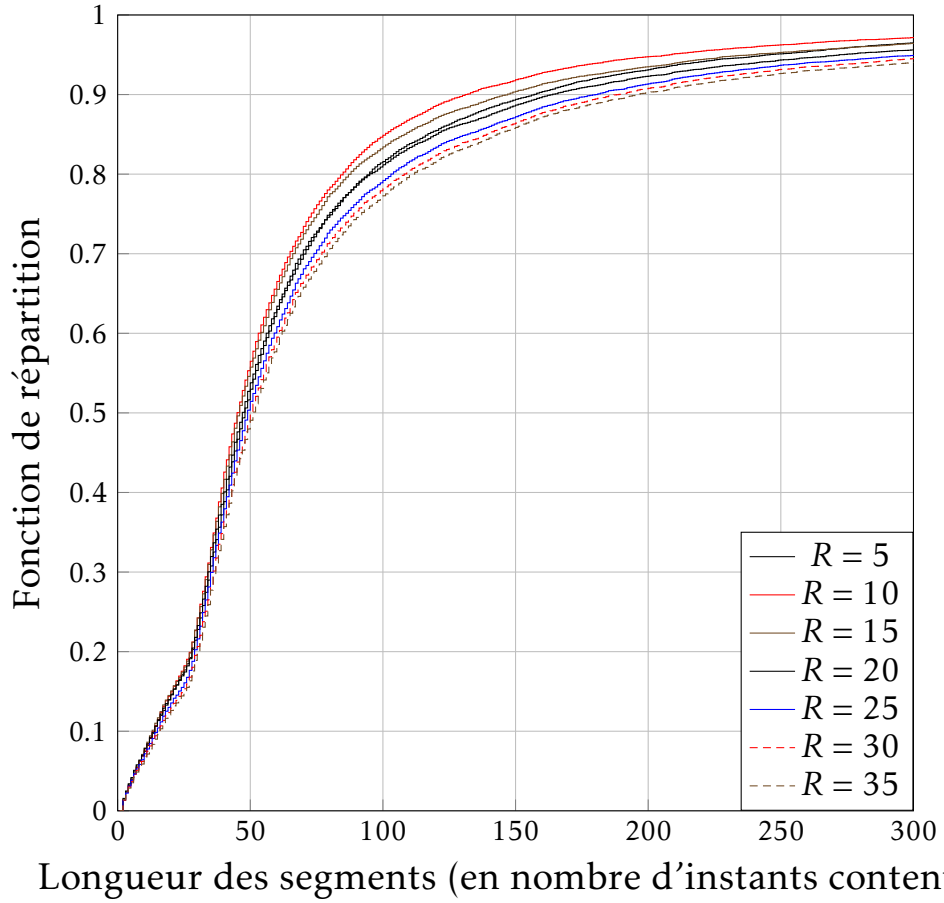



FIGURE 5.9 – Fonction de répartition empirique des longueurs des segments

Nous avons donc créé des découpes avec différentes valeurs de $R$ et conservé celles donnant les meilleurs résultats. Une intuition concernant la durée des actions élémentaires que l'on cherche à générer est qu'elles doivent être de l'ordre de quelques secondes : par exemple, pour l'activité manger, une action élémentaire serait "amener la main à la bouche puis la redescendre". Sachant que les vidéos ont été tournées à 15 images par seconde, on s'attend à des segments de longueurs entre 15 et 75 images. Afin d'analyser l'effet du rayon sur les découpes générées, la Figure 5.9 représente les fonctions de répartition empiriques pour 7 valeurs de $R$, toutes activités confondues. L'abscisse représente donc la longueur des segments en nombre d'images tandis que l'ordonnée représente le nombre de segments générés d'une taille inférieure à cette longueur dans toute la base.

$R$ a finalement assez peu d'influence sur le nombre de segments. Comme le nombre de segments de 1 à 3 secondes est plus important avec $R = 10$, nous conserverons cette valeur pour le reste des expériences.

La Figure 5.10 présente un histogramme des longueurs des segments obtenues avec $R = 10$ pour chaque action. On constate une certaine homogénéïté dans la répartition

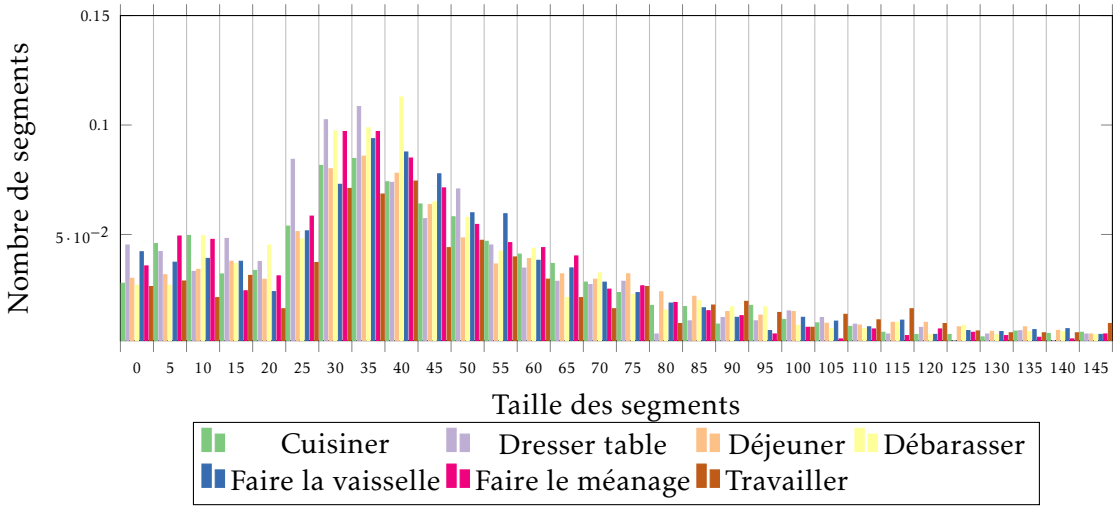



FIGURE 5.10 – Histogrammes des longueurs des segments pour différentes actions. Pour chaque action, l'histogramme a été normalisé par le nombre total de découpes générées

des tailles de segments au travers des activités pour cette valeur de rayon. D'autre part, la grande majorité des segments obtenus a bien une longueur comprise entre 1 et 3s (15 et 75 images). Des segments de petites tailles (inférieures à 20 instants) apparaissent quelques soient les activités. Ils sont dûs aux intervalles inter-actions comme représenté sur la figure 5.4 où, bien que seulement 2 actions soient présentes, 3 segments ont été obtenus.

### 5.3.2 La détection hiérarchique

Après l'étape précédente, nous avons obtenu une découpe de chaque vidéo en un ensemble de segments qui, après quantification, représente les actions élémentaires recherchées. Nous évaluons dans cette section la détection hiérarchique d'activités que nous proposons. Nous présentons les paramètres utilisés, les performances que nous obtenons sur le jeu de données DAHLIA et évaluons l'apport de l'étape d'apprentissage de métrique sur les taux de bonne reconnaissance.

La représentation des squelettes se fait à l'aide d'un VLAD, décrit en section 5.1.2. Pour générer ce VLAD, nous avons appliqué les quantifications à l'aide de $C = 4$ initialisations de l'algorithme des *k-moyennes* comportant chacun $k = 32$ centres. Ces deux paramètres n'ont pas été optimisés et une étude de leur influence devrait être menée.

Après l'étape d'apprentissage de métrique, une nouvelle étape de *k-moyennes* est mise en place afin de définir les actions élémentaires où $k$ représente le nombre souhaité d'actions élémentaires. Ce nombre est difficile à déterminer *a priori* et est très important puisqu'il influence directement la description des activités : utiliser un nombre trop faible amène à une description très pauvre pas forcément discriminante mais utiliser un nombre trop important amène à une description trop riche qui peut nuire à la

généralisation. Nous testons donc notre approche hiérarchique pour différentes valeurs de ce paramètre puis conservons celui donnant les meilleurs résultats (en pratique, nous avons testé pour des valeurs $k = 2^i$ avec $i \in \{4, 5, \ldots, 9\}$).

Il est délicat, voire impossible, d'évaluer les sous-étapes de la méthode hiérarchique que nous proposons. En effet, il est complexe d'estimer la qualité d'un ensemble d'actions élémentaires n'ayant pas de sémantique. Nous évaluons donc directement le résultat final de la détection d'activité.

L'approche que nous proposons se fait en deux étapes : la première retrouve les actions élémentaires et la seconde détermine les activités. En entraînement, nous avons mis en place une méthode permettant de déterminer les actions élémentaires, composée d'une segmentation non supervisée en segments puis d'un étiquetage des segments. Or, comme évoqué plus haut (section 5.3.1), cette méthode de ne peut pas être utilisée *en ligne* où les images sont traitées au fur et à mesure de leur extraction.

Le premier niveau de DOHT a donc un intérêt majeur. Il renvoie, à chaque instant, une étiquette d'action élémentaire à partir des squelettes extraits, générant une segmentation et annontation simultanées d'actions élémentaires. Cette étape est efficace en terme de temps de calcul et est temps réel, comme démontré lors de l'étude sur le jeu de données *TUM Kitchen dataset* présentée en section 3.3.3.

Le paramètre principal de ce premier DOHT est la taille maximale $2M$ des intervalles considérés lors du processus de vote (section 3.1). Rigoureusement, $2M$ devrait être fixé à la taille maximale des activités élémentaires à reconnaître et donc, des segments obtenus. En pratique, on utilise une durée couvrant 90% des segments. En reprenant la Figure 5.9, on choisit une durée maximale de 150 images, soit $M = 75$. Pour tous les autres paramètres, nous décidons de conserver les mêmes valeurs que dans les études précédentes menées sur la base de données TUM, à savoir $C = 4$ pour le SVM et un nombre d'intervalles $2n_I = 10$ déterminé expérimentalement pour le DOHT classique.

Le second niveau du DOHT détecte quant à lui les activités. En vue de comparer les résultats de la détection d'activités hiérarchique proposée ici avec celle de la détection à l'aide d'un simple DOHT, nous conservons les mêmes paramètres pour l'étape de détection d'activités (DOHT 2ème niveau). Notamment, nous conservons $M = 1000$ instants pour définir les intervalles considérés.

