Université de Montréal

Vidéosurveillance intelligente pour la détection de chutes chez les personnes âgées

par
Caroline Rougier

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

Thèse présentée à la Faculté des arts et des sciences
en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.)
en informatique

mars, 2010

Université de Montréal
Faculté des arts et des sciences

Cette thèse intitulée:

Vidéosurveillance intelligente pour la détection de chutes chez les personnes âgées

présentée par:

Caroline Rougier

a été évaluée par un jury composé des personnes suivantes:

Neil Stewart,	président-rapporteur
Jean Meunier,	directeur de recherche
Sébastien Roy,	membre du jury
Richard Lepage,	examinateur externe
Neil Stewart,	représentant du doyen de la FES

Thèse acceptée le: 18 mai 2010

RÉSUMÉ

Les pays industrialisés comme le Canada doivent faire face au vieillissement de leur population. En particulier, la majorité des personnes âgées, vivant à domicile et souvent seules, font face à des situations à risques telles que des chutes. Dans ce contexte, la vidéosurveillance est une solution innovante qui peut leur permettre de vivre normalement dans un environnement sécurisé.

L'idée serait de placer un réseau de caméras dans l'appartement de la personne pour détecter automatiquement une chute. En cas de problème, un message pourrait être envoyé suivant l'urgence aux secours ou à la famille via une connexion internet sécurisée. Pour un système bas coût, nous avons limité le nombre de caméras à une seule par pièce ce qui nous a poussé à explorer les méthodes monoculaires de détection de chutes.

Nous avons d'abord exploré le problème d'un point de vue 2D (image) en nous intéressant aux changements importants de la silhouette de la personne lors d'une chute. Les données d'activités normales d'une personne âgée ont été modélisées par un mélange de gaussiennes nous permettant de détecter tout événement anormal. Notre méthode a été validée à l'aide d'une vidéothèque de chutes simulées et d'activités normales réalistes.

Cependant, une information 3D telle que la localisation de la personne par rapport à son environnement peut être très intéressante pour un système d'analyse de comportement. Bien qu'il soit préférable d'utiliser un système multi-caméras pour obtenir une information 3D, nous avons prouvé qu'avec une seule caméra calibrée, il était possible de localiser une personne dans son environnement grâce à sa tête. Concrêtement, la tête de la personne, modélisée par une ellipsoide, est suivie dans la séquence d'images à l'aide d'un filtre à particules. La précision de la localisation 3D de la tête a été évaluée avec une bibliothèque de séquence vidéos contenant les vraies localisations 3D obtenues par un système de capture de mouvement (Motion Capture). Un exemple d'application utilisant la trajectoire 3D de la tête est proposée dans le cadre de la détection de chutes.

En conclusion, un système de vidéosurveillance pour la détection de chutes avec une seule caméra par pièce est parfaitement envisageable. Pour réduire au maximum les risques de fausses alarmes, une méthode hybride combinant des informations 2D et 3D

pourrait être envisagée.

Mots clés: vision par ordinateur, vidéo surveillance, détection de chutes, détection de mouvement, suivi d'une cible, analyse de forme, localisation 3D.

ABSTRACT

Developed countries like Canada have to adapt to a growing population of seniors. A majority of seniors reside in private homes and most of them live alone, which can be dangerous in case of a fall, particularly if the person cannot call for help. Video surveillance is a new and promising solution for healthcare systems to ensure the safety of elderly people at home.

Concretely, a camera network would be placed in the apartment of the person in order to automatically detect a fall. When a fall is detected, a message would be sent to the emergency center or to the family through a secure Internet connection. For a low cost system, we must limit the number of cameras to only one per room, which leads us to explore monocular methods for fall detection.

We first studied 2D information (images) by analyzing the shape deformation during a fall. Normal activities of an elderly person were used to train a Gaussian Mixture Model (GMM) to detect any abnormal event. Our method was tested with a realistic video data set of simulated falls and normal activities.

However, 3D information like the spatial localization of a person in a room can be very useful for action recognition. Although a multi-camera system is usually preferable to acquire 3D information, we have demonstrated that, with only one calibrated camera, it is possible to localize a person in his/her environment using the person's head. Concretely, the head, modeled by a 3D ellipsoid, was tracked in the video sequence using particle filters. The precision of the 3D head localization was evaluated with a video data set containing the real 3D head localizations obtained with a Motion Capture system. An application example using the 3D head trajectory for fall detection is also proposed.

In conclusion, we have confirmed that a video surveillance system for fall detection with only one camera per room is feasible. To reduce the risk of false alarms, a hybrid method combining 2D and 3D information could be considered.

Keywords: computer vision, videosurveillance, fall detection, motion detection, tracking, shape analysis, 3D localization.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES ANNEXES

LISTE DES SIGLES

2D	Deux dimensions
3D	Trois dimensions
ACP	Analyse en composante principale
CCD	Charged Coupled Device
CCTV	Closed Circuit Television
CMOS	Complementary Metal Oxide Semi-conductor
CSS	Curvature Scale Space
DTW	Dynamic Time Warping
GMM	mélange de gaussiennes (Gaussian Mixture Model)
IP	IP pour "Internet Protocol", les caméras IP correspondent à des caméras réseau
HMM	Hidden Markov Model
kNN	k-plus proches voisins (k-nearest neighbour)
LED	Light Emitting Diode
PETS	Performance Evaluation of Tracking and Surveillance (système de vidéo surveillance)
LVQ	Learning Vector Quantization
MLP	Multi-Layer Perceptron
MoCap	Motion Capture
RBF	Radial Basis Function
SOM	Self Organizing Maps
SVM	Support Vector Machines
TDNN	Time Delay Neural Network
VSAM	Visual Surveillance and Monitoring (système de vidéo surveillance)
W^4	Who ? When ? Where ? What ? (système de vidéo surveillance)

REMERCIEMENTS

Je remercie tout particulièrement Jean Meunier de m'avoir accueillie au Laboratoire de traitement d'images du Département d'Informatique et de Recherche Opérationnelle (DIRO) et d'avoir dirigé mes travaux de recherche. J'ai apprécié tes conseils et tes commentaires sur mon travail de recherche, ainsi que ton enthousiasme dans les périodes de doutes. Je te remercie aussi de m'avoir confié le cours de traitement d'images pendant que tu étais directeur du DIRO. Cette expérience a été très enrichissante pour moi et m'a permis de faire mes premières armes en enseignement. Je remercie également Jacqueline Rousseau, du Centre de Recherche de l'Institut Universitaire de Gériatrie de Montréal, et Alain St-Arnaud, du Centre de Santé et de Services Sociaux Lucille-Teasdale, qui m'ont apporté leur aide au niveau de la compréhension de la chute chez les personnes âgées. Merci aussi Alain pour avoir joué les cascadeurs en simulant des chutes dans nos séquences vidéos.

Merci aux membres du jury d'avoir accepté d'évaluer mon travail de thèse.

Je tiens aussi à remercier toutes les personnes que j'ai pu côtoyer dans le laboratoire, les professeurs Sébastien Roy et Max Mignotte, ainsi que tous mes collègues et amis du laboratoire. Merci Isabelle pour ton accueil à mon arrivée dans le laboratoire. Merci Edouard de m'avoir prété main forte pour le dataset de chutes alors que j'étais enceinte de presque neuf mois. Merci Étienne pour ton enthousiasme fédérateur avec le projet PETS. Merci Myriam d'avoir partagé avec moi une des conférences les plus sympathiques auxquelles j'ai pu assister. Et merci de m'avoir fait découvrir qu'il y a bien pire que le bus pour aller à New York (Amtrak). Merci Amani pour tes petites pâtisseries Tunisiennes bloquées par les douanes Canadiennes pendant 2 mois. Pour rester dans les sucreries, merci Sébastien pour les délicieux baklavas. Merci Mélissa pour ta bonne humeur contagieuse. Et merci aussi à tous les autres trop nombreux pour être cités, mais notamment Mohamed, Di, Lucie, Vincent, Hung, Hoang Anh, Jamil et Nicolas.

Merci au CRSNG et au gouvernement du Canada pour leur support financier.

À mon conjoint Matthieu, qui s'est lancé avec moi dans cette aventure, et à ma fille, Méline, qui ensoleille ma vie.

CHAPITRE 1

INTRODUCTION

1.1 Une population vieillissante

Le Canada, comme de nombreux pays occidentaux, fait face au vieillissement de sa population. D'après l'agence de santé publique du Canada [89], un Canadien sur huit était âgé de plus de 65 ans en 2001. En 2026, cette proportion passera à un sur cinq, dû notamment aux « baby boomers » de l'aprés guerre et à l'espérance de vie qui augmente (voir Fig. 1.1).

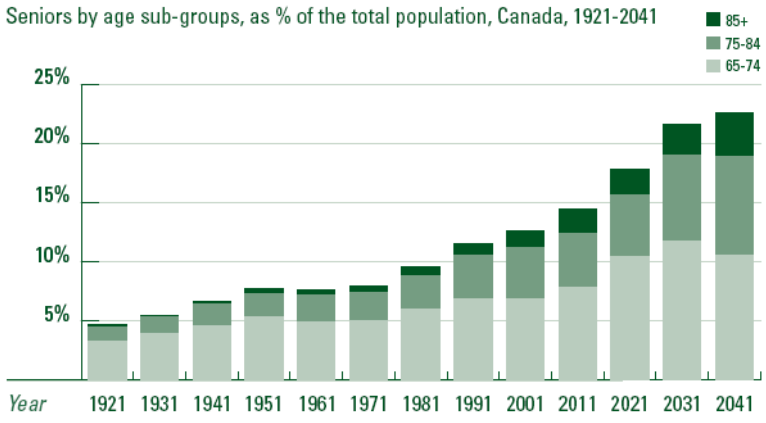


Figure 1.1 – Le vieillissement de la population au Canada [89].

La situation est identique au Québec puisque la proportion de personnes âgées va doubler d'ici vingt-cinq ans [31]. Le nombre croissant de personnes âgées nous incite à développer des solutions innovantes pour les aider à vivre mieux et en sécurité dans leur environnement. En effet, les aînés vivent majoritairement à domicile, notamment 90% des 65-85 ans [31], ce qui constitue pour eux un des aspects importants de leur qualité de vie. Plusieurs études [22, 54, 101] ont montré l'intérêt de maintenir les personnes âgées à domicile d'un point de vue humain, mais aussi d'un point de vue financier.

1.2 Le maintien à domicile

Les personnes âgées en perte d'autonomie, vivant seules à domicile, font face à de nombreux risques, tels que des risques de chutes, de mauvaise médication ou de troubles cognitifs. Ces risques peuvent être diminués par l'utilisation de capteurs qui permettent de sécuriser l'environnement de la personne âgée :

Capteurs portés par la personne Afin d'avoir des informations plus précises sur l'activité et le comportement de la personne, le développement de capteurs portés par la personne s'est généralisé. Il existe différents types de capteurs portables : capteurs pour contrôler des paramètres physiologiques (pression artérielle, température, fréquence cardiaque, etc), des accéléromètres, des boutons d'alerte, etc. Ces capteurs donnent des informations sur l'état de la personne et peuvent être utilisés pour alerter les secours. Cependant, le port d'un capteur par les personnes âgées est problématique, car elles oublient souvent de le porter. De plus, ils ont besoin de piles pour fonctionner qui doivent être changées régulièrement.

Capteurs intégrés à l'environnement Les maisons intelligentes peuvent être équipées de toutes sortes de capteurs, par exemple, des capteurs infrarouge pour la détection de mouvement, des détecteurs de contact de portes/fenêtres, des capteurs de température, etc. Ces capteurs donnent des informations sur l'environnement de la personne. Dans le cas d'un détecteur de mouvement, il indique seulement si la personne est dans la pièce ou non, mais pas ce qu'elle est en train de faire. Depuis quelques années, l'utilisation de caméras commence à prendre de l'ampleur permettant une analyse plus fine du comportement de la personne par rapport à son environnement.

1.3 La chute

La chute est un des risques les plus graves pour les personnes âgées vivant seules à domicile, et peuvent causer de graves blessures d'où l'intérêt des systèmes de détection de chutes. D'après l'Agence de Santé Publique du Canada [90], 62% des hospitalisations

par blessure chez les personnes âgées sont dues à des chutes. Elles sont la cause de 90% des blessures de la hanche chez les personnes âgées, et 20% d'entre elles décèdent dans l'année qui suit. Même sans blessure, une chute peut causer une perte de confiance chez la personne âgée et une réduction de ses activités.

1.3.1 Caractéristiques d'une chute

Dans de récents papiers [85, 86], Noury *et al.* ont essayé de définir une chute et de classifier les différentes méthodes de détection de chutes. L'événement « chute » peut être découpée en 4 phases comme montré sur la Fig. 1.2 :

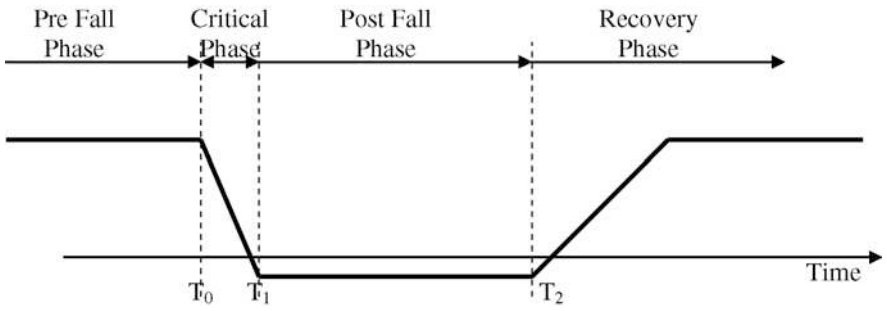


Figure 1.2 – Les différentes phases d'une chute [86].

Phase de pré-chute La personne exerce ses activités quotidiennes, avec occasionnellement des mouvements soudains tels que « s'asseoir » ou « s'accroupir ». Ces activités ne doivent pas générer d'alarme par le détecteur de chutes.

Phase critique Cette phase, très courte, correspond à un mouvement soudain du corps en direction du sol, terminé par un impact au sol.

⇒ *Méthodes « directes » de détection de chutes :*

1. Détection précoce de la chute (vitesse du corps par exemple)
2. Détection de l'impact au sol

Phase de post-chute Juste après la chute, le personne reste le plus souvent immobile, incapable de bouger et allongée sur le sol.

⇒ *Méthodes « indirectes » de détection de chutes :*

1. Personne allongée
2. Absence de mouvement

Phase de rétablissement Éventuellement, la personne peut réussir à se relever toute seule ou avec l'aide de quelqu'un.

La difficulté dans la détection de chutes n'est pas seulement de détecter un maximum de chutes, mais aussi de faire attention à ce que le système ne renvoie pas trop de fausses alarmes avec des activités normales similaires à des chutes. Par exemple, les caractéristiques d'une personne âgée se laissant brutalement tomber dans le canapé sont assez similaires aux caractéristiques d'une chute. La vitesse peut être assez forte, presque aussi forte que lors d'une chute où la personne se retient à un meuble.

1.3.2 Méthodes de détection de chutes

a - Capteurs portés par la personne

Les méthodes habituelles de détection de chutes font principalement appel à des capteurs ou boutons poussoirs portés par la personne. Les boutons d'alerte [36] permettent à la personne âgée d'appeler de l'aide en cas de problèmes. Dans le cas de la chute, ce type de technologie n'est efficace que si la personne est consciente après la chute et si elle n'est pas immobilisée ou dans l'incapacité de pouvoir actionner un bouton. Les capteurs automatiques de détection de chutes [14, 64, 65, 88] sont plus intéressants car ils ne nécéssitent pas d'intervention humaine. Certains sont basés sur des accéléromètres [64, 65] qui détectent la magnitude et la direction de l'accélération ; d'autres utilisent des gyroscopes [14] qui mesurent l'orientation du corps. Des produits commercialisés exploitent déjà ces technologies [119, 125]. La combinaison d'accéléromètres et de gyroscopes a permis à Nyan *et al.* [88] de pouvoir détecter une chute très précocement avant l'impact au sol, ce qui pourrait être utilisé pour l'utilisation de coussins gonflables pour la protection des hanches [102].

b - Capteurs au sol

Alwan *et al.* [4] ont proposé d'utiliser les vibrations au sol pour détecter des chutes. Cette idée a été reprise avec succès par Zigel *et al.* [137] en y ajoutant un détecteur de sons (combinaison d'un accéléromètre et d'un microphone). Cependant, ils admettent que leur système n'est pas suffisament sensible pour les chutes lentes et les chutes à partir d'une chaise. Par ailleurs, ce type de méthode est dépendant de la dynamique du sol (dynamique différente pour un plancher en bois ou une moquette) et n'en est qu'à ses balbutiements. La détection d'une personne au sol pourrait aussi être détecté avec un dallage sensible à la pression au sol [111], mais ce type de technologie est difficile à mettre en oeuvre dans un appartement et est certainement couteux.

c - La vidéosurveillance

L'utilisation de la vidéosurveillance est relativement récente, la section 2.7 du chapitre 2 portant sur la vidéosurveillance présente les différents travaux de détection de chutes utilisant cette technologie.

1.4 Structure du document

1.4.1 Plan de la thèse

Cette thèse par articles est composée de deux articles de journaux.

Le *Chapitre 2* présente une revue de littérature des techniques utilisées en vidéosurveillance pour détecter et suivre du mouvement, ainsi que pour la reconnaissance d'actions. Ce chapitre présente aussi les travaux actuels en vidéosurveillance pour la détection de chutes ainsi qu'un aperçu de notre système de vidéosurveillance. Le *Chapitre 3* présente notre première contribution avec un article portant sur la détection de chutes en utilisant seulement des informations 2D issues de la caméra. Le *Chapitre 4* présente les outils nécessaires pour récupérer une information 3D à partir d'une seule caméra qui seront utilisés dans le deuxième article. Le *Chapitre 5* présente notre deuxième contribution avec un article portant sur la localisation 3D d'une personne dans une pièce en utilisant une seule caméra calibrée, avec une application pour la détection de chutes. Une discussion et conclusion sur nos travaux est présentée dans le *Chapitre 6*.

1.4.2 Publications

Les principales communications dans des conférences et journaux internationaux reliées à nos travaux sont les suivantes :

- **Travaux sur les méthodes 2D de détection de chutes**
  - *« Occlusion Robust Video Surveillance for Fall Detection »* C. Rougier, J. Meunier, A. St-Arnaud et J. Rousseau, soumis au journal *IEEE Transactions on Circuits and Systems for Video Technology*, 2009.
    $\Rightarrow$ Article présenté dans le *Chapitre 3*.
  - *« GMM classification for fall detection »* C. Rougier, J. Meunier, A. St-Arnaud et J. Rousseau, publié lors de l'école d'été *International Computer Vision Summer School (ICVSS 2009)*, Sicile, Italie, Juillet 2009.
  - *« Procrustes Shape Analysis for Fall Detection »* C. Rougier, J. Meunier, A. St-Arnaud et J. Rousseau, publié lors de la conférence *ECCV 8th International Workshop on Visual Surveillance (VS 2008)*, Marseille, France, Octobre 2008.
  - *« Fall Detection from Human Shape and Motion History using Video Surveillance »* C. Rougier, J. Meunier, A. St-Arnaud et J. Rousseau, publié lors de la conférence *IEEE 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW)*, Niagara Falls, Canada, Mai 2007.
- **Travaux sur les méthodes 3D de détection de chutes**
  - *« 3D Head Tracking using a Single Calibrated Camera »* C. Rougier, J. Meunier, A. St-Arnaud et J. Rousseau, soumis au journal *Image and Vision Computing*, 2010.
    $\Rightarrow$ Article présenté dans le *Chapitre 5*.
  - *« 3D head trajectory using a single camera »* C. Rougier et J. Meunier, accepté à la conférence *International Conference on Image and Signal Processing (ICISP)* , Trois-Rivières, Canada, Juillet 2010.
  - *« Monocular 3D Head Tracking to Detect Falls of Elderly People »* C. Rougier, J. Meunier, A. St-Arnaud et J. Rousseau, publié lors de la conférence *International Conference of the IEEE Engineering in Medicine and Biology Society*,

New York, USA, Septembre 2006.

- *« Fall Detection Using 3D Head Trajectory Extracted From a Single Camera Video Sequence »* C. Rougier et J. Meunier, publié lors de la conférence *Démo, First International Workshop on Video Processing for Security (V4PS-06)*, Québec, Canada, Juin 2006.

1.4.3 Co-auteurs

Ce travail s'inscrit dans un projet en collaboration entre le Département d'Informatique et de Recherche Opérationnelle (DIRO) et le Centre de Recherche de l'Institut Universitaire de Gériatrie de Montréal (CRIUGM). Les co-auteurs des articles présentés dans cette thèse sont :

Jean Meunier est le directeur de cette thèse et travaille dans le laboratoire de traitement d'images du Département d'Informatique et de Recherche Opérationnelle de l'université de Montréal.

Alain St-Arnaud**,** neuropsychologue, travaille au Centre de Santé et de Services Sociaux Lucille-Teasdale.

Jacqueline Rousseau**,** professeure en ergothérapie, travaille au Centre de Recherche de l'Institut Universitaire de Gériatrie de Montréal.

Alain St-Arnaud et Jacqueline Rousseau nous ont apporté leur aide au niveau de la compréhension de la chute chez les personnes âgées.

CHAPITRE 2

LA VIDÉOSURVEILLANCE POUR LA DÉTECTION DE CHUTES

2.1 La vidéosurveillance

2.1.1 Évolution des systèmes de vidéosurveillance

La vidéosurveillance a connu plusieurs développements depuis la première génération de caméras basées sur les CCTV (Closed Circuit Television) analogiques [121]. Les systèmes CCTV font partie des systèmes dit « manuels » car ils nécessitent un opérateur pour contrôler tous les écrans CCTV (surveillance de parking, surveillance de magasins, etc). Cependant, contrôler plusieurs écrans est un travail fastidieux et un événement peut facilement être manqué par un opérateur. Ce type de système est donc moyennement fiable et aussi coûteux.

Est alors arrivée l'utilisation de la vision par ordinateur pour analyser la scène (détection et suivi de personnes, reconnaissance de comportements, etc). Les systèmes sont devenus semi-automatiques aidant l'opérateur en lui signalant des événements suspects (par exemple, pour la surveillance de métro ou d'un aéroport, etc). Le système est plus fiable car il sélectionne les événements suspects pour faciliter le travail de surveillance.

La troisième génération s'oriente vers des systèmes entièrement automatisés capables de distribuer l'information au travers d'un vaste réseau de caméras.

2.1.2 Applications en vidéosurveillance

De nombreux systèmes de vidéosurveillance automatisés de personnes ont été développés dans des domaines d'application variés comme le montre le tableau 2.I.

Domaine d'application	Exemples
Suivi d'objets	Suivi de personnes ou de véhicules [47, 103] Suivi de joueurs de soccer [132]
Surveillance des lieux publics	Surveillance de personnes [24] Surveillance d'aéroports [21, 114], de métros [29, 104], de stationnements [92] ou de magasins [72] Détection de noyades [94] Détection d'accidents et analyse du trafic routier [124]
Reconnaissance d'activités	Analyse de trajectoire [82, 83] Analyse de la prise de repas [45] ou de la prise de médicaments [122] Reconnaissance de gestes, de visages et d'expressions faciales [25, 110] Détection de chutes [5, 6, 8, 70, 81, 98, 107, 115, 116]

Tableau 2.I – Quelques domaines d'application de la vidéosurveillance

Parmi ces applications, certains travaux font office de référence dans le domaine de la vidéosurveillance :

- *Projet ADVISOR* [1] [104] : c'est un projet de vidéosurveillance de stations de métro impliquant des partenaires académiques et industriels dont le but est de détecter automatiquement des situations dangereuses telles que des accidents, des actes de violence ou du vandalisme. La partie détection est basée sur la différence d'images entre l'image courante et une image de fond afin d'extraire les régions en mouvement. Le suivi de la personne se fait alors en combinant les régions en mouvement, un détecteur de tête et la forme du contour de la personne.
- *Projet VSAM* [2] [24] : c'est un projet de système de vidéosurveillance développé par l'université Carnegie Mellon dans le but de permettre à un seul opérateur de surveiller de nombreux endroits. Pour détecter et suivre des objets en mouvement, ils utilisent une méthode basée sur la différence par rapport à une image de fond associée à la différence temporelle. Les objets en mouvement sont classifiés comme « personne » ou « véhicule » à l'aide de réseaux de neurones. L'étape finale de reconnaissance d'activités permet de discriminer une personne qui court d'une

[1] Annotated Digital Video for Surveillance and Optimised Retrieval
[2] Visual Surveillance and Monitoring

personne qui marche, mais aussi d'analyser les intéractions entre les objets (rencontres de personnes, véhicule déposant quelqu'un, etc).

- *Projet W^4* [3] [47] : le but du projet W^4 est de détecter et suivre de multiples personnes dans un environnement extérieur. Sans utiliser d'information couleur, à l'aide de caméras monochromatiques, ils sont capables d'analyser les activités et les interactions entre les personnes en combinant l'analyse de la silhouette et le suivi des membres des personnes.
- *Projet Pfinder* [4] [129] : système temps réel pour extraire et suivre une personne afin d'obtenir une description 3D de cette personne.

Face au nombre croissant de travaux en vidéosurveillance et face à la nécéssité de trouver des outils pour évaluer et comparer les différents algorithmes, des bibliothèques de vidéos et des ateliers en vidéosurveillance sont apparus servant de référence à la communauté scientifique :

- La base de données *CAVIAR* [5] [20] contient des vidéos pour l'analyse de comportements dans des espaces publics tels qu'un centre commercial. Les actions sont diverses : marche, rencontre de personnes, entrée/sortie de magasins, bagarres, oubli d'un bagage, etc.
- *PETS* [6] est définitivement le plus connu des ateliers en vidéosurveillance. Tous les participants travaillent sur la même base de données de vidéos en un temps limité. Plus d'une dizaine d'ateliers ont déjà eu lieu avec différents thèmes, comme par exemple l'analyse de la foule (PETS 2009 [2]) ou la détection de bagages abandonnés (PETS 2006 [1]), atelier auquel nous avons participé [7].
- La base de données *HumanEva* [7] [106] contient des vidéos pour valider les algorithmes d'estimation de pose et de mouvement d'une personne. Des vidéos, issues d'un système multi-caméras calibré et synchronisé, ainsi que la vraie pose 3D, is-

[3] Who ? When ? Where ? What ?
[4] Person finder
[5] Context Aware Vision using Image-based Active Recognition
[6] Performance Evaluation of Tracking and Surveillance
[7] Synchronized Video and Motion Capture Dataset for Evaluation of Articulated Human Motion

sue d'un système de « Motion Capture », sont fournies pour plusieurs personnes effectuant différentes actions. Comme nous allons utiliser cette bibliothèque dans nos travaux, l'annexe II présente une description plus approfondie de cet ensemble de données.

2.2 Vision de jour et/ou vision de nuit

Le choix de la caméra dépend des objectifs du système de vidéosurveillance, à savoir si le système doit fonctionner uniquement de jour ou aussi de nuit. Les systèmes de vidéosurveillance classiques pour la vision de jour utilisent des capteurs en niveaux de gris ou en couleur, la technologie étant soit de type CCD[8] ou CMOS[9].

De nombreux systèmes de vidéosurveillance fonctionnent avec des capteurs en niveaux de gris [47] car ils sont peu dispendieux. Cependant, on perd une information importante, la couleur, qui peut être très utile pour la détection ou le suivi de personnes. Ainsi, de nombreux systèmes de vidéosurveillance utilisent plutôt des capteurs couleur [87, 103] car ils deviennent de moins en moins dispendieux pour une bonne qualité d'image. Cependant, les capteurs couleur utilisés en vidéosurveillance sont souvent bas de gamme, et il ne faut pas espérer pouvoir faire des traitements colorimétriques aussi performants qu'avec une caméra haut de gamme.

Pour une application qui doit fonctionner en vision de nuit, dans le noir total ou en faible luminosité, il faut se tourner vers d'autres systèmes : systèmes qui amplifient la lumière, caméras thermiques ou éclairage infrarouge [40] :

En cas de faible luminosité, on peut utiliser des systèmes qui intensifient la lumière sur le capteur. Mais ce type de système est souvent bruité et parasité par les lumières ponctuelles, et il a besoin d'une faible lumière ambiante pour fonctionner parfaitement.

Les caméras thermiques sont intéressantes car elles ne nécessitent pas de lumière ambiante. En effet, les caméras thermiques détectent la chaleur émise (radiation infrarouge) par les objets de la scène. Plus l'objet est chaud, plus son image sur le capteur sera claire. Plusieurs applications pour surveiller des parkings ou des lieux publics la nuit utilisent

[8] Charged Coupled Device
[9] Complementary Metal Oxide Semi-conductor

des caméras thermiques, car elles sont capables de détecter des personnes à de grandes distances. Le projet W^4 [47] est un exemple d'application fonctionnant de jour comme de nuit car il est capable de fonctionner aussi bien avec une caméra thermique qu'avec une caméra en niveaux de gris. Sixsmith et Johnson [107] proposent d'utiliser un capteur infrarouge pour détecter des chutes et Treptow *et al.* [117] présentent un robot capable de suivre une personne avec une caméra thermique. Un des inconvénients majeurs de cette technologie est son prix. Pour un système de vidéosurveillance moins dispendieux, on s'orientera plutôt vers un éclairage infrarouge.

Ce type de système est très populaire car peu dispendieux comparé aux autres systèmes de vision de nuit. Il consiste à éclairer la scène avec un éclairage dans le proche infrarouge, l'image est alors acquise par un capteur sensible pour ces longueurs d'onde. Le proche infrarouge correspond à des longueurs d'onde bien particulières : ce que l'oeil humain est capable de voir, le domaine du visible, correspond à des longueurs d'onde comprises entre 400 et 700 nm, le proche infrarouge correspond à des longueurs d'onde légèrement supérieures à 700 nm donc invisibles pour l'oeil humain. Pour réaliser un tel éclairage, il suffit de disposer des LEDs autour de la caméra afin d'éclairer la scène de manière uniforme, et de disposer d'un capteur sensible au proche infrarouge. En exemple d'application, un éclairage infrarouge peut être utilisé pour suivre les pupilles des yeux d'un conducteur [61] ou pour de la reconnaissance de visages [73].

2.3 Structure d'un système de vidéosurveillance

Un système de vidéosurveillance est en général composé de trois modules :

- **Détection**

  A chaque nouvelle image, la personne en mouvement doit être détectée et segmentée en régions. La section 2.4 présente les différentes méthodes de détection de mouvement.

- **Suivi**

  Il s'agit alors de suivre une personne ou une trajectoire tout au long de la séquence vidéo. Les techniques de suivi sont détaillées dans la section 2.5.

- **Reconnaissance**

  La reconnaissance de comportement permet de détecter un événement suspect et fournit en sortie un résultat correspondant à une prise de décision à savoir si le comportement est normal ou anormal. Les outils pour la reconnaissance de comportements sont expliqués dans la section 2.6.

2.4 Détection de personnes

La détection de personnes consiste à segmenter les régions en mouvement. Le mouvement peut notamment être détecté par rapport à une image de fond, par la différence temporelle ou par le flux optique.

2.4.1 Différence par rapport à une image de fond

Une des méthodes les plus utilisées pour détecter des objets en mouvement est la différence entre l'image courante et une image de fond qui a été modélisée préalablement [47, 129]. La qualité des régions extraites dépend de la bonne modélisation de l'image de fond.

