Détection de la présence humaine par vision

Yannick Benezeth

PUCVL
Pôle Universités
Centre Val de Loire

UNIVERSITÉ D'ORLÉANS

UNIVERSITE D'ORLEANS

ÉCOLE DOCTORALE SCIENCES ET TECHNOLOGIES

Institut PRISME

THÈSE présentée par :

Yannick BENEZETH

soutenue le : **28 octobre 2009**

pour obtenir le grade de : **Docteur de l'université d'Orléans**

Discipline/ Spécialité : **Sciences et Technologies Industrielles**

Détection de la présence humaine par vision

THÈSE dirigée par :

Christophe ROSENBERGER	Professeur des Universités, ENSICAEN

RAPPORTEURS :

Jean-Marc CHASSERY	Directeur de Recherche CNRS - Grenoble
Frédéric TRUCHETET	Professeur des Universités, Université de Bourgogne

JURY :

Jean-Marc CHASSERY	Directeur de Recherche CNRS - Grenoble
Frédéric TRUCHETET	Professeur des Universités, Université de Bourgogne
Pierre-Marc JODOIN	Professeur Adjoint, Université de Sherbrooke (Canada)
Gérard POISSON	Professeur des Universités, Université d'Orléans - Président du jury
Christophe ROSENBERGER	Professeur des Universités, ENSICAEN
Bruno EMILE	Maître de Conférences, Université d'Orléans

Remerciements

Cette thèse a été effectuée au sein de l'équipe Images et Signaux pour les Systèmes (*ISS*) de l'institut PRISME et financée par le projet *CAPTHOM* du pôle de compétitivité de la région Centre $S2E2$.

J'exprime tout d'abord ma reconnaissance à Frédéric Truchetet et Jean-Marc Chassery d'avoir accepté d'être les rapporteurs de ce mémoire. Mes remerciements vont ensuite à Gérard Poisson et Pierre-Marc Jodoin pour avoir accepté d'examiner ces travaux.

Je souhaite ensuite exprimer de volumineux remerciements à Christophe Rosenberger pour avoir accepté de diriger ma thèse, à Bruno Emile pour son encadrement et également à Hélène Laurent. Je les remercie chaleureusement ici pour leur disponibilité, leurs conseils avisés et leur sempiternelle bonne humeur. J'ai eu de la chance de travailler à leurs côtés.

Mes vifs remerciements vont ensuite à Pierre-Marc Jodoin pour m'avoir si bien accueilli durant trois mois à l'Université de Sherbrooke, au Canada, et pour avoir partager avec moi son enthousiasme et sa passion de la recherche. Je remercie également Venkatesh Saligrama pour m'avoir fait l'honneur de m'accueillir durant deux mois à Boston University. Je les remercie vigoureusement pour ces expériences internationales qui m'ont enrichi scientifiquement et humainement.

Les travaux présentés dans ce mémoire font état des recherches menées au sein du projet *CAPTHOM*. J'exprime alors toute ma gratitude envers tous les partenaires industriels et universitaires de ce projet ainsi qu'envers ST Imaging pour leur collaboration.

Je tiens à saluer les stagiaires, thésards, post-docs, permanents et personnels administratifs de l'Université de Sherbrooke, de Boston University, de l'ENSI de Bourges et de l'IUT de Bourges pour tous les instants de détente, repas ou pauses café partagés ensemble. Je présente particulièrement d'éléphantesques remerciements à Antoine, Damien et Pierre, mes collègues et amis thésards du projet *CAPTHOM*, à Baptiste mon ancien collègue de bureau, à Adel et Hazem.

Et j'exprime finalement de gigantesques remerciements à ma famille ainsi qu'à Juliette pour leur amour, leur soutien et leurs encouragements.

Résumé

Les travaux présentés dans ce manuscrit traitent de la détection de personnes dans des séquences d'images et de l'analyse de leur activité. Ces travaux ont été menés au sein de l'Institut PRISME dans le cadre du projet *CAPTHOM* du pôle de compétitivité $S2E2$ (Sciences et Systèmes de l'Énergie Électrique).

Après un état de l'art sur l'analyse de séquences d'images pour l'interprétation automatique de scènes de vidéo-surveillance et une étude comparative de différents modules mis en place dans ce contexte, nous présentons la méthode de détection de personnes proposée dans le cadre du projet *CAPTHOM*. Celle-ci s'articule autour de trois étapes : la détection de changement, le suivi d'objets mobiles et la classification. La détection de changement est réalisée avec une soustraction de l'arrière-plan. Une mise à jour à trois niveaux différents nous permet de gérer les changements de l'environnement les plus fréquents (variation d'illumination *etc.*). Ensuite, un suivi des objets mobiles basé sur l'analyse des composantes connectées et le suivi de points d'intérêt nous permet d'obtenir un historique des déplacements des objets dans le plan image. Finalement, la nature de ces objets est déterminée en utilisant plusieurs classifieurs par partie, un indice de confiance sur cette information est alors construit. Ce système a été évalué sur une large base de vidéos correspondant à des scénarios de cas d'usage de *CAPTHOM* établis par les partenaires du projet.

Ensuite, nous présentons des méthodes permettant d'obtenir, à partir du flux vidéo d'une ou deux caméras, d'autres informations de plus haut-niveau sur l'activité des personnes détectées. Nous présentons tout d'abord une mesure permettant de quantifier leur activité. Ensuite, un système de stéréovision multi-capteurs combinant une caméra infrarouge et une caméra visible est utilisé pour augmenter les performances du système de détection mais aussi pour permettre la localisation dans l'espace des personnes et donc accéder à une cartographie des déplacements effectués. Finalement, une méthode de détection d'événements anormaux, basée sur des statistiques de distributions spatiales et temporelles des pixels de l'avant-plan est détaillée. Les méthodes proposées offrent un panel de solutions performantes sur l'extraction d'informations haut-niveau à partir de séquences d'images.

Abstract

The work presented in this manuscript deals with people detection and activity analysis in images sequences. This work has been done in the PRISME institut within the framework of the *CAPTHOM* project of the French Cluster $S2E2$.

After a state of the art on video analysis and a comparative study of several video surveillance tools, we present the people detection method proposed in the *CAPTHOM* project. This method is based on three steps : change detection, mobile objects tracking and classification. Change detection is carried out with a background subtraction. The background model is updated based on three different levels in order to manage the most current changes in the environnement (illumination variation *etc*). Then, we track mobile objects using the analysis of connected components and the tracking of points of interest. Finally, the nature of these objects is determined using several part-based classifiers, a confidence index on the obtained result is built. This system was assessed on a wide videos dataset corresponding to use cases established by the industrial partners of the *CAPTHOM* project.

Then, we present methods used to obtain other high-level information concerning the activity of detected persons. A criterion for characterizing their activity is presented. Then, a multi-sensors stereovision system combining an infrared and a daylight camera is used to increase performances of the people detection system but also to localize persons in the 3D space and build the moving cartography. Finally, an abnormal events detection method based on statistics about spatio-temporal foreground pixel distribution is presented. These proposed methods offer robust and efficient solutions on high-level information extraction from images sequences.

Table des matières

Introduction Générale

Le secteur du bâtiment est aujourd'hui un des secteurs qui consomme le plus d'énergie, devant les secteurs des transports et de l'industrie. En France par exemple, le secteur du bâtiment est responsable de 21% des émissions de CO2 et de 43% de la consommation d'énergie totale [4]. Les pays industrialisés s'étant engagés à réduire leurs émissions de gaz à effet de serre, le secteur du bâtiment offre de grandes possibilités d'économie d'énergie. Plusieurs pistes sont actuellement envisagées pour réaliser ces économies. Il y a tout d'abord la production décentralisée d'énergie à partir d'énergie renouvelable, les solutions passives d'économie d'énergie (isolation *etc.*) puis les solutions de gestion active des consommations d'énergie. Pour permettre le développement de ces dernières solutions, il est indispensable de disposer d'informations fiables sur l'occupation des bâtiments.

Les détecteurs de présence actuellement sur le marché sont majoritairement des détecteurs à Infra-Rouge Passif (IRP). Une description complète de cette technologie a été réalisée par J.F. Gobeau [63]. Le principe de ce capteur repose sur le fait que le corps humain émet un rayonnement électromagnétique de longueur d'ondes dans l'infrarouge (entre 6 et 14 μm). Lorsqu'une personne se déplace dans le champ du détecteur, son rayonnement infrarouge est focalisé par des lentilles de Fresnel sur le capteur pyroélectrique, qui produit en retour un signal électrique. Après traitement, celui-ci déclenche l'action commandée par le détecteur.

Le rôle des lentilles de Fresnel est capital puisque le capteur pyroélectrique n'est sensible qu'aux variations de flux infrarouge. Autrement dit, le capteur ne réagit pas en réponse à un flux infrarouge constant. Les lentilles de Fresnel jouent alors le rôle d'un système optique capable de moduler le flux infrarouge. Elles découpent l'espace surveillé par le capteur en zones de détection et en zones aveugles pour le détecteur (cf. figure 1). Lorsqu'une personne se trouve dans une zone de détection, les lentilles de Fresnel créent alors son "image" sur la surface du capteur qui reçoit donc une fraction de son rayonnement infrarouge. En se déplaçant, la personne traverse successivement des zones de détection et des zones aveugles. Le flux infrarouge incident sur le capteur pyroélectrique varie et un signal électrique est généré en retour.

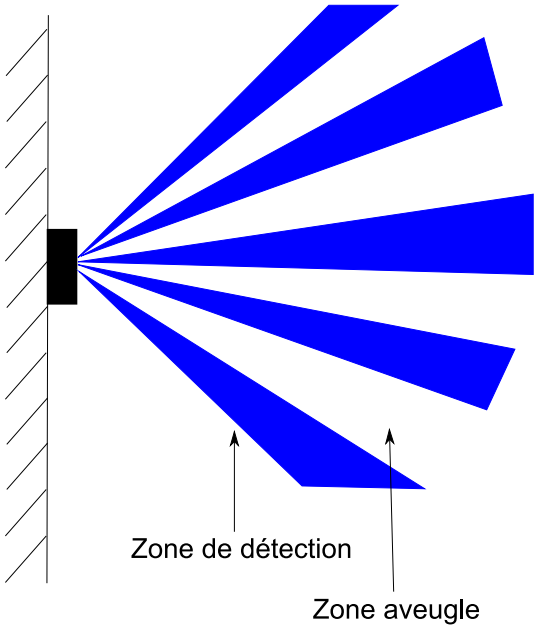


Figure 1: Répartition des zones de détection et des zones aveugles en vue de dessus d'un capteur à IRP.

Cette technologie est aujourd'hui bien connue et est couramment utilisée pour la gestion de l'éclairage, l'ouverture automatique des portes *etc.* Un exemple de détecteur de présence utilisant cette technologie, actuellement disponible sur le marché, est présenté dans la figure 2. Alors que ce type de détecteur est actuellement très répandu, celui-ci présente cependant plusieurs défauts majeurs :

- les personnes immobiles ne peuvent pas être détectées,
- le capteur est sensible aux courants d'air,
- le capteur ne peut distinguer les animaux domestiques des humains.

Les limitations technologiques de ce détecteur de présence, qui n'est en fait qu'un détecteur de mouvement, sont un frein au développement de solutions de gestion de la consommation énergétique innovantes.

Figure 2: Exemple de détecteur de présence (infrarouge passif) actuellement sur le marché.

C'est dans ce contexte que se situe le projet *CAPTHOM* (CAPteur universel

de présence HUMaine pour le bâtiment et l'habitat). Ce projet s'inscrit dans la thématique "Gestion de l'énergie dans le bâtiment" du pôle de compétitivité de la région Centre "Sciences et Systèmes de l'Énergie Électrique" ($S2E2$). Les membres du consortium impliqués dans ce projet sont STMicroelectronics, Legrand, Agilicom, Thermor, Sorec, Wirecom Technologies, Pôle Capteurs, CRESITT Industrie et l'institut PRISME.

L'objectif du projet est de développer un capteur fiable et performant détectant la présence d'un humain dans un environnement intérieur. Le capteur doit s'insérer dans des bâtiments résidentiels ou tertiaires et se substituer aux détecteurs actuels. Les objectifs principaux d'un tel système sont la gestion de la consommation d'énergie et le confort des habitants (adaptation du chauffage à leur activité). De plus, le composant *CAPTHOM* doit pouvoir être facilement intégré dans d'autres domaines d'applications liés à la sécurité ou à la surveillance et à l'assistance des personnes âgées ou handicapées, à domicile ou en institution spécialisée.

Le capteur doit être capable de détecter la présence d'une personne dans son environnement sans être perturbé par la présence d'animaux, d'autres objets mobiles ou encore par la morphologie ou l'activité des personnes. Le capteur doit être robuste aux changements de luminosité, aux sources de chaleur et doit être capable de détecter des personnes jusqu'à 15 mètres. Les traitements s'effectuant au sein d'une architecture embarquée, la complexité des algorithmes proposés et la mémoire utilisée doivent être compatibles avec ces contraintes matérielles. La solution proposée doit être économiquement acceptable et avoir un temps de réponse très court. Afin de respecter les règles de la directive européenne *2005/32/CE* appelée *EuP* pour *Eco-Design of Energy using Products*, traitant de l'éco-conception des produits consommateurs d'énergie et fixant des contraintes sur la consommation des appareils électriques, l'énergie consommée par le capteur doit être très faible.

C'est dans ce contexte que s'inscrit cette thèse, l'objectif étant de proposer des solutions algorithmiques pour détecter des personnes à partir de séquences d'images. Grâce aux récentes avancées technologiques concernant tout d'abord les systèmes d'acquisition vidéo mais surtout les capacités de calcul continuellement grandissantes des unités de traitement embarquées, il est aujourd'hui totalement réaliste de concevoir une caméra et son unité de traitement associé, comme étant un capteur à part entière. Dans ce cas, l'objectif de la caméra n'est plus de renvoyer une image à un opérateur qui traitera celle-ci, mais de renvoyer le flux vidéo à une unité de traitement qui renverra des informations plus "haut-niveau" comme par exemple : *une personne est présente dans mon champ de vision*. Le principal avantage lié à l'utilisation de caméras est le nombre très important d'informations qu'il est possible d'obtenir à partir de l'analyse d'une vidéo.

L'utilisation de caméras "intelligentes" dans notre quotidien soulève le problème du respect de la vie privée et également le problème de l'acceptation de ce capteur par les utilisateurs finaux. Ces questions ont été largement considérées au sein du projet *CAPTHOM* et il a été décidé que la caméra et son unité de traitement associée ne renverront que des informations utiles aux systèmes de gestion technique du bâtiment. Pour l'application visée, aucune image ne sera donc transmise par le capteur vers l'extérieur.

Inconsciemment, le cerveau humain traite continuellement des flux d'images pour en extraire des informations utiles à la perception de son environnement. Nous sommes par exemple capables de reconnaître des milliers d'objets, des caractères, nous pouvons nous situer dans l'espace ou encore reconnaître des actions impliquant des informations spatio-temporelles. Depuis plusieurs années, les chercheurs en vision par ordinateur ont pour objectif de permettre aux systèmes informatisés d'extraire automatiquement des informations haut-niveau, d'un point de vue sémantique, à partir du contenu d'une vidéo. Cependant, considérant le fait qu'une vidéo n'est qu'une succession de tableaux contenant des valeurs discrètes, le challenge reste entier.

Lorsque l'on souhaite détecter un objet dans une image ou une vidéo, on est alors confronté à plusieurs difficultés :

- premièrement, une image est une représentation 2D d'une scène 3D. Un même objet, observé avec un point de vue de la caméra légèrement différent, peut avoir une apparence très différente sur l'image. Si on ne dispose pas d'information *a priori* sur les positions relatives des personnes et de la caméra, on doit alors être capable de gérer le nombre infini de configurations possibles.
- Deuxièmement, les conditions d'acquisition de l'image peuvent varier d'un environnement à un autre. Celles-ci peuvent également varier dans le temps.
- Troisièmement, les arrière-plans peuvent être très complexes. Les possibilités de fausses détections sont donc nombreuses et le contraste entre les personnes à détecter et l'arrière-plan peut éventuellement être très faible.
- Ensuite, il peut y avoir de nombreuses occultations entre la personne et son environnement, entre plusieurs personnes et également entre la personne et elle-même.
- Finalement, la principale difficulté que l'on rencontre lorsque l'on veut détecter des personnes est la très grande variabilité intra-classe. De par leurs vêtements, leurs tailles, leurs poids, leurs coupes de cheveux, l'apparence de deux personnes peut être très différente. De plus, le corps humain étant hautement articulé, le nombre de poses possibles est très grand et la silhouette d'une personne change au cours du temps.

