Reconnaissance d'activités humaines à partir de séquences multi-caméras : application à la détection de chute de personne

Ange Mikaël Mousse

Université d'Abomey-Calavi
Institut de Mathématiques et de Sciences Physiques

Université du Littoral Côte d'Opale

Reconnaissance d'activités humaines à partir de séquences multi-caméras : application à la détection de chute de personne

THÈSE

présentée et soutenue publiquement le 10 Décembre 2016

pour l'obtention du

Doctorat délivré conjointement par l'Université d'Abomey-Calavi et l'Université du Littoral Côte d'Opale

(spécialité : Génie Informatique, Automatique et Traitement du Signal)

par

Mikaël Ange Mousse

Composition du jury

Présidents :	**Antoine Vianou** Professeur à l'Université d'Abomey-Calavi, Bénin
Rapporteurs :	**Pierre Gouton** Professeur à l'Université de Bourgogne, France **Marc Kokou Assogba** Maître de conférences HDR à l'Université d'Abomey-Calavi, Bénin
Examinateurs :	**Patrice Wira** Professeur à l'Université de Haute Alsace, France
Directeurs de thèse :	**Cina Motamed** Maître de conférences HDR à l'Université du Littoral Côte d'Opale, France **Eugène C. Ezin** Maître de conférences HDR à l'Université d'Abomey-Calavi, Bénin

Unité de Recherche d'Informatique et Sciences Appliquées
&
Laboratoire d'Informatique, Signal et Image de la Côte d'Opale

Mis en page avec la classe thloria.

Remerciements

Ce travail est le fruit d'une thèse en cotutelle effectuée au sein du laboratoire LISIC, à l'Université du Littoral Côte d'Opale, et l'Unité de Recherche en Informatique et Sciences Appliquées, à l'Institut de Mathématiques et de Sciences Physiques. Il n'aurait jamais pu voir le jour sans le soutien moral et intellectuel de nombreuses personnes auxquelles je voudrais exprimer ma profonde reconnaissance.

Je voudrais dans un premier temps adresser toutes mes gratitudes à mes directeurs de thèse M. Cina MOTAMED et M. Eugène C. EZIN. Vous avez cru en moi dès le début de cette aventure et je vous remercie beaucoup pour votre soutien et votre encadrement. Votre bonne humeur, votre patience et vos encouragements ont été une source intarissable qui me poussait à avancer.

Je remercie également les mebres du jury qui ont accepté apporter leurs expertises pour parfaire et améliorer le présent travail.

Je tiens à remercier le Professeur Joël TOSSA pour son accompagnement et son soutien. A travers lui, j'exprime ma reconnaissance à tous les enseignants de l'IMSP pour leur participation durant tout le cursus de ma formation.

Je m'en voudrais de ne pas remercier Monsieur Abdoulaï YAYA. Vous avez su me donner goût à l'informatique. Merci pour tout ce que vous avez eu à faire pour moi.

Ma gratitude va à l'encontre de tout le personnel du LISIC qui n'a ménagé aucun effort pour m'accueillir lors de mes différents séjours dans leurs locaux.

Enfin, je ne trouve pas de mots assez forts pour exprimer mon sentiment de reconnaissance et de profonde gratitude à ma famille. Je voudrais citer surtout mon père M. Eric MOUSSE afin de lui dire que cette reconnaissance lui sera toujours éternelle et de le remercier aussi de l'effort et des sacrifices financiers qu'il a dû faire pour me permettre de poursuivre mes études en France. Ses encouragements m'ont toujours permis de m'accrocher dans les moments difficiles et de ne pas me décourager. Merci aussi à ma soeur Sandrine.

Un merci spécial à ma Queen, ceux qui la connaissent savent pourquoi.

Un grand merci à mon camarade de lutte M. Fréjus A. A. LALEYE pour les multiples discussions que nous avons eu dans le cadre de la réalisation de ce travail et ceci malgré le climat qui, parfois nous était pas toujours favorable.

A tous ceux qui n'ont cesser de me soutenir (Fawaz, Arnaud, Bethel, Christian, Roméo, Muriel, Calice,...), je voudrais leur rendre un vibrant hommage.

Je voudrais aussi remercier l'ambassade de France au Bénin pour la bourse de mobilité offerte afin de faciliter la réalisation de ce travail. Je n'oublie pas mes amis boursiers du SCAC en particulier Léonide M. SINSIN qui m'a aidé à plusieurs reprrises notamment lors de mes séjours en France.

A Tous je vous dis encore merci.

In loving memory of

ATTERE Colette Patricia Kossilatè.

Dear Mother, may your soul continue to rest in peace.

Table des figures

Liste des tableaux

Glossaire

MoG : Mixture of Gaussian

CB : Codebook

MCBSb : Combinaison de l'algorithme "Codebook" et de l'opérateur de détection de contour "Sobel"

MCBLp : Combinaison de l'algorithme "Codebook" et de l'opérateur de détection de contour "Laplacian of Gaussian"

MCBCa : combinaison de l'algorithme "Codebook" et de l'opérateur de détection de contour "Canny"

FPR : Taux de faux positif

TPR : Taux de vrai positif

PR : Précision

FM : F-Mesure

fps : Image par seconde

HMM : Modèle de Markov Cachée

LHMM : Modèle de Markov Cachée Hiérarchisé

DBN : Réseaux Bayésiens Dynamiques

LBP : Motifs binaires locaux

SVM : Support Vector Machine.

Table des matières

Chapitre 1

Introduction générale

Sommaire

La vidéosurveillance intelligente a connu un attrait important du fait des nombreuses avantages qu'elle offre. Elle est utilisée dans la plupart des cas pour la reconnaissance de comportements. Cette reconnaissance se fait à l'aide des techniques d'analyses et d'interprétation automatique des séquences vidéos par un système informatique. Dans nos travaux nous avons proposé un waffer système multi-cameras de reconnaissance de comportements humaines avec une application à la détection de chute.

Le présent chapitre est subdivisé en trois (3) sections. La première section présente les motivations de ce travail de thèse. Après la présentation de ces motivations, nous faisons le point sur les principales contributions de nos travaux de recherches dans la seconde section. Enfin la troisième section du chapitre présente la structure du manuscrit.

1.1 Motivations

La reconnaissance d'activités humaines dans des séquences vidéos est l'une des thématiques les plus en vogues dans le domaine de la vision par ordinateur. Elle permet le développement des applications dans le domaine de surveillance aussi bien des endroits sensibles (gares, aéroports, ports, supermarchés, sites militaires,...), des environnements industriels que des environnements médicaux [2, 3, 4]. Plus récemment plusieurs applications de surveillance de personne à l'aide de maisons intelligentes ont vu le jour. L'objectif principal de ces systèmes est de permettre le monitoring des personnes. Ils permettent de suivre les comportements de ces personnes en vue d'extraire des activités typiques [5, 6, 7]. Ainsi toutes activités sortant du cadre de celles extraites sont considérées comme suspectes et méritent d'être traitées avec une attention particulière. Une des activités suspectes à détecter est la chute de personnes. On parle de chute de personne lorsqu'un individu réalisant une activité normale (marché, mangé, cuisiner,...) se retrouve accidentellement au sol. Il peut avoir la force de se relever ou non. Dans le cas où il parvient à se relever, il peut lui même appeler les secours pour réparer les éventuels problèmes occasionnés par sa chute. Mais dans le second cas, cette chute peut occasionner de nombreuses conséquences néfastes car l'individu ne peut pas appeler le service d'urgences ou un centre médical pour lui venir en aide. Par exemple selon MacCulloch et al. [1], la chute est la sixième cause de décès chez les personnes âgées [1]. Les conséquences des chutes (hospitalisations, décès), sont schématisées par la pyramide reportée figure 1.1, extraite du rapport de Boulmier [8].

La gravité des conséquences de chute dépend en grande partie du temps mis pour aider la personne qui est tombée accidentellement. Ainsi plus la réponse à l'accident est rapide, moins lourdes seront les conséquences sur la vie du sujet. Pour faire face a cet état des choses, plusieurs laboratoires de recherche ont mis en place des solutions afin de limiter les conséquences de ce problème de santé, qui engendre d'importants problèmes humains

1. personnes dont l'age est supérieur ou égale à 65 ans

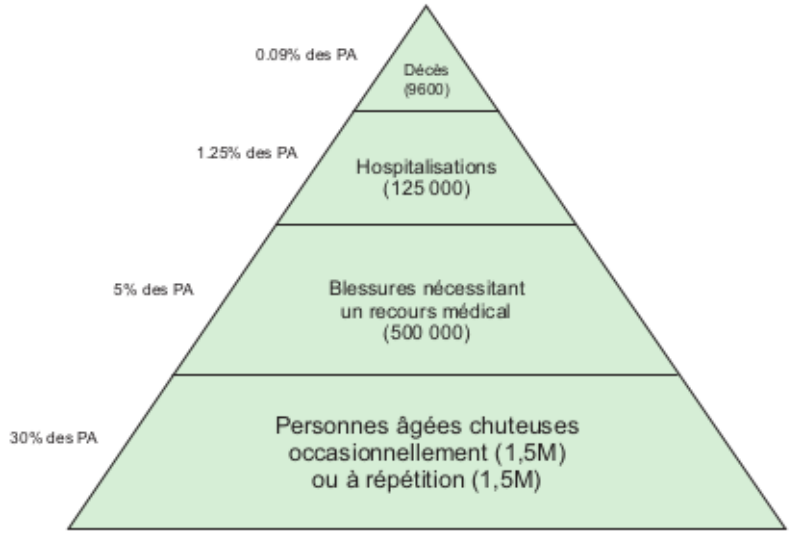

FIGURE 1.1 – Importance des chutes et de leurs conséquences annuelles chez les personnes âgées [8]

et financiers.

Cette thèse aussi s'inscrit dans le contexte. L'objectif global est la proposition d'un système de vidéo surveillance pour la détection automatique de chute de personne. Bien que de dizaines d'équipes de recherche à travers le monde s'intéressent à l'utilisation de la vision artificielle pour la détection automatique des chutes de personnes, dans le but de permettre le déclenchement rapide des secours. Plusieurs approches assurent une détection prometteuse de chutes. Cependant la réduction du temps de calcul demeure toujours un challenge dans ces genres de systèmes toujours dans l'optique de rendre plus instantané l'appel des secours. Ainsi, ces dernières années, Nous avons donc poursuivi nos recherches dans le but d'avoir un système de détection de chutes performant basé sur la vision par ordinateur mais avec une complexité moindre. Cela réduira le temps de calcul et par conséquences le temps d'intervention des secours. Étant basé sur un système de vision artificiel, nous devions dès lors étudier les diverses couches de ce type de système.

1.2 Principales contributions

Les principales fonctionnalités technologiques d'un système de vidéosurveillance intelligent dans un contexte multi-caméras ou non peuvent se résumer notamment au calibrage de la scène, la détection automatique et le suivi d'objets mobiles dans des séquences vidéos, la reconnaissance et la classification d'objets d'intérêt, l'analyse d'activités humaines et l'interprétation vidéo pour la compréhension de scène. Ces traitements dans un système de vidéosurveillance intelligent sont subdivisés de deux : les traitements bas niveaux et les traitements haut niveaux. Les traitements bas niveau regroupent : le calibrage de la scène, la détection automatique, le suivi d'objets mobiles, la reconnaissance et la classification d'objets d'intérêt. Les traitements haut niveau quant à eux regroupent l'analyse d'activités humaines et l'interprétation vidéo pour la compréhension de scène. Dans cette thèse, nous avons abordé ces deux couches d'un système d'interprétation automatique de scénarios avec une application à la détection de chute de personnes. A chacun des deux niveaux, nous avons suggéré des algorithmes en vue d'obtenir un système globale robuste et respectant les contraintes liées à la détection de chutes.

Les traitements bas niveaux ont généralement pour but le filtrage et l'extraction de caractéristiques. Ainsi au niveau des traitement bas niveaux, nous avons suggérés des méthodes pour une extraction plus efficiente des objets en mouvement. Ces deux méthodes de détection sont basées sur l'approche "Codebook" qui est une approche répandue pour l'extraction des pixels de premier plan. Le premier algorithme proposé combine l'algorithme basé sur l'approche "Codebook" avec un algorithme de détection de contours. L'algorithme de détection de contour est utilisé en vue de valider si les pixels de premier plan détecté par l'algorithme utilisant l'approche "Codebook" le sont vraiment. Nous avons fait nos expérimentations en utilisant trois algorithmes de détection de contour différents. Mais dans le but de réduire le temps de calcul tout en gardant des performances correctes, nous avons proposés une seconde approche basée sur la région. Les régions uniformes de la scène

sont regroupées en superpixels et la méthode d'extraction est basée sur ces superpixels. Vu que le système à mettre en place est un système multi-cameras, nous nous sommes intéressés à la gestion de ces genres de systèmes. Ainsi nous avons proposé une stratégie de fusion pour agréger les informations de premier plan de chaque camera. Le but de cette stratégie est de de détecter de manière adéquate la surface au sol d'un objet et d'identifier la caméra qui donne la meilleur vue de l'objet. Le dernier travail que nous avons réalisé dans cette partie bas niveau a été choisir de choisir un algorithme de suivi d'objet en mouvement dans un scène et de l'adapter en vue d'une implémentation respectant les contraintes d'un système de détection de chutes de personnes.

Les traitements haut niveau permettent l'analyse des caractéristiques extraites. Cette analyse conduit la plupart du temps à la reconnaissance d'événements dans la scène. Dans notre travail le scénario à détecter est : la chute d'une personne. Nous avons proposés un algorithme qui utilise la surface au sol obtenue à l'aide du traitement bas niveau. A l'aide de cette surface au sol, nous définissons des seuils qui ont pour rôle de donner une information sur l'état de la personne. Les valeurs de ces seuils sont obtenues après une phase d'apprentissage supervisé sur l'individu dans la scène. L'utilisation des seuils que nous proposons permettent de déterminer une éventuelle chute chez la personne mais donne aussi une idée générale de sa posture.

1.3 Organisation du manuscrit

La présentation de nos travaux dans ce mémoire est structurée en six chapitres regroupés en deux parties. Après le chapitre introductif nous avons la première partie du manuscrit dénommée détection de mouvements. Composée de deux chapitres, cette partie est essentiellement consacrée au traitement bas niveau. En effet, le premier chapitre de cette partie présente les différentes approches de détection d'objet en mouvement dans

un système de vidéosurveillance intelligent (mono-caméra et multi-caméras) tandis que le second expose les différentes approches de détection de mouvements (à l'aide d'un système mono camera) proposées au cours de nos recherches. Nous exposons aussi les critères utilisés pour évaluer les performances des différents algorithmes. Enfin les résultats de nos expérimentations sont présentés et nous interprétons les différentes valeurs obtenues.

La deuxième partie est consacrée à l'application de la thèse. Elle est constituée de trois chapitres. Le premier chapitre présente les approches de détection de chute en mettant l'accent sur les méthodes utilisant un système de vidéosurveillance intelligent. Ce chapitre nous permet de justifier les diverses "orientations" données dans la proposition de l'application résultant de nos travaux. Enfin le deuxième chapitre présente notre algorithme de détection de chute de personnes par un système multi-cameras de vidéosurveillance surveillance intelligent. Dans ce dernier nous présentons les séquences utilisées pour le test de la stratégie de détection de chutes tout en précisant leurs caractéristiques. Nous exposons aussi les critères utilisés pour évaluer les performances des différents algorithmes. Enfin les résultats de nos expérimentations sont présentés et nous interprétons les différentes valeurs obtenues. Pour terminer le manuscrit, nous avons le dernier chapitre qui est consacré à la conclusion et à la présentation des perspectives. Il résume les travaux effectués dans le cadre de cette thèse tout en précisant les pistes futures que nous explorerons.

Première partie

Détection de mouvements

Chapitre 2

Techniques de détection de mouvements dans les séquences d'images

Sommaire

2.1 Introduction

La détection de mouvements représente une tâche importante pour tout système de vidéosurveillance intelligente. De son résultat dépend tous les autres traitements. Le but du module de détection de mouvements est d'extraire à partir d'une séquence les pixels qui contiennent des objets en mouvement de la scène observée par un ou plusieurs cameras. Le résultat de la détection dépend de plusieurs variables. On peut citer par exemple : la variation de la luminosité, la présence d'ombre d'objets, etc... Dans la littérature d'innombrables stratégies de détection de mouvements plus ou moins robustes ont été proposé. Dans ce chapitre nous recensons les diverses méthodes de détection de mouvements existants dans l'état de l'art.

Ce chapitre est subdivisé comme suit. Les deux premières sections font le recensement des méthodes de détection de mouvements disponibles dans l'état de l'art. La première méthode présente les méthodes utilisées pour extraire le premier plan dans une séquence mono camera tandis que la seconde présente les méthodes pour extraire le premier plan dans une séquence multi-cameras. La troisième section présente une évaluation comparative des approches et permet de justifier nos choix en matière de proposition d'algorithme de détection de mouvement. Enfin nous terminons le chapitre par une synthèse.

2.2 Méthodes de détection d'objets en mouvement dans les séquences mono-caméra

Les systèmes de vidéosurveillances mono-camera ont pris un ampleur à cause de deux raisons majeurs. La première cause est que les cameras sont devenus beaucoup moins chers. En plus de cela ils faut ajouter qu'il est beaucoup plus facile de mettre en place de tels systèmes. La couche intelligente permet au système d'être beaucoup plus autonome

et comme nous avons dis précédemment la première étape de tout système de vidéosurveillance intelligente est la détection d'objets en mouvements. Depuis la naissance de cet axe de recherche qui s'articule autour de la détection automatique d'objets en mouvement, plusieurs algorithmes ont été proposés et plusieurs travaux ont été publiés. Le but de ces travaux est de proposer des méthodes robustes aux conditions complexes de capture comme : les objets non-rigide, arrière-plan dynamique, changement d'éclairage, etc.. Vis-à-vis la multitude des méthodes proposées dans la littérature pour la détection d'objets en mouvement, la classification de ces méthodes n'est pas une tâche aisée. La plus part de ces travaux sont orientés vers des applications précises qui traitent les systèmes complets de surveillance. Ces algorithmes peuvent être regrouper en trois grandes catégories :

— algorithmes de détection de mouvements sans modélisation de l'arrière plan ;
— algorithmes de détection de mouvements avec modélisation de l'arrière plan ;
— combinaison d'un algorithme de modélisation de fond et d'un autre algorithme.

2.2.1 Algorithmes de détection de mouvements sans modélisation de l'arrière plan

Les algorithmes de cette catégorie sont les plus utilisés à cause de la facilité avec laquelle ils peuvent être implémenter. Les techniques les plus utilisées sont le seuillage d'image, la différence d'images et le gradient spatio-temporel. Ces outils exploitent dans leur plus grande partie les informations de l'image courante et la précédente (ou un certain nombre d'images précédentes) des séquences vidéos.

Détection de mouvements basée sur le seuillage d'image

Le seuillage d'image est la technique de détection d'objet la plus simple. Les algorithmes de cette catégorie se basent sur le principe selon lequel la couleur des objets en mouvement est différent de celle du fond de l'image. Ainsi chaque pixel de l'image en fonction du seuil défini est soit pixel d'un objet de premier plan ou du fond [18]. Le seuil

utilisé peut être global (pour toute l'image) [18] ou local (une partie de l'image) [19]. Dans le second cas le choix du seuil est dynamique et s'adapte aux différentes zones identifiées dans l'image. Le choix du seuil se fait en basant sur des techniques comme l'histogramme des couleurs [20]. Les auteur dans [20] ont mené des discussions sur le choix des types de seuillage.

Dans la majeure partie des cas, les objets et le fond de l'image partage beaucoup de couleurs. Par conséquent il est difficile de déterminer un seuil optimal en vue de séparer les objets du fond de l'image. Pour cela certains travaux de recherches ont porté sur des stratégies pour l'obtention des seuils. Su et Amer [21] ont recensé deux types de seuillage et ont proposé un algorithme non paramétrique pour le calcul du seuil. Leurs résultats ont montré que l'approches suggérée était beaucoup moins rapide que les approches traditionnelles (Poisson, Euler) mais permet d'obtenir un meilleur résultat en matière de détection d'objets en mouvement. Le seuillage d'image est une technique très utilisée surtout lorsque les objets ont des couleurs différentes de celles de l'arrière plan.

Détection de mouvements basée sur la différence d'images

La différence d'images est l'une des techniques les plus utilisés pour extraire les pixels de premiers plan d'un fond statique. Dans la plupart des méthodes utilisant cette technique, se base sur les différences entre deux images consécutives de la séquence vidéo. Dans ces cas seuls les pixels qui ont beaucoup variés (en fonction d'une valeur prédéfinie) sur la scène sont considérés comme pixels de premier plan. Cette technique marche très sur des scènes avec arrière plan statique mais cela impose à l'objet d'être en perpétuel mouvement [22]. D'autre améliorations à la différence basique ont été proposées. Une des améliorations a été de considérer l'image courante et les neuf (09) images précédentes de la séquence [23]. Ils utilisent l'analyse en composante principale (PCA) pour réduire les données à manipuler et appliquent la différence d'images sur les données issues des dix

(10) images. Cette méthode est beaucoup plus robuste que la différence d'image basique. Tout comme [23], d'autres travaux proposent aussi d'actualiser l'image de référence après un certain temps. Les auteurs de [24], proposent de détecter les objets en sauvegardant une historique des valeurs des pixels de la scène. Cette historique permet de déterminer les positions actuelles de tous les objets de la scène en générant une représentation 2D des images de la séquence.

Pour finir l'état de l'art des algorithmes de cette catégorie, nous soulignons qu'une comparaison de certains algorithmes de détection de mouvements basés sur la différence d'images a été réalisée par Rosin et Ioannidis dans [25]. La figure 2.1 présente les résultats de détection des algorithmes étudiés par [25]. D'après leur étude comparative, la technique de détection proposée par [26] donnait de meilleurs résultats que les autres approches étudiées.

Détection de mouvements basée gradient spatio-temporel

Le gradient spatio-temporel est utilisé pour détecter les objets en mouvement à partir des scènes avec arrière plan statique. Les algorithmes cette catégorie implémentent en majorité les techniques permettant l'obtention de flot optique. Des revues de littérature regroupant les diverses techniques de cette catégorie ont été par [27, 28]. La figure 2.2 montre les résultats de l'estimation du flot optique. Dans [28], les auteurs ont présentés ces algorithmes tout en comparant leur complexité et leur précision. La figure 2.3 présente les résultats qu'ils ont obtenu. Ces résultats montrent que les algorithmes donnent des résultats acceptables mais leurs complexités étaient trop importantes du fait du calcul du flot optique. Les résultats de la détection dépendent aussi de l'objet. En effet, si l'objet est homogène (voitures, avions,...) la détection est beaucoup plus meilleurs. Mais dans le cas contraire (homme en marche) la détection devient compliquée [29]. Aussi les objets non homogènes peuvent causer des bruits lors du calcul du flot optique par région. Ces

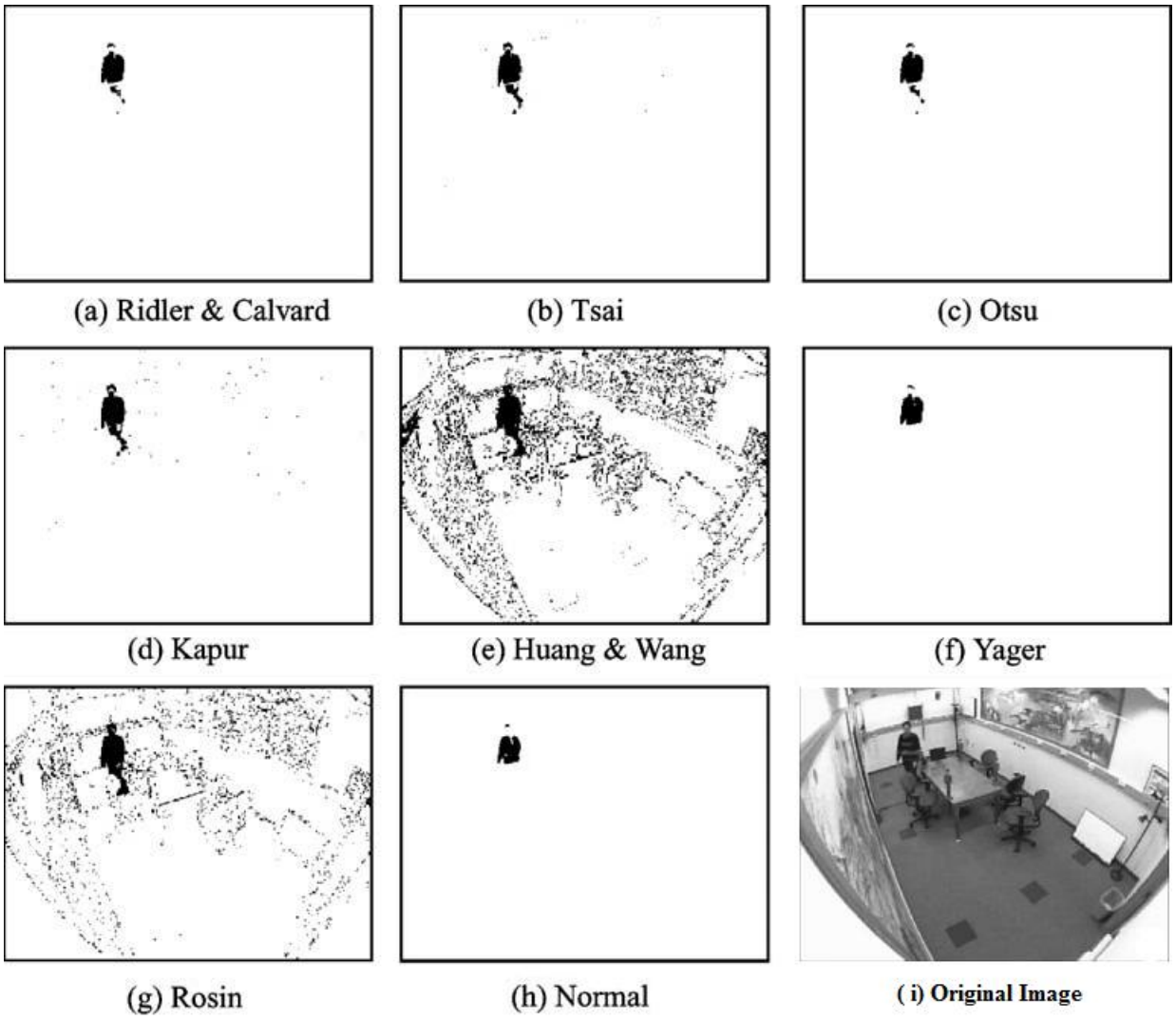

FIGURE 2.1 – Résultats de la détection de divers algorithmes basés sur la différence d'images. [25]

conditions font que l'utilisation du flot optique n'est pas adapté au objet non homogène [30].

La seconde classe de techniques est basée sur l'entropie de l'image. Dans ce cas, l'entropie est obtenu en comparant pendant une période bien définie la variation de l'intensité d'un pixel à la variation des pixels qui lui sont proches. Dans la littérature, l'entropie de l'image a été exploité de divers manière pour l'extraction du premier plan. Kapur et al. [31] considère l'extraction de pixels de premier plan comme étant un processus de clas-

FIGURE 2.2 – Exemple d'estimation du flot optique.