Soient $I(i,j)$ la valeur du pixel à la position (i,j) du pixel courant, $B(i,j)$ la valeur du pixel de l'image de fond et $T(i,j)$ la valeur de seuil du pixel considéré, alors la différence par rapport à une image de fond s'écrit sous la forme :

$$\begin{aligned} \text{Si } |I(i,j) - B(i,j)| \geq T(i,j) \quad & \text{alors pixel } (i,j) \text{ en mouvement} \\ & \text{sinon pixel } (i,j) \text{ appartient au fond} \end{aligned}$$

Il ne faut pas oublier de mettre à jour régulièrement le modèle de fond [24, 28] afin de tenir compte des changements dans l'image de fond (objets déplacés, changement d'éclairage, etc).

Cette méthode est très simple mais est aussi très sensible aux ombres et aux surexpositions. Plusieurs personnes ont travaillé sur la détection des ombres afin de rendre cette méthode plus robuste [30, 56]. Jabri *et al.* [59] ont proposé de combiner une soustraction d'images basée sur la couleur avec une soustraction d'images basée sur le gradient, après avoir supposé qu'une ombre ne possède pas de gradient.

Récemment, une nouvelle méthode [67] est trés populaire. Elle permet de tenir compte des variations présentes dans l'image de fond. Pour chaque pixel, un *catalogue de codes* est construit contenant un ou plusieurs *codes* (forme compressée du modèle de fond).

Un *code* est défini par 9 paramètres : $(\bar{R},\bar{V},\bar{B})$ le vecteur couleur, $(\check{I},\hat{I})$ la brillance min et max pour ce *code*, λ le plus long intervalle durant lequel le *code* n'est pas apparu pendant la phase d'entraînement et (p,q) le premier et le dernier temps d'accès où le *code* a été utilisé.

Pour chaque pixel, les *codes* sont obtenus lors d'une phase d'entraînement à partir d'une séquence d'images de fond. Les *codes* sont créés par une technique de quantification de vecteurs dont la mesure de similarité est basée sur la distorsion de couleur et le champ de brillance. Les objets en mouvement sont alors détectés par comparaison avec les *codes* de l'image de fond. La figure 2.1 montre la variabilité de l'intensité d'un pixel au cours du temps lorsque l'on a un fond en mouvement (dans ce cas, une branche d'arbre en mouvement). L'algorithme est capable de segmenter la personne sans être sensible au fond en mouvement.

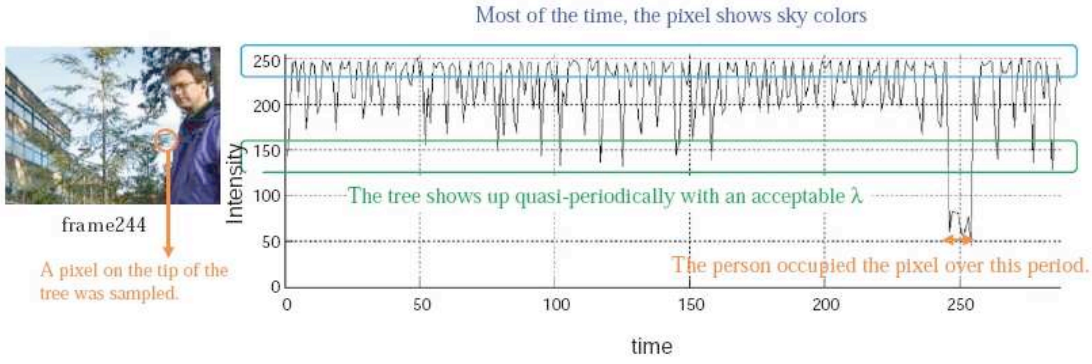


Figure 2.1 – Principe de la méthode basée sur un *catalogue de codes* [67] : la branche d'arbre en mouvement sur un fond de ciel bleu génère un fond variable quasi-périodique avec un paramètre λ faible alors que la personne génère un fort λ.

D'après l'article, cette méthode est très rapide et fonctionne particulièrement bien lorsque l'arrière plan change à cause d'objets en mouvement ou de changement d'illumination. Cet algorithme est aussi très efficace pour les vidéos compressées comme

montré dans l'article par une comparaison très convaincante avec d'autres algorithmes connus notamment [56].

2.4.2 Différence temporelle

La différence temporelle consiste à faire des différences entre deux ou trois images consécutives pour extraire les changements temporels. Si l'objet suivi est rapide, la région en mouvement à l'instant t ne sera pas superposée à celle de l'instant $t-1$. Si l'objet est lent, les zones uniformes de l'objet risquent de ne pas être segmentées comme étant un objet en mouvement. Cette technique est utilisée par le projet VSAM [24] sur trois images consécutives :

Soit $I_t(i,j)$ la valeur de l'intensité du pixel à la position (i,j) au temps t et $T_t(i,j)$ la valeur du seuil pour le pixel (i,j) pour décrire un changement significatif. Un pixel est en mouvement si :

$$(|I_t(i,j) - I_{t-1}(i,j)| \geq T_t(i,j)) \text{ et } (|I_t(i,j) - I_{t-2}(i,j)| \geq T_t(i,j))$$

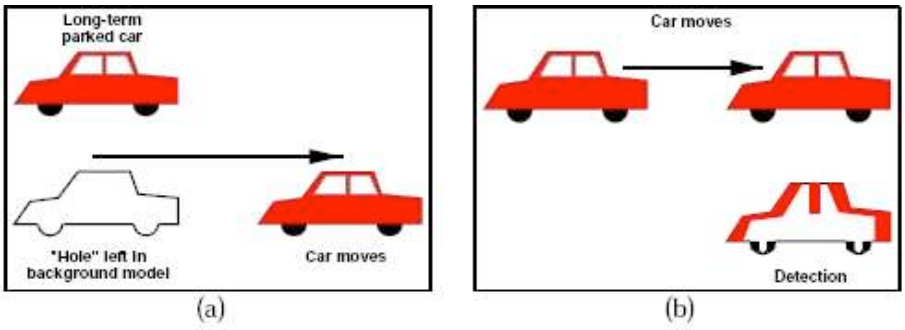


Figure 2.2 – Cette figure [24] présente les problèmes liés à la méthode de soustraction par rapport à une image de fond (a) et celle de la différence temporelle (b)

Lorsque l'image de fond change (changement d'éclairage, objet déplacé, etc), la différence par rapport à cette image va créer temporairement des objets en mouvement, jusqu'à ce que l'image de fond soit mise à jour (cf figure 2.2 (a)). On n'aura pas ce problème avec la différence temporelle car on considère des images consécutives. Cependant, dans le cas de la différence temporelle, ce sont les zones uniformes intérieures

à l'objet qui ne seront pas détectées en mouvement (cf figure 2.2 (b)). La combinaison de ces deux techniques permet de contrer ces problèmes.

2.4.3 Flux optique

Le flux optique [9] permet de calculer les champs de vecteurs des objets en mouvement. A partir de ces champs de vecteurs, il est possible de segmenter les régions en mouvement. Le flux optique est particulièrement utile lorsque la caméra est en mouvement, mais ce n'est pas le cas dans notre application. Calculer le flux optique est souvent complexe en temps de calcul, ce qui fait qu'il est peu utilisé en vidéosurveillance. De plus, le flux est sensible au bruit dans l'image. Quelques articles utilisent quand même le flux optique pour détecter du mouvement, en limitant la zone de recherche pour augmenter la rapidité de l'algorithme. Par exemple, Gao *et al.* [45] utilisent le flux optique pour suivre les avant-bras et la tête d'une personne âgée prenant son repas. Jean *et al.* [60] proposent d'utiliser le flux optique pour suivre les pieds d'une personne en limitant leur recherche dans une zone restreinte.

2.5 Suivi d'une personne

Le but de notre système de vidéosurveillance est d'analyser et d'interpréter le comportement de la personne. Pour cela, il est nécessaire de récupérer des informations dans la vidéo telles que la trajectoire du centre de gravité de la personne ou de sa tête, sa démarche, etc. Le suivi est donc une étape importante du système car il va servir à récupérer les données nécessaires à l'étape suivante de reconnaissance. Les techniques de suivi sont très diverses, on peut suivre une partie du corps (tête, visage, jambes, etc) ou la totalité de la personne. Parfois, il est nécessaire de faire un suivi plus fin en cherchant à modéliser la personne par une représentation de type squelette 2D/3D ou autre.

Dans cette partie, nous allons donc explorer quatre grandes approches : suivi par approche région, suivi à l'aide d'une modèle, suivi par approche contours, suivi à l'aide d'attributs. Pour chacune de ces méthodes, il faut garder en tête qu'une bonne méthode de suivi doit être robuste, précise et surtout rapide pour pouvoir suivre un objet en temps

réel. La qualité du suivi est aussi dépendante d'une bonne détection des objets en mouvement.

2.5.1 Outils mathématiques pour le suivi

Avant de présenter les différentes techniques de suivi, nous allons faire une parenthèse pour aborder quelques outils mathématiques très utilisés dans ce domaine : le filtre de Kalman et le filtre à particules.

a - Filtre de Kalman

Le filtre de Kalman [127], basé sur une distribution Gaussienne, est un outil récursif linéaire pour estimer l'état d'un système. Appliqué au problème du suivi de personnes, le filtre de Kalman sert à estimer la position d'une personne à l'instant $t+1$ à partir de celle observée à l'instant t. Pour chaque nouvelle position à estimer, le filtre de Kalman se déroule en 2 étapes de façon cyclique (fig. 2.3).

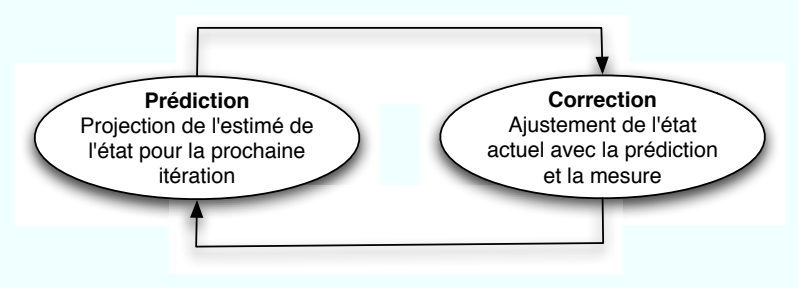


Figure 2.3 – Le cycle de Kalman

Ce processus cyclique consiste à prédire la position de la prochaine itération (phase de 'prédiction'), puis à ajuster la position actuelle en fonction de la position prédite et de celle qui a été mesurée (phase de 'correction'). Le filtre de Kalman a été utilisé avec succès pour suivre des véhicules [62] ou des personnes [105]. Cependant, la trajectoire d'une personne peut être aléatoire avec des brusques changements de direction, et comme le filtre de Kalman est un filtre prédictif, il risque de perdre sa trace. De plus, comme le filtre de Kalman est basé sur une distribution Gaussienne, il devient inadéquat en présence d'occlusions ou de suivi multi-cibles.

b - Filtre à particules

Contrairement à l'algorithme de Kalman, les filtres à particules, basés sur l'algorithme Condensation (Conditional Density Propagation) [58], sont utilisés pour des systèmes non linéaire avec des modèles d'observation non gaussiens. Ainsi, ils sont bien adaptés pour suivre une trajectoire avec des brusques changement de direction, et pour le suivi de multiples personnes.

L'algorithme du filtre à particules se déroule en trois étapes (fig. 2.4) :

1. ***Sélection :*** Un nouvel ensemble de particules est construit aléatoirement en favorisant les meilleures particules de l'ensemble précédent.
2. ***Prédiction :*** Chaque échantillon est propagé à l'aide d'un modèle dynamique stochastique.
3. ***Mesure :*** Un poids est calculé pour chaque échantillon du nouvel ensemble.

Une fois que tous les échantillons ont été construits, le résultat du suivi est une estimation pondérée de l'ensemble des particules.

Les filtres à particules ont été utilisés pour suivre des personnes [81], mais aussi des parties du corps telles que la tête ou les mains [58, 87, 117]. Cette méthode de suivi semble très prometteuse et est utilisée dans de nombreux travaux récents. Ils donnent d'excellents résultats à condition d'avoir suffisamment de particules pour avoir un bon maximum de vraisemblance pour l'état estimé. Deutscher *et al.* [34] ont incorporé le principe du recuit simulé au filtre à particules classique avec leur algorithme appelé « annealed particle filter ». Le fait de gérer les particules sous la forme de plusieurs couches permet d'obtenir un résultat plus précis avec le même nombre de particules.

2.5.2 Suivi par approche régions

L'approche région est l'une des plus classiques : les personnes ont été segmentées dans l'image, et le suivi consiste à faire correspondre des régions entre deux images consécutives.

Un des suivis les plus simples est le suivi par recouvrement [77]. Il ne nécessite pas de prédiction de position, les objets sont appariés par recouvrement de leur boîte

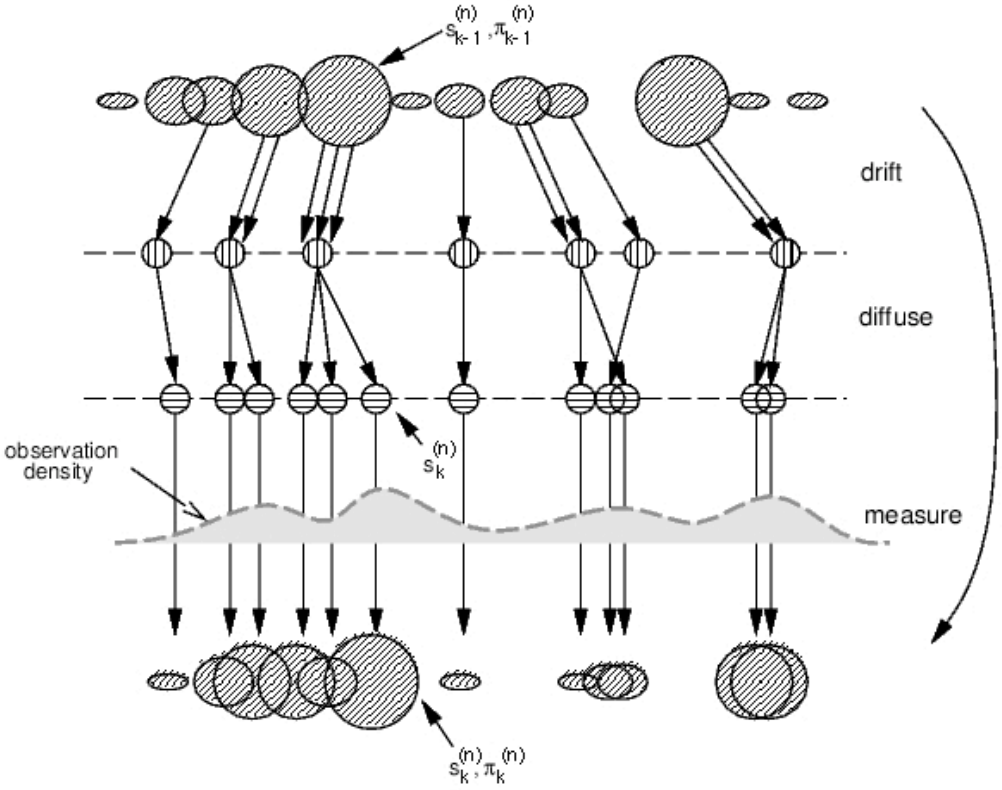


Figure 2.4 – Principe d'un filtre à particules [58] où l'on retrouve les trois grandes étapes : sélection, prédiction et mesure

englobante entre l'image $t-1$ et l'image t. Cette méthode nécessite que le mouvement ne soit pas trop important pour qu'il y ait recouvrement.

Fuentes et Velastin [42] utilisent des matrices d'appariements entre les régions de l'image *t-1* et celles de l'image *t* et inversement, qui permettent de suivre des objets selon des critères colorimétriques. L'avantage par rapport à la méthode précédente est que la quantité de mouvement entre deux images peut être plus importante puisque l'on se base sur l'information couleur.

Toute technique de suivi peut être améliorée par la combinaison d'autres informations telles que la texture, la forme, la vitesse de la personne, etc.

Le suivi de régions ne se limite pas au suivi de personnes, il peut parfois être intéressant de suivre un élément caractéristique de la personne tel que ses pieds, ses mains ou

sa tête [12, 47, 58, 60, 87]. Pour la détection de chute, il peut être intéressant de suivre la tête car elle est quasiment toujours présente dans l'image, et lors d'une chute, la tête aura une vitesse et une accélération qui vont augmenter et qui seront déterminantes pour détecter cet événement.

Plusieurs techniques existent pour le suivi de tête : Birchfield [12] propose de modéliser la tête par une ellipse et de la suivre par une méthode basée sur le gradient et les histogrammes couleur. C'est une méthode simple qui fonctionne relativement bien et rapidement à condition que le mouvement ne soit pas trop important entre deux images car ils utilisent une fenêtre de recherche pour retrouver la tête dans l'image suivante. Une ellipse représentant la tête est aussi utilisée par Nummiaro *et al.* [87] en utilisant un filtre à particules basé sur la couleur.

2.5.3 Suivi à l'aide d'un modèle

Une autre méthode de suivi consiste à ajuster un modèle à la personne. Il est possible de classifier ces modèles en deux catégories : modèles 2D et modèles 3D. Un modèle 2D s'ajuste assez facilement mais peut être erroné en fonction du point de vue de la caméra. Un modèle 3D sera plus précis mais aussi plus compliqué à déterminer. Ajuster un modèle ne nécessite pas forcément plusieurs images, les informations de l'image courante peuvent être suffisantes. Cependant, on peut améliorer un modèle en utilisant des informations temporelles qui vont permettre d'imposer des contraintes sur le modèle. Nous allons voir quelques travaux qui ont été réalisés dans ce domaine.

a - Modèle 2D

Dans la cas d'un modèle 2D, le but est d'ajuster ce modèle à la projection de la personne dans l'image. Le modèle peut être composé d'une combinaison de segments 2D ou de formes (ellipse, polygone, etc).

Une technique simple pour ajuster un modèle 2D est de rechercher des caractéristiques telles que des points ou des segments dans la région représentant la personne. Fujiyoshi et Lipton [43] du projet VSAM [24] utilisent la silhouette de la personne pour extraire un squelette 2D en forme d'étoile. Le squelette est ajusté par une recherche des

points extrêmes de la silhouette. Ce type de technique est intéressante mais nécessite une bonne détection de la personne. La silhouette de la personne ne doit pas être perturbée par des ombres, des changements d'éclairage ou des occlusions.

Un modèle 2D peut aussi être composé de formes ou de régions 2D telles que des ellipses. Des régions représentant chacune des parties du corps telles que la tête ou les mains sont utilisées dans le système Pfinder [129] en se basant sur une analyse des contours et de la couleur. La silhouette de la personne est utilisée dans le projet W^4 [47] pour extraire les parties de son corps (tête, pieds, mains et buste). Chacun de ces éléments est alors modélisé par une ellipse.

b - Modèle 3D

Le problème des modèles 2D est qu'ils sont souvent limités par l'angle de vue de la caméra. Aussi certains chercheurs ont proposé une approche 3D dont le but est d'ajuster des formes 3D de type cylindre, sphères, ou des modèles plus complexes. Nous nous intéressons ici seulement aux techniques mono-caméras et sans marqueurs placés sur la personne.

Horain et Bomb [55] proposent d'utiliser un modèle articulé 3D basé sur des images couleur et des contraintes biomécaniques pour représenter le haut de la personne (buste et tête). La limite de leur système est surtout computationnelle car leur modèle 3D requiert de nombreux calculs pour être bien ajusté. Ils ont eu aussi des problèmes au niveau de la segmentation couleur, et imposent que la peau, les vêtements et le fond soient de couleurs différentes. Sminchisescu et Triggs [109] présentent une méthode pour ajuster un modèle 3D avec une seule caméra basée sur les contours de la personne et sur le mouvement extrait par flux optique. Ce type de méthode donne de bons résultats, mais ne peut pas être utilisé pour des applications temps réel étant donné la complexité des algorithmes. Le problème des modèles 3D est qu'ils sont souvent difficiles à estimer car il est nécessaire d'estimer beaucoup de paramètres. Plus le modèle est raffiné, plus les temps de calculs sont longs pour l'ajuster, ce qui est difficilement compatible avec du temps réel. De plus, comme pour un modèle 2D, si la personne est mal segmentée (occultation, etc), le modèle risque d'être mal ajusté.

2.5.4 Suivi par approche contours

L'approche contour consiste à suivre un objet par ses contours. Par exemple, pour le projet Advisor [103], le système suit la forme de la personne approximée par une B-spline en utilisant un filtre de Kalman. Un filtre à particules est utilisé par Isard et Blake [58] pour suivre la tête ou la main d'une personne à l'aide d'un modèle déformable. Paragios et Deriche [93] présentent une méthode basée sur les contours actifs géodésiques pour détecter et suivre plusieurs objets dans une vidéo. L'intérêt de suivre un contour est que le modèle suivi est souvent simple, ce qui rend la méthode rapide en temps de calculs. Cependant cette technique nécessite une initialisation robuste et automatique, ce qui n'est pas forcément évident.

2.5.5 Suivi à l'aide d'attributs

Cette approche consiste à suivre des attributs tels que des lignes ou des points, qui sont, en général, faciles à extraire. Par exemple, il est possible de suivre les yeux d'un conducteur grâce à un éclairage infrarouge [61] : les pupilles du conducteur apparaissent très claires sur l'image, et à l'aide d'un filtre de Kalman, le système est capable de suivre les pupilles et de voir l'état d'éveil du conducteur.Un avantage de cette méthode est que même en cas d'occlusion partielle, le reste des caractéristiques visibles peut permettre de poursuivre le suivi.

2.6 Reconnaissance de comportements

Contrairement à la détection et au suivi de personnes où de nombreux travaux ont été faits, la reconnaissance de comportements est un domaine de recherche où il reste encore beaucoup de choses à explorer. En effet, la reconnaissance de comportements est une chose complexe, les activités des personnes sont si diverses qu'il n'est pas évident de détecter un événement suspect. Nous allons aborder dans cette partie deux types de méthodes de reconnaissance : la reconnaissance statique basée sur des critères obtenus à partir d'une seule image, et la reconnaissance dynamique basée sur une analyse temporelle de la séquence d'images.

2.6.1 Reconnaissance statique

Les méthodes pour la reconnaissance statique sont basées sur des comparaisons entre l'image courante et des informations préalablement enregistrées. Une des applications type est la reconnaissance de posture où l'on cherche à reconnaître la posture d'une personne à partir d'éléments connus ou d'informations extraites des images précédentes.

a - Approche région

L'approche région consiste à reconnaître une posture à l'aide d'un descripteur de régions. Par exemple, l'utilisation de la forme de la région est une technique simple et qui donne une information intéressante sur la position de la personne, sa taille, sa forme, etc. Lee et Mihailidis [70] utilisent la forme de la silhouette de la personne et sa vitesse pour détecter une chute. La caméra est placée au plafond ce qui permet de détecter si une personne est couchée au sol par la forme allongée de sa boîte englobante. Nait-Charif et McKenna [81] suivent la personne à l'aide d'un filtre à particules et approximent sa silhouette par une ellipse. Une fois encore, la caméra est située au plafond et permet de détecter une chute lorsque l'ellipse s'allonge. Approximer une personne par une forme est une technique simple mais qui peut être faussée par le point de vue de la caméra ou par des occlusions. C'est aussi une méthode globale qui ne permet pas d'avoir beaucoup de précision sur la posture.

b - Approche contour

L'approche contour est une analyse plus fine. Elle consiste à extraire la silhouette de la personne sous la forme d'un contour, puis la description du contour va permettre par exemple de déterminer l'identité de la personne ou sa pose. Une des méthodes les plus anciennes pour coder un contour est le codage de Freeman [41] qui consiste à coder les directions du contour à partir d'une origine donnée.

Mokhtarian [78] utilise le descripteur de courbure CSS (Curvature Scale Space) pour décrire les contours d'un objet. On compare alors des silhouettes par appariement entre des courbes CSS en considérant la position des maximums de courbure. Díaz de León et Sucar [71] utilisent les descripteurs de Fourier pour représenter une silhouette humaine. Les descripteurs de Fourier sont obtenus à partir d'une transformée de Fourier discrète.

Il suffit de sélectionner un certain nombre de descripteurs pour caractériser la forme, sachant que plus on prend de descripteurs, plus on aura d'information sur les détails, donc plus la forme sera complexe. Pour comparer des formes, il s'agit alors de comparer leurs descripteurs par ordre croissant. Pour leur application, ils utilisent 40 descripteurs normalisés pour caractériser une silhouette humaine.

Veeraraghavan *et al.* [123] se basent sur la théorie mathématique des formes de Kendall pour décrire la silhouette de la personne. Ils utilisent ces données pour analyser la nature des déformations de la forme pour reconnaître sa démarche. Bauckhage *et al.* [10] décrivent la forme de la personne par un treillis 2D qui leur permet d'extraire des points caractéristiques de la silhouette. Cet ensemble de points caractéristiques est ensuite utlisé pour classifier si la personne a une démarche normale ou anormale à l'aide d'un SVM[10].

2.6.2 Reconnaissance dynamique

La reconnaissance dynamique s'effectue sur plusieurs images, on s'intéresse alors à la reconnaissance d'actions. Une des techniques les plus utilisées pour reconnaître une action est d'analyser la trajectoire de l'objet. Le problème devient alors un problème de classification de trajectoires avec d'un côté, un grand nombre de trajectoires dites "normales" et de l'autre, un petit nombre ("outlier") de trajectoires suspectes. Les trajectoires normales sont en général bien représentatives de cette classe, mais parfois il manque des exemples dans la classe des trajectoires anormales. Il existe de nombreux types de classifieurs [37, 39], nous allons aborder les méthodes les plus connues : les méthodes de recalage temporel, les modèles de Markov caché, les méthodes connexionnistes et quelques autres classifieurs.

a - Approche recalage temporel

La technique la plus connue de recalage temporel est la déformation temporelle dynamique, DTW[11], qui consiste à rechercher le meilleur alignement entre deux trajectoires à l'aide de la programmation dynamique (fig.2.5). Cette méthode a été utilisée

[10] Support Vector Machine
[11] Dynamic Time Warping

initialement pour la reconnaissance de la parole, et depuis, est utilisée pour d'autres applications telles que le recalage de trajectoires [96], la reconnaissance de caractères [84] ou la reconnaissance de gestes [25].

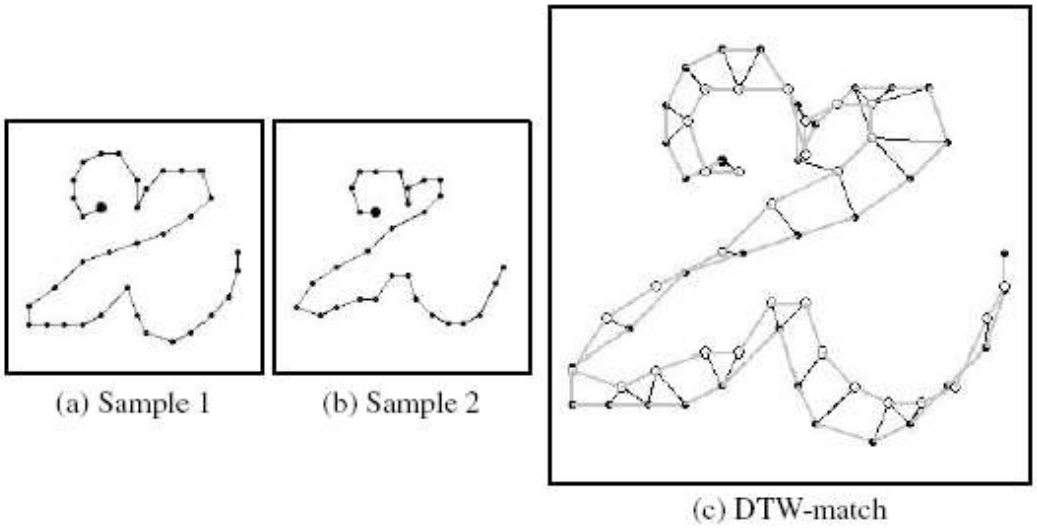


Figure 2.5 – Exemple de recalage avec la DTW [84]

Une autre méthode de recalage temporel est celle proposée par Rao *et al.* [97] qui utilisent la courbure spatio-temporelle d'une trajectoire 2D pour détecter ce qu'ils appellent des instants dynamiques, c'est à dire des changements importants dans le mouvement (vitesse, direction et accélération). Ces caractéristiques sont ensuite utilisées lors du recalage temporel de la trajectoire afin de reconnaître l'action d'une personne. Jiao *et al.* [62] proposent de reconstruire une trajectoire 3D à partir de ses projections 2D extraites de plusieurs caméras. Pour cela, ils utilisent le fait qu'une courbe 3D est décrite de façon unique par sa courbure et ses vecteurs de torsion.

b - Les modèles de Markov cachés

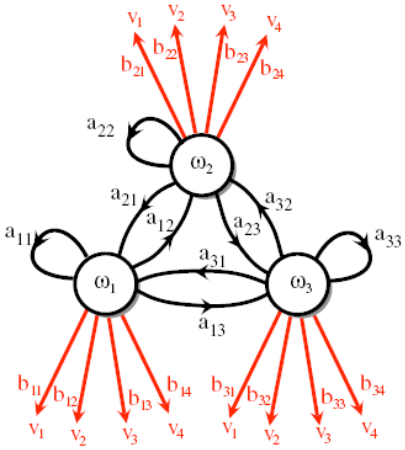
Les modèles de Markov cachés (MMC, plus connus sous le nom de HMM[12]) sont des automates probabilistes, particulièrement bien adaptés pour classifier des séquences d'actions. Ils sont par exemple utilisés pour la reconnaissance de la parole, l'analyse

[12]Hidden Markov Model

d'écriture ou pour la reconnaissance de gestes. Ils se basent sur l'hypothèse de Markov qui énonce que l'état futur ne dépend que de l'état présent.

1 - Structure d'un HMM et apprentissage

Le but d'un HMM est d'identifier une séquence d'événements étant donné un ensemble d'observations V (états visibles). Les paramètres d'un modèle HMM de 1^{er} ordre (ne dépendant que de l'état précédent) sont montrés sur la figure 2.6.



- États visibles : $V^T = \{v(1), v(2), ..., v(T)\}$
- États cachés : $\omega^T = \{\omega(1), \omega(2), ..., \omega(T)\}$
- $a_{ij} = P\left(\omega_j(t+1) \mid \omega_i(t)\right)$: probabilité de transition pour passer de l'état $\omega_i(t)$ à l'état $\omega_j(t+1)$
- $b_{jk} = P\left(v_k(t) \mid \omega_j(t)\right)$: probabilité d'émission c'est à dire probabilité d'observer un état $v_k(t)$ étant donné un état $\omega_i(t)$

Figure 2.6 – Exemple de structure d'un HMM [39]

Les paramètres du HMM sont déterminés par un algorithme d'apprentissage. Cet algorithme sert à entraîner le HMM à partir d'une séquence d'événements. Les algorithmes d'apprentissage les plus connus sont l'algorithme de Viterbi et l'algorithme de Baum Welch (algorithme de "retour arrière").

Chaque action à reconnaître est modélisée par un HMM, donc il faudra autant de HMMs qu'il y a d'actions à reconnaître.