Nous nous sommes intéressés dans ce travail à l'ensemble de ces difficultés afin de développer un système permettant de gérer un grand nombre de ces situations.

La suite du mémoire est organisée comme suit :

- **Le chapitre 1** présente un état de l'art sur l'analyse de séquences d'images en vidéo-surveillance. Nous présentons ici plus particulièrement les méthodes utilisées actuellement pour la détection de changement, le suivi de personnes, la classification et la reconnaissance d'activités humaines.

- **Le chapitre 2** présente une étude comparative de différents modules de vidéo-surveillance utilisés dans le cadre de ces travaux. Nous insistons plus particulièrement sur les méthodes de soustraction de l'arrière-plan et les méthodes de classification.

- **Le chapitre 3** présente la méthode proposée dans le cadre du projet

CAPTHOM pour la détection de personnes.

- **Le chapitre 4** présente quelques applications possibles d'un *CAPTHOM*. Nous présentons tout d'abord un critère sur l'activité des personnes détectées. Nous présentons ensuite un système de stéréovision multi-capteurs utilisé pour augmenter les performances de détection du système précédent mais également pour localiser les personnes dans l'espace. Une méthode pour détecter des événements anormaux est ensuite présentée.

- Les conclusions et les perspectives de ces travaux sont ensuite présentées dans la dernière partie.

CHAPITRE 1

État de l'art sur l'analyse de séquences d'images pour la vidéo-surveillance

Nous présentons dans ce chapitre les méthodes de l'état de l'art les plus fréquemment utilisées pour l'analyse de séquence d'images en vidéo-surveillance. Nous nous intéressons plus particulièrement à la détection de personnes et à la reconnaissance d'activités. Un exemple de processus d'analyse vidéo est présenté dans la figure 1.1. Dans ce chapitre, nous détaillons les méthodes de détection de changement (plus particulièrement la soustraction de l'arrière-plan), le suivi d'objets mobiles, la reconnaissance d'humains et ensuite les méthodes de reconnaissance d'activités humaines.

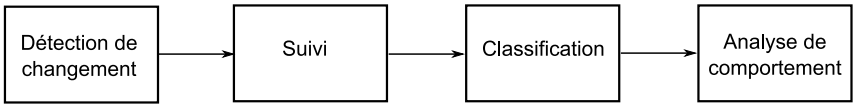


FIGURE 1.1: Exemple de processus généralement mis en place pour l'analyse de séquences d'images en vidéosurveillance.

1.1 Détection de changement

Beaucoup d'applications en vision par ordinateur commencent par une étape de détection de changement. L'objectif est en général double. Tout d'abord, il s'agit de simplifier les traitements ultérieurs en localisant les régions d'intérêt dans l'image pour optimiser le temps de calcul. Ensuite, le nombre de fausses détections est également réduit puisque les régions de l'image où la probabilité de trouver une personne est très faible sont éliminées.

C'est une phase très importante car les étapes suivantes se baseront sur ce résultat. À partir d'un modèle de l'environnement et d'une observation ou d'une série d'observations successives, on cherche à détecter ce qui a changé. Pour notre application, les régions d'intérêt détectées correspondent aux régions de l'image où il y a une forte probabilité qu'il y ait un humain.

Il existe principalement trois classes de méthodes dans la littérature pour détecter les régions d'intérêt. Il y a tout d'abord les méthodes qui utilisent le flot optique. Ces méthodes extraient un champ de vitesses à partir d'une séquence d'images en posant l'hypothèse que l'intensité lumineuse d'un point est conservée au cours du déplacement. On peut citer par exemple Meyer *et al.* [108] ou Shin *et al.* [133]. Le principal avantage de ces méthodes est qu'il est possible d'obtenir le mouvement global de la caméra et donc de segmenter les régions d'intérêt avec une caméra mobile. Dans le contexte de notre application, le calcul du flot optique sur toute l'image ne semble pas nécessaire puisque nous disposons d'une caméra statique. De plus, ces méthodes nécessitent beaucoup de calcul et les ressources du matériel visé par notre application sont limitées. Ces méthodes ne seront donc pas détaillées ici.

Il existe ensuite les méthodes qui calculent une différence temporelle, pixel par pixel, entre deux ou trois images successives. On peut citer par exemple Lipton *et al.* [99] et Huwer et Niemannqui [79]. La valeur absolue de cette différence est seuillée pour détecter les changements. Ensuite, les pixels labellisés "en mouvement" sont regroupés en objets avec une analyse en composantes connectées. Cette méthode présente l'avantage d'être adaptée aux environnements dynamiques puisqu'elle n'est pas influencée par les variations d'illumination mais ne permet pas de récupérer tous les pixels de l'objet en mouvement. Un exemple synthétique est présenté dans la figure 1.2.

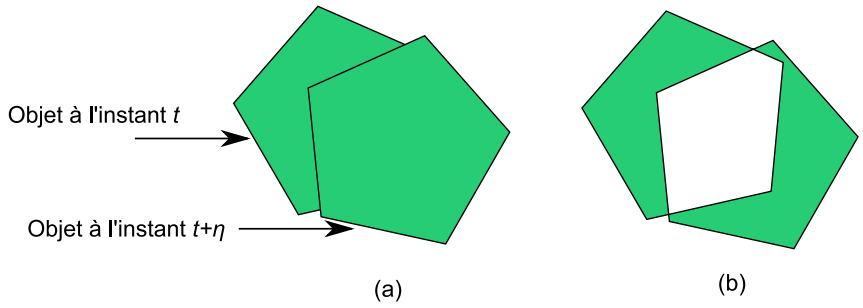


Figure 1.2: Exemple de résultat obtenu avec une différence temporelle (a) objets en mouvement (b) avant-plan détecté.

Il y a enfin les méthodes de soustraction de l'arrière-plan. Basées sur l'hypothèse que l'objet en mouvement est d'une couleur différente de l'arrière-plan, ces méthodes identifient comme appartenant à l'avant-plan les pixels à l'instant t dont la couleur diffère de la couleur de ce même pixel dans l'arrière-plan [164]. Nous appelons "arrière-plan" l'union de toutes les zones statiques de l'image et "avant-plan" les zones de l'image où un changement a été détecté. Dans ce cas, la soustraction de l'arrière-plan peut simplement se formuler par :

$$\mathcal{X}_t(s) = \begin{cases} 1 & \text{si } d(I_{s,t}, B_s) > \tau \\ 0 & \text{sinon,} \end{cases} \tag{1.1}$$

où $\mathcal{X}_t$ est le masque de l'avant-plan à l'instant t, d est la distance entre $I_{s,t}$ la valeur du pixel s à l'instant t et B_s le modèle de l'arrière-plan au pixel s ; τ est un seuil. Les pixels de $\mathcal{X}_t$ à 1 correspondent à l'avant-plan et les pixels à 0 correspondent à l'arrière-plan. Dans sa formulation la plus simple, le modèle de l'arrière-plan peut être une image correspondant à un instant où il n'y avait pas d'objet en mouvement dans la scène et la distance d peut être une distance euclidienne. Cette formulation n'est évidemment pas adaptée pour les applications réelles car le modèle doit être capable de gérer les difficultés suivantes :

- changement de luminosité (brusque ou progressif),
- ombres et réflexions,
- textures animées dans l'arrière-plan (branches d'arbres, écrans de télévision etc.),
- introduction ou retrait d'objets statiques dans la scène,
- secousses de la caméra,
- ...

De nombreux modèles de l'arrière-plan ont été présentés dans la littérature pour gérer ces difficultés. Ces méthodes sont alors plus robustes (du moins en théorie) aux instabilités de l'arrière-plan que les méthodes basiques. Nous présentons dans la suite de ce chapitre, les méthodes de l'état de l'art les plus utilisées pour réaliser la soustraction de l'arrière-plan.

1.1.1 Modélisation avec un filtre moyenneur temporel

La manière la plus simple pour modéliser l'arrière-plan est d'utiliser une image estimée par un filtre moyenneur temporel [164, 114]. Dans ce cas, le modèle est itérativement mis à jour par un filtre récursif :

$$B_{s,t+1} = (1 - \alpha)B_{s,t} + \alpha.I_{s,t} \tag{1.2}$$

où α est un paramètre de mise à jour dont la valeur varie entre 0 et 1. Plus la valeur de α est grande, plus le modèle s'adapte rapidement aux changements. Cependant, le choix de cette valeur est délicat car les objets de l'avant-plan immobiles sont inclus

dans le modèle de l'arrière-plan plus ou moins rapidement selon la valeur de α. Les pixels de l'avant-plan sont détectés en seuillant une des distances suivantes :

$$d_0(I_{s,t}, B_{s,t}) = |I_{s,t} - B_{s,t}|, \tag{1.3}$$
$$d_1(\boldsymbol{I}_{s,t}, \boldsymbol{B}_{s,t}) = |I_{r,s,t} - B_{r,s,t}| + |I_{v,s,t} - B_{v,s,t}| + |I_{b,s,t} - B_{b,s,t}|, \tag{1.4}$$
$$d_2(\boldsymbol{I}_{s,t}, \boldsymbol{B}_{s,t}) = (I_{r,s,t} - B_{r,s,t})^2 + (I_{v,s,t} - B_{v,s,t})^2 + (I_{b,s,t} - B_{b,s,t})^2, \tag{1.5}$$
$$d_3(\boldsymbol{I}_{s,t}, \boldsymbol{B}_{s,t}) = \max\{|I_{r,s,t} - B_{r,s,t}|, |I_{v,s,t} - B_{v,s,t}|, |I_{b,s,t} - B_{b,s,t}|\}. \tag{1.6}$$

où les indices r, v et b correspondent aux composantes *rouge*, *verte* et *bleue*. $I_{s,t}$ est la valeur en niveaux de gris du pixel s à l'instant t utilisée par la distance d_0 et $\boldsymbol{I}_{s,t}$ correspond au vecteur des trois composantes de couleur utilisées par les autres distances qui opèrent toutes sur des images couleur.

1.1.2 Modélisation gaussienne de l'arrière-plan

De nombreux auteurs modélisent chaque pixel de l'arrière-plan par une densité de probabilité déterminée à partir d'une série d'images d'apprentissage, en général une distribution gaussienne est utilisée [168]. Dans ce cas, la soustraction de l'arrière-plan est réalisée en seuillant la fonction de densité de probabilité. Par exemple, Wren *et al.* [153] modélisent chaque pixel de l'arrière-plan par une distribution gaussienne. La distance utilisée peut être le logarithme de la densité de probabilité :

$$d_G = -\frac{1}{2}(\boldsymbol{I}_{s,t} - \boldsymbol{\mu}_{s,t})\boldsymbol{\Sigma}_{s,t}^{-1}(\boldsymbol{I}_{s,t} - \boldsymbol{\mu}_{s,t})^T - \frac{1}{2}\log(|\boldsymbol{\Sigma}_{s,t}|) - \frac{3}{2}\log(2\pi) \tag{1.7}$$

ou la distance de Mahalanobis :

$$d_M = (\boldsymbol{I}_{s,t} - \boldsymbol{\mu}_{s,t})\boldsymbol{\Sigma}_{s,t}^{-1}(\boldsymbol{I}_{s,t} - \boldsymbol{\mu}_{s,t})^T \tag{1.8}$$

où $\boldsymbol{\mu}_{s,t}$ est le vecteur correspondant aux moyennes des composantes *rouge*, *verte* et *bleue* du pixel s à l'instant t, $\boldsymbol{\Sigma}_{s,t}$ est la matrice de covariance. Par la pondération de l'inverse de la covariance, un pixel situé dans une zone où l'image est bruitée, devra avoir une grande variation pour être labellisé appartenant à l'avant-plan. Pour prendre en compte les variations d'illumination, la moyenne et la covariance peuvent être itérativement mises à jour par :

$$\boldsymbol{\mu}_{s,t+1} = (1-\alpha).\boldsymbol{\mu}_{s,t} + \alpha.\boldsymbol{I}_{s,t}, \tag{1.9}$$
$$\boldsymbol{\Sigma}_{s,t+1} = (1-\alpha).\boldsymbol{\Sigma}_{s,t} + \alpha.(\boldsymbol{I}_{s,t} - \boldsymbol{\mu}_{s,t})(\boldsymbol{I}_{s,t} - \boldsymbol{\mu}_{s,t})^T. \tag{1.10}$$

On note que la matrice de covariance est une matrice 3×3 que l'on peut considérer comme diagonale pour limiter les temps de calcul. Pour une matrice de covariance diagonale,

$$\Sigma_{s,t} = \begin{pmatrix} \sigma^2_{r,s,t} & 0 & 0 \\ 0 & \sigma^2_{v,s,t} & 0 \\ 0 & 0 & \sigma^2_{b,s,t} \end{pmatrix} \tag{1.11}$$

les termes diagonaux sont mis à jour par :

$$\begin{cases} \sigma^2_{r,s,t+1}=(1-\alpha)\sigma^2_{r,s,t} + \alpha(I_{r,s,t} - \mu_{r,s,t})^2 \\ \sigma^2_{v,s,t+1}=(1-\alpha)\sigma^2_{v,s,t} + \alpha(I_{v,s,t} - \mu_{v,s,t})^2 \\ \sigma^2_{b,s,t+1}=(1-\alpha)\sigma^2_{b,s,t} + \alpha(I_{b,s,t} - \mu_{b,s,t})^2 \end{cases} \tag{1.12}$$

où $\sigma^2_{r,s,t}$, $\sigma^2_{v,s,t}$ et $\sigma^2_{b,s,t}$ correspondent respectivement aux variances des composantes rouge, verte et bleue au pixel de coordonnées s à l'instant t.

1.1.3 Modélisation avec un mélange de gaussiennes

Modéliser un pixel de l'arrière-plan avec une distribution gaussienne n'est quelques fois pas suffisant. Un même pixel peut par exemple représenter le ciel à un instant donné puis une feuille d'arbre à un autre instant. Pour prendre en compte les arrière-plans qui ont des textures animées (les vagues sur la mer, les branches d'un arbre agitées par le vent, un ventilateur, les écrans de télévision ou d'ordinateur *etc.*), des fonctions de densité de probabilité multimodales ont été proposées (*e.g.* [95]). Par exemple, Stauffer et Grimson [137] modélisent chaque pixel par un mélange de K gaussiennes. La probabilité d'occurrence d'une couleur au pixel s est représentée par :

$$P(\boldsymbol{I}_{s,t}) = \sum_{i=1}^{K} \omega_{i,s,t}.\eta(\boldsymbol{I}_{s,t}, \boldsymbol{\mu}_{i,s,t}, \boldsymbol{\Sigma}_{i,s,t}) \tag{1.13}$$

où $\omega_{i,s,t}$ est le poids de la $i^{\text{ème}}$ gaussienne et $\eta(\boldsymbol{I}_{s,t}, \boldsymbol{\mu}_{i,s,t}, \boldsymbol{\Sigma}_{i,s,t})$ est le $i^{\text{ème}}$ modèle gaussien défini par :

$$\eta(\boldsymbol{I}_{s,t}, \boldsymbol{\mu}_{i,s,t}, \boldsymbol{\Sigma}_{i,s,t}) = \frac{1}{\sqrt{(2\pi)^3|\boldsymbol{\Sigma}_{i,s,t}|}}\ e^{-\frac{1}{2}(\boldsymbol{I}_{s,t}-\boldsymbol{\mu}_{i,s,t})\boldsymbol{\Sigma}^{-1}_{i,s,t}(\boldsymbol{I}_{s,t}-\boldsymbol{\mu}_{i,s,t})^T}. \tag{1.14}$$