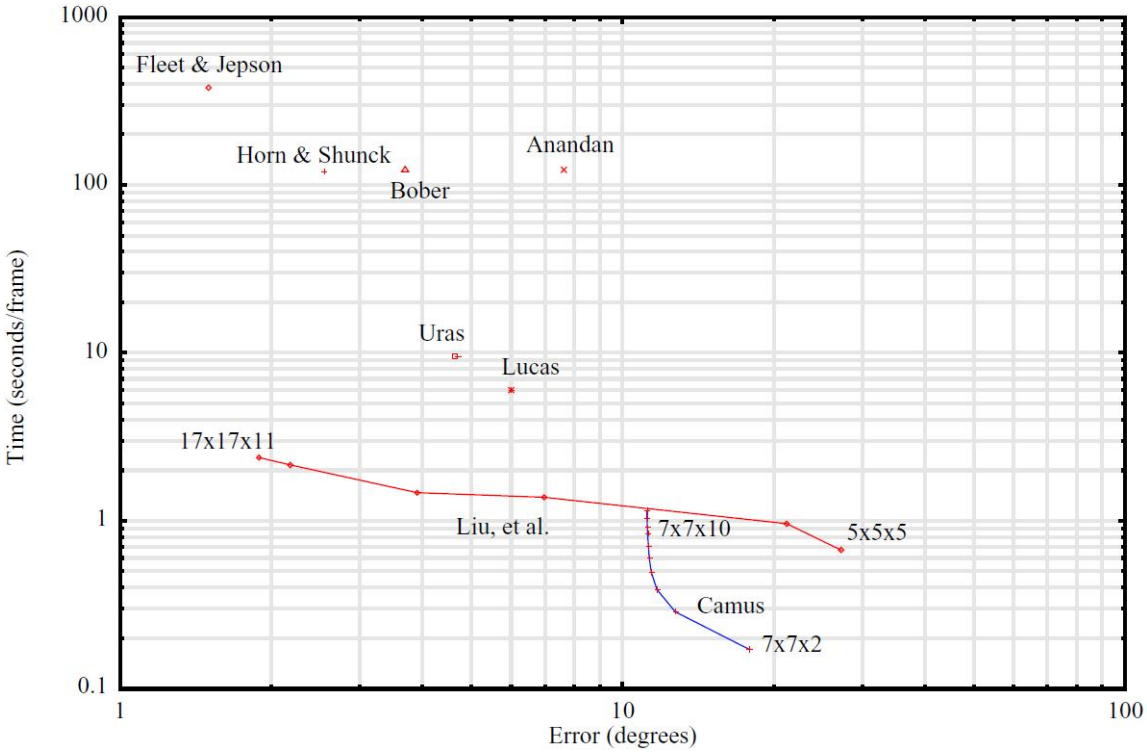

FIGURE 2.3 – Performances de divers algorithmes basées sur le flot optique. [28]

sification des pixels en deux classes. Ils utilisent donc l'entropie de l'image et proposent une fonction de densité de probabilité pour caractériser chacune des deux classes. Les auteurs de [32], proposent d'utiliser l'histogramme de l'entropie de l'intensité en utilisant des règles obtenus à l'aide de la logique floue. Ces règles sont issues des travaux proposés par

Yager [34] et Huang et Wang [33]. Ma et Zhang [35] suggèrent d'utiliser les histogrammes "temps-espace" pour le calcul de l'entropie. Les zones de premier plan sont les parties de l'image où l'entropie ainsi calculée atteint une valeur maximale.

2.2.2 Algorithmes de détection de mouvements avec modélisation de l'arrière plan

La modélisation de l'arrière plan pour la détection de mouvements permet de segmenter l'image en deux classes (premier plan et arrière plan) dans les scènes complexes (arrière plan non statique, variation de luminosité, ...). Ces techniques construisent un modèle pour l'arrière plan en se basant sur l'historique des valeurs prises par les pixels. La construction de ces modèles d'arrière plan se fait la plupart du temps en utilisant des modèles statistiques ou probabilistes. Tous les pixels n'ayant pas de concordance avec un pixel de l'arrière plan sont considérés comme pixels du premier plan. La validité de ces techniques reste conditionnée par l'arrière-plan utilisé, le développement des arrière-plans valides est une tâche très complexe :

— premièrement, l'arrière-plan doit être robuste au changement de la luminosité et aux mouvements lents ;

— deuxièmement, il faut éviter la détection des objets mobiles dans le fond et les ombres projetées par les objets mobiles.

Un bon modèle de fond devrait rapidement répondre aux changements en arrière-plan et s'adapter automatiquement aux changements survenant dans le fond comme :

— la stationnarité dans la mobilité des objets ;

— changement de luminosité,...

Les algorithmes de détection de mouvements avec modélisation de l'arrière peuvent être subdiviser en trois grandes classes : modélisation de l'arrière plan, estimation de l'arrière plan et soustraction de l'arrière plan. En général, les techniques de modélisation de fond améliore les performances de la détection de l'avant-plan de manière significative dans

presque tous les environnements (intérieur/extérieur).

Modélisation de l'arrière plan

La modélisation de l'arrière plan se fait en utilisant chaque pixel de l'image. Une fois le modèle d'arrière plan obtenu lors de la phase d'apprentissage, les pixels de premier plan sont obtenus en comparant la valeur de chaque pixel avec le le modèle. Si la valeur correspond a une valeur dans le modèle d'arrière plan alors ce pixel est un pixel d'arrière plan. Dans le cas contraire, il est un pixel de premier et par conséquent appartenant à un objet en mouvement. Le modèle de l'arrière plan est obtenu en utilisant la densité de probabilité des caractéristiques de la scène. Dans cette catégorie, l'un des algorithmes le plus utilisé est celui proposé par Stauffer et al. [36]. Stauffer et al. ont proposé d'utiliser un mélange gaussien (Mixture of Gaussians (MoG)) pour la modélisation de l'arrière plan. Cette modélisation permet la représentation de l'image en arrière-plan avec une distribution multimodale. Dans un mélange gaussien, la distribution temporelle au fil du temps de l'intensité de chaque pixel est modélisée paramétriquement par un mélange de K gaussiennes. Le mélange gaussien est caractérisé par une moyenne, une matrice de covariance et une probabilité a priori de chaque k gaussiennes. Ces paramètres sont mis à jour dans chaque image de la séquence vidéo. Pour chaque pixel p_t , chaque gaussienne des K distributions correspond à la probabilité d'observation d'une intensité particulière, ce qui rend le mélange gaussien (MoG) plus générale par rapport à un simple gaussien. Cette approche donne de bon résultat mais n'est pas adaptée aux scènes présentant des ombres et/ou des variations de luminosité. De nombreux travaux de recherches [37, 39, 38] ont eu pour objectif l'amélioration des performances du MoG (notamment par rapport aux deux cas précédemment énoncés qui posent des problèmes). Au lieu d'utiliser une distribution semi paramétrique, Elgammal et al. suggèrent d'utiliser une distribution non paramétrique [40]. Ils ont démontré dans leur travail que la distribution non paramétrique était beaucoup plus flexible que la semi paramétrique. Leur modèle devient beaucoup plus

robuste surtout lorsque l'arrière plan contient de petits objets. L'inconvénient majeur de cette solution est le temps de calcul. En effet l'obtention du modèle d'arrière plan avec la distribution non paramétrique est beaucoup plus complexe. La dernière approche de cette catégorie que nous présentons dans notre état de l'art est celle proposée par Kim et al. [9]. Les expérimentations ont démontré que cette approche donne des résultats acceptables en matière d'extraction de premier plan (confère figure 2.4). En observant la

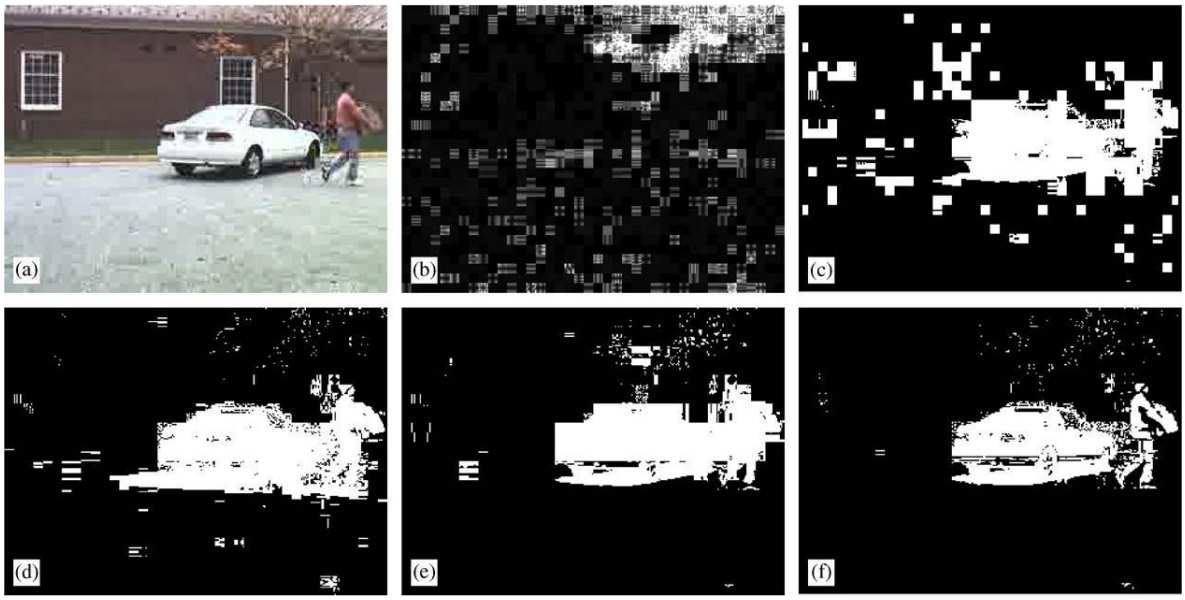

FIGURE 2.4 – Résultats de la détection de divers algorithmes basées sur la modélisation de l'arrière plan. [9] (a) image original, (b) déviations standard, (c) Horprasert et al. [41], (d) Stauffer et al. [36], (e) Elgammal et al. [40], (f) Kim et al. [9]

figure 2.4, nous remarquons que l'approche "Codebook" proposée par Kim et al. donne une meilleur segmentation que les autre approches classiques ([36, 40]) de l'état de l'art. Comme toutes les approches de cette catégorie, en utilisant l'algorithme "Codebook", les objets stationnaires sont absorbés dans l'arrière-plan s'ils restent quelque secondes et la technique de modélisation de fond adopte la couleur de l'objet stationnaire arrière-plan sol. Notons que les variations soudaines de l'intensité lumineuse font que le modèle de fond est instable.

Estimation de l'arrière plan

Ces techniques utilisent des méthodes de prédiction dans le but d'estimer l'arrière plan. Cette estimation est faite en se basant sur les valeurs prises par les pixels dans les précédentes images de la séquence. L'une des méthodes de cette catégorie est celle dénommée "VuMeter" et proposée par Goyat et al. [42]. C'est une méthode probabiliste qui permet d'obtenir un modèle en utilisant l'estimation de la fonction de la densité de probabilité. Le modèle obtenu est un modèle non paramétrique. Les filtres adaptatifs sont aussi utilisés pour l'estimation de l'arrière plan. Ainsi Ridler et al. [43] et Zhong et Sclaroff [44] ont proposé des méthodes qui utilisent le filtre de Kalman pour la modélisation de chaque pixel.

D'autres techniques statistiques ont été utilisées. On peut par exemple donner l'exemple de Soatto et al. [45] et Doretto et al. [46] qui utilise une fonction d'auto-régressive pour estimer la moyenne des mouvements sur la scène. Cela permet entre autre d'extraire des objets en mouvement des arrières plans dynamiques.

Soustraction de l'arrière plan

Cette dernière catégorie d'algorithmes avec modélisation de l'arrière plan permettent de générer une image d'arrière plan en se basant sur les images précédentes de la séquence vidéo. Une fois l'image d'arrière plan obtenue, elle est soustraite de l'image courante. Et le résultat de cette soustraction représente les endroits de l'image sur lequel on a observé du mouvement. Plusieurs approches de l'état de l'art suivent ce principe. On peut citer par exemple les travaux effectués par Horprasert et al. [41]. Horprasert et al. propose de séparer dans le modèle de couleur la luminance de la chromaticité. Le but de cette séparation est de pouvoir obtenir une image d'arrière plan robuste et faisant face aux problèmes causés par les ombres. Nous pouvons aussi citer les travaux de Lo et Velastin [47] et Cucchiara et al. [48]. Ils utilisent les valeurs moyennes et médianes de chaque pixel depuis le début

de la séquence. Ces valeurs sont exploitées pour l'obtention de l'image de d'arrière plan. Ces algorithmes ont des résultats acceptables mais présentent un inconvénient majeur. En effet pour avoir l'image de l'arrière plan, il faut sauvegarder toutes les images à partir du début de la séquence. Ce qui requiert beaucoup d'espaces mémoires.

2.2.3 Algorithmes de détection de mouvements combinant un algorithme de modélisation de fond et un autre algorithme

Les algorithmes de détection de mouvement de cette catégorie combine un algorithme de modélisation de fond ("Codebook", MoG, ...) avec d'autres algorithmes. Le but est de minimiser les erreurs de détection. Cong et al. [49] proposent d'utiliser l'approche de modélisation d'arrière plan MoG avec la méthode de différence d'images. Izadi et Saeedi [50] proposent d'utiliser la même approche basée sur le MoG tout en la combinant le gradient spatial. Ils utilisent aussi des opérateurs morphologiques pour supprimer les ombres éventuelles sur les images. Ces approches donnent de bons résultats comme en témoigne la figure 2.5. Li et al. [53] proposent un framework Bayésien qui modélise les caractéristiques spectrales, spatiales, et temporelles des pixels. Ces caractéristiques sont utilisés pour réaliser le modèle d'arrière plan. Heikkilä et Pietikäinen [51] modélisent les pixels par un groupe d'histogrammes des caractéristiques locales (LBP). Ces caractéristiques sont utilisées pour la modélisation des dépendances spatiales entre les pixels. Tian et Men [52] ont étendu l'approche proposée par Heikkilä et Pietikäinen en utilisant une variante plus avancée du LBP. Tous les algorithmes présentés dans cette catégorie ont de très bon résultats en matière de détection d'objets mobiles. Mais du fait de la combinaison de plusieurs méthodes, leur complexité devient trop élevée. Ainsi il est difficile de les implémenter dans une application temps réel.

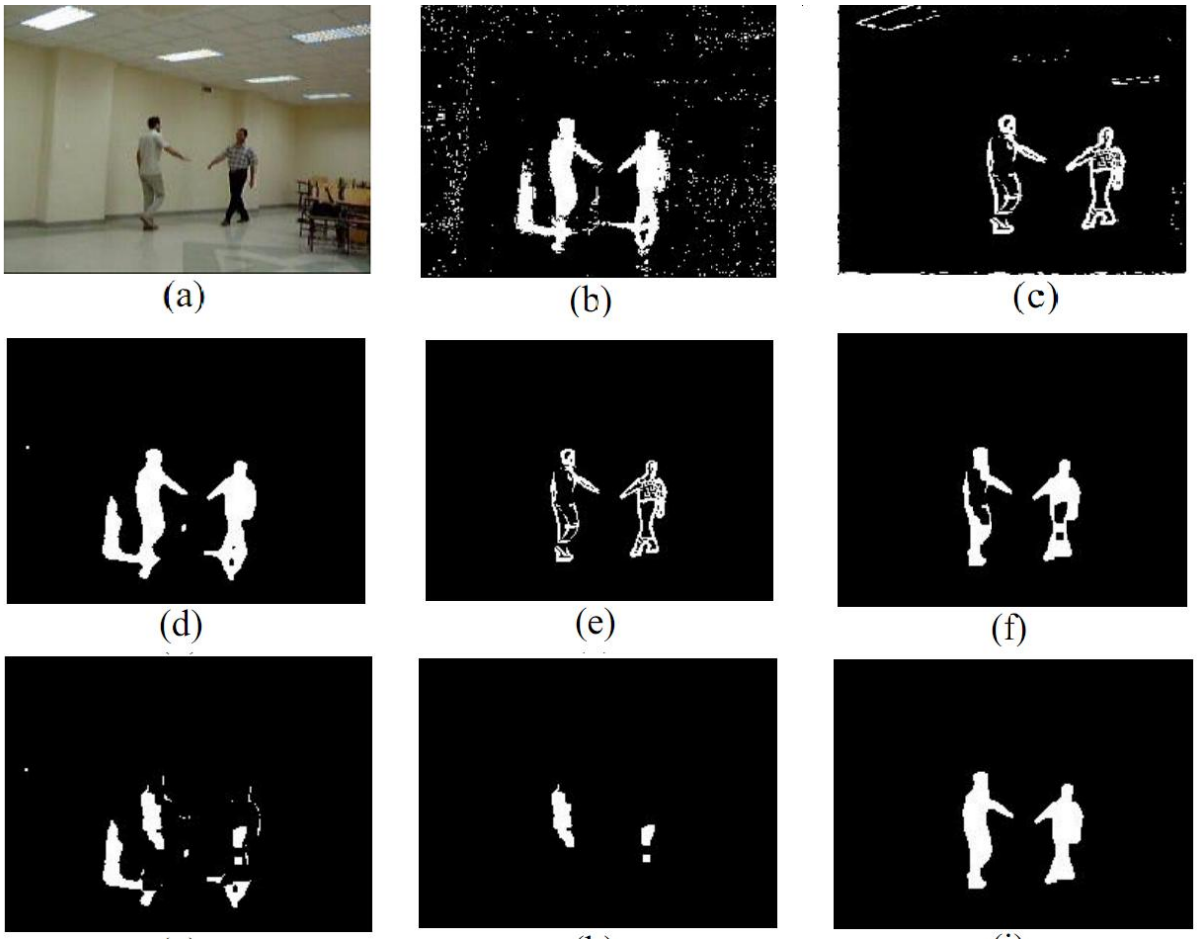

FIGURE 2.5 – Illustration étape par étape de l'approche proposée par Izadi et Saedi. [50]

2.3 Méthodes de détection d'objets en mouvement dans les séquences multi-caméras

Dans la pratique pour la surveillance des espaces étendus (aéroports, ports, ...), il faut plus d'une caméra. Une des stratégies adoptées est de réaliser un système multi-caméras pour la surveillance. Ces cameras peuvent avoir des vues chevauchantes où non. Dans cette section nous reportons les algorithmes de détection de mouvement à partir de plusieurs cameras avec des vues chevauchants. Car en utilisant des cameras avec des vues chevauchants on arrive à faire face aux problèmes d'occultations et de fausse détections. Selon Xu et al. [61], les stratégies de détections de mouvements depuis les systèmes multi-caméras peuvent être classer en trois catégories.

— La première catégorie regroupe les stratégies qui font la fusion d'information bas niveau. Ainsi les mouvements sont détectés dans chaque caméra. La stratégie permet de permet d'un caméra à une autre lorsqu'elle prédit que la caméra courante n'aura pas une bonne vue de l'objet [54, 55]. Ces méthodes sont fortement dépendantes des caméras car la détection est faite avec une seule des cameras du système.

— Dans la seconde catégorie, le système extrait les caractéristiques des détections individuelles. Ces caractéristiques sont fusionnées en vue d'obtenir une information beaucoup plus globale [56, 58, 57]. Ces méthodes aussi sont fortement dépendant de la détection de mouvements au niveau de chaque camera.

— La troisième catégorie fait une fusion haut niveau d'information. En effet dans ces systèmes les cameras individuelles ne procèdent pas à l'extraction de caractéristiques. Mais elles mettent à disposition d'un centre de fusion leur information de premier plan. La détection finale est obtenue après la fusion [63, 62, 60, 64, 61]. La fusion se fait après projection dans le plan de masse des diverses détections monocaméra. La projection dans le plan de masse quant à elle, se fait la plupart du temps en utilisant l'homographie. Ces algorithmes donnent de très bon résultats et sont adaptés pour faire face aux problèmes d'occultations et de fausses détections (confère figure 2.6).

2.4 Discussion

Dans ce chapitre, nous présentons l'état de l'art pour la détection d'objets en mouvement à partir d'un camera fixe où à partir de plusieurs cameras fixes avec des vues chevauchantes. La détection de mouvement est le premier module dans la réalisation d'un

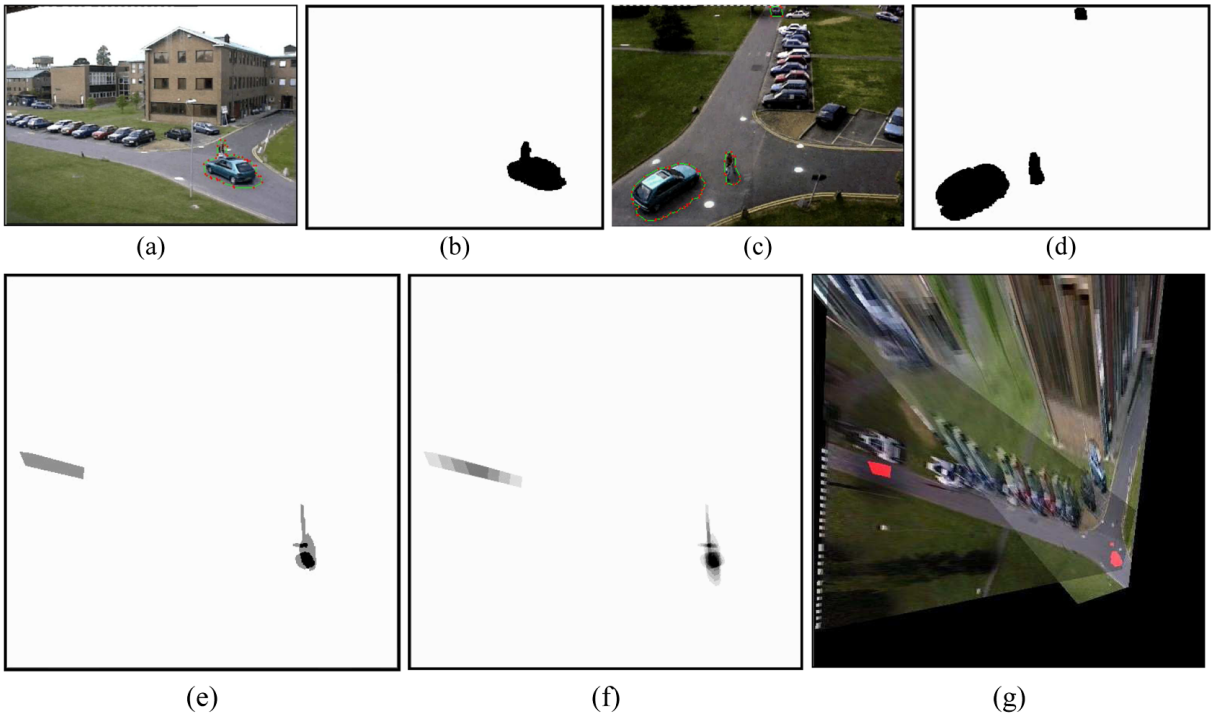

FIGURE 2.6 – Illustration étape par étape de l'approche proposée par Xu et al. [61]

système de vidéosurveillance intelligent.

Les approches de détection de mouvements mono-camera sont subdivisés en trois groupes. Le tableau 2.1 fait un résumé des avantages et inconvénients de chaque groupe. A partir de ce tableau, nous concluons que la détection de mouvements avec modélisation de l'arrière plan possède un certains avantages sur les autres. En se basant sur nos expérimentations, nous avons constaté que l'approche MoG et celle de Codebook présentaient de très bon résultats (détection et temps d'exécution) comparativement aux autres approches de modélisation d'arrière plan. Dans les scènes plus complexes, le modèle de fond obtenu en utilisant "Codebook" occupe moins d'espace mémoire. Ainsi nous avons décider de creuser encore plus la détection d'objet basée sur le "Codebook". Cet apport a été présenté dans le chapitre 3.

Aussi en regardant de plus près les approches de fusion d'informations pour la détection

Techniques	Avantages	Inconvénients
Détection de mouvements sans modélisation de l'arrière plan	- Faible complexité - Flexibilité d'usage - Souplesse d'initialisation	- Non adaptée pour scène avec arrières plans dynamiques - Détection incomplète - Mouvement obligatoire
Détection de mouvements avec modélisation de l'arrière plan	- Adaptée pour tout type de scène - Bonne classification - Résultats clairs	- Initialisation
Détection de mouvements combinant la modélisation de l'arrière plan et un autre algorithme	- Adaptée pour tout type de scène - Bonne classification - Résultats clairs	- Initialisation - Forte complexité

TABLE 2.1 – Tableau comparatif des techniques de détection de mouvements

de mouvements multi-caméras nous pouvons conclure que les approches de la troisième catégorie (fusion haut niveau d'information) sont les plus adaptées. Car cela tire le plus d'avantages possibles de l'utilisation des systèmes multi-cameras. De ce fait les systèmes implémentant ces approches sont plus robustes et font mieux face aux problèmes d'occultations et de fausses détections. En implémentant les approches de cette catégorie nous remarquons que celle qui donne de meilleurs résultat plus vite est celle de Xu et al. [61]. Ce constat confirme la conclusion que Xu et al. avaient émis dans leurs travaux.

2.5 Conclusion

L'importance de la détection de mouvements n'est plus à démontrer. Les divers challenges auxquels sont confrontés les chercheurs font naître de jour en jour des solutions en vue d'optimiser un aspect précis. Ce chapitre, recense les grandes approches dévelop-

pées dans le cadre d'extraction de mouvement dans une scène aussi bien en utilisant un système mono-caméra qu'un système multi-caméras. Une comparaison de ces approches a été proposé en vue de choisir les éléments de l'état de l'art sur lesquels porteront nos études. Le prochain chapitre présente l'un des aspects approfondi durant nos travaux de recherche.

Chapitre 3

Méthodes de détection d'objets mobiles basées sur l'approche "Codebook"

Sommaire

3.1 Introduction

La détection d'objets mobiles est la première étape d'un système de vidéosurveillance intelligente. Dans le chapitre précédent, l'évaluation des techniques de détection de mouvement dans une scène a permis de constater que les algorithmes de détection de mouvements modélisant l'arrière plan peuvent être utiliser dans les scènes simples ou complexes. L'un des algorithmes le plus connu et le plus utilisé dans cette catégorie est celui proposé par Kim et al. [9]. Dans leurs travaux, Kim et al. proposent un algorithme Codebook qui n'utilise pas de paramètre d'apprentissage et qui offre généralement de bons résultats [9].De ce fait, nous nous sommes basés sur cet algorithme dans la réalisation de nos travaux. Au cours de nos travaux, nous avons apporté des modifications à l'algorithme proposé par Kim et al. dans le but d'améliorer aussi bien au niveau du taux de détection qu'au niveau de la complexité de la stratégie de détection des objets mobiles.

Le présent chapitre est subdivisé en quatre (04) sections. La première section présente l'approche "Codebook" tandis que la seconde et la troisième montrent respectivement les deux (02) stratégies de détection de mouvement développées au cours de nos travaux. La première stratégie est une combinaison de la méthode "Codebook" avec un détecteur de contour et la seconde est une extension de la méthode "Codebook" qui exploite les superpixels. Enfin la dernière section fait une conclusion du chapitre.