2 - Quelques applications utilisant des HMMs

Les HMMs ont été très utilisés pour la reconnaissance d'écriture ou la reconnaissance de la parole. Ils s'appliquent dorénavant dans un domaine qui leur est proche, l'analyse de trajectoires. Gao *et al.* [45] présentent une méthode basée sur un HMM pour analyser la prise de repas d'une personne âgée. Le HMM utilise des caractéristiques de mouvement et de distance (tête, main) pour classifier les actions de la personne. D'autres

travaux proposent d'utiliser un HMM pour détecter un événement suspect sur le tarmac d'un aéroport [21]. Les actions des véhicules autour de l'avion doivent respecter un certain ordre qui est modélisé par le HMM.

Différentes variantes du HMM existent pour reconnaître des activités. Par exemple, Brand *et al.* [17] utilisent un CHMM (Coupled Hidden Markov Model) c'est à dire plusieurs HMMs assemblés pour modéliser les interactions entre eux. Quant à Galata *et al.* [3], ils utilisent un VLMM (Variable Length Markov Model) qui est un modèle d'ordre supérieur permettant de coder des dépendances temporelles sur une échelle de temps variable. Cependant d'après Nguyen *et al.* [83], ces modèles se limitent à la reconnaissance d'activités simples. Ils proposent donc d'utiliser des structures plus complexes telle que le HHMM, c'est à dire un HMM hiérarchique, pour reconnître des activités plus complexes.

c - Méthodes connexionnistes

L'approche connexionniste consiste à utiliser un réseau de neurones pour classifier les données en une ou plusieurs classes. Dans notre cas, la majorité des applications ont pour but de différencier les comportements normaux des comportements anormaux. Pour cela, il est nécessaire d'avoir un nombre suffisant et bien représentatif de données d'entraînement. Il existe plusieurs types de réseaux de neurones, chacun avec ses particularités. Commençons par examiner ce qu'est un réseau de neurones.

1 - Structure d'un réseau de neurones

Un neurone possède plusieurs entrées x_i, chacune étant affectée d'un poids ω_i, et une sortie y. Le passage des entrées vers la sortie se fait par une fonction d'activation f de type sigmoïde, radiale ou autres (fig.2.7).

Un réseau de neurones est composé d'un ensemble de neurones connectés entre eux par des liaisons affectées d'un poids. En modifiant les poids de certaines de ces connexions, on peut adapter le réseau pour qu'il donne en sortie des réponses différentes. La modification des poids est faite à l'aide d'un algorithme d'apprentissage.

2 - Apprentissage

L'apprentissage d'un réseau de neurones consiste à adapter les poids des neurones

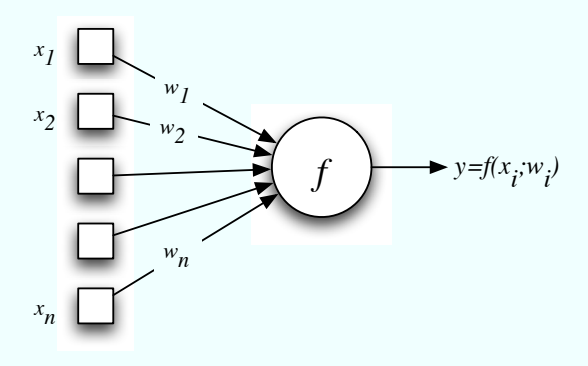


Figure 2.7 – Un neurone

du réseau afin que le réseau réponde au mieux à la tâche qui lui est demandée. Les deux types d'apprentissage les plus connus sont : l'apprentissage supervisé et l'apprentissage non supervisé.

Apprentissage supervisé : Dans le cas d'un apprentissage supervisé, le réseau reçoit en entrée des données connues, il doit alors s'adapter afin de fournir la sortie attendue. Le réseau va donc se modifier au fur et à mesure des entrées pour donner les sorties voulues.

Apprentissage non supervisé : Dans le cas d'un apprentissage non supervisé, on cherche à construire un réseau dont on ne connaît pas a priori la sortie correspondant aux données en entrée. Les données en entrée sont fournies au réseau qui va chercher à les regrouper selon des critères de ressemblance, ceci revient donc à un problème de classification.

L'apprentissage est une étape cruciale d'un réseau de neurones. Si l'on fait apprendre à un réseau de neurones toujours le même type de données, le réseau va se spécialiser et il ne généralisera plus correctement. On parle alors de surapprentissage. Lors des itérations d'apprentissage, les poids des connexions des neurones sont mis à jour à l'aide de règle d'apprentissage, dans le but d'adapter le réseau à la tâche de reconnaissance voulue. Voici quelques règles connues :

- *Règle de Hebb*

  La plus ancienne règle d'apprentissage est la règle de Hebb qui est basée sur des connaissances neurobiologiques. Plus un neurone est activé de façon synchrone et répétée, plus la force de la connexion synaptique va être importante. Ainsi, la règle de Hebb dit que si deux neurones connectés sont activés simultanément, le poids de leur connexion sera augmenté. C'est un apprentissage local.

- *Règles de correction d'erreur*

  Cette méthode est utilisée dans le cas d'un apprentissage supervisé. Soit d la sortie désirée et y la sortie calculée par le réseau, alors l'erreur $(d-y)$ est utilisée afin de modifier les connexions et de diminuer l'erreur globale du système. Le réseau va donc s'adapter par une minimisation d'un critère d'erreur pour que y se rapproche de d. Cette règle se retrouve sous plusieurs noms : Adaline (Adaptive linear Element), règle de Widrow-Hoff, règle Delta, ou rétropropagation par descente de gradient.

- *Apprentissage de Boltzmann*

  C'est une règle de type stochastique qui consiste à mettre à jour les neurones de façon probabiliste.

- *Règles d'apprentissage par compétition*

  L'apprentissage par compétition est du type « Tout pour le vainqueur » (winner-take-all). C'est à dire qu'il favorise le neurone gagnant en le rapprochant du vecteur d'entrée à qui il doit sa victoire. Pour chaque entrée, il y a un seul neurone gagnant.

3 - Les différents types de réseaux de neurones

Il existe de nombreux types de réseaux de neurones, nous allons aborder dans cette partie les réseaux les plus populaires : les réseaux de neurones non récurrents, les réseaux de neurones récurrents et les cartes topologiques.

- *Réseaux de neurones non récurrents*

  Dans un réseau de neurones non récurrents (réseau multicouche à propagation

directe), l'organisation des neurones est faite en couches successives (fig. 2.8). L'information circule des entrées vers les sorties sans jamais retourner en arrière.

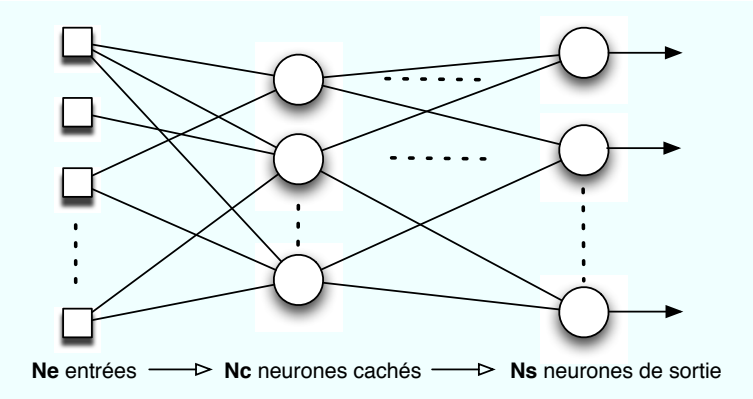


Figure 2.8 – Réseau de neurone non récurrent

Le perceptron multicouches (ou MLP pour Multi-Layer Perceptron) est un des réseaux de neurones non récurrents les plus simples. Il possède une couche d'entrée, une couche de sortie et une ou plusieurs couches cachées. Leurs caractéristiques est d'avoir des neurones cachés avec une fonction d'activation de type sigmoïde. Sixsmith et Johnson [107] ont par exemple utilisé un perceptron multicouches pour détecter des chutes avec un capteur infrarouge en se basant sur la vitesse verticale 2D de la personne. Leur perceptron multicouches était entrainé avec des scénarios prédéfinis effectués par une actrice couvrant de nombreux types de chutes, vues de plusieurs points de vue. Au final, un total de 10000 vecteurs classifiés en chute ou non chute ont été utilisés pour l'entrainement. Seulement un tiers des chutes sont détectées, et ils expliquent ces résultats par une instabilité de l'estimation de la vitesse verticale de la personne, et par le fait que les données d'entraînement n'étaient pas suffisament représentatives.

On peut aussi citer les réseaux RBF (Radial Basis Function) qui sont des réseaux à une couche cachée qui utilisent des fonctions radiales. Les réseaux multicouches à retard (TDNN, Time Delay Neural Network) sont aussi des réseaux de neurones non récurrents qui ont été beaucoup utilisés en reconnaissance de la parole et en reconnaissance de gestes, car ils prennent en compte la dimension temporelle des

données. Par exemple, Yang et Ahuja [133] utilisent un TDNN pour reconnaître des gestes du langage des signes Américain. Le TDNN est particulièrement approprié car un geste est une séquence spatio-temporelle de vecteurs de caractéristiques définis tout le long de la trajectoire.

- *Réseaux de neurones récurrents*

  La différence des réseaux de neurones récurrents (feedback networks) par rapport aux précédents est qu'ils autorisent le retour arrière et les connexions latérales au sein d'une même couche (fig.2.9).

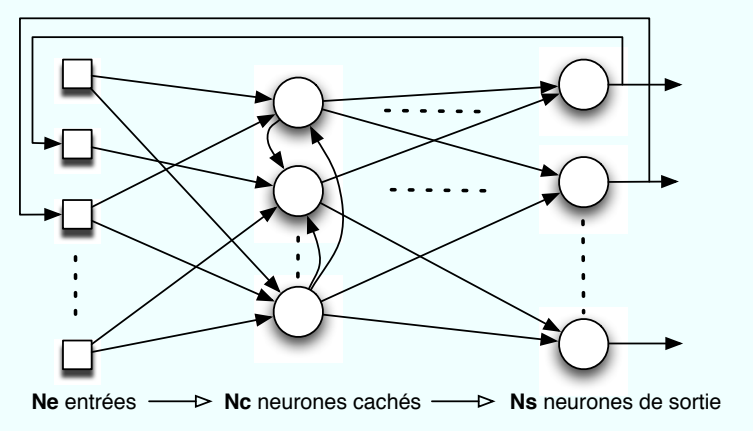


Figure 2.9 – Réseau de neurone récurrent

Les réseaux de Hopfield sont un exemple de réseaux récurrents. Il s'agit de réseaux entièrement connectés où les entrées et les sorties sont confondues. Il existe d'autres types de réseaux récurrents tels que les réseaux ART (Adaptative Resonnance Theorie) qui sont des réseaux à apprentissage par compétition. Dû à la caractéristique de récurrence, ces réseaux peuvent être plus difficiles à mettre en oeuvre.

- *Les réseaux à apprentissage compétitif*

  Les réseaux à apprentissage compétitif cherchent à regrouper des vecteurs d'entrées similaires, de telle sorte que lorsqu'un nouveau vecteur entre dans le réseau, il puisse être classifié dans la classe la plus similaire. Les cartes topologiques font partie des réseaux compétitifs, elles sont formées d'un treillis de neurones

qui permettent de regrouper les données similaires en classes. Les cartes auto-organisatrices de Kohonen, aussi appelée SOM (Self Organizing Maps), font partie des méthodes de classification non supervisées les plus utilisées. Le principe d'une carte de Kohonen est que le vecteur en entrée est comparé à tous les neurones de la carte, et seul les poids du neurone gagnant et de son voisinage sont modifiés. Une version supervisée des cartes topologiques existe, c'est une méthode de quantification vectorielle nommée LVQ (Learning Vector Quantization).

Un exemple d'application des cartes de Kohonen est la détection de trajectoires suspectes [30, 92]. Le principe est simple : la carte de Kohonen se construit par apprentissage à partir d'un ensemble de trajectoires normales. Une trajectoire sera alors classifiée comme anormale si elle n'est pas similaire aux données d'entraînement. Le système fonctionne bien mais nécessite beaucoup de données d'entraînement. Par exemple, si une trajectoire normale ne correspond à aucune des données d'entraînement, elle risque d'être classifiée comme suspecte.

- *Autres types de classifieurs*

  Il existe d'autres types de classifieurs qui ont été utilisés dans la littérature pour la reconnaissance de trajectoires et l'analyse de comportements. Par exemple, les SVM (Support Vector Machines) dont l'idée est de rechercher l'hyperplan qui sépare le mieux des ensembles de données. Le choix du noyau du SVM va influencer la qualité du classifieur : un mauvais noyau risque de donner de mauvaises classifications si les données ne sont pas suffisamment représentatives de chaque classe. Jiao *et al.* [62] proposent un SVM avec un nouveau noyau pour détecter des trajectoires suspectes. Leurs trajectoires sont représentées par une séquence de descripteurs associés à des vitesses et des accélérations.

2.6.3 Le cas de la reconnaissance de chutes

Avec la détection de chutes, nous sommes dans le cas bien particulier des classifieurs à une classe. En effet, les données normales sont nombreuses (personne qui marche, s'assoit, s'accroupit, fait son ménage, etc), et les données anormales (chutes) sont très

peu nombreuses voir inexistantes. Certains [107] ont utilisé un acteur pour simuler des chutes comme données d'entrainement, mais une fois le système final positionné, les données d'entraînement se sont révélées insuffisantes pour avoir un bon détecteur de chutes. Les méthodes de détection de chutes sont donc basées principalement sur la détection d'un événement nouveau par rapport à des données connues.

Markou et Singh [74][75] ont classifié les méthodes de détection de nouveautés en deux classes :

Approches statistiques La plupart du temps, cette approche consiste à modéliser la distribution des données pour ensuite estimer la probabilité qu'un exemple test appartienne à cette distribution. Parmi les approches statistiques, les méthodes de classification par rapport au voisinage peuvent être utilisées, comme les k-plus proches voisins kNN[13] ou les fenêtres de Parzen. Il faut alors choisir k le nombre de voisins dans le cas des kNN ou la taille du voisinage dans le cas des fenêtres de Parzen. La distribution peut aussi être modélisée avec un mélange de gaussiennes GMM[14] dont il faut estimer les paramètres.

L'utilisation des HMM[15] peut se faire par la modélisation de chaque action normale par un HMM. Cependant, les activités normales sont si diverses qu'il n'est pas évident de modéliser toutes les activités possibles.

La qualité et la quantité des données d'entrainement des approches statistiques est importante pour bien déterminer les paramètres de la distribution lors de la phase d'entrainement.

Approches réseaux de neurones Comparé aux méthodes statistiques, les réseaux de neurones sont moins adaptés à la détection d'événement nouveau dû à leur habilité à généraliser. Ils sont aussi plus complexes à mettre en oeuvre. Les réseaux de neurones les plus utilisés pour la détection de nouveautés sont les perceptrons multicouches MLP, les SVM[16], les réseaux de Hopfield et les cartes de Kohonen. Par

[13] k-nearest neighbour
[14] Gaussian Mixture Model
[15] Hidden Markov Model
[16] Support Vector Machines

exemple, les cartes de Kohonen ont été utilisées pour la détection de trajectoires suspectes [30, 92].

2.7 La détection de chutes par vidéosurveillance

Depuis quelques années, de plus en plus de travaux de recherche utilisent la vidéosurveillance pour la détection de chutes. Elle apporte en effet beaucoup d'attraits, notamment parce qu'elle ne nécéssite pas le port d'un quelconque instrument de mesure. De plus, une caméra apporte énormément d'information sur la personne mais aussi sur son environnement. On peut connaître sa localisation dans une pièce, son mouvement et ses actions. Ainsi, en plus de détecter des chutes, un système de vidéosurveillance pourrait permettre d'analyser le comportement de la personne, à savoir si elle prend correctement ses médicaments, si elle mange et dort à des heures réguliéres, etc. Toutes ces données sont des indicateurs du bien être de la personne et peuvent révéler des problèmes qui néssecitent un suivi.

2.7.1 Systèmes monoculaires de détection de chutes

Certains [5, 116] ont utilisé le ratio de la boite englobante de la silhouette de la personne dans l'image pour détecter des chutes. Cette méthode très simple, ne fonctionne toutefois que si la caméra est placée de côté et que si le champ de la caméra n'est pas occulté par des objets. Pour éviter les problèmes d'occultations, d'autres chercheurs [70, 81] ont placé la caméra au plafond. Lee et Mihailidis [70] détectent les chutes en analysant la silhouette de la personne et sa vitesse 2D dans l'image, avec des seuils spéciaux pour les zones d'inactivités habituelles telles que le lit, un canapé ou un fauteuil. Nait-Charif et McKenna [81] utilisent aussi des zones d'inactivités connues. Dans leur travaux, la personne est suivie à l'aide d'une ellipse et la trajectoire résultante est utilisée pour détecter une inactivité inhabituelle en dehors des zones normales d'inactivités. Cependant, avec une caméra au plafond, ils se privent d'une information importante pour la détection de chute, la vitesse verticale de la personne, et ne sont capables de couvrir que des champs restreints.

Pour avoir un champ de vue plus large, la caméra est préférablement monté dans un coin de la pièce en hauteur. Sixsmith et Johnson [107] détectent les chutes à partir de la vitesse verticale 2D de la personne. Cependant, la vitesse 2D est plus grande dans l'image quand la personne est proche de la caméra que quand elle en est éloignée, ce qui rend difficile le choix d'un seuil de détection pour discriminer une chute d'une personne qui s'assoit brutalement dans son fauteuil. Il est plus intéressant de calculer des vitesses 3D, comme proposé par Wu [131] dans une étude biomédicale avec des marqueurs sur la personne. En se basant sur cette idée, nous avons proposé de calculer ces vitesses 3D, sans utiliser de marqueurs, à l'aide de la trajectoire 3D de la tête de la personne [98]. Dans cette étude préliminaire, la tête de la personne était détectée et suivie par une ellipse dans l'image à l'aide de filtres à particules. Connaissant les paramètres de la caméra, les proportions 3D de la tête et sa projection dans l'image (ellipse), il est possible de calculer la trajectoire 3D de la tête avec une seule caméra calibrée. Les vitesses verticale V_v et horizontale V_h de la tête étaient alors calculées pour détecter des chutes. Le chapitre 5 présente une version améliorée de cette méthode en prennant le raisonnement d'une manière différente : au lieu de suivre une ellipse 2D, nous suivons une ellipsoide 3D représentant la tête à l'aide d'un filtre à particules qui est projetée en une ellipse 2D dans le plan image.

2.7.2 Systèmes multi-caméras de détection de chutes

Avec un système multi-caméras, Thome et Miguet [115] utilisent un modèle de Markov caché (HMM) à plusieurs couches pour distinguer des chutes par rapport à des activités de marche. Des caractéristiques pour l'analyse du mouvement sont extraites par une rectification métrique d'images dans chaque vue. Avec deux caméras non calibrées, Hazelhoff *et al.* [50] détectent des chutes en se basant sur l'analyse en composante principale de la forme de la silhouette, notamment la direction de son axe principal et la variance de son ratio. Un module de suivi de tête est utilisé pour réduire les fausses alarmes. Anderson *et al.* [6] reconstruisent un volume 3D de la personne à l'aide de voxels (volumetric pixel) à partir de 2 caméras. La détection de chutes est faite par une analyse du volume 3D à l'aide d'une hierarchie floue. Un volume 3D de la personne

est aussi reconstruit par Auvinet *et al.* [8] à partir de plusieurs caméras. La chute est détectée quand une grande partie du volume est concentrée près du sol. Les systèmes multi-caméras fournissent une information 3D fiable, mais sont aussi plus compliqués à mettre en oeuvre car ils ont besoin d'être calibrés et synchronisés. Ils sont aussi plus chers car il est nécessaire d'avoir plusieurs caméras par pièce et suffisamment de ressources informatiques pour traiter tous les flux vidéos.

2.8 Notre système de détection de chutes

Notre système de vidéosurveillance est du type « caméra intelligente », capable de détecter une situation anormale automatiquement, et de contacter la famille ou les secours en cas de problème. Pour limiter le coût du système nous avons utilisé des caméras IP (caméras réseaux) et limité leur nombre à une seule caméra par pièce.

Concrètement, les caméras IP équipant l'appartement seront reliées à un routeur. Une plateforme de traitement d'images sera aussi reliée au réseau pour traiter localement les images. Si une situation anormale est détectée, un message pourra alors être envoyé suivant l'urgence aux secours ou à la famille via une connexion internet sécurisée. La figure 2.10 donne un aperçu du système envisagé.

Notre système est d'abord orienté pour la vision de jour, mais nous pourrions envisager une application fonctionnant en vision de nuit avec l'utilisation de caméras IP équipées d'un éclairage infrarouge.

Comme nous avons décidé de nous limiter à une seule caméra par pièce, nous allons donc nous intérresser aux méthodes monoculaires pour détecter des chutes. Par ailleurs, nous avons simplifié le problème en considérant que la personne est seule dans son appartement, en supposant que si une autre personne est présente, elle peut prévenir les secours en cas de problème.

Pour nos méthodes de détection de chutes, nous avons exploré deux voies : la première consistant à se limiter à une information 2D (image) pour détecter une chute, la deuxième en essayant d'inférer une information 3D à partir d'une seule caméra.

Le Chapitre 3 présente notre première contribution avec un article sur notre méthode

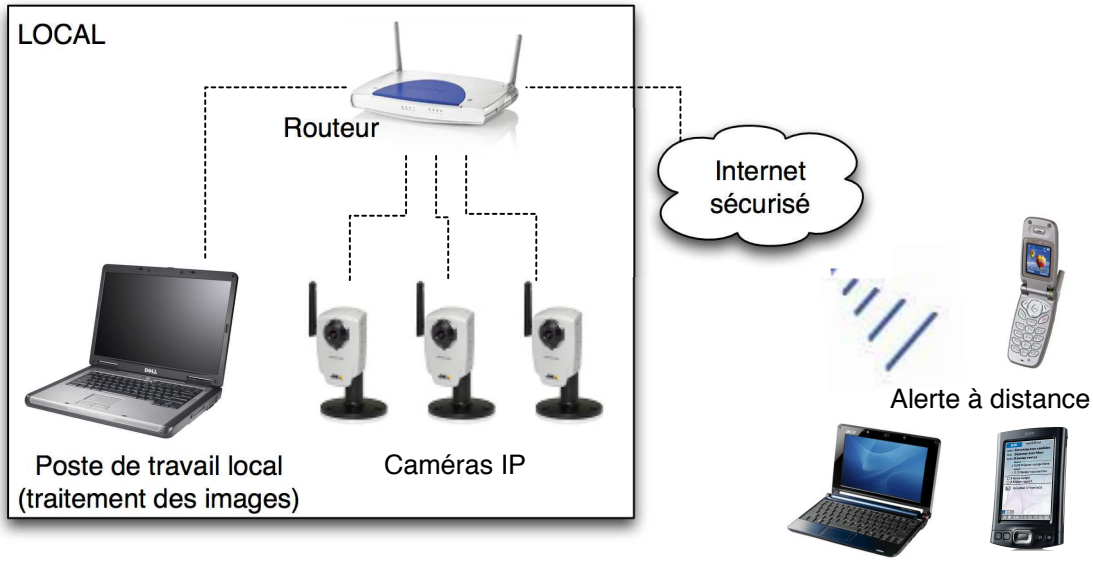


Figure 2.10 – Le système de vidéosurveillance

de détection de chutes basée sur l'analyse de la déformation de la silhouette de la personne. Après un rappel des techniques pour récupérer une information 3D à partir d'un système monoculaire dans le Chapitre 4, nous présentons notre deuxième contribution avec un article sur l'extraction d'une trajectoire 3D de la personne à partir d'un système monoculaire dans le Chapitre 5.

CHAPITRE 3

OCCLUSION ROBUST VIDEO SURVEILLANCE FOR FALL DETECTION (ARTICLE)

Ce chapitre présente le manuscript intitulé *« Occlusion Robust Video Surveillance for Fall Detection »* soumis au journal *IEEE Transactions on Circuits and Systems for Video Technology*, par Caroline Rougier, Jean Meunier, Alain St-Arnaud et Jacqueline Rousseau (2009).

3.1 Avant-propos

En premier lieu, nous nous sommes intéressés aux méthodes 2D (image) pour la détection de chutes. Plusieurs travaux [5, 70, 81, 99, 116] s'intéressent à la forme de la silhouette pour détecter des chutes. Cependant, les méthodes actuelles sont souvent dépendantes du point de vue de la caméra ou des occlusions par des objets de la scène.

Nous avons donc développé une nouvelle méthode basée sur la déformation de la silhouette lors d'une chute. Notre méthode consiste à apparier les parties similaires de la silhouette d'une personne entre deux images consécutives. Ceci nous permet d'être robuste aux occlusions en nous intéressant seulement aux parties non occultées pour quantifier la déformation. Pour la détection de chutes, nous utilisons un mélange de gaussienne (GMM) pour modéliser la distribution des données d'activités normales d'une personne âgée. Une chute est alors détectée comme étant un événement anormal par rapport à la distribution des données. Notre méthode de détection de chutes a été évaluée à l'aide d'une analyse ROC (Receiver Operating Characteristic) détaillée dans l'annexe III. Notre méthode a été testée avec succès sur des vidéos réalistes de chutes et d'activités normales comprenant de nombreuses difficultés en traitement d'images (forte compression, ombres, objets en mouvement, etc) et différents points de vue d'une même action.

Les notations utilisées dans cet article sont internes à l'article et n'ont pas de lien

avec les notations utilisées dans le reste de la thèse.

3.2 Abstract

Facing the growing population of seniors, developed countries need to establish new healthcare systems to ensure the safety of elderly people at home. Computer vision provides a promising solution to analyze people behavior and detect some unusual events like falls. In this paper, a new method is proposed to detect falls by analyzing the human shape deformation during the video sequence. A shape matching technique is used to track the elderly silhouette along the video sequence. The shape deformation is then quantified from these silhouettes based on shape analysis methods. Finally, falls are detected from normal activities using a Gaussian Mixture Models (GMM). This study has been conducted on a realistic data set of daily activities and simulated falls, and gives very good results (down to 0% error with a multi-camera setup) compared with other common image processing methods.

Keywords : fall detection, video surveillance, shape context, Procrustes shape analysis, Gaussian Mixture Model, GMM, novelty detection.

3.3 Introduction

According to the Public Health Agency of Canada [89], one Canadian out of eight was older than 65 years old in 2001. In 2026, this proportion will be one out of five. Similar figures are observed in other industrialized countries. Faced with the growing population of seniors, developed countries need to establish new healthcare systems to ensure the safety of elderly people at home. Indeed, a majority of seniors, 93%, reside in private homes, and of these, 29% live alone [89]. Moreover, falls are one of the major risks for old people living alone, often causing severe injuries. The gravity of the situation can increase if the person cannot call for help, being unconscious or immobilized.

Most fall detection techniques are based on accelerometers [57, 64, 88] or help buttons [36]. But the major problem with these types of technology is that older people often forget to wear them, and in the case of a help button, it is useless if the person is

unconscious after the fall. Moreover, batteries are needed for these devices and must be replaced or recharged regularly for adequate functioning. Floor vibration-based detectors could be a promising solution but they depend upon the floor dynamics and are still in their infancy [4].

Recently, the emergence of computer vision systems has allowed us to overcome these problems. The main advantage of computer vision systems is that the person doesn't need to wear any special device. Moreover, a camera provides a vast amount of information on the person and his/her environment. For example, we can extract information on the location, the motion or the actions of the person. Thus, we can imagine a computer vision system providing information on falls, but also, checking other daily behaviors like medication intake, or meal/sleep time and duration. Typically, these systems would be powered by conventional electrical wall outlets with possibly a back-up power supply (battery pack). The reader can find a good study on fall detection techniques in a recent article by Noury *et al.* [85].

3.4 Related Works in Computer Vision

3.4.1 Monocular systems

Among fall detection methods, one of the simplest and commonly used techniques is to analyze the bounding box representing the person [5, 116] in the image. However, this method is efficient only if the camera is placed sideways, and can fail because of occluding objects. For more realistic situations, the camera has to be placed higher in the room to avoid occluding objects and to have a larger field of view.

Lee and Mihailidis [70] detected falls by analyzing the silhouette and the 2D velocity of the person, with special thresholds for inactivity zones like the bed. Nait-Charif and McKenna [81] track the person using an ellipse, and analyze the resulting trajectory to detect inactivity outside the normal zones of inactivity like chairs or sofas. However, they both used a camera mounted on the ceiling and therefore did not have access to the vertical motion of the body, which provides useful information for fall detection.

The 2D (image) velocity of the person has also been used to detect falls [70, 107].

However, a problem with the 2D velocity is that it is higher when the person is near the camera, so that the thresholds to discriminate falls from a person sitting down abruptly, for instance, can be difficult to define.

Recently, we have shown promising preliminary results on how a fall detector could be based on simple shape analysis or head tracking [98, 99]. More elaborate shape analysis will be considered in this paper based on the person's silhouette.

3.4.2 Multi-camera systems

Other work has been done using multi-camera systems. Thome and Miguet [115] proposed to use a Layered Hidden Markov Model (LHMM) to distinguish falls from walking activities. The features used for motion analysis were extracted from a metric image rectification in each view. Anderson *et al.* [6] analyzed the states of a voxel person obtained from two cameras. Fall detection was achieved with a fuzzy hierarchy. Auvinet *et al.* [8] proposed to exploit the reconstructed 3D silhouette of an elderly person for fall detection. An alarm was triggered when most of this volume was concentrated near the floor.

An important point about multi-camera systems is that they need to be calibrated to compute a reliable 3D information. Another problem is that the video sequences of each camera need to be synchronized, which makes the system more difficult to implement than a monocular one.

Most fall detection systems are tested in a controlled environment. An attempt was made to use more realistic data sets by Hazelhoff *et al.* [50], with two uncalibrated cameras. Based on a Principal Component Analysis (PCA), falls were detected using the direction of the principal component and the variance ratio of the human silhouette. A head tracking module helped to reject false alarms. These authors obtained a 100% detection rate when large occlusions were absent, but their recognition results decreased drastically to 55% for occluded activities. In this work, we will show the performance of our method with a realistic data set containing large occlusions.

3.4.3 Our system

As seen previously, several methods exist to detect falls with good detection rates, but only a few of them take into account truly realistic data sets. The main difficulty of fall detection is not to detect falls versus walking, but rather falls versus lure activities like sitting down brutally or crouching down. Moreover, the video data set should contain occlusions, object carrying, change of clothes and different viewpoints which are well-known sources of problems in computer vision.

In this paper, we present a new robust method to detect falls from a realistic data set. The main characteristic of our method is the analysis of shape deformation through a video sequence assuming that falls will increase the shape deformation with time. Our method can work with only one uncalibrated camera, but we also tested an uncalibrated multi-camera system using an ensemble classifier to improve our results.

3.5 Data set

In this section, we first introduce the data set to illustrate the main difficulties of realistic video sequences. To better test our system, each action was taken from several view points. Figure 3.1 shows the configuration of the cameras in the room.