Les tests ont été effectués dans les configurations mono-vues et multi-vues. Pour cette dernière configuration, nous fusionnons les informations dans le deuxième niveau du DOHT (cf Figure 5.11). Prenant ainsi les actions élémentaires générées sur les trois vues indépendamment Cette fusion a été faite au niveau des cartes de présence, puisqu'il s'agit du niveau ayant présenté les meilleurs résultats sur le DOHT classique (section 3.3.2).

Comme évoqué plus haut, le nombre d'actions élémentaires à considérer est délicat à estimer *a priori*, nous avons évalué cette méthode pour différentes valeurs de ce paramètre. Les résultats obtenus en utilisant toutes les vues sont résumés Tableau 5.2.

Les meilleurs résultats sont obtenus pour 64 actions élémentaires selon la métrique $\mathcal{FA}_1$ et 512 selon les métrique F-Score et $IoU$. Dans la suite, les résultats obtenus considèrent 512 actions élémentaires.

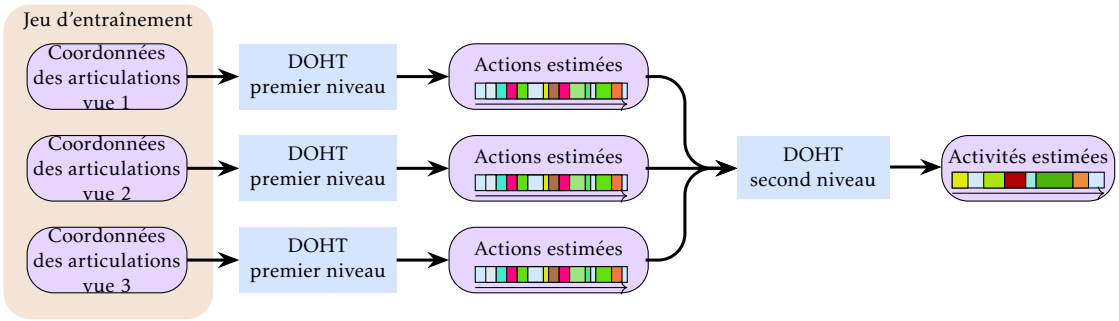



FIGURE 5.11 – Paradigme de fusion des vues en DOHT hierarchique

| | $\mathcal{FA}_1$ | F-Score | IoU |
|---|---|---|---|
| **DOHT Unique** | 0.77 | 0.75 | 0.60 |
| **16 actions** | 0.65 | 0.63 | 0.46 |
| **32 actions** | 0.70 | 0.67 | 0.51 |
| **64 actions** | **0.75** | 0.71 | **0.56** |
| **128 actions** | 0.74 | 0.71 | **0.56** |
| **256 actions** | 0.73 | 0.70 | 0.55 |
| **512 actions** | 0.74 | **0.72** | **0.56** |

TABLEAU 5.2 – Résultats du DOHT hiérarchique multi-vues en fonction du nombre d'actions élémentaires considérées

Le Tableau 5.3a présente les résultats obtenus avec le DOHT hiérarchique sur chacune des vues indépendamment et en multi-vues. De la même façon que pour le DOHT classique, l'utilisation conjointe de toutes les vues améliore les résultats.

Pour démontrer l'apport de l'étape d'apprentissage de métriques sur la reconnaissance d'activités et donc l'amélioration du caractère discriminant des actions par la méthode proposée, nous présentons dans le Tableau 5.3b les performances obtenues lorsque l'on génère les étiquettes d'actions dès la sortie de l'algorithme de segmentation. Pour cela, nous appliquons un algorithme des *k-moyennes* sur les segments générés par la première étape puis apprenons le DOHT hiérarchique avec ces nouvelles étiquettes d'ac-

| | $\mathcal{FA}_1$ | F-Score | IoU |
|---|---|---|---|
| **Vue 1** | 0.66 | 0.62 | 0.46 |
| **Vue 2** | 0.62 | 0.59 | 0.42 |
| **Vue 3** | 0.70 | 0.67 | 0.51 |
| **Multi-vues** | 0.75 | 0.71 | 0.56 |

(a) Avec étape d'apprentissage de métrique

| | $\mathcal{FA}_1$ | F-Score | IoU |
|---|---|---|---|
| **Vue 1** | 0.57 | 0.51 | 0.35 |
| **Vue 2** | 0.50 | 0.45 | 0.30 |
| **Vue 3** | 0.67 | 0.61 | 0.46 |
| **Multi-vues** | 0.74 | 0.70 | 0.55 |

(b) Sans apprentissage de métrique

TABLEAU 5.3 – Résultats mono et multi-vues de l'apprentissage de métrique

tions. On constate des résultats inférieurs à ceux obtenus avec notre méthode complète. Cela montre l'intérêt de l'étape d'apprentissage de métrique.

Pour analyser plus précisément ces résultats et constater l'augmentation du caractère discriminant des actions apportée par l'étape d'apprentissage de métriques, nous présentons sur la Figure 5.12 une comparaison des matrices de confusions avec et sans *metric learning*.

On constate une diminution de la confusion pour toutes les classes, sur toutes les vues. Cette amélioration est plus flagrante pour les classes *Débarasser* et *Faire le ménage*. En effet, sur les vues 1 et 2, on constate une augmentation de plus de 20 points sur la classe débarasser.

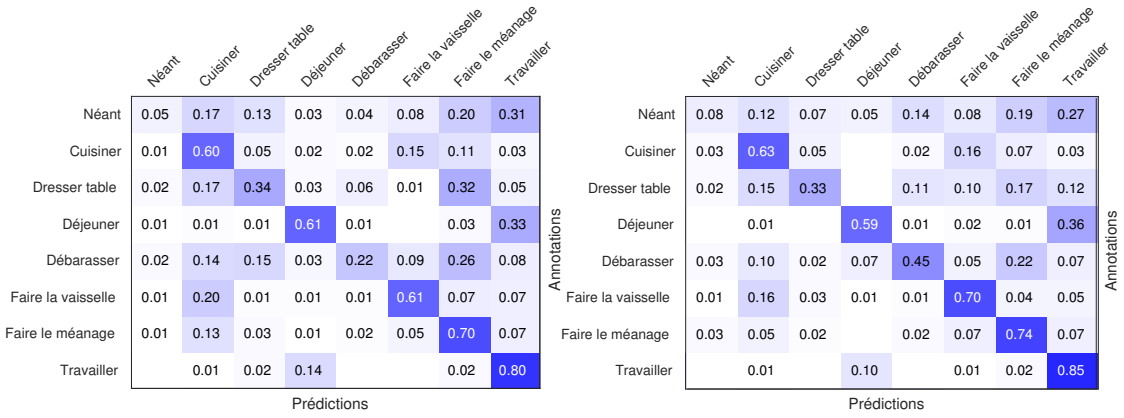
Plus spécifiquement, la Figure 5.12c présente une confusion importante pour la classe *"Débarasser"* avec les classes *"Cuisiner"*, *"Faire la vaisselle"* et *"Faire le ménage"*. Après apprentissage de métrique, pour les instants associés à la classe *"Débarasser"*, le taux de réponses (fausses détections) pour les classes *"Cuisiner"* et *"Faire la vaisselle"* chutent respectivement de 14 et 13 points (Figure 5.12d). Ces résultats semblent confirmer le regroupement, dans l'espace des descripteurs, des segments associés à la classe *"Débarasser"*. De la même façon, pour les instants associés à la classe *"Dresser la table"* on constate une diminution de 10 points sur la confusion avec la classe *"Cuisiner"*.

Les observations précédentes confirment l'apport de l'étape d'apprentissage de métrique pour la définition des actions élémentaires. Cependant, cette amélioriation n'est pas aussi nette pour toutes les activités, suggérant qu'un réglage plus fin des paramètres de cette étape d'apprentissage devrait être réalisé pour une amélioration généralisée des résultats.

La Figure 5.13 présente la matrice de confusion obtenue lorsque les différentes vues sont utilisées. On constate une amélioration des performances pour toutes les activités. Les deux classes pour lesquelles cette fusion améliore le plus les résultats sont les classes *"Débarasser"* et *"Dresser la table"*. Ce sont celles qui sont les plus affectées par les déplacements de la personne dans la pièce ainsi que par les occultations provoquées par l'ouverture d'une porte de placard, par exemple. La fusion des vues permet donc de compenser une occultation intervenant sur une des vues par les informations enregistrées par les autres.

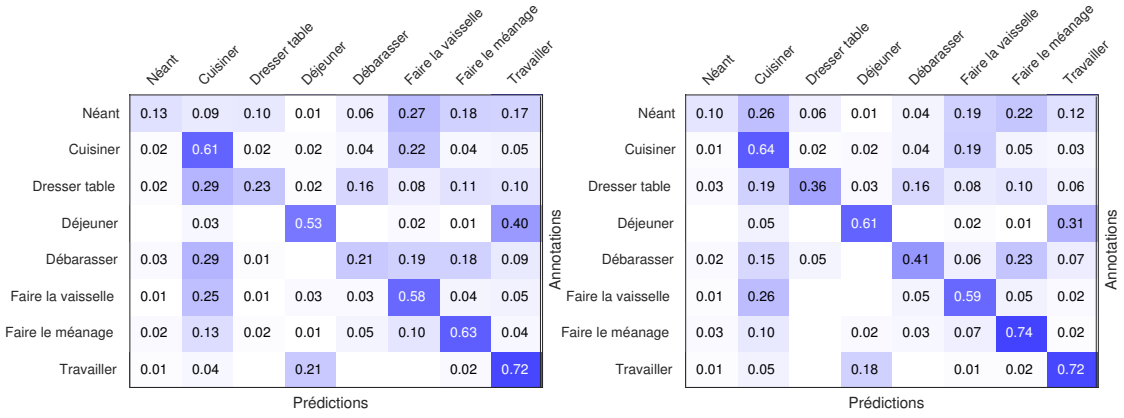
Cette méthode hiérarchique est compatible avec une approche temps réel puisque, après quantification, le traitement de chaque instant dans le cas multi-vues prend 8 ms sur un "Intel(R) Xeon(R) CPU E5-2687W 0 @ 3.10GHz" en utilisant 16 coeurs pour la génération de la carte de présence et sans optimisation poussée. Ces temps de calcul sont similaires (voire légèrement inférieurs) à ceux du DOHT classique.