Dans [137], la matrice de covariance $\boldsymbol{\Sigma}_{i,s,t}$ est estimée par une matrice diagonale :

$$\boldsymbol{\Sigma}_{i,s,t} = \sigma^2_{i,s,t}.\boldsymbol{Id} \tag{1.15}$$

où $\boldsymbol{Id}$ représente ici la matrice identité. La mise à jour du modèle peut être réalisée avec la méthode d'*Expectation Maximisation* (*EM*) [112] ou alors en mettant à jour les distributions qui "correspondent" (il y a correspondance si la distance de Mahalanobis entre $\boldsymbol{I}_{s,t}$ et $\boldsymbol{\mu}_{i,s,t}$ est inférieure à un seuil) avec :

$$\omega_{i,s,t} = (1-\alpha).\omega_{i,s,t-1} + \alpha \tag{1.16}$$
$$\boldsymbol{\mu}_{i,s,t} = (1-\rho).\boldsymbol{\mu}_{i,s,t-1} + \rho.\boldsymbol{I}_{s,t} \tag{1.17}$$
$$\sigma^2_{i,s,t} = (1-\rho).\sigma^2_{i,s,t-1} + \rho.(\boldsymbol{I}_{s,t} - \boldsymbol{\mu}_{i,s,t-1})^T(\boldsymbol{I}_{s,t} - \boldsymbol{\mu}_{i,s,t-1}) \tag{1.18}$$

où α correspond au taux d'apprentissage défini par l'utilisateur ($0 \leq \alpha \leq 1$) et ρ correspond à un second taux d'apprentissage défini par $\rho = \alpha\eta(\boldsymbol{I}_{s,t}, \boldsymbol{\mu}_{i,s,t}, \boldsymbol{\Sigma}_{i,s,t})$. Les paramètres $\boldsymbol{\mu}$ et σ de la composante qui ne correspond pas restent inchangés et $\omega_{i,s,t} = (1-\alpha)\omega_{i,s,t-1}$. Si aucune composante ne correspond, celle avec le poids le plus faible est remplacée par une nouvelle gaussienne, initialisée avec une moyenne $\boldsymbol{I}_{s,t}$, une variance σ_0^2 et un poids ω_0. Après avoir mis à jour chaque gaussienne, les poids $\omega_{i,s,t}$ sont normalisés. Ensuite, les K distributions sont ordonnées selon leur valeur de $\omega_{i,s,t}/\sigma_{i,s,t}$. Une distribution modélise l'arrière-plan si elle apparaît fréquemment ($\omega_{i,s,t}$ grand) et varie peu ($\sigma_{i,s,t}$ faible). Les distributions de l'arrière-plan les plus probables restent alors les premières. Donc, seules les H premières distributions les plus fiables ($\omega_{i,s,t}$ grand et $\sigma_{i,s,t}$ faible) sont conservées avec :

$$H = \underset{h}{\operatorname{argmin}}(\sum_{i=1}^{h} \omega_i > T) \tag{1.19}$$

où T est un seuil. Si T est faible, une distribution unimodale représentera l'arrière-plan. Inversement, si T est grand, plusieurs distributions représenteront l'arrière-plan (dans la limite du nombre maximal de gaussienne K). La détection est finalement réalisée par :

$$\mathcal{X}_t(s) = \begin{cases} 1 & \text{si } d_M(\boldsymbol{I}_{s,t}, \boldsymbol{\mu}_{i,s,t}) > \tau \quad \forall i \leq H \\ 0 & \text{sinon.} \end{cases} \tag{1.20}$$

Modéliser l'arrière-plan avec un mélange de gaussiennes permet de prendre en compte les arrière-plans multimodaux, cette méthode ne demande pas beaucoup d'espace mémoire. Cependant, cette méthode manque de flexibilité car le nombre de gaussiennes (en général $K = 3$ ou 5) doit être préalablement fixé. De plus, il y a un grand nombre de paramètres à fixer empiriquement ce qui peut rendre très délicat le fait de trouver la combinaison optimale pour une situation donnée.

1.1.4 Modélisation non-paramétrique

Une approche non-paramétrique (ou *Kernel Density Estimation*) peut aussi être utilisée pour modéliser les densités de probabilité multimodales. À partir de N images

$I_{s,i}$, avec $i \in [1, N]$, Elgammal *et al.* [44] construisent une estimation de la densité de probabilité. Si $I_{s,t}$ est une observation à l'instant t et au pixel s, une estimation de sa probabilité d'observation est donnée par :

$$P(I_{s,t}) = \frac{1}{N} \sum_{i=1}^{N} K_\sigma(I_{s,t} - I_{s,i}) \tag{1.21}$$

où K_σ est une fonction noyau. Lorsqu'on travaille avec des vidéos en couleur, on peut utiliser des produits de noyaux 1-dimension :

$$P(\boldsymbol{I}_{s,t}) = \frac{1}{N} \sum_{i=1}^{N} \prod_{j=\{r,v,b\}} K_{\sigma_j}(I_{j,s,t} - I_{j,s,i}). \tag{1.22}$$

Si le noyau K est une gaussienne, la probabilité d'observation du pixel $\boldsymbol{I}_{s,t}$ est donnée par :

$$P(\boldsymbol{I}_{s,t}) = \frac{1}{N} \sum_{i=1}^{N} \prod_{j=\{r,v,b\}} \frac{1}{\sqrt{2\pi\sigma_j^2}} \; e^{-\frac{1}{2}\frac{(I_{j,s,t} - I_{j,s,i})^2}{\sigma_j^2}}. \tag{1.23}$$

Un pixel appartient à l'avant-plan lorsque la probabilité $P(\boldsymbol{I}_{s,t})$ est plus petite qu'un seuil prédéfini. Cette technique peut être utilisée pour modéliser des arrière-plans avec des textures animées. Ce modèle non-paramétrique est beaucoup plus flexible que les mélanges de gaussiennes, car le nombre de modes n'est pas déterminé *a priori*. Cependant, il reste très gourmand en espace mémoire. Elgammal *et al.* proposent dans [44] des pistes d'optimisation de la méthode en utilisant une table dans laquelle est pré-calculée l'estimation de probabilité de l'équation 1.23 pour toutes les valeurs d'intensité.

1.1.5 Minimum, maximum et maximum de différence intertrame

Le système de vidéo-surveillance W^4 (*Who, When, Where, What*) [68] utilise un modèle de l'arrière-plan composé d'un minimum m_s, d'un maximum M_s et d'un maximum de différence entre images consécutives D_s. Dans ce système, un pixel s appartient à l'arrière-plan si :

$$|M_s - I_{s,t}| < \tau d_\mu \quad \text{ou} \quad |m_s - I_{s,t}| < \tau d_\mu \tag{1.24}$$

où τ est un seuil prédéfini et d_μ est la moyenne de la plus grande différence entre images consécutives sur tous les pixels. Haritaoglu *et al.* [68] ont proposé une méthode pour mettre à jour le triplet $[M_s, m_s, D_s]$. La valeur de d_μ est recalculée pour chaque nouvelle image. Notons que cette méthode travaille sur des images en niveaux de gris. De la même manière que la modélisation de l'arrière-plan avec une gaussienne, un pixel situé dans une zone où l'image est bruitée, devra avoir une grande variation pour être déclaré appartenant à l'avant-plan.

1.1.6 Table de codage

Une autre méthode présentée par Kim *et al.* [92], basée sur une table de codage (ou *codebook*), a pour objectif de gérer les arrière-plans multimodaux. À partir d'une séquence d'apprentissage, cette méthode assigne à chaque pixel de l'arrière-plan plusieurs séries de valeurs clés (ou *codewords*). L'ensemble des valeurs clés décrit toutes les couleurs qu'un pixel de l'arrière-plan peut prendre. Par exemple, un pixel dans une zone où l'image est stable sera modélisé par une seule valeur clé alors qu'un pixel dans une zone dynamique sera représenté par plusieurs valeurs clés. Dans la méthode proposée par Kim *et al.* [92], une valeur clé est composée du vecteur moyenne $\boldsymbol{\mu}$ et de 6 autres valeurs numériques (les valeurs minimales et maximales des niveaux de gris, la fréquence d'apparition, la durée de non apparition, le premier et le dernier instant d'apparition de la valeur clé). Avec l'hypothèse que les ombres correspondent à une variation de luminosité et les objets de l'avant-plan à une variation de chrominance, on évalue séparément la variation de la chrominance :

$$\sqrt{I_{r,s,t}^2 + I_{v,s,t}^2 + I_{b,s,t}^2 - \frac{(\mu_{r,i,s}.I_{r,s,t} + \mu_{v,i,s}.I_{v,s,t} + \mu_{b,i,s}.I_{b,s,t})^2}{\mu_{r,i,s}^2 + \mu_{v,i,s}^2 + \mu_{b,i,s}^2}} < \tau \tag{1.25}$$

et la variation de la luminosité :

$$\alpha_{i,s} \leq I_{r,s,t}^2 + I_{v,s,t}^2 + I_{b,s,t}^2 \leq \beta_{i,s} \tag{1.26}$$

où $\mu_{r,i,s}$, $\mu_{v,i,s}$, $\mu_{b,i,s}$, $\alpha_{i,s}$ et $\beta_{i,s}$ sont des paramètres de la i^{eme} valeur clé du pixel s. Si un pixel s satisfait les équations 1.25 et 1.26, il correspond avec la i^{eme} valeur clé et est donc labellisé comme "statique".

1.1.7 Réduction statistique de l'arrière-plan

Oliver *et al.* [116] ont proposé l'utilisation d'un espace propre pour modéliser l'arrière-plan. Avec cette méthode, il est possible de réaliser l'apprentissage de l'arrière-plan dans un environnement non-contraint, c'est-à-dire avec des objets en mouvement pendant la phase d'apprentissage. Alors que les méthodes précédentes proposent une modélisation au niveau des pixels de l'arrière-plan, cette méthode prend en compte les statistiques des pixels voisins dans le modèle.

Soit $\{\boldsymbol{I}_i\}_{i=1:N}$ une représentation en vecteur des N-images utilisées pour l'apprentissage. La moyenne $\boldsymbol{\mu}$ peut être simplement calculée avec :

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{I}_i \tag{1.27}$$

et, à chaque image, on peut soustraire cette moyenne pour construire $\{\boldsymbol{X}_i\}_{i=1:N}$ où $\boldsymbol{X}_i = \boldsymbol{I}_i - \boldsymbol{\mu}$. Ensuite, la matrice de covariance $\boldsymbol{\Sigma}$ est construite avec $\boldsymbol{\Sigma} = E[\boldsymbol{X}\boldsymbol{X}^T]$,

avec $\boldsymbol{X} = [\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n]$. À partir de la transformée de Karhunen-Loeve, on peut calculer la matrice des vecteurs propres $\boldsymbol{\phi}$ qui diagonalise la matrice de covariance $\boldsymbol{\Sigma}$:

$$D = \boldsymbol{\phi}\boldsymbol{\Sigma}\boldsymbol{\phi}^T \tag{1.28}$$

où D est la matrice diagonale correspondante. Selon les principes de l'analyse en composantes principales, une nouvelle matrice rectangulaire $\boldsymbol{\phi}_M$ est construite avec les M vecteurs propres qui ont les plus grandes valeurs propres. Une fois que $\boldsymbol{\phi}_M$ (appelé *Eigen Background*) et la moyenne $\boldsymbol{\mu}$ sont connues, chaque image d'entrée $\boldsymbol{I}_t$ (représentée en colonne) est projetée sur les M vecteurs propres avec :

$$\boldsymbol{B}_t = \boldsymbol{\phi}_M(\boldsymbol{I}_t - \boldsymbol{\mu}) \tag{1.29}$$

et ensuite $\boldsymbol{I}'_t$ est reconstruit avec :

$$\boldsymbol{I}'_t = \boldsymbol{\phi}_M^T\boldsymbol{B}_t + \boldsymbol{\mu}. \tag{1.30}$$

Finalement, les pixels de l'avant-plan sont détectés en calculant la distance entre l'image d'entrée $\boldsymbol{I}_t$ et l'image reconstruite $\boldsymbol{I}'_t$:

$$\mathcal{X}_t(s) = \begin{cases} 1 & \text{si } d_2(\boldsymbol{I}_t, \boldsymbol{I}'_t) > \tau \\ 0 & \text{sinon.} \end{cases} \tag{1.31}$$

La décomposition (équation 1.28) peut être rapidement calculée avec la décomposition en valeur singulière. Cependant, il est difficile de maintenir à jour ϕ_M. Des solutions basées sur l'analyse en composantes principales incrémentale [129, 96] ont cependant été proposées.

1.1.8 Autres méthodes

Nous avons décrit ici quelques unes des méthodes les plus utilisées dans l'état de l'art pour détecter l'avant-plan. Nous pouvons citer également *Mean-Shift* [125] où les maxima locaux des densités de probabilité de la distribution multimodale sont estimés à chaque itération. Cette méthode est très souple et demande moins d'espace mémoire que le modèle non-paramétrique proposé par [44] mais le nombre de calculs nécessaires pour chaque pixel est prohibitif. D'autres méthodes ne considèrent pas la probabilité d'observer une valeur de pixel mais utilise le filtre de Kalman [163] pour faire une prédiction, l'avant-plan est ensuite détecté en comparant la prédiction avec l'observation. Un autre modèle, appelé *Wallflower* a également été proposé dans [143].

1.1.9 Discussion

Nous avons présenté dans cette partie les méthodes de soustraction de l'arrière-plan les plus utilisées dans l'état de l'art. Il en existe un nombre très important. Les différences entre les méthodes, en terme de complexité de calcul, d'espace mémoire utilisé et du nombre de paramètres à fixer empiriquement, sont très importantes. Cependant, il est difficile d'estimer les performances de ces méthodes en terme de qualité de l'avant-plan détecté. Afin de répondre à cette question, nous présentons dans le chapitre suivant une étude comparative de toutes les méthodes citées ci-dessus.

1.2 Suivi d'objets

Le suivi d'objets est une tâche fréquemment rencontrée en vision par ordinateur et la littérature sur le suivi de personnes est abondante. Plusieurs articles de référence sont disponibles [55, 8, 76, 159, 110]. Les objectifs du suivi d'objets sont de déterminer les trajectoires de ceux-ci dans le plan image et d'assigner à chaque objet de la scène une étiquette consistante dans le temps. Le suivi d'humains est une tâche difficile pour plusieurs raisons particulières :

- les personnes suivies peuvent avoir des mouvements complexes et difficilement prévisibles,
- le corps humain est très articulé,
- de nombreuses occultations peuvent survenir (de la personne par elle-même, par les autres objets en mouvement ou par des objets de l'arrière-plan),
- des changements d'illumination de la scène peuvent entraîner une non-consistance des valeurs des pixels représentant une personne,
- etc.

D'une manière générale, on peut trouver dans la littérature deux approches majeures pour réaliser le suivi. La première approche consiste à rechercher dans l'image courante l'objet présent sur les images précédentes [31]. Il se pose alors le problème de l'initialisation et de la terminaison du chemin. La deuxième approche consiste à détecter les objets sur l'image courante et à les associer avec les objets présents à l'instant précédent [68, 137]. Ces méthodes peuvent être également classées en fonction de leur environnement de travail (station de métro [135], environnement intérieur [136] ou extérieur [82], stade [80] etc.), de leur système de vision (monoculaire ou stéréovision [139]), du spectre utilisé par la caméra (visible ou infrarouge [158])... Néanmoins, ces méthodes se décomposent toutes en deux parties : le choix d'une représentation des personnes suivies et une méthode de suivi.

1.2.1 Les différentes représentations

Les représentations possibles sont nombreuses, nous pouvons citer par exemple :

- le centroïde de la personne [38],
- un ensemble de points [107, 144],

- des primitives géométriques plus ou moins complexes, par exemple une simple boîte englobante,
- la silhouette ou le contour de la personne [91, 157],
- un modèle articulé [162] ; dans cette représentation, plusieurs parties du corps sont modélisées par une primitive géométrique (par exemple une ellipse) et sont reliées entre elles par des connexions,
- un squelette [67],
- une représentation statistique avec des densités de probabilité, par exemple utilisation d'un histogramme des couleurs [122, 167],
- des descripteurs basés sur l'avant-plan, par exemple les moments de Zernike [78],
- le mouvement, avec par exemple l'utilisation du flot optique [41].

Le choix de la représentation est crucial car les modèles de déplacement qu'il sera possible d'utiliser en dépendront. Les informations récupérées à partir du déplacement d'un centroïde ou des déplacements des parties d'un modèle articulé n'auront pas la même importance. Nous présentons dans la figure 1.3 quelques illustrations des représentations utilisées pour le suivi de personnes.

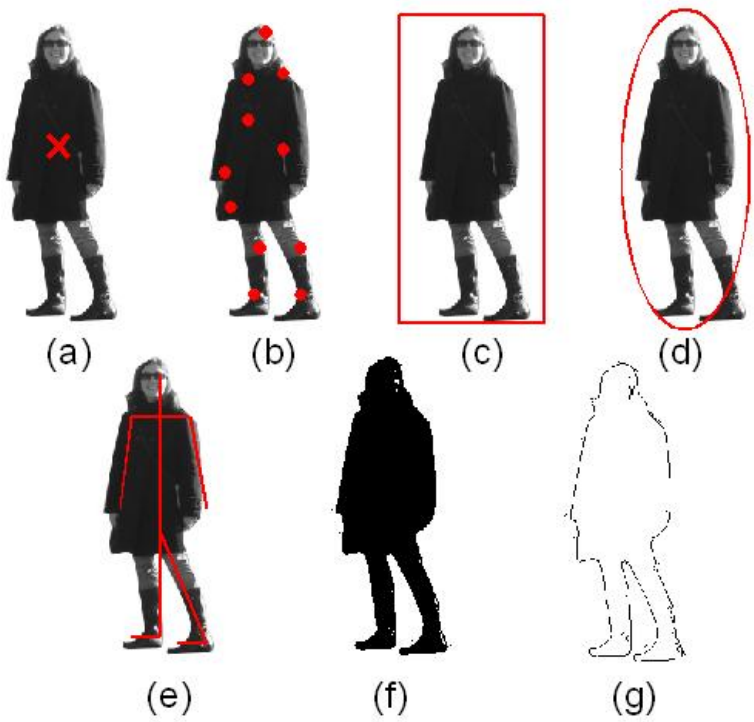


Figure 1.3: Illustration de quelques représentations utilisées pour le suivi (a) le centroïde (b) un ensemble de points caractéristiques (c) une boîte englobante (d) une ellipse (e) un squelette (f) un masque (ou silhouette) (g) un contour.