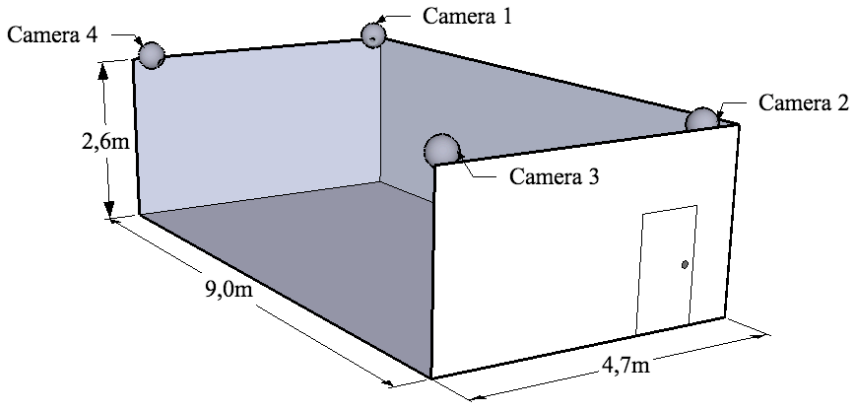


Figure 3.1 – Camera configuration.

Our data set was composed of :

- **Daily normal activities**
  - With no difficulties (walking in different directions).
  - With some little difficulties (housekeeping, small occlusions).
  - With some big difficulties (moving objects, large occlusions)
  - With characteristics similar to falls (sitting down/standing up, crouching down)
- **Simulated falls**
  - Forward falls, backward falls, falls when inappropriately sitting down, loss of balance. Falls were done in different directions with respect to the camera point of view. Notice that a mattress was used to protect the person during the simulated falls.

Figure 3.2 shows the proportion for each type of event in the dataset. We have a total of 75 different events for a total duration of more than 12 minutes for each camera. Some examples are shown in Fig. 3.3.

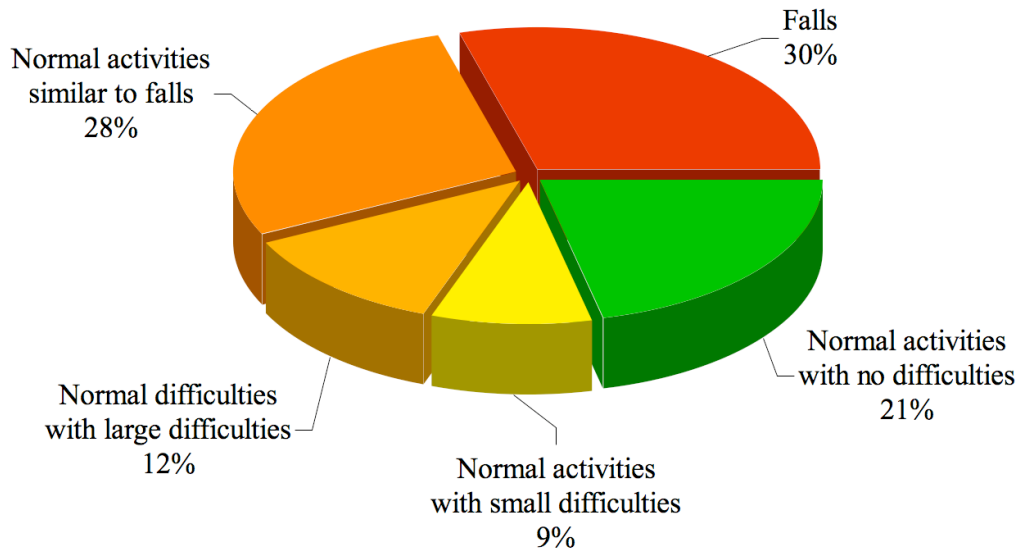
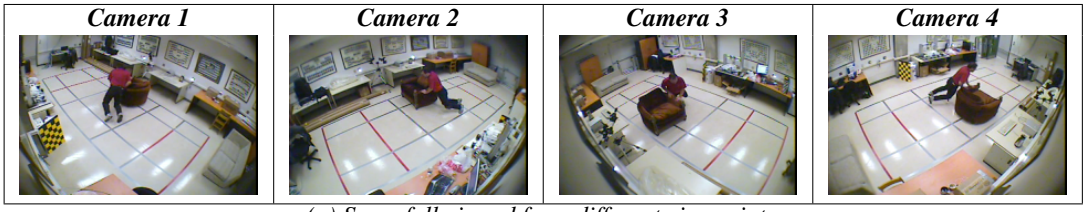
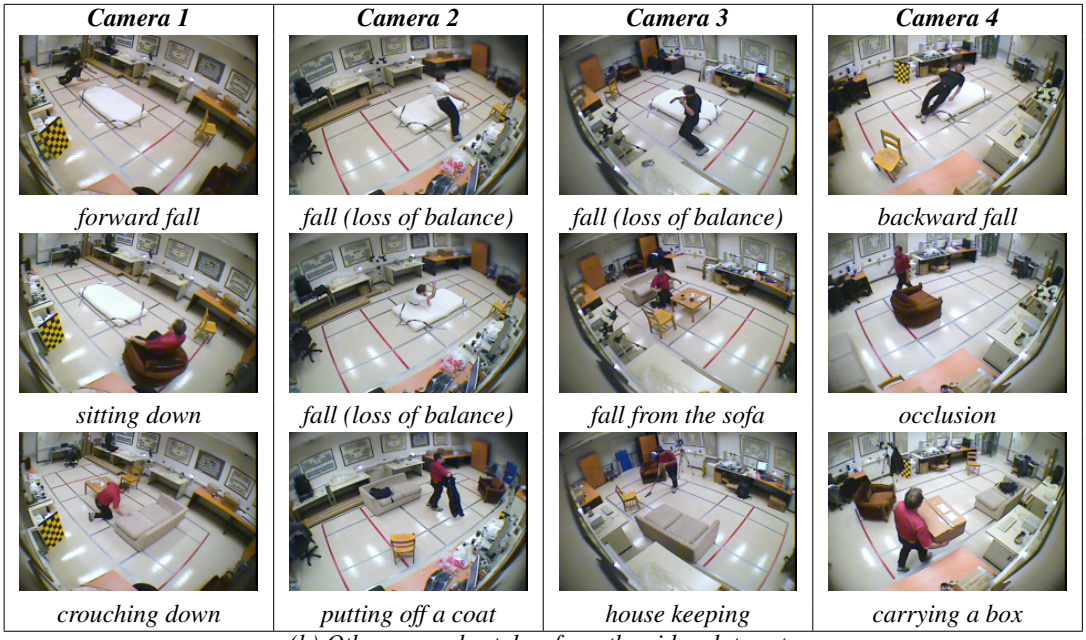


Figure 3.2 – Proportion of each type of event.



(a) Same fall viewed from different viewpoints



(b) Other examples taken from the video data set

Figure 3.3 – The first line (a) shows a fall, slowed down by grabbing an armchair, from different viewpoints. The other lines (b) show other examples of falls, lures and data set difficulties.

We wish our final system to be low-cost, so our video sequences were acquired using inexpensive IP cameras (Gadspot gs-4600 [44]) with a wide angle to cover all the room. The acquisition frame rate was 30 fps and the image size was 720x480 pixels.

Our video sequences contained typical difficulties which can lead to segmentation errors like :

– **High video compression** (MPEG4) which can give artifacts in the image.

– **Shadows** and **reflections** which can be detected as moving objects during a seg-

mentation process.

- **Cluttered and textured background**.
- **Variable illumination** which must be taken into account during the background updating process.
- **Carried objects** (bags, clothes, etc) which must also be taken into account during the background updating process.
- **Occlusions** (chairs, sofa, etc).
- **Entering/leaving** the field of view.
- **Different clothes** with different color and texture. Putting on and taking off a coat.

3.6 Falls Characteristics

To better design our system, we must first understand how to detect a fall. According to Noury *et al.* [85], automatic methods for fall detection are based on the detection of :

- **Lack of movement :** usually, the person will remain immobile on the ground after a serious fall, at least for some time.
- **A lying position :** according to the authors, this method is prone to many « false positives ». For example, if the person sleeps outside the bedroom at irregular hours.
- **A person lying on the ground :** this method is better than the previous one unless the fall doesn't end on the ground.
- **Vertical speed :** an appropriate threshold could allow us to distinguish falls from normal activities (sitting down, crouching down, etc).
- **An impact shock :** easily detectable with an accelerometer or vibration detector, but more difficult with a computer vision system.

We can add another characteristic which, unlike other sensors, can be quantified using a camera :

- **body shape change :** Indeed, the human shape will progressively and slowly change during usual activities, while during a fall, it will change drastically and rapidly.

Currently, two main technologies are used to detect falls : wearable devices [57, 64, 85, 88] and camera based devices [5, 6, 8, 50, 70, 81, 98, 99, 107, 115, 116]. Table 3.I summarizes the sensor performance to detect fall characteristics with :

- Wearable devices : accelerometers and other sensors.
- 2D vision system : only one camera is required, the system doesn't need to be calibrated.
- 3D vision system : 3D information can be recovered from one calibrated camera. But for better performance, a calibrated multi-camera system is preferable.

	Wearable device	2D vision system	3D vision system
Lack of movement	++	++	++
Lying position	+	+	++
Lying on the ground	-	+	++
Vertical speed	++	+	++
Impact shock	++	-	+
Body shape change	-	++	++

Tableau 3.I – Comparison between wearable devices and vision systems

Lack of movement is easily detectable with all sensors. With a 3D vision system, it is possible to localize precisely a person relatively to his/her environment. Therefore, a 3D vision system can easily detect the characteristics of *lying position* and *lying position on the ground*. This is more difficult with a 2D vision system since we don't have any accurate 3D information without calibration. Wearable sensors do not give any information about the person's position relative to the ground. However, it is possible to have the body orientation using a 3D gyroscope which could detect a *lying position*. For a more precise localization, such as *lying position on the ground*, it is necessary to couple these wearable sensors with floor sensors. The *vertical speed* depends on the camera point of view with 2D vision system, but is easily measurable with wearable devices and 3D vision systems. Detecting the *impact shock* with vision systems is not easy since the frame

rate is usually not sufficiently high. It could be possible to detect some changes in the person's acceleration, but this change is generally not sufficiently accurate to distinguish a fall from a person sitting down. On the other hand, vision systems are outstanding for the analysis of the actions of the person and to quantify *body shape change*.

Based on these observations, we chose to combine two fall characteristics to increase the robustness of our system : human shape deformation during the fall, followed by a lack of movement just after the fall. These choices are justified because the body shape change includes somehow the information produced by the vertical speed and impact shock characteristics. Furthermore, the two *lying* characteristics are not sufficiently robust while the *lack of movement* feature is easy to compute and adds robustness to the *body shape change*.

While 3D vision systems are better than 2D systems, for some of the features listed in Table 3.I, they are not significantly superior for *body shape change* and *lack of movement*. Furthermore, 3D vision systems are more difficult to implement and they need to be calibrated. Thus, we develop here a 2D vision system with only one or with a few uncalibrated cameras.

3.7 Method Overview

A fall is characterized by a large movement and some changes in the human shape. More precisely, during usual activities, the human shape will change progressively and (relatively) slowly, while during a fall, the human shape will change drastically and rapidly. Thus, we chose to detect falls during the video sequence by quantifying human shape deformation.

The main steps of our system are :

1. **Silhouette edge points extraction**

   For body shape change analysis, we need to compare two consecutive silhouettes of a person. As landmarks, we chose to extract some edge points from the silhouette of the person. The silhouette is obtained by a foreground segmentation method, and some edge points are extracted from the silhouette by a Canny edge

detector. This step is described in section 3.8.

2. **Matching with Shape Context**

   The silhouette edge points are then matched through the video sequence. The shape matching is useful to track and to quantify the silhouette deformation. For this purpose, we use the shape context matching method [11] described in section 3.9.

3. **Shape analysis**

   For body shape change analysis, we chose to compare two deformation measures :

   (a) The mean matching cost obtained from the Shape Context matching which has been used for shape recognition [11].

   (b) The full Procrustes distance [38] which is a well-known tool for shape analysis, and which has been widely used to compare shapes in biology and medicine.

   The shape analysis step is described in section 3.10.

4. **Fall detection using GMM**

   Finally, we use a Gaussian Mixture Model (GMM) to classify the different activities as a fall or not, based on shape deformation during the fall followed by a lack of movement after the fall. This step is described in section 3.11.

3.8 Silhouette Edge Point Extraction

Usually, the whole silhouette of the person is used for shape analysis [63, 126]. The silhouette is extracted by a background subtraction method which consists of comparing the current image with an updated background image. We chose the method described by Kim *et al.* [67] which takes into account shadows, highlights and high image compression. In addition to the foreground silhouette contour, we chose to extract edge points inside the silhouette using a Canny edge detector [18] to provide additional shape information. An example of edge point extraction is shown in Fig. 3.4.

The moving edge points will be used to match two consecutive human shapes. Since we don't need as many points, we select N landmarks regularly-spaced for each sil-

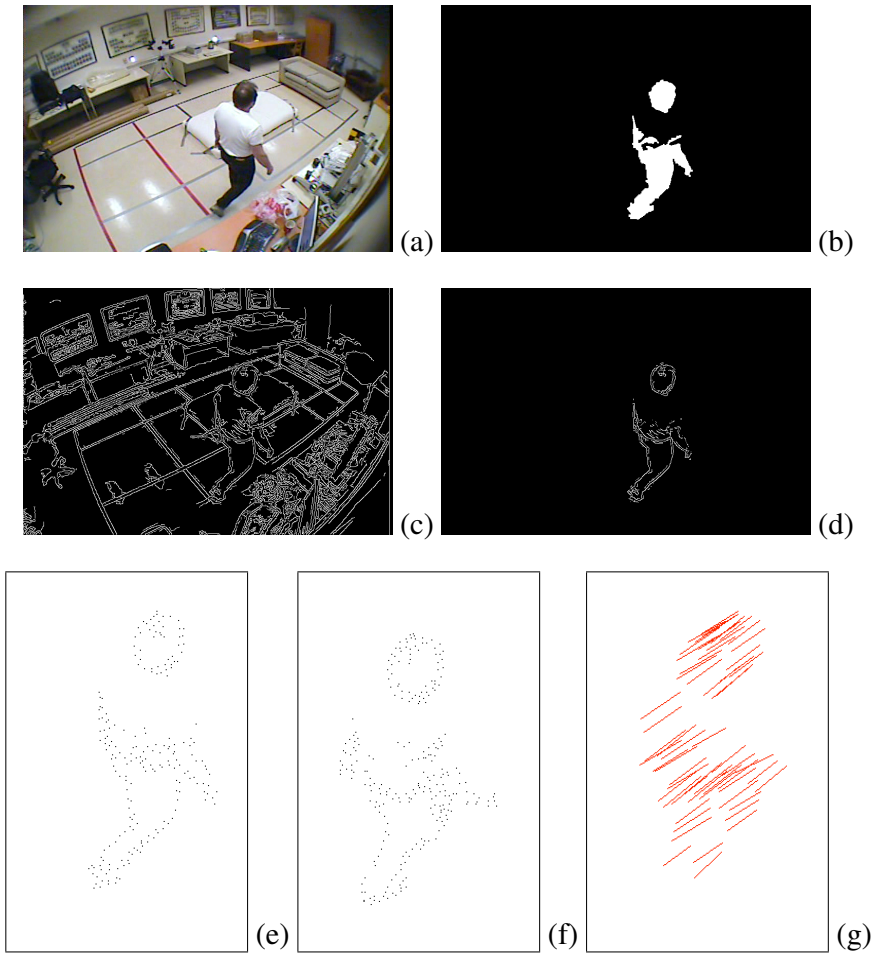


Figure 3.4 – From the original image (a), the foreground silhouette is extracted (b). Due to compression, occlusions and segmentation problems, this silhouette is not clean enough to be used for shape analysis. By combining the moving silhouette (b) with the Canny edge image (c), we obtain the moving edge points (d) which are used to choose a set of selected edge points (e). By matching with the previous selected edge points (f), we obtain the matching points (g).

houette. We used $N = 250$ landmarks for our experiment. The increment to select regularly the landmarks is the same for the two shapes. If n_{i-1} and n_i are respectively the total number of edge points from the previous and the current silhouette, then the increment is set to :

$$inc = \frac{\max(n_{i-1}, n_i)}{N} \tag{3.1}$$

An example of moving edge points and selected edge points is also shown in Fig. 3.4.

3.9 Matching using Shape Context

The moving edge points extracted from two consecutive images are then matched using Shape Context [11]. Shape Context is a shape descriptor that encodes local information about each point relative to its neighbours. An algorithm for Shape Context matching with edge points has been proposed by Mori and Malik [79]. Unlike them, however, we discard unnecessary background edge points with the background subtraction segmentation. As Shape Context is sensitive to background edges, we improve it by considering only moving edge points for cluttered scenes.

The shape matching process consists of finding for each point p_i of the first shape, the best corresponding point q_j of the second shape.

For each point p_i on the shape, we compute a log-polar histogram h_i of the relative coordinates of the remaining $n-1$ points :

$$h_i(k) = \#\{q \neq p_i : (q - p_i) \in bin(k)\} \tag{3.2}$$

The log-polar histogram is obtained by positionning the polar coordinate system on each edge landmark p_i as shown in Fig. 3.5.

Then, to find similar points on the two shapes, we compute a matching cost $C_{ij} = C\left(p_i, q_j\right)$ for each pair of points $\left(p_i, q_j\right)$. This matching cost is computed with the χ^2 statistic :

$$C_{ij} = \frac{1}{2}\sum_{k=1}^{K} \frac{\left[h_i(k) - h_j(k)\right]^2}{h_i(k) + h_j(k)} \tag{3.3}$$

where $h_i(k)$ and $h_j(k)$ denote the K-bin histograms respectively for p_i and q_j.

Given the set of costs C_{ij} between all pairs of points, the best corresponding points are obtained by minimizing the total matching cost $H(\pi)$ given a permutation $\pi(i)$:

$$H(\pi) = \sum_i C\left(p_i, q_{\pi(i)}\right) \tag{3.4}$$

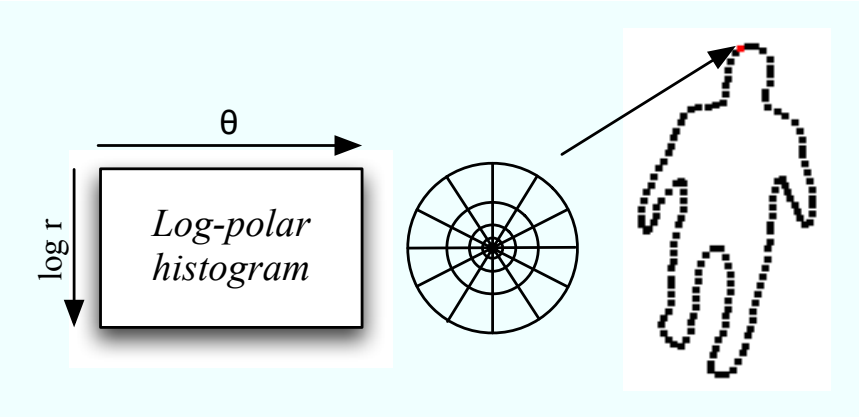


Figure 3.5 – Log-polar histogram computation for a point p_i. The log-polar histogram has 5 bins for $\log r$ and 12 bins for θ as proposed by the authors in [11].

The authors in [11] proposed to use the Hungarian algorithm [69] for bipartite matching. The input of this algorithm is a square cost matrix with entries C_{ij}, and the result corresponds to the permutation $\pi(i)$ minimizing $H(\pi)$. As we can have some bad landmarks in our selected edge points (due to segmentation errors and/or partial occlusions), we need to add some dummy points or outliers in the matching process. However, the number of these dummy points is not easy to choose, especially in the case of severe occlusion where some bad landmarks can still remain.

In our implementation, we propose to match only the most reliable points by finding those that have their cost minimal for the row and the column of the matrix $\left(\min_i C_{ij} = \min_j C_{ij}\right)$.

To quantify the shape deformation, we need reliable landmarks, so we also clean the set of matching points based on the motion of the person. The mean motion vector $\bar{v}$ and the standard deviation σ_v are computed with the set of matching points. We keep the vectors within 1.28 standard deviations from the mean which corresponds to 80% of the motion vectors.

An example of Shape Context matching is shown in Fig. 3.4. With the Hungarian algorithm, some bad matching points can appear in spite of the inclusion of dummy points. With our method, only reliable landmarks are kept which is important in the quantification of the shape deformation. Another advantage is that the computational

time is reduced with our method compared to the Hungarian algorithm.

3.10 Shape Analysis

3.10.1 Mean matching cost

The *mean matching cost* $\bar{C}$ of the best corresponding points is obtained during the shape matching step (see Section 3.9) and we propose here to use it to quantify abnormal shape deformation. In fact, the mean matching cost should be high during the fall and low just after the fall.

3.10.2 Full Procrustes distance

Procrustes analysis [38] is a well-known tool for shape analysis, which has been widely used to compare shapes in biology or medicine. Some researchers have also used this method for gait recognition [15, 63, 126]. We propose to use it here to detect abnormal shape deformation for fall detection.

The main characteristic of Procrustes shape analysis [38] is that the shapes are compared once translational, rotational and scaling components are removed to normalize the shapes. Concretely, two sets of landmarks are obtained from the shapes to be compared. Then, the full Procrustes distance (defined below) can be computed to quantify the deformation between the two shapes. This distance will be high in case of a fall.

Each human shape is represented by k landmarks and can be described by a k dimensional complex vector Z :

$$Z = [z_1, z_2, \ldots, z_i, \ldots, z_k], \quad z_i = x_i + jy_i \tag{3.5}$$

The centered landmarks Z_C are obtained by multiplying the coordinates Z with the centring matrix C :

$$Z_C = CZ \qquad with \;\; C = I_k - \frac{1}{k} 1_k 1_k^T \tag{3.6}$$

where I_k is a $k \times k$ identity matrix and 1_k is a k dimensional vector of ones.

Consider now two centered configurations $v = (v_1, \cdots, v_k)$ and $w = (w_1, \cdots, w_k)$. A suitable distance between them can be obtained by choosing the best registration of v and w with a similarity transformation and then by computing the remaining distance between these complex vectors. For two centered complex configurations, this full Procrustes distance is simply [38] :

$$D_f(v,w) = \left\{ 1 - \frac{|v^* w|^2}{\|v\|^2 \|w\|^2} \right\}^{1/2} \tag{3.7}$$

The *full Procrustes distance* D_f is computed between the matching points of two consecutive images. D_f should increase in case of a fall, and should be low after the fall.

3.11 Fall Detection using GMM

The fall detection problem can be seen as an outlier detection problem. So the fall recognition system needs to be a one-class classifier, which is trained with normal activities with the aim of detecting abnormal events like falls. A survey of outlier detection methods can be found in the article [53]. For our experiment, we model our normal activity data with a Gaussian Mixture Model (GMM).

3.11.1 Gaussian Mixture Model (GMM)

A Gaussian Mixture Model [80] can be defined by a weighted sum of Gaussian distributions :

$$p(x) = \sum_{j=1}^{M} P(j)\, p(x \mid j) \tag{3.8}$$

where M is the number of components in the mixture and $P(j)$ are the mixing coefficients. The jth Gaussian probability density function $p(x \mid j)$ has the form :

$$p(x \mid j) = \frac{1}{\left(2\pi \prod_{i=1}^{d} \sigma_{j,i}^2\right)^{d/2}} \exp\left\{ -\sum_{i=1}^{d} \frac{\left(x_i - \mu_{j,i}\right)^2}{2\sigma_{j,i}^2} \right\} \tag{3.9}$$

where d is the dimensionality of the input space.

The parameters to be estimated are the mixing coefficients $P(j)$, the mean vector μ of dimension d and the diagonal covariance matrix $\sum_j = diag\left(\sigma_{j,1}^2, \ldots, \sigma_{j,d}^2\right)$. The parameters are determined using the EM (Expectation-Maximisation) algorithm by maximizing the data likelihood. Specifically in our case, the parameters of the GMM are estimated from a training data set of normal daily activities (walking, sitting down, crouching down, housekeeping, etc).

3.11.2 Leave-One-Out Cross-Validation

A leave-one-out cross-validation is used to train and test our dataset. The dataset is divided into N video sequences which contain some falls and/or normal activities (including lures). For testing, one sequence is removed from the dataset, and the training is done using the $N-1$ remaining sequences (where falls are deleted because the training is only done with normal activities). This sequence is then classified with the resulting GMM. This is repeated N times by removing each sequence in turn. By counting the number of errors, classification error rate and other measurements can be computed (see Section 3.11.4).

3.11.3 GMM Features

A fall is characterized by a peak on the smoothed *full Procrustes distance* curve or *mean matching cost* curve followed by a lack of movement of the person just after the fall, as shown in Fig. 5.7.

We therefore consider two features (F_1, F_2) for the GMM classification representing these characteristics :

- F_1 *representing the fall*
  This corresponds to $D_f(t)$ or $\bar{C}(t)$ at time t. F_1 will be high in case of a fall.
- F_2 *representing the lack of movement after the fall*
  This corresponds to the mean value of D_f or $\bar{C}$ just after the fall, between $t+1s$ and $t+5s$. This time interval was determined experimentally.

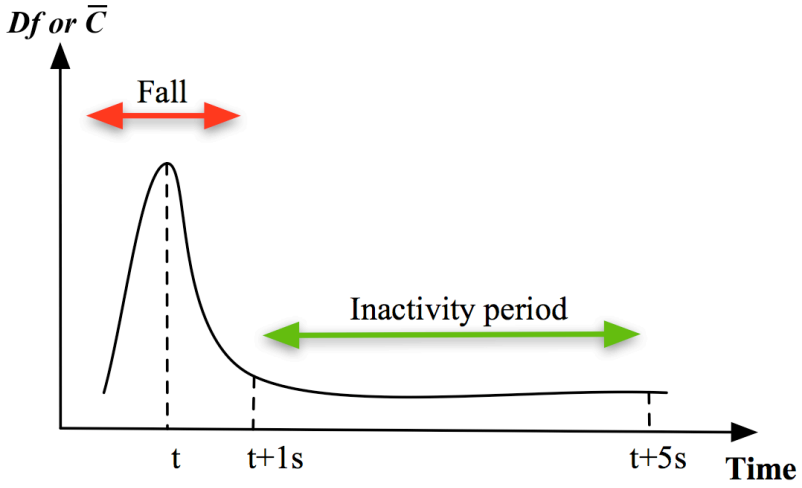


Figure 3.6 – Fall features based on the full Procrustes distance D_f or mean matching cost $\bar{C}$.

3.11.4 GMM Analysis

To analyze our recognition results, we compute the sensitivity, the specificity, the accuracy and the error rate obtained with our GMM classifier as follows :

- True Positives (TP) : number of falls correctly detected.
- False Negatives (FN) : number of falls not detected.
- False Positives (FP) : number of normal activities (including lures) detected as a fall.
- True Negatives (TN) : number of normal activities (including lures) not detected as a fall.
- Sensitivity : $Se = TP/(TP+FN)$
- Specificity : $Sp = TN/(TN+FP)$
- Accuracy : $Ac = \frac{(TP+TN)}{(TP+TN+FP+FN)}$
- Classification error rate : $Er = \frac{(FN+FP)}{(TP+TN+FP+FN)}$

A good fall detection system must have a high sensitivity, which means that a majority of falls are detected, and a high specificity, which means that normal activities and lures are not detected as falls. Similarly, the accuracy must be high while the error rate

must be low.

3.12 Experimental Results

Our method works with a single uncalibrated camera. The shape matching is implemented in C++ using the OpenCV library [91] and the fall detection step is done with Matlab® using the Netlab toolbox [80] to perform the GMM classification.

The original video sequences were acquired with a frame rate of 30 fps, but 5 fps was sufficient to detect a fall. The computational time of the shape matching step is about 200*ms* on an Intel Core 2 Duo processor (2.4 GHz), which is adequate for our application with a frame rate of 5fps.

3.12.1 Number of GMM Components

We explored the relationship between the number of components of the GMM and the classification results.

The initialization of the EM algorithm [80] can influence the resulting GMM, so we repeated the cross-validation 10 time for each number of components and each camera. A ROC (Receiver Operating Characteristic) analysis is performed by varying the threshold, and the EER (Equal Error Rate) is computed.

Figure 3.7 shows the EER as a function of the number of GMM components. This demonstrates that one or two components for the GMM are not sufficient because they give poorer recognition results. If we take too many GMM components, the EER and its standard deviation increase, as well as the computation time. Thus, we chose to train a GMM with 3 components for our experiment, which is a good compromise between a low classification error rate, a good repeatability of the results and a reasonable computation time.

3.12.2 Classification results

Figure 3.8 shows an example of the log-likelihood obtained with a 3-component GMM with full Procrustes distance features. The input features are normalized to unit

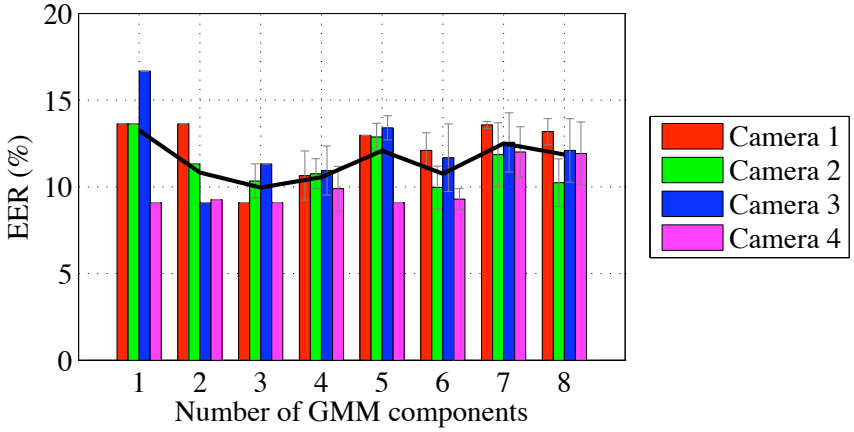


Figure 3.7 – EER as a function of number of GMM components for each camera for D_f features. The black curve represents the mean.

standard deviations and zero means. An example of a trajectory generated from a video sequence where a fall occurs is also superimposed on the graphic.