3.2 Approche "Codebook"

L'approche "Codebook" pour l'extraction des pixels de premier plan a été proposée par Kim et al. [9]. De part ses performances, elle est devenue une référence dans les domaines de la détection et du suivi d'objets mobiles. Elle est robuste et efficace aussi

bien pour les arrières-plans statiques que pour les arrière-plans dynamiques (feuillages, les fontaines, les bords de mers, les drapeaux...) et les légers changements d'illumination. Comme les méthodes de soustraction d'arrière-plan, la méthode "Codebook" se fait en deux (2) phases : une phase d'apprentissage et une phase de détection.

3.2.1 Phase d'apprentissage

Durant la période d'apprentissage, le principe de la méthode "Codebook" est de diviser l'image afin de construire un modèle d'arrière-plan. Ce modèle est représenté par une liste de "codebooks". Chaque "codebook" correspond à un pixel et contient N "codewords". Le "codeword" est créé ou est mis à jour (si le pixel observé est similaire à un "codeword" existant) à chaque itération de l'apprentissage. Le "codeword" est défini par deux vecteurs. Le premier vecteur contient respectivement les valeurs R, G, B du "codeword" (pixel). Le second vecteur contient des données telles que les valeurs de luminosité minimum et maximum, des informations temporelles et de fréquence d'observation du "codeword". Pendant cette phase, tout nouveau "codeword" obtenu (pour un pixel donné) est intégré dans le modèle d'arrière-plan s'il satisfait deux conditions. La première condition est une contrainte sur la distorsion de la luminosité tandis que la seconde est une contrainte sur la distorsion de couleur. Après l'apprentissage, la taille du modèle d'arrière-plan ainsi obtenue est assez importante. Mais Kim et al. ont démontré qu'une taille de 6.5 "codewords" par "codebook" est suffisante pour avoir une bonne qualité du modèle [9]. Ainsi, la derniere étape est d'épurer les "codewords" dont on détecte qu'ils pourraient appartenir à des objets mobiles observés pendant l'apprentissage. Pour ce faire, l'algorithme utilise la valeur "Maximum Negative Run-Length" (λ). Cette valeur est définie comme étant le plus long intervalle dans la période d'apprentissage pendant lequel le "codeword" n'a pas été observé. Ainsi, si cette valeur est importante, cela signifie que ce "codeword" est moins fréquemment observé et donc qu'il n'appartient à priori pas à l'arrière-plan. Notons aussi qu'un objet resté immobile très longtemps peut entraîner une valeur élevé de λ. Le pseudo-

code du processus est présenté par l'algorithme 1. Dans ce pseudo-code $I = \sqrt{R + G + B}$ et la séquence contient N images de taille $(m \times n)$ chacune. Les discussions autour de ϵ_1 et de λ_i sont menées dans [9].

Algorithme 1 : Modélisation du fond

```
1  l ← 0
2  for t = 1 to N do
3      for each frame F_t do
4          for each pixel p_t(R,G,B) of frame F_t do
5              Find the matched codeword c_i in codebook matching to p_t based on two
                conditions (a) and (b).
                   (a) colordist (p_t, v_i) <= ε_1
                   (b) (brightness (I, Ǐ_i, Î_i)) = true
6              if l = 0 or there is no match then
7                  l ← l + 1
8                  create codeword c_L by setting parameter
                   v_L ← (R,G,B) and aux_L ←{I, I, 1, t − 1, t, t}
9              else
10                 update codeword c_i by setting v_i ← (f_iR_i+R/f_i+1, f_iG_i+G/f_i+1, f_iB_i+B/f_i+1) and
                    aux_i ←{min(I, Ǐ_i), max(I, Î_i), f_i + 1, max(λ_i, t − q_i), p_i, t}
11 for each codeword c_i do
12     λ_i ← max{λ_i, ((m × n × t) − q_i + p_i − 1)}
```

3.2.2 Phase de détection

Après la phase d'apprentissage, le modèle obtenu représente l'arrière-plan de l'image. Il correspond à la partie "sans mouvement" de l'image. Il est donc utiliser pour caractériser chaque pixel d'une nouvelle image. On vérifie l'existence dans le modèle d'un codeword répondant aux mêmes contraintes précédemment décrites que celles du pixel observé. S'il existe un codeword du modèle correspondant au pixel observé, alors il est étiqueté comme appartenant au fond et le codeword correspondant est mis à jour. Sinon, il est étiqueté comme appartenant à un objet mobile. Algorithme 2 résume le processus d'extraction. La figure 3.1 présente les résultats de l'utilisation de cet algorithme sur des séquences. Ces

Algorithme 2 : Extraction des pixels de premier plan

1 $p_t(R, G, B)$
2 **for** *all codewords* **do**
3 find the codeword c_m matching to p_t based on :
(a) colordist $(p_t, v_m) <= \epsilon_2$
(b) (brightness $(I, \check{I}_m, \hat{I}_m)$) = true
Update the matched codeword as in Step 10 in the algorithm of background modeling.
4

$$BGS(Su_k) = \begin{cases} \textbf{foreground} & \text{if there is no match} \\ \textbf{background} & \text{otherwise} \end{cases}$$

FIGURE 3.1 – Résultats de la détection. Sur la première ligne nous avons les images originales, sur la seconde nous avons les vérités de terrain (détection idéale). La troisième ligne montre l'extraction de premier plan réalisée par la méthode de "Codebook".

résultats sont obtenus en implémentant l'algorithme avec le langage de programmation C++ avec la bibliothèque OpenCv. Vu les performances de cet algorithme, de nombreux travaux de recherche ont essayé d'améliorer le taux de détection et/ou de réduire le taux de fausses détections. Dans cette optique nous avons aussi investiguer deux approches pour

améliorer le taux de détection. La première approche fait une combinaison de la méthode de "Codebook" et d'un algorithme de détection de contour. Tandis que la seconde approche est une approche beaucoup plus basée sur les régions homogènes de l'image.

3.3 Combinaison de la méthode basée sur le "Codebook" et d'un algorithme de détection de contour

Dans cette section nous détaillons la première contribution pour la détection des objets mobiles. Ce changement consiste à combiner l'algorithme Codebook avec un algorithme de détection de contour. En effet plusieurs travaux de recherches ont exploré la possibilité de combiner l'algorithme Codebook avec d'autres algorithmes :

— combinaison avec un modèle gaussien [12] ;

— combinaison avec les caractéristiques locales de l'image (LBP) [10, 11].

Le fonctionnement de ces algorithmes est globalement le même. Les auteurs utilisent les méthodes (modèle gaussien et LBP) pour confirmer ou infirmer les pixels de premier plan détectés par l'algorithme de Codebook. Ces algorithmes ont des performances acceptables. Dans cette même optique nous avons proposé une combinaison avec les détecteurs de contour. L'idée d'utiliser des détecteurs de contour provient du fait que l'on retrouve des zones non détectées à l'intérieur des objets dont le contour est pourtant bien extraites avec l'algorithme de "Codebook" proposé par [9].

Dans le domaine de l'analyse d'images, la détection de contours est une étape préliminaire et importante à de nombreuses applications. C'est une technique de réduction d'information dans les images. Elle consiste a extraire les parties les plus informatives d'une image. Sa fonction est d'identifier les frontières des régions homogènes dans l'image. Dans nos travaux [13], nous avons utilisés trois algorithmes de détection de contours (détecteur de Sobel, méthode du Laplacien, détecteur de Canny).

3.3.1 Algorithme de Sobel

Le filtre de Sobel est un opérateur utilisé pour le traitement d'image et dont le but est de détecter les contours. Il s'agit d'un des opérateurs les plus simples qui donne toutefois des résultats corrects. Il se base sur le calcule du gradient de l'intensité de chaque pixel. Cette valeur indique la direction de la plus forte variation du clair au sombre, ainsi que le taux de changement dans cette direction. Pour cela, on utilise des matrices de convolution G_x et G_y données respectivement par les expressions 3.1 et 3.2.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{3.1}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{3.2}$$

En utilisant ces deux matrices, et en considérant une image originale I on a : $I_x = G_x \times I$ et $I_y = G_y \times I$. I_x et I_y représentent respectivement les approximations des gradients horizontaux et verticaux. Ainsi la norme du gradient est donnée par $\sqrt{I_x^2 + I_y^2}$ et la direction du gradient est donnée par $atan2(I_x, I_y)$.

3.3.2 Algorithme du Laplacien

La méthode consiste à calculer le passage par zéro de la valeur du Laplacien. Dans le cas d'une approche dérivée seconde, on dispose donc de la valeur du laplacien en chaque point de l'image soit de la fonction D On considére que les points de contours sont localisés aux passages par zéro du laplacien. Si le calcul du laplacien était exact il suffirait de sélectionner les points M tels que D(M)=0. Mais comme généralement l'approximation du laplacien est assez bruitée, on détecte les points où il change de signe. Une dernière

étape de seuillage est là encore nécessaire afin d'éliminer les points de trop faible gradient. L'extraction de ces passages par zéro s'effectue classiquement en trois étapes :

1. détermination d'une image de polarité.
2. détection du passage par zéro.On calcule une image Iz telle que Iz(M)=1 correspond à une transition 0-1 ou 1-0 dans Ip. On remarque que le choix de la localisation du passage par zéro au point de laplacien positif ou négatif revient, comme pour l'extraction des extréma locaux, à définir les points de contour dans la région la plus claire ou la plus foncée.
3. seuillage des passages par zéro. L'élimination des passages par zéro de faible norme de gradient peut s'effectuer par un algorithme de seuillage quelconque. L'algorithme de seuillage par hystérésis décrit pour l'approche dérivée première peut par exemple être utilisé. On peut aussi se servir du fait que les passages par zéro extraits définissent des lignes fermées délimitant les régions de points connexes où le laplacien est positif ou négatif. Des méthodes reposant sur le suivi de ces frontières et sur un calcul local du gradient peuvent aussi être utilisées

3.3.3 Algorithme de Canny-Deriche

Le filtre de Canny aussi est utilisé pour la détection des contours. Il a été proposé par Canny [67]. Pour sa mise en oeuvre, il faut suivre un certain nombre d'étapes. La première étape est la réduction du bruit. Ceci permet d'éliminer les pixels isolés qui pourraient conduire à l'obtention de fortes réponses lors du calcul du gradient, conduisant ainsi à de faux positifs. Ainsi un filtrage gaussien 2D est utilisé. Ce filtrage se base sur l'opérateur de convolution suivante :

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.3}$$

et un masque 5×5 discret avec $\sigma = 1.4$. Il faut souligner que plus le masque est grand, moins le détecteur est sensible au bruit et plus l'erreur de localisation grandit. Après le

filtrage, l'étape suivante est d'appliquer un gradient qui retourne l'intensité des contours. L'opérateur utilisé permet de calculer le gradient suivant les directions X et Y, il est composé de deux masques de convolution, un de dimension 3×1 et l'autre 1×3. On extrait donc la valeur du gradient pour produire la carte des gradients d'intensité ainsi que l'orientation des contours. Cette carte fournit une intensité en chaque point de l'image. Une forte intensité indique une forte probabilité de présence d'un contour. Toutefois, cette intensité ne suffit pas à décider si un point correspond à un contour ou non. Seuls les points correspondant à des maxima locaux sont considérés comme correspondant à des contours, et sont conservés pour la prochaine étape de la détection. Un maximum local est présent sur les extrema du gradient, c'est-à-dire là où sa dérivée s'annule. La différenciation des contours sur la carte générée se fait par seuillage à hysteresis. Cela nécessite deux seuils, un haut et un bas ; qui seront comparés à l'intensité du gradient de chaque point. Pour chaque point, si l'intensité de son gradient est :

— inférieur au seuil bas, le point est rejeté ;
— supérieur au seuil haut, le point est accepté comme formant un contour ;
— entre le seuil bas et le seuil haut, le point est accepté s'il est connecté à un point déjà accepté.

3.3.4 Algorithme de détection proposé

L'algorithme que nous proposons se base sur l'un de ces trois détecteurs. Après l'utilisation de l'algorithme de Codebook, nous détectons les enveloppes convexes des contours c_{1n} de l'image seuillée. L'enveloppe convexe d'un contour est le plus petit polygone convexe pouvant contenir tous les points du contours. La détection de l'enveloppe convexe permet l'obtention des contours fermés. Pour trouver l'enveloppe convexe d'un ensemble de points nous utilisons le parcours de Graham comme l'illustre la figure 3.2. Nous partons du point de l'ensemble ayant la plus petite valeur sur l'axe des abscisses. Ce point est choisi car il est l'un des sommets du polygone représentant l'enveloppe convexe. S'il y a égalité entre

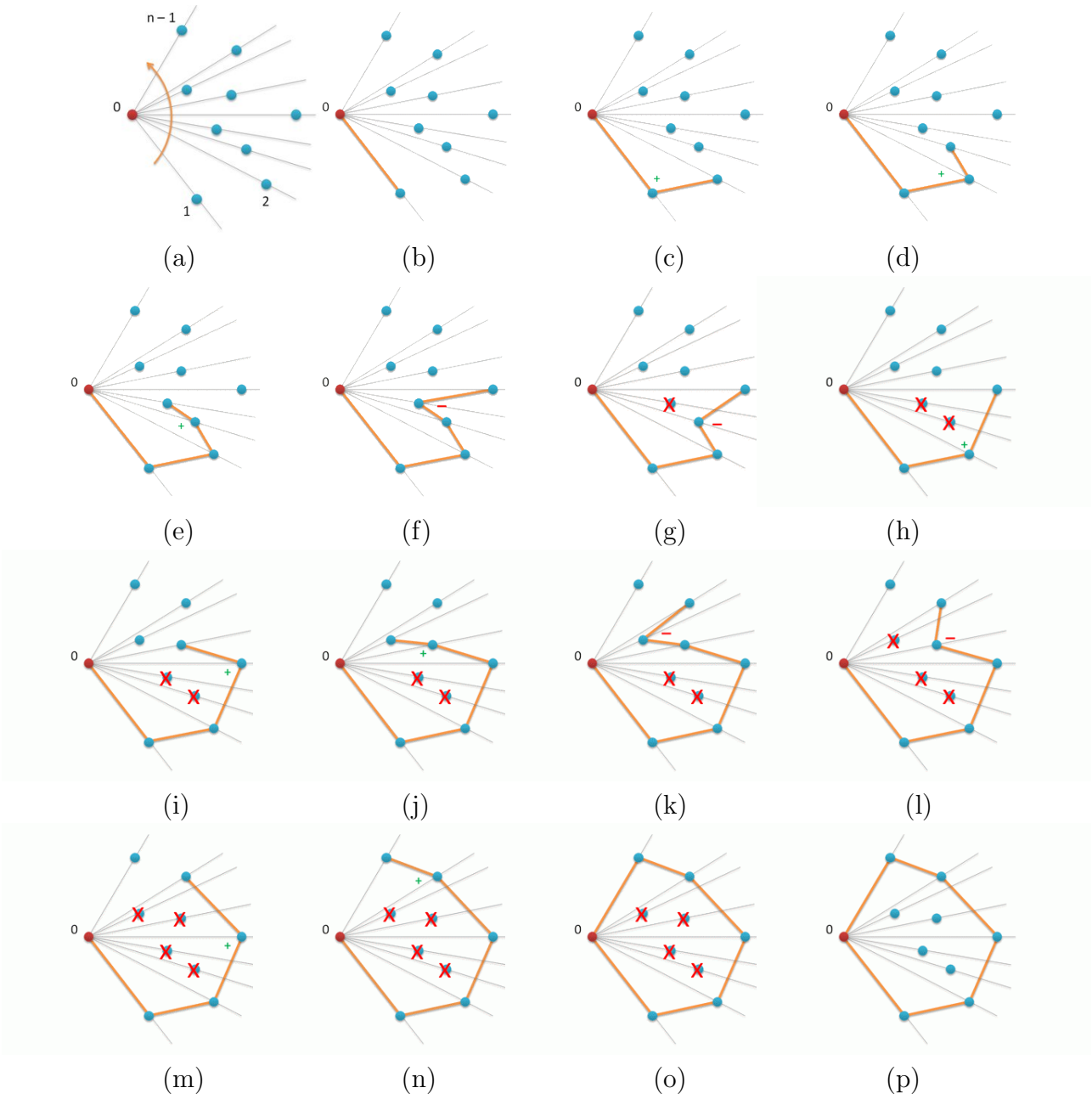

FIGURE 3.2 – Exemple de détection de l'enveloppe convexe d'un ensemble de points.

un ou plusieurs points, l'algorithme choisit parmi eux le point de plus petite ordonnée. L'ensemble des points est trié en fonction de l'angle que chacun d'entre eux fait avec l'axe des abscisses relativement au point de départ. Pour ce faire dans notre travail, nous avons implémenté le tri par tas car il a une complexité de O(n log(n)) et utilisé le produit en

croix des coordonnées pour connaître les positions relatives des points. Ainsi, nous disposons d'un tableau contenant les points triés. On considère ensuite successivement les séquences de trois points contigus dans le tableau de points triés, vus comme deux couples successifs. Pour chacune de ces paires de couples, on évalue si passer du premier couple au second constitue un "tournant à gauche" ou un "tournant à droite". Si c'est un "tournant à droite", cela signifie que l'avant dernier point considéré (le deuxième des trois) ne fait pas partie de l'enveloppe convexe, et qu'il doit être rejeté. Cette analyse se répète ensuite, tant que l'ensemble des trois derniers points est un "tournant à droite". Dès que l'on rencontre un "tournant à gauche", l'algorithme passe au point suivant du tableau. Si l'on rencontre trois points alignés, à n'importe quelle étape que ce soit, on peut choisir de conserver ou de rejeter le point considéré, au choix, suivant la définition que l'on choisit pour l'enveloppe convexe. Dans nos travaux, nous avons décidé de conserver ses points. Pour déterminer si trois points constituent un "tournant à gauche" ou un "tournant à droite" on procède comme suit. Considérons les trois points (x_1, y_1), (x_2, y_2) et (x_3, y_3), il faut calculer le sens du produit vectoriel des deux vecteurs définis par les points (x_1, y_1), (x_2, y_2) et (x_1, y_1), (x_3, y_3), donné par le signe de l'expression 3.4.

$$(x2 - x1)(y3 - y1) - (y2 - y1)(x3 - x1) \tag{3.4}$$

Si le résultat est nul alors les points sont alignés. S'il est positif, les trois points constituent un "tournant à gauche", dans le cas contraire c'est un "tournant à droite". Ce processus retournera finalement au point auquel il a commencé. Alors l'algorithme sera terminé et on obtiendra alors les points formant l'enveloppe convexe.

Dans le même temps nous détectons les contours de l'image originale convertie en niveaux de gris en utilisant l'un des trois détecteurs. Le seuillage avec le détecteur se fait à deux niveaux comme le montre le pseudo code présenté par Algorithme 3. Dans

Algorithme 3 : Procédure de seuillage

Input : grayscale image G
Output : thresholded image t

```
t ← detectedge (G)
for each pixel p_i of t do
    if intensity of p_i ≥ maximum of t's pixel intensity×(1−θ) then
        intensity of p_i ← 255
    else
        intensity of p_i ← 0
```

l'algorithme 3, detectedge est une fonction qui implémente un des 3 détecteurs de contours étudiés. Cela permet de ressortir les contours les plus intéressants. Le premier niveau de seuillage se fait à l'aide du détecteur. Une seconde sélection se fait en utilisant la valeur φ (confère 3.5).

$$\varphi = G(1 - \theta) \tag{3.5}$$

Dans l'équation 3.5, G représente le gradient maximal de l'image tandis que θ est une variable dont la valeur est comprise entre 0 et 1. La valeur de θ dépends des caractéristiques de la scène. Ainsi plus la scène est texturée plus grande doit être la valeur de θ. Après ce double seuillage, nous recherchons aussi les enveloppes convexes c_{2n}. Pour finir une comparaison entre les différentes valeurs des pixels est faite. Le but de cette comparaison est de voir si les pixels détectés par l'algorithme de Codebook sont réellement des pixels d'un objet. Un pixel sera considéré comme étant pixel de premier plan s'il est considéré comme pixel de premier plan à la fois par c_1 et c_2. Un pixel est considérer comme pixel de premier plan s'il appartient à une enveloppe convexe. L'algorithme détaillé est présenté dans [13] et les diverses étapes sont schématiquement représentées par la figure 3.3.

Le choix du détecteur dépend de l'application visée. Par exemple si l'application est une application temps réelles alors on utilisera le détecteur de Sobel. Sa complexité est moindre et donc la combinaison avec le Codebook sera moins coûteuse en terme de temps de calcul. Les autres cas pouvant favoriser le choix d'un détecteur au détriment d'un autre

sont spécifier dans [13].

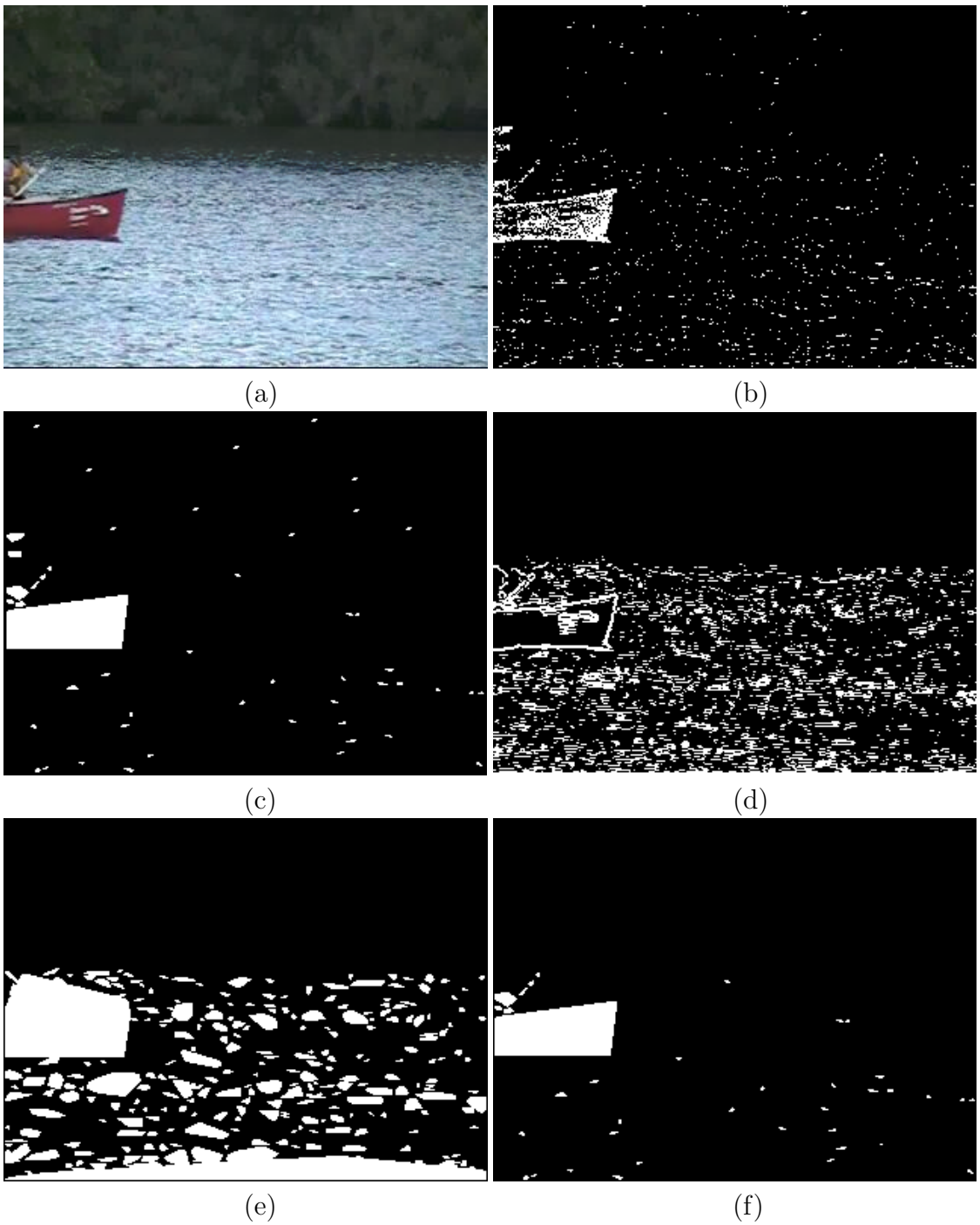

FIGURE 3.3 – Représentation schématique de l'algorithme proposé. (a) : image originale - (b) : résultat de la détection avec Codebook - (c) : enveloppe convexe des contours de (b) - (d) : seuillage avec le détecteur de contour (Sobel) - (e) : enveloppe convexe des contours de (d) - (f) : résultat final de la détection

3.4 Algorithme basé sur l'approche "Codebook" et l'utilisation des superpixels

Cette section présente la seconde méthode de détection de mouvements basée sur l'approche "Codebook" que nous avons proposée. Nous avons pensé à cet algorithme à cause de deux raisons fondamentales. Premièrement l'observation des résultats de l'approche "Codebook" montre que les fausses détections sont généralement situées dans les zones sombres de l'image. Les couleurs sombres (et donc moins lumineuses) sont par nature plus difficiles à différencier et conduisent à une incertitude plus grande sur leur classification finale. Ainsi, la luminosité est un facteur très important dans la comparaison des distorsions de couleurs entre deux pixels. Pour cela nous avons choisi un espace de couleur qui sépare la luminosité de la couleur comme l'ont fait Doshi et al. [14]et Fang et al. [15]. Contrairement à [14], dans lequel les auteurs convertissent les pixels en HSV et à [15] dans lequel les pixels sont considérés dans l'espace HSL, nous avons décider de convertir les coordonnées des pixels dans l'espace CIE L*a*b*. Défini en 1976 par la commission internationale de l'éclairage (CIE), CIE L*a*b* est un espace colorimétrique qui caractérise les couleurs par trois grandeurs. La première grandeur (L*) représente la luminance et les deux autres grandeurs (a* et b*) expriment l'écart de la couleur par rapport à celle d'une surface grise de même clarté, comme la chrominance de la vidéo. Pour convertir les pixels de l'espace de couleur RGB dans l'espace de couleur CIE L*a*b*, nous transformons les coordonnées dans l'espace CIE XYZ en utilisant l'expression 3.6.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.433953 & 0.376219 & 0.189828 \\ 0.212671 & 0.715260 & 0.0772169 \\ 0.017758 & 0.109477 & 0.872765 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.6}$$

Après cela la conversion vers l'espace CIE L*a*b* se fait en utilisant expression 3.7.

$$\begin{aligned} L^*= &\quad 116f(Y/Y_n) - 16 \\ a^*=&500[f(X/X_n) - f(Y/Y_n)] \\ b^*=&\ 200[f(Y/Y_n - f(Z/Z_n)] \end{aligned} \tag{3.7}$$

Dans l'expression 3.7, la fonction $f(t)$ est définie comme le montre l'expression 3.8, les valeurs $X_n = 95.0456$, $Y_n = 100$ et $Z_n = 108.8754$.

$$f(t) = \begin{cases} t^{\frac{1}{3}} & \text{si } t > (\frac{6}{29})^3 \\ \frac{1}{3}(\frac{29}{6})^2 t + \frac{4}{29} & \text{sinon} \end{cases} \tag{3.8}$$

La deuxième raison est la réduction du nombre de données manipulé par l'algorithme. Ce qui nous a conduit à penser à l'utilisation des superpixels. L'utilisation des superpixels a pris de l'ampleur dans la réalisation des applications de vision par ordinateur. La segmentation en superpixels est la simplification de l'image en un nombre K de régions homogènes. Chaque région est un superpixel et les pixels de chaque région ont des caractéristiques très voisines. Le nombre de superpixels est assez grand mais nettement inférieur au nombre de pixels. De nos nombreuses stratégies de regroupement des superpixels existent. Mais nous avons choisi celle proposée par Schick et al. [16]. Le pseudo code de leur stratégie est donné par l'algorithme 4. Dans cet algorithme, nous considérons que les images de la séquence ont pour taille $N \times M$, et que ces images vont être subdivisées en K superpixels. Chaque superpixel possède approximativement $\frac{N \times M}{K}$ pixels et la région centrale est approximativement $S = \sqrt{\frac{N \times M}{K}}$.