Enfin, le dernier test consiste à comparer le DOHT hiérarchique mis en place avec un DOHT unique (Tableau 5.4). Ces résultats sont assez décevants puisque seule la vue 1 améliore légèrement les résultats, alors que les autres résultats sont très similaires à ceux obtenus précédemment.



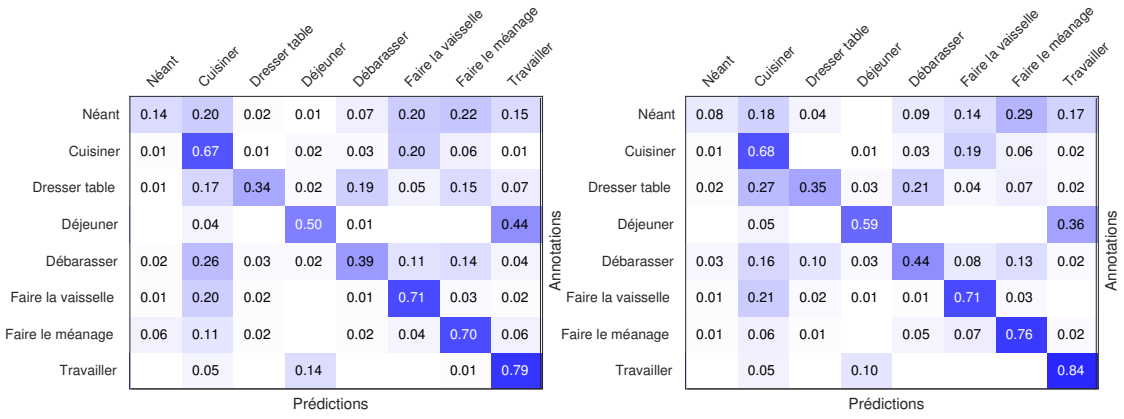

(a) Vue 1 sans apprentissage de métrique

(b) Vue 1




(c) Vue 2 sans apprentissage de métrique

(d) Vue 2




(e) Vue 3 sans apprentissage de métrique

(f) Vue 3

FIGURE 5.12 – Matrices de confusion obtenues avec le DOHT hiérarchique, avec et sans étape d'apprentissage de métrique


FIGURE 5.13 – Matrice de confusion après fusion des vues dans le DOHT hiérarchique

| | DOHT initial | DOHT hiérarchique |
|---|---|---|
| **Vue 1** | 0.60 | **0.66** |
| **Vue** | **0.63** | 0.62 |
| **Vue** | **0.73** | 0.70 |
| **Multi-vues** | **0.77** | 0.75 |

TABLEAU 5.4 – Comparaison du DOHT hiérarchique et du DOHT initial

## 5.4 Perspectives

Même si ces premiers résultats ne sont pas ceux escomptés, ils laissent supposer différentes pistes à explorer.

Tout d'abord, une optimisation plus poussée des paramètres devrait être faite pour permettre d'améliorer le DOHT hiérarchique. En effet, bien des paramètres peuvent influencer le résultat comme il a été mentionné dans ce chapitre :

— le nombre de plus proches voisins lors du pré-traitement des données, section 5.1.1 ;

— le rayon de recherche généralisé $R$ utilisé pour la segmentation, section 5.1.1 ;

— le nombre de centres et le nombre d'initialisations utilisés lors du VLAD, section 5.1.2 ;

— le nombre d'actions élémentaires à définir ;

— le paramètre de pondération $\nu$ utilisé lors de la recherche de métrique ;

— les paramètres des deux DOHT, à savoir, la taille maximale de la fenêtre $M$ et le coefficient $C$ utilisé lors de l'apprentissage des SVM.

Ainsi, si tous les paramètres ont été optimisés sur le DOHT classique, il n'en est pas de même pour le DOHT hiérarchique par manque de temps, ce qui ne peut qu'améliorer les résultats.

On pourrait aussi imaginer une méthode définissant un nombre d'actions élémentaires possibles différent en fonction de l'activité (paramètre de l'algorithme des *k-moyennes* après apprentissage de métrique). Cela permettrait une découpe non homogène de l'espace des descripteurs sans augmenter de façon trop conséquente le nombre d'actions à définir.

Une autre piste consiste à garder plusieurs actions en sortie du premier DOHT. En effet, l'entrée du second DOHT ne nécessite pas obligatoirement d'avoir imposé l'action élémentaire reconnue à chaque instant. On pourrait ainsi garder les probabilités de toutes les actions élémentaires en entrée du second DOHT. Cela augmenterait la quantité d'information et pourrait lever certaines ambiguïtés.

## 5.5 Conclusion

Nous avons décrit dans ce chapitre une méthode de détection hiérarchique d'activités. Elle consiste en une extraction à deux niveaux. Un premier DOHT prédit la présence d'actions élémentaires qu'un second DOHT prend en entrée pour la détection d'activités à haut niveau sémantique.

Pour la phase d'entraînement du premier DOHT générant les actions élémentaires, on définit de façon semi-supervisée des actions élémentaires de tailles variables. Cette génération se fait par une segmentation non-supervisée des vidéos puis par un apprentissage de métriques supervisé par les étiquettes d'activités. Enfin, un algorithme de clustering non supervisé (*k-moyennes*) associe une étiquette à chaque segment.

L'aspect hiérarchique de cette méthode permet notamment un apprentissage plus fin du comportement humain, sans augmenter les ressources mémoires nécessaire. En effet, le premier niveau dédié aux actions considère des intervalles à court terme en vue d'une détection d'actions alors que le second considère l'évolution à plus long terme pour la détection d'activités.

Nous avons présenté une étude préliminaire des performances d'une telle méthode. Les résultats observés sont proches de ceux du DOHT initial sans optimisation des paramètres. Ces premiers résultats nous rendent cependant confiants quant aux bénéfices de l'approche hiérarchique qui, intuitivement, semble pertinente. En effet, l'idée est de dissocier une première recherche d'actions élémentaires où la temporalité à court terme est importante d'une recherche d'activités mettant plus en avant une temporalité à long terme entre actions élémentaires.

Plusieurs pistes ont été proposées pour améliorer les résultats.

Chapitre 6

# Conclusion et perspectives

## Synthèse

Les travaux menés dans cette thèse font partie d'un projet de grande ampleur mené au CEA portant sur le développement d'habitations intelligentes. Dans ce cadre nous nous sommes plus particulièrement intéressés à la détection d'activités humaines. Ces travaux considèrent et adaptent un algorithme de détection ayant fait ses preuves pour la détection temporelle d'actions au sein de vidéos : le *DOHT (Deeply Optimized Hough Transform)*, un algorithme de transformée de Hough fortement optimisé.

La première contribution de cette thèse consiste en une fusion d'informations à trois niveaux différents au sein de cet algorithme DOHT. Cette fusion a un double intérêt : celui de combiner des informations en provenance de plusieurs caméras et la prise en considération de différentes modalités afin de détecter de façon robuste des actions ou des activités humaines au sein d'une vidéo. On considère dans cette thèse simultanément des positions d'articulations (*squelettes*) ainsi que des informations visuelles agrégées temporellement (*trajectoires de points*, *histogrammes de gradient et de flux optique*), le tout en provenance de plusieurs vues différentes. Une étude est alors réalisée pour comparer les différents niveaux de fusion d'information et les différentes combinaisons de modalités. Les résultats montrent qu'il est plus performant de fusionner les informations au plus bas niveau possible de l'algorithme. D'autre part, combiner des modalités de nature différentes (forme et trajectoire par exemple) permet d'améliorer les scores de détection. Il en est de même pour la fusion d'informations en provenance de plusieurs capteurs qui permet de s'affranchir des problèmes d'occultation.

Une étude a également été menée concernant la robustesse de la méthode face à une indisponibilité de capteurs ayant servi lors de l'apprentissage. Elle montre que le deuxième niveau de fusion proposé reste robuste face à cette perte d'informations, ce qui est un atout pour le développement d'une application réelle. Dans un dernier temps, nous nous sommes intéressés aux temps de calcul et à la latence de l'algorithme. Là encore, les résultats obtenus sont encourageant pour le déploiement de la méthode proposée.

L'algorithme a été testé sur des bases de la littérature qui ne reflètent pas des conditions réelles d'utilisation dans le cadre d'habitat intelligent. Face à ce manque de données réalise pour notre application, un nouvel ensemble de données adapté à la détection d'activités a été créé. Cette base, appelée *DAHLIA* (*DAily Home LIfe Actitivy*), a été pensée pour des applications orientées vers la détection temporelle d'activités à haut niveau sémantique. Elle se différencie des jeux de la littérature par la longueur de ses vidéos, le type d'activités considérées ainsi que l'environnement et les comportements réalistes qui la composent.

Cette base a été acquise dans des conditions de vie réelles, dans l'appartement instrumenté MobileMii mis en place au CEA. Ainsi, 44 personnes ont été filmées pendant qu'elles réalisaient 7 activités autour de la période du déjeuner pour une durée moyenne de 39 minutes par séquence. Comme assez peu de consignes étaient données aux participants, la variabilité des vidéos est très forte et représente bien la diversité rencontrée dans des conditions naturelles (par exemple, les durées extrêmes sont de 24 et 64 minutes). Afin de proposer cette base à la communauté de chercheurs, nous avons également introduit des métriques de performances et des premiers résultats sur des algorithmes de la littérature qui serviront de point de comparaison.