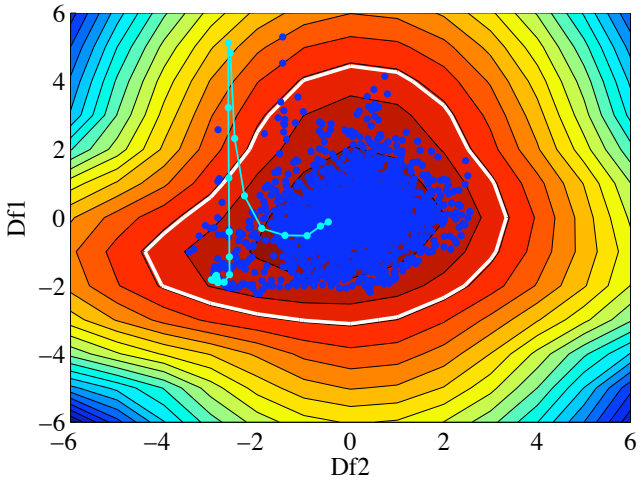
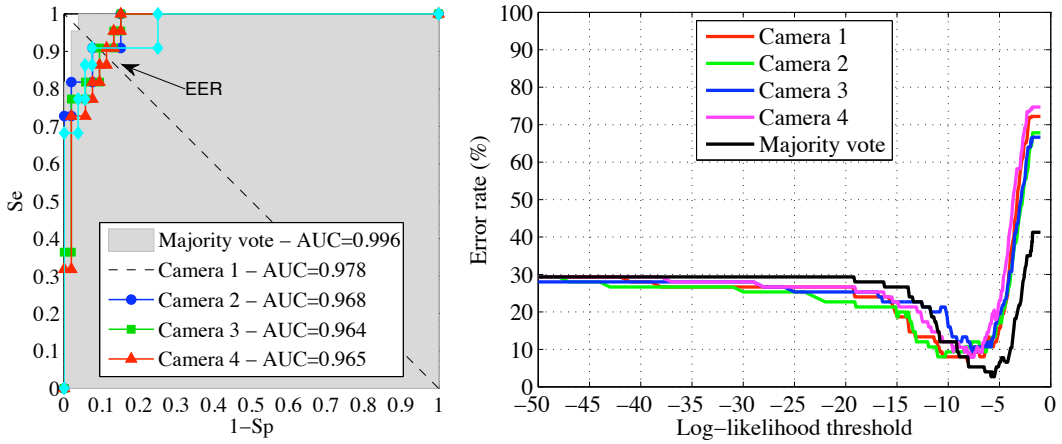


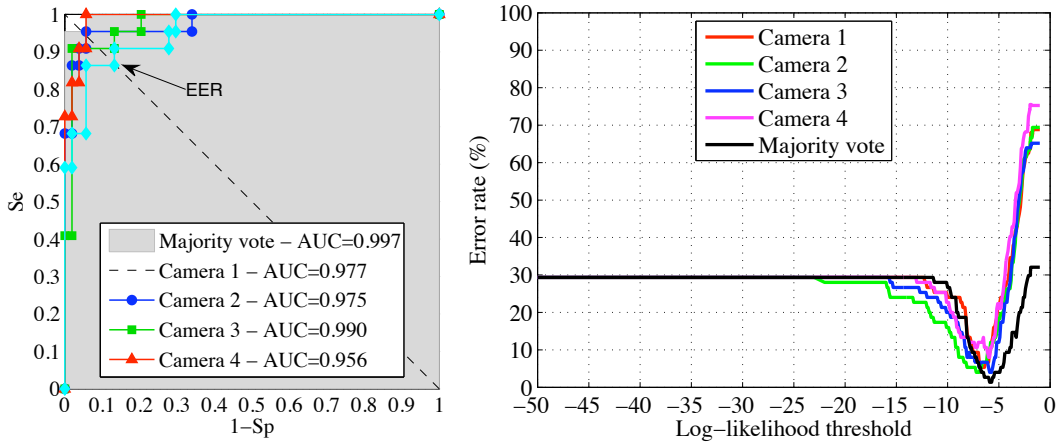
Figure 3.8 – Example of log-likelihood obtained with a 3-component GMM. The dark blue points represent the normalized training data. The light blue points correspond to a sequence where a fall occurs. The white line represents the boundary obtained for the log-likelihood threshold. The light blue points which are outside the boundary represent the detected fall.

The choice of a detection threshold depends on the sensitivity required for the sys-

tem. We perform a ROC analysis to study the influence of the GMM log-likelihood threshold. Figure 3.9 shows the ROC curves and the classification error rates obtained for each camera independently for the full Procrustes distance and mean matching cost features. They are obtained with a 3-component GMM and a log-likelihood threshold ranging from -50 to -1.



(a) Results using the full Procrustes distance features (D_{f_1}, D_{f_2})



(b) Results using the mean matching cost features $(\bar{C}_1, \bar{C}_2)$

Figure 3.9 – On the left, the ROC curves obtained for the full Procrustes distance (a) and for the mean matching cost (b). On the right, the classification error rate curves obtained by varying the GMM log-likelihood threshold. The results were computed for each camera independently and for a majority vote (at least 3 of 4 cameras).

3.12.3 Ensemble Classifier

The ROC curves show that our recognition results are good for each camera independently. But we also tried to improve our results by combining the results of all cameras. This was done with an ensemble classifier as shown in Fig. 3.10.

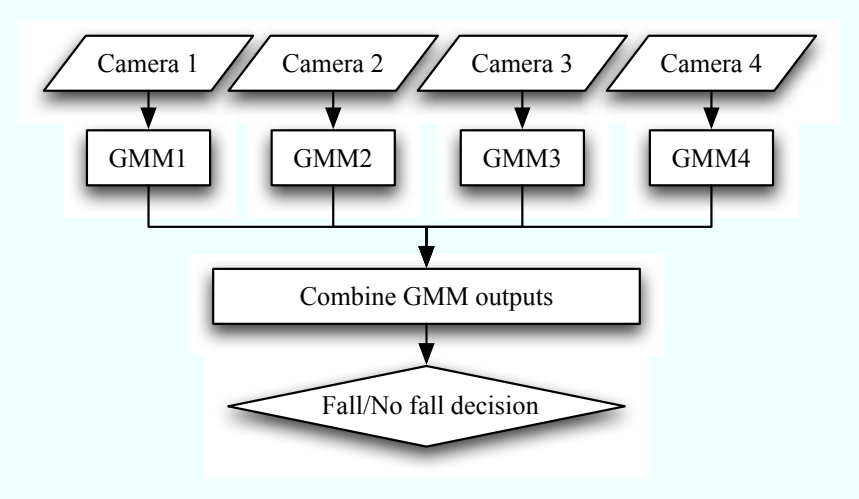


Figure 3.10 – Overview of the ensemble classifier.

The rule used is simply a majority vote on all cameras. Each camera GMM classifier has one vote, and if an abnormal event occurs in a majority of cameras, the event is considered as abnormal. Therefore, if at least 3 of the 4 cameras detect a fall, the event is considered as a fall. By combining all camera results, our system accuracy increased as shown in Fig. 3.9.

By choosing the best threshold, two events still remain misclassified with the D_f majority vote. One lure is detected as a fall, because of a high D_f peak, when the person brutally sits down on the armchair. One fall is not detected when the person gets up from the sofa and falls on the small table. The reason is that this fall is rather smooth and the resulting D_f peak is not sufficiently high.

As expected, the *mean matching cost* gave good results similar to the *full Procrustes distance*. Figure 3.9 showed that ROC curves with these features are similar, which provides evidence that our method is view-independent for the two features. For the *full*

Procrustes distance, the best classification error rate is less than 10% for each camera independently, and decreases to 2.7% using a majority vote. For the *mean matching cost*, the majority vote gave more than 98% accuracy.

3.12.4 Comparative study with other 2D features

In this section, we compare our shape feature with other commonly used 2D features :

- The **aspect ratio ρ of the bounding box**. Several works used this feature to detect falls [116][5] because of its simplicity. The bounding box should change from a vertical to a horizontal orientation after the fall.
- The **2D vertical velocity** v_y. This feature has been used for fall detection in [107] with an infrared sensor. The 2D vertical velocity is computed from the motion of the centroid of the person's silhouette. It should be high during the fall and low just after the fall.
- The **normalized 2D vertical velocity** v_{yn}. An object moving at a constant 3D velocity will display a higher 2D image velocity near the camera than far away from it due to perspective projection. One possible solution to this inconsistency problem could be to normalize the 2D velocity by the person's 2D size. To estimate this size, we compute the best fitting ellipse of the silhouette using moments [99] as shown in Fig. 3.11. The normalized 2D vertical velocity is then obtained by dividing it by the size of the ellipse major axis. It should be high during the fall and low just after the fall.

For a fair comparison, our 3-component GMM classifier is used with the new features representing the fall and the same inactivity period after the fall. Table 3.II (a) summarizes the classification results obtained with the cross-validation for each feature. The computed values are the EER (Equal Error Rate) and the AUC (Area Under the Curve) for each camera, and the EER and AUC corresponding to the majority vote for 4 cameras (events detected for at least 3 out of 4 cameras).

Figure 3.11 – Example of the bounding box (red) and of the approximated ellipse (green) of the silhouette. We can see here that the bounding box is very sensible to carried objects.

Features*	Cam. 1 EER (AUC)	Cam. 2 EER (AUC)	Cam. 3 EER (AUC)	Cam. 4 EER (AUC)	Majority vote EER(AUC)
(D_{f_1}, D_{f_2})	9.1% (0.978)	9.4% (0.968)	11.3% (0.964)	9.1% (0.966)	3.8% (0.996)
$(\bar{C}_1, \bar{C}_2)$	5.7% (0.977)	9.1% (0.975)	5.7% (0.990)	13.2% (0.956)	4.6% (0.997)
(ρ_1, ρ_2)	45.5% (0.631)	29.6% (0.729)	40.9% (0.578)	31.8% (0.719)	43.4% (0.488) †
(v_{y_1}, v_{y_2})	36.4% (0.697)	22.7% (0.817)	22.7% (0.745)	40.7% (0.653)	11.3% (0.889) ‡
(v_{yn_1}, v_{yn_2})	44.4% (0.622)	36.4% (0.701)	32.1% (0.675)	50.0% (0.538)	22.7% (0.776) †
D_{f_1}	27.3% (0.813)	11.3% (0.937)	18.2% (0.886)	24.1% (0.849)	13.6% (0.879) †
$\bar{C}_1$	27.3% (0.778)	17.0% (0.907)	22.7% (0.846)	36.4% (0.676)	17.0% (0.881) †

(a) Features comparison (Best matching points, no inactivity zones)

(D_{f_1}, D_{f_2})	13.2% (0.963)	9.4% (0.965)	7.6% (0.988)	13.6% (0.930)	9.1% (0.907)
$(\bar{C}_1, \bar{C}_2)$	11.3% (0.953)	9.1% (0.979)	9.4% (0.979)	13.6% (0.935)	0% (1)

(b) Comparison with Hungarian matching (Hungarian matching, no inactivity zones)

(D_{f_1}, D_{f_2})	5.7% (0.983)	9.1% (0.979)	7.6% (0.971)	9.1% (0.983)	0% (1)
$(\bar{C}_1, \bar{C}_2)$	4.6% (0.984)	0% (1)	5.7% (0.988)	9.4% (0.972)	1.9% (0.999)

(c) Influence of inactivity zones (Best matching points, inactivity zones)

* *Full Procrustes distance D_f, mean matching cost $\bar{C}$, bounding box ratio ρ, 2D vertical velocity v_y, normalized 2D vertical velocity v_{yn}*
† *Significantly different w.r.t. (D_{f_1}, D_{f_2}) using a one-sided binomial test ($p < 0.05$)*
‡ *Almost significantly different p=0.0898*

Tableau 3.II – EER and AUC values for different features (a), using Hungarian shape matching (b) and using inactivity zones (c).

The *ratio of the bounding box* gave poor results. Indeed, segmentation errors can affect the bounding box, because of shadows, highlights, occlusions, object carrying or simply if the person extends his arms as shown in Fig.3.11. The *2D vertical velocity* is sensitive to the camera view point, the velocity is higher when the person is near the camera. Normalizing the 2D vertical velocity doesn't improve the recognition results mainly because the size of the person is unreliable as explained for the bounding box aspect ratio. However, with a better assessment of the person's size, this approach could be more effective.

The *mean matching cost* and the *full Procrustes distance* gave the best recognition results with, respectively, an Equal Error Rate of 4.6% and 3.8% with a majority vote.

Table 3.II (a) shows also that the inactivity period is important to confirm the fall. If we only consider the first feature D_{f_1} or $\bar{C}_1$, the results deteriorate considerably as expected.

The results obtained using the Hungarian algorithm [69] for bipartite matching with 20% dummy points are also shown in Table 3.II (b) for comparison with our method using only the best matching points. The results are not statistically different from those obtained with our methodology. However, Hungarian matching is more time consuming, requires choosing the percentage of dummy points (a parameter that affects considerably the quality of the results) and can leave bad matching points.

A solution to improve recognition results could be to define normal inactivity zones (for example, the bed or the sofa) as in the work of Lee and Mihailidis [70], where the detection thresholds were less sensitive. We defined manually normal inactivity zones in our sequences, and when the centroid of the person was inside one of these zones, the detection threshold was fixed at 1.5 times the normal threshold. The results using inactivity zones are better, as shown in table 3.II (c). Normal inactivity zones could be automatically learned before installing the system.

To summarize, D_f or $\bar{C}$ features gave the best results compared to other features. A multi-camera system with 4 uncalibratred cameras increased the performance and so did the inclusion of known inactivity zones.

3.13 Discussion and Conclusion

In this work, we have presented a new GMM classification method to detect falls by analyzing human shape deformation.

The edge point matching step (with Shape Context) is robust to occlusions and other segmentation difficulties. We also observed that the addition of edge points within the silhouette (Canny filter), while not absolutely necessary, generally helped to improve the results (e.g. reduction of the EER from 9.1% to 3.8% for the majority vote with D_f).

Human shape deformation is a useful tool to detect falls. We have demonstrated that the full Procrustes distance and the mean matching cost are really discriminant features for classification. Because they can be sensitive to bad matching points, only reliable matching points were kept for shape deformation assessment.

The *mean matching cost* $\bar{C}$ can be sensitive in case of occlusion, segmentation problem or moving object. Figure 3.12 shows an example where $\bar{C}$ is incorrect when the person moved an armchair. Since $\bar{C}$ is based on shape context, and the context is not the same when an object moves or in case of occlusion, the matching cost increases similarly to a fall. The full Procrustes distance D_f is more robust in this case since it measures the shape deformation of reliable matched points.

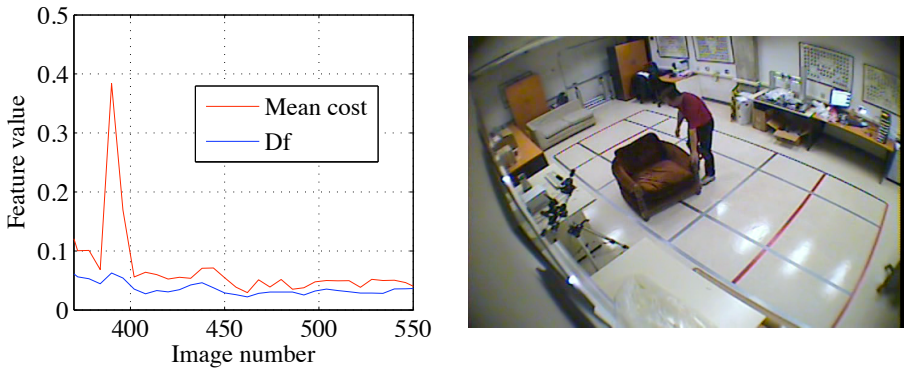


Figure 3.12 – Example of full Procrustes distance D_f and mean cost $\bar{C}$ curves for a video sequence where a person moved an armchair. The armchair is seen as a moving object which changes drastically the context of the object producing a false fall detection (peak) for $\bar{C}$.

This work was done with a realistic data set, and in spite of the low-quality images (high compression artifacts, noise) and segmentation difficulties (occlusions, shadows, moving objects, different clothes, etc), the recognition results are excellent. The system can run in real time at 5 fps which is fast and sufficient to detect a fall. Finally, compared with other 2D features, the shape deformation features are significantly superior tools to detect falls.

When developing such systems, we must ensure the privacy of the person. This requirement is satisfied with our system as it is entirely automated, and nobody has access to the images except in case of emergency. The system will be activated to send an alarm signal toward an outside resource (e.g. via a cell phone or Internet) if and only if an abnormal event is detected (e.g. falling). Moreover, this is a technique that does not require the person to wear any device.

We hope that this research study will set the ground for the development of healthcare video surveillance systems to improve the quality of life and care for the elderly so that they can preserve their autonomy and enjoy a greater degree of comfort in their daily lives. This corresponds to the hopes of the elderly themselves, their families, caregivers and governments. Indeed, a recent study on the acceptability by older people of such vision systems has been conducted by our team [100], and has revealed an encouraging high rate of acceptance among elderly people and care givers. When the intelligent videomonitoring system is well explained, 83.3% of caregivers and 86.7% of seniors are in favor of such system. Various studies [22] have also shown the economic advantages of support in the home setting instead of placing the elderly in a specialized long-term care establishment.

Finally, we believe that such a video system will complement advantageously other types of sensors for healthcare surveillance by overcoming many of their limitations.

Acknowledgment

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

CHAPITRE 4

OBTENIR UNE INFORMATION 3D À PARTIR D'UN SYSTÈME MONOCULAIRE

Les méthodes monoculaires se limitent souvent à des informations 2D (image) pour analyser des comportements telles que la forme de la silhouette de la personne ou sa trajectoire. Cependant, des informations 3D seraient très intéressantes car il serait alors possible de localiser une personne dans une pièce et de récupérer sa trajectoire 3D. Par exemple, les trajectoires 2D sont souvent limitées par l'angle de vue de la caméra, et un même mouvement 3D donnera un mouvement 2D dans l'image plus important si la personne est proche de la caméra.

Habituellement, il est nécessaire d'utiliser un système multi-caméras pour récupérer de l'information 3D. Mais les systèmes multi-caméras sont plus complexes à mettre en oeuvre, ils ont besoin d'être calibrés et synchronisés, et ils sont aussi plus coûteux. Pour un système bas coût, nous avons préféré nous limiter à une caméra par pièce dans l'appartement. Nous allons donc nous intéresser dans cette partie aux techniques qui pourraient nous aider à récupérer une information 3D à partir d'un système monoculaire, telle que la trajectoire 3D de la personne.

4.1 Modélisation géométrique d'une caméra

Un système monoculaire donne par défaut des informations 2D. Pour obtenir une information 3D, le système a besoin d'être calibré, c'est à dire soit par une caméra calibrée (connaissance des paramètres internes de la caméra tels que le centre optique et la focale), soit par un calibrage de la pièce (mesures et proportions connues de la pièce).

4.1.1 Formation d'une image

La formation d'une image consiste à transformer un point 3D $\mathrm{M}(X,Y,Z)$ du repère objet R en un point image (u,v) sur le capteur :

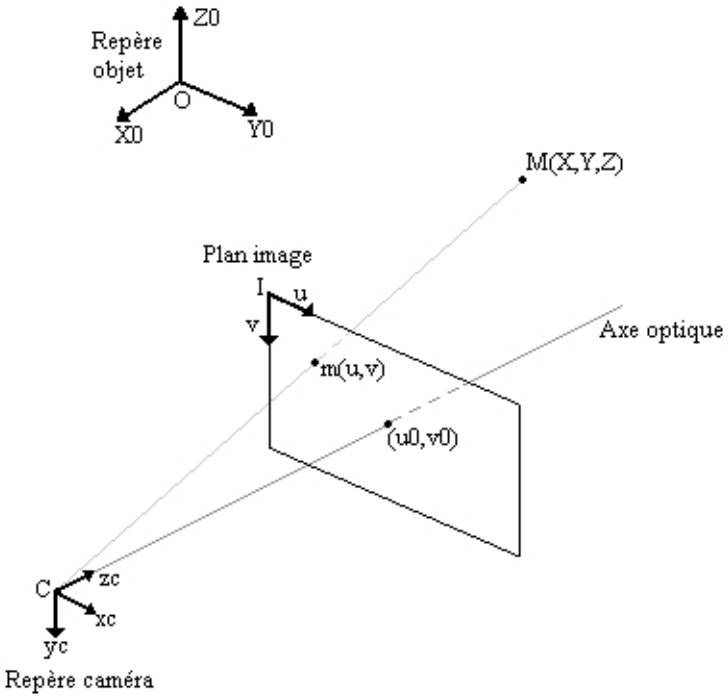


Figure 4.1 – Modèle sténopé de la caméra

Notations :

O$(X0, Y0, Z0)$ l'origine du repère objet 3D que l'on notera R.

C(xc, yc, zc) le repère caméra 3D que l'on notera Rc. Ce repère a pour origine le centre optique C de la caméra et son axe zc est confondu avec l'axe optique de la caméra.

$(I, \vec{u}, \vec{v})$ représente le repère image associé au plan image de la caméra.

$(u0, v0)$ représente les coordonnées en pixels dans le repère image de l'intersection de l'axe optique et du plan image.

M un point de coordonnées (X, Y, Z) dans le repère objet R, qui a pour coordonnées (Xc, Yc, Zc) dans le repère caméra Rc.

La formation d'une image se déroule en trois étapes successives :

Transformation objet-caméra :

La transformation objet-caméra permet de passer du repère 3D objet au repère ca-

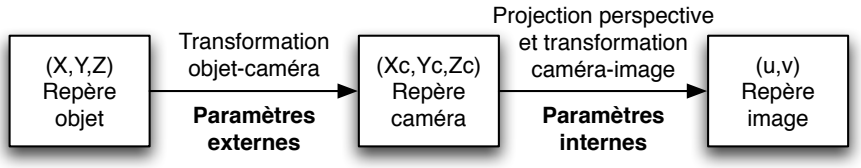


Figure 4.2 – Les différentes étapes de la formation d'une image

méra. Elle est représentée par les paramètres extrinsèques de la caméra, c'est à dire la position et l'orientation de la caméra par rapport au repère de référence. Cette transformation peut s'écrire sous la forme d'une matrice homogène exprimant la position de l'objet dans le repère caméra *Rc* :

$$\begin{pmatrix} Xc \\ Yc \\ Zc \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & Tx \\ r_{21} & r_{22} & r_{23} & Ty \\ r_{31} & r_{32} & r_{33} & Tz \\ 0 & 0 & 0 & 1 \end{pmatrix} . \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = M_{ext} . \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{4.1}$$

Projection perspective :

La projection perspective, basée sur le modèle sténopé, est utilisée pour passer d'un point 3D exprimé dans le repère caméra en un point 2D exprimé dans le repère caméra. Dans le repère caméra *Rc*, les coordonnées de la projection du point objet $\mathrm{M}(Xc, Yc, Zc)$ dans le plan image sont :

$$\begin{cases} x = f.\frac{Xc}{Zc} \\ y = f.\frac{Yc}{Zc} \\ z = f \end{cases} \tag{4.2}$$

En coordonnées homogènes, le système s'écrit sous la forme :

$$\begin{pmatrix} s.x \\ s.y \\ s \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} . \begin{pmatrix} Xc \\ Yc \\ Zc \\ 1 \end{pmatrix} \tag{4.3}$$

Transformation caméra-image :

La transformation caméra-image permet de passer d'un point 2D du repère caméra en un point $m(u,v)$ du repère image, ce qui correspond aux coordonnées pixel du point. Elle est représentée par les paramètres intrinsèques de la caméra. Les coordonnées du point m sont :

$$\begin{cases} u = x/dx + u0 \\ v = y/dy + v0 \end{cases} \tag{4.4}$$

Où dx et dy sont les dimensions respectivement en largeur et en hauteur d'un pixel du capteur CCD.

Avec les équations (4.3) et (4.4), la matrice de projection perspective s'écrit alors :

$$\begin{pmatrix} s.u \\ s.v \\ s \end{pmatrix} = \begin{pmatrix} f/dx & 0 & u0 & 0 \\ 0 & f/dy & v0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} . \begin{pmatrix} Xc \\ Yc \\ Zc \\ 1 \end{pmatrix} = M_{int} . \begin{pmatrix} Xc \\ Yc \\ Zc \\ 1 \end{pmatrix} \tag{4.5}$$

Où M_{int} est la matrice des paramètres internes de la caméra.
On notera par la suite : $fx = f/dx$ et $fy = f/dy$

Le système complet de formation d'image s'écrit alors :

$$\begin{pmatrix} s.u \\ s.v \\ s \end{pmatrix} = M_{int} . M_{ext} . \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{4.6}$$

4.1.2 Correction de la distorsion

Le système de formation d'images décrit précédemment représente un cas idéal où le système ne présente pas de distorsion, c'est à dire que l'image n'est pas déformée en sortie du système optique. Mais dans la réalité, tout système optique présente plus ou moins de la distorsion, et il convient d'en tenir compte lorsque l'on calibre une caméra.

On peut regrouper les distorsions en deux catégories [128] :

La distorsion radiale C'est la courbure des lentilles qui entraîne de la distorsion radiale. Cette déformation est symétrique par rapport au centre de l'image et dépend de la position du diaphragme vis à vis de la lentille :

- *Distorsion en barillet :* Si le diaphragme est placé entre la l'objet et la lentille, alors le grandissement diminue pour les rayons éloignés de l'axe, ce qui donne une image en forme de barillet.
- *Distorsion en coussinet :* Si le diaphragme est placé entre la lentille et l'image, alors le grandissement augmente pour les rayons éloignés de l'axe, ce qui donne une image en forme de coussinet.

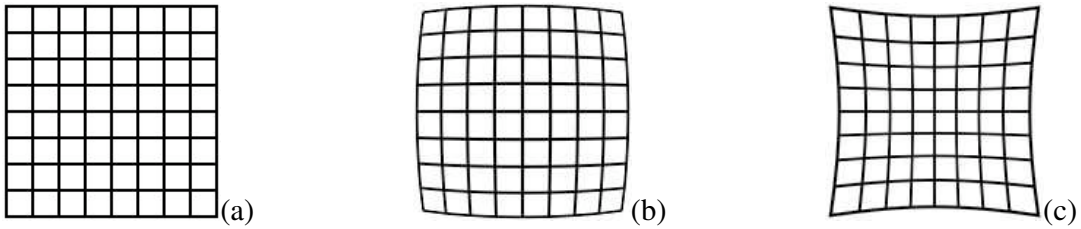


Figure 4.3 – La distorsion radiale : pas de distorsion (a), avec distorsion en barillet (b), avec distorsion en coussinet (c)

La distorsion tangentielle Deux types de défauts peuvent être à l'origine de la distorsion tangentielle : il se peut que la lentille ne soit pas centrée par rapport à l'axe optique, ou qu'elle ne soit pas placée perpendiculairement à l'axe optique. Tsai [118] a montré que la distorsion radiale est plus importante que la distorsion tangentielle, et que par conséquent, elle peut en général être négligée dans les techniques de correction de distorsion.

La distorsion est donc la combinaison d'une composante radiale et d'une composante tangentielle. Elle peut être modélisée [51] par les équations :

$$\begin{cases} u_d = u_u + \delta u_r + \delta u_t \\ v_d = v_u + \delta v_r + \delta v_t \end{cases} \tag{4.7}$$

Avec (u_d, v_d) un point image avec de la distorsion et (u_u, v_u) ce même point corrigé de la distorsion. La distorsion radiale est exprimée par le paramètre $(\delta u_r, \delta v_r)$ et la distorsion tangentielle par $(\delta u_t, \delta v_t)$. Ces paramètres ont pour expression :

$$\begin{cases} \delta u_r = u_u \left(k_1 r^2 + k_2 r^4 + ...\right) \\ \delta v_r = v_u \left(k_1 r^2 + k_2 r^4 + ...\right) \end{cases} \tag{4.8}$$

$$\begin{cases} \delta u_t = 2p_1 u_u v_u + p_2 \left(r^2 + 2u_u^2\right) \\ \delta v_t = p_1 \left(r^2 + 2v_u^2\right) + 2p_2 u_u v_u \end{cases} \tag{4.9}$$

avec k_i les coefficients de la distorsion radiale et p_i ceux de la distorsion tangentielle et $r^2 = \sqrt{u_u^2 + v_u^2}$,

4.2 Calibrage d'une caméra

Pour notre projet, nous avons utilisé des caméras bas de gamme de type caméras IP avec un grand angle de vue. Cependant, pour avoir un grand angle de vue, la caméra doit être équipée d'une petite focale ce qui implique beaucoup de distorsion. Il est donc nécessaire de calibrer la caméra afin de connaître ses paramètres internes (focale et centre optique) et la correction de la distorsion si l'on veut avoir une information 3D de qualité.

Zhang [136] classifie les méthodes de calibrage en trois catégories :

Calibrage avec un objet 3D Les méthodes traditionnelles de calibrage de caméra utilisent une mire de calibrage qui possède deux ou trois plans orthogonaux les uns aux autres. Un exemple typique de mire non coplanaire est celle utilisée par Tsai [118], une composition de deux plans perpendiculaires contenant chacun un damier. Devernay [35] présente une méthode qui pourrait être intéressante dans le

futur pour notre application : aucune mire de calibrage n'est requise, le calibrage est fait à partir de segments 3D tels que les arêtes des murs d'une pièce. Ce calibrage est automatique et permet de calculer les paramètres internes de la caméra, mais aussi d'estimer la distorsion.

Calibrage avec un objet 2D Le calibrage avec un objet 2D consiste à prendre plusieurs images de ce même objet prises sous des angles de vue différents comme montré sur la Fig. 4.4. Zhang [134, 135] utilise une mire en damier pour extraire les paramètres intrinsèques et extrinsèques de la caméra. Cette méthode a été implémentée avec succès dans deux bibliothèques très utilisées dans le milieu de la vision par ordinateur qui sont la boîte à outils Matlab de calibration de caméra [13] et la bibliothèque OpenCV [16] en C++ de Intel. Ce sont ces bibliothèques que nous utiliserons par la suite dans nos travaux pour calibrer nos caméras étant donné leurs bons résultats et leur simplicité d'utilisation.

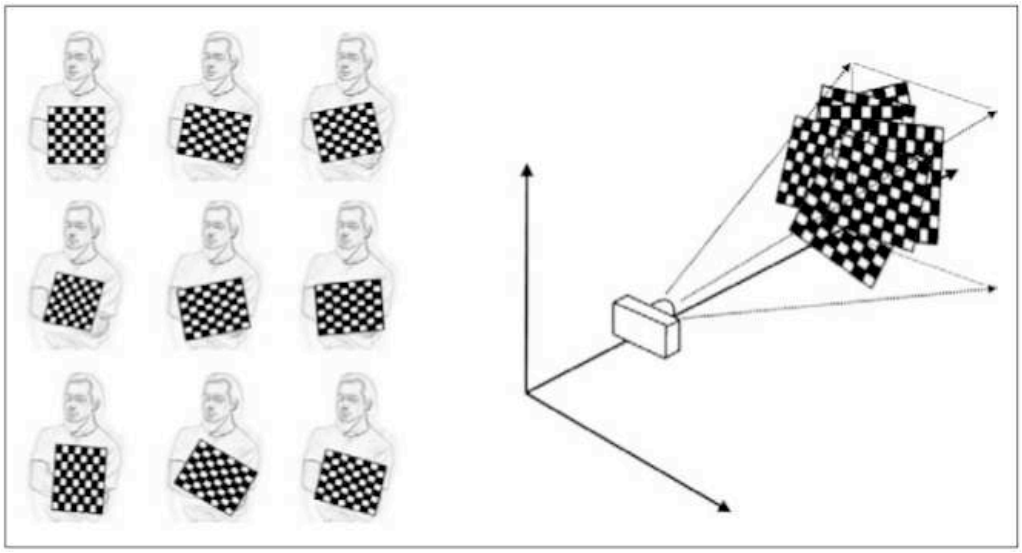


Figure 4.4 – Principe du calibrage avec un damier [16].

Auto-calibrage Dans cette catégorie, on n'utilise pas d'objet de référence pour calibrer, c'est pourquoi Zhang [136] appelle cette catégorie l'approche 0D. Dans les méthodes précédentes, on utilisait pour calibrer la correspondance entre des

points objet connus et leurs correspondants image, alors que dans le cas de l'auto-calibrage [49, 76], seules des correspondances entre points image sont utilisées. Les points image sont reliés par une homographie entre deux images de la caméra. Cette méthode est surtout utilisée dans le cas de systèmes stéréo ou multi-caméras, ou dans le cas d'une caméra en mouvement. L'auto-calibrage ne sera donc pas utilisée pour notre application basée sur des caméras monoculaires fixes.