Algorithme 4 : segmentation en superpixels

1 Initialize cluster centers $C_k = [l_k, a_k, b_k, x_k, y_k]^T$ by sampling pixels at regular grid steps S.
2 Perturb cluster centers in an $n * n$ neighborhood, to the lowest gradient position using expression 3.9.
3 **repeat**
4 **for** *each cluster center* C_k **do**
5 Assign the best matching pixels from a $2S \times 2S$ square neighborhood around cluster center according to the distance measure (using expression 3.10).
6 Compute new cluster centers and residual error E {L1 distance between previous centers and recomputed centers}.
7 **until** $E <= threshold$
8 Enforce connectivity.

$$\begin{aligned} G(x,y) &= ||I(x+1,y) - I(x-1,y)||^2 \\ &\quad + ||I(x,y+1) - I(x,y-1)||^2 \end{aligned} \tag{3.9}$$

$$d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2}$$

$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2}$$

$$D_s = d_{lab} + \frac{m}{S} d_{xy} \tag{3.10}$$

Dans 3.9, 3.10, d_{lab} est la distance *lab*, d_{xy} est la distance euclidienne et $I(x, y)$ est le vecteur lab correspondant au pixel à la position (x, y). Les auteurs dans [16] ont prouver l'efficacité de la stratégie. Ils obtiennent une bonne qualité de segmentation comme le montre la figure 3.4 et une meilleure complexité que les méthodes de l'état de l'art dans la catégorie segmentation en superpixels.

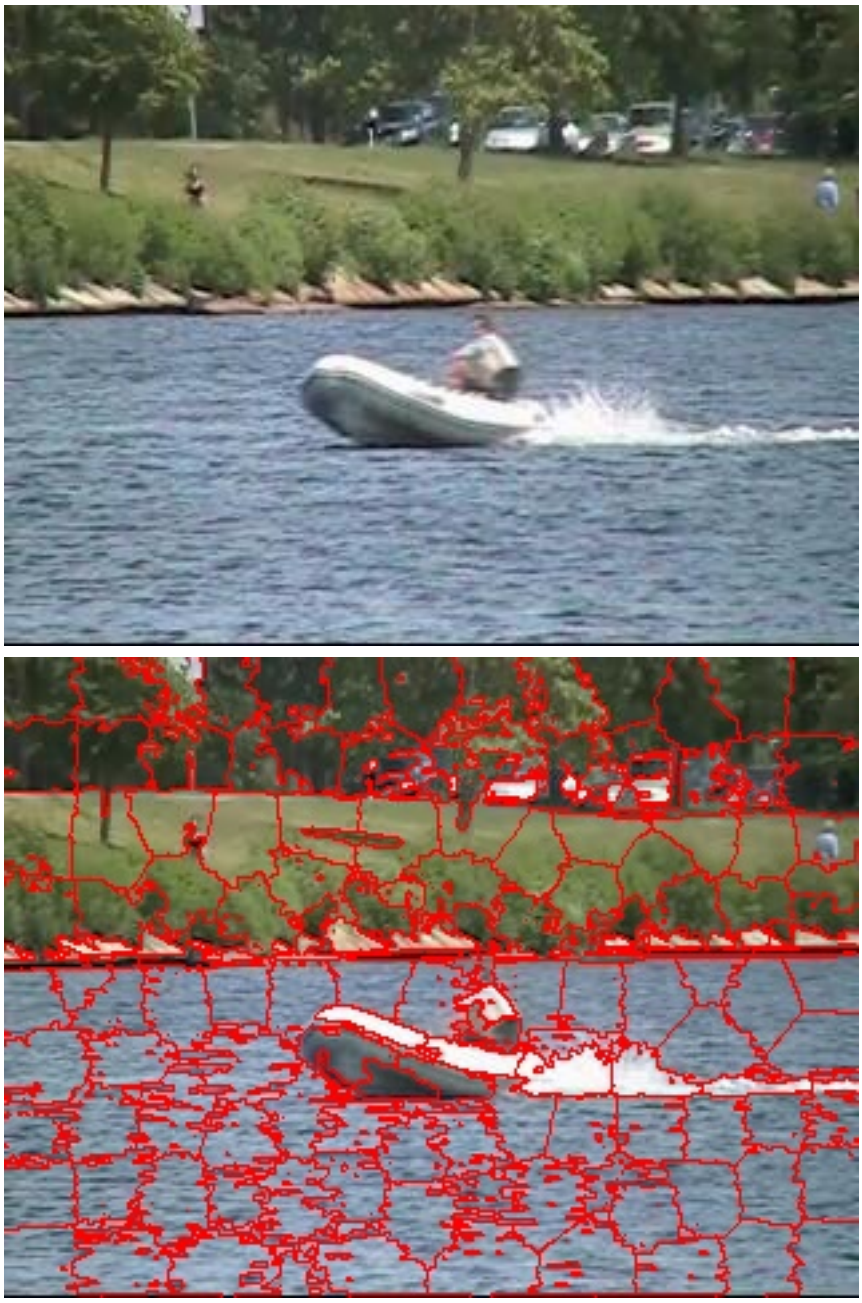

FIGURE 3.4 – Segmentation en superpixels. La première ligne représente l'image originale tandis que la seconde représente le résultat de la segmentation en superpixels.

Ainsi nous intégrons les superpixels dans l'algorithme de Codebook initial tout en faisant le changement d'espace de couleur. Le modèle de fond est donc construit sur les superpixels. Soit $P = \{s_1, s_2, ..., s_k\}$ les K superpixels obtenus après la segmentation en

superpixels. Une discussion sur le nombre de superpixels est menée dans l'article Mousse et al. [17].Chaque superpixel $s_j, j \in \{1, 2, ..., k\}$ est composé approximativement de m pixels. Avec chaque superpixel we construisons un "codebook" $C = \{c_1, c_2, ..., c_L\}$ qui contiens L "codewords" $c_i, i \in \{1, 2, ..., L\}$. Chaque "codewords" c_i est composé d'un vecteur $v_i = (\bar{a}_i, \bar{b}_i)$ et d'un 6-uplet $aux_i = \{\check{L}_i, \hat{L}_i, f_i, p_i, \lambda_i, q_i\}$ in which $\check{L}_i, \hat{L}_i$. La distorsion de la luminosité est évaluée seulement sur la composante L* tandis que la distorsion de la couleur est évaluées en utilisant les composantes a* et b*. Ainsi la distorsion de la couleur est obtenue en utilisant l'expression 3.11 et la distorsion de la luminosité suivant l'expression 3.12.

$$Colordist(p_t, c_i) = \sqrt{||p_t||^2 - C_p^2} \tag{3.11}$$

Dans l'expression 3.11 :

— $||p_t||^2 = \bar{a}^2 + \bar{b}^2$;

— $C_p^2 = \frac{(\bar{a}_i\bar{a} + \bar{b}_i\bar{b})^2}{\bar{a}_i^2 + \bar{b}_i^2}$.

$$L_{low} \leq \bar{L} \leq L_{hi} \tag{3.12}$$

Dans l'expression 3.12, $L_{low} = \alpha\hat{L_i}$, $I_{hi} = min\{\beta\hat{L}, \frac{L'}{\alpha}\}$. $\bar{L}, \bar{a}, \bar{b}$ représentent respectivement les valeurs moyennes des composantes L*, a* and b* des pixels du superpixel. Après la construction de fond l'extraction des pixels de premier plan se font aussi sur les superpixels.

3.5 Expérimentations et analyse des performances

Pour la validation de nos algorithmes, nous avons utilisé des banques de données publiques. Ces banques de données ont pour objectif de tester l'efficacité des algorithmes de détection proposés par les équipes de recherches. La présente section est subdivisée

en deux sous sections. La première montre le cadre expérimental tandis que la seconde section présente et analyse les performances.

3.5.1 Expérimentations

Les séquences utilisées pour tester nos propositions sont des séquences reconnues et largement utilisées par la communauté de vision par ordinateur. Ce sont des séquences disponibles dans une banque de données disponible à l'adresse `http://www.changedetection.net/`. La banque de données contient plusieurs sortes de séquences qui sont mis en place pour évaluer les algorithmes de détection par rapport à divers challenges. Chaque séquence est fournie avec un nombre d'image pour apprentissage, un nombre d'image de test et une détection idéale pour les images de test. Les caractéristiques de ces séquences sont détaillés dans Goyette et al. [65]. Toutes les expériences ont été conduites en utilisant une machine ayant un processeur Intel-Core7@2.13Ghz avec une mémoire RAM de 4GB. Les algorithmes ont été implémentés en utilisant le langage de programmation C++ avec la bibliothèque OpenCv. Cette bibliothèque a été choisie car elle offre de nombreux avantages pour la réalisation des modules de vision par ordinateur.

La figure 3.5 présente les captures d'écran des détections obtenues en utilisant le premier algorithme proposé (combinaison "Codebook" et d'un détecteur de contour), l'approche "Codebook" et l'approche MoG sur les séquences "canoe" et "fountain01". La valeur de θ est 0.85 pour la séquence "canoe" et 0.80 pour la séquence "fountain01". Les valeurs de θ sont obtenues à partir de nos expérimentations. Pour ces séquences ce sont les valeurs qui permettent d'avoir les meilleurs taux de détection.

Les figures 3.6 et 3.7 présentent les captures d'écran des détections obtenues en utilisant le second algorithme proposé respectivement sur les séquences "boats" et "fall". Les paramètres de l'algorithme de segmentation en superpixels utilisés sont ceux suggérés par

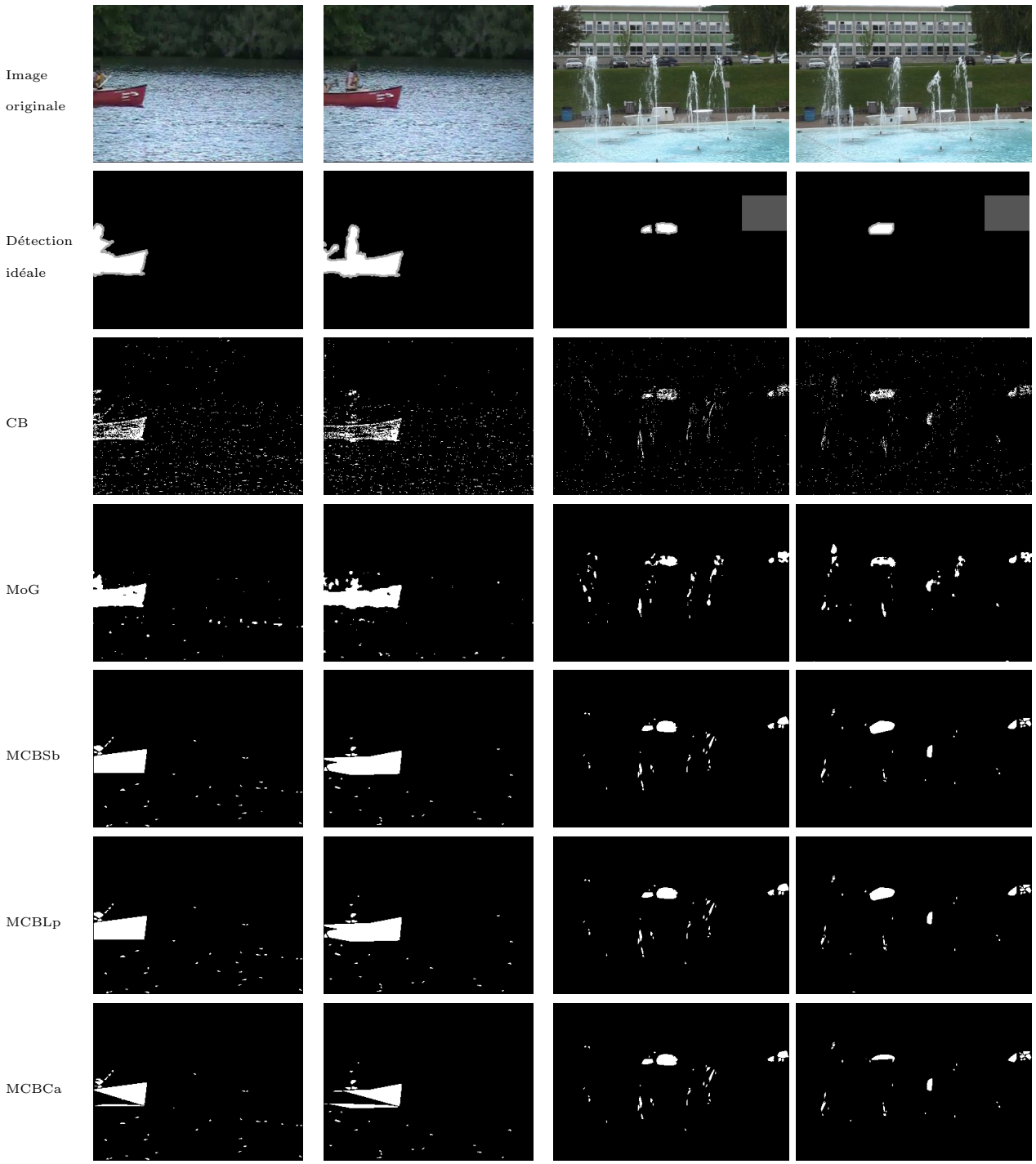

FIGURE 3.5 – Résultat de détection

Schick et al. [16]. Le nombre de superpixels utilisé dépends de la taille de l'image. En effet pour chaque image le nombre de pixels est divisé par 50. Ainsi si la taille de l'image est

$m \times n$ alors nous procédons à la construction de $(\frac{m \times n}{50})$ superpixels. Cette valeur a été obtenue en se basant sur nos expérimentations. Ces expérimentations ont démontrées qu'en utilisant cette valeur $(\frac{m \times n}{50})$, on obtient un meilleur compromis entre le temps d'exécution et le taux de détection.

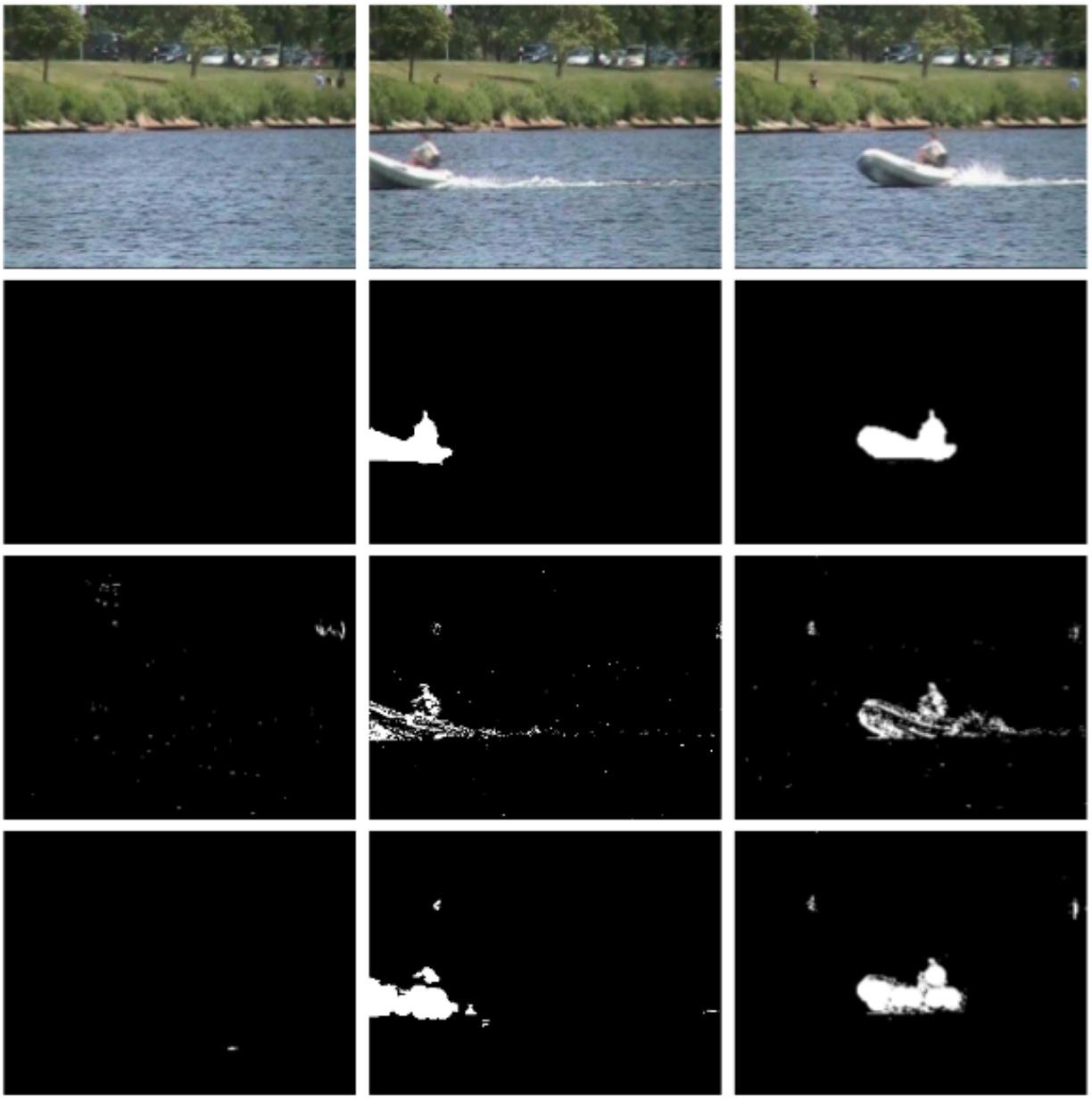

FIGURE 3.6 – Résultat de détection. La première ligne présente l'image originale, la seconde ligne présente les détections idéales associées à chaque image. La troisième ligne présente la détection basée sur le "Codebook" [9]. La dernière ligne présente les résultats de détection basée sur notre algorithme.

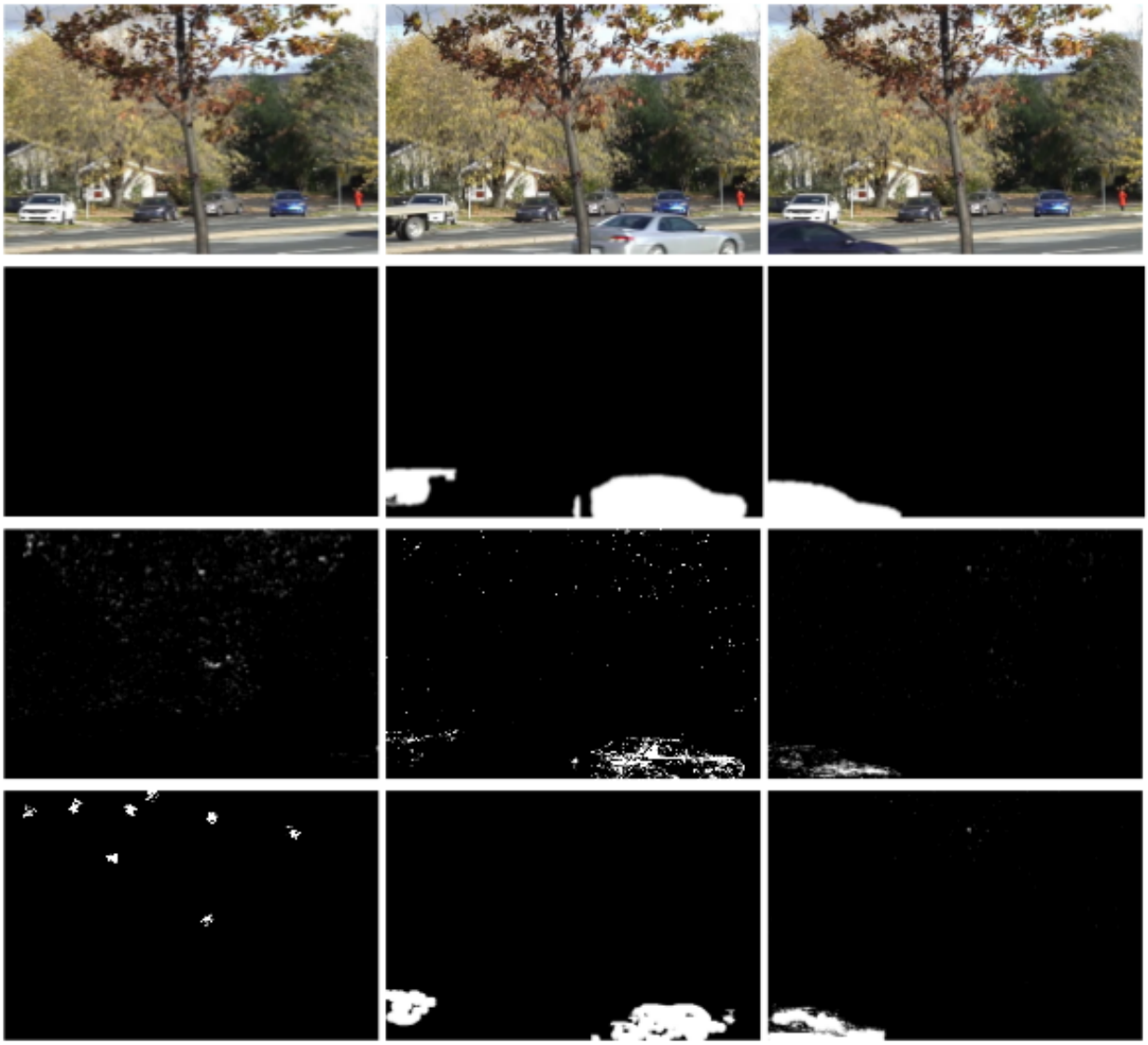

FIGURE 3.7 – Résultat de détection. La première ligne présente l'image originale, la seconde ligne présente les détections idéales associées à chaque image. La troisième ligne présente la détection basée sur le "Codebook" [9]. La dernière ligne présente les résultats de détection basée sur notre algorithme.

3.5.2 Analyse des performances

Pour la validation de nos algorithmes, nous avons utilisés des critères bien connus dans la communauté. Ces critères sont basés sur des métriques : Vrai Négatif (TN), Vrai Positif (TP), Faux Négatif (FN), Faux Positif (FP). Ces métriques sont obtenues en se basant sur les résultats de détection des algorithmes et les résultats de la détection idéales.

L'obtention de ces métriques est résumée par le tableau 3.1. D'après ce tableau :

		Résultat de détection utilisant le système	
		Premier plan	Arrière plan
Vérité de terrain	Premier plan	TP	FN
	Arrière plan	FP	TN

TABLE 3.1 – Identification des métriques

— un pixel est un considéré comme pixel Vrai Négatif si le résultat de la détection obtenue en utilisant l'algorithme et le résultat de la détection idéale classe le pixel comme étant un pixel n'appartenant pas à un objet en mouvement ;

— un pixel est un considéré comme pixel Vrai Positif si le résultat de la détection obtenue en utilisant l'algorithme et le résultat de la détection idéale classent le pixel comme étant un pixel appartenant à un objet en mouvement ;

— un pixel est un considéré comme pixel Faux Négatif si le résultat de la détection obtenue en utilisant l'algorithme classe à tort le pixel comme étant un pixel n'appartenant pas à un objet en mouvement ;

— un pixel est un considéré comme pixel Faux Positif si le résultat de la détection obtenue en utilisant l'algorithme classe à tort le pixel comme étant un pixel appartenant à un objet en mouvement ;

En utilisant ces métriques, nous calculons :

— le **Taux de faux positif** (FPR) en utilisant l'expression (3.13) ;

$$FPR = 1 - \frac{TN}{TN + FP} \tag{3.13}$$

— le **Taux de vrai positif** (TPR) en utilisant l'expression (3.14) ;

$$TPR = \frac{TP}{TP + FN} \tag{3.14}$$

— la **Précision** (PR) en utilisant l'expression (3.15) ;

$$PR = \frac{TP}{TP + FP} \tag{3.15}$$

— la **F-mesure** (FM) en utilisant l'expression (3.16) ;

$$FM = \frac{2 \times PR \times TPR}{PR + TPR} \tag{3.16}$$

Ces valeurs une fois obtenues servent de critères de comparaison entre différentes approches de détection de mouvements.

Nous calculons ces valeurs pour la première approche proposée (section 3.3) tout en comparant les résultats obtenus avec ceux de Kim et al. [9] et de Stauffer et al. [36]. Les résultats sont présentés dans les tableaux 3.2 et 3.3. Dans ces tableaux :

— sur une ligne, les valeurs gras sont les plus optimales.

En analysant ses valeurs, nous constatons que la combinaison de l'algorithme "Codebook"

Critères	CB	MoG	MCBSb	MCBLp	MCBCa
FPR	1.62	0.36	0.29	0.31	**0.24**
PR	41.01	**89.82**	86.29	86.01	81.89
FM	35.08	**88.17**	63.08	65.09	43.39

TABLE 3.2 – Comparaison des différentes valeurs obtenues en faisant l'expérimentation sur la séquence "canoe"

Critères	CB	MoG	MCBSb	MCBLp	MCBCa
FPR	1.09	1.59	**0.43**	0.45	**0.43**
PR	2.24	4.01	6.82	**7.31**	6.79
FM	4.17	7.63	11.58	**12.50**	11.53

TABLE 3.3 – Comparaison des différentes valeurs obtenues en faisant l'expérimentation sur la séquence "fountain01".

et d'un détecteur de contour est toujours plus performant que l'algorithme de "Codebook". Ainsi la détection des contours apporte une certaines à l'algorithme de détection basé sur "Codebook". Ces résultats aussi démontrent l'importance de l'utilisation de l'algorithme de détection de contour pour la réduction du taux de fausses alarmes. Aussi, il convient de souligner que dans certains cas où l'algorithme de MoG a de meilleur performance que celui de CB, la combinaison avec un détecteur de contour permet de rivaliser en terme de performance. L'utilisation de l'opérateur de Sobel le temps de traitement de l'algorithme "Codebook" de 19.55% (23.33% pour l'opérateur "laplacian of Gaussian" et 28.15% pour le détecteur de contour canny). Ainsi un des critères pour motiver le choix d'un algorithme de détection de contour il faut tenir compte du type d'application. Si le type d'application visé est une application où l'aspect temps-réel est important il faut porter son choix sur l'algorithme de Sobel.