L'algorithme DOHT a été conçu pour détecter des actions de courte durée où la temporalité des gestes est très importante, ce qui n'est plus forcément le cas des activités. C'est pourquoi nous avons proposé une approche hiérarchique de détection d'activités. Elle décompose le problème de la détection d'activités en deux niveaux. Le premier segmente et classifie des actions élémentaires de façon semi-supervisée puis le second utilise ces actions pour détecter les activités en elle-même. Ces derniers travaux, effectués en toute fin de thèse, montrent des performances encourageantes et méritent une analyse plus approfondie des paramètres pour confirmer l'apport de cette méthode.

Les travaux réalisés dans cette thèse se sont concrétisés par une implémentation dans l'appartement intelligent MobileMii. On peut ainsi voir la détection d'activités en temps réel à partir de capteurs Kinectv2 disposés autour de la scène.

## Perspectives

Ces travaux ouvrent plusieurs perspectives dont la première se situe au niveau des descripteurs. Nos expériences ont montré que l'exploitation des positions 3D des squelettes des personnes améliore les performances de détection d'activités. Ces squelettes ont été acquis à l'aide de cartes de profondeur imposant l'utilisation de capteurs adaptés. Une perspective à court terme serait alors celle de l'utilisation de squelettes extraits à partir d'images en 2 dimensions, par deep learning par exemple. La perte de l'information de profondeur devrait alors être compensée par un apprentissage adapté.

Concernant le paradigme de détection en lui-même, une perspective immédiate est l'enrichissement de la méthode hiérarchique proposée qui montre des résultats encourageant sans étude poussée des paramètres. En plus d'une étude approfondie de l'influence des nombreux paramètres de la méthode, une attention pourrait être portée à la segmentation en actions élémentaires. En effet dans les travaux que nous proposons,

la segmentation se fait de façon non supervisée et ne prend donc pas en compte les spécificités inhérentes à chacune des activités. Or nous constatons que les paramètres de l'algorithme de segmentation influent de façon hétérogène sur les tailles des segments générés selon l'activité considérée. Cela suggère une évolution de l'algorithme de segmentation vers un paradigme semi-supervisé (au lieu de non-supervisé) afin d'obtenir des segments, et donc des actions élémentaires, plus discriminants vis-à-vis des activités haut niveau.

Une autre perspective est l'exploration des différentes stratégies de décision au sein de l'architecture de DOHT hiérarchique. Dans la version présentée dans cette thèse, le DOHT détectant les actions élémentaires sélectionne celles recevant le score maximum à partir des primitives extraites. Cela permet ensuite de considérer directement ces actions comme des mots en entrée du DOHT de niveau supérieur. On pourrait envisager un choix moins strict de l'action en sortie du premier niveau, à l'image des méthodes de codage doux. Cela permettrait d'augmenter la quantité d'informations prise en compte pour la détection d'activités par le DOHT de niveau supérieur. Il paraît alors naturel de considérer une autre perspective plus importante consistant en un apprentissage simultané des poids des deux étages du DOHT. Cela peut par exemple se faire par une formulation équivalente du DOHT sous la forme d'un réseau de neurones et par application des méthodes correspondantes.

Ensuite, la généralisation de cette méthode à des problèmes de détections d'activités multiples au sein d'une même vidéo semble pertinente. En effet, on ne considère dans cette thèse que des actions réalisées par une seule personne dans chaque vidéo. La méthode de fusion d'informations proposée est toutefois compatible avec la détection simultanée de plusieurs activités dans une vidéo, mais aussi avec la détection d'interaction de ces activités. On pourrait par exemple incorporer des informations issues d'un suivi de personne afin d'associer à chaque piste une activité ou d'en déduire une activité commune. Le paradigme de Hough est d'ailleurs directement compatible avec une détection spatiale de l'activité. Cela impose alors d'être en possession de données contenant diverses activités simultanées, impliquant ou non des interactions entre les personnes ; à l'image du jeu de données **LIRIS** [158] pour la détection d'actions.

Par ailleurs, on peut envisager l'intégration d'une compréhension plus fine de l'environnement par la prise en compte de descripteurs modélisant le contexte comme les objets impliqués dans les actions. L'amélioration constatée lorsque l'on ajoute les descripteurs *HOG* modélisant l'apparence spatiale laisse supposer qu'un tel ajout d'informations augmenterait de façon significative les performances de l'algorithme de détection.

A un niveau supérieur et plus proche de l'application, on pourrait également considérer des éléments de contexte plus haut globaux intégrant les habitudes des personnes en termes d'horaires et de succession temporelles des activités.

Enfin, les méthodes travaillant à partir de réseaux de neurones profonds ont prouvé leur efficacité dans la majorité des domaines de l'apprentissage statistique. Jusque récemment, ces méthodes était difficilement applicables à des problèmes de reconnaissance et de détection d'actions ou d'activités par manque de jeu de données suffisamment impor-

tants en terme de nombre d'exemples. Aujourd'hui, avec la mise à disposition de jeux de données de tailles conséquentes et le développement de méthodes d'apprentissage par transfert (*transfer learning*), ce paradigme devient une perspective majeure dans l'élaboration de méthodes de détections d'activités humaines.

# Bibliographie


[1] G. Vaquette, C. Achard et L. Lucat, « Robust information fusion in the doht paradigm for real-time action detection », *Journal of Real-Time Image Processing*, p. 1–14, 2016, issn : 18618200. doi : 10.1007/s11554-016-0660-5. adresse : https://www.engineeringvillage.com/share/document.url?mid=cpx%7B%5C_%7DM67b0df341591c903b5bM776f10178163171%7B%5C&%7Ddatabase=cpx.

[2] ——, « Information fusion for action recognition with deeply optimised hough transform paradigm », in *11th International Conference on Computer Vision and Applications (VISAPP)*, 2016.

[3] G. Vaquette, A. Orcesi, L. Lucat et C. Achard, « The daily home life activity dataset : a high semantic activity dataset for online recognition », *Proceedings - 12th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2017 - 1st International Workshop on Adaptive Shot Learning for Gesture Understanding and Production, ASL4GUP 2017, Biometrics in the Wild, Bwild 2017, Heteroge*, p. 497–504, 2017. doi : 10.1109/FG.2017.67.

[4] T. M. Mitchell, *Machine learning*. McGraw-Hill Boston, MA : 1997.

[5] F. Keller, « Introduction to machine learning connectionist and statistical language processing a sample data set learning rules learning rules », *Machine Learning*, 1997.

[6] I. Goodfellow, Y. Bengio et A. Courville, *Deep Learning*. MIT Press, 2016. adresse : qq.

[7] X. Peng, L. Wang, X. Wang et Y. Qiao, « Bag of visual words and fusion methods for action recognition : comprehensive study and good practice », *Computer Vision and Image Understanding*, t. 150, p. 109–125, 2015, issn : 1090235X. doi : 10.1016/j.cviu.2016.03.013. arXiv : 1405.4506. adresse : http://dx.doi.org/10.1016/j.cviu.2016.03.013.

[8] M. Selmi, « Reconnaissance d'activités humaines à partir de séquences vidéo », thèse de doct., Institut National des Télécommunications, 2014.

[9] J. Yamato, J. Ohya et K. Ishii, *Recognizing human action in time-sequential images using hidden markov model*, 1992. doi : 10.1109/CVPR.1992.223161. adresse : http://ieeexplore.ieee.org/xpls/abs%7B%5C_%7Dall.jsp?arnumber=223161.

[10] A. F. Bobick et J. W. Davis, « The recognition of human movement using temporal templates », *IEEE Transactions on pattern analysis and machine intelligence*, t. 23, n° 3, p. 257–267, 2001.

[11] D. Weinland, R. Ronfard et E. Boyer, « Free viewpoint action recognition using motion history volumes », *Computer Vision and Image Understanding*, t. 104, n° 2-3 SPEC. ISS. P. 249–257, 2006, issn : 10773142. doi : 10.1016/j.cviu.2006.07.013.

[12] A. A. Chaaraoui, P. Climent-Pérez et F. Flórez-Revuelta, « Silhouette-based human action recognition using sequences of key poses », *Pattern Recognition Letters*, t. 34, n° 15, p. 1799–1807, 2013.

[13] I. Laptev et T. Lindeberg, « Space-time interest points », in *9th International Conference on Computer Vision, Nice, France*, IEEE conference proceedings, 2003, p. 432–439.

[14] P. Dollár, V. Rabaud, G. Cottrell et S. Belongie, « Behavior recognition via sparse spatio-temporal features », *Proceedings - 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, VS-PETS*, t. 2005, p. 65–72, 2005. doi : 10.1109/VSPETS.2005.1570899.

[15] G. Willems, T. Tuytelaars et L. Van Gool, « An efficient dense and scale-invariant spatio-temporal interest point detector », *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, t. 5303 LNCS, n° PART 2, p. 650–663, 2008, issn : 03029743. doi : 10.1007/978-3-540-88688-4-48.

[16] H. Wang, A. Kläser, C. Schmid et C.-L. Liu, « Action recognition by dense trajectories », in *IEEE Conference on Computer Vision & Pattern Recognition*, Colorado Springs, United States, 2011, p. 3169–3176. adresse : http://hal.inria.fr/inria-00583818/en.

[17] C. Harris et M. Stephens, « A combined corner and edge detector », in *Procedings of the Alvey Vision Conference 1988*, Citeseer, t. 15, 1988, p. 147–151. doi : 10.5244/C.2.23. adresse : http://www.bmva.org/bmvc/1988/avc-88-023.html.

[18] D. G. Lowe, « Object recognition from local scale-invariant features », *Proceedings of the Seventh IEEE International Conference on Computer Vision*, t. 2, n° [8, p. 1150–1157, 1999, issn : 0-7695-0164-8. doi : 10.1109/ICCV.1999.790410. arXiv : 0112017 [cs]. adresse : http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=790410.

[19] D. Hall, V. C. de Verdière et J. L. Crowley, « Object recognition using coloured receptive fields », in *European conference on computer vision*, Springer, 2000, p. 164–177.

[20] K. Mikolajczyk, C. Schmid, J. Ralyté, R. Deneckère et C. Rolland, « An affine invariant interest point detector », *Computer Vision—ECCV 2002*, p. 128–142, 2002. doi : 10.1007/3-540. adresse : https://hal.inria.fr/inria-00548252.