1.2.2 Les méthodes de suivi

Bien entendu, les méthodes de suivi dépendent directement de la représentation de la personne. Pour les méthodes où la représentation choisie est un ensemble de points, la première façon de faire la correspondance entre les points sur l'image à l'instant t et les points présents sur l'image à l'instant $t-1$ est de formuler le problème comme étant un problème d'optimisation [126, 147, 130]. On recherche alors pour chaque point son correspondant optimal. Les contraintes utilisées peuvent être la distance, la régularité de la vitesse, la rigidité (les mouvements des points proches doivent être similaires), l'homogénéité des mouvements des points d'un même objet etc. Il est également possible de faire la correspondance de manière statistique (et non plus déterministe comme expliqué ci-dessus). La méthode qui est la plus utilisée est sans conteste le filtre de Kalman [100, 81, 62, 49]. Partant de l'hypothèse que le bruit de mesure suit une distribution gaussienne, le filtre de Kalman est un algorithme récursif en deux étapes, la prédiction et la correction. Des versions modifiées comme le filtre de Kalman étendu [115] ont également été proposées dans la littérature et permettent de prendre en compte des modèles non-linéaires. Il a récemment été proposé une méthode basée sur les estimateurs à horizon glissant dont la particularité est de prendre en compte une fenêtre temporelle pour l'estimation et également de pouvoir gérer les occultations comme étant des contraintes visuelles [20]. Le filtre à particules [87, 22] permet lui de considérer les cas où les variables ne suivent pas une distribution normale.

Si la représentation utilisée est basée sur la forme ou sur des densités de probabilité, la méthode la plus simple, mais aussi la plus coûteuse en terme de calcul est le simple *template matching* [98] en calculant par exemple l'inter-corrélation. On peut également dans ce cas utiliser la méthode du *mean shift* [31], cette méthode présentant l'avantage par rapport au *template matching* d'optimiser la phase de recherche du correspondant optimal. En se basant sur le même raisonnement, si on utilise une représentation du contour, il est possible de réaliser une correspondance simple entre les contours en calculant par exemple la distance de Hausdorff [9]. Ensuite, Shi *et al.* [132] présentent une méthode basée sur le flot optique. Se basant sur des contraintes d'absence de variation de la luminance, ils calculent le vecteur déplacement d'un pixel ou l'étendent à une région de pixels.

1.3 Reconnaissance d'humains

L'objectif de la reconnaissance est de déterminer la nature d'un objet particulier. La reconnaissance d'humains en est simplement un cas particulier. Elle est basée sur l'utilisation combinée, classique en reconnaissance de formes, de descripteurs et d'une méthode de classification. L'idée est de coder l'information d'une image en un vecteur de descripteurs et d'utiliser les techniques d'apprentissage pour la classification. Nous présentons ici les différentes représentations et les différentes techniques de classification communément utilisées dans le cas de la reconnaissance d'humains.

1.3.1 Les différentes représentations

La première étape du processus de reconnaissance est donc de choisir une représentation des humains qui les caractérise clairement. Le choix de la représentation est très délicat. En effet, la représentation choisie doit permettre aisément la généralisation à toutes les instances de la classe "humain" mais doit être suffisamment discriminante pour permettre la séparation entre les exemples de la classe "humain" et les exemples négatifs. La représentation doit être peu sensible aux variations entre les individus, aux variations dues aux vêtements et aux mouvements des articulations. En d'autres termes, la représentation choisie doit apporter une haute variabilité inter-classes et une basse variabilité intra-classe. Pour ces raisons, au lieu d'utiliser directement l'intensité de chaque pixel de l'image, il est plus judicieux et efficace d'utiliser une représentation plus avancée. Cette représentation peut être globale [56] ou locale [101, 120, 34]. Nous présentons dans cette partie les représentations utilisées pour la détection de la présence humaine.

1.3.1.1 Représentation globale

Une première approche rencontrée pour la détection d'objets dans une image est une approche globale. Cette méthode, beaucoup utilisée en reconnaissance de formes, est difficile à appliquer dans le cas de la détection d'humains, principalement parce que le corps est hautement articulé et que la forme du corps humain varie au cours du temps.

Cependant, on peut trouver dans la littérature quelques exemples de méthodes qui utilisent une représentation globale. Par exemple, Gavrila *et al.* [56] ont proposé un système de détection de piétons, appelé PROTECTOR, basé sur l'extraction des contours (cf. figure 1.4), la distance de Chamfer est utilisée pour mesurer la similarité avec des exemples préalablement appris. Ce système a été utilisé avec succès en temps-réel.

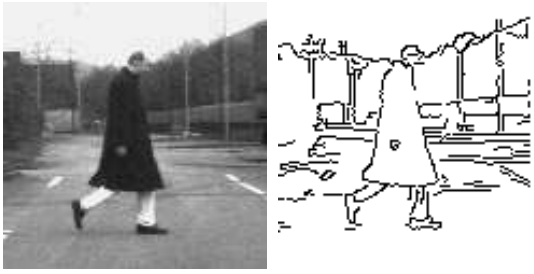


Figure 1.4: Exemples d'images utilisées pour le système PROTECTOR (image originale à gauche, contours à droite). Images tirées de [57].

Un autre exemple est le système de vidéo-surveillance VASM [29] qui utilise la dispersion, l'aire des objets détectés et le rapport hauteur/largeur de la boîte englobante pour la classification. De manière générale, les approches globales ont des difficultés à gérer les occultations.

1.3.1.2 Représentation locale

Les représentations locales sont moins sensibles aux occultations puisque seulement une partie des descripteurs est affectée.

Points d'intérêt Une première approche possible est d'utiliser une représentation locale et clairsemée de l'image basée sur des descripteurs locaux. L'image est décrite par un ensemble de descripteurs calculés au voisinage de points saillants (ou points d'intérêt). Le détecteur final se base donc sur l'ensemble de ces vecteurs descripteurs. Les points d'intérêt doivent décrire de façon stable les régions de l'image où l'information est importante. La performance du détecteur dépend de la répétabilité de ces points d'intérêt et de l'importance de l'information contenue dans la région détectée.

Les points d'intérêt généralement utilisés sont ceux proposés par Förstner et Harris [52, 53, 69] ou les Différences de Gaussiennes [102]. Pour le calcul du vecteur descripteur au voisinage des points détectés, il existe de nombreuses méthodes. Nous pouvons citer *Scale Invariant Feature Transformation* (SIFT) [102, 101], *Shape Contexts* [12], *Speed Up Robust Features* (SURF) [11] ou les moments invariants tel que les moments de Hu [75, 140], la transformée de Fourier-Mellin [61] ou les moments de Zernike [27, 90, 26]. Le descripteur SIFT est le plus utilisé actuellement. Il est constitué d'un histogramme des orientations des gradients contenus dans un voisinage de chaque point. Sur les images de la figure 1.5, nous avons extrait et affiché les points d'intérêt associés aux descripteurs SIFT avec l'orientation de leurs gradients.

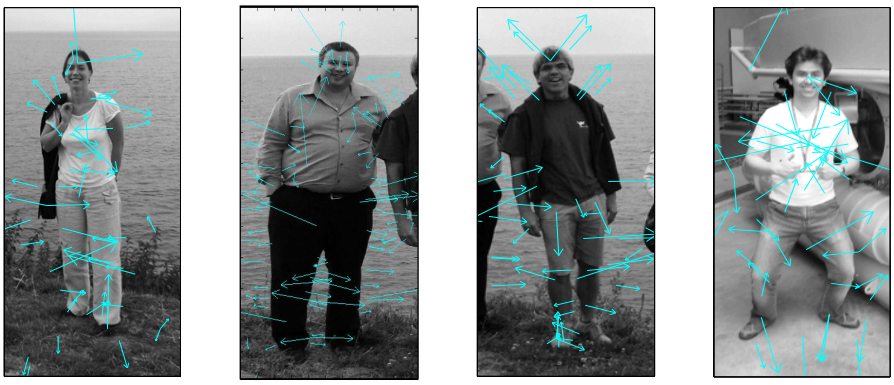


Figure 1.5: Extraction des points d'intérêt (associés aux descripteurs SIFT) sur quatres images différentes contenant des humains.

Filtres de Haar Une autre approche très populaire en détection d'objets est l'utilisation d'une représentation locale et dense basée sur les ondelettes de Haar (appelées aussi "les filtres rectangles" ou "haar-like filters"). Oren *et al.* [117], Papageorgiou et Poggio [120] ont été les premiers à proposer un système de détection d'objets basé sur les filtres de Haar, appliqué au cas de la détection d'humains, de voitures et

de visages. Ils utilisent les coefficients des filtres de Haar à différentes orientations et échelles comme descripteur local (cf. figure 1.6). L'objet à détecter est donc décrit par un dictionnaire sur-complet de différence en intensité de régions adjacentes. Mohan *et al.* [111] étendent ce système en apprenant plusieurs composantes (bras, jambes, têtes), des contraintes géométriques sont utilisées pour valider la sortie du classifieur.

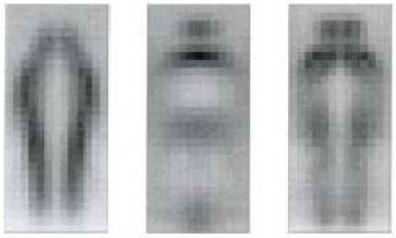


Figure 1.6: Moyenne des coefficients des filtres de Haar (de gauche à droite : filtre vertical, horizontal et diagonal). Les images sont tirées de [120].

Viola et Jones [148] utilisent également les filtres de Haar pour la détection d'objets. Leur système a été développé initialement pour la détection de visages puis étendu à la détection d'humains [149, 85].

Histogrammes de gradients orientés Dalal et Triggs [34, 35] ont proposé des descripteurs directement inspirés de *SIFT*, les histogrammes de gradients orientés (HOG). Ils ont utilisé avec succès ces descripteurs pour la détection d'humains dans des images puis dans des vidéos en y ajoutant des informations temporelles. Les descripteurs sont calculés sur une grille dense et uniformément répartie, l'information locale de la forme est encodée en calculant les histogrammes de gradients orientés. Une étude approfondie a été présentée dans [33] avec plusieurs variantes des histogrammes de gradients orientés.

Autres méthodes Si dans la littérature, les histogrammes des gradients orientés [34] et les représentations basées sur les filtres de Haar [120] sont les deux représentations les plus utilisées pour la détection d'humains, nous pouvons cependant citer d'autres méthodes de l'état de l'art. Par exemple, Kuno *et al.* [94] utilisent un histogramme de la silhouette calculé sur l'image de l'avant-plan. Abramson [6] utilise les points de contrôle qui sont une version modifiée des filtres de Haar. Utsumi et Tetsutani [146] utilisent une représentation basée sur une analyse statistique de la distance de Mahalanobis entre différentes parties de l'image. Wu et Nevatia [155] utilisent une représentation basée sur les *Edgelets*. Certains auteurs utilisent directement le mouvement pour la caractérisation. Par exemple, Sidenbladh [134] utilise le flot optique.

1.3.1.3 Représentation par composantes

On a vu précédemment des représentations holistiques où l'objet est décrit avec un ensemble de représentations locales ou avec une description globale. Dans tous

les exemples présentés ci-dessus, le corps humain entier est recherché dans l'image. Il a été présenté dans la littérature des systèmes de détection d'humains recherchant non pas le corps en entier mais recherchant certaines parties du corps humain (par exemple la tête, les bras, le torse ou les jambes). L'algorithme de détection recherche chacune de ces composantes indépendamment puis fusionne les résultats de détection [109].

Par exemple, Wu et Nevatia [155], Mikolajczyk *et al.* [109], Papageorgiou *et al.* [111], Zhao *et al.* [162] et Felzenszwalb *et al.* [50] utilisent une représentation en plusieurs parties du corps humain pour augmenter la robustesse du système global et gérer les occultations partielles.

1.3.2 La classification

Il existe principalement deux types d'apprentissage statistique : supervisé ou non-supervisé. Lors d'un apprentissage supervisé, on fournit à l'algorithme un ensemble de données avec la classe associée à chacune d'entre elles. Pour un apprentissage non-supervisé, l'algorithme opère des regroupements en fonction des ressemblances entre les exemples et forme les classes. Dans le cas de la détection de personnes, les algorithmes utilisent majoritairement un apprentissage supervisé. Si on dispose d'un grand nombre d'exemples d'images positives et négatives, l'algorithme d'apprentissage cherche une fonction de décision capable de séparer les exemples positifs et les exemples négatifs dans l'espace des descripteurs (ou espace des attributs). Par la suite, nous présentons les méthodes d'apprentissage couramment utilisées dans la détection de la présence humaine.

1.3.2.1 Machines à vecteurs de support

C'est en 1979 que Vladimir Vapnik introduit le Séparateur à Vastes Marges (SVM). Cependant, c'est depuis le début des années 90 que le SVM est beaucoup utilisé par la communauté scientifique. L'acronyme SVM remplace *Support Vectors Machine* (Machine à Vecteurs de Support) ou Séparateur à Vastes Marges. Dans ce cas, la classification est faite en recherchant un hyper-plan optimal qui maximise les marges entre les exemples de la classe positive et de la classe négative. La séparation entre les classes est réalisée dans un espace de redescription des données de très grande dimension.

Papageorgiou *et al.* [120] ont utilisé avec succès les SVM dans le cas de la détection d'humains. Dans les premières versions de leur système de détection, ils utilisent les SVM avec une représentation des humains basée sur les filtres de Haar. Dans les versions suivantes, ils utilisent plusieurs SVM, un pour chaque partie du corps humain [111] et ils combinent ensuite les résultats de chaque classifieur avec des contraintes géométriques. Dalal et Triggs [34] utilisent aussi un SVM linéaire pour détecter la présence humaine en combinaison avec les histogrammes des gradients orientés. Sidenbladh [134] utilise les SVM pour apprendre un modèle de la marche humaine réalisé avec le flot optique. Yoon et Kim [161] utilisent aussi les SVM en combinaison avec une carte des distances de Mahalanobis.

1.3.2.2 Adaboost

Adaboost (pour *Adaptive Boosting*) est la méthode la plus utilisée en boosting. Le principe du boosting est de combiner les résultats de classifieurs "faibles" pour obtenir un classifieur plus efficace (ou classifieur boosté). Un classifieur faible est un classifieur dont la probabilité de donner un bon résultat est un peu plus importante que 1/2 , autrement dit un peu meilleur que le hasard.

Viola et Jones [148] ont été les premiers à utiliser Adaboost pour apprendre une cascade de classifieurs boostés basés sur les coefficients des filtres de Haar. Ils ont illustré leurs travaux avec la détection de visages dans [148] puis Viola *et al.* [149] ont étendu le classifieur avec des informations temporelles pour la détection d'humains. Abramson [6] utilise également Adaboost pour la détection de piétons avec les points de contrôle comme classifieurs faibles. Récemment, Zhu *et al.* [165] ont utilisé les histogrammes de gradients orientés dans une cascade de classifieurs boostés. Même si l'apprentissage avec Adaboost est très lent [28], l'architecture en cascade du détecteur permet de rejeter beaucoup d'images négatives dés les premiers étages de la cascade et augmente donc logiquement la rapidité du système.

1.3.2.3 On-line boosting

Pour utiliser les techniques classiques d'apprentissage citées ci-dessus, il faut construire une très grande base d'images. Cette étape peut être laborieuse et demander beaucoup de temps. La qualité de cette base est primordiale et l'objectif est très difficile puisqu'il faut être capable de généraliser avec un nombre d'images fini à tous les exemples possibles sur tous les arrière-plans possibles. Or, dans le contexte de la vidéo-surveillance ou de la domotique, la caméra est fixe. Cette tâche peut être simplifiée si l'apprentissage est réalisé en utilisant les conditions d'utilisation du système et le classifieur adapté à un scénario particulier. Plusieurs systèmes ont récemment été présentés n'utilisant qu'une petite base d'apprentissage et ensuite, le modèle est mis à jour et se spécialise pour répondre à une tâche précise [64, 124, 24, 156].