Le second algorithme proposé (section 3.4) aussi a été évalué. Ces performances ont été comparées aux performances des autres algorithmes basés sur l'approche "Codebook"

Critères	CB	CB_HSV	CB_HSL	CB_YUV	CB_LAB
FPR	0.23	0.25	**0.21**	0.27	**0.21**
PR	0.87	0.86	0.89	0.80	**0.91**
FM	0.60	0.62	0.64	0.65	**0.66**

TABLE 3.4 – Comparaison des différentes valeurs obtenues en faisant l'expérimentation sur la séquence "boats".

Critères	CB	CB_HSV	CB_HSL	CB_YUV	CB_LAB
FPR	0.31	0.33	0.25	0.38	**0.23**
PR	0.56	0.60	0.63	0.41	**0.67**
FM	0.41	0.47	0.51	0.43	**0.54**

TABLE 3.5 – Comparaison des différentes valeurs obtenues en faisant l'expérimentation sur la séquence "fall".

Critères	CB	CB_HSV	CB_HSL	CB_YUV	CB_LAB
FPR	0.16	0.18	0.15	0.21	**0.13**
PR	0.41	0.46	0.46	**0.52**	0.48
FM	0.35	0.38	0.37	**0.41**	0.39

TABLE 3.6 – Comparaison des différentes valeurs obtenues en faisant l'expérimentation sur la séquence "canoe".

avec une extension au niveau des pixels. Ces valeurs sont reportées dans les tableaux 3.4, 3.5, 3.6. Dans ces tableaux :

— CB signifie codebook [9] ;

— CB_HSV signifie l'approche proposée par Doshi et al. [14] ;

— CB_HSL signifie l'approche proposée par Fang et al. [15] ;

— CB_YUV l'approche proposée par Cheng et al. [66] ;

— CB_LAB signifie l'approche proposée par la section 3.4 ;

— sur une ligne, les valeurs gras sont les plus optimales.

L'analyse de ces résultats montre que l'utilisation de l'espace de couleur CIE L*a*b* permet d'avoir le plus bas taux de fausse alarme. La plupart du temps cet algorithme permet d'améliorer la précision de l'algorithme est améliorée lorsqu'on passe dans l'espace de couleur CIE L*a*b**. L'utilisation des superpixels réduit la complexité de l'algorithme et ainsi nous avons un gain non négligeable en temps d'exécution. Cela peut s'avérer très important dans la réalisation des applications de surveillance tenant particulièrement compte de l'aspect temps réel.

3.6 Conclusion

Dans ce chapitre nous avons présenté l'algorithme de Codebook ainsi que nos contributions basées sur cet algorithme. Ces contributions se basent sur l'algorithme de Codebook car il présente de très bon résultats en terme de détection. La première exploite l'information sur les contours dans l'image en combinant l'algorithme de Codebook avec un détecteur de contour et la seconde est une approche beaucoup plus axée sur les régions homogènes de l'image. Cette dernière approche est réalisée en utilisant un algorithme de segmentation en superpixels. Ces deux algorithmes ont pour but d'améliorer la détection de mouvements dans un système de vidéosurveillance intelligent. Nous avons effectué des expérimentations en utilisant des séquences publiques dans le but de valider nos approches. Les résultats de ces expérimentations ont démontré l'efficacité des diverses approches.

Deuxième partie

Système de détection de chute

Chapitre 4

Techniques de détection de chute

Sommaire

4.1 Introduction

Dans ce chapitre nous présentons un état de l'art des systèmes de détection de chutes. Les chutes chez les personnes âgées représentent un problème important de santé. Nous estimons qu'environ 30 % des personnes âgées de 65 ans et plus chutent chaque année, entraînant des conséquences néfastes sur le plan individuel, familial et social. Plusieurs interventions efficaces ont été développées dans le but de prévenir les chutes chez les personnes âgées et les médecins sont appelés à les inclure dans leur pratique. Dans le présent manuscrit, nous ne présentons que les systèmes de vidéosurveillance intelligents pour la

détection de chute. Une revue de littérature plus complète est fournie par Mubashir et al. [68]. Depuis quelques années, plusieurs travaux de recherche utilisent les systèmes de vidéosurveillance intelligents pour la détection de chutes. L'avantage majeur de ces types de système, est qu'ils ne nécessitent pas le port d'un instrument de mesure avant de détecter la chute. De plus, avec la caméra, on obtient beaucoup plus d'informations sur la personne (comme sa localisation, sa posture, ses actions,...) mais aussi sur son environnement. Ainsi, en plus de détecter des chutes, un système de vidéosurveillance intelligent pourrait permettre d'analyser le comportement de la personne. On pourra donc vérifier si elle prend correctement ses médicaments, si elle mange et dort à des heures régulières, etc. Toutes ces données sont des indicateurs du bien être de la personne et peuvent révéler des problèmes qui nécessitent un suivi et qui ont peut-être entraîné la chute.

Dans ce chapitre nous présentons un état de l'art des systèmes de vidéosurveillance pour la détection de chutes. Après avoir présenté ces systèmes, une discussion est menée en vue de motiver le choix de l'approche que nous avons fait lors de nos travaux.

4.2 Méthodes de détection de chutes à partir des séquences mono-caméra

Dans cette section, nous présentons les approches mono-caméra de détection de chute. Dans cette catégorie, Anderson et al. [69] analyse la taille des silhouettes. Le ratio largeur sur hauteur est utilisé pour détecter la chute. Ce ratio est inséré en entrée à une chaîne de Markov cachée. Liu et al. [70] utilisent une méthode basée sur les k plus proches voisins pour classifier la posture de l'intéressé. Ils utilisent comme critères de classification les histogrammes des silhouettes des images, ratios et différence entre la hauteur et la largeur.

D'autres approches se basent sur les informations contextuelles de la scène. Cela se

fait en divisant la scène en deux zones. Les zones d'activités et les zones d'inactivité. Ainsi quand l'individu est au sol dans une zone d'activités, le système conclura qu'il s'agit d'une chute. Dans cette perspective, Lee et Mihailidis [71] identifient les zones d'inactivités à partir d'une camera placé au plafond. Ces zones sont la plupart du temps les zones où sont placés les objets à utiliser (sofa, chaise, ...). Lors de la détection le périmètre et le diamètre de Feret, la position du centre et la vitesse de l'objet obtenu sont calculés. Ces informations sont mis ensemble pour détecter la position (assis ou couché) de l'individu. Dans le même sens Charif et McKenna [72] définissent un temps d'inactivité dans les zones d'inactivité. Ainsi lorsque ce temps d'inactivité est dépassé le système considère que l'individu a chuté. Toutes les formes d'inactivité dans les zones d'activités sont considérées comme des chutes. L'utilisation d'une camera placée au plafond ne permet pas toujours d'identifier de façon correct les chutes. Pour remédier à cela, Shoaib et al. [73, 74] ont utilisé des cameras placées obliquement vis à vis de la scène et ont proposés d'utiliser une stratégie non supervisée prenant en compte l'environnement. Cependant un inconvénient demeure toujours. En effet lorsque l'individu est accroupi, le système le détecte comme étant un individu qui a chuté.

Ma chute est définie cliniquement par un mouvement rapide et anormal du corps de l'homme vers le bas. Certains approches exploitent cette caractéristique de chutes. Ainsi Rougier et al. [75] suggèrent de sauvegarder l'historique des silhouettes détectées pour évaluer la posture de l'homme. La silhouette est obtenue en approximant la détection par une ellipse. L'angle d'inclinaison et le ratios entre les deux diagonales sont sauvegarder au niveau de chaque silhouette.La chute est alors détectée si on constate que l'axe est vers le bas et que l'angle d'inclinaison devient plus petit et ceci très rapidement. Dans leur article ils mentionnent que le système évolue à une vitesse d'environ 10 fps (images par seconde) pour une séquence dont les images ont pour taille 320×240. De façon similaire, Liao et al. [76] exploitent l'historique des détections aussi pour détecter les chutes. Mais

eux ils proposent de calculer l'énergie spatio-temporelle et l'angle d'inclinaison pour quantifier la posture de l'individu. Ces attributs sont transmis à un réseau Bayésien en vue de distinguer une chute accidentelle de l'action de se coucher normalement par exemple. Ils ont démontré que leur approche est plus adaptée à certains types de séquences. Chen et al. [77] présentent une combinaison de distance entre des exemples de squelettes de la personne et la variation de l'ellipse englobante de la détection obtenu. Cette distance permet de comparer l'état actuel aux états initialement définis par les squelettes prédéfinis. Lorsque l'individu se trouve dans une position inactive pendant un temps donné, le système vérifie l'état pour prendre une décision par rapport à un éventuelle chute. Plusieurs travaux ont tenté d'extraire les caractéristiques efficaces pour représenter la silhouette. Par exemple, Rougier et al. [78, 79] utilisent les histogrammes de Log-polar et Rougier et al. [78] utilisent aussi la distance Full Procustes. Ces méthodes permettent d'obtenir de bon résultats mais sont d'une complexité non négligeable. En effet selon [79], la vitesse de traitement est de 5 fps. Htike et al. [80] proposent de reconnaître la posture des individus en utilisant un algorithme d'apprentissage basée sur les chaînes de Markov cachées floues. Ils transmettent en entrée au système d'apprentissage une distribution d'histogrammes représentant la forme 2D de la silhouette. Khan and Sohn [81] utilisent par contre une chaîne de Markov cachée simple pour la détection de l'événement relatif à la chute de personnes avec une modélisation préalable des silhouettes. D'autres approches utilisent différents classificateurs avec comme entrée des paramètres 2D extraites pour caractériser les silhouettes. Nous pouvons par exemple citer les travaux de Yu et al. [82]. En effet Yu et al. utilisent un système basé sur les machines à vecteurs de support (SVM) pour détecter les chutes. Les paramètres d'entrée du système d'apprentissage sont : angle d'inclinaison de l'ellipse englobant, le ratio des axes de l'ellipse et la projection des histogrammes suivant les axes des ellipses. Les méthodes décrites jusqu'à présent n'utilisent que les informations 2D de la silhouette.

La seconde grande catégorie d'approches est celle utilisant les informations 3D de la scène. Pour ces algorithmes, la calibration de la scène est nécessaire. Grâce a la calibration, il est possible de savoir comment les objets de la scène sont projetés sur le plan focal de la camera. Inversement on peut aussi savoir a quel point physique de la scène correspond un point de l'image. Pour obtenir les paramètres de calibration, il faut mettre en correspondance les points physiques de la scène avec les points images correspondants. Pour réussir cette opération, il faut prendre des mesures fiables dans la scène réelle. Plus les mesures sont fiables et nombreuses plus les paramètres de calibration sont précis et utilisable. Ces paramètres de calibration permettent de calculer la position dans l'espace physique d'une personne détectée, sa taille et sa largeur. Ces divers informations utilisés rendent fiable les systèmes de détection de chutes. Ainsi, Rougier et al. [83] ont proposé une méthode de suivi 3D de la tête de la personne détectée. A partir du suivi de la tête, ils extraient la vitesse et le sens de déplacement de la tête. Une valeur limite est définie et en fonction de cette dernière le système déclenche l'alerte d'une chute. Leur approche marche très bien sauf que cela est sensible aux actions proches de la chute (comme s'asseoir rapidement). Cucchiara et al. [84, 85] proposent aussi un système utilisant les caractéristiques 3D de la scène. Par contre eux, ils utilisent une méthode de classification pour apprendre les divers événements. Ces événements sont vus comme étant une succession de postures (debout, assis, couché, accroupi). Les résultats sont meilleurs que ceux des algorithmes n'utilisant que des informations 2D. Mais l'obtention des caractéristiques 3D de la détection a une forte complexité algorithmique. Ce qui fait que ces systèmes demandent beaucoup de ressources pour une exécution temps réel. Aussi précisons qu'il est compliqué d'obtenir de bon paramètres de calibration en utilisant un seul caméra.

4.3 Méthodes de détection de chutes à partir des séquences multi-caméras

Dans cette section, nous présentons les algorithmes de détection de chutes utilisant un système multi-caméras. En effet l'utilisation de plusieurs caméras pour la surveillance d'un espace prend de plus en plus d'ampleur car cela permet de gérer le problème d'occultations notamment dans une maison intelligente. En effet, il est rare qu'une personne soit cachée dans la vue de plusieurs caméras à la fois dans une chambre. Nous avons deux types de systèmes multi-caméras : les systèmes multi-caméras avec des caméras ayant des vues chevauchants et les systèmes dont les caméras n'ont pas de vues chevauchants. Dans cette partie nous présentons les algorithmes utilisant des cameras avec des vues chevauchantes.

Thome et al. [86] utilisent une logique floue pour la fusion des décisions individuelles des caméras. Pour prendre la décision mono caméra ils procèdent à une correction des distorsions dues aux caméras qu'ils utilisent et estiment l'angle entre l'axe principale de l'ellipse englobant la détection de la personne et la verticale. Les décisions prises le sont à l'aide d'un LHMM qui modélise les différents états ainsi que leurs transitions. Ces décisions sont fusionnées en fonction des critères tels que la position de l'individu dans la scène, etc. Hazelhoff et al. [87] proposent d'utiliser l'analyse en composante principale pour déterminer la direction de l'axes principales de l'homme ainsi que les ratios des variances suivant les directions x et y. Un classificateur gaussien est exploité pour classifier les différents événements lier à la gestion de la maison intelligente. Le système proposé se base sur un couple de caméras perpendiculaire et il n'y a pas de contraintes de calibration de la scène. Anderson et al. [88] proposent de faire une reconstruction des voxels de la personne. A partir de cette reconstruction, un ensemble de stratégie basé sur la logique floue est exploité. Le premier niveau permet la classification de l'état de l'individu et le second état permet la reconnaissance des événements. Ils ont mis à la disposition un fra-

mework implémentant cette hiérarchie de logique floue très flexible qui peut être adapté dans divers contexte. Cette stratégie a été approfondie par les travaux de Zambanini et al. [89] et de Zweng et al. [90]. Zambanini et al. se sont beaucoup plus intéressés à réduire la complexité algorithmique du système initial proposé par Anderson et al. Ils ont proposés l'utilisation d'une méthode de raisonnement beaucoup plus souple en vue d'obtenir un meilleur rendement en matière de vitesse d'exécution sans pour autant dégrader les résultats de la détection. Zweng et al. quant à eux proposent d'extraire les caractéristiques directement depuis l'image 2D. Ils utilisent la stratégie de raisonnement proposé par Zambanini et al. Yu et al. [91] font aussi une reconstruction des voxels pour classifier les événements de chutes des autres en utilisant une machine à vecteurs de support à une classe. Auvinet et al. [94, 92] suggèrent une méthode basée sur la reconstruction 3D de la forme de la personne à partir d'un réseau de caméras. Ils ont proposés une grandeur pour quantifier le volume de voxels. Ils ont constater qu'en fonction de la distribution verticale des voxels, la posture de la personne peut être distinguée. Cette méthode permet d'obtenir de très bons résultats lorsque la couverture de la scène est bien assurée. Hung et Saito [95] proposent d'utiliser un système de deux caméras relativement orthogonaux. Ils estiment la hauteur et la largeur de l'individu en utilisant la projection homographique. En fonction des valeurs obtenues pour ces caractéristiques ils utilisent un enchaînement d'état pour détecter la chute. Ils obtiennent de bon résultats tout en minimisant la complexité du système.

L'avantage majeure de cette méthode et de toutes les méthodes utilisant des systèmes multi-caméras est la possibilité d'extraire de façon robuste les caractéristiques 3D (voxels, reconstruction 3D,..) de la personne. Cela induit un inconvénient qui n'est pas négligeable. En effet, l'obtention de bons paramètres 3D nécessite l'utilisation de plusieurs caméras. Il faut donc acquérir plus de caméras. Il faut aussi souligner que la reconstruction est très coûteuse en temps de calcul. Dans la plupart du temps pour leur mise en oeuvre,

il faut recourir à des outils informatique plus évolués (GPU par exemple) pour faire une application temps réel. Finalement, une approche multi-vues nécessite des étapes initiales : une étape de synchronisation, une étape de calibration et une étape d'enregistrement [93]. Il faudra alors faire toutes ses étapes préalables avant d'espérer avoir de bons résultats.

4.4 Méthodes de détection de chutes à partir des caméras avec vue en profondeur

Les cameras avec vue en profondeur offrent beaucoup plus d'informations sur la scène que les caméras simples. En effet, elles permettent d'obtenir à la fois les images couleurs ainsi qu'une carte de la profondeur de la scène. De plus l'introduction des nouveaux caméras avec vue en profondeur à de faible coût telles que celles de la gamme Microsoft Kinect a conduit à l'adoption de ces caméras. Dans cette section, nous présentons les algorithmes récents de détection de chutes de personnes utilisant des caméras avec vue en profondeur.

Rougier et al. [96] propose une méthode efficace d'extraction du barycentre de l'individu en se basant sur sa hauteur par rapport au sol en utilisant les séquences obtenues à l'aide d'une caméra Kinect. Leur stratégie se base principalement sur le fait que la plupart des chutes finissent au sol ou près du sol. Pour gérer les occlusions dues à la présence de meubles, il se base sur la vitesse du corps 3D obtenue avant que le corps ne soit immobile. Cela permet de minimiser les fausses détections. Les événements de chutes sont détectés en utilisant un seuil qui est obtenu de façon manuelle. Planinc et Kampel [97] se basent sur un squelette pour estimer l'orientation 3D principale du corps humain. Si l'orientation principale de la personne est parallèle au sol et si la hauteur de la personne est proche du sol alors la personne est considérée comme ayant chuter. Cette approche fonctionne mais ne prend pas en compte les événements très proches des chutes. Cet aspect a été pris en considération dans leur second travail [98]. Dans [98], Planinc et Kampel ont uti-

lisée les mêmes caractéristiques que le précédent travail mais emploient une logique floue pour la reconnaissance des événements. Zhang et al. [99], choisissent huit (8) points au niveau des articulations du corps en utilisant le SDK de Microsoft Kinect pour le calcul des caractéristiques cinématiques de l'individu ainsi que de sa taille. Il propose une hiérarchie de machines à vecteurs de support (SVM) pour reconnaître cinq types d'événements. Par contre Zhang et al. [100] proposent d'utiliser d'autres caractéristiques en vue de caractériser les événements et utilisent un système basé sur les réseau Bayésien pour prendre la décision finale. Mastorakis et Makris [101] ont proposés un modèle basé sur la vitesse. La vitesse est calculée en fonction du cube 3D englobant la détection. Dubey et al. [102] proposent d'extraire une caractéristique 3D qui permet de modéliser l'historique des mouvements sur l'image. La caractéristique est utiliser avec un système de machines à vecteurs de support pour la classification des événements.

4.5 Discussion

Dans ce chapitre, nous présentons l'état de l'art pour la détection automatique des chutes de personne à partir d'une ou plusieurs caméras fixes. Nous avons subdivisé ces approches en trois classes. Et les diverses méthodes de cette classe présentent des résultats plus ou moins acceptables en fonction de l'environnement d'application. Dans une maison intelligente, pour la détection des chutes l'utilisation d'un caméra (avec vue en profondeur ou non) n'est pas toujours suffisante pour obtenir des bons systèmes. Il faut se donner des hypothèses nécessaires et suffisantes pour pouvoir avoir des résultats acceptables. Il est donc plus recommandé d'utiliser des systèmes multi-caméras. L'utilisation de ces systèmes permet aussi la gestion des occlusions dues à la présence des meubles. Le coût des caméras simples étant devenu plus abordable, la mise en oeuvre de ces systèmes ne sont plus trop complexe. La majeure partie des approches proposées utilisent plusieurs caméras (allant de deux (02) caméras à sept (07) caméras) pour la détection de chutes.

En outre les caractéristiques extraites (les informations 3D) dans la plupart des travaux de recherches font que la complexité des systèmes est très grand. Cela induit que la mise en oeuvre de ces systèmes ne permet pas toujours d'avoir des applications temps réels en utilisant des équipements pas très sophistiqués.

Il est donc important d'affronter les deux challenges que posent la mise en oeuvre d'un système de vidéosurveillance pour la détection automatique de chute de personnes. Ces deux challenges sont :

— la réduction du nombre de cameras ;
— l'extraction des caractéristiques moins complexes.

La réduction du nombre de caméras a été discuté dans les travaux menés par Hung et Saito [95]. En effet, ils proposent après expérimentations d'utiliser un système de caméras composé de deux (02) caméras avec des vues complémentaires. Ces expérimentations ont été valider durant les nôtres. En effet nous avons constaté qu'en utilisant deux caméras avec des vues relativement perpendiculaires, les informations 2D robustes de la scène peuvent être extraites. L'autre aspect de la discussion est la réalisation des systèmes moins complexes. Dans l'état de l'art, seul le système de Hung et Saito [95] est implémenté sur un PC présentant des caractéristiques pas trop sophistiquées tout en donnant des résultats acceptables. Ils obtiennent plus de 95.8% de taux de reconnaissance des événements de chutes et 100% de taux de reconnaissance des événements de non chutes en utilisant un PC possédant un possesseur Intel Core i7 950 ayant une vitesse 3.07 GHz et 3 GB de mémoire RAM. Il est donc intéressant de penser à la réalisation d'un système moins complexe. Réaliser un système moins complexe, reviens à extraire des caractéristiques beaucoup plus simple en vue d'une détection robuste des chutes.

4.6 Conclusion

La revue de littérature ici présentée montre toute l'importance et l'attrait des systèmes de détection automatique de chutes de personnes via les systèmes de vidéosurveillance intelligente. Après avoir recensé les divers approches existantes, nous avons fait une analyse qui nous permet de justifier le système que nous avons conçu. Ce système sera présenté dans le prochain chapitre.

Chapitre 5

Proposition d'un système de détection de chutes dans un environnement multi-cameras

Sommaire

5.1 Introduction

Ce chapitre présente notre contribution en matière de la détection de chutes en utilisant un système multi-caméras. Comme démontré dans le chapitre précédent, pour détecter la chute d'un individu deux caméras ayant des vues complémentaires suffisent pour avoir

des résultats acceptables. Le système utilisé dans notre travail est basé sur deux cameras avec des vues relativement orthogonales. Le but principal de notre approche est d'obtenir de bon résultats de détection en tenant compte de l'aspect temps réel. En effet bien que la détection de la chute soit importante, il faut aussi que cette dernière soit très vite détectée pour qu'on puisse prendre les dispositions idoines pour secourir au besoin l'individu.

La prochaine section présente l'approche proposée tandis que la troisième présente les expérimentations et l'analyse des performances. La dernière section du présent chapitre le conclut.

5.2 Algorithme de détection de chutes

La détection de chutes dans une scène observée par deux (02) caméras proposée dans ce manuscrit est déduis d'une observation. En effet la surface au sol de l'individu varie en fonction de sa posture. S'il est debout la surface au sol est petite tandis que s'il est couché la surface au sol est très grande. Cela peut être utilisé comme étant un élément important dans la détection de chutes. Comme le montre la figure 5.1, la surface au sol peut être obtenue en en considérant l'intersection de la projection homographique dans le plan de masse des surfaces des détections des caméras. Notre approche est composée de cinq (05) modules :

— extraction par caméra des pixels de premier plan ;
— fusion des premier plans des caméras ;
— extraction de caractéristiques ;
— suivi de l'individu ;
— prise de décision.

L'extraction des pixels de premier plan est très important. Cela se fait en se basant sur l'approche proposée dans la section 3.4 du chapitre 3. La figure 5.2 présente une illustration

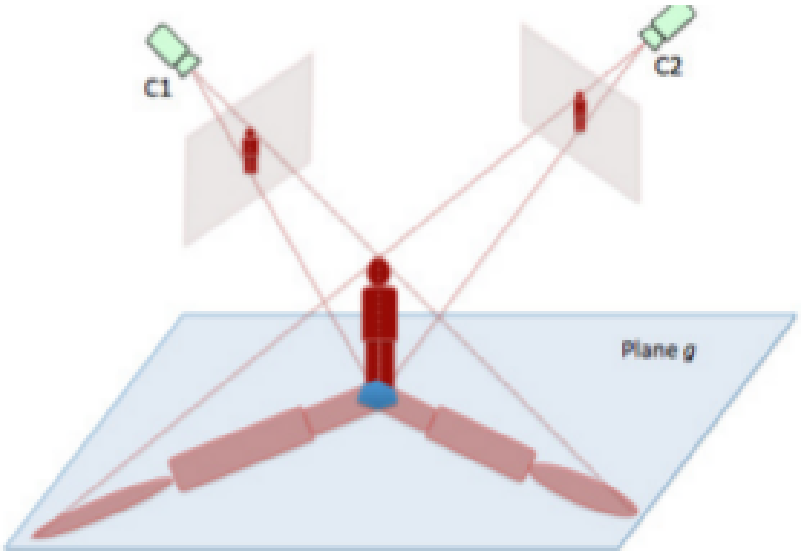

FIGURE 5.1 – Personne détectée par deux caméras avec des vues chevauchantes

de la détection. Une fois les pixels d'avant plan obtenus nous faisons une approximation

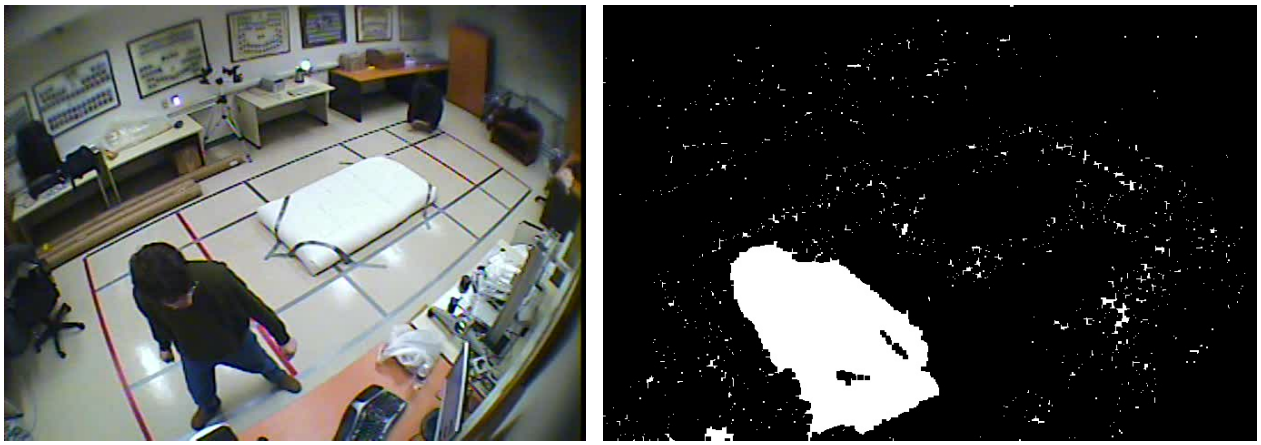

FIGURE 5.2 – Résultat de la détection. La première image représente l'image originale et la seconde présente le résultat de la détection.

de ces pixels par un polygone. Les polygones sont obtenus en recherchant les enveloppes convexes des pixels de premier plan. Cette approximation permet de réduire le nombre des données à manipuler. Cela permet aussi de régler les problèmes de liés aux trous de détection (confère figure 5.3). Nous constatons que sur la figure 5.3 les trous de détection sont fermés. Aussi les sommets du polygone (point en noir sur la seconde image) représente le polygone.

FIGURE 5.3 – polygone formé à partir de la détection de l'image de la figure 3.8.