[21] T. Tuytelaars et L. Van Gool, « Wide baseline stereo matching based on local, affinely invariant regions », *British Machine Vision Conference*, p. 412–425, 2000. doi : 10.5244/C.14.38. adresse : http://www.bmva.org/bmvc/2000/papers/p42.html.

[22] S. Smith, « Real-time motion segmentation and shape tracking, 1995 », in *Proc. 5th Int. Conf. on Computer Vision*.

[23] I. Laptev, « On space-time interest points », *International Journal of Computer Vision*, t. 64, n° 2-3, p. 107–123, 2005.

[24] L. Fei-Fei et P. Perona, « A bayesian hierarchical model for learning natural scene categories », *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, t. 2, p. 524–531, 2005, issn : 10636919. doi : 10.1109/CVPR.2005.16. adresse : http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1467486.

[25] E. Nowak, F. Jurie et B. Triggs, « Sampling strategies for bag-of-features image classification », in *Computer Vision–ECCV 2006*, Springer, 2006, p. 490–503.

[26] H. Wang, M. M. Ullah, A. Käser, I. Laptev et C. Schmid, « Evaluation of local spatio-temporal features for action recognition », *20th British Machine Vision Conference*, 2009.

[27] M. Garrigues et A. Manzanera, « Real time semi-dense point tracking », *Image Analysis and Recognition*, p. 245–252, 2012, issn : 03029743. doi : 10.1007/978-3-642-31295-3_29.

[28] B. D. Lucas, T. Kanade et al., « An iterative image registration technique with an application to stereo vision », 1981.

[29] H. Wang, A. Kläser, C. Schmid et C.-L. Liu, « Dense trajectories and motion boundary descriptors for action recognition », *Int J Comput Vis*, t. 103, p. 60–79, 2013. doi : 10.1007/s11263-012-0594-8. adresse : https://link.springer.com/content/pdf/10.1007%7B%5C%%7D2Fs11263-012-0594-8.pdf.

[30] J. Shi et al., « Good features to track », in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, IEEE, 1994, p. 593–600.

[31] H. Wang et C. Schmid, « Action recognition with improved trajectories », in *Computer Vision (ICCV), 2013 IEEE International Conference on*, IEEE, 2013, p. 3551–3558.

[32] N. Dalal et B. Triggs, « Histograms of oriented gradients for human detection », in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, IEEE, t. 1, 2005, p. 886–893, isbn : 0769523722. doi : 10.1109/CVPR.2005.177. adresse : http://scholar.google.com/scholar?hl=en%7B%5C&%7DbtnG=Search%7B%5C&%7Dq=intitle:Histograms+of+oriented+gradients+for+human+detection%7B%5C#%7D0.

[33] I. Laptev, M. Marszałek, C. Schmid et B. Rozenfeld, « Learning realistic human actions from movies », in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, IEEE, 2008, p. 1–8.

[34] A. Klaser, M. Marszalek et C. Schmid, « A spatio-temporal descriptor based on 3d-gradients », *Proceedings of the British Machine Conference*, p. 99.1–99.10, 2008. doi : 10.5244/C.22.99. adresse : http://eprints.pascal-network.org/archive/00005291/.

[35] N. Dalal, B. Triggs, C. Schmid, N. Dalal, B. Triggs, C. Schmid, H. Detection et U. Oriented, « Human detection using oriented histograms of flow and appearance to cite this version : human detection using oriented histograms of flow and appearance », p. 428–441, 2010.

[36] W. Li, Z. Zhang et Z. Liu, « Action recognition based on a bag of 3d points », in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, IEEE, 2010, p. 9–14.

[37] L. Xia et J. K. Aggarwal, « Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera », in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, IEEE, 2013, p. 2834–2841.

[38] J. Wang, Z. Liu, J. Chorowski, Z. Chen et Y. Wu, « Robust 3d action recognition with random occupancy patterns », *Computer Vision {\textendash} {ECCV} 2012*, p. 872–885, 2012. doi : 10.1007/978-3-642-33709-3_62. adresse : https://doi.org/10.1007%7B%5C%%7D2F978-3-642-33709-3%7B%5C_%7D62.

[39] J. Wang, Z. Liu, Y. Wu et J. Yuan, « Mining actionlet ensemble for action recognition with depth cameras », *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, p. 1290–1297, 2012.

[40] A. W. Vieira, E. R. Nascimento, G. L. Oliveira, Z. Liu et M. F. M. Campos, « Stop : space-time occupancy patterns for 3d action recognition from depth map sequences », in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, Springer, 2012, p. 252–259.

[41] O. Oreifej et Z. Liu, « Hon4d : histogram of oriented 4d normals for activity recognition from depth sequences », in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2013, p. 716–723, isbn : 978-0-7695-4989-7. doi : 10.1109/CVPR.2013.98. arXiv : 1604.01753.

[42] Y. Song, J. Tang, F. Liu, S. Yan et S. Member, « Body surface context : a new robust feature for action recognition from depth videos », *IEEE Transactions on Circuits and Systems for Video Technology*, t. 24, n° 6, p. 952–964, 2014.

[43] X. Yang, C. Zhang et Y. Tian, « Recognizing actions using depth motion maps-based histograms of oriented gradients », *Proceedings of the 20th ACM international conference on Multimedia - MM '12*, n° c, p. 1057, 2012. doi : 10.1145/2393347.2396382. adresse : http://dl.acm.org/citation.cfm?doid=2393347.2396382.

[44] C. Chen, K. Liu et N. Kehtarnavaz, « Real-time human action recognition based on depth motion maps », *Journal of Real-Time Image Processing*, t. 12, n° 1, p. 155–163, 2016, issn : 18618200. doi : 10.1007/s11554-013-0370-1.

[45] J.-F. Hu, W.-S. Zheng, J. Lai et J. Zhang, « Jointly learning heterogeneous features for rgb-d activity recognition », in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, p. 5344–5352.

[46] B. Ni, Y. Pei, P. Moulin et S. Yan, « Multilevel depth and image fusion for human activity detection », *Cybernetics, IEEE Transactions on*, t. 43, n° 5, p. 1383–1394, 2013.

[47] Y. Song, S. Liu et J. Tang, « Describing trajectory of surface patch for human action recognition on rgb and depth videos », *Signal Processing Letters, IEEE*, t. 22, n° 4, p. 426–429, 2015.

[48] T. B. Moeslund, A. Hilton et V. Kr ? ?ger, « A survey of advances in vision-based human motion capture and analysis », *Computer Vision and Image Understanding*, t. 104, n° 2-3 SPEC. ISS. P. 90–126, 2006, issn : 10773142. doi : 10.1016/j.cviu.2006.08.002.

[49] L. Lo Presti et M. La Cascia, « 3d skeleton-based human action classification : a survey », *Pattern Recognition*, t. 53, p. 130–147, 2016, issn : 00313203. doi : 10.1016/j.patcog.2015.11.019. arXiv : arXiv:1212.0402. adresse : http://dx.doi.org/10.1016/j.patcog.2015.11.019.

[50] R. Girshick, J. Shotton, P. Kohli, A. Criminisi et A. Fitzgibbon, « Efficient regression of general-activity human poses from depth images », *Proceedings of the IEEE International Conference on Computer Vision*, p. 415–422, 2011, issn : 1550-5499. doi : 10.1109/ICCV.2011.6126270.

[51] V. Belagiannis et A. Zisserman, « Recurrent human pose estimation », in *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, IEEE, 2017, p. 468–475.

[52] A. Haque, B. Peng, Z. Luo, A. Alahi, S. Yeung et L. Fei-Fei, « Towards viewpoint invariant 3d human pose estimation », in *European Conference on Computer Vision*, Springer, 2016, p. 160–177.

[53] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook et R. Moore, « Real-time human pose recognition in parts from single depth images », *Communications of the ACM*, t. 56, n° 1, p. 116–124, 2013, issn : 0001-0782. doi : 10.1145/2398356.2398381.

[54] C. Wang, Y. Wang et A. L. Yuille, « An approach to pose-based action recognition », *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 915–922, 2013, issn : 10636919. doi : 10.1109/CVPR.2013.123.

[55] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy et R. Vidal, « Bio-inspired dynamic 3d discriminative skeletal features for human action recognition », *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, p. 471–478, 2013, issn : 21607508. doi : 10.1109/CVPRW.2013.153.

[56] D. Wu et L. Shao, « Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition », *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 724–731, 2014, issn : 10636919. doi : 10.1109/CVPR.2014.98.

[57] M. Raptis, D. Kirovski et H. Hoppe, « Real-time classification of dance gestures from skeleton animation », in *Proceedings of the 2011 ACM SIGGRAPH/Eurographics symposium on computer animation*, ACM, 2011, p. 147–156.

[58] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal et R. Bajcsy, « Sequence of the most informative joints (smij) : a new representation for human skeletal action recognition », *Journal of Visual Communication and Image Representation*, t. 25, n° 1, p. 24–38, 2014, issn : 10473203. doi : 10.1016/j.jvcir.2013.04.007. adresse : http://dx.doi.org/10.1016/j.jvcir.2013.04.007.

[59] X. Yang et Y. Tian, « Effective 3d action recognition using eigenjoints », *Journal of Visual Communication and Image Representation*, t. 25, n° 1, p. 2–11, 2014, issn : 10473203. doi : 10.1016/j.jvcir.2013.03.001. adresse : http://dx.doi.org/10.1016/j.jvcir.2013.03.001.

[60] D. Carbonera Luvizon, H. Tabia et D. Picard, « Learning features combination for human action recognition from skeleton sequences », *Pattern Recognition Letters*, 2016, issn : 01678655. doi : 10.1016/j.patrec.2017.02.001.

[61] B. B. Amor, J. Su et A. Srivastava, « Action recognition using rate-invariant analysis of skeletal shape trajectories », *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, t. 38, n° 1, p. 1–13, 2016.

[62] D. G. Kendall, « Shape manifolds, procrustean metrics, and complex projective spaces », *Bulletin of the London Mathematical Society*, t. 16, n° 2, p. 81–121, 1984, issn : 14692120. doi : 10.1112/blms/16.2.81.