Zhang [136] définit même une nouvelle catégorie en proposant une méthode avec un objet 1D (points alignés sur une ligne). D'autres techniques existent encore, notamment en utilisant les points de fuite [19] dans les trois directions orthogonales. Cependant, l'estimation des points de fuite est parfois imprécise et donc le calibrage n'est pas toujours optimal. Il est aussi possible de calibrer une caméra en rotation pure [112], c'est à dire que la caméra effectue une rotation autour de son centre optique sans aucune translation.

4.3 Information 3D avec une seule caméra

Récupérer une information 3D avec une seule caméra peut se traduire sous plusieurs formes : cela peut s'inscrire dans un problème de calcul de pose d'un objet par rapport à la caméra, ou de positionnement d'un objet dans son environnement grâce à des techniques de photogrammétrie. Nous allons passer en revue les différentes techniques qui pourraient nous fournir cette information dans cette section.

a - Estimation de pose

Estimer la pose d'un objet signifie calculer sa position et son orientation par rapport à une caméra, c'est ce qu'on appelle les paramètres extrinsèques de la caméra. C'est un vieux problème pour lequel de nombreux travaux en photogrammétrie et en vision par ordinateur ont été réalisés, et qui connaît donc plusieurs solutions.

Il est possible de retrouver la pose d'un objet à partir de points, de lignes ou d'autres primitives. Les méthodes utilisant des points sont connues sous le nom de PnP c'est à dire *Perspective from n Points*, et nécessitent au moins $n \geq 3$ points connus. La pose d'un objet peut être calculée en utilisant une caméra calibrée, et en connaissant les points 3D de l'objet de référence et les points 2D correspondants (projection des points 3D dans

l'image).

Parmi les méthodes à 3 points, l'algorithme de Church [108], utilisée initialement en photogrammétrie aérienne, est une technique itérative pour calculer la pose à partir d'un triangle. D'autres variantes à partir de 3 points existent, Haralick *et al.* [46] proposent un comparatif de quelques solutions possibles. Les solutions sont en général soit itératives, soit algébriques. Le problème des méthodes itératives est qu'elles nécessitent une bonne initialisation, alors que les méthodes algébriques sont souvent sensibles au bruit des données [95]. Avec 3 points, le problème peut donner jusqu'à 4 solutions possibles [49], et le problème devient alors de choisir la bonne pose. Pour éviter le problème des solutions multiples, il faut alors ajouter une contrainte pour obtenir la bonne solution, comme ajouter un point. Une solution unique existe à partir de 4 points non coplanaires. Il faut noter que plus le nombre de points est important, plus la pose sera précise.

Un exemple de méthode algébrique à 4 points est proposé par Quan et Lan [95]. Leur méthode utilise la redondance des 4 points pour extraire une solution unique. L'algorithme POSIT de Dementhon [32] fait partie des solutions itératives mais qui ne souffre pas de problème d'initialisation. En effet, initialement, la pose est calculée à partir d'un modèle de projection perspective à l'échelle (caméra para-perspective), puis par itération, l'algorithme converge vers un modèle de projection perspective pure. Cette méthode est très utilisée car c'est un algorithme rapide en temps de calculs et qui permet une estimation de la pose beaucoup plus précise que les méthodes algébriques.

b - Information 3D à partir de la connaissance de l'environnement

Pour utiliser les méthodes précédentes d'estimation de pose, il est nécessaire d'avoir les proportions de l'objet. Dans le cas où elles ne seraient pas connues, il est possible de localiser l'objet par rapport à son environnement. Par exemple, on pourrait localiser une personne par la position de ses pieds dans un plan de référence connu. On peut citer les travaux de Criminisi [26, 27] qui utilise les propriétés de la géométrie projective pour reconstruire en 3D une scène à partir d'une seule image. Les seules données dont il a besoin sont : un ensemble de lignes de fuite orthogonales, 4 points connus sur le sol et 1 hauteur connue dans la scène. Ses résultats sont très intéressants et peuvent être

appliqués à notre problématique pour positionner la personne dans la pièce, à condition de pouvoir voir ses pieds dans l'image. Dans le cas où les pieds de la personne seraient temporairement cachés à l'instant t, il serait possible par une interpolation temporelle entre $t-1$ et $t+1$ d'estimer leur position à l'instant t. Ou, si la personne est droite et que l'on connaît sa taille, il est possible de prédire la position de ses pieds par une interpolation spatiale.

CHAPITRE 5

3D HEAD TRACKING USING A SINGLE CALIBRATED CAMERA (ARTICLE)

Ce chapitre présente le manuscript intitulé *« 3D Head Tracking using a Single Calibrated Camera »* soumis au journal *Image and Vision Computing*, par Caroline Rougier, Jean Meunier, Alain St-Arnaud et Jacqueline Rousseau (2010).

5.1 Avant-propos

Dans une étude en biomécanique avec des marqueurs sur le corps, Wu [131] a montré que les vitesses 3D étaient très utiles pour détecter des chutes. En partant de cette idée, nous avons voulu récupérer des vitesses 3D par des techniques de vision par ordinateur mais sans l'utilisation de marqueurs. Nous nous sommes intéressés à la trajectoire 3D de la tête car c'est la partie du corps qui aura le plus de vitesse lors d'une chute. De plus, la tête à la forme caractéristique d'une ellipsoïde 3D ou d'une ellipse 2D dans le plan image [12], et est en général toujours visible dans l'image (moins de problèmes d'occultations pour le haut du corps).

Dans nos précédents travaux [98], nous avions vu le problème sous la forme d'un suivi 2D de la tête nous permettant d'estimer la pose 3D de la tête connaissant ses proportions. La tête modélisée par une ellipse était suivie à l'aide d'un filtre à particules [58] dans la séquence vidéo. La pose 3D de la tête était alors obtenue à l'aide de l'algorithme POSIT [32] connaissant les proportions réelles de la tête (données anthropométriques), sa projection dans l'image (ellipse 2D suivie) et le calibrage de la caméra. Cette méthode donnait de bons résultats lorsque la personne était debout, mais pouvait donner des poses erronnées lors d'une chute suivant le point de vue de la caméra.

Pour mieux gérer les inclinaisons de la personne, nous avons étudié le problème d'une manière différente : la tête est vue comme étant une ellipsoïde 3D qui est projetée dans le plan image à l'aide des paramètres internes de la caméra (voir Annexe IV). Un

filtre à particules [58] est utilisé pour suivre le modèle 3D dans la séquence vidéo. Cette méthode est expliquée dans le deuxième article de cette thèse décrit dans ce chapitre.

Pour localiser un objet par rapport au sol, nous avons besoin de connaître la position de la caméra par rapport au sol. Pour nos expérimentations, nous avons utilisé des marquages au sol de dimensions connues. Les paramètres extrinsèques de la caméra peuvent alors être retrouvés à partir de la matrice homographique liant le repère 3D au sol (marquage planaire) à sa projection dans l'image [134].

Pour évaluer la précision de notre algorithme au niveau de la localisation 3D, nous avons utilisé la bibliothèque de vidéos HumanEva décrite dans l'annexe II. Nous avons aussi testé notre algorithme sur des mouvements rapides tels que lors d'une chute avec notre bibliothèque de vidéos de chute décrite dans l'annexe I.

Les notations utilisées dans cet article sont internes à l'article et n'ont pas de lien avec les notations utilisées dans le reste de la thèse.

5.2 Abstract

The head trajectory is an interesting source of information for behavior recognition and can be very useful for video surveillance applications. Consequently, much work has been done to track the head in the 2D image plane using a single camera or in a 3D world using multiple cameras. In the same vein, we propose here an original method to extract the 3D head trajectory of a person in a room but this time using only one calibrated camera. The head is represented as a 3D ellipsoid, which is tracked with a hierarchical particle filter based on color histograms and shape information. Experiments demonstrated that this method can run in quasi-real-time, providing reasonable 3D errors for a monocular system. An application example is also presented for fall detection using the head 3D vertical velocity or height obtained from the 3D trajectory.

Keywords : computer vision, 3D, head tracking, monocular, particle filter, video surveillance, fall detection

5.3 Introduction

The human head is an easy human body part to track in a scene as it is usually visible from different camera points of view and its shape is relatively simple. The head trajectory can be very useful for video surveillance applications, and has already been used for action recognition in the 2D image plane [23] or in 3D [66].

Tracking the head in the 2D image plane has been widely investigated. One of the most famous studies was done by Birchfield [12] who showed that a head is well-approximated by an ellipse in the 2D image plane. In that case the tracker was based on a local search using gradient and color information. A head ellipse was also tracked by Nummiaro *et al.* [87] using a color-based particle filter. An ellipse was also used in the work of Nait Charif and McKenna [23] to track the head of multiple persons in a room using a particle filter with measurement based on the color histogram and gradient information along the head boundary.

However, a 3D trajectory gives more information about the localization of the person in a room and its 3D movement. 3D head tracking from video can be divided into two main classes of strategy : single-camera and multi-camera.

5.3.1 Multi-Camera Systems

Usually, a multi-camera system is required to provide some 3D information. Moreno *et al.* [79] have used a calibrated stereo rig for their 3D head tracker. The head shape was modeled by an ellipse and was tracked with an updated color histogram. The depth information obtained from the stereo system was used to scale the ellipse. Kawanaka *et al.* [66] have tracked the head in a 3D voxel space with four stereo cameras, using particle filtering with depth and color information. Usabiaga *et al.* [120] have proposed to use the 3D head trajectory obtained from multiple cameras to recognize simple human actions like sitting-down or bending-down. The head was first detected with an elliptical head tracker [12] in the 2D image plane of each camera, and the 3D head location was then obtained by triangulation. The action recognition step was done by aligning the 3D trajectories using Dynamic Time Warping (DTW). Wu and Aghajan [130] have

recovered both the 3D trajectory of the head and its pose with a multi-camera system without accurate calibration of the camera network. Their method was based on Chamfer matching refined with a probabilistic framework. Kobayashiet *et al.* [68] have used AdaBoost-based cascaded classifiers to extract the 3D head trajectory using several cameras.

5.3.2 Monocular Systems

Very few attempts have been done to track the 3D head trajectory in real-time with a single camera. Hild [52] recovered a 3D motion trajectory from a walking person, but he supposed that the person was standing and that the camera optical axis was parallel to the (horizontal) ground plane. These assumptions cannot be made in video surveillance applications : the camera needs to be placed higher in the room to avoid occluding objects and to have a larger field of view, and the person is not supposed to be standing or facing the camera. In our previous work [98], we tried to recover the 3D head localization from a 2D head ellipse tracked in the image plane. The 3D head pose was computed using a single calibrated camera, a 3D model of the head and its projection in the image plane. This method worked well with a standing person, but the 3D pose can be wrongly estimated when the person falls.

5.4 Method Overview

In this work, we explore a different method : the head is viewed as a 3D ellipsoid which is projected as an ellipse in the 2D image plane, and tracked through the video sequence using a particle filter. Therefore, contrary to our previous work [98], it is now the 3D ellipsoid that is directly tracked instead of its 2D ellipse projection. The 3D head model and its projection in the image plane is described in Section 5.5. The head tracking module using a particle filter is shown in section 5.6. Section 5.7 presents several tests conducted with the HumanEva-I dataset [106]. An illustrative application for fall detection is also shown.

5.5 Head Model Projection

We chose to represent the head by a 3D ellipsoid, which is projected in the image plane as an ellipse [113]. The 3D head model will be tracked in the world coordinate system attached to the XY ground plane. To project the 3D head model in the image plane, we need to know the camera characteristics (intrinsic parameters) and the pose of the XY ground plane relative to the camera (extrinsic parameters).

5.5.1 The intrinsic parameters

The intrinsic parameters can be computed using a chessboard calibration pattern and the camera calibration toolbox for Matlab [13]. The camera's intrinsic matrix K is defined by :

$$K = \begin{pmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{5.1}$$

with the focal length (f_x, f_y) and the optical center (u_0, v_0) in pixels. Notice that image distortion coefficients (radial and tangential distortions) are also computed with our methodology and used to correct the images for distortion before processing.

5.5.2 The extrinsic parameters

The extrinsic parameters are obtained from corresponding ground points between the real word and the projected image points. The plane-image homography obtained from these two sets of points is used to compute the extrinsic parameters :

$$M_{ext} = \begin{pmatrix} R & T \\ 0\,0\,0 & 1 \end{pmatrix} \tag{5.2}$$

where R and T are respectively a 3D rotation matrix and a 3D translation vector, as described in the work of Zhang [135].

5.5.3 Ellipsoid projection

An ellipsoid is a quadric described by a positive definite matrix Q_C in the camera coordinate system such that :

$$[x,y,z,1]^T Q_C [x,y,z,1] = 0 \tag{5.3}$$

for a point (x,y,z) belonging to the ellipsoid.

This ellipsoid is projected in the image plane with the projection matrix P as a conic C [48, 113] :

$$C = Q_{C_{44}} Q_{C_{1:3,1:3}} - Q_{C_{1:3,4}} Q_{C_{1:3,4}}^T \tag{5.4}$$

The ellipse is described by the conic by $[u,v,1]^T C [u,v,1] = 0$ for a point (u,v) in the image plane.

Concretely, the ellipsoid matrix representing the head in the head coordinate system has the form :

$$Q_H = \begin{pmatrix} \frac{1}{B^2} & 0 & 0 & 0 \\ 0 & \frac{1}{B^2} & 0 & 0 \\ 0 & 0 & \frac{1}{A^2} & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{5.5}$$

With the semi-major A and the semi-minor B ellipsoid head axes.

The head ellipsoid is projected as Q_C in the camera coordinate system with the projection matrix P :

$$Q_C = P^{-1T} Q_H P^{-1} \tag{5.6}$$

The projection matrix P represents here the transformation from the head ellipsoid coordinate system to the image plane :

$$P = K M_{ext} M_{Head/World} \tag{5.7}$$

The matrix $M_{Head/World}$ will be defined during the tracking process by the translation and rotation of the head in the world coordinate system (see Section 5.6). Finally, the conic obtained from eq. 5.4 is used to defined the ellipse parameters. Our 3D ellipsoid model is shown in Fig. 5.1. Notice that the 3D point corresponding to the centroid of the person, which will be used during the tracking process, corresponds to the half-height of the person.

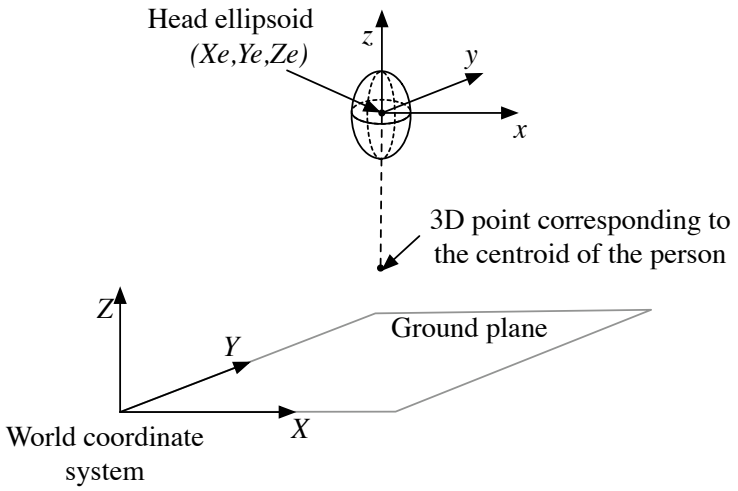


Figure 5.1 – The 3D ellipsoid model.

5.6 3D Head Tracking with Particle Filter

5.6.1 Particle filter

Particle filters have been used with success in several applications, and in particular to track a head with an ellipse [87, 98], or a parametric spline curve [58] using color information or edge contours. In our work, a particle filter is particularly suitable as it allows abrupt variations of the trajectory and can deal with small occlusions.

The main idea of particle filters is to estimate the probability distribution $p(S_t|Z_t)$ of the state vector S_t of the tracked object given Z_t, representing all the observations. This probability can be approximated from a set of N weighted samples (also called particles).

The main steps of our particle filter algorithm are shown in Fig. 5.2.

Each particle $(n = 1, \ldots, N)$ of our filter is an ellipsoid, represented by the state vector :

$$s_t^n = [X_e, Y_e, Z_e, \theta_{Xe}, \theta_{Ye}]_t^n \tag{5.8}$$

where (X_e, Y_e, Z_e) is the 3D head ellipsoid centroid expressed in the world coordinate system (translation component of the matrix $M_{Head/World}$), and $(\theta_{X_e}, \theta_{Y_e})$ are respectively the rotation around the X and the Y axes (rotation component of the matrix $M_{Head/World}$) [1].

Our dynamical model doesn't include motion (the known velocity between two successive images is added to the particles before propagating the particles), so the deterministic component A_l is represented by an identity matrix (see Fig. 5.2). The stochastic components B_l varies for each layer as shown in the section 5.6.5.

5.6.2 Particles Weights

The particle weights are based on foreground, color and body coefficients.

- **Foreground coefficient** C_F

  To compute the foreground coefficient, we first need to extract the person silhouette in the image. For this purpose, we use a background subtraction method which consists in comparing the current image with an updated background image. We use the codebook method from the work of Kim *et al.* [67] which takes into account shadows, highlights and high image compression. The foreground coefficient is computed by searching for silhouette contour points along N_e line segments normal to the ellipse. These segments are distributed uniformly along the ellipse and centered on its contour (Fig. 5.3, right). For an ellipse defined by the semi-axis parameters a and b, the starting point of each segment is on an inner ellipse defined by the parameters $a/2$ and $b/2$ and the ending point is on the outer ellipse defined by $a + a/2$ and $b + b/2$ (see Fig. 5.3, right). For each normal seg-

[1] Notice that two angles (instead of three) are sufficient to define the position and orientation of the ellipsoid since its minor axes have both the same length in our model.

Notation

$S_t = \{s_t^n, n = 1, \ldots, N\}$	The set of samples at time t
$\Pi_t = \{\pi_t^n, n = 1, \ldots, N\}$	The corresponding weights
$C_t = \{c_t^n, n = 1, \ldots, N\}$	The normalized cumulative probabilities

Given the previous sample set $\{s_{t-1}^n, \pi_{t-1}^n, c_{t-1}^n, n = 1, \ldots, N\}$ at time $t-1$, a new sample set $\{s_t^n, \pi_t^n, c_t^n, n = 1, \ldots, N\}$ at time t is constructed through several layers.

Initially, $S_{old} = S_{t-1}$ the previous sample set.

For each layer l=1 ... L

1. ***Selection :***
   N new samples are selected from the old sample set S_{old} by favoring the best particles.
   - Generate a random number $r \in [0, 1]$, uniformly distributed
   - Find, by binary search, the smallest j for which $c_{old}^j \geq r$
   - Set $s_t'^n = s_{old}^j$
2. ***Prediction :***
   The new samples are predicted with a stochastic dynamical model $s_t^n = A_l s_t'^n + B_l w_t^n$ where w_t^n is a vector of standard normal random variables. A_l and B_l are, respectively, the deterministic and stochastic components of the dynamical model.
3. ***Measurement :***
   Compute the new weights $\pi_t^n = p(z_t | s_t^n)$ and normalize so that $\sum_n \pi^n = 1$.
   Compute the cumulative probabilities :
   - $c_t^0 = 0$
   - $c_t^n = c_t^{n-1} + \pi_t^n \quad with\ n = 1, \ldots, N$
4. ***Reduce the stochastic component :***
   - The stochastic component decreases for the next step : $B_{l+1} = B_l/2$
   - The current sample set becomes the old sample set : $S_{old} = S_t$

Estimation

The mean state of the system is then estimated at time t using the N final weighted samples :

$$E[S_t] = \sum_{n=1}^{N} \pi_t^n s_t^n$$

Figure 5.2 – The hierarchical particle filter algorithm

ment, the distance d_e from the ellipse point to the detected silhouette point is used to compute the foreground coefficient :

$$C_F = \frac{1}{N_e} \sum_{n=1}^{N_e} \frac{D_e(n) - d_e(n)}{D_e(n)} \ , \quad C_F \in [0 \dots 1] \tag{5.9}$$

where D_e, the half length of the normal segment, is used to normalize the distances. An example of foreground coefficient is shown in Fig. 5.3.

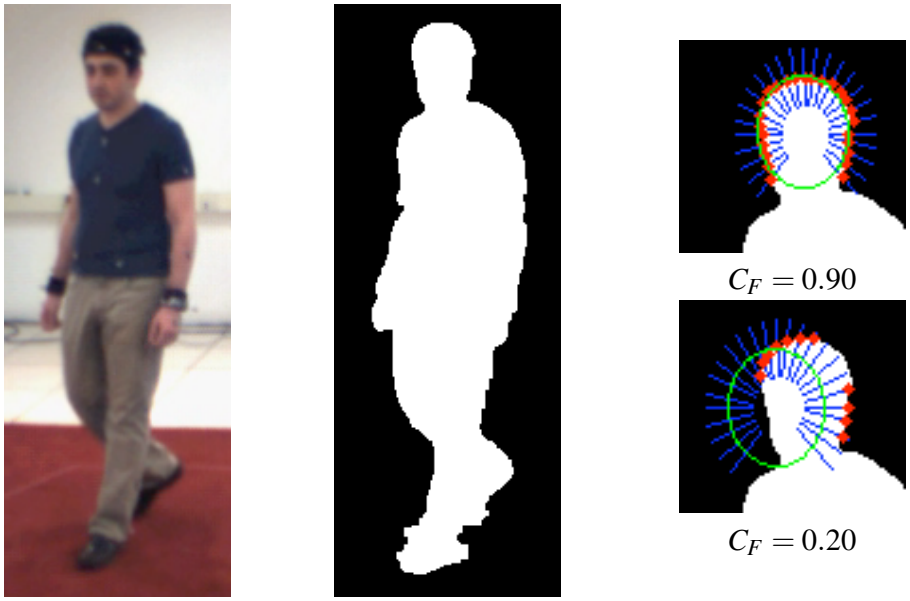


Figure 5.3 – Example of foreground segmentation and foreground coefficient computation.

- **Color coefficient** C_C

  The color coefficient is based on the normalized 3D color histogram H of the head ellipse [87]. The color histogram, in the RGB color space, is composed of $N_b = 8 \times 8 \times 8$ bins and is computed inside a rectangular zone included in the ellipse. The comparison between an updated color head model and the target model is then done by calculating the normalized histogram intersection :

$$C_C = \sum_{i=1}^{N_b} \min \left(H(i), H_{ref}(i) \right) \ , \quad C_C \in [0 \dots 1] \tag{5.10}$$

- **Body coefficient** C_B

  The body coefficient is used to link the head to the body through the body center. The projection of the 3D point corresponding to the centroid of the person (see Fig. 5.1) should be near the centroid of the 2D silhouette (distance d_b compared to the half-major axis of the bounding box D_b). This coefficient is useful to avoid unrealistic 3D ellipsoid rotation ; it is only used when the bounding box is valid (and thus not used in case of occlusion for example).

$$C_B = \frac{D_b - d_b}{D_b} \ , \quad C_B \in [0 \dots 1] \tag{5.11}$$

- **Final coefficient**

  Finally, the ellipsoid coefficient is a combination of these coefficients amplified by a Gaussian to give larger weights to the best particles ($\sigma = 0.15$) :

$$C_{final} = \frac{1}{\sqrt{2\pi}\sigma} \exp^{(C_F C_C C_B)/2\sigma^2} \tag{5.12}$$

Each coefficient is important. The foreground coefficient is used for the 3D pose precision when the ellipsoid is matched to the head contour. The color coefficient becomes very useful during large movement to prevent the ellipsoid from hanging on something else inside the silhouette. The body coefficient fixes a base for the head to prevent an abnormal rotation of the ellipsoid around its center.

5.6.3 Ellipsoid Calibration

From the top head point of the foreground silhouette, we manually initialize an ellipse representing the head. This head ellipse is used to calibrate the ellipsoid size relative to the body height (supposed to be known). The aspect ratio of the ellipse is fixed at 1.2 for a human head [12]).

5.6.4 Initialization

A head detection module is used to automatically initialize our system. When the person stands up, the top head point is detected in the foreground silhouette. From this point, several 2D ellipses are tested and the one which has the biggest foreground coefficient C_F is kept. If $C_F > 0.7$, the ellipse is supposed sufficiently reliable to begin the tracking with the particle filters.

The initial 3D head centroid localization can then be computed with an iterative algorithm, called the *POSIT algorithm* [32], which takes as input arguments :

- The 3D proportions of the head model (ellipsoid model) ;
- The 2D corresponding points projected in the image (detected ellipse) ;
- The camera calibration parameters.

The POSIT algorithm returns the relative position of the head in the camera coordinate system. This position can be transformed in the world coordinate system attached to the XY ground plane using :

$$P_{Head/World} = M^{-1}_{World/Cam} P_{Head/Cam} \tag{5.13}$$

with the matrix $M_{World/Cam}$ representing the known position of the world coordinate system in the camera coordinate system, the point $P_{Head/Cam}$ representing the position of the head in the camera coordinate system computed by *POSIT algorithm*, and the point $P_{Head/World}$ the desired position of the head in the world coordinate system. This position is then refined using the particle filter.

5.6.5 Tracking

To have a reliable 3D head localization, we need to precisely estimate the head projection in the image. With a conventional particle filter, many particles are needed for precision, which severely affects the computational performance and is incompatible with real-time operation. We prefer instead to use an improved particle filter based on a hierarchical scheme with several layers, similar to the annealed particle filter [34].

We chose to work with 250 particles and 4 layers, which provides a good compromise between performance and computational time.

The known velocity between two successive images is added to the particles to predict the next 3D ellipsoid localization before propagating the particles. Each layer has a different stochastic component $B_l = \left[B_{X_e}, B_{Y_e}, B_{Z_e}, B_{\theta_{X_e}}, B_{\theta_{Y_e}}\right]$ for the model propagation, sufficiently large for the first layer and decreasing for the next layers, such as $B_{l+1} = B_l/2$ with l the current layer and $l+1$ the next layer.

At the first frame, the person is supposed to be standing up, so that Z_e is approximately known and, θ_{X_e} and θ_{Y_e} are close to zero. Thus, at the beginning, to refine the initial head position, B_l will be large for the X and Y components, and of moderate size for the Z component. Our initial values are fixed to $B_l = [0.5\ 0.5\ 0.3\ 0\ 0]$ which corresponds to a diffusion of $\pm 50cm$ on the horizontal plane and $\pm 30cm$ for the Z component.

For the next images, B_l is reinitialized from the current velocity, which produces movement of the particles towards the 3D trajectory direction (a minimum for B_l components of $0.1m$ or $0.1rad$).

Figure 5.4 shows some examples of tracking with the hierarchical particle filter. The first row shows an example of a jogging video sequence (from the HumaneEva-I dataset described in section 5.7) with an horizontal diffusion of the particles due to the motion of the person. The second row shows the importance of using several layers during large motion, such as during a fall. Indeed, considering only the first layer, the ellipsoid is badly estimated, but with four layers, the ellipsoid finally recovers the correct position.

5.7 Experimental Results

Our method was first evaluated to test the precision of the 3D localization using the HumanEva-I dataset [106]. This is described in Section 5.7.1. Then, Section 5.7.2 presents an application example for fall detection realized in our laboratory.

Our method is implemented in C++ using the OpenCV library [16] and can run in quasi-real time (130ms/frame on an Intel Core 2 Duo processor (2.4 GHz), non optimized code and image size of 640x480).

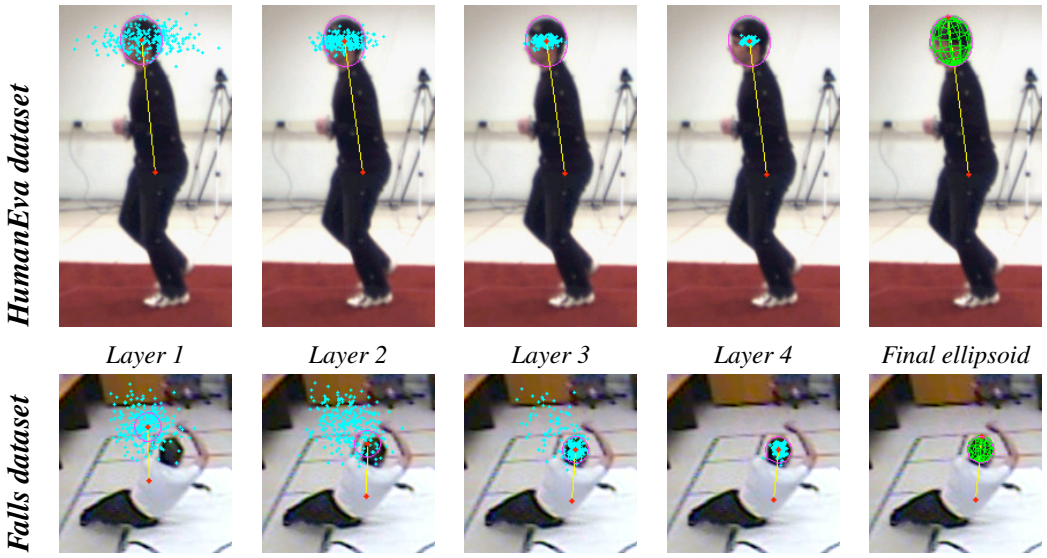


Figure 5.4 – Layers examples with a jogging sequence in the first row and a large motion during a fall in the second row. From the left to the right, we can see the particles and the mean state ellipsoid for each layer, and the final ellipsoid obtained at the end of the process.

5.7.1 3D evaluation using HumanEva dataset

The precision of our method is evaluated with the HumanEva-I dataset [106], which contains synchronized multi-view video sequences and corresponding MoCap data (ground truth 3D localizations). As our method works with a single camera, we compared the results obtained from three different view points (color cameras C_1, C_2 and C_3) using the video sequences of 3 subjects (S1, S2 and S3). The HumanEva dataset contains several actions. We used the motion sequences « walking » and « jogging » to evaluate our 3D trajectories at 30Hz, 20Hz and 10Hz.

Figure 5.5 gives an example of 3D head trajectories (top head point) obtained from different view points, and shows that the curves are similar to the MoCap trajectory. The location error is lower for the Z axis, the height is quite well estimated and the curves for the Z location show the periodic motion of the walking person. The depth error (X and Y location) is a little higher due to the ellipsoid detection and depending on the orientation of the person relative to the camera (the frontal/back and lateral views of the head are

considered identical in our 3D ellipsoid model which is not always the case in reality).

Table 5.I summarizes the 3D mean errors obtained for each subject and each camera.This corresponds to a mean error of about 5% at a 4 to 6 meters distance. As expected, when the movement was larger, the error tended to be slightly higher, but the 3D ellipsoid remained well tracked.