1.3.2.4 Autres méthodes

Nous n'avons présenté ici que les deux méthodes les plus utilisées dans le cas de la détection d'humains mais nous pouvons en citer quelques autres. Certains auteurs utilisent les réseaux de neurones pour la classification. Après une estimation de l'avant-plan, Collins *et al.* [29] utilisent des descripteurs de formes simples, la dispersion de la silhouette ($\frac{perimetre^2}{aire}$), son aire et le rapport de la boîte englobante de la silhouette, comme entrées d'un réseau de neurones. Branca *et al.* [17] utilisent aussi un réseau de neurones à trois couches, en rétro-propagation. Fahmy *et al.* [48] utilisent un modèle de Markov caché pour la détection d'humains. Ullman *et al.* [145] et Wu et Nevatia [154] utilisent une classification Bayesienne. Utsumi et Tetsutani [146] et Ghidary *et al.* [60] utilisent un simple modèle statistique (moyenne, variance) pour la classification. Gall et Lempitsky [54] utilisent les *Random Forest*, composés de plusieurs arbres de décision, pour la détection de personnes.

1.4 Reconnaissance d'activités humaines

La reconnaissance d'activités humaines est une autre étape très étudiée ces dernières années par la communauté scientifique en vision par ordinateur. Des articles de référence [55, 8, 76] décrivent avec soin les méthodes utilisées dans l'état de l'art. La reconnaissance d'activités consiste, à partir d'informations bas-niveau comme la valeur numérique d'un ensemble de pixels, à obtenir une représentation sémantique, en langage naturel de la scène. Le processus de la reconnaissance d'activité peut donc être considéré comme un problème de classification où interviennent les différentes représentations des activités et les techniques de reconnaissance. C'est un problème très complexe. Les difficultés sont nombreuses, nous pouvons en citer quelques unes ici :

- Il y a une grande variabilité intra-classe. Une même action réalisée par la même personne à deux moments différents, même avec des conditions identiques, peut être légèrement différente. Deux personnes peuvent réaliser la même action différemment ou le temps de réalisation peut être différent.
- Il se pose souvent le problème de déterminer le début et la fin de l'action pour effectuer la correspondance entre l'action observée et les modèles.
- Une action ou un scénario à reconnaître est une combinaison d'actions atomiques (courir, tomber etc.) qui peuvent être détectées indépendamment les unes des autres, il faut donc ensuite analyser leur répartition temporelle pour remonter au niveau de scénario ou d'activité.

Plusieurs hypothèses simplificatrices et méthodes ont été proposées dans la littérature pour résoudre le problème du changement de point de vue. Tout d'abord, certaines méthodes utilisent plusieurs caméras [151, 89]. Cette hypothèse simplifie grandement le problème mais n'est pas utilisable dans beaucoup de situations où seulement un système monoculaire est utilisé [104]. Avec une seule caméra, certains auteurs ont proposé l'utilisation d'un modèle de la caméra (modèle affine [128] ou projectif [121] par exemple). D'autres auteurs se sont basés sur la géométrie épipolaire entre une même pose vue sous deux angles différents [65, 160]. Par exemple, Rao *et al.* [127] et Syeda-Mahmood *et al.* [138] calculent la correspondance entre deux actions en évaluant la matrice fondamentale avec quelques points sélectionnés.

Les représentations utilisées peuvent être formulées de manière explicite ou déterminées par l'apprentissage. Certaines méthodes utilisent les propriétés physiques de l'action comme par exemple son caractère périodique [71]. Beaucoup de méthodes utilisent directement le mouvement [43, 32] ou se basent sur les images de l'avant-plan pour construire par exemple les *Motion Energy Images* (MEI), les *Motion History Images* (MHI) [40] (cf. figure 1.7) ou plus récemment les *Motion History Volumes* (MHV) [151].

Les méthodes utilisées pour la modélisation sont également nombreuses. Nous pouvons tout d'abord citer les classifieurs bayesiens [74, 105] qui sont bien adaptés pour modéliser les incertitudes avec des probabilités liées à la reconnaissance d'activité mais ne sont pas adaptés pour prendre en compte les relations temporelles. Les modèles de Markov cachés (ou *Hidden Markov Models* - HMM) sont certainement

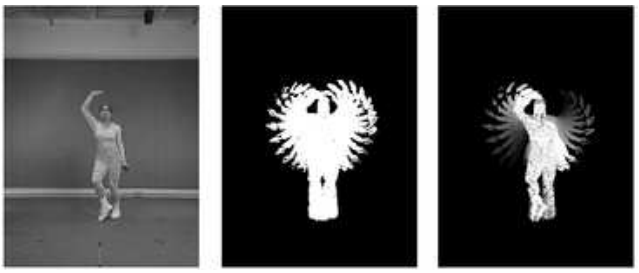


Figure 1.7: Exemple de représentation utilisée pour la reconnaissance d'activité. De gauche à droite : image originale, MEI et MHI. Les images sont tirées de [13].

les plus couramment utilisés dans l'état de l'art pour la reconnaissance d'activités [18, 21, 150]. Contrairement aux classifieurs bayesiens, un HMM permet de modéliser les incertitudes des relations temporelles des actions. Les réseaux de Petri ont également été utilisés pour reconnaître des scénarios prédéfinis [59]. Ils permettent aisément de paralléliser et de séquencer les éléments d'un scénario.

1.5 Conclusion

Nous avons présenté dans ce chapitre les méthodes généralement utilisées dans l'état de l'art pour détecter des humains dans une séquence d'images en portant particulièrement notre attention sur la soustraction de l'arrière-plan, le suivi d'objets mobiles et la classification. Nous avons ensuite présenté les méthodes de reconnaissance d'activités.

Un système de vision est donc un ensemble de sous-systèmes pouvant avoir des interactions plus ou moins complexes les uns avec les autres. Le choix de chaque sous-système a un impact fort sur la performance du système global et cette interdépendance entre les sous-systèmes entraîne le risque d'observer une erreur se propager aux autres éléments du système. De plus, le système global ayant une capacité de calcul et de mémoire finie, un sous-système ne doit pas utiliser toutes les ressources matérielles au détriment des autres composants. Le choix des méthodes les plus adaptées au projet *CAPTHOM* est délicat puisqu'il s'agit de trouver un compromis optimal entre complexité algorithmique et performances.

S'il est possible d'estimer *a priori* la complexité algorithmique de ces méthodes, il est assez difficile d'estimer leurs performances dans les différentes conditions d'usage de *CAPTHOM*. Nous présentons dans le chapitre suivant, une étude comparative de méthodes de soustraction de l'arrière-plan et de méthodes de classification. Ces résultats nous permettrons de définir les méthodes les plus adaptées aux contraintes du projet *CAPTHOM*.

CHAPITRE 2

Étude comparative de modules de vidéo-surveillance

Nous avons présenté dans le chapitre précédent les différentes étapes généralement mises en place pour l'analyse de vidéos (détection de personnes et reconnaissance d'activités). Nous présentons dans ce chapitre les résultats de deux études concernant deux étapes clés présentées dans le chapitre 1, à savoir la soustraction de l'arrière-plan et la reconnaissance d'humains. Ces deux étapes sont très importantes pour notre application puisque la première permettra de satisfaire les contraintes matérielles et d'atteindre les performances attendues tandis que la seconde apportera une rupture technologique par rapport aux capteurs du marché qui ne sont pas capables de différencier un humain d'un autre objet mobile.

Nous présentons dans ce chapitre les résultats d'une large étude comparative des performances des méthodes de soustraction de l'arrière-plan. Cette étude a été menée sur une base de vidéos réelles, synthétiques et semi-synthétiques. Sept méthodes, parmi les plus couramment utilisées et référencées dans la littérature sont évaluées. Les temps de calcul et l'espace mémoire sont également considérés.

Dans un deuxième temps, nous nous intéressons plus particulièrement à l'évaluation de la reconnaissance d'humains. Nous comparons tout d'abord les résultats de deux méthodes de détection de personnes. Ensuite, nous analysons dans le détail les résultats de détection obtenus avec une de ces deux méthodes selon plusieurs paramètres (taille de la base d'apprentissage, variante de la méthode d'apprentissage, spectre utilisé etc.*).*

2.1 Algorithmes de soustraction de l'arrière-plan

Une application de vision par ordinateur est souvent initialisée par le résultat de la détection de changement ou plus particulièrement par le résultat de la soustraction de l'arrière-plan. Cette étape fondamentale sera utilisée ultérieurement par le suivi et/ou par la reconnaissance. Le nombre de méthodes présentes dans la littérature est très important, chacune présentant des caractéristiques différentes. Si nous pouvons déduire certains avantages et inconvénients de ces méthodes en analysant leur structure algorithmique, nous avons besoin de plus de résultats pour quantifier leurs performances. Nous présentons donc ici une étude comparative des méthodes de soustraction de l'arrière-plan.

L'évaluation a été réalisée sur des vidéos réelles, semi-synthétiques et synthétiques. Les objectifs de l'étude sont les suivants :

1. évaluer les bénéfices apportés par l'utilisation de méthodes complexes par rapport aux plus simples,
2. comparer les temps de calcul et l'espace mémoire nécessaires de chaque méthode,
3. déterminer les conditions d'utilisation favorables de chaque méthode,
4. comparer les méthodes de post-traitement.

La soustraction de l'arrière-plan étant une technique très utilisée en vision par ordinateur, il est possible de trouver dans la littérature quelques études référençant ou évaluant ces techniques. Nous pouvons citer par exemple Toyama *et al.* [143] qui ont été, à notre connaissance, les premiers à présenter une étude comparative des algorithmes de soustraction de l'arrière-plan. Leur évaluation est basée sur l'étude du nombre de faux négatifs (FN) et de faux positifs (FP). Cependant, ces deux valeurs étant dépendantes l'une de l'autre, lorsque le nombre de faux négatifs diminue, le nombre de faux positifs augmente en retour. Puisqu'ils ne présentent qu'une valeur du couplet $\{FN, FP\}$, ces résultats sont difficiles à interpréter. Chalidabhongse *et al.* [23] ont proposé une méthode pour évaluer les algorithmes de soustraction de l'arrière-plan basée sur l'analyse de perturbations. Dans leur étude, le taux de fausses détections est fixé sur les séquences d'apprentissage. Ensuite, l'arrière-plan de ces mêmes séquences est modifié par un vecteur de perturbation dans toutes les directions de l'espace RGB simulant un avant-plan. Le taux de détection est affiché en fonction de la magnitude de ce vecteur exprimant ainsi la sensibilité des algorithmes. Le principal avantage de cette méthode d'évaluation est que la vérité terrain de l'avant-plan n'est pas nécessaire. Cependant, dans le cas d'arrière-plan multimodaux, l'avant-plan simulé sera une combinaison de la distribution multimodale et de la perturbation alors que les avant-plans sont généralement unimodaux. On notera également que cette méthode ne permet pas d'évaluer les méthodes qui ne sont pas basées sur l'analyse indépendante de chaque pixel et qu'elle ne permet pas l'étude des méthodes de post-traitement. Les auteurs comparent quatre méthodes dans cet article. Plus récemment, Panahi *et al.* [119] ont présenté une étude comparant 6 méthodes mais ces méthodes sont comparées en ce basant seulement sur une valeur du couplet $\{FN, FP\}$.

Ces études ne nous permettent pas d'avoir suffisamment d'informations pour la sélection d'une méthode. Nous avons donc choisi de mener une nouvelle étude comparative. Celle-ci devra prendre en compte l'inter-dépendance entre le nombre de faux négatifs et faux positifs, cette étude devra utiliser suffisamment de vidéos représentant des difficultés variées et enfin la complexité algorithmique devra également être considérée.

2.1.1 Protocole de l'étude

2.1.1.1 Principe

Le principe de l'étude comparative est illustré dans la figure 2.1. Pour chaque vidéo, nous disposons de la vérité terrain correspondant au masque de l'avant-plan ou aux boîtes englobantes de chaque objet de l'avant-plan. Nous appliquons les algorithmes de soustraction de l'arrière-plan sur les vidéos et nous comparons les résultats obtenus avec la vérité terrain en comptant le nombre de pixels qui ont été correctement labellisés comme appartenant à l'avant-plan.

Les algorithmes sont comparés en se basant sur la courbe Précision/Rappel définie par :

$$Précision = \frac{\#VraisPositifs}{\#VraisPositifs + \#FauxPositifs}, \tag{2.1}$$

$$Rappel = \frac{\#VraisPositifs}{\#VraisPositifs + \#FauxNégatifs}, \tag{2.2}$$

Par définition, le meilleur algorithme est celui qui présente un faible nombre de faux positifs et de faux négatifs, donc conjointement une grande valeur de Précision et de Rappel.

2.1.1.2 Méthodes sélectionnées

Le choix des méthodes évaluées est très délicat. Il est clairement impossible de réaliser une étude de toutes les méthodes présentes dans la littérature. Nous avons donc choisi d'évaluer les méthodes qui nous paraissent être les méthodes les plus utilisées (*e.g.* [153]), les plus citées (*e.g.* [137]) ou qui présentent une approche originale (*e.g.* [116] ou [92]).

Ensuite, il est possible de trouver pour chaque méthode plusieurs variantes ou optimisations. Par exemple, la méthode de Stauffer et Grimson [137] a fait l'objet de nombreux articles proposant des optimisations (*e.g* [95, 166, 88, 70]). Nous avons fait le choix d'évaluer les méthodes originales et non les variantes. La table de codage est un cas particulier. En effet, cette méthode utilise une base de couleur particulière afin de séparer l'évaluation de la chrominance et de la luminance. Ici, nous avons fait

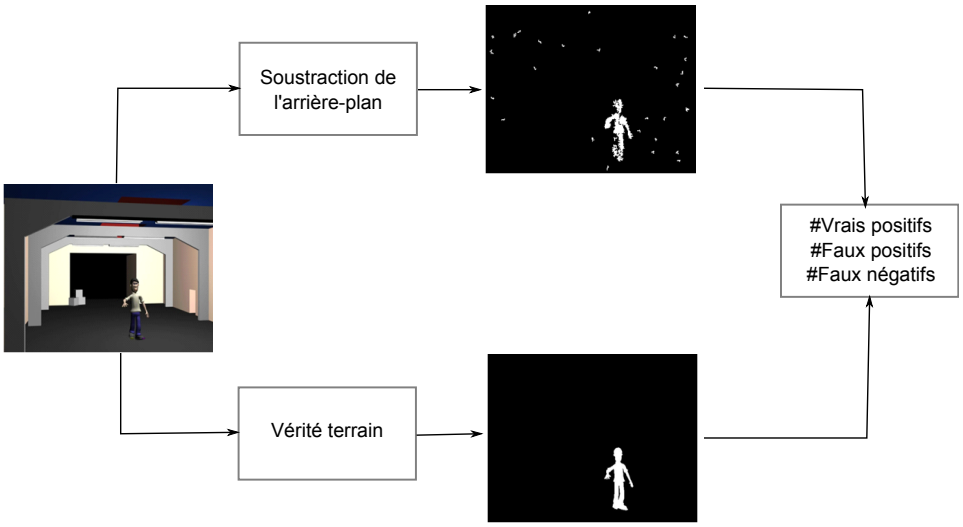


Figure 2.1: Principe de l'évaluation supervisée des algorithmes de soustraction de l'arrière-plan.

le choix de modifier la méthode originale afin d'évaluer l'utilisation de la table de codage et non pas d'évaluer la table de codage avec un espace de couleur particulier. Plusieurs études ont déjà été réalisées pour évaluer l'influence des bases de couleur sur la qualité de la soustraction de l'arrière-plan [93, 131]. Dans la méthode originale de Kim *et al.* [92], une valeur clé est composée du vecteur moyenne $\boldsymbol{\mu}$ et de 6 autres valeurs numériques et l'évaluation de la différence de chrominance et de luminance est réalisée séparément afin de distinguer les objets de l'avant-plan et leurs ombres. Nous avons simplifié le modèle, chaque valeur clé est alors composée des paramètres d'une distribution gaussienne. Soit N le nombre d'images dans la séquence d'apprentissage et $C_s = \{c_{1,s}, \ldots, c_{L,s}\}$, une table de codage associée au pixel s, composée de L valeurs clés. Chaque valeur clé $c_{i,s}$ est une gaussienne définie par une moyenne $\boldsymbol{\mu}_{i,s}$ et une matrice de covariance $\boldsymbol{\Sigma}_{i,s}$ (nous utilisons une matrice de covariance diagonale). Les paramètres des gaussiennes et le nombre de valeurs clés par pixel sont estimés lors de l'apprentissage. Pendant cette phase, la table de codage de chaque pixel est tout d'abord initialisée avec sa couleur à l'instant $t = 0$, c'est-à-dire $\boldsymbol{\mu}_{1,s} = \boldsymbol{I}_{s,0}$ et $\boldsymbol{\Sigma}_{1,s} = \sigma_0^2.\mathrm{Id}$, où σ_0^2 est une constante choisie empiriquement et Id est la matrice identité. Ensuite, chaque nouvelle couleur $\boldsymbol{I}_{s,t}$ est comparée avec les valeurs clés $c_{i,s}$ grâce à la distance de Mahalanobis. Si une valeur clé correspond, celle-ci est mise à jour. Si aucune valeur clé j ne correspond, une nouvelle valeur clé est créée et initialisée avec $\boldsymbol{\mu}_{j,s} = \boldsymbol{I}_{s,t}$ et $\boldsymbol{\Sigma}_{j,s} = \sigma_0^2.\mathrm{Id}$. Pendant la phase de détection, chaque pixel $\boldsymbol{I}_{s,t}$ est classifié avec :

$$\mathcal{X}_t(s) = \begin{cases} 1 & \text{si } d_M(\boldsymbol{I}_{s,t}, c_{i,s}) > \tau \ \ \forall i, \\ 0 & \text{sinon.} \end{cases} \tag{2.3}$$

La liste des méthodes utilisées et les abréviations associées est présentée dans le tableau 2.1.