[63] A. Chan-Hon-Tong, C. Achard et L. Lucat, « Deeply optimized hough transform : application to action segmentation », in *Image Analysis and Processing–ICIAP 2013*, Springer, 2013, p. 51–60.

[64] J. Macqueen, « Some methods for classification and analysis of multivariate observations », *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, t. 1, n° 233, p. 281–297, 1967, issn : 00970433. doi : citeulike-article-id:6083430.

[65] C. Bishop et N. Nasrabadi, « Pattern recognition and machine learning », *Pattern Recognition*, t. 4, n° 4, p. 738, 2006, issn : 10179909. doi : 10.1117/1.2819119. arXiv : 0-387-31073-8. adresse : http://www.library.wisc.edu/selectedtocs/bg0137.pdf.

[66] L. Kaufman et P. J. Rousseeuw, *Finding groups in data : an introduction to cluster analysis*. John Wiley & Sons, 2009, t. 344.

[67] S. C. Johnson, « Hierarchical clustering schemes* », t. 32, 1967. adresse : http://hbanaszak.mjr.uw.edu.pl/TempTxt/Johnson%7B%5C_%7D1967%7B%5C_%7DHierarchicalClusteringSchemes.pdf.

[68] A. Y. Ng, M. I. Jordan et Y. Weiss, « On spectral clustering : analysis and an algorithm », *Advances in Neural Information Processing Systems 14*, p. 849–856, 2002, issn : <null>. doi : 10.1.1.19.8100. adresse : http://papers.nips.cc/paper/2092-on-spectral-clustering-analysis-and-an-algorithm.pdf.

[69] U. Von Luxburg, « A tutorial on spectral clustering », *Statistics and computing*, t. 17, n° 4, p. 395–416, 2007. arXiv : arXiv:0711.0189v1.

[70] J. C. Van Gemert, J.-M. Geusebroek, C. J. Veenman et A. W. M. Smeulders, « Kernel codebooks for scene categorization », in *European conference on computer vision*, Springer, 2008, p. 696–709.

[71] J. Yang, K. Yu, Y. Gong et T. Huang, « Linear spatial pyramid matching using sparse coding for image classification », in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, 2009, p. 1794–1801.

[72] H. Jégou, M. Douze, C. Schmid et P. Pérez, « Aggregating local descriptors into a compact representation », *IEEE Conference on Computer Vision & Pattern Recognition*, p. 3304–3311, 2010. doi : 10.1109/CVPR.2010.5540039>. adresse : http://lear.inrialpes.fr/pubs/2010/JDSP10/jegou%7B%5C_%7Dcompactimagerepresentation%7B%5C_%7Dslides.pdf.

[73] F. Perronnin et C. Dance, « Fisher kernels on visual vocabularies for image categorization », in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, IEEE, 2007, p. 1–8.

[74] M. Blank, L. Gorelick, E. Shechtman, M. Irani et R. Basri, « Actions as space-time shapes », in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, IEEE, t. 2, IEEE, 2005, p. 1395–1402.

[75] D. Batra, T. Chen et R. Sukthankar, « Space-time shapelets for action recognition », *2008 IEEE Workshop on Motion and Video Computing, WMVC*, p. 1–6, 2008. doi : 10.1109/WMVC.2008.4544051.

[76] J. Sullivan et S. Carlsson, « Recognizing and tracking human action », *ECCV '02 : Proceedings of the 7th European Conference on Computer Vision-Part I*, p. 629–644, 2002.

[77] Z. Lin, Z. Jiang et L. S. Davis, « Recognizing actions by shape-motion prototype trees », in *2009 IEEE 12th international conference on computer vision*, IEEE, 2009, p. 444–451.

[78] D. Weinland et E. Boyer, « Action recognition using exemplar-based embedding », in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, IEEE, 2008, p. 1–7.

[79] A. Mokhber, C. Achard et M. Milgram, « Recognition of human behavior by space-time silhouette characterization », *Pattern Recognition Letters*, t. 29, n° 1, p. 81–89, 2008.

[80] A. Chan-hon-tong, « Segmentation supervisée d'actions à partir de primitives haut niveau dans des flux vidéos », thèse de doct., Université Pierre et Marie Curie-Paris VI, 2014.

[81] A. M. J. Sarkar, Y.-K. Lee et S. Lee, « A smoothed naive bayes-based classifier for activity recognition », *IETE Technical Review*, t. 27, n° 2, p. 107–119, 2010. doi : 10.4103/02564602.10876586.

[82] D. J. C. MacKay et L. C. B. Peto, « A hierarchical dirichlet language model », *Natural language engineering*, t. 1, n° 3, p. 289–308, 1995.

[83] F. Jelinek, « Interpolated estimation of markov source parameters from sparse data », in *Proc. Workshop on Pattern Recognition in Practice, 1980*, 1980.

[84] C. Schuldt, I. Laptev, B. Caputo, L. Barbara et S.-. Stockholm, « Recognizing human actions : a local svm approach », in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, IEEE, t. 3, 2004, p. 32–36, isbn : 0769521282. doi : 10.1109/ICPR.2004.1334462. arXiv : 1505.04868.

[85] H. Jhuang, T. Serre, L. Wolf et T. Poggio, « A biologically inspired system for action recognition », *Ieee Iccv*, t. 11, p. 1–8, 2007, issn : 1550-5499. doi : 10.1109/ICCV.2007.4408988. adresse : papers2://publication/uuid/5AD5F124-2606-43A2-B3BD-61E6FE94AE88.

[86] J. C. Niebles, H. Wang et L. Fei-Fei, « Unsupervised learning of human action categories using spatial-temporal words », *International Journal of Computer Vision*, t. 79, n° 3, p. 299–318, 2008, issn : 09205691. doi : 10.1007/s11263-007-0122-4.

[87] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt et B. Scholkopf, « Support vector machines », *IEEE Intelligent Systems*, t. 13, p. 18–28, 1998, issn : 1094-7167. doi : 10.1109/5254.708428. arXiv : arXiv:1011.1669v3.

[88] B. E. Boser, I. M. Guyon et V. N. Vapnik, « A training algorithm for optimal margin classifiers », *Proceedings of the fifth annual workshop on Computational learning theory - COLT '92*, p. 144–152, 1992, issn : 0-89791-497-X. doi : 10.1145/130385.130401. arXiv : arXiv:1011.1669v3. adresse : http://portal.acm.org/citation.cfm?doid=130385.130401.

[89] C. Cortes et V. Vapnik, « Support-vector networks », *Machine Learning*, t. 20, n° 3, p. 273–297, 1995, issn : 15730565. doi : 10.1023/A:1022627411411. arXiv : arXiv:1011.1669v3.

[90] J. Platt et al., « Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods », *Advances in large margin classifiers*, t. 10, n° 3, p. 61–74, 1999.

[91] C.-C. Chang et C.-J. Lin, « Libsvm : a library for support vector machines chih-chung », *ACM Transactions on Intelligent Systems and Technology*, t. 2, n° 3, p. 1–27, 2011, issn : 21576904. doi : 10.1145/1961189.1961199. arXiv : 0-387-31073-8. adresse : http://dl.acm.org/citation.cfm?doid=1961189.1961199.

[92] F. Rosenblatt, « The perceptron : a probabilistic model for information storage and organization in ... », *Psychological review*, t. 65, n° 6, p. 386–408, 1958, issn : 1939-1471(Electronic) ;0033-295X(Print). doi : 10.1037/h0042519. arXiv : arXiv : 1112.6209. adresse : http://psycnet.apa.org/journals/rev/65/6/386/%7B%5C%%7D5Cnhttp://psycnet.apa.org/journals/rev/65/6/386/%7B%5C%%7D5Cnhttp://www2.fiit.stuba.sk/%7B~%7Dcernans/nn/nn%7B%5C_%7Dtexts/neuronove%7B%5C_%7Dsiete%7B%5C_%7Dpriesvitky%7B%5C_%7D02%7B%5C_%7DQ.pdf%7B%5C%%7D5Cnhttp://psycnet.apa.org/journals/rev/65/6/386.pdf%7B%5C%%7D5Cnpapers://c53d1644-c.

[93] A. Blum et J. Dunagan, « Smoothed analysis of the perceptron algorithm for linear programming », in *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, Society for Industrial et Applied Mathematics, 2002, p. 905–914, isbn : 0-89871-513-X. adresse : http://dl.acm.org/citation.cfm?id=545381.545499.

[94] C. Achard, X. Qu, A. Mokhber et M. Milgram, « Action recognition with semi-global characteristics and hidden markov models », in *Advanced Concepts for Intelligent Vision Systems*, Springer, 2007, p. 274–284.

[95] F. Lv et R. Nevatia, « Recognition and segmentation of 3-d human action using hmm and multi-class adaboost », *Computer Vision–ECCV 2006*, p. 359–372, 2006.

[96] S. Fine, Y. Singer et N. Tishby, « The hierarchical hidden markov model : analysis and applications », *Machine learning*, t. 32, n° 1, p. 41–62, 1998.

[97] H. H. Bui, D. Q. Phung et S. Venkatesh, « Hierarchical hidden markov models with general state hierarchy », in *Proceedings of the national conference on artificial intelligence*, Menlo Park, CA ; Cambridge, MA ; London ; AAAI Press ; MIT Press ; 1999, 2004, p. 324–329.

[98] N. T. Nguyen, D. Q. Phung, S. Venkatesh et H. Bui, « Learning and detecting activities from movement trajectories using the hierarchical hidden markov model », in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, IEEE, t. 2, 2005, p. 955–960.

[99] C. Wu, J. Zhang, O. Sener, B. Selman, S. Savarese et A. Saxena, « Watch-n-patch : unsupervised learning of actions and relations », in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, p. 4362–4370. adresse : https://arxiv.org/pdf/1603.03541.pdf.

[100] P. Dai, H. Di, L. Dong, L. Tao et G. Xu, « Group interaction analysis in dynamic context », *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, t. 38, n° 1, p. 275–282, 2008, issn : 1941-0492. doi : 10.1109/TSMCB.2008.2009559. adresse : http://www.ncbi.nlm.nih.gov/pubmed/19150758.