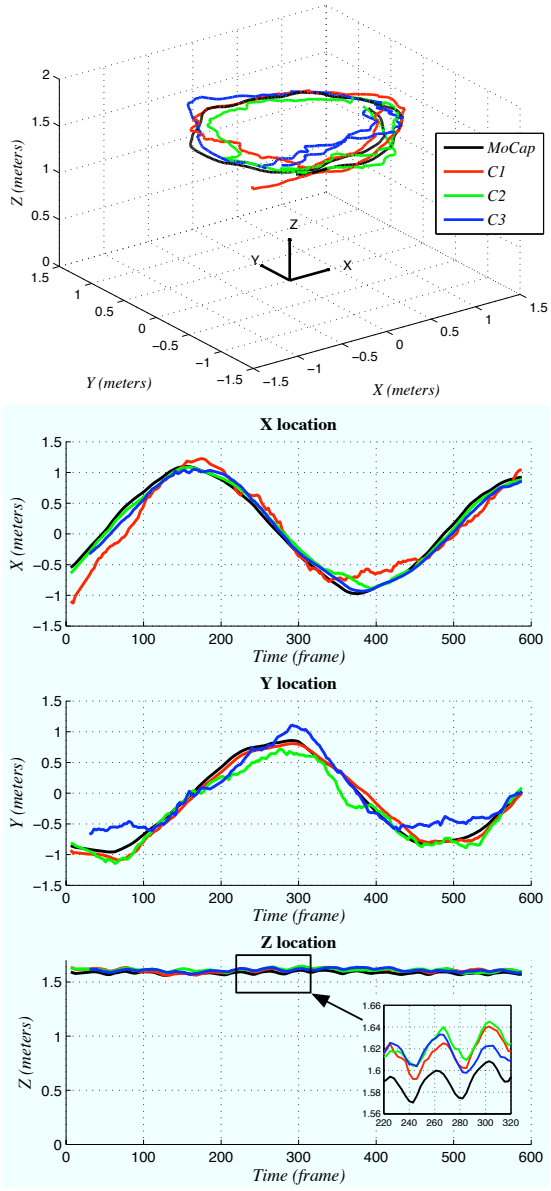
Frame rate	Camera	Walking sequences			Jogging sequences		
		S1	S2	S3	S1	S2	S3
30Hz	C1	20.6 ± 13.6	20.5 ± 7.3	21.3 ± 12.9	19.2 ± 9.5	24.1 ± 16.3	25.9 ± 15.3
	C2	17.1 ± 13	21.3 ± 7.8	23.6 ± 10.7	20.6 ± 12.4	24.5 ± 10.6	17.4 ± 9.7
	C3	17.3 ± 11.6	21.4 ± 8.4	25.9 ± 17	21.5 ± 13.6	23.7 ± 11.6	28.8 ± 17.2
20Hz	C1	20 ± 12.4	21.2 ± 7.4	19.7 ± 11.5	16.6 ± 10.3	25.4 ± 17.4	25.5 ± 16.7
	C2	15.1 ± 9.6	22.8 ± 8.5	22.8 ± 11.1	22.8 ± 10.1	26.3 ± 12	16.9 ± 10.3
	C3	19 ± 11.5	20.6 ± 8.4	28.8 ± 19.4	15.3 ± 9.8	23 ± 11.5	30 ± 19
10Hz	C1	24 ± 12.8	22.8 ± 8.2	21 ± 13	21 ± 11.4	25.9 ± 16.2	23.2 ± 16.1
	C2	18.3 ± 12.6	22.5 ± 9.9	18.3 ± 14	22.1 ± 13.8	29.2 ± 13.7	22.8 ± 16.8
	C3	22.6 ± 14	19.5 ± 8.5	28.8 ± 17.7	22 ± 10.5	24.1 ± 14.2	33.7 ± 20.7

Tableau 5.I – Mean 3D errors (in cm) obtained from walking and jogging sequences for different subjects (S1, S2, S3) and several view points (C1, C2, C3).

5.7.2 3D head trajectory for fall detection

With the HumanEva-I dataset, we have shown that our tracker works well with a standing person. In this section, we will show that our tracker is able to track large motion, such as during a fall.

Falls are one of the major risks for seniors living alone at home, often causing severe injuries. Nowadays, the favored solution to detect falls is to use wearable fall detectors like accelerometers [64, 88, 119] or help buttons [36]. But the drawback of these types of technologies is that seniors often forget to wear them, and a help button is useless if the person is unconscious after the fall. Moreover, batteries are needed for these devices and must be replaced or recharged regularly for adequate functioning. Computer vision systems offer a new promising solution to detect falls as they are perfectly adapted to acquire information on the person but also about his/her environment (location, motion or actions of the person).



3D errors	Camera 1	Camera 2	Camera 3
X error	17.7 ± 12.6	7.1 ± 4.8	8.3 ± 4.8
Y error	7.2 ± 4.4	11.9 ± 9.8	15.6 ± 12.3
Z error	2.3 ± 1.1	2.7 ± 0.9	2.4 ± 1

Figure 5.5 – 3D head trajectories for a walking sequence of subject S1 (20Hz). The table shows the mean 3D errors (in cm) for X, Y and Z location.

In a biomechanical study with wearable markers, Wu [131] showed that falls can be distinguished from normal activities using 3D velocities (computed from the trunk). Kangas *et al.* [64] showed that a triaxial accelerometer is more efficient if it is worn at the waist or the head. With a computer vision system, the head is a better feature to track than the waist, because the head is well-defined and the head is usually visible in the scene (no occlusion problem). Moreover, computer vision systems provide additional information relative to accelerometers, providing the possibility of knowing the position of the head relative to the ground.

Our method to recover the 3D head trajectory, without markers, is used here for fall detection. Two fall detection methods are explored :

Vertical velocity The duration of the critical phase of a fall is about 500 ms [86]. The vertical velocity V_v of the head centroid is then computed as a height difference on 500 ms :

$$V_v = Z_e(t) - Z_e(t - 500\,ms) \tag{5.14}$$

Head height The centroid head height relative to the ground Z_e can also be used directly as the person is supposed to be near the ground at the end of the fall.

The video sequences were acquired using low cost IP cameras (Gadspot gs-4600 [44]) with a wide angle (110 degrees) to cover the entire room. One camera was placed at the entrance of the space, the other, on the opposite wall, and the two cameras were separated by 9 meters (deep field of view). The acquisition frame rate was 30 fps, but 10 fps was sufficient to detect a fall. The image size was 720x480 pixels. Due to the wide angle, the images needed to be corrected for distortion before processing with our methodology as shown in Fig. 5.6.

Our tracker has been tested on 10 falls (forward falls, backward falls, loss of balance) which were done in different directions with respect to the two camera points of view. Notice that a mattress was used to protect the person during the simulated falls.

Figure 5.7 shows an example of vertical velocity V_v and head height Z_e obtained for a fall viewed from the two different cameras. Our 3D tracker was able to deal with the deep field of view. Even if the person was not entirely in the image (camera 1 near the

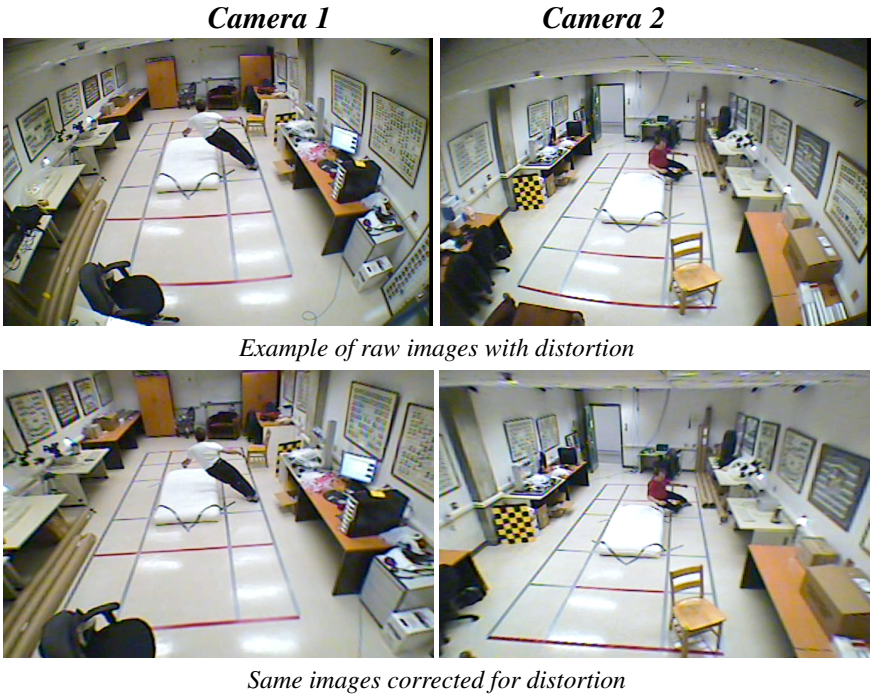


Figure 5.6 – Images from the two viewpoints before and after distortion correction.

entry), the head was detected and the tracking could be done.

With this dataset acquired with inexpensive cameras, the foreground images are noisy due to the high video compression (MPEG4) which gave artifacts on the contour of the objects. This explains why the head depth was sometimes not accurate, however the head height was well estimated as shown in Fig. 5.7. Indeed, this figure shows that the two views gave similar head height although 9 meters separates the two cameras (deep field of view).

In spite of the low image quality, our tracker was able to detect all 10 falls from the two views with the vertical velocity V_v (with a threshold at $-1m/s$). An example of lure when the person sits down is shown in Fig 5.7 and produced a vertical velocity equal to $-0.53m/s$ which was not detected as a fall. The head height Z_e can also be used to detect a fall. Indeed, when the head height is below $50cm$, the person is near the ground which is a suspicious event for an old person. Notice in Fig. 5.7 that the head height is about $1m$ when the person is seated. By considering the head height, only one fall was

not detected because of a tracking failure due to a noisy silhouette. This tracking error did not affect too much the vertical velocity with a peak of $-1.17m/s$, which was just sufficient to detect a fall.

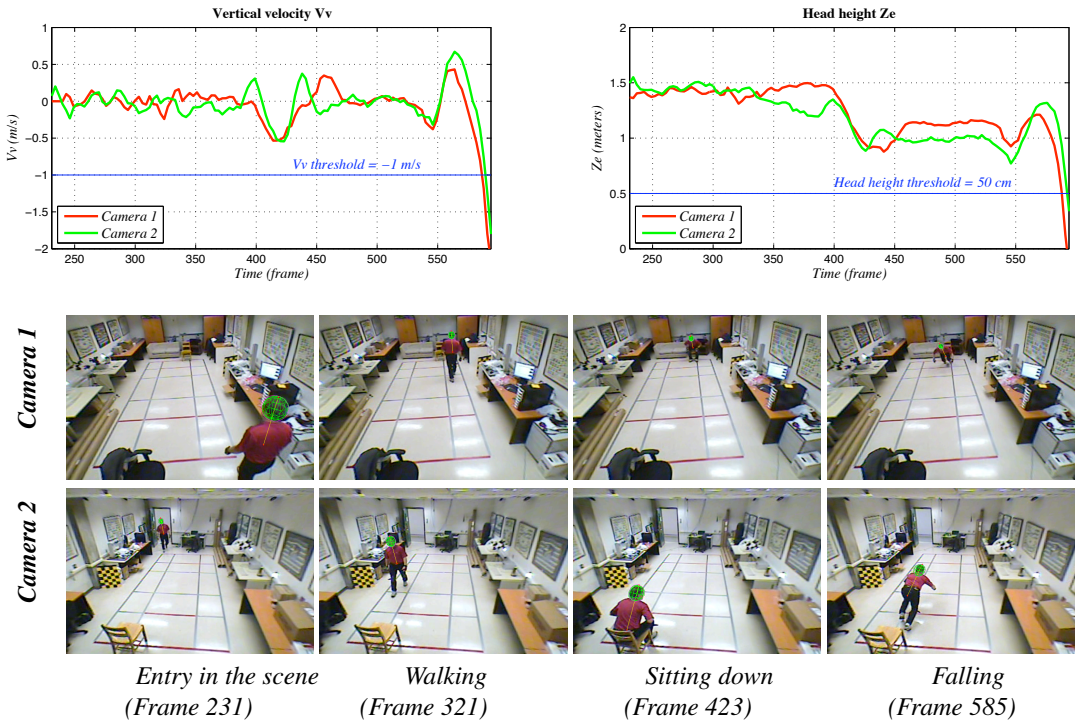


Figure 5.7 – When the person walks, the vertical velocity is about 0m/s and the head height is about $1.5m$. A little peak on the V_v curve occurs when the person sits down. The head height is about $1m$ from the ground when the person is seated. The person stands up again and falls with a high vertical velocity (V_v less than $-1m/s$) and the head height Z_e is below $50cm$ at the end of the fall.

As the 3D localization is inferred from the head foreground detection, the 3D pose can be unreliable if the head is not appropriately visible. For example, when the person falls towards the camera, the head tends to be merged with the body of the person which can give some 3D errors. However, even if the 3D pose is not well estimated, a high vertical velocity generally occurs at the beginning of a fall.

Finally, the vertical velocity is a better criterion to detect a fall than the location of the head because with the vertical velocity, the detection occurs at the beginning of the fall. In contrast, with the head height, we must wait until the head is near the ground

which can lead to failure because of occlusion or tracking problem.

5.8 Discussion and Conclusion

In this paper, we have shown that a 3D head trajectory can be computed with only a single calibrated camera. The method gave similar results for different viewpoints, different frame rates and different subjects. The 3D locations were estimated with a mean error of around 25cm (5% at 5 meters) which is sufficient for most activity recognition based on trajectories.

Our 3D tracker is automatically initialized, so that even if the tracking fails, it can be reinitialized when a good 2D head ellipse is detected. The head tracking precision is important in order to have a reliable 3D head pose. As the mean state of the particle filter is a weighted combination of all particles, the weights amplification with eq. 5.12 is important to favour the best particles. Moreover, the hierarchical particle filters with 4 layers allows us to track more precisely the head, as shown in Fig. 5.4, with a reasonable computational time.

This method can deal with body occlusions (for example with chairs or occlusion due to entry into the scene), however the head needs to be well defined and not occluded. Finally, the use of computer vision is well-adapted to detect falls and permits working with 3D velocity and position characteristics without any wearable sensor.

5.9 Acknowledgement

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

CHAPITRE 6

CONCLUSION GÉNÉRALE ET PERSPECTIVES

6.1 Bilan de nos travaux de recherche

Les travaux de cette thèse ont permis de mettre en évidence qu'il était possible de détecter des chutes chez les personnes âgées avec une seule caméra par pièce.

Nous nous sommes tout d'abord intéressés aux méthodes 2D pour détecter des chutes par l'analyse de la déformation de la silhouette de la personne lors d'une chute. Notre méthode donne d'excellents résultats comparativement aux méthodes classiques d'analyse de formes. Nous sommes capables de détecter un maximum de chutes sans générer trop de fausses alarmes. Notre système a été validé avec des vidéos réalistes comprenant de nombreuses difficultés en traitement d'images (forte compression, ombres, objets en mouvement, etc) et fonctionne quelque soit le point de vue de la caméra.

Notre deuxième travail nous a permis de montrer qu'il était possible de localiser en 3D une personne dans une pièce à l'aide d'une seule caméra calibrée. La localisation 3D est estimée avec une erreur raisonnable pour un système monoculaire (5% d'erreur à 5 mètres) ce qui est suffisant pour détecter des chutes en se basant sur la trajectoire 3D de la tête. Il est évident que notre approche nécessite que la tête soit entièrement visible dans l'image. Une occlusion de la tête sur plusieurs images peut causer un arrêt du suivi par le filtre à particules. Cependant, notre système étant entièrement automatisée, il est capable de se réinitialiser facilement en cas de perte de suivi par la détection d'une nouvelle ellipse 2D représentant la tête.

Nous sommes convaincus que les méthodes hybrides combinant des informations 2D avec un système de localisation 3D pourraient permettre de détecter un maximun de chutes en évitant de générer trop de fausses détections. Par exemple, avec notre système d'analyse de la déformation de la silhouette de la personne, nous avons parfois du mal à discriminer une chute peu forte d'une personne qui s'assoit brutalement sur la canapé. Mais cette indécision peut être levée en combinant cette information avec la localisa-

tion 3D de la tête (vitesse ou position de la tête par rapport au sol). Inversement, si notre suivi de tête 3D devient temporairement peu fiable (dû par exemple à une occlusion de la tête), nous pourrions nous raccrocher à l'information 2D du mouvement de la silhouette pour détecter une chute.

6.2 Vidéosurveillance, vie quotidienne et vie privée

Des études ont montré que les personnes âgées sont plutôt réceptives aux nouvelles technologies utilisées dans les maisons intelligentes [33]. Dans le cadre de notre projet, une étude a été menée [100] au Centre de Recherche de l'Institut Universitaire de Gériatrie de Montréal (CRIUGM) qui a montré un taux surprenant d'acceptation du système de vidéosurveillance pour la détection de chute. En effet, dans l'étude, 83.3% des proche-aidants et 86.7% des personnes âgées étaient favorables à un tel système (après avoir eu une explication claire du système).

Notre système de vidéosurveillance est entièrement automatisé et aucune image n'est diffusée à moins d'être dans une situation d'urgence. Au besoin, l'image envoyée peut être rendue floue afin de préserver l'intimité de la personne, comme par exemple pour les images provenant de la salle de bain.

Un avantage de cette technologie est qu'elle n'entrave pas la liberté de la personne dans le sens où la personne n'a pas de capteur à porter et qu'il n'y a pas de bouton à actionner. La personne âgée peut continuer à vivre dans son appartement en conservant ses habitudes quotidiennes sans avoir à penser à porter un quelconque équipement. Le système de vidéosurveillance est assez discret avec seulement une caméra par pièce fixée dans un coin. Les caméras, étant du type caméra réseau, peuvent être branchées en sans fil pour limiter l'encombrement dans l'appartement.

6.3 Perspectives d'avenir

Pour se placer dans un contexte au plus proche de la réalité, un appartement du CRIUGM[1] va être équipé de caméras. Tout un travail pour gérer le réseau de caméras et

[1] Centre de Recherche de l'Institut Universitaire de Gériatrie de Montréal

suivre une personne dans plusieurs pièces va devoir être fait. Une nouvelle bibliothèque de chutes va être acquise, avec l'aide d'un cascadeur, qui pourra servir à la communauté scientifique pour comparer les différents algorithmes de détection de chutes. D'autres aspects pourraient être étudiés : par exemple, le cas où plusieurs personnes sont présentes dans la pièce, ainsi que le passage vers l'utilisation d'un éclairage infrarouge pour avoir une application fonctionnant de nuit. De plus, il faudrait penser à une méthode de calibrage automatique sur place pour le système final.

Dans un contexte plus général, nous pourrions envisager un projet plus large de 'monitoring' d'une personne âgée sur une plus longue période. En effet, l'analyse des changements de comportement de la personne pourrait permettre de détecter des troubles du comportement signes d'une pathologie pouvant nécessiter un suivi. Concrètement, le système pourrait apprendre les habitudes de vie de la personne pendant une période suffisante (ex : 2 semaines). Une fois que le système aura appris les habitudes de la personne, il sera capable de détecter des événements anormaux dans le style de vie de la personne. Par exemple, il serait possible de savoir si la personne dort à des heures régulières et combien de temps dure son sommeil. Il serait aussi possible de vérifier la durée et le moment d'autres activités quotidiennes telles que les repas, la télévision, la salle de bain, etc. Toutes ces données vont être des indicateurs du bien être de la personne. Dans cette optique, nous pourrions réutiliser notre algorithme pour récupérer la trajectoire 3D de la personne âgée, auquel nous pourrions aussi ajouter des caractéristiques plus fines telles que la position des membres supérieurs, la forme de la silhouette, etc.

BIBLIOGRAPHIE

[1] PETS 2006. Performance evaluation of tracking and surveillance. `http ://pets2006.net`, 2010.

[2] PETS 2009. Performance evaluation of tracking and surveillance. `http ://www.visualsurveillance.org`, 2009.

[3] N. Johnson A. Galata et D. Hogg. Learning variable-length markov models of behavior. *Computer Vision and Image Understanding*, 81(3):398–413, Mars 2001.

[4] M. Alwan, P.J. Rajendran, S. Kell, D. Mack, S. Dalal, M. Wolfe et R. Felder. A smart and passive floor-vibration based fall detector for elderly. Dans *2nd Information and Communication Technologies*, volume 1, pages 1003–1007, 2006.

[5] D. Anderson, J. Keller, M. Skubic, X. Chen et Z. He. Recognizing falls from silhouettes. Dans *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS 2006)*, pages 6388–6391, 2006.

[6] Derek Anderson, Robert H. Luke, James M. Keller, Marjorie Skubic, Marilyn Rantz et Myra Aud. Linguistic summarization of video for fall detection using voxel person and fuzzy logic. *Computer Vision and Image Understanding*, 113 (1):80–89, Janvier 2009.

[7] E. Auvinet, E. Grossmann, C. Rougier, M. Dahmane et J. Meunier. Left-luggage detection using homographies and simple heuristics. Dans *Proceedings of the Ninth IEEE International Workshop on Performance Evaluation in Tracking and Surveillance (PETS)*, pages 51–58, Juin 2006.

[8] E. Auvinet, L. Reveret, A. St-Arnaud, J. Rousseau et J. Meunier. Fall detection using multiple cameras. Dans *30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2554–2557, 2008.

[9] J. L. Barron, D. J. Fleet et S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[10] C. Bauckhage, J.K. Tsotsos et F.E. Bunn. Detecting abnormal gait. Dans *The 2nd Canadian Conference on Computer and Robot Vision*, pages 282–288, Mai 2005.

[11] S. Belongie, J. Malik et J. Puzicha. Shape matching and object recognition using shape context. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[12] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. Dans *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 232–237, Juin 1998.

[13] Jean-Yves Bouguet. Camera calibration toolbox for matlab. `http ://www.vision.caltech.edu/bouguetj/calib_doc`, 2008.

[14] A.K. Bourke et G.M. Lyons. A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Medical Engineering & Physics*, 30(1):84–90, Janvier 2008.

[15] J.E. Boyd. Video phase-locked loops in gait recognition. Dans *Proc. IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 696–703, 2001.

[16] G. Bradski et A. Kaehler. *Learning OpenCV : Computer Vision with the OpenCV Library*. O'Reilly, Septembre 2008.

[17] M. Brand, N. Oliver et A. Pentland. Coupled hidden markov models for complex action recognition. Dans *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 994–999, Juin 1997.

[18] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[19] B. Caprile et V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127–139, Mars 1990.

[20] CAVIAR. Context aware vision using image-based active recognition. `http ://homepages.inf.ed.ac.uk/rbf/CAVIAR`, 2010.

[21] M.T. Chan, A. Hoogs, J. Schmiederer et M. Petersen. Detecting rare events in video using semantic primitives with hmm. Dans *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, volume 4, pages 150–154, 2004.

[22] Neena L. Chappell, Betty Havens Dlitt, Marcus J. Hollander, Jo Ann Miller et Carol McWilliam. Comparative costs of home care and residential care. *The Gerontologist*, 44:389–400, 2004.

[23] Hammadi Nait Charif et Stephen J. McKenna. Tracking the activity of participants in a meeting. *Machine Vision and Applications*, 17(2):83–93, 2006.

[24] Robert Collins, Alan Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yanghai Tsin, David Tolliver, Nobuyoshi Enomoto et Osamu Hasegawa. A system for video surveillance and monitoring. Rapport technique, tech. report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, Mai 2000.

[25] A. Corradini. Dynamic time warping for off-line recognition of a small gesture vocabulary. Dans *IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 82–89, 2001.

[26] A. Criminisi. *Accurate visual metrology from single and multiple uncalibrated images*. Springer-Verlag London Ltd., Septembre 2001.

[27] A. Criminisi. Single-view metrology : Algorithms and applications. Dans *Proceedings of the 24th DAGM Symposium on Pattern Recognition table of contents*, pages 224–239, 2002.

[28] Rita Cucchiara, Costantino Grana, Massimo Piccardi et Andrea Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1337–1342, Octobre 2003.

[29] F. Cupillard, A. Avanzi, F. Brémond et M. Thonnat. Video understanding for metro surveillance. Dans *The IEEE ICNSC 2004 in the special session on Intelligent Transportation Systems*, volume 1, pages 186–191, 2004.

[30] Mohamed Dahmane et Jean Meunier. Real-time video surveillance with self-organizing maps. Dans *The 2nd Canadian Conference on Computer and Robot Vision (CRV'05)*, pages 136–143, 2005.

[31] Ministère de la Santé et des Services sociaux du Québec. Portrait de santé du québec et de ses régions. Rapport technique, Institut national de santé publique du Québec, 2006.

[32] D.F. Dementhon et L.S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1-2):123–141, Juin 1995.

[33] G. Demiris, M. Rantz, M.A. Aud, K. Marek, H. Tyrer, M. Skubic et A. Hussam. Older adults' attitudes towards and perceptions of 'smart home' technologies : a pilot study. *Medical Informatics and the Internet in Medicine*, 29(2):87–94, Juin 2004.

[34] J. Deutscher, A. Blake et I.D. Reid. Articulated body motion capture by annealed particle filtering. Dans *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 126–133, 2000.

[35] F. Devernay et O. Faugeras. Straight lines have to be straight. *Machine Vision and Applications*, 13(1):14–24, 2001.

[36] DirectAlert. Wireless emergency response system. `http ://www.directalert.ca/emergency/help-button.php`, 2010.

[37] G. Dreyfus, J.M. Martinez, M. Samuelides, M.B. Gordon, F. Badran, S. Thiria et L. Hérault. *Réseaux de neurones : méthodologies et applications*. Éditions Eyrolles, 2002.

[38] I.L. Dryden et K.V. Mardia. *Statistical Shape Analysis*. John Wiley and Sons, Chichester, 1998.

[39] Richard O. Duda, Peter E. Hart et David G. Stork. *Pattern Classification*. Wiley, 2nd édition, 2001.

[40] Electrophysics. How night vision works. `http ://www.hownightvisionworks.com`, 2009.

[41] H. Freeman. On the encoding of arbitrary geometric configurations. *IEEE Transactions on Electronic Computers*, EC-10(2):260–268, 1961.

[42] L.M. Fuentes et S.A. Velastin. People tracking in surveillance applications. Dans *Proceedings of the 2nd IEEE Workshop on Performance Evaluation of Tracking and Surveillance (PETS2001)*, 2001.

[43] H. Fujiyoshi et A. Lipton. Real-time human motion analysis by image skeletonization. Dans *In Proceedings of the IEEE Workshop on Applications of Computer Vison, WACV'98*, pages 15–21, 1998.

[44] Gadspot. Ip camera. `http ://gadspot.com`, 2010.

[45] J. Gao, A.G. Hauptmann, A. Bharucha et H.D. Wactlar. Dining activity analysis using a hidden markov model. Dans *Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004*, volume 2, pages 915–918, 2004.

[46] B.M. Haralick, C.-N. Lee, K. Ottenberg et M. Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13(3):331–356, Décembre 1994.

[47] I. Haritaoglu, D. Harwood et L.S. Davis. W^4 : real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, Août 2000.

[48] R. I. Hartley et A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2nd édition, 2004.

[49] R.I. Hartley. An algorithm for self calibration from several views. Dans *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'94*, pages 908–912, Juin 1994.

[50] Lykele Hazelhoff, Jungong Han et Peter H. N. de With. Video-based fall detection in the home using principal component analysis. Dans *Advanced Concepts for Intelligent Vision Systems*, volume 1, pages 298–309, 2008.

[51] J. Heikkila et O. Silven. A four-step camera calibration procedure with implicit image correction. Dans *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, ICPR'97*, pages 1106–1112, Juin 1997.

[52] M. Hild. Estimation of 3d motion trajectory and velocity from monocular image sequences in the context of human gait recognition. Dans *International Conference on Pattern Recognition (ICPR)*, volume 4, pages 231–235, 2004.

[53] Victoria J. Hodge et Jim Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22:85–126, 2004.

[54] M. Hollander et N. Chappell. Final report of the national evaluation of the cost-effectiveness of home care. Rapport technique, Health Transition Fund, Health Canada, 2002.

[55] P. Horain et M. Bomb. 3D model based gesture acquisition using a single camera. Dans *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV 2002)*, pages 158–162, 2002.

[56] T. Horprasert, D. Harwood et L.S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. Dans *Proceedings of the 7th IEEE International Conference on Computer Vision, Frame Rate Workshop (ICCV '99)*, pages 1–19, Septembre 1999.

[57] iLife. Fall detection sensor. `http ://www.falldetection.com`, 2010.

[58] M. Isard et A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[59] S. Jabri, Z. Duric, A. Rosenfeld et H. Wechsler. Detection and location of people in video images using adaptive fusion of color and edge information. Dans *Proceedings of International Conference on Pattern Recognition (ICPR'00)*, volume 4, pages 627–630, 2000.

[60] F. Jean, R. Bergevin et A.B. Albu. Body tracking in human walk from monocular video sequences. Dans *The 2nd Canadian Conference on Computer and Robot Vision, CRV'05*, pages 144–151, Mai 2005.

[61] Q. Ji et X. Yang. Real-time eye, gaze, and face pose tracking for monitoring driver vigilance. *Real-Time Imaging*, 8(5):357–377, Octobre 2002.

[62] L. Jiao, Yi Wu, G. Wu, E.Y. Chang et Y.-F. Wang. Anatomy of a multicamera video surveillance system. *Multimedia Systems*, 10(2):144–163, Août 2004.

[63] Ning Jin et Farzin Mokhtarian. Human motion recognition based on statistical shape analysis. Dans *Proc. IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS 2005)*, pages 4–9, 2005.

[64] Maarit Kangas, Antti Konttila, Per Lindgren, Ilkka Winblad et Timo Jämsä. Comparison of low-complexity fall detection algorithms for body attached accelerometers. *Gait & Posture*, 28(2):285–291, 2008.

[65] D.M. Karantonis, M.R. Narayanan, M. Mathie, N.H. Lovell et B.G. Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE Transactions on Information Technology in Biomedicine*, 10(1):156–167, 2006.

[66] H. Kawanaka, H. Fujiyoshi et Y. Iwahori. Human head tracking in three dimensional voxel space. Dans *International Conference on Pattern Recognition (ICPR)*, volume 3, pages 826–829, 2006.

[67] K. Kim, T.H. Chalidabhongse, D. Harwood et L. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11(3):172–185, Juin 2005.

[68] Y. Kobayashi, D. Sugimura, K. Hirasawa, N. Suzuki, H. Kage, Y. Sato et A. Sugimoto. 3d head tracking using the particle filter with cascaded classifiers. Dans *Proc. of British Machine Vision Conference (BMVC)*, pages 37–46, 2006.

[69] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955.

[70] T. Lee et A. Mihailidis. An intelligent emergency response system : preliminary development and testing of automated fall detection. *Journal of telemedicine and telecare*, 11(4):194–198, 2005.

[71] R. Diaz De Leon et L.E. Sucar. Human silhouette recognition with fourier descriptors. Dans *15th International Conference on Pattern Recognition*, volume 3, pages 709–712, Septembre 2000.

[72] A. Leykin et M. Tuceryan. A vision system for automated customer tracking for marketing analysis : Low level feature extraction. Dans *In Proc. of the International Workshop on Human Activity Recognition and Modelling, HAREM'05*, pages 1–7, 2005.

[73] S.Z. Li, Chu RuFeng, Liao ShengCai et Zhang Lun. Illumination invariant face recognition using near-infrared images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):627–639, 2007.

[74] Markos Markou et Sameer Singh. Novelty detection : a review - part 1 : statistical approaches. *Signal Processing*, 83(12):2481–2497, Décembre 2003.

[75] Markos Markou et Sameer Singh. Novelty detection : a review - part 2 : : neural network based approaches. *Signal Processing*, 83(12):2499–2521, Décembre 2003.

[76] S.J. Maybank et O.D. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151, Août 1992.

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

[78] F. Mokhtarian. Silhouette-based isolated object recognition through curvature scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17 (5):539–544, 1995.

[79] F. Moreno, A. Tarrida, J. Andrade-Cetto et A. Sanfeliu. 3d real-time head tracking fusing color histograms and stereovision. Dans *16th International Conference on Pattern Recognition (ICPR)*, volume 1, pages 368–371, 2002.