Abréviation	Description
Basic	Modélisation avec un filtre moyenneur temporel, détaillée dans la section 1.1.1.
1-G	Modélisation gaussienne de l'arrière-plan, détaillée dans la section 1.1.2
GMM	Modélisation avec un mélange de gaussiennes, détaillée dans la section 1.1.3 (*GMM* pour *Gaussian Mixture Model*)
KDE	Modélisation non-paramétrique, détaillée dans la section 1.1.4 (*KDE* pour *Kernel Density Estimation*)
MinMax	Minimum, maximum et maximum de différence intertrames, détaillée dans la section 1.1.5
CB$_{RGB}$	Table de codage où chaque valeur clé est une gaussienne, détaillée dans la section 1.1.6 et 2.1.1.2 (CB$_{RGB}$ pour *Code-Book*)
Eigen	Réduction statistique de l'arrière-plan, détaillée dans la section 1.1.7 (*Eigen* pour *Eigen Backgrounds*)

Table 2.1: Abréviations utilisées dans cette étude comparative.

2.1.1.3 Base de vidéos

Pour évaluer quantitativement les performances des algorithmes décrits précédemment, ces méthodes ont été appliquées sur une large base de vidéos réelles, synthétiques et semi-synthétiques (*cf.* quelques exemples figure 2.2). Notre base de vidéos est composée de 29 vidéos (15 réelles, 10 semi-synthétiques et 4 synthétiques). Nous avons créé quelques vidéos synthétiques et semi-synthétiques, les autres sont extraites de la base de vidéos PETS2001 [123], de la base de vidéos d'IBM [19] et la compétition VSSN 2006 [2]. Les vidéos synthétiques ont été réalisées avec le logiciel *3DSMAX*. Les vidéos semi-synthétiques sont faites d'avant-plans synthétiques (personnes, voitures) se déplaçant devant un arrière-plan réel. L'ensemble de la base de vidéos contient des vidéos de scènes intérieures (20 vidéos) et extérieures (9 vidéos). De plus, 6 de ces 29 vidéos contiennent des arrière-plans en mouvement.

La vérité terrain des vidéos synthétiques et semi-synthétiques est facile à obtenir. Pour les vidéos réelles, la vérité terrain est disponible pour des images de référence, manuellement annotées ou fournie avec la base de vidéos.

Nous avons choisi de diviser la base de vidéos en plusieurs parties illustrant une caractéristique particulière :

- vidéos avec des arrière-plans statiques (15 vidéos),
- vidéos avec des arrière-plans multimodaux (6 vidéos),
- vidéos dégradées par un bruit gaussien (bruit blanc et additif) (15 vidéos).

Certaines séquences de la base de vidéos dégradées proviennent de la base de

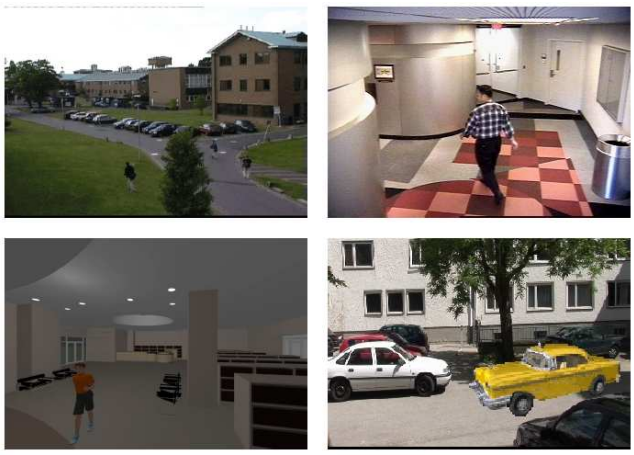


Figure 2.2: Exemples des vidéos utilisées pour l'évaluation supervisée.

vidéos avec arrière-plans statiques auxquelles nous avons dégradé la qualité. L'évaluation est réalisée sur chacune de ces bases. Ensuite, nous présentons les résultats obtenus sur l'ensemble de la base.

2.1.1.4 Paramètres utilisés

Nous présentons dans le tableau 2.2 les différents paramètres utilisés pour l'étude comparative. Les valeurs de ces paramètres ont été déterminées empiriquement. La variation du seuil τ de chaque méthode est utilisée pour obtenir les différents points de la courbe Précision/Rappel. Les algorithmes testés ont les origines suivantes : *KDE* a été fourni par les auteurs [44], *GMM* provient de la librairie *OpenCV* [16], le code de *Eigen* est disponible dans [1]. Nous avons codé les autres algorithmes.

Algorithme	**Paramètres**
Basic	distance d_2, $\alpha = 10^{-3}$
1-G	distance d_M, $\alpha = 10^{-3}$ la matrice de covariance est diagonale
GMM	$K = 3$, $\alpha = 10^{-2}$ la matrice de covariance est diagonale
KDE	$N = 100$
CB$_{\text{RGB}}$	distance d_M, $\alpha = 10^{-3}$ la matrice de covariance est diagonale
Eigen	$N = 100$, $M = 20$

Table 2.2: Paramètres utilisés pour l'étude comparative.

2.1.2 Résultats expérimentaux

Comme mentionné précédemment, les algorithmes ont été évalués sur différents groupes de vidéos. Nous présentons tout d'abord les résultats obtenus sur des vidéos avec arrière-plans statiques, puis sur des vidéos avec arrière-plans multimodaux et finalement sur des vidéos fortement bruitées. Nous présentons ensuite les résultats des méthodes de post-traitement, les résultats obtenus sur l'ensemble de la base puis les temps de calcul et la mémoire utilisée pour chaque méthode.

2.1.2.1 Base de vidéos avec arrière-plans statiques

Pour ce premier test, les méthodes de soustraction de l'arrière-plan sont évaluées dans des conditions idéales. Les vidéos utilisées présentent un grand rapport signal sur bruit, aucun bruit n'a été rajouté et les arrière-plans sont statiques. Nous utilisons ici 15 vidéos. Les résultats sont présentés dans la figure 2.3.

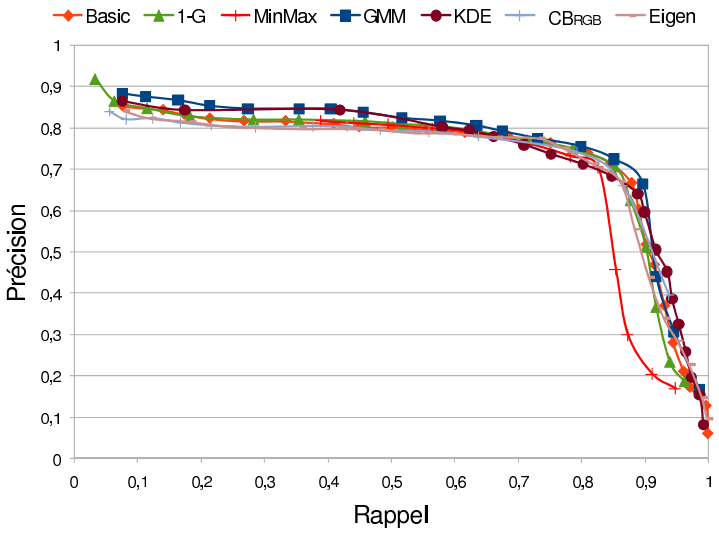
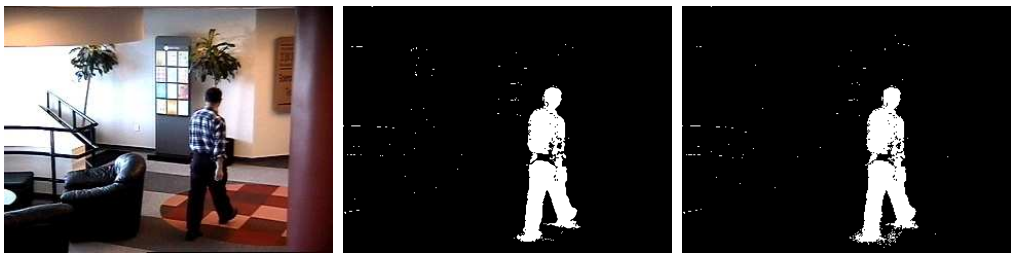


Figure 2.3: Courbes Précision/Rappel obtenues sur une base de 15 vidéos avec arrière-plans statiques.

Comme expliqué précédemment, le meilleur algorithme est celui qui présente conjointement une grande valeur de Précision et de Rappel. Donc plus une courbe se situe dans la partie supérieure du graphique, plus cette courbe présente des bons résultats d'un algorithme de soustraction de l'arrière-plan.

Deux conclusions principales peuvent être tirées de ces résultats. Tout d'abord, les résultats de *MinMax* sont légèrement inférieurs aux autres méthodes. Ceci peut s'expliquer par le fait que cette méthode travaille sur des images en niveaux de gris. Ensuite, toutes les autres méthodes ont des résultats homogènes. Ce résultat est très intéressant puisque la complexité de certaines méthodes (comme *GMM* ou *KDE*) ne permet pas, dans le cas de vidéos simples, d'obtenir de meilleurs résultats qu'une

modélisation simple de l'arrière-plan. Une illustration est présentée dans la figure 2.4 dans laquelle les résultats obtenus avec *Basic* et *GMM* sont visuellement équivalents.



(a) (b) (c)

Figure 2.4: Exemples d'avant-plans obtenus sur une vidéo avec arrière-plan simple. (a) image d'entrée (b) avant-plan obtenu avec *Basic* (c) avant-plan obtenu avec *GMM*.

2.1.2.2 Base de vidéos avec arrière-plans multimodaux

L'objectif de ce test est d'évaluer la robustesse des méthodes de soustraction de l'arrière-plan lorsqu'elles sont utilisées sur des vidéos avec des arrière-plans multimodaux. Nous utilisons ici 6 vidéos qui contiennent toutes des arbres agités par le vent. Les résultats sont présentés dans la figure 2.5.

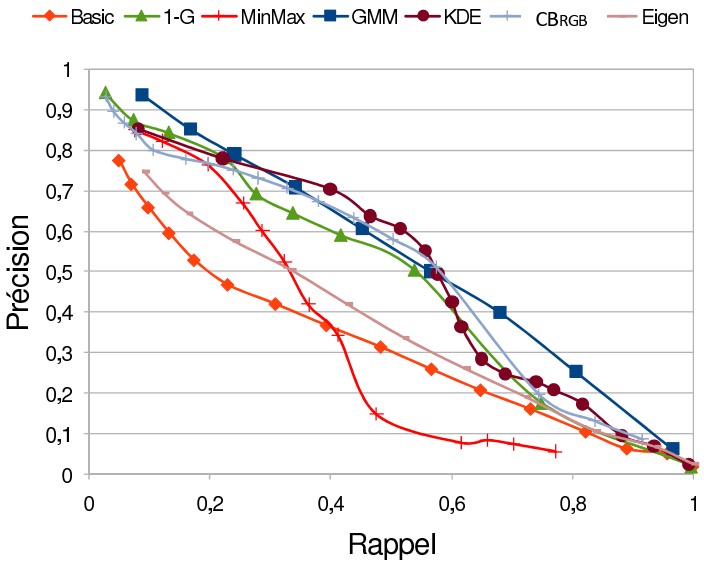


Figure 2.5: Courbes Précision/Rappel obtenues sur une base de 6 vidéos avec arrière-plans multimodaux.

Les résultats sont dans ce cas très hétérogènes. Tout d'abord, les résultats obtenus avec les méthodes simples *Basic* et *MinMax* sont fortement inférieurs à ceux obtenus

par les autres méthodes. Leur seuil global et non-adaptatif n'est pas adapté aux vidéos avec des textures animées. La méthode *Eigen* présente également des résultats médiocres. La méthode *1-G*, pourtant structurellement adaptée aux arrière-plans unimodaux, présente des performances étonnamment bonnes sur notre base de vidéos multimodales. Ceci peut s'expliquer par le fait que le seuil est localement pondéré par la matrice de covariance qui compense bien certaines instabilités. Ensuite, *KDE*, *GMM* et CB_{RGB} sont structurellement adaptés pour pouvoir gérer les arrière-plans multimodaux, leurs performances sont logiquement meilleures dans ce contexte.

Nous présentons dans la figure 2.6, des exemples d'avant-plans obtenus avec les différentes méthodes. Dans cette figure, on voit clairement la différence de performance entre *Basic*, *MinMax*, *Eigen* et les autres méthodes.

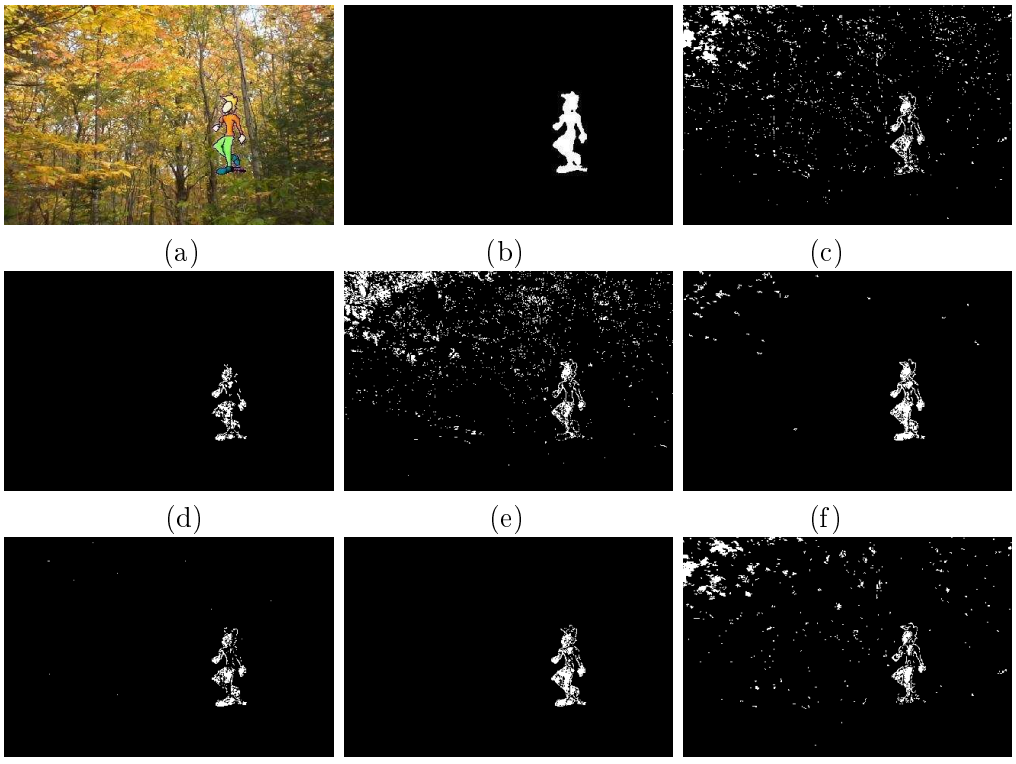


Figure 2.6: Avant-plans obtenus sur une vidéo avec un arrière-plan fortement multimodal (a) image originale (b) Vérité terrain (c) *Basic* (d) *1-G* (e) *MinMax* (f) *GMM* (g) *KDE* (h) CB_{RGB} (i) *Eigen*.