[101] D. Damen et D. Hogg, « Recognizing linked events : searching the space of feasible explanations », in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, 2009, p. 927–934.

[102] A. McCallum, D. Freitag et F. C. N. Pereira, « Maximum entropy markov models for information extraction and segmentation. », in *Icml*, t. 17, 2000, p. 591–598.

[103] J. Lafferty, A. McCallum et F. C. N. Pereira, « Conditional random fields : probabilistic models for segmenting and labeling sequence data », 2001.

[104] C. Sminchisescu, A. Kanaujia et D. Metaxas, « Conditional models for contextual human motion recognition », *Computer Vision and Image Understanding*, t. 104, n° 2, p. 210–220, 2006.

[105] M. Á. Mendoza et N. P. De La Blanca, « Applying space state models in human action recognition : a comparative study », in *International Conference on Articulated Motion and Deformable Objects*, Springer, 2008, p. 53–62.

[106] S. B. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian et T. Darrell, « Hidden conditional random fields for gesture recognition », in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, IEEE, t. 2, 2006, p. 1521–1527.

[107] A. Quattoni, S. Wang, L.-P. Morency, M. Collins et T. Darrell, « Hidden conditional random fields », *IEEE transactions on pattern analysis and machine intelligence*, t. 29, n° 10, 2007.

[108] Y. Wang et G. Mori, « Max-margin hidden conditional random fields for human action recognition », in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, 2009, p. 872–879.

[109] J. Zhang et S. Gong, « Action categorization with modified hidden conditional random field », *Pattern Recognition*, t. 43, n° 1, p. 197–203, 2010.

[110] L. Chen, N. der Aa, R. T. Tan et R. C. Veltkamp, « Hidden conditional random fields for action recognition », in *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, IEEE, t. 2, 2014, p. 240–247, isbn : 978-0-7695-5125-8. doi : 10.1109/CSA.2013.85.

[111] A.-A. Liu, W.-Z. Nie, Y.-T. Su, L. Ma, T. Hao et Z.-X. Yang, « Coupled hidden conditional random fields for rgb-d human action recognition », *Signal Processing*, t. 112, p. 74–82, 2015, issn : 01651684. doi : 10.1016/j.sigpro.2014.08.038.

[112] M. Selmi et M. A. El-Yacoubi, « Multimodal sequential modeling and recognition of human activities », p. 541–548, 2016. doi : 10.1007/978-3-319-41267-2_76. adresse : http://link.springer.com/10.1007/978-3-319-41267-2%7B%5C_%7D76.

[113] L. Han, X. Wu, W. Liang, G. Hou et Y. Jia, « Discriminative human action recognition in the learned hierarchical manifold space », *Image and Vision Computing*, t. 28, n° 5, p. 836–849, 2010.

[114] L. Wang et 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, p. 1–8.

[115] H. Kjellström, J. Romero et D. Kragić, « Visual object-action recognition : inferring object affordances from human demonstration », *Computer Vision and Image Understanding*, t. 115, n° 1, p. 81–90, 2011.

[116] P. V. C. Hough, *Method and means for recognizing complex patterns*, 1962.

[117] R. O. Duda et P. E. Hart, « Use of the hough transformation to detect lines and curves in pictures », *Communications of the ACM*, t. 15, n° 1, p. 11–15, 1972.

[118] D. H. Ballard, « Generalizing the hough transform to detect arbitrary shapes », *Pattern Recognition*, t. 13, n° 2, p. 111–122, 1981, issn : 00313203. doi : 10.1016/0031-3203(81)90009-1.

[119] A. Yao, J. Gall et L. Van Gool, « A hough transform-based voting framework for action recognition », in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, IEEE, 2010, p. 2061–2068.

[120] B. Leibe, A. Leonardis et B. Schiele, « Combined object categorization and segmentation with an implicit shape model », in *Workshop on Statistical Learning in Computer Vision*, 2004.

[121] S. Maji et J. Malik, « Object detection using a max-margin hough transform », in *Internationnal Conference on Computer Vision and Pattern Recognition*, 2009.

[122] Y. Zhang et T. Chen, « Implicit shape kernel for discriminative learning of the hough transform detector », in *Conference of British Machine Vision Conference*, 2010.

[123] P. Wohlhart, S. Schulter, M. Kostinger, P. Roth et H. Bischof, « Discriminative hough forests for object detection », in *Conference of British Machine Vision Conference*, 2012.

[124] A. Chan-Hon-Tong, C. Achard et L. Lucat, « Simultaneous segmentation and classification of human actions in video streams using deeply optimized hough transform », *Pattern Recognition*, t. 47, n° 12, p. 3807–3818, 2014.

[125] N. Dalal, B. Triggs et C. Schmid, « Human detection using oriented histograms of flow and appearance », in *Computer Vision–ECCV 2006*, Springer, 2006, p. 428–441.

[126] M. Tenorth, J. Bandouch et M. Beetz, « The tum kitchen data set of everyday manipulation activities for motion tracking and action recognition », in *International Conference on Computer Vision Workshops*, 2009.

[127] A. Yao, J. Gall, G. Fanelli et L. Van Gool, « Does human action recognition benefit from pose estimation ? », in *Conference of British Machine Vision Conference*, 2011.

[128] L. Gorelick, M. Blank, E. Shechtman, M. Irani et R. Basri, « Actions as space-time shapes », *IEEE transactions on pattern analysis and machine intelligence*, t. 29, n° 12, p. 2247–2253, 2007.

[129] R. Messing, C. Pal et H. Kautz, « Activity recognition using the velocity histories of tracked keypoints », in *Computer Vision, 2009 IEEE 12th International Conference on*, IEEE, 2009, p. 104–111.

[130] B. W. Hwang, S. Kim et S. W. Lee, « A full-body gesture database for automatic gesture recognition », *FGR 2006 : Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*, t. 2006, p. 243–248, 2006. doi : 10.1109/FGR.2006.8.

[131] Y. Song, D. Demirdjian et R. Davis, « Tracking body and hands for gesture recognition : natops aircraft handling signals database », *2011 IEEE International Conference on Automatic Face and Gesture Recognition and Workshops, FG 2011*, p. 500–506, 2011. doi : 10.1109/FG.2011.5771448.

[132] Y. Ma, H. M. Paterson et F. E. Pollick, « A motion capture library for the study of identity, gender, and emotion perception from biological motion », *Behavior Research Methods*, t. 38, n° 1, p. 134–141, 2006, issn : 1554-351X. doi : 10.3758/BF03192758. adresse : http://www.springerlink.com/index/10.3758/BF03192758.

[133] T. Mori, M. Shimosaka et K. Tsujioka, *Ics action database*, 2003. adresse : http://www.ics.t.u-tokyo.ac.jp/action/.

[134] M. Rohrbach, M. Regneri, M. Andriluka, S. Amin, M. Pinkal et B. Schiele, « Script data for attribute-based recognition of composite activities », in *European Conference on Computer Vision*, Springer, 2012, p. 144–157.

[135] M. Rohrbach, A. Rohrbach, M. Regneri, S. Amin, M. Andriluka, M. Pinkal et B. Schiele, « Recognizing fine-grained and composite activities using hand-centric features and script data », *International Journal of Computer Vision*, t. 119, n° 3, p. 346–373, 2016, issn : 15731405. doi : 10.1007/s11263-015-0851-8. arXiv : 1502.06648.

[136] M. Bregonzio, S. Gong et 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, p. 1948–1955.

[137] M. D. Rodriguez, J. Ahmed et 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, p. 1–8.

[138] L. Wang et C. Leckie, « Encoding actions via quantized vocabulary of averaged silhouettes », in *Pattern Recognition (ICPR), 2010 20th International Conference on*, IEEE, 2010, p. 3657–3660.

[139] M. Marszalek, I. Laptev et C. Schmid, « Actions in context », in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, 2009, p. 2929–2936.

[140] J. Liu, J. Luo et M. Shah, « Recognizing realistic actions from videos ıin the wildȷ », in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, 2009, p. 1996–2003.

[141] J. Liu, Y. Yang et M. Shah, « Learning semantic visual vocabularies using diffusion distance », in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, 2009, p. 461–468.

[142] K. K. Reddy et M. Shah, « Recognizing 50 human action categories of web videos », *Machine Vision and Applications*, t. 24, nº 5, p. 971–981, 2013, issn : 09328092. doi : 10.1007/s00138-012-0450-4.

[143] H. Kuehne et T. Serre, « A large video database for human motion recognition », *Proceedings of the International Conference on Computer Vision (ICCV)*, p. 2556–2563, 2011, issn : 1550-5499. doi : 10.1109/ICCV.2011.6126543.

[144] F. C. Heilbron, V. Escorcia, B. Ghanem et J. C. Niebles, « Activitynet : a large-scale video benchmark for human activity understanding », *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, t. 07-12-June, p. 961–970, 2015, issn : 10636919. doi : 10.1109/CVPR.2015.7298698.

[145] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar et L. Fei-Fei, « Large-scale video classification with convolutional neural networks », *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, p. 1725–1732, 2014, issn : 978-1-4799-5118-5. doi : 10.1109/CVPR.2014.223. arXiv : 1412.0767. adresse : http://www-cs.stanford.edu/groups/vision/pdf/karpathy14.pdf%7B%5C%%7D5Cnhttp://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6909619%7B%5C%%7D0Apapers3://publication/doi/10.1109/CVPR.2014.223.

[146] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger et A. Weber, « Documentation mocap database hdm05 », *Database*, p. 34, 2007, issn : 16108892. doi : 10.1016/0260-4779(90)90032-9. adresse : http://cg.cs.uni-bonn.de/en/publications/paper-details/cg-2007-2/.

[147] J. Zhang, W. Li, P. O. Ogunbona, P. Wang et C. Tang, « Rgb-d-based action recognition datasets : a survey », *Pattern Recognition*, t. 60, p. 86–105, 2016.

[148] Z. Cheng, L. Qin, Y. Ye, Q. Huang et Q. Tian, « Human daily action analysis with multi-view and color-depth data », in *European Conference on Computer Vision*, Springer, 2012, p. 52–61.