[80] Ian T. Nabney. *NETLAB - Algorithms for Pattern Recognition*. Springer, 2001.

[81] H. Nait-Charif et S.J. McKenna. Activity summarisation and fall detection in a supportive home environment. Dans *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, volume 4, pages 323–326, 2004.

[82] J.C. Nascimento, M.A.T. Figueiredo et J.S. Marques. Recognition of human activities using space dependent switched dynamical models. Dans *International Conference on Image Processing, ICIP 2005*, volume 3, pages 852–855, 2005.

[83] N.T. Nguyen, D.Q. Phung, S. Venkatesh et H.H. Bui. Learning and detecting activities from movement trajectories using the hierarchical hidden markov models. Dans *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 955–960, 2005.

[84] R. Niels et L. Vuurpijl. Dynamic time warping applied to tamil character recognition. Dans *Eighth International Conference on Document Analysis and Recognition*, volume 2, pages 730–734, 2005.

[85] N. Noury, A. Fleury, P. Rumeau, A.K. Bourke, G.O. Laighin, V. Rialle et J.E. Lundy. Fall detection - principles and methods. Dans *29th Annual International*

Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), pages 1663–1666, 2007.

[86] N. Noury, P. Rumeau, A.K. Bourke, G. ÓLaighin et J.E. Lundy. A proposal for the classification and evaluation of fall detectors. *IRBM*, 29(6):340–349, Décembre 2008.

[87] K. Nummiaro, E. Koller-Meier et L. Van Gool. An adaptive color-based particle filter. *Image and Vision Computing*, 21(1):99–110, Janvier 2003.

[88] M.N. Nyan, Francis E.H. Tay et E. Murugasu. A wearable system for pre-impact fall detection. *Journal of Biomechanics*, 41(16):3475–3481, Décembre 2008.

[89] Division of Aging et Seniors. Canada's aging population. Rapport technique, Public Health Agency of Canada, 2002.

[90] Division of Aging et Seniors. Report on senior's falls in canada. Rapport technique, Public Health Agency of Canada, 2005.

[91] OpenCV. Open source computer vision library. `http ://opencv.willowgarage.com/wiki`, 2010.

[92] J. Owens et A. Hunter. Application of the self-organizing map to trajectory classification. Dans *Proceedings of the Third IEEE International Workshop on Visual Surveillance (VS'2000)*, pages 77–83, 2000.

[93] N. Paragios et R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):266–280, 2000.

[94] Poséidon. Le troisième oeil du maître-nageur. `http ://www.poseidon-tech.com/fr/`, 2010.

[95] L. Quan et Z. Lan. Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):774–780, 1999.

[96] C. Rao, A. Gritai, M. Shah et T. Syeda-Mahmood. View-invariant alignment and matching of video sequences. Dans *Ninth IEEE International Conference on Computer Vision*, volume 2, pages 939–945, Octobre 2003.

[97] C. Rao, A. Yilmaz et M. Shah. View-invariant representation and recognition of actions. *International Journal of Computer Vision*, 50(2):203–226, 2002.

[98] C. Rougier, J. Meunier, A. St-Arnaud et J. Rousseau. Monocular 3d head tracking to detect falls of elderly people. Dans *International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6384–6387, 2006.

[99] C. Rougier, J. Meunier, A. St-Arnaud et J. Rousseau. Fall detection from human shape and motion history using video surveillance. Dans *IEEE 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW)*, volume 2, pages 875–880, 2007.

[100] J. Rousseau, A. St-Arnaud, F. Ducharme, J. St-Arnaud, J. Meunier, S. Turgeon-Londei et M. Jobidon. Videomonitoring at home : Do you want it ? Dans *Canadian Association of Occupational Therapy Annual conference - CAOT Conference 2008 : Exploring the frontiers of occupation*, Juin 2008.

[101] Comité sénatorial spécial sur le vieillissement. Rapport final sur le vieillissement de la population, un phénomène à valoriser. Rapport technique, Sénat Canada, 2009.

[102] Guangyi Shi, Cheung Shing Chan, Wen Jung Li, Kwok-Sui Leung, Yuexian Zou et Yufeng Jin. Mobile human airbag system for fall protection using mems sensors and embedded svm classifier. *IEEE Sensors Journal*, 9(5):495–503, 2009.

[103] Nils T. Siebel et Steve Maybank. Real-time tracking of pedestrians and vehicles. Dans *2nd IEEE Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, 2001.

[104] Nils T. Siebel et Steve Maybank. The advisor visual surveillance system. Dans *In Proceedings of the ECCV 2004 workshop Applications of Computer Vision*, pages 103–111, 2004.

[105] N.T. Siebel et S. Maybank. The application of colour filtering to real-time person tracking. Dans *Proceedings of the 2nd European Workshop on Advanced Video-Based Surveillance Systems (AVBS2001)*, pages 227–234, Septembre 2001.

[106] L. Sigal et M. J. Black. Humaneva : Synchronized video and motion capture dataset for evaluation of articulated human motion. Rapport technique CS-06-08, Brown University, Department of Computer Science, Providence, RI, 2006.

[107] A. Sixsmith et N. Johnson. A smart sensor to detect the falls of the elderly. *IEEE Pervasive Computing*, 3(2):42–47, 2004.

[108] C. Slama, C. Theurer et S.W. Henriksen. *Manual of photogrammetry*. American Society of Photogrammetry and Remote Sensing, Falls Church, Virginia, USA, 4th edition édition, 1980.

[109] C. Sminchisescu et B. Triggs. Covariance scaled sampling for monocular 3D body tracking. Dans *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01)*, volume 1, pages 447–454, Décembre 2001.

[110] K. Sobottka et I. Pitas. Face localization and facial feature extraction based on shape and color information. Dans *International Conference on Image Processing (ICIP)*, volume 3, pages 483–486, 1996.

[111] P. Srinivasan, G. Qian, D. Birchfield et A. Kidané. Design of a pressure sensitive floor for multimodal sensing. Dans *Proceedings of the Ninth International Conference on Information Visualisation*, pages 41–46, Juillet 2005.

[112] G.P. Stein. Accurate internal camera calibration using rotation, with analysis of sources of error. Dans *Fifth International Conference on Computer Vision, ICCV'95*, pages 230–236, Juin 1995.

[113] B. Stenger, P.R.S. Mendonça et R. Cipolla. Model-based hand tracking using an unscented kalman filter. Dans *Proc. BMVC*, volume 1, pages 63–72, Septembre 2001.

[114] D. Thirde, M. Borg, J. Ferryman, F. Fusier, V. Valentin, F. Bremond et M. Thonnat. Video event recognition for aircraft activity monitoring. Dans *In Proceedings of 8th IEEE International Conference on Intelligent Transportation Systems*, pages 1102–1107, Septembre 2005.

[115] N. Thome, S. Miguet et S. Ambellouis. A real-time, multiview fall detection system : A LHMM-based approach. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1522–1532, 2008.

[116] B.U. Töreyin, Y. Dedeoglu et A.E. Çetin. HMM based falling person detection using both audio and video. Dans *Proceedings of the International Workshop on Computer Vision in Human-Computer Interaction (ICCV-HCI)*, volume 3766, Octobre 2005.

[117] A. Treptow, G. Cielniak et T. Duckett. Active people recognition using thermal and grey images on a mobile security robot. Dans *International Conference on Intelligent Robots and Systems*, pages 2103–2108, 2005.

[118] R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, Août 1987.

[119] Tunstall. Fall detector.
`http ://www.tunstall.co.uk/Our-products`, 2010.

[120] Jorge Usabiaga, George Bebis, Ali Erol, Mircea Nicolescu et Monica Nicolescu. Recognizing simple human actions using 3d head movement. *Computational Intelligence*, 23(4):484–496, 2007.

[121] M. Valera et S.A. Velastin. Intelligent distributed surveillance systems : a review. *IEE Proceedings - Vision, Image and Signal Processing*, 152(2):192–204, 2005.

[122] M. Valin, J. Meunier, A. St-Arnaud et J. Rousseau. Video surveillance of medication intake. Dans *28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6396–6399, 2006.

[123] A. Veeraraghavan, A.K. Roy-Chowdhury et R. Chellappa. Matching shape sequences in video with applications in human movement analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1896–1909, Décembre 2005.

[124] H. Veeraraghavan, O. Masoud et N.P. Papanikolopoulos. Computer vision algorithms for intersection monitoring. *IEEE Transactions on Intelligent Transportation Systems*, 4(2):78–89, 2003.

[125] Visonic. Fall detector. `http ://www.visonic.com/seniors`, 2010.

[126] Liang Wang, Tieniu Tan, Weiming Hu et Huazhong Ning. Automatic gait recognition based on statistical shape analysis. *IEEE transactions on image processing*, 12(9):1120–1131, 2003.

[127] G. Welch et G. Bishop. An introduction to the kalman filter. Rapport technique Technical report TR95-041, University of North Carolina at Chapel Hill, 2006.

[128] J. Weng, P. Cohen et M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):965–980, Octobre 1992.

[129] C. R. Wren, A. Azarbayejani, T. Darrell et A. P. Pentland. Pfinder : Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[130] Chen Wu et H. Aghajan. Head pose and trajectory recovery in uncalibrated camera networks - region of interest tracking in smart home applications. Dans *ACM/IEEE International Conference on Distributed Smart Cameras*, pages 1–7, 2008.

[131] G. Wu. Distinguishing fall activities from normal activities by velocity characteristics. *Journal of Biomechanics*, 33(11):1497–1500, 2000.

[132] M. Xu, J. Orwell, L. Lowey et D.J. Thirde. Architecture and algorithms for tracking football players with multiple cameras. *IEE Proceedings - Vision, Image and Signal Processing*, 152(2):232–241, 2005.

[133] M.-H. Yang et N. Ahuja. Recognizing hand gesture using motion trajectories. Dans *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, Juin 1999.

[134] Z. Zhang. A flexible new technique for camera calibration. Rapport technique Technical Report MSR-TR-98-71, Microsoft Research, Décembre 1998.

[135] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI*, 22(11):1330–1334, Novembre 2000.

[136] Z. Zhang. Camera calibration with one-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 2004*, 26(7):892–899, Juillet 2004.

[137] Y. Zigel, D. Litvak et I. Gannot. A method for automatic fall detection of elderly people using floor vibrations and sound - proof of concept on human mimicking doll falls. *IEEE Transactions on Biomedical Engineering*, 56(12):2858–2867, 2009.

Annexe I

Bibliothèque de vidéos de chutes

Nous avons travaillé sur des vidéos de chutes réalisées dans les locaux du DIRO, et simulées par notre spécialiste des chutes, Alain St-Arnaud. Pour nos premières expérimentations, nous avions utilisé des caméras de type webcams branchées en USB. Le flux vidéo n'étant pas stable, nous nous sommes tournés vers les caméras IP qui sont finalement beaucoup plus intéressantes pour gérer tout un réseau de caméras dans un appartement. Nos caméras IP(Gadspot gs-4600 [44]) ont un grand angle de vue permettant de couvrir toute la pièce. La fréquence d'acquisition est de 30 images par seconde et la taille de l'image est de 720x480 pixels.

La Figure I.1 montre la configuration des différentes caméras dans la pièce.

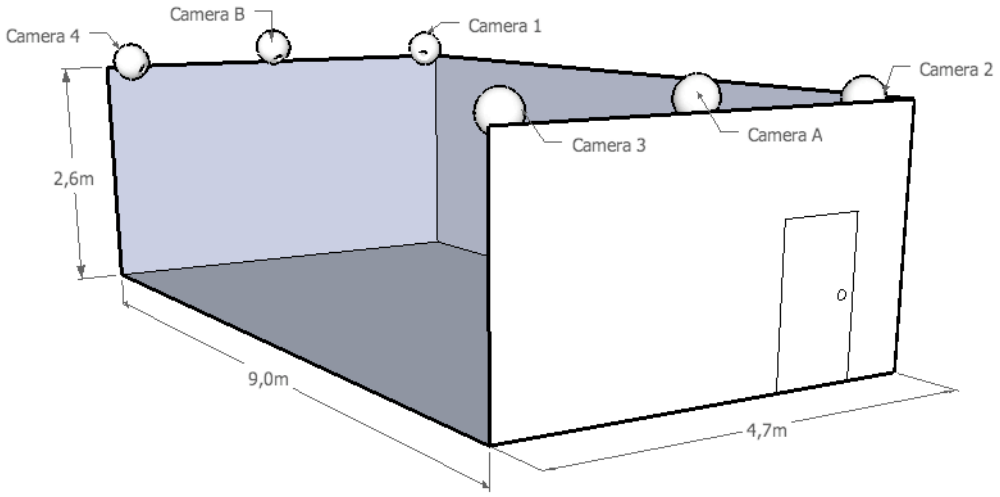


Figure I.1 – Configuration des caméras de la vidéothèque de chutes. Les caméras 1, 2, 3 et 4 dans les quatre angles de la pièce ont été utilisées dans le premier article (voir Chapitre 3). Après correction de la distorsion, les caméras A et B sont celles qui permettent d'avoir une vue entière de la pièce et ont donc été utilisées pour le deuxième article (voir Chapitre 5) où la correction de la distorsion est nécessaire.

Les caméras utilisées étant des caméras bas coût, nous avons été confrontés à plu-

sieurs problèmes dont une forte compression vidéo (MPEG4) qui génére des artéfacts sur les contours des objets, ainsi qu'une illumination variable dans le flux vidéo.

De plus, pour être au plus proche des conditions réelles, nos vidéos contiennent différents types de difficultés pouvant générer des problèmes de segmentation :

- Des ombres et des réflexions qui peuvent être détectées comme des objets en mouvement lors de la segmentation.
- Le fond n'est pas uniforme, de nombreux objets sont présents dans l'arrière plan.
- La personne déplace des objets dans la scène (ex : sac, carton, balai, manteau), enlève et dépose son manteau. Un objet déplacé devient un objet en mouvement avec une méthode de segmentation d'images. Il faut donc penser à mettre à jour le fond avec le nouvel objet si celui-ci devient immobile en faisant attention de ne pas mettre à jour le fond avec une personne endormie sur le canapé.
- Occlusions de la personne avec des objets (ex : chaise, canapé) ou avec le bord de l'image (entrée/sortie de la scène).
- Différents vêtements de différentes couleurs (parfois identique à la couleur du fond), enlève et remet son manteau.

La Figure I.2 montre des exemples de chutes et d'activités normales vues par les différents points de vue.

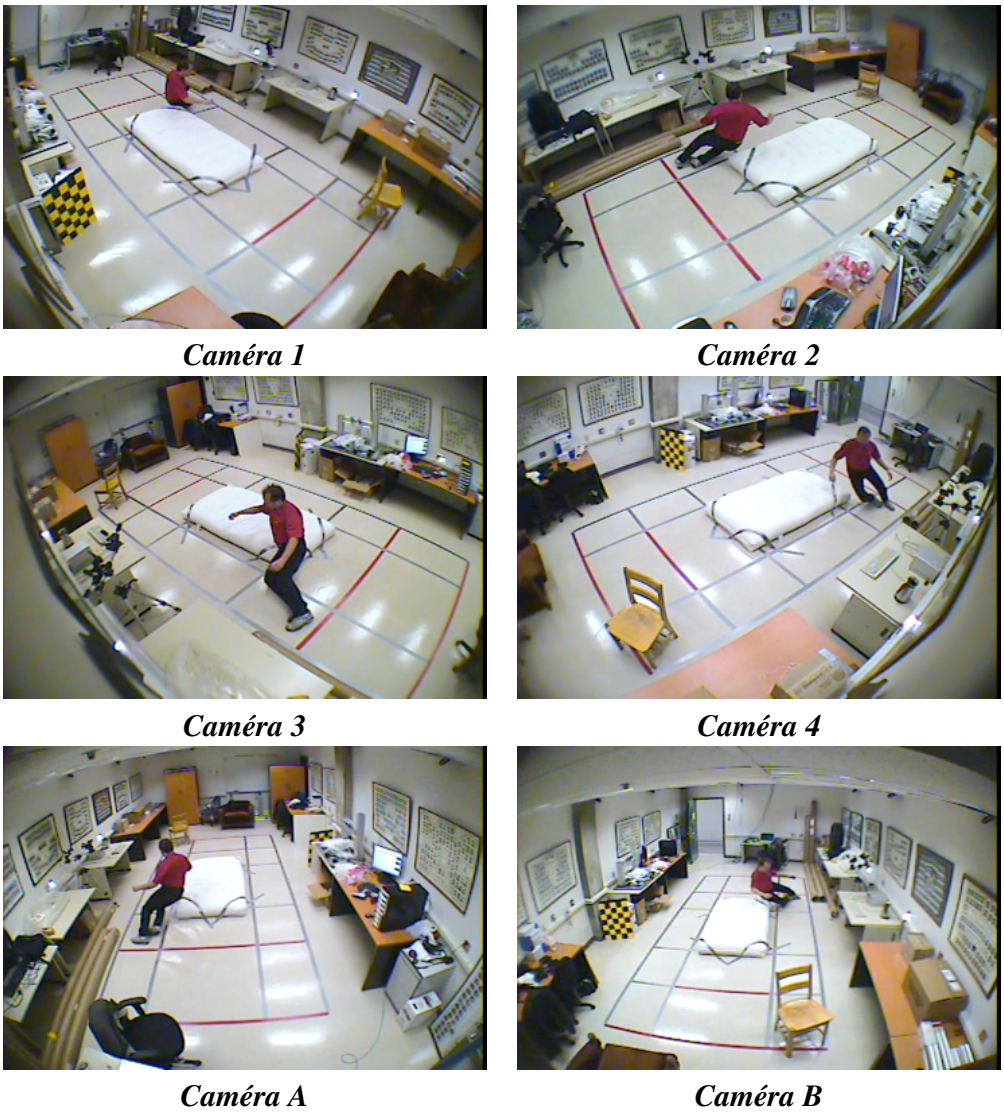


Figure I.2 – Exemple d'une même chute vue par nos différentes caméras.

Annexe II

Bibliothèque de vidéos HumanEva

La bibliothèque HumanEva-I (HumanEva : Synchronized Video and Motion Capture Dataset for Evaluation of Articulated Human Motion) a été proposée par Sigal et Black [106] pour l'évaluation de l'estimation de pose 3D d'une personne.

Les données ont été acquises avec un système multi-caméras synchronisé associé à un système de Motion Capture (MoCap) pour récupérer la « vraie » localisation 3D servant de référence aux calculs d'erreurs 3D.

Plusieurs caméras (3 couleurs C_1, C_2, C_3 et 4 en niveaux de gris BW_1, BW_2, BW_3, BW_4) sont disposées autour de la scène comme montré sur la Fig. II.1. Les caméras ont été calibrées en utilisant la toolbox Camera Calibration pour Matlab [13] basée sur une méthode standard de calibrage avec un damier.

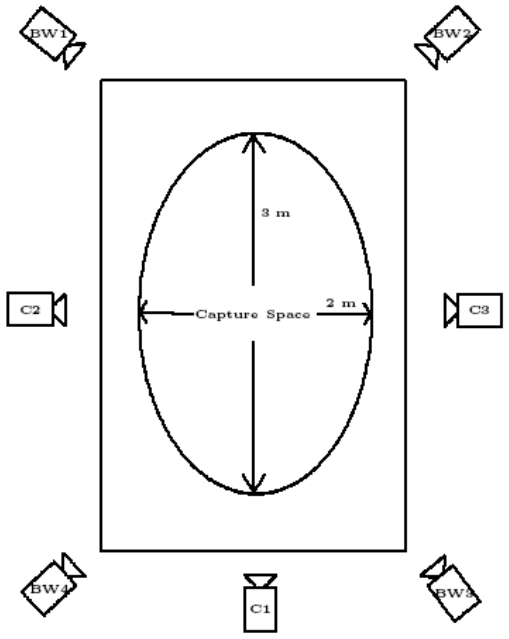


Figure II.1 – Configuration des caméras de la vidéothèque HumanEva [106]

Les données vidéo ont été acquises à 60Hz. Diférentes actions sont effectuées par quatre personnes (une femme (S_1) et 3 hommes (S_2, S_3, S_4)). Notez que nous n'utilisons pas les données du sujet S_4 dans nos expériences car les données MoCap ne sont pas didsponibles pour ce sujet.

L'ensemble de données HumanEva contient différentes actions :

– Des séquences de mouvement : « walking » and « jog ».
– Des séquences de gestes : « throw/catch », « gestures » and « box ».
– Des séquences diverses « combo » (qui ne contiennent pas de données MoCap).

Un exemple de séquence de marche (« walking ») est montré sur la Fig. II.2.

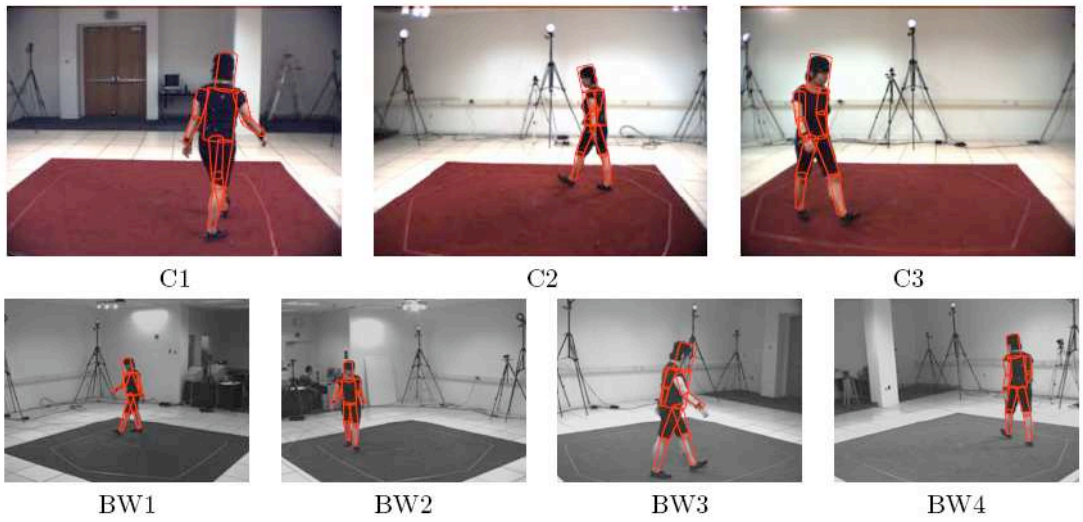


Figure II.2 – Séquence de marche (sujet S_1) vue par les différentes caméras avec les données MoCap superposées. [106]

Annexe III

Analyse ROC

L'analyse ROC (Receiver Operating Characteristic) est un outil très utilisé pour quantifier la qualité d'un classifieur. Pour tracer la courbe ROC, le nombre d'évènements réussis ou manqués doit être comptabilisé pour chaque seuil du détection comme présenté dans le tableau III.I.

	Chute	Non chute
Détecté « chute »	Vrai Positif (VP)	Faux Positif (FP)
Aucune détection	Faux Négatif(FN)	Vrai Négatif (VN)

Tableau III.I – Classification des évènements.

Il est alors possible de calculer la sensitivité et la spécificité du système :

- **Sensitivité** S_e

  La sensitivité mesure la probabilité qu'une chute soit détectée. Plus la sensitivité est forte, plus le système sera capable de détecter des chutes.

$$S_e = \frac{VP}{VP+FN} \tag{III.1}$$

- **Spécificité** S_p

  La spécificité mesure la probabilité qu'un évènement normal ne soit pas détecté comme une chute. Un système efficace de détection de chute générant peu d'alarmes aura une spécificité forte.

$$S_p = \frac{VN}{FP+VN} \tag{III.2}$$

La courbe ROC correspond à un tracé du taux des vrais positifs (sensitivité) en fonction du taux de faux positifs (1-spécificité). En faisant varier le seuil de détection, un ensemble de paire (S_p, S_e) est obtenu servant au tracé de la courbe ROC comme montré sur la Fig. III.1. Un moyen pour comparer des courbes ROC est de calculer l'aire sous la courbe ROC (AUC pour Area Under the Curve) qui doit tendre vers 1 lorsque le clas-

sifieur est efficace. Un autre moyen pour comparer des classifieurs est d'utiliser le taux d'erreur égal (EER pour Equal Error Rate) qui correspond au point de la courbe ROC tel que $S_e = S_p$.

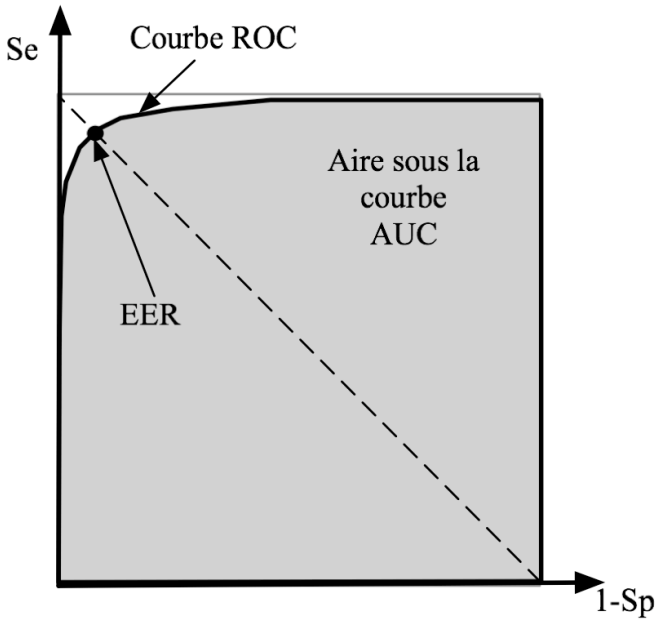


Figure III.1 – Exemple de courbe ROC avec deux caractéristiques importantes : l'aire sous la courbe (AUC) et le point correspondant au taux d'erreur égal (EER).

Le seuil de détection sera choisi par un compromis entre la sensitivité et la spécificité du système, à savoir si l'on préfère détecter un maximum de chutes quitte à avoir un certain nombre de fausses détections, ou alors si l'on préfère limiter la quantité de fausses alarmes.

Annexe IV

Projection de l'ellipsoïde 3D dans le plan image

La tête est représentée par une ellipsoïde 3D, comme montrée sur la figure IV.1, qui est projetée en une ellipse dans le plan image. L'ellipsoïde reprśentant la tête a pour équation : $\frac{x^2}{B^2} + \frac{y^2}{B^2} + \frac{z^2}{A^2} = 1$.

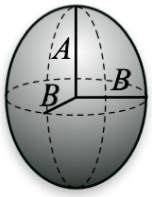


Figure IV.1 – Ellipsoïde 3D avec pour demi-grand axe A et demi-petit axe B.

L'ellipsoïde s'écrit aussi sous la forme matricielle en la matrice Q_H dans le repère de l'ellipsoïde :

$$Q_H = \begin{pmatrix} \frac{1}{B^2} & 0 & 0 & 0 \\ 0 & \frac{1}{B^2} & 0 & 0 \\ 0 & 0 & \frac{1}{A^2} & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{IV.1}$$

Dans le repère caméra, l'ellipsoïde est représentée par une quadrique Q_C telle que pour un point (x,y,z) appartenant à l'ellipsoïde, on a :

$$[x,y,z,1]^T Q_C [x,y,z,1] = 0 \tag{IV.2}$$

La matrice Q_C est obtenue à partir de la matrice Q_H en utilisant la matrice de projection P, soit $Q_C = P^{-1T} Q_H P^{-1}$. La Figure IV.2 montre les différentes transformations composant la matrice de projection $P = KM_{ext}M_{Tête/Monde}$.

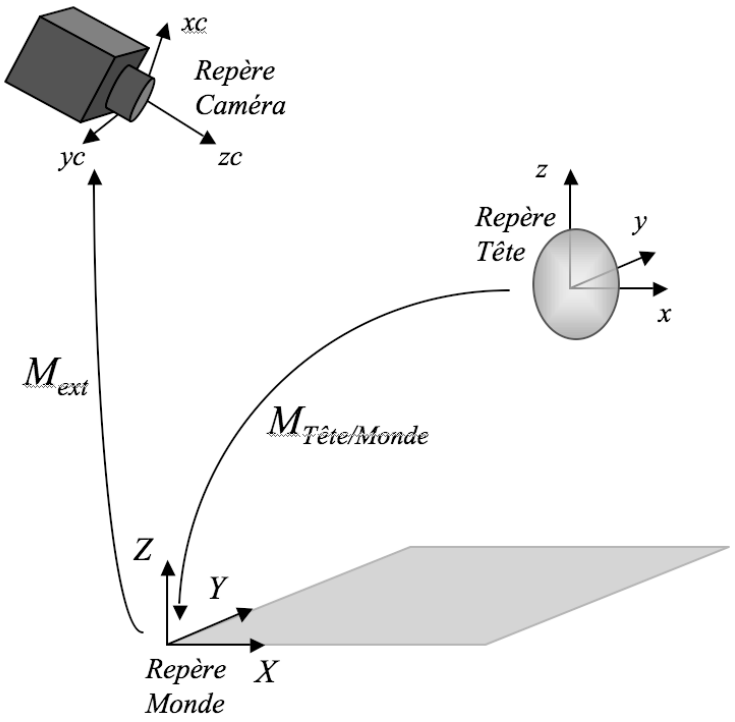


Figure IV.2 – Projection de l'ellipsoïde dans le repère caméra.

La matrice K représente les paramètres internes de la caméra :

$$K = \begin{pmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{IV.3}$$

avec la distance focale (f_x, f_y) en x et en y et le centre optique (u_0, v_0) en pixels.

La matrice M_{ext} représente la matrice des paramètres externes, c'est à dire la position

du repère monde par rapport au repère caméra :

$$M_{ext} = \begin{pmatrix} R & T \\ 0\,0\,0 & 1 \end{pmatrix} \tag{IV.4}$$

avec R et T respectivement la matrice de rotation et le vecteur de translation [135].

Quant à $M_{Tête/Monde}$, elle est déterminée lors du suivi avec le filtre à particules à partir des paramètres d'une ellipsoïde, c'est à dire le centre de l'ellipsoïde (X_E, Y_E, Z_E) et la rotation de l'ellipsoïde $(\theta_{Xe}, \theta_{Ye})$.

La conique C dans le plan image, montrée sur la Figure IV.3, est obtenue à partir de Q_C [48, 113], et permet de déterminer les paramètres de l'ellipse représentant la tête dans le plan image :

$$C = Q_{C_{44}} Q_{C_{1:3,1:3}} - Q_{C_{1:3,4}} Q_{C_{1:3,4}}^T \tag{IV.5}$$

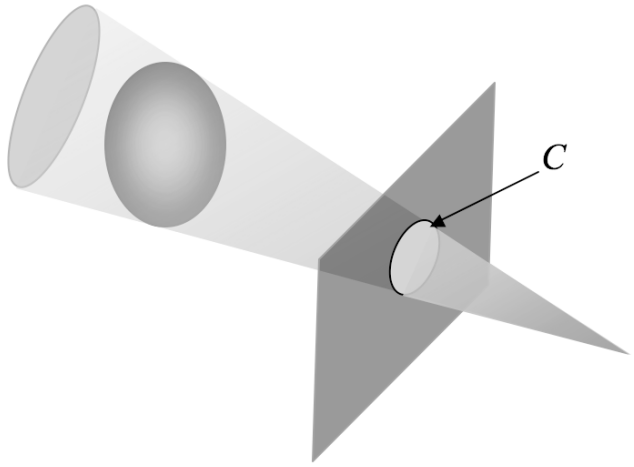


Figure IV.3 – Conique obtenue dans le plan image.