Une fois les polygones détectés, nous faisons leur projection dans le plan de masse (ou plan de référence). Cela se fait en utilisant le second module de notre système. La projection se fait en utilisant les principes de l'homographie planaire. Il est donc important de procéder à l'étape de calibration. En utilisant l'homographie chaque polygone est projeté dans le plan de masse (ou plan de référence). Les sommets du polygone projeté sont obtenus en projetant les sommets du polygone dans le plan de masse (ou plan de référence). L'homographie est une transformation linéaire entre deux plans projectifs. Cette fusion permet d'obtenir une information beaucoup plus globale. La projection dans le même plan des polygones permet d'avoir une estimation de la surface au sol qui est l'intersection des polygones issus des caméras. Nous avons proposé une stratégie de fusion des polygones beaucoup plus rapides dans Mousse et al. [106]. Les figures 5.4 et 5.5 montrent des illustrations de fusion de polygones. Après cela, il est important de définir des paramètres robustes en vue de caractériser la posture.

Les paramètres pour caractériser la posture, doit tenir compte de la détection de chaque caméra. Mais il est important que la décision soit la plus globale possible. Pour concevoir le système de détection de chutes, il est important de différencier la posture "couché au

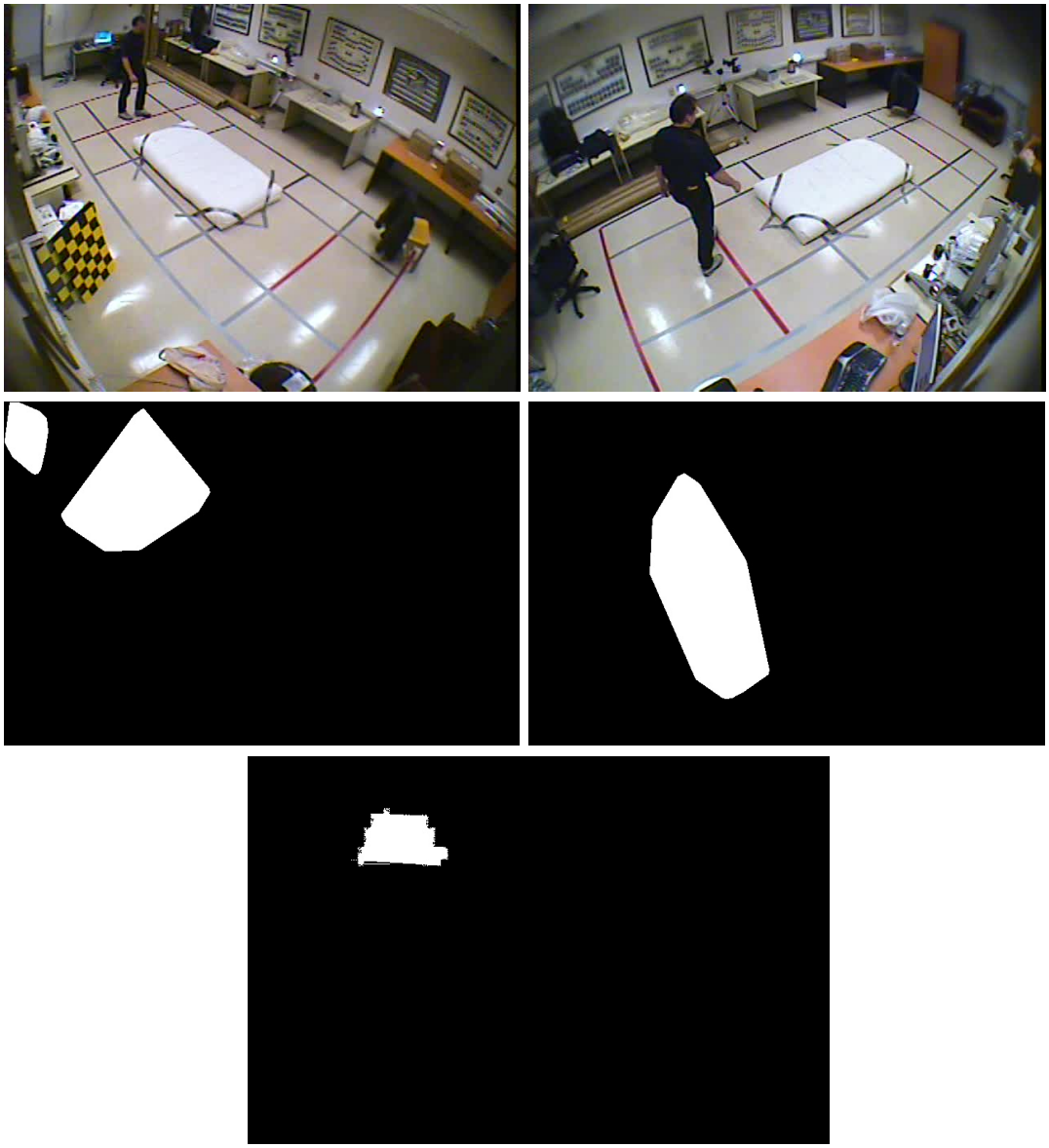

FIGURE 5.4 – La première ligne représente la vue des caméras, la seconde montre les polygones détectés à partir des pixels de premier plan et la dernière ligne représente la surface en contact avec le sol.

sol" des autres ("debout", "assis", "accroupi"). Pour la caractérisation de la posture d'un individu, nous considérons les mesures suivantes :

— la surface du polygone détecté par la première caméra ω_1 ;

— la surface du polygone détecté par la seconde caméra ω_2 ;

— la surface de l'intersection σ.

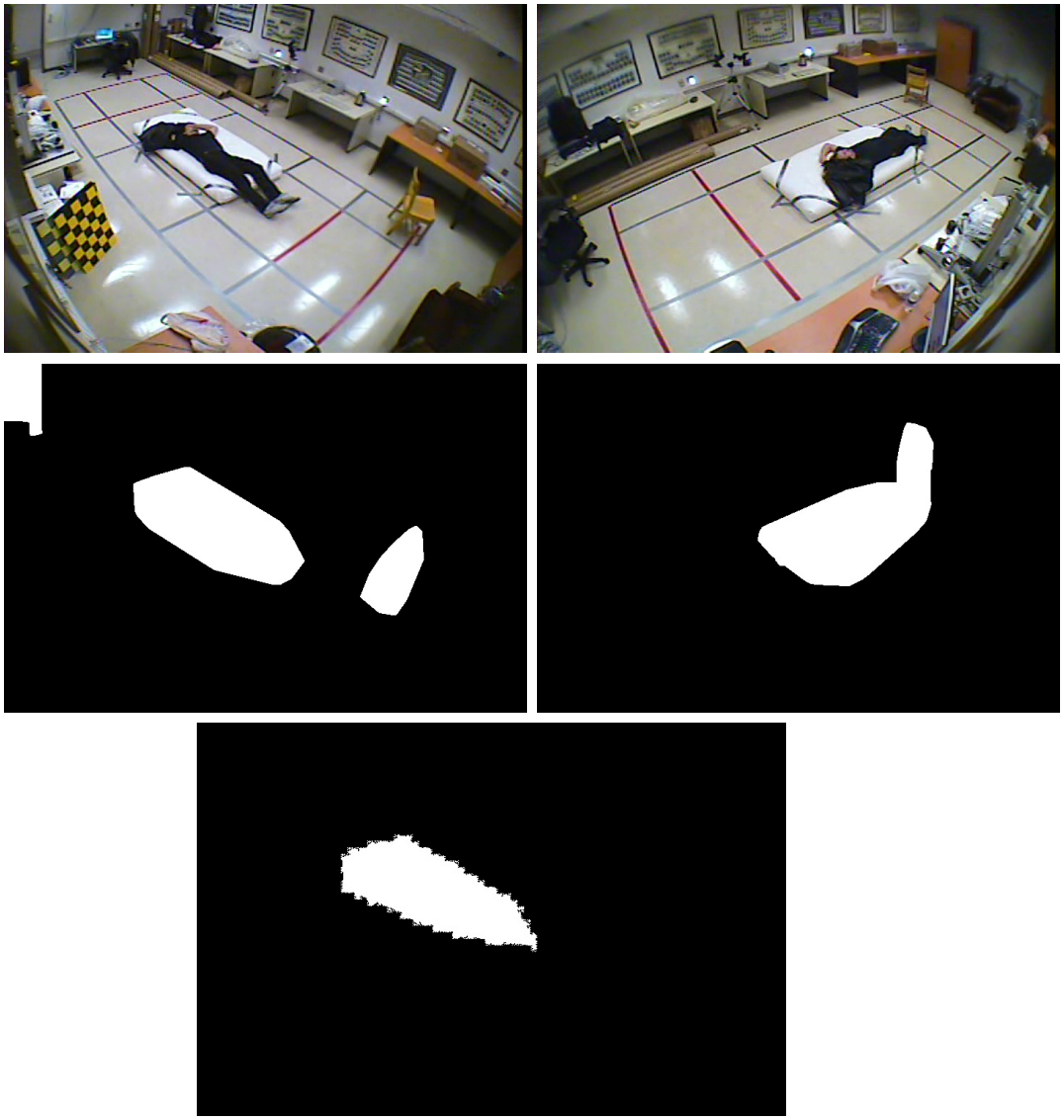

FIGURE 5.5 – La première ligne représente la vue des caméras, la seconde montre les polygones détectés à partir des pixels de premier plan et la dernière ligne représente la surface en contact avec le sol.

A partir de ces mesures, on extrait les caractéristiques ϱ_1 et ϱ_2 obtenues respectivement en utilisant les expressions 5.1 et 5.2.

$$\varrho_1 = \frac{\sigma}{\omega_1} \tag{5.1}$$

$$\varrho_2 = \frac{\sigma}{\omega_2} \tag{5.2}$$

Ces caractéristiques permettent au système d'être robuste face aux occultations. En effet dans une maison il est courant qu'une zone ne soit pas accessible par la camera du fait de la présence des meubles et autres. L'une valeurs entre ω_1 et ω_2 sera biaisée. Mais à l'intérieur d'une maison, il est rare que la personne soit invisible dans les deux (2) cameras. Donc même si l'une des valeurs est biaisée, la seconde ne le sera pas. Ainsi les valeurs ϱ_1 et ϱ_2 sont plus efficaces et plus adéquats car elles sont obtenues après fusion de ω_1 et ω_2. Les figures 5.6, 5.7, 5.8 montrent comment différencier la position "couché au sol" correspondant à une chute éventuelle. Une fois les caractéristiques extraites, il faut

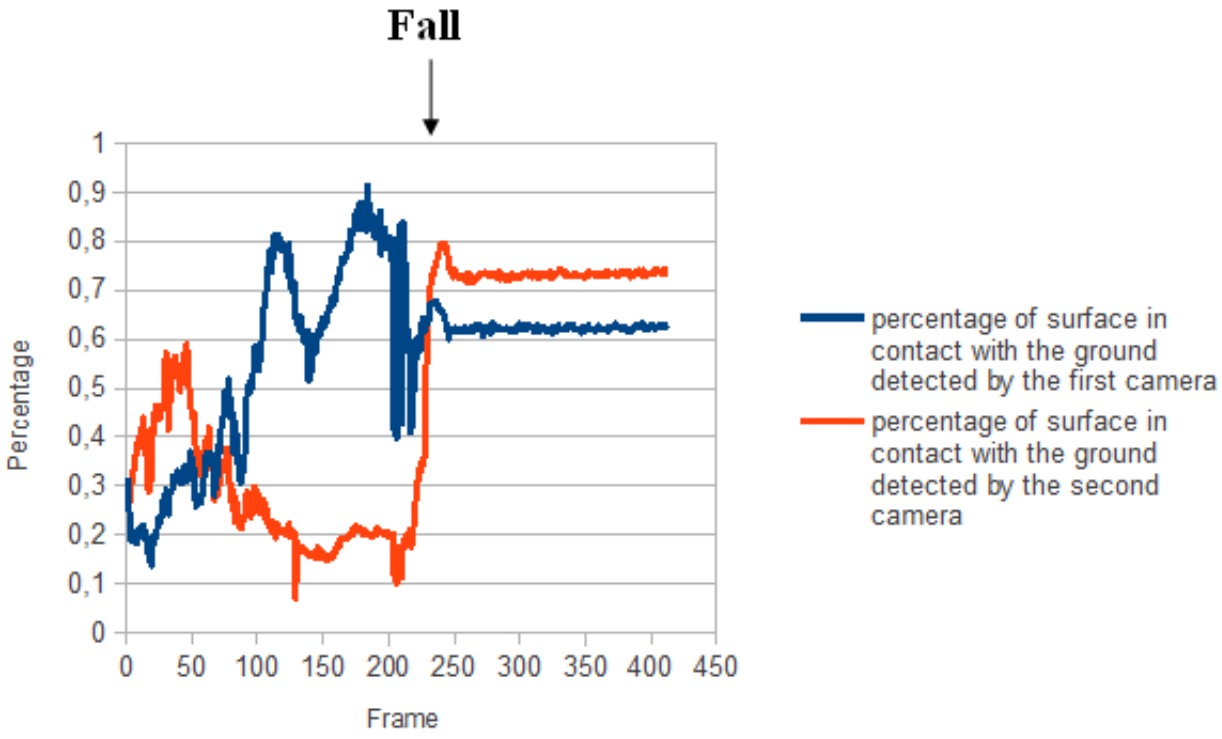

FIGURE 5.6 – Utilisation des caractéristiques pour détecter la chute (scenario 1).

maintenir l'identité de la détection. Cela se fait à travers le module de suivi.

Le suivi permet d'avoir l'information sur la trajectoire de l'individu. Cela permet

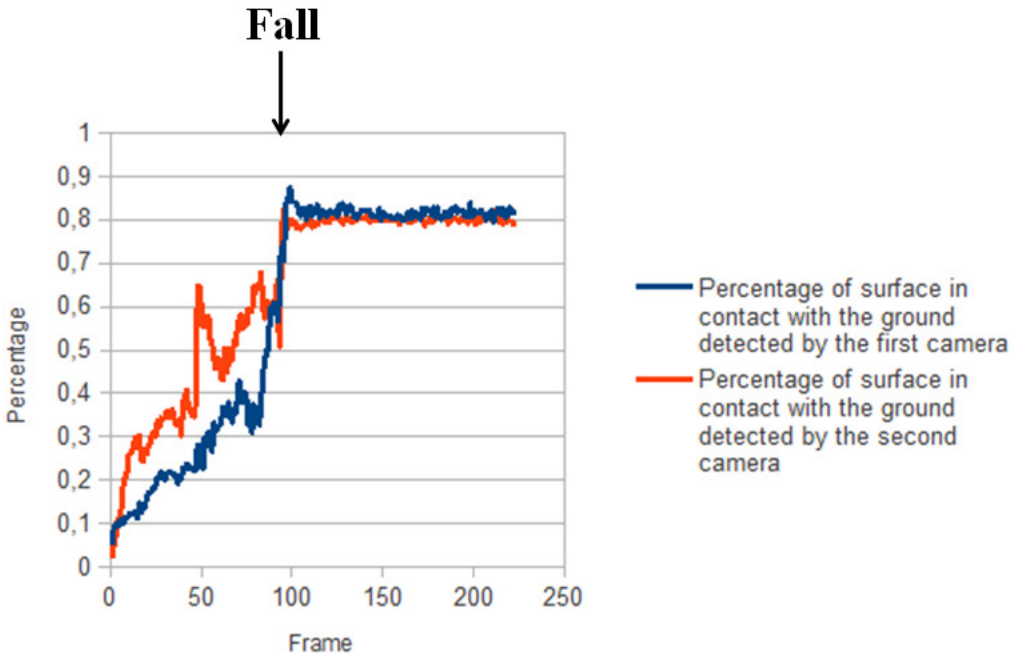

FIGURE 5.7 – Utilisation des caractéristiques pour détecter la chute (scenario 2).

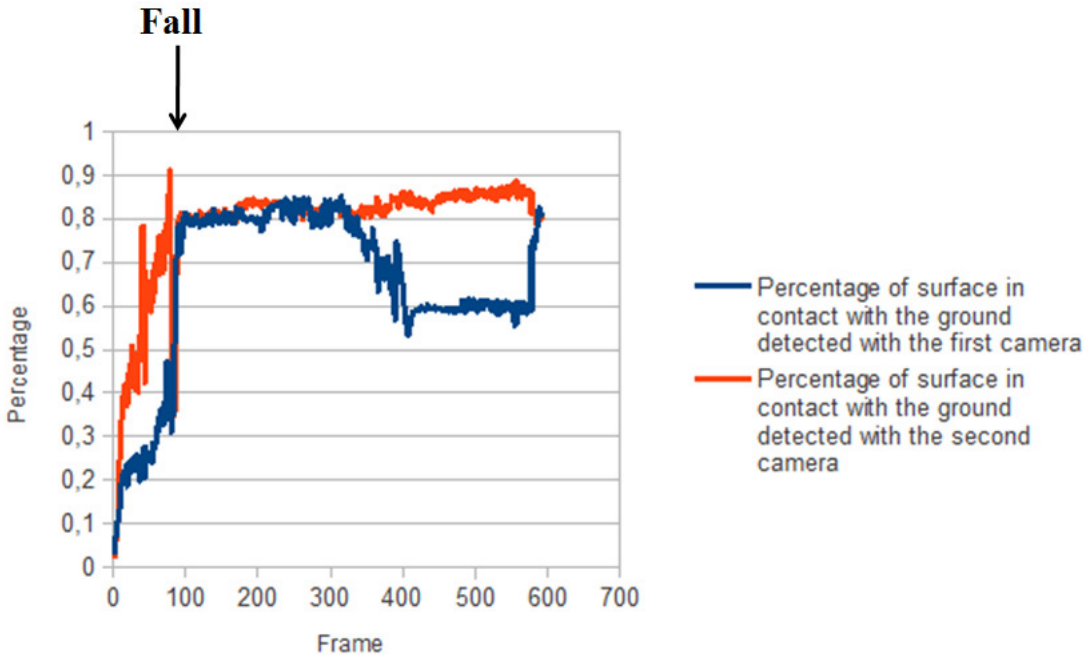

FIGURE 5.8 – Utilisation des caractéristiques pour détecter la chute (scenario 3).

aussi de suivre l'historique des postures de l'individu. La problématique de suivi présente d'importants challenges pour la communauté de vision par ordinateur. Mais au vu de

l'application on n'aura pas beaucoup d'individus à suivre. Un algorithme de suivi simple peut être adapter pour garder l'identité de l'individu dans le temps. Dans notre approche, le suivi est assuré par le centre de fusion. Il est basé sur l'aspect spatial. La méthode utilisent les caractéristiques de l'image pour converger vers les meilleures positions des cibles dans des séquences d'images en fonction du temps. Étant donné les caractéristiques extraites à partir de l'image courante et une fonction d'évaluation de l'état courant, on raffine itérativement l'état estimé à travers la convergence vers le maximum global. Ainsi chaque objet (individu) prends l'identité l'objet (individu) le plus proche de lui dans l'image précédente de la séquence. L'estimation des paramètres du vecteur de déplacement temporelle, se base sur le calcul du flot optique. La zone cible d'intérêt est représentée par une fenêtre carrée de taille N, N étant obtenu en utilisant l'expression 5.3.

$$N = (2w - 1) \times (2w - 1) \tag{5.3}$$

Lorsqu'un nouveau objet apparaît sur la scène, le nombre d'objet est augmenter de un. Et un nouveau identifiant est attribué au nouveau objet. Lorsque l'objet quitte la scène le nombre d'objet présent sur la scène est diminué de un. Et l'identité de l'objet de l'image précédente n'ayant pas de correspondance sur l'image actuelle est supprimée. Une fois les paramètres des individus extraites et que ces individus sont identifiés dans le temps, nous précédons à détection de la chute.

Pour ce faire nous détectons des seuils au delà desquels des décisions de chute seront prises ou non. Ces seuils sont obtenus de manière supervisés. Et leur objectif est de classifier les postures des individus en deux (2) grandes classes. L'apprentissage se fait en utilisant un scénario (scénario 9) de la banque de données présentée dans la sous section 5.3.1. Ce scénario a été choisi car elle regroupe toutes les postures possibles d'un homme et a été justifier dans nos travaux [103, 104]. Les résultats issus de nos expérimentations

nous ont permis d'obtenir les postures en fonction des couples de valeur (ϱ_1, ϱ_2). Ces valeurs sont représentées par la figure 5.9. Sur cette figure, l'axe des abscisses représente

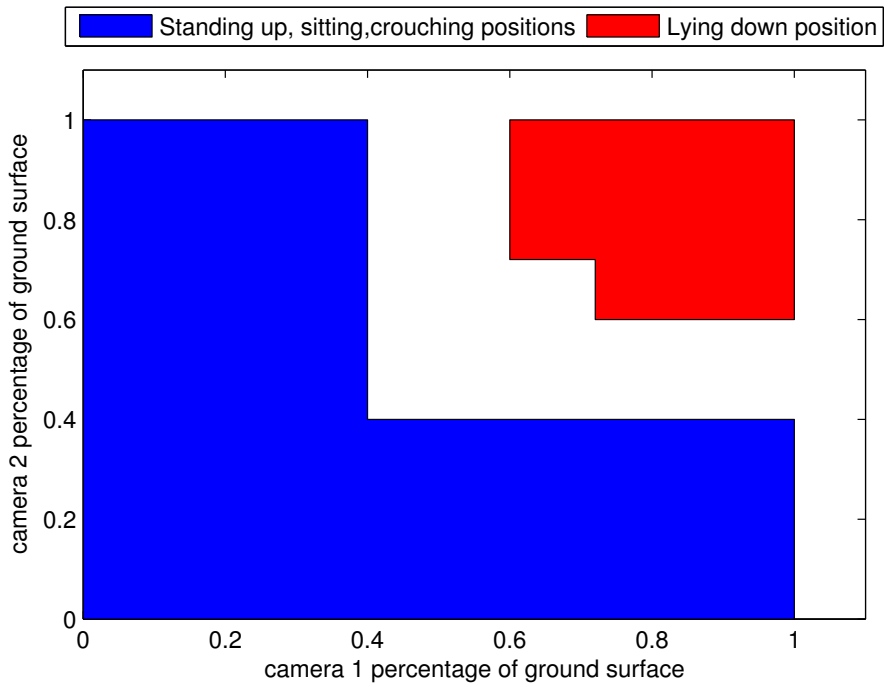

FIGURE 5.9 – Classification des postures.

la valeur ϱ_1 et l'axe des ordonnées représente la valeur ϱ_2. En se basant sur cette figure, on constate aisément que les postures sont clairement divisées en deux. La première classe qui est coloriée en bleu. Cette classe représente les postures : assis, debout et accroupi. La seconde en rouge quand à elle représente la posture "couché au sol" qui peut correspondre éventuellement à une chute. Ainsi lorsque l'une des valeurs entre ϱ_1 et ϱ_2 est inférieure à 0.4 alors la posture est classé dans la première catégorie. L'individu ne présente donc pas aucun problème de chute. Mais lorsque $((\varrho_1 \geq 0.6$ et $\varrho_2 \geq 0.72)$ ou $(\varrho_2 \geq 0.6$ et $\varrho_1 \geq 0.72))$ alors l'individu est dans la position "couché au sol". Dans cette position il faudra vérifier s'il est parti au sol de manière accidentelle ou non. Il est aussi important de s'assurer qu'il est toujours au sol ou non. La personne va être considéree comme ayant chuté si sa posture quitte la première catégorie rapidement pour la seconde. Ainsi nous définissons

un temps $t = 1.5$ secondes. Donc si la personne change de posture en un temps inférieur ou égal à t alors un avertissement est émis. Car la personne a chuté. Après cette chute il peut retrouver ses idées et se relever automatiquement. Une seconde durée $t' = 3$ secondes est alors défini. Lorsque la personne reste au sol plus de 3 secondes alors l'alarme de chute est confirmé et les stratégies pour remédier à la chute sont mises en oeuvre. Ainsi nous détectons les chutes en utilisant la surface au sol.

Une extension de ces travaux ont été menée dans Mousse et al. [105]. Le but de cette extension est de pouvoir étendre le modèle en vue de distinguer les autres postures de la première catégorie. Il est donc important d'estimer la hauteur de l'individu à l'entrée de la scène et de ré-estimer la hauteur lorsque la posture est classée dans la première catégorie. Pour l'estimation de la hauteur nous considérons la figure 5.10. La hauteur est la

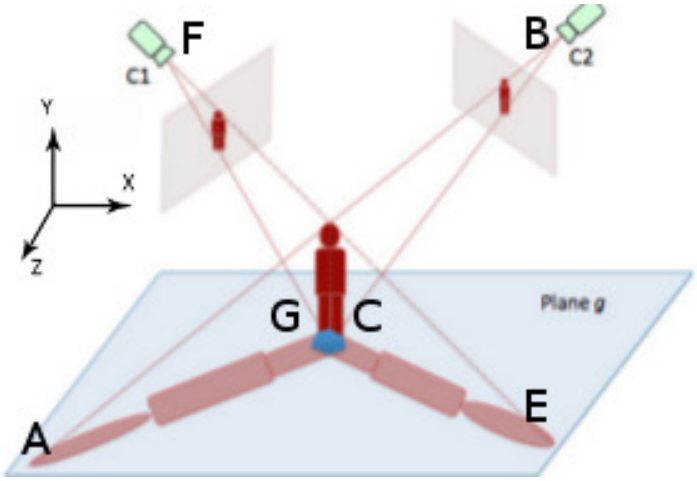

FIGURE 5.10 – Estimation de la hauteur.

longueur de l'intersection des triangles ABC et EFG. L'intersection est obtenue en utilisant l'algorithme proposé par Tropp et al. [107]. Ils ont proposés un algorithme pour faire l'intersection des triangles à partir des coordonnées 3D des sommets. La méthode proposée est rapide et permet d'obtenir le segment de l'intersection. L'obtention de l'intersection se fait en des étapes majeures qui sont décrites comme suit. La première étape consiste à vérifier s'il existe une intersection entre un segment d'un des deux (2) triangles et le plan

du second triangle. Si c'est le cas alors on passe à la seconde étape qui consiste à calculer le point de percée dans le plan. Dans notre cas, ce point se retrouve forcément à l'intérieur du triangle. Il faut alors considérer le second segment qui possède une intersection avec le triangle. Et comme dans le premier cas détecter le point de percée dans le plan. En utilisant les deux points de percée, on obtient le segment qui est le résultat de l'intersection des deux triangles. La hauteur s'obtient donc en calculant la distance euclidienne entre les deux (02) points. Soit h la hauteur initiale et h' la hauteur obtenue lors de la classification dans la première classe. Suivant h et h', nous avons les 3 sous cas ci après :

— Si ($h' = h$ et ($\varrho_1 \leq 0.274$ ou $\varrho_2 \leq 0.274$)) alors la personne est en position debout ;

— Si ($h' < h$ et ($\varrho_1 \leq 0.274$ ou $\varrho_2 \leq 0.274$)) alors la personne est en position penchée ;

— Si $\varrho_1 > 0.274$ et $\varrho_2 > 0.274$ alors la personne est en position assise.

5.3 Expérimentations et analyse des performances

Dans cette section nous présentons le cadre expérimental et faisons une analyse des performances de notre système. La section est subdivisée en deux sous sections. La première sous section présente le cadre expérimental tandis que la seconde présente et analyse les performances.