[149] Z. Zhang, W. Liu, V. Metsis et V. Athitsos, « A viewpoint-independent statistical method for fall detection », in *Pattern Recognition (ICPR), 2012 21st International Conference on*, IEEE, 2012, p. 3626–3630.

[150] B. Ni, G. Wang et P. Moulin, « Rgbd-hudaact : a color-depth video database for human daily activity recognition », in *Consumer Depth Cameras for Computer Vision*, Springer, 2013, p. 193–208.

[151] J. Sung, C. Ponce, B. Selman et A. Saxena, « Human activity detection from rgbd images. », *Plan, activity, and intent recognition*, t. 64, 2011.

[152] H. S. Koppula, R. Gupta et A. Saxena, « Learning human activities and object affordances from rgb-d videos », *The International Journal of Robotics Research*, t. 32, n° 8, p. 951–970, 2013.

[153] P. Wei, N. Zheng, Y. Zhao et S.-C. Zhu, « Concurrent action detection with structural prediction », in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, p. 3136–3143.

[154] S. M. Amiri, M. T. Pourazad, P. Nasiopoulos et V. C. M. Leung, « Non-intrusive human activity monitoring in a smart home environment », in *E-Health Networking, Applications & Services (Healthcom), 2013 IEEE 15th International Conference on*, IEEE, 2013, p. 606–610.

[155] P. Wei, Y. Zhao, N. Zheng et S.-C. Zhu, « Modeling 4d human-object interactions for event and object recognition », in *2013 IEEE International Conference on Computer Vision*, IEEE, 2013, p. 3272–3279.

[156] C. van Gemeren, R. T. Tan, R. Poppe et R. C. Veltkamp, « Dyadic interaction detection from pose and flow », *Human Behavior Understanding*, t. 8749, p. 101–115, 2014, issn : 16113349. doi : 10.1007/978-3-319-11839-0_9.

[157] C. Van Gemeren, R. Poppe et R. C. Veltkamp, « Spatio-temporal detection of fine-grained dyadic human interactions », in *International Workshop on Human Behavior Understanding*, Springer, 2016, p. 116–133. adresse : http://link.springer.com/chapter/10.1007/978-3-319-46843-3%7B%5C_%7D8.

[158] C. Wolf, E. Lombardi, J. Mille, O. Celiktutan, M. Jiu, E. Dogan, G. Eren, M. Baccouche, E. Dellandréa, C.-E. Bichot et al., « Evaluation of video activity localizations integrating quality and quantity measurements », *Computer Vision and Image Understanding*, t. 127, p. 14–30, 2014, issn : 1090235X. doi : 10.1016/j.cviu.2014.06.014.

[159] N. Xu, A. Liu, W. Nie, Y. Wong, F. Li et Y. Su, « Multi-modal & multi-view & interactive benchmark dataset for human action recognition », *Proceedings of the 23rd ACM international conference on Multimedia*, p. 1195–1198, 2015. doi : 10.1145/2733373.2806315.

[160] H.˜Jhuang, J.˜Gall, S.˜Zuffi, C.˜Schmid et M.˜J.˜Black, « Towards understanding action recognition », *Proc. IEEE International Conference on Computer Vision (ICCV)*, p. 3192–3199, 2013.

[161] A. Shahroudy, J. Liu, T.-T. Ng et G. Wang, « Ntu rgb+ d : a large scale dataset for 3d human activity analysis », *ArXiv preprint arXiv :1604.02808*, 2016.

[162] M. Rohrbach, S. Amin, M. Andriluka et B. Schiele, « A database for fine grained activity detection of cooking activities », in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2012, p. 1194–1201, isbn : 9781467312264. doi : 10.1109/CVPR.2012.6247801. adresse : https://www.mpi-inf.mpg.de/fileadmin/inf/d2/amin/rohrbach12cvpr.pdf.

[163] G. Vaquette, A. Orcesi, L. Lucat et C. Achard, « The daily home life activity dataset : a high semantic activity dataset for online recognition », in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, IEEE, mai 2017, p. 497–504, isbn : 978-1-5090-4023-0. doi : 10.1109/FG.2017.67. adresse : http://ieeexplore.ieee.org/document/7961782/.

[164] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, M. Boonstra et V. Korzhova, « Performance evaluation protocol for text, face, hands, person and vehicle detection & tracking in video analysis and content extraction (vace-ii) », *Protocol Document*, 2005.

[165] J. Munkres, « Algorithms for the assignment and transportation problems », *Journal of the society for industrial and applied mathematics*, t. 5, n° 1, p. 32–38, 1957.

[166] R. Collins, X. Zhou et S. K. Teh, « An open source tracking testbed and evaluation web site », in *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, t. 35, 2005.

[167] J. A. Ward, P. Lukowicz et G. Tröster, « Evaluating performance in continuous context recognition using event-driven error characterisation », in *International Symposium on Location-and Context-Awareness*, Springer, 2006, p. 239–255.

[168] D. P. Young et J. M. Ferryman, « Pets metrics : on-line performance evaluation service », in *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, 2005, p. 317–324.

[169] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn et A. Zisserman, « The pascal visual object classes challenge : a retrospective », *International Journal of Computer Vision*, t. 111, n° 1, p. 98–136, 2015.

[170] M. Meshry, M. E. Hussein et M. Torki, « Linear-time online action detection from 3d skeletal data using bags of gesturelets », in *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, IEEE, 2016, p. 1–9.

[171] C.-Y. Chen et K. Grauman, « Efficient activity detection with max-subgraph search », in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2012, p. 1274–1281.

[172] J. Bentley, « Programming pearls : algortihm design techniques », *Communications of the ACM*, t. 27, n° 9, 1984.

[173] S. Nowozin et J. Shotton, « Action points : a representation for low-latency online human action recognition », *Microsoft Research Cambridge, Tech. Rep. MSR-TR-2012-68*, 2012.

[174] M. Zanfir, M. Leordeanu et C. Sminchisescu, « The moving pose : an efficient 3d kinematics descriptor for low-latency action recognition and detection », in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, p. 2752–2759.

[175] B. Krüger, A. Vögele, T. Willig, A. Yao, R. Klein et A. Weber, « Efficient unsupervised temporal segmentation of motion data », *IEEE Transactions on Multimedia*, t. 19, n° 4, p. 797–812, 2017, issn : 15209210. doi : 10.1109/TMM.2016.2635030. arXiv : 1510.06595.

[176] G. Yu, Z. Liu et J. Yuan, « Discriminative orderlet mining for real-time recognition of human-object interaction », in *Asian Conference on Computer Vision*, Springer, 2014, p. 50–65.

[177] K. Q. Weinberger et L. K. Saul, « Distance metric learning for large margin nearest neighbor classification », *The Journal of Machine Learning Research*, t. 10, p. 207–244, 2009, issn : 1532-4435. doi : 10.1126/science.277.5323.215. arXiv : 1407.4979.

[178] K. Q. Weinberger, J. Blitzer et L. K. Saul, « Distance metric learning for large margin nearest neighbor classification », 2006.

[179] K. Q. Weinberger et L. K. Saul, « Fast solvers and efficient implementations for distance metric learning », in *Proceedings of the 25th international conference on Machine learning*, ACM, 2008, p. 1160–1167.

[180] J. Goldberger, G. E. Hinton, S. T. Roweis et R. R. Salakhutdinov, « Neighbourhood components analysis », in *Advances in neural information processing systems*, 2005, p. 513–520.

[181] S. Chopra, R. Hadsell et Y. LeCun, « Learning a similarity metric discriminatively, with application to face verification », in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, IEEE, t. 1, 2005, p. 539–546.

# Table des matières

**Reconnaissance robuste d'activités humaines par vision**


**Résumé**

Cette thèse porte sur la segmentation supervisée d'un flux vidéo en fragments correspondant à des activités de la vie quotidienne. En différenciant geste, action et activité, cette thèse s'intéresse aux activités à haut niveau sémantique telles que "Cuisiner" ou "Prendre son repas" par opposition à des actions comme "Découper un aliment".
Pour cela, elle s'appuie sur l'algorithme DOHT (Deeply Optimized Hough Transform), une méthode de l'état de l'art utilisant un paradigme de vote (par transformée de Hough). Dans un premier temps, nous adaptons l'algorithme DOHT pour fusionner les informations en provenance de différents capteurs à trois niveaux différents de l'algorithme. Nous analysons l'effet de ces trois niveaux de fusion et montrons son efficacité par une évaluation sur une base de données composée d'actions de la vie quotidienne. Ensuite, une étude des jeux de données existant est menée. Constatant le manque de vidéos adaptées à la segmentation et classification (détection) d'activités à haut niveau sémantique, une nouvelle base de données est proposée. Enregistrée dans un environnement réaliste et dans des conditions au plus proche de l'application finale, elle contient des vidéos longues et non découpées adaptées à un contexte de détection. Dans un dernier temps, nous proposons une approche hiérarchique à partir d'algorithmes DOHT pour reconnaître les activités à haut niveau sémantique. Cette approche à deux niveau décompose le problème en une détection non-supervisée d'actions pour ensuite détecter les activités désirées.



**Mots clés :** détection d'activités, fusion d'informations, transformée de hough fortement optimisée (doht), jeu de données


---


**Abstract**

This thesis focuses on supervised activity segmentation from video streams within application context of smart homes. Three semantic levels are defined, namely gesture, action and activity, this thesis focuses mainly on the latter. Based on the Deeply Optimized Hough Transform paridigm, three fusion levels are introduced in order to benefit from various modalities. A review of existing action based datasets is presented and the lack of activity detection oriented database is noticed. Then, a new dataset is introduced. It is composed of unsegmented long time range daily activities and has been recorded in a realistic environment. Finaly, a hierarchical activity detection method is proposed aiming to detect high level activities from unsupervised action detection.



**Keywords:** activity detection, information fusion, deeply optimized hough transform (doht), database


---


**Laboratoire Vision et Ingénierie des Contenus**
CEA Saclay – Centre d'intégration Nano-INNOV – Avenue de la Vauve – Bâtiment 861 – 91120 Palaiseau – France