2.1.2.3 Base de vidéos dégradées

Avec ce troisième test, nous évaluons les performances des algorithmes de soustraction de l'arrière-plan sur une base de vidéos fortement bruitées. Nous utilisons 15 vidéos dont la qualité a été dégradée par un bruit gaussien (bruit additif et blanc). Les résultats sont présentés dans la figure 2.7.

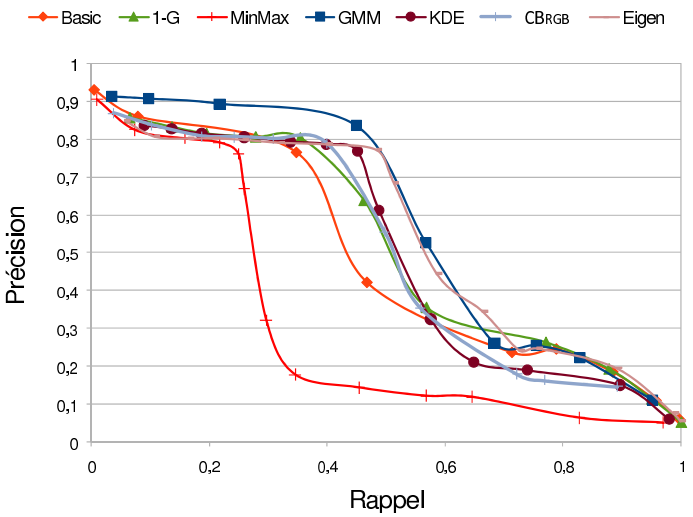


Figure 2.7: Courbes Précision/Rappel obtenues sur 15 vidéos fortement bruitées.

La méthode *MinMax* ne semble pas adaptée pour les vidéos bruitées. Son seuil global dépend du maximum de différence intertrame qui est grand pour des vidéos bruitées. La méthode *Basic* est aussi pénalisée par son seuil global. Les méthodes statistiques comme *1-G*, *GMM*, *KDE* ou CB_{RGB} donnent de meilleurs résultats, particulièrement *GMM*.

2.1.2.4 Influence de la distance d

Comme mentionné dans la section 1.1.1, il existe plusieurs distances possibles pour calculer l'écart entre un pixel de l'image courante et ce même pixel dans le modèle. Pour évaluer ces distances, nous les avons appliquées sur 12 vidéos présentant un bruit additif ou un arrière-plan multimodal. Les résultats sont présentés dans la figure 2.8.

Chaque composante de l'espace de couleur *RGB* contient des informations de luminance et de chrominance. En utilisant la distance d_0, on perd des informations de chrominance. Les performances sont donc moins bonnes avec cette distance. Les résultats obtenus avec d_1, d_2 et d_3 sont globalement homogènes. Les résultats obtenus avec d_M et d_G sont légèrement meilleurs, ceci peut s'expliquer par le fait que, dans ce cas, le calcul devient local et dépend de la quantité de bruit pour le pixel considéré.

2.1.2.5 Influence du post-traitement

Classiquement, les régions de l'avant-plan correspondent à des formes compactes et les faux positifs correspondent à des pixels isolés répartis sur l'image. Il est donc possible d'appliquer une étape de post-traitement après la soustraction de l'arrière-plan pour lisser l'avant-plan. On peut alors remplir les "trous" dans les objets détectés (diminuer les faux négatifs) et diminuer le bruit (diminuer les faux positifs). En

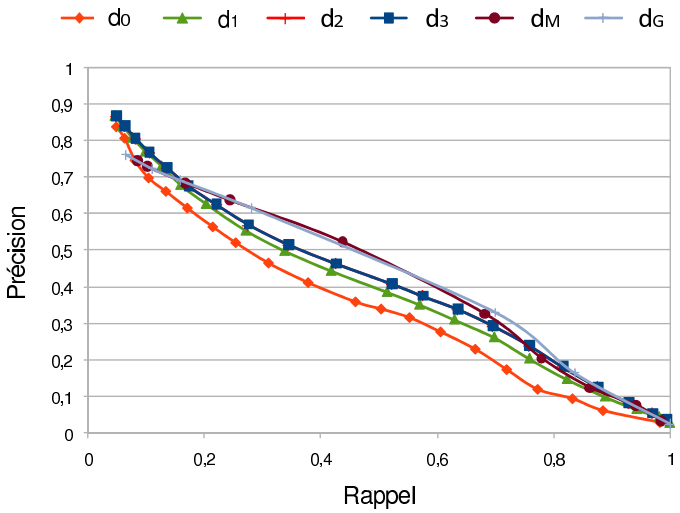


Figure 2.8: Évaluation des différentes distances d : courbes Précision/Rappel obtenues avec 12 vidéos avec bruit gaussien ou arrière-plans multimodaux.

pratique, la décision d'appartenance ou non d'un pixel à l'avant-plan est prise au regard de sa valeur numérique et aussi des labels des pixels de son voisinage. Dans cette étude, nous évaluons trois méthodes de post-traitement :

- le filtre médian [15],
- le filtre morphologique [15],
- les champs de Markov [5].

Lors des expérimentations, nous avons utilisé différentes tailles de masque pour le filtre médian et pour le filtre morphologique. Nous avons également utilisé différentes combinaisons d'érosion et de dilatation. Les résultats présentés ont été obtenus avec des filtres de taille 5×5 et les opérations morphologiques sont définies par : $fermeture(ouverture(\mathcal{X}, \mathcal{W}), \mathcal{W})$ où $\mathcal{X}$ est l'avant-plan et $\mathcal{W}$ le filtre morphologique utilisé. Enfin, nous utilisons les champs de Markov avec la formulation du potentiel de Ising au pixel s défini par :

$$L(\mathcal{X}(s), x) = \sum_{s' \in \eta_s} (1 - \delta(\mathcal{X}(s'), x)), \tag{2.4}$$

où $\delta(a, b)$ est une fonction à deux variables qui vaut 1 si celles-ci sont égales ou 0 sinon, η_s est le voisinage du pixel s et $x = \{0, 1\}$. La détection avec le terme *a priori* s'écrit :

$$\mathcal{X}_t(s) = \begin{cases} 1 & \text{si } d(I_{s,t}, B_{s,t}) > \tau' \\ 0 & \text{sinon,} \end{cases} \tag{2.5}$$

où

$$\tau' = \tau - \beta(L(\mathcal{X}(s), 0) - L(\mathcal{X}(s), 1)), \tag{2.6}$$

β est un seuil prédéfini. Plus il y a de pixels de l'avant-plan à 1 qui entourent un pixel, plus le seuil τ' sera petit et donc plus il aura de chance d'appartenir lui aussi à l'avant-plan.

Les résultats présentés dans la figure 2.9 ont été obtenus avec l'algorithme *Basic* suivi des différentes méthodes de post-traitement. Ces résultats sont obtenus sur une base de 12 vidéos avec des arrière-plans multimodaux et un bruit additif.

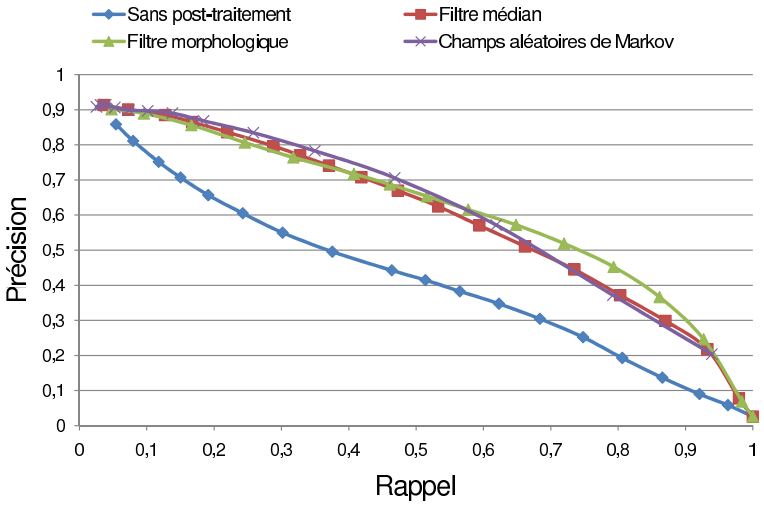
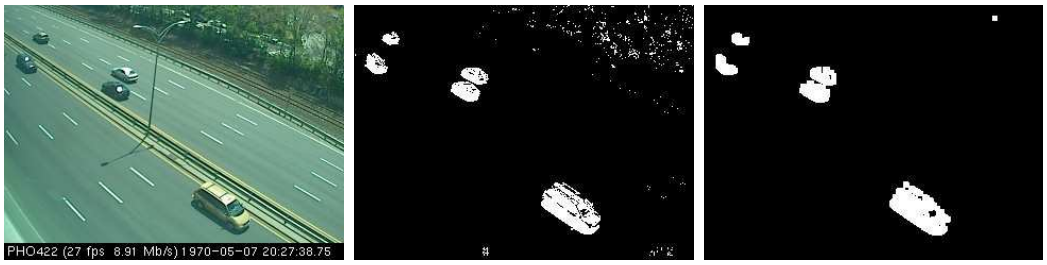


Figure 2.9: Courbes Précision/Rappel. Évaluation des méthodes de post-traitement.

Le post-traitement augmente clairement les performances de la soustraction de l'arrière-plan. Les différences entre les méthodes de post-traitement ne sont pas très importantes. Cependant, pour des performances équivalentes, on peut simplement remarquer que l'*a priori* markovien est un processus itératif et donc demande plus de calcul qu'un simple filtrage. Un exemple de soustraction de l'arrière-plan, illustrant les bénéfices de l'utilisation du post-traitement, est présenté dans la figure 2.10 dans le contexte de la vidéo-surveillance de trafic routier.



(a) (b) (c)

Figure 2.10: (a) image d'entrée (b) avant-plan obtenu avec l'algorithme *B*asic seul (c) avant-plan obtenu après post-traitement par filtrage morphologique.

2.1.2.6 Performance après optimisation

Nous présentons désormais les résultats obtenus par l'ensemble des méthodes sur toute la base, soit 36 vidéos contenant des vidéos avec arrière-plans statiques, arrière-plans multimodaux et les vidéos dégradées. Certaines vidéos de la base de vidéos dégradées sont des vidéos de la base de vidéos avec arrière-plans statiques auxquelles nous avons dégradé la qualité. Nous appliquons un filtre morphologique sur tous les résultats de détection. Les résultats sont présentés dans la figure 2.11.

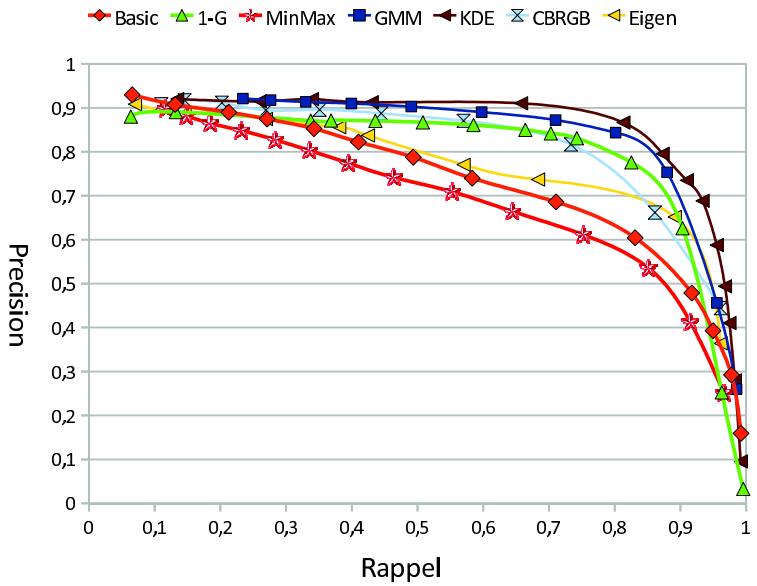


Figure 2.11: Courbes Précision/Rappel. Évaluation des méthodes sur l'ensemble de la base après post-traitement.

Nous pouvons tout d'abord remarqué que les écarts entre les différentes méthodes se sont réduits par rapport aux résultats très hétérogènes obtenus sur les bases de vidéos bruitées et multimodales (*cf.* figure 2.5 et 2.7). Le post-traitement tend à uniformiser les performances des méthodes. Il est cependant possible d'observer des différences entre tout d'abord *MinMax*, *Basic* et *Eigen* et ensuite *1-G*, *GMM*, *KDE* et CB_{RGB}. Le premier groupe de méthodes présente des résultats inférieurs au second groupe. Cette observation est intéressante puisque *1-G*, qui est une méthode structurellement simple, obtient des performances similaires à d'autres méthodes complexes comme *GMM* et *KDE* après post-traitement. Il faut cependant noter qu'il y a seulement 6 vidéos avec des arrière-plans multimodaux dans cette base de vidéos. S'il y avait plus de vidéos multimodales dans la base, les résultats auraient été différents mais cette observation est cependant intéressante pour *CAPTHOM* puisque nous travaillons dans des environnements intérieurs et donc *a priori* unimodaux.

2.1.2.7 Contraintes matérielles

La soustraction de l'arrière-plan n'est en général qu'une partie d'une application d'analyse vidéo. Dans le cadre du projet *CAPTHOM*, les contraintes sur les temps de calcul et l'espace mémoire sont très importantes. En effet, les capacités de calcul des cibles embarquées sont assez limitées et l'espace mémoire disponible est restreint. Nous présentons donc ci-dessous les performances relatives des algorithmes de soustraction de l'arrière-plan en considérant les temps de calcul et l'espace mémoire utilisé.

Temps de calcul La méthode CB_{RGB} n'apparaît pas dans cette comparaison puisque le nombre d'opérations par pixel et l'espace mémoire dépend directement de la complexité de la vidéo. Pour un arrière-plan uni-modal, CB_{RGB} est aussi rapide que *1-G* et utilise les mêmes ressources mémoire mais sa vitesse de calcul diminue et son espace mémoire utilisé augmente à mesure que la complexité de l'arrière-plan augmente.

Les valeurs de vitesse d'exécution obtenues ne sont significatives que lorsqu'elles sont considérées relativement. Chaque valeur est donc divisée par le temps de calcul de l'algorithme de référence (*Basic*). Bien entendu, les temps de calcul sont directement liés à l'implémentation mais les résultats présentés dans le tableau 2.3 permettent néanmoins de tirer quelques conclusions.

Algorithme	**Temps relatif**
Basic	1
1-G	1.32
MinMax	1.47
GMM	4.91
KDE	13.80
Eigen	11.98

Table 2.3: Temps d'exécution relatif des méthodes de soustraction de l'arrière-plan.

Dans le tableau 2.3, nous pouvons voir que les résultats sont très hétérogènes. Nous pouvons répartir les algorithmes en deux groupes. Il y a tout d'abord les méthodes rapides comme *Basic*, *1-G* et *MinMax*. Ces algorithmes sont structurellement simples. Il y a ensuite, *GMM*, *KDE* et *Eigen*. Ces méthodes sont entre 5 et 14 fois plus lentes que *Basic* !

Espace mémoire Une autre information très importante est l'espace mémoire utilisé. L'espace utilisé influe directement sur l'architecture matérielle des cibles embarquées et donc sur le prix du capteur. Nous comparons dans le tableau 2.4, l'espace mémoire minimal nécessaire pour modéliser un pixel de l'arrière-plan. Par exemple, si

nous considérons l'algorithme *Basic*, pour chaque pixel, il y a 3 valeurs qui modélisent l'arrière-plan (une valeur par composante de couleur).

Algorithme	**Nombre de flottants par pixel**
Basic	3, soit μ_r, μ_v et μ_b
1-G	6, soit μ_r, μ_v, μ_b, σ_r, σ_v et σ_b
MinMax	3, soit m_s, M_s et D_s
GMM	$K \times 5$, soit $K \times (\mu_r, \mu_v, \mu_b, \omega, \sigma)$
KDE	$N \times 3 + 3$, soit $N \times (I_r, I_v, I_b)$, σ_r, σ_v et σ_b
CB$_{\text{RGB}}$	$L \times 6$, soit $L \times (\mu_r, \mu_v, \mu_b, \sigma_r, \sigma_v, \sigma_b)$
Eigen	$M \times 3 + 3$ soit $M \times 3$ composantes de couleur, μ_r, μ_v et μ_b

Table 2.4: Espace mémoire utilisé.