5.3.1 Expérimentations

Pour valider notre algorithme, nous avons utilisé une banque de données publiques de détection de chutes à partir des séquences multi-caméras. Cette banque de données a été proposée par le Département d'Informatique et de Recherche Opérationnelle de l'université de Montréal. Les séquences sont recueillies en utilisant un système composé de 8 caméras IP (Gadspot gs-4600) à bas coût comme le montre la figure 5.11 et équipée d'un objectif grand angle pour couvrir toute la salle. La disposition des caméras dans la salle de test ainsi que la taille de la salle sont données par la figure 5.12. Cette banque

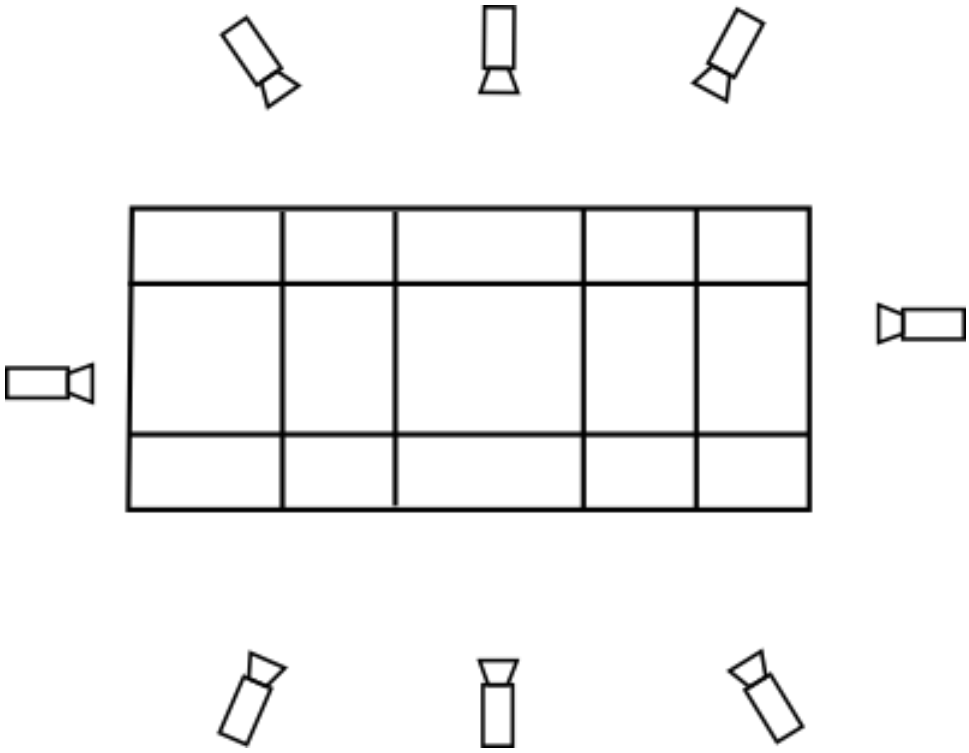

FIGURE 5.11 – Architecture du réseau de caméras.

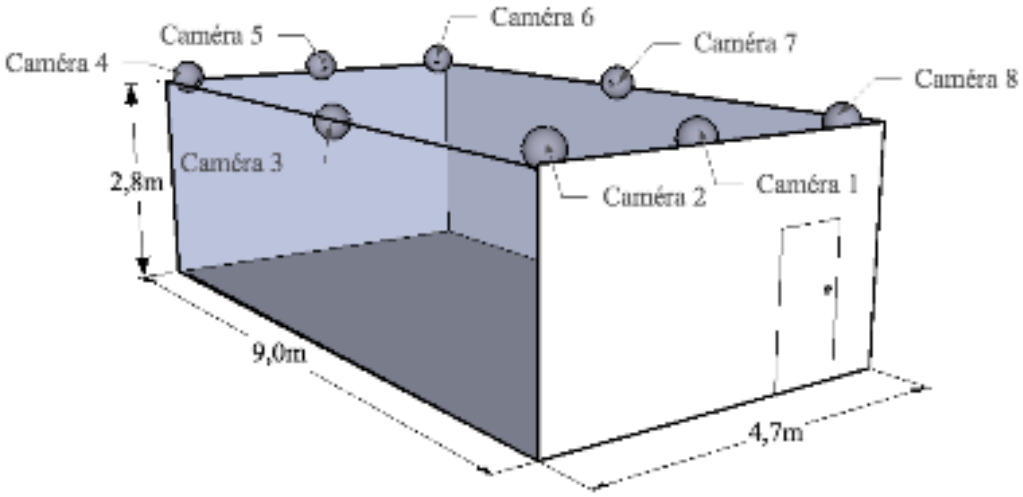

FIGURE 5.12 – Cadre expérimental. [108]

de données contient 24 séquences correspondant à 24 différents scénarios. Ces scénarios contiennent des difficultés qui peuvent engendrer des erreurs de segmentation telles que :

— des artefacts dus à une compression (MPEG4) ;

— des réflexions de lumière ou d'ombres qui peuvent être détectées comme objets en

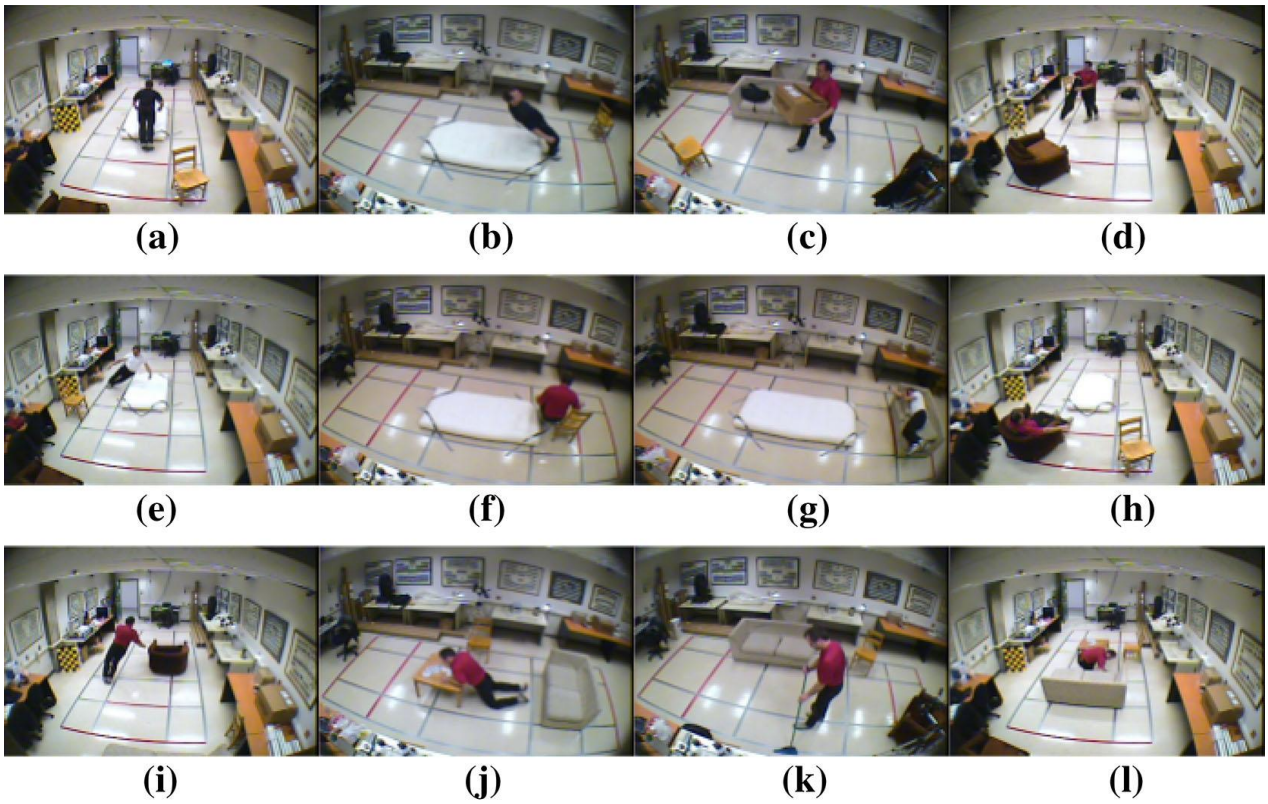

FIGURE 5.13 – Exemples d'images issus de la séquence.

mouvement durant la phase de segmentation ;

— des occlusions, etc.

Les chutes et les activités non chutes ont été simulées par un seul sujet pour chaque séquence (scénario). Des exemples de chutes vers l'avant et vers l'arrière sont données. Ces chutes sont parfois dues à une perte d'équilibre depuis une position debout ou dues à une assise instable. Les activités ordinaires comme marcher, s'asseoir ou se lever complètent cette base de vidéos. Les auteurs ont annoté cette base en repérant le début et la fin de chaque action de chacune de ses vidéos. La figure 5.13 présente des images issus de la séquence. La description de quelques scénarios est présentée par la figure 5.14.

Plus de détails sont donnés dans le rapport technique [108] qui accompagne la banque de données.

Toutes les expériences ont été conduites en utilisant une machine ayant un processeur

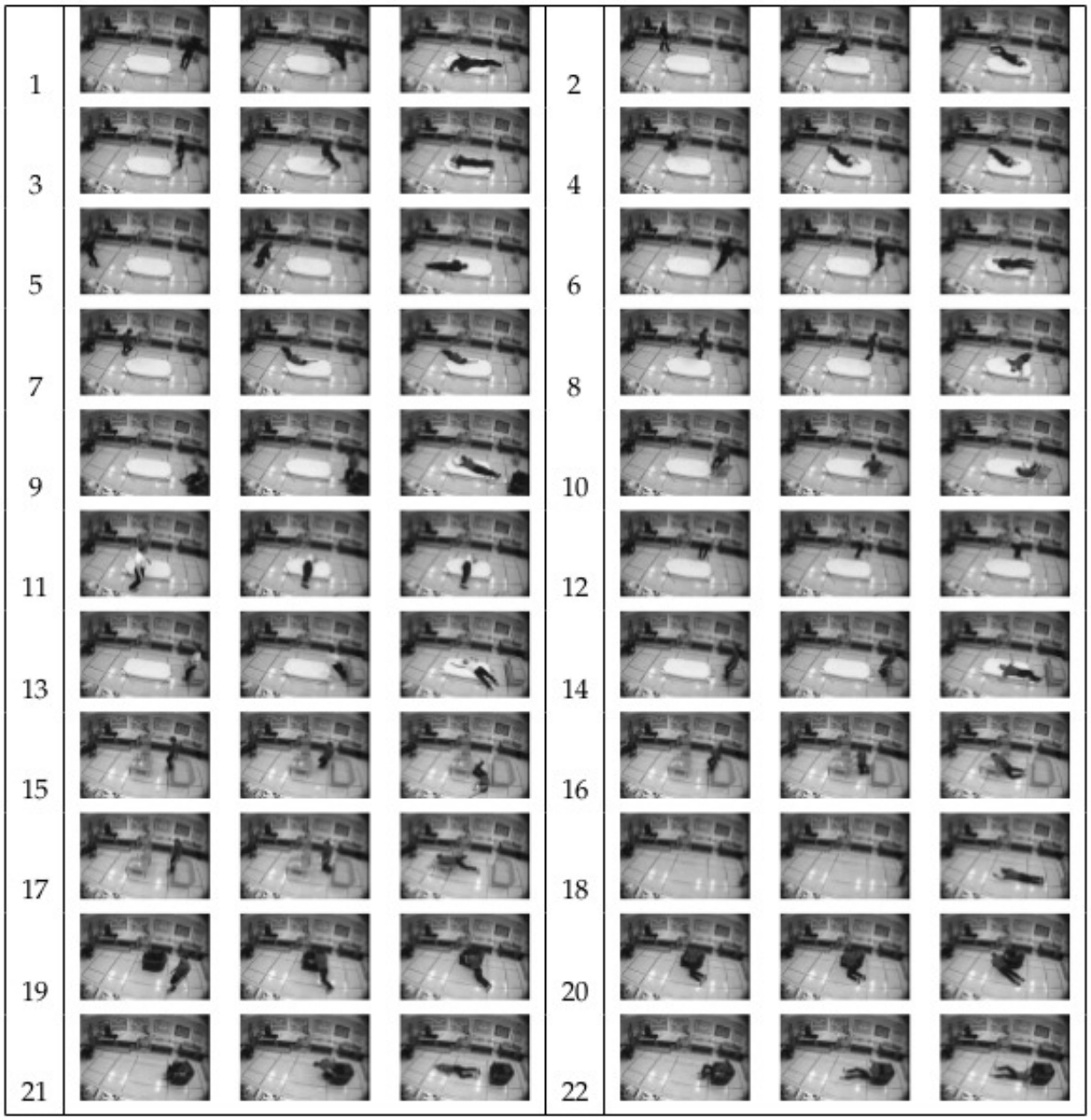

FIGURE 5.14 – Description de quelques scénarios de la séquence. [94]

Intel-Core7@2.13Ghz avec une mémoire RAM de 4GB. Les algorithmes ont été implémentés en utilisant le langage de programmation C++ avec la bibliothèque OpenCv. Notre algorithme n'a besoin que de deux caméras relativement complémentaire pour détecter les chutes. Pour ce faire nos expérimentations ont été faites avec les couples de cameras ci après :

— caméra 1 et caméra 3 ;
— caméra 2 et caméra 5 ;
— caméra 4 et caméra 7 ;
— caméra 6 et caméra 3 ;
— caméra 5 et caméra 8.

5.3.2 Analyse des performances

Une fois les expérimentations effectuées il est important de caractériser la méthode de reconnaissance à l'aide de différentes grandeurs. Ces dernières permettent de voir l'utilité de l'approche proposée. Les plus couramment utilisées en matière d'évaluation des systèmes de détection de chutes sont : la sensibilité et la spécificité. Elles sont définis en utilisant les variables suivantes :

— vrais positifs (TP) : nombre de chutes correctement détectées ;
— faux négatifs (FN) : nombre de chutes non détectées ;
— faux positifs (FP) : nombre d'images ou de suite d'images détectées comme chutes bien qu'il s'agisse de non-chutes ;
— vrais négatifs (TN) : nombre d'images ou de suites d'images d'activités normales, détectées correctement comme non-chute.

La sensibilité Se est obtenu à partir de l'expression 5.4 tandis que la spécificité est obtenue en utilisant l'expression 5.5.

$$Se = \frac{TP}{TP + FN} \tag{5.4}$$

	Sensitivité Se	Specificité Sp
Notre méthode	95.8%	100%
Auvinet et al. [94]	80.6%	100%
Rougier et al. [79]	95.4%	95.8%
Hung et Saito [95]	95.8%	100%

TABLE 5.1 – Comparaison des performances

$$Sp = \frac{TN}{TN + FP} \tag{5.5}$$

La sensibilité indique le taux d'événements de chute reconnus comme tels par le système proposé tandis que la spécificité montre le taux d'événements non chute bien identifiés. Ces valeurs ainsi calculées nous servent d'éléments de comparaison entre les approches de l'état de l'art [79, 94, 95] et la nôtre. Il convient de souligner que d'après nos expériences, les valeurs de notre système ne varient pas en fonction du couple de caméras choisi. Le système n'est donc pas dépendant des caméras dans la salle, juste qu'il faut veiller à ce que les deux caméras aient des vues complémentaires. Les valeurs que nous avons obtenues sont reportées dans le tableau 5.1.

En analysant ce tableau, nous pouvons conclure que l'approche proposée a des performances comparables à ceux de l'approche de l'état de l'art présentant de meilleurs résultats. Ainsi tous les événements ne correspondant à une chute ont été bien détecter comme tel. C'est ce qui explique le taux de 100% obtenu pour la spécificité. Un seul des situation de chute sur vingt quatre n'a pas été reconnu par notre système. Ce qui conduit à l'obtention de la valeur 95.8%.

Le deuxième aspect de comparaison est le temps d'exécution. En effet comme souligné plus haut, il est important qu'un système de détection de chute minimise les temps de calcul. Nous avons donc comparé les deux approches (notre approche et celui de Hung et

	Hung et Saito [95]	Méthode proposée
Vitesse (fps)	10,95	**15,25**

TABLE 5.2 – Comparaison des vitesses de traitement

Saito [95]) les plus performants de notre tableau comparatif. La comparaison des vitesses de traitement est reportée dans le tableau 5.2. De cette comparaison, il ressort que notre approche est plus rapide en matière de temps de traitement que celle proposée par Hung et Saito [95]. Cela est du principalement à deux causes. Premièrement l'utilisation des superpixels dans le module d'extraction de pixels de premier plan qui rends cette partie beaucoup plus rapide. Aussi l'approximation par les polygones réduit considérablement le nombre d'informations manipulées.

La faiblesse de notre système est qu'il est sensible à l'objet présent dans la scène. En effet si un non humain passe, le système va le considérer et faire les traitements pour voir s'il a chuté ou pas. Ce qui peut conduire à l'émission des fausses alarmes.

5.4 Conclusion

Dans ce chapitre nous avons présenté notre contribution dans la conception d'un système de détection de chutes. Nous proposons de faire une estimation robuste de la surface à partir des détections pour la détection de chutes. Cette surface est obtenue en utilisant la projection homographique des polygones représentant les détections de chaque caméra du réseau. Des seuils ont été obtenus à partir d'un modèle d'apprentissage et permettent de reconnaître les situations de chutes de personnes. Nous avons testés notre approche sur une banque de données publiques qui a été utilisé dans beaucoup de travaux de recherches pour la validation des algorithmes de détection de chutes à partir des séquences multi-caméras. Ces expériences ont démontré la qualité du système proposé et la comparaison de notre approche avec ceux de l'état de l'art montre qu'il donne le meilleur résultat en

tenant compte de la performance et du temps de calcul.

Chapitre 6

Conclusion et perspectives

Sommaire

6.1 Conclusion et contributions

Les applications de vidéosurveillance intelligente ont pris beaucoup d'ampleurs. L'utilisation de ces systèmes dans la gestion des activités humaines. Les travaux initiés dans cette thèse s'inscrivent dans le cadre de la mise en oeuvre d'un système de de détection automatique de chutes de personnes en utilisant un système multi-caméras. Ces travaux sont subdivisés en deux (02) grandes parties. La première partie est consacrée à la détection des objets mobiles et la seconde présente la reconnaissance de chutes de personnes. Dans chaque partie, nos recherches nous ont permis de dresser un état de l'art des approches et solutions proposées. Une fois présenté, nous avons analysé ces approches pour proposer des contributions. Ces contributions ont pour but d'améliorer les aspects de l'état de l'art.

Dans la première partie, deux contributions ont été proposées dans le but d'améliorer la détection. La première contribution consiste à la proposition d'un algorithme de détection de mouvements basé sur la combinaison de l'algorithme de "Codebook" [9] et d'un algorithme de détection de contour. Cette combinaison consiste à faire valider les pixels de premier plan obtenu en utilisant l'algorithme de Codebook par les résultats de la détection de contour. Un seuil a été défini pour permettre la fusion efficace des informations. Ce seuil est dépendant de la séquence. Le choix de l'algorithme de détection de contour dépend de l'application visée. La seconde contribution exploite les caractéristiques de l'espace de couleur CIE L*a*b* ainsi que les avantages de la segmentation en superpixels. En utilisant l'espace de couleur CIE L*a*b* les fausses détections sont limitées et la segmentation en superpixels permets la réduction de la complexité algorithmique. Ainsi toutes les images des séquences sont segmentées en utilisant un algorithme de superpixels. Cette segmentation permet de mettre ensemble les pixels homogènes. Après le regroupement, les pixels des clusters sont convertis dans l'espace de couleur CIE L*a*b* et nous utilisons un algorithme modifié de "Codebook" pour la modélisation de l'arrière plan. Les pixels de premier plan sont déduis à partir de l'arrière plan généré. Cette stratégie améliore le temps d'exécution du module de détection d'objets mobiles tout en gardant des taux de détection acceptables.

Dans la seconde partie, nous avons proposé un système de détection de chutes de personnes via un système de caméras. Le but de notre approche est de proposer un système ayant des performances acceptables en un temps record. Le système est composé de deux caméras ayant des vues relativement orthogonales. Cette condition permet d'utiliser le moins de caméra possible pour pouvoir tirer les informations les plus pertinentes de la scène. L'information extraite est l'estimation de la surface au sol. La surface au sol est obtenu en se basant sur la projection homographique dans le plan de masses des détection de chaque caméra. Le système proposé est composé de cinq (05) modules. Ces cinq (05)

modules mis en ensemble permettent une détection automatique des chutes de personnes. Le système proposé premièrement est un système qui fait la classification des postures en deux classes : Les postures de “chute” et les postures de “non chute”. Dans un second temps, nous avons proposé une extension des travaux de bases. Dans cette extension nous avons discriminé les événements de la catégorie “non chute”. Ainsi dans cette approche non avons ajouter d’autres conditions qui permettent de distinguer les postures “assis”, “debout” et “accroupi”. Nos expériences ont prouvé que le système proposé réponds efficacement au besoin de la détection de chute.

6.2 Travaux et perspectives de recherches

Les résultats de cette thèse ouvrent la voie à d’autres perspectives de recherches. Ces perspectives peuvent être classifier en deux parties. Ces parties sont présentées par les deux sous-sections de cette section. La première sous-section présente les perspectives dans le cadre de la détection de mouvement et la seconde présente les perspectives en matière de détection de chutes.

6.2.1 Détection de mouvement

La problématique de réduction des temps de traitements est un challenge toujours d’actualité dans les modules de détection de mouvement. Il est donc nécessaire de penser à améliorer les gains en temps de calcul. Pour ce faire, nous pensons qu’il faille définir une stratégie plus efficientes de gestion des superpixels. En effet nous pensons que la segmentation en superpixels peut être automatiser en se basant sur les aspects visuels de la scène. Ainsi une stratégie de multi-résolution peut être adopter pour cette automatisation. Cette stratégie permettra d’agrandir la taille des superpixels si la taille des objets est très grand et de réduire la taille des superpixels dans le cas contraire. Cela permettra de réduire les traitements inutiles et donc de gagner en temps d’exécution. En effet si la taille des

objets est grand, les petits superpixels vont constituer des traitements supplémentaires pour le module de détection.

6.2.2 Détection de chutes

Le module de détection de chutes proposé la reconnaissance de l'événement pour une personne. Le premier problème est de permettre au module de distinguer les personnes des autres objets. Ceci permettra de ne pas déclencher des alarmes dans le cas d'apparition d'objet non humain. Une autre évolution majeure de système sera d'étendre le modèle pour détecter les chutes multiples. En effet, la détection multi chutes implique le traitement simultané d'au moins deux personnes présentes sur la scène. Cela requiert une stratégie de suivi d'objet plus complexe car les occlusions entre objets en mouvement peuvent intervenir peuvent intervenir. Il est donc indispensable de penser à gérer les cas de groupage de la détection. En effet lorsque les personnes seront proches les uns des autres le module de détection va les mettre ensemble. Il est donc important de pouvoir reconnaître chaque individu du groupe pour ne pas perdre les informations concernant l'individu.

Annexe A

Liste des publications

A.1 Revues internationales avec comité de lecture

1. M. A. Mousse, C. Motamed, E. C. Ezin. *People counting via multiple views using a fast information fusion approach.* Multimedia Tools and Applications, 2016. DOI : 10.1007/s11042-016-3352-z.
2. M. A. Mousse, C. Motamed, E. C. Ezin. *Percentage of human occupied areas for fall detection from two views.* The Visual Computer, 2016. DOI : 10.1007/s00371-016-1296-y.
3. M. A. Mousse, C. Motamed, E. C. Ezin. *Enhanced codebook algorithm for fast moving object detection from dynamic background using scene visual perception.* Journal of Electronic Imaging 25 (6), 2016. DOI : 10.1117/1.JEI.25.6.061618.

A.2 Conférences internationales avec comité de lecture

1. M. A. Mousse, E. C. Ezin, C. Motamed. *Foreground-Background Segmentation Based on Codebook and Edge Detector.* IEEE 10th International Conference on

Signal-Image Technology and Internet-Based Systems, pp. 119-124, 2014. DOI : 10.1109/SITIS.2014.55.

2. M. A. Mousse, C. Motamed, E. C. Ezin. *Fast Moving Object Detection from Overlapping Cameras.* 12th International Conference on Informatics in Control, Automation and Robotics, pp. 296-303, 2015. DOI : 10.5220/0005541402960303.
3. M. A. Mousse, C. Motamed, E. C. Ezin. *Video-Based People Fall Detection via Homography Mapping of Foreground Polygons from Overlapping Cameras.* IEEE 11th International Conference on Signal-Image Technology and Internet-Based Systems, pp. 164-169, 2015. DOI : 10.1109/SITIS.2015.56.
4. M. A. Mousse, C. Motamed, E. C. Ezin. *A Multi-View Human Bounding Volume Estimation for Posture Recognition in Elderly Monitoring System.* International Conference on Pattern Recognition Systems, 2016. DOI : 10.1049/ic.2016.0026.
5. M. A. Mousse, C. Motamed, E. C. Ezin. *Fast polygons fusion for multi-views moving object detection from overlapping camera.* 13th African Conference on Research in Computer Science and Applied Mathematics, pages 262-268.
6. M. A. Mousse, C. Motamed, E. C. Ezin. *An Adaptive Algorithm for Fast Moving Object Detection from Dynamic Background based on Cobebook.* International Conference on Robotics and Mechatronics, 2016.

Annexe B

Homographie planaire

L'homographie est une transformation linéaire entre deux plans projectifs. Dans le cas où la scène observée est planaire, la relation qui existe entre deux caméras est définie par une homographie H (ou transformation projective) 2D qui représente une transformation linéaire inversible de P^2 dans P^2 qui conserve l'alignement. Le théorème suivant permet de caractériser de façon matricielle les homographies 2D :

Théorème 1 : Une fonction $H : P^2 \rightarrow P^2$ est une homographie si et seulement si il existe une matrice H de taille 3×3 telle que pour tout point q de P^2, $H(q) = Hq$.

La matrice H est homogène et est définie à un facteur près et possède donc 8 degrés de liberté. Un des cas courants d'utilisation des homographies 2D est celui décrit dans la figure B.1. Nous nous plaçons ici dans le cas de deux caméras observant un plan Π de l'espace. La projection centrale du plan de l'espace au plan image (et réciproquement) définit une homographie 2D (les coordonnées des points 2D étant exprimées dans le repère 2D relatif à chacun des plans). Un résultat intéressant est alors que, pour tout point 3D appartenant au plan Π, la fonction passant des coordonnées de son observation q_1 dans l'image 1 aux coordonnées de son observation q_2 dans l'image 2 est également une homographie. En effet, la composition de 2 homographies est une homographie. Le lien

entre les observations peut donc s'écrire :

$$q_2 = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} q_1$$

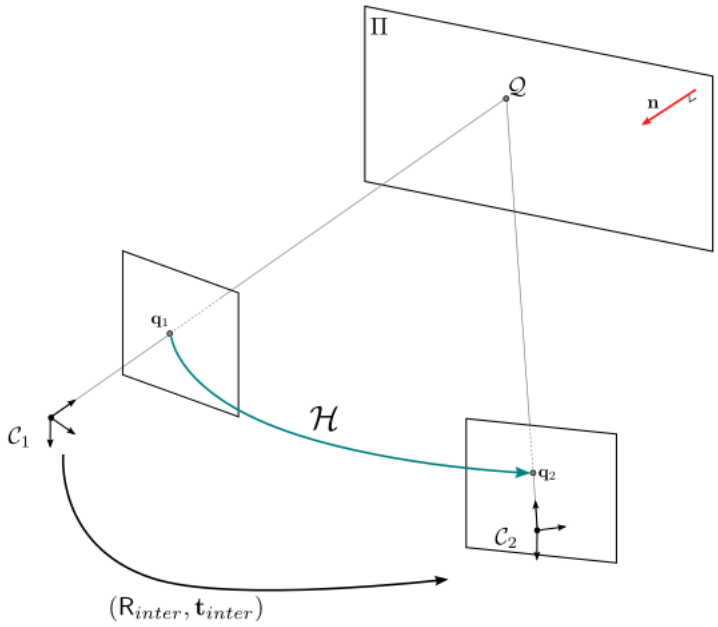

FIGURE B.1 – Homographies 2D. Les coordonnées des observations correspondantes de points 3D situés sur un même plan de l'espace sont reliées par une homographie 2D.

Il est donc important de définir une manière d'obtention de la matrice.

Lorsque la structure de l'environnement est inconnue, il est tout de même possible de calculer le déplacement relatif entre deux caméras. Cela nécessite d'associer les observations des 2 caméras qui correspondent aux mêmes points 3D de l'environnement. Ceci permet de calculer la matrice fondamentale (algorithme des 8 points Hartley [109]) ou essentielle (algorithme des 5 points, Nister [110]). Dans le cas de caméras non-calibrées,les transformations sont calculées à partir de la matrice fondamentale (Hartley and Zisserman [111]). Dans ce cas, le déplacement inter-caméra ne peut être retrouvé qu'à une transforma-

tion projective près. Lorsque le calibrage des caméras est connu, il est préférable d'utiliser la matrice essentielle.

Bibliographie

[1] P. A. MacCulloch, T. Gardner, and A. Bonner. *Comprehensive fall prevention programs across settings : a review of the literature.* Geriatric Nursing, 28(5) pp. 306–311, 2007.

[2] J. R. Agustina, and G. G. Clavell. *The impact of CCTV on fundamental rights and crime prevention strategies : The case of the Catalan Control Commission of Video surveillance Devices.* Computer law & security review, Vol 27, pp. 168–74, 2011

[3] Y. W. Bai, Z. L. Xie, and Z. H. Li. *Design and Implementation of a Home Embedded Surveillance System with Ultra-Low Alert Power.* IEEE Transactions on Consumer Electronics. 75, pp. 153-159, 2011.

[4] M. J. H. Loomans, C. J.Koeleman, and P. H. N. de With. *Low-Complexity Wavelet-Based Scalable Image & Video Coding for Home-Use Surveillance.* IEEE Transactions on Consumer Electronics, pp. 507-15, 2011.

[5] N. Zouba, F. Bremond, and M. Thonnat. *An Activity Monitoring System for Real Elderly at Home : Validation Study.* In the 7th IEEE International Conference on Advanced Video and Signal-Based Surveillance, AVSS, Boston, pp.1-8, 2010.

[6] P. Robert, E. Castelli, P. C. Chung, T. Chiroux, C. F. Crispim-Junior, P. Mallea, and F. Bremond. *SWEET HOME ICT technologies for the assessment of elderly*

subjects. IRBM BioMedical Engineering and Research, Ref. No. : IRBM-D-13-00003, 2013.

[7] M. A. Hossain, and D. T. Ahmed. *caregiver : an ambient-aware elderly monitoring system*. IEEE Transactions on Information Technology in Biomedicine, 16(6) pp. 1024–1031, 2012.

[8] M. Boulmier. *Bien vieillir à domicile : Enjeux d'habitat, enjeux de territoires*. Rapport remis à Monsieur Benoist APPARU Secrétaire d'Etat au Logement et à l'Urbanisme.

[9] K. Kim, T. H. Chalidabhonse, D. Harwood, and L. Davis. *Real-time foreground-background segmentation using codebook model*. Real-Time Imaging, 11(3) pp. 172–185, 2005.

[10] B. Li, Z. Tang, B. Yuan, and Z. Miao. *Segmentation of moving foreground objects using codebook and local binary patterns*. In Proceedings of the 2008 Congress on Image and Signal Processing, pp. 239–243, 2008.

[11] W. Yu, D. Zeng, and H. Li. *Layered video objects detection based on LBP and codebook*. In Proceedings of First International Workshop on Education Technology and Computer Science, pp. 207–213, 2006.

[12] Y. Li, F. Chen, W. Xu, and Y. Du. *Gaussian-based codebook model for video background subtraction*. Lecture Notes in Computer Science, pp. 762–765, 2006.

[13] M. A. Mousse, E. C. Ezin and C. Motamed. *Foreground-Background Segmentation Based on Codebook and Edge Detector*. In Proceedings of IEEE 10th International Conference on Signal-Image Technology and Internet-Based Systems, pp 119–124, 2014.

[14] A. Doshi, and M. M. Trivedi. *Hybrid cone-cylinder codebook model for foreground detection with shadow and highlight suppression*. In Proceedings of IEEE

International Conference on Advanced Video and Signal Based Surveillance, pp. 121–133, 2006.

[15] X. Fang, C. Liu, S. Gong, and Y. Ji. *Object Detection in Dynamic Scenes Based on Codebook with Superpixels.* In Proceedings of Asian Conference on Pattern Recognition, pp. 430–434, 2013.

[16] A. Schick, M. Fischer, and R. Stiefelhagen. *Measuring and evaluating the compactness of superpixels.* In Proceedings of International Conference on Pattern Recognition, pp. 930–934, 2012.

[17] M. A. Mousse, C. Motamed, and E. C. Ezin. *Fast Moving Object Detection from Overlapping Cameras.* In Proceedings of 12th International Conference on Informatics in Control, Automation and Robotics, pp. 296-303, 2015.

[18] G. X. Ritter and J. N. Wilson. *Handbook of Computer Vision Algorithms in Image Algebra.* CRC Press, USA, 2nd edition, 2000.

[19] L. G. Shapiro and G. C. Stockman. *Computer Vision.* Prentice Hall, 2002.

[20] M. Petrou and P. Bosdogianni. *Image Processing : The Fundamental.* John Wiley and Sons, Inc, NewYork, USA, 2nd edition, 2010.

[21] C. Su and A. Amer. *A real-time adaptive thresholding for video change detection.* In International Conference on Image Processing, pp. 157-160, 2006.

[22] M. K. Leung and Y. H. Yang. *Human body motion segmentation in a complex scene.* Pattern Recognition Letter, 20, pp. 55-64, 1987.

[23] N. Verbeke and N. Vincent. *A pca-based technique to detect moving objects.* In SCIA, pp. 641-650, 2007.

[24] G. Bradski and J. Davis. *Motion segmentation and pose recognition with motion history gradients.* Machine Vision and Applications, 13(3) pp. 74-184, 2002.

[25] P. L. Rosin and E. Ioannidis. *Evaluation of global image thresholding for change detection.* Pattern Recognition Letter, 24 pp. 2345-2356, 2003.

[26] P. Rosin. *Unimodal thresholding.* Pattern Recognition Letter, 34(11) pp. 2083-2096, 2001.

[27] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. *Performance of optical flow techniques.* Int. Journal of Computer Vision, 12(1) pp. 43-77, 1994.

[28] H. Liu, T.H. Hong, M. Herman, and R. Chellappa. *Accuracy vs. efficiency trade-offs in optical flow algorithms.* Computer Vision and Image Understanding, 7(3) pp. 271-286, 1998.

[29] H. Mori, N.M. Charkari, , and T. Matsushita. *On-line vehicle and pedestrian detections based on sign pattern.* IEEE Transactions on Industrial Electronics, 41 pp. 384-391, 1994.

[30] A. Broggi, M. Bertozzi, A. Fascioli, and M. Sechi. *Shape based pedestrian detection.* In IEEE Intelligent Vehicles Symposium, pp. 215-220, 2000.

[31] J. Kapur, P. Sahoo, and A. Wong. *A new method for gray level picture thresholding using the entropy of the histogram.* Computer Vision Graphics Image Process, 29(3), pp.273-285, 1985.

[32] J. Parker. *Algorithms for Image Processing and Computer Vision.* John Wiley and Sons, Inc, NewYork, USA, 2nd edition, 1996.

[33] L. Huang and M. Wang. *Image thresholding by minimizing the measures of fuzziness.* Pattern Recognition Letter, 28, pp. 41-51, 1995.

[34] R. Yager. *On the measure of fuzziness and negation.* International Journal on General Systems, 5, pp. 221-229, 1979.

[35] Y.F. Ma and H.J. Zhang. *Detecting motion object by spatio-temporal entropy.* In IEEE Int. Conf. on Multimedia and Expo (ICME 2001), pp. 379-382, 2001.

[36] C. Stauffer, W. Eric, and L. Grimson. *Learning patterns of activity using real time tracking.* Transaction on Pattern Analysis and Machine Intelligence, 22(8), pp. 747-757, 2000.

[37] P. Kaewtrakulpong and R. Bowden. *An improved adaptive background mixture model for real time tracking with shadow detection.* In 2nd European Workshop on Advanced Video Based Surveillance Systems, 2001.

[38] N. Thome and S. Miguet. *A robust appearance model for tracking human motion.* In Advanced Video and Signal Based Surveillance AVSS, pp. 528-533, 2005.

[39] P. Dickinson, A. Hunter, and K. Appiah. *Segmenting foreground objects from a dynamic textured background via a robust kalman filter.* In IEEE International Conference on Computer Vision, pp. 44-50, 2003.

[40] A. Elgammal, D. Harwood, and L. Davis. *Non-parametric model for background subtraction.* In 6th European Conference on Computer Vision, pp. 751-767, 2000.

[41] T. Horprasert, D. Harwood, and L.S. Davis. *A statistical approach for real-time robust background subtraction and shadow detection.* In IEEE International Conference on Computer Vision, 1999.

[42] Y. Goyat, T. Chateau, L. Malaterre, and L. Trassoudaine. *Vehicle trajectories evaluation by static video sensors.* In 9th IEEE International Conference on Intelligent Transportation Systems, pp. 864-869, 2006.

[43] C. Ridder, O. Munkelt, and H. Kirchner. *Adaptive background estimation and foreground detection using kalman-filtering.* In Proceedings of International Conference on recent Advances in Mechatronics, pp. 193-199, 1995.

[44] J. Zhong and S. Sclaroff. *Segmenting foreground objects from a dynamic textured background via a robust kalman filter.* In IEEE International Conference on Computer Vision (ICCV), pp. 44-50, 2003.

[45] S. Soatto, G. Doretto, and Y.N. Wu. *Dynamic textures.* In Proceedings of International Conference on Computer Vision, pp. 439-446, 2001.

[46] G. Doretto, A. Chiuso, Y.N. Wu, and S. Soatto. *Dynamic textures.* IJCV, 51(2) pp. 91-109, 2003.

[47] B.P.L. Lo and S.A. Velastin. *Automatic congestion detection system for underground platforms.* In International Symposium on Intelligent Multimedia, Video and Speech Processing, pp. 158-161, 2001.

[48] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. *Detecting moving objects, ghosts and shadows in video streams.* PAMI, 25(10) pp. 1337-1342, 2003.

[49] D.N.T. Cong, L. Khoudour, C. Achard, and P. Phothisane. *People re-identification by means of a camera network using a graph-based approach.* Machine Vision and Application, 90 pp. 2362-2374, 2009.

[50] M. Izadi and P. Saeedi. *Robust region-based background subtraction and shadow removing using color and gradient information.* In 19th International Conference on Pattern Recognition, pp. 1-5, 2008.

[51] M. Heikkilä and M. Pietikäinen. *A texture based method for modeling the background and detecting moving objects.* Transaction on Pattern Analysis and Machine Intelligence, 28(4) pp. 657-662, 2006.

[52] G.D. Tian and A.D. Men. *An improved texture-based method for background subtraction using local binary patterns.* In CISP, pp. 1-4, 2009.

[53] L. Li, W. Huang, I.Y.H. Gu, , and Q. Tian. *Statistical modeling of complex backgrounds for foreground object detection.* IEEE Transaction on Image Processing, 13(11), pp. 1459-1472, 2004.

[54] Q. Cai and J. Aggarwal. *Automatic Tracking of Human Motion in Indoor Scenes Across Multiple Synchronized Video Streams.* In Proc. of IEEE International Conference on Computer Vision, 1998, pp. 356-362.

[55] S. Khan and M. Shah. *Consistent Labeling of Tracked Objects in Multiple*

Cameras with Overlapping Fields of View. IEEE Trans. on Pattern Analysis and Machine Intelligence, volume : 25, issue : 10, 2003, pp. 1355-1360.

[56] J. Kang, I. Cohen and G. Medioni. *Continuous Tracking within and across Camera Streams.* Proc. of International Conference on Computer Vision Pattern Recognition, volume : 1, 2003, pp. 267-272.

[57] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou and S. Maybank. *Principal Axis-Based Correspondence between Multiple Cameras for People Tracking.* IEEE Trans. on Pattern Analysis and Machine Intelligence, volume : 28, issue : 4, 2006, pp. 663-671.

[58] M. Xu, J. Orwell, L. Lowey and D. Thirde. *Architecture and Algorithms for Tracking Football Players with Multiple Cameras*". IEE Proc. of Vision, Image and Signal Processing, volume : 152, issue : 2, 2005 pp. 232-241.

[59] N. Krahnstoever, T. Yu, K. Patwardhan and D. Gao. *Multi-camera person tracking in crowded environments.* Proc. of 12th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, 2009, pp. 1-7.

[60] R. Eshel and Y. Moses. *Homography based multiple camera detection and tracking of people in a dense crowd.* Proc. of 18th IEEE International Conference on Computer Vision and Pattern Recognition, 2008, pp. 1-8.

[61] M. Xu, J. Ren, D. Chen, J. Smith and G. Wang. *Real-time detection via homography mapping of foreground polygons from multiple.* Proc. of 18th IEEE International Conference on Image Processing, 2011, pp. 3593-3596.

[62] S. M. Khan and M. Shah. *A multi-view approach to tracking people in crowded scenes using a planar homography constraint.* Proc. of 9th European Conference on Computer Vision, 2006, pp. 133-146.

[63] D. B. Yang, H. H. Gonzalez-Banos and L. J. Guibas. *Counting People in*

Crowds with a Real-Time Network of Simple Image Sensors. Proc. 9th IEEE International Conference on Computer Vision, volume : 1, 2003, pp. 122-129.

[64] S. M. Khan and M. Shah. *Tracking multiple occluding people by localizing on multiple scene planes.* IEEE Transactions on Pattern Analysis and Machine Intelligence, volume : 31, issue : 3, 2009, pp. 505-519.

[65] N. Goyette, P. M. Jodoin, F. Porikli, J. Konrad and P. Ishwar. *changedetection.net : A new change detection benchmark dataset.* In Proc. IEEE Workshop on Change Detection (CDW12), 2012. Providence, RI.

[66] X. Cheng, T. Zheng and L. Renfa. *A fast motion detection method based on improved codebook model.* Journal of Computer and Development 47, 2010, pp. 2149-2156.

[67] J. A. Canny. *Computational Approach To Edge Detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, pp. 679-714.

[68] M. Mubashir, L. Shao and L. Seed. *A survey on fall detection : principles and approaches.* Neurocomputing , 100, 2013, pp. 144-152.

[69] D. Anderson, J. M. Keller, M. Skubic, X. Chen and Z. H. He. *Recognizing falls from silhouettes.* In IEEE International Conference on Engineering in Medicine and Biology Society , 2006, pp. 6388-6391.

[70] C. L. Liu, C. H. Lee and P. M. Lin. *A fall detection system using k-nearest neightbor classifier.* Expert Systems with Applications , 37, 2010, pp. 7174-7181.

[71] T. Lee and A. Mihailidis. *An intelligent emergency response system : preliminary development and testing of automated fall detection.* Journal of Telemedicine and Telecare , 11(4), 2005, pp. 194-198.

[72] H. N. Charif and S. J. McKenna. *Activity summarization and fall detection in a supportive home environment.* In International Conference on Pattern Recognition, 2004, pp. 323-326.

[73] M. Shoaib, R. Dragon and J. Ostermann. *View-invariant fall detection for elderly in real home environment.* In Pacific-Rim Symposium on Image and Video Technology, 2010, pp. 52-57.

[74] M. Shoaib, R. Dragon and J. Ostermann. *Context-aware visual analysis of elderly activity in a cluttered home environment.* EURASIP Journal on Advances in Signal Processing, 2011, pp. 1-14.

[75] C. Rougier, J. Meunier, A. St-Arnaud and J. Rousseau. *Fall detection from human shape and motion history using video surveillance.* In International Workshop on Advanced Information Networking and Applications, 2007, pp. 875-880.

[76] Y. T. Liao, C. L. Huang and S. C. Hsu. *Slip and fall event detection using bayesian belief network.* Pattern Recognition, 45, 2012, pp. 24-32.

[77] Y. T. Chen, Y. C. Lin and W. H. Fang. *A hybrid human fall detection scheme.* In International Conference on Image Processing, 2010, pp. 3485-3488.

[78] C. Rougier, J. Meunier, A. St-Arnaud and J. Rousseau. *Procrustes shape analysis for fall detection.* In International Workshop on Visual Surveillance, 2007.

[79] C. Rougier, J. Meunier, A. St-Arnaud and J. Rousseau. *Robust video surveillance for fall detection based on human shape deformation.* IEEE Transactions on Circuits and Systems for Video Technology, 21(5), 2011, pp. 611-622.

[80] Z. Z. Htike, S. Egerton and K. Y. Chow. *A monocular view-invariant fall detection system for the elderly in assisted home environments.* In Conference on Intelligent Environments, 2011, pp. 40-46.

[81] Z. A. Khan and W. Sohn. *Abnormal human activity recognition system based on r-transform and kernel discriminant technique for elderly home care.* IEEE Transactions on Consumer Electronics, 2011, 57(4), pp. 1843-1850.

[82] M. Yu, A. Rhuma, S. M. Naqvi, L. Wang and J. Chambers. *A posture*

recognition-based fall detection system for monitoring an elderly person in a smart home environment. IEEE Transactions on Information Technology in Biomedicine,16(6), 2012, pp. 1274-1286.

[83] C. Rougier, J. Meunier, A. St-Arnaud and J. Rousseau. *Monocular 3d head tracking to detect falls of elderly people.* In IEEE International Conference on Engineering in Medicine and Biology Society, 2006, pp. 6384-6387.

[84] R. Cucchiara, C. Grana, A. Prati and R. Vezzani. *Probabilistic posture classification for human behavior analysis.* IEEE Transactions on Systems, Man, and Cybernestics, Part A : Systems and Humans , 35(1), 2005, pp. 42-54.

[85] R. Cucchiara, A. Prati and R. Vezzani. *A multi-camera vision system for fall detection and alarm generation.* Expert Systems, 24(5), 2007, pp. 334-345.

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

[87] L. Hazelhoff, J. Han and P. H. N. de With. *Video-based fall detection in the home using principal component analysis.* In Advanced Concepts for Intelligent Vision Systems, 2008, pp. 298-309.

[88] D. Anderson, R. H. Luke, J. M. Keller, M. Skubic, M. Rantz, and M. Aud. *Linguistic summarization of video for fall detection using voxel person and fuzzy logic.* Computer Vision and Image Understanding, 113, 2009, pp. 80-89.

[89] S. Zambanini, J. Machajdik, and M. Kampel. *Detecting falls at homes using a network of low-resolution cameras.* In IEEE International Conference on Information Technology and Applications in Biomedicine, 2010.

[90] A. Zweng, S. Zambanini, and M. Kampel. *Introducing a statistical behavior model into camera-based fall detection.* In International Symposium on Visual Computing, 2010, pp. 163-172.

[91] M. Yu, A. Rhuma, S. M. Naqvi and J. Chambers. *Fall detection for the elderly in a smart room by using an enhanced one class support vector machine.* In International Conference on Digital Signal Processing, 2011, pp. 1-6.

[92] E. Auvinet, F. Multon, A. St-Arnaud, J. Rousseau, and J. Meunier. *Fall detection using multiple cameras.* In IEEE International Conference on Engineering in Medicine and Biology Society, 2008, pp. 2554-2557

[93] O. Javed and M. Shah. *Practice . Automated Multicamera Surveillace : Algorithm and pratice.* Springer, 2008, 1st edition.

[94] E. Auvinet, F. Multon, A. St-Arnaud, J. Rousseau, and J. Meunier. *Fall detection with multiple cameras : An occlusion-resistant method based on 3-d silhouette vertical distribution.* IEEE Transactions on Information Technology in Biomedicine, 2011, 15 :290-300.

[95] D. Hung and H. Saito. *The estimation of heights and occupied areas of humans from two orthogonal views for fall detection.* IEEJ Transactions on Electronics and Information and Systems, 2013.

[96] C. Rougier, E. Auvinet, J. Rousseau, M. Mignotte and J. Meunier. *Fall detection from depth map video sequences.* In International Conference on Smart Homes and Health Telematics , pp. 121-128.

[97] R. Planinc and M. Kampel. *Introducing the use of depth data for fall detection.* Personal and Uniquitous Computing, 17(6) ; 2012, pp. 1063-1072.

[98] R. Planinc and M. Kampel. *Robust fall detection by combining 3d data and fuzzy logic.* ACCV Workshop on Color Depth Fusion in Computer Vision, volume 2, 2012, pages 109-120.

[99] C. Zhang, Y. Tian and E. Capezuti. *Privacy preserving automatic fall detection for elderly using rgbd cameras.* In International Conference on Computers Helping People with Special Needs, 2012, pp. 625-633.

[100] Z.Zhang, W. Liu, V. Metsis and V. Athitsos. *A viewpoint-independent statistical method for fall detection.* In International Conference on Pattern Recognition, 2012, pp. 3626-3630.

[101] G. Mastorakis and D. Makris. *Fall detection system using kinect's infrared sensor.* Journal of Real-Time image Processing, 2012, pp. 1-12.

[102] R. Dubey, B. Ni, and P. Moulin. *A depth camera based fall recognition system for the elderly.* In International Conference on Image Analysis and Recognition, 2012, pp. 106-113.

[103] M. A. Mousse, C. Motamed, E. C. Ezin. *Percentage of human occupied areas for fall detection from two views.* The Visual Computer, 2016. DOI : 10.1007/s00371-016-1296-y.

[104] M. A. Mousse, C. Motamed, E. C. Ezin. *Video-Based People Fall Detection via Homography Mapping of Foreground Polygons from Overlapping Cameras.* IEEE 11th International Conference on Signal-Image Technology and Internet-Based Systems, pp. 164-169, 2015. DOI : 10.1109/SITIS.2015.56.

[105] M. A. Mousse, C. Motamed, E. C. Ezin. *A Multi-View Human Bounding Volume Estimation for Posture Recognition in Elderly Monitoring System.* International Conference on Pattern Recognition Systems, 2016. DOI : 10.1049/ic.2016.0026.

[106] M. A. Mousse, C. Motamed, E. C. Ezin. *Fast polygons fusion for multi-views moving object detection from overlapping camera.* 13th African Conference on Research in Computer Science and Applied Mathematics, 2016, pp. 262-268.

[107] O. Tropp, A. Tal and I. Shimshoni. *A fast triangle to triangle intersection test for collision detection.* Comp. Anim. Virtual Worlds, 2006, pp. 527-535.

[108] J. Meunier E. Auvinet, C. Rougier and J. Rousseau St-Arnaud. *Multiple came-*

ras fall dataset. Technical Report 1350, Université de Montréal, DIRO, LISA, 2010.

[109] R. Hartley. *In defense of the eight-point algorithm*. Pattern Analysis and Machine Intelligence,19(6), 1997, pp. 580-593.

[110] D. Nister. *An efficient solution to the five-point relative pose problem*. Pattern Analysis and Machine Intelligence, 26(6), 2004, pp. 756-777,

[111] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Second edition, 2004.

Résumé

La vision artificielle est un domaine de recherche en plein évolution. Les nouvelles stratégies permettent d'avoir des réseaux de caméras intelligentes. Cela induit le développement de beaucoup d'applications de surveillance automatique via les caméras. Les travaux développés dans cette thèse concernent la mise en place d'un système de vidéo-surveillance intelligente pour la détection de chutes en temps réel. La première partie de nos travaux consiste à pouvoir estimer de façon robuste la surface d'une personne à partir de deux (02) caméras ayant des vues complémentaires. Cette estimation est issue de la détection de chaque caméra. Dans l'optique d'avoir une détection robuste, nous avons fait recours à deux approches. La première approche consiste à combiner un algorithme de détection de mouvements basé sur la modélisation de l'arrière plan avec un algorithme de détection de contours. Une approche de fusion a été proposée pour rendre beaucoup plus efficiente le résultat de la détection. La seconde approche est basée sur les régions homogènes de l'image. Une première segmentation est effectuée dans le but de déterminer les régions homogènes de l'image. Et pour finir nous faisons la modélisation de l'arrière plan en se basant sur les régions. Une fois les pixels de premier plan obtenu, nous faisons une approximation par un polygone dans le but de réduire le nombre d'informations à manipuler. Pour l'estimation de cette surface nous avons proposé une stratégie de fusion dans le but d'agréger les détections des caméras. Cette stratégie conduit à déterminer l'intersection de la projection des divers polygones dans le plan de masse. La projection est basée sur les principes de l'homographie planaire. Une fois l'estimation obtenue, nous avons proposé une stratégie pour détecter les chutes de personnes. Notre approche permet aussi d'avoir une information précise sur les différentes postures de l'individu. Les divers algorithmes proposés ont été implémentés et testés sur des banques de données publiques

dans le but de juger l'efficacité des approches proposées par rapport aux approches existantes dans l'état de l'art. Les résultats obtenus et qui ont été détaillés dans le présent manuscrit montre l'apport de nos algorithmes.

Mots-clés: détection de chute, classification des postures, homographie planaire, détection de mouvement, Codebook, vidéosurveillance intelligente.

Abstract

Artificial vision is an evolving field of research. The new strategies make it possible to have some autonomous networks of cameras. This leads to the development of many automatic surveillance applications using the cameras. The work developed in this thesis concerns the setting up of an intelligent video surveillance system for real-time people fall detection. The first part of our work consists of a robust estimation of the surface area of a person from two (02) cameras with complementary views. This estimation is based on the detection of each camera. In order to have a robust detection, we propose two approaches. The first approach consists in combining a motion detection algorithm based on the background modeling with an edge detection algorithm. A fusion approach has been proposed to make much more efficient the results of the detection. The second approach is based on the homogeneous regions of the image. A first segmentation is performed to find homogeneous regions of the image. And finally we model the background using obtained regions. Once the foreground pixels are obtained, we approximate it by a polygon to reduce the number of information. For the estimation of the surface, we proposed a fusion strategy in order to aggregate the detections of the cameras. This strategy leads to determine the intersection of the projection of the various polygons into ground plane. The projection is based on the principles of Planar homography. Once the estimate was obtained, we use a strategy to detect falls. Our approach also allows to have precise information about the different postures of the individual. All proposed algorithms have been implemented and tested on public databases in order to judge the effectiveness of these approaches and to make camparision with existing approaches. The results obtained which are detailed in the present manuscript show the contribution of our algorithms.

Keywords: fall detection, posture classification, planar homography, motion detection, Codebook, intelligent video surveillance