Dans le tableau 2.4, L, K, M et N sont respectivement le nombre de valeurs clés (généralement de 1 à 4), le nombre de gaussiennes utilisées dans le mélange de gaussiennes (de 3 à 5), le nombre de vecteurs propres conservés (par exemple 20) et le nombre d'images dans la mémoire (généralement de 100 à 200 images). Si nous considérons l'espace mémoire utilisé, *KDE* et *Eigen* sont pénalisés ici.

2.1.3 Discussion

Nous pouvons tirer diverses conclusions de cette étude comparative. Tout d'abord et sans surprise, on peut remarquer que les méthodes qui travaillent sur des vidéos en niveaux de gris sont moins performantes que celles qui travaillent avec des vidéos couleurs.

Deuxièmement, il n'y a pas de meilleure méthode. Aucune n'est significativement plus performante que les autres dans toutes les catégories de l'évaluation. Le choix d'une méthode dépend donc de l'application et de l'environnement. Il faut trouver un compromis entre performance, vitesse d'exécution et espace mémoire.

Ensuite, toutes les méthodes utilisées dans cette étude comparative ont un certain nombre de paramètres dont la valeur est à déterminer empiriquement. À chaque situation spécifique correspond un jeu de paramètre optimal. Or, il n'est pas toujours possible d'ajuster les paramètres pour chaque situation. Il est donc préférable dans beaucoup de cas d'utiliser une méthode où peu de paramètres sont à régler. Le tableau 2.5 résume le nombre de paramètres à fixer pour chaque méthode.

Une dernière conclusion intéressante est que le bénéfice que l'on peut retirer de l'utilisation des méthodes complexes n'est pas toujours significatif. Il y a beaucoup de situations où une méthode simple sera aussi performante (en terme de qualité de détection) que les méthodes complexes. Nous pouvons prendre l'exemple d'une vidéo où l'image est de bonne qualité et où l'arrière-plan est *a priori* statique. Les

Algorithme	**Nombre de paramètres**
Basic	2, soit τ, α
1-G	3, soit τ, α et σ_0
MinMax	3, soit τ et deux paramètres utilisés lors de la mise à jour
GMM	6, soit τ, α, T, K, ω_0 et σ_0
KDE	3, soit τ, σ et N
CB$_{\text{RGB}}$	3, soit τ, α et σ_0
Eigen	3, soit τ, M et N

Table 2.5: Nombre de paramètres à fixer empiriquement.

méthodes simples sont dans ce cas aussi performantes que les méthodes complexes. Si l'on ajoute à cela les contraintes induites par les méthodes complexes en termes de vitesse d'exécution et de mémoire utilisée, l'intérêt des méthodes simples devient encore plus important.

Le tableau 2.6 présente une synthèse de tous les résultats obtenus dans cette étude comparative, le nombre d'étoiles est associé à la performance de la méthode. L'attribution de trois étoiles correspond à de bonnes performances dans ce contexte d'usage et une étoile à de plus mauvaises. L'attribution de ces étoiles est faite de manière subjective.

	Basic	**1-G**	**MinMax**	**GMM**	**KDE**	**CB$_{RGB}$**	**Eigen**
Arrière-plan statique	***	***	**	***	***	***	***
Arrière-plan multimodal	*	**	*	***	***	***	*
Arrière-plan bruité	*	***	*	***	***	***	***
Temps de calcul	***	***	***	**	*	-	*
Espace mémoire	***	***	***	**	*	**	*
Adaptatif	***	***	**	***	***	**	*
Nombre de paramètres	***	***	**	*	**	***	*

Table 2.6: Synthèse des résultats de l'étude comparative des algorithmes de soustraction de l'arrière-plan.

2.2 La reconnaissance d'humains

Nous présentons dans cette partie plusieurs résultats concernant la reconnaissance d'humains dans une image. Nous commençons tout d'abord par présenter le protocole de cette étude en détaillant la méthode d'évaluation et les différentes bases d'images utilisées. Ensuite, nous comparons les résultats obtenus avec les deux méthodes de détection de personnes les plus utilisées dans la littérature, à savoir Dalal *et al.* [34] et Viola *et al.* [148], puis nous étudions l'influence de plusieurs paramètres (*e.g.* base d'apprentissage *etc.*).

2.2.1 Protocole de l'étude

Principe Pour comparer les résultats, nous utilisons à nouveau les courbes Précision/Rappel et également la valeur maximale du *f-score* définie par :

$$\textit{f-score} = \frac{2.\textit{Précision}.\textit{Rappel}}{\textit{Précision} + \textit{Rappel}}. \tag{2.7}$$

La valeur du *f-score* a été utilisée, pour l'évaluation de système de vidéo-surveillance, en autre dans le projet ETISEO [113].

Pour déterminer si un résultat de détection est un vrai ou un faux positif, nous utilisons la règle suivante : soit $A = \{a_i, i \in \{1; \ldots; M\}\}$ l'ensemble des M résultats de détection de l'algorithme évalué et $B = \{b_j, j \in \{1; \ldots; N\}\}$ l'ensemble des N personnes dans la vérité terrain, la détection a_i est vérifiée s'il existe un j tel que :

$$\frac{card(a_i \cap b_j)}{card(a_i \cup b_j)} > 0,7. \tag{2.8}$$

Base d'images Plusieurs bases d'images ont été utilisées pour entraîner les différents classifieurs et pour réaliser des tests. Elles sont décrites ci-dessous :

- MIT : une base de 924 images contenant des personnes debout, vue de face ou de dos, proposée par le *MIT* [117, 120, 111], cf. figure 2.12. Cette base est relativement simple puisque les poses et les points de vue sont assez limités.
- INRIA : Une base de 1208 images contenant des personnes debout, vue sous tous les angles, adoptant différentes poses, proposée par Dalal [33] à l'*INRIA*, cf. figure 2.13. Cette base est actuellement la plus couramment utilisée dans la littérature.
- NICTA : Une base de 25551 images contenant des personnes debout, vue sous tous les angles, adoptant différentes poses, proposée par le centre de recherche *NICTA* [118], cf. figure 2.14.
- IR : Nous avons construit une quatrième base d'images composée de 1175 images contenant des personnes debout dans le spectre infrarouge. Elle est constituée d'images provenant de la base *OTCBVS* [39, 37] et d'images collectées avec notre caméra infrarouge, cf. figure 2.15.

– Une base d'images négatives composée de 3415 images provenant de la base *INRIA* et d'autres sources Internet, cf. figure 2.16.
– TEST : Une base de test, provenant de diverses sources Internet, composée de 215 images contenant au moins une personne dans le spectre visible avec une grande variété d'arrière-plans, de postures ou de vêtements, cf. figure 2.17.

Figure 2.12: Exemples d'images extraites de la base d'images *MIT*.

Figure 2.13: Exemples d'images extraites de la base d'images *INRIA*.

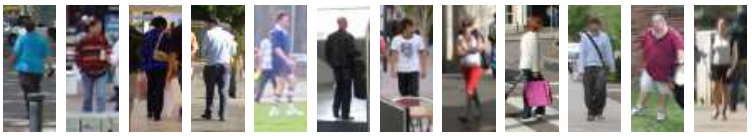


Figure 2.14: Exemples d'images extraites de la base d'images *NICTA*.

Figure 2.15: Exemples d'images extraites de la base d'images *IR*

Figure 2.16: Exemples d'images extraites de la base d'images négatives

Figure 2.17: Exemples d'images extraites de la base d'images *TEST*.

On notera que tous les apprentissages ont été réalisés en doublant la taille de la base d'images en considérant les symétries verticales.

Il existe également d'autres bases d'images de personnes dans la littérature qui ne sont pas utilisées ici :

- la base *Caltech* [42] composée d'environ 10 heures de vidéo prise par un véhicule en mouvement. Cette base est composée de 350000 boîtes englobantes et 2300 piétons uniques.
- la base *CVC* [58], composée d'environ 1000 personnes,
- la compétition *Pascal* [47], proposant des bases d'images de personnes, de voitures *etc.*
- la base *TUDMotionPairs* [152] composée de 1092 paires d'images avec 1776 piétons annotés,
- la base proposée par *ETH Zurich* [45] composée d'environ 13000 piétons,
- ...

2.2.2 Résultats expérimentaux

2.2.2.1 Comparaison de *Haar-Boost* et de *HOG-SVM*

Nous présentons ici une comparaison des performances de deux méthodes très utilisées dans l'état de l'art pour détecter des personnes dans des images. Nous comparons la méthode de Viola et Jones [148] basée sur les filtres de Haar et adaboost, appelée *Haar-Boost*, avec la méthode de Dalal et Triggs [34] basée sur les histogrammes de gradients orientés et les machines à vecteurs de support, appelée *HOG-SVM*.

Il existe dans la littérature quelques études ayant comparées ces deux méthodes (par exemple [42]). Celles-ci ont alors observées que *HOG-SVM* présentait de meilleures performances que *Haar-Boost*. Ces deux méthodes sont logiquement comparées sur la même base d'images. Cependant, nous avons observé qu'elles ne réagissaient pas de la même manière au contexte autour de l'objet dans la base d'apprentissage. Pour vérifier cela, nous avons modifié la base d'apprentissage *INRIA* afin d'obtenir trois sous-bases :

1. *Grand contexte* correspond à la base originale où se trouve un grand contexte d'arrière-plan autour de chaque personne (approximativement 16 pixels),
2. *Moyen contexte* correspond approximativement à 8 pixels autour de chaque personne,
3. *Petit contexte* correspond à un contexte réduit au minimum autour de chaque personne.

Un exemple d'image de la base d'apprentissage avec différentes tailles est donné dans la figure 2.18. Les résultats obtenus avec *Haar-Boost* sont illustrés dans la figure 2.19 et les résultats obtenus avec *HOG-SVM* dans la figure 2.20. On peut remarquer que le comportement de ces deux algorithmes en fonction du contexte autour des objets de la base d'apprentissage est opposé. Pour *Haar-Boost*, les performances sont

Figure 2.18: Modification du contexte des images de la base *INRIA*. De gauche à droite : grand, moyen et petit contexte.

meilleures lorsque le contexte est faible. Inversement, pour *HOG-SVM*, les meilleures performances ont été obtenues avec un grand contexte. Les mêmes conclusions concernant l'influence du contexte sur *HOG-SVM* ont été observées dans [33].

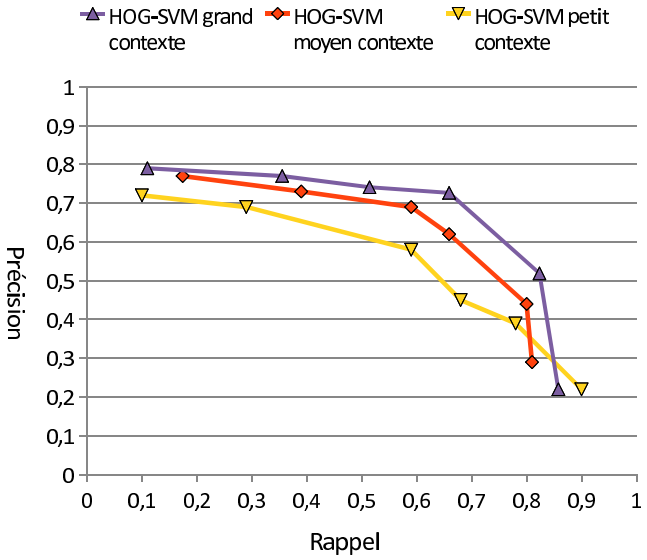


Figure 2.19: Courbes Précision/Rappel. Analyse de l'influence du contexte sur *HOG-SVM*.

Les résultats présentés dans la figure 2.21 et le tableau 2.7 sont finalement obtenus en comparant, sur la base *INRIA*, *HOG-SVM* en grand contexte et *Haar-Boost* en petit contexte. Les différences entre les deux méthodes sont ici moins prononcées que le présentait [42]. Il semble, au regard de ces courbes que la méthode proposée par Dalal [34] est légèrement plus performante sur notre base *TEST*. Cependant, pour une valeur de rappel inférieure à 0.5, la courbe *Haar-Boost* est au-dessus de

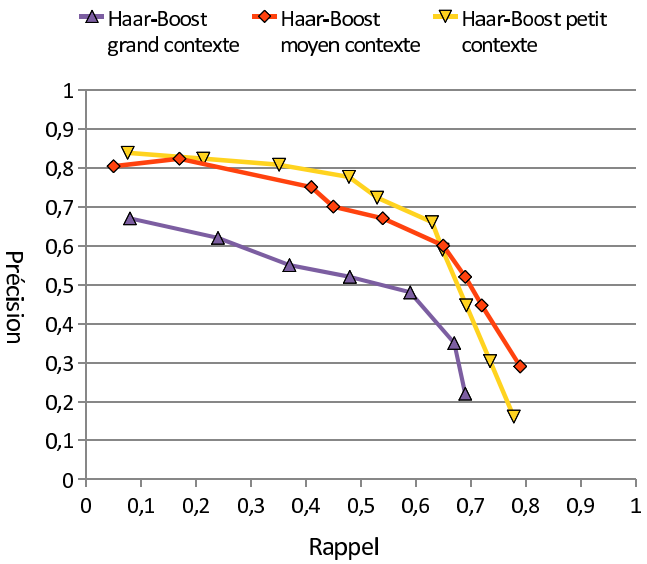


Figure 2.20: Courbes Précision/Rappel. Analyse de l'influence du contexte sur *Haar-Boost*.

HOG-SVM et inversement lorsque la valeur du rappel est supérieure à 0.5. D'après le tableau 2.7, on voit tout de même que le *f-score* de *HOG-SVM* est supérieure.

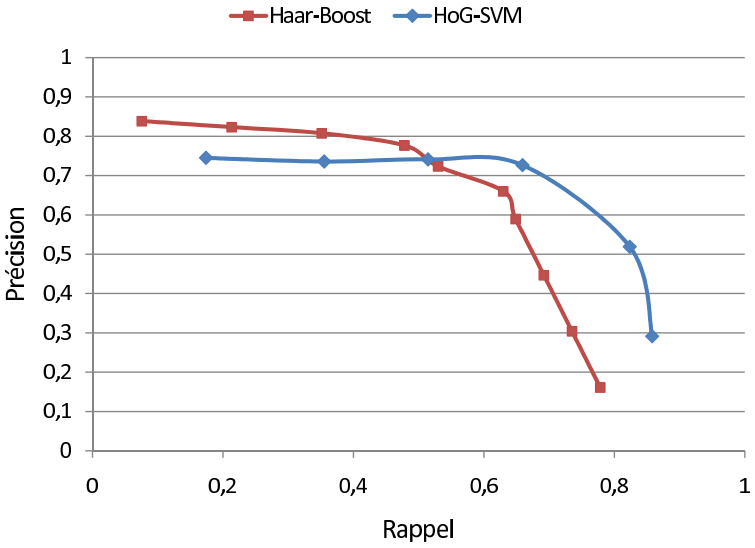


Figure 2.21: Courbes Précision/Rappel. Comparaison de *Haar-Boost* et *HOG-SVM*

Puisque les différences en terme de performance entre ces deux méthodes ne sont pas significatives et compte tenu des avantages apportés en terme de rapidité par *Haar-Boost*, nous avons donc choisi d'utiliser cette dernière pour la classification.

	HOG-SVM	*Haar-Boost*
f-score	**0.69**	0.65

Table 2.7: *f-score* des méthodes *Haar-Boost* et *HOG-SVM*

En effet, l'architecture en cascade permet de rejeter beaucoup d'exemples négatifs très rapidement et, les filtres de Haar peuvent être calculés avec seulement quelques opérations en utilisant les images intégrales.

2.2.2.2 Étude de l'influence de différents paramètres sur *Haar-Boost*

Tous les résultats présentés par la suite concernent la méthode de Viola *et al.* [148] à partir de laquelle plusieurs tests ont été effectués :

1. étude des différentes bases d'apprentissage,
2. étude des différentes variantes du boosting,
3. étude de la classification par parties,
4. étude du domaine spectral utilisé.

Test 1 : Étude des différentes bases d'apprentissage Nous présentons tout d'abord les résultats de détection obtenus avec plusieurs bases d'apprentissage. La base d'apprentissage a une importance cruciale sur la performance du classifieur. Un classifieur a été créé à partir de chacune des trois bases *MIT*, *INRIA* et *NICTA* composées respectivement de 924, 1208 et 25551 images positives. L'apprentissage sur la base *NICTA* a été réalisé par *ST Imaging*. Les résultats sont présentés dans la figure 2.22 et le tableau 2.8.

	INRIA	*NICTA*	*MIT*
f-score	**0.69**	0.57	0.48

Table 2.8: *f-score* des résultats de détection en utilisant différentes bases d'apprentissage.

Dans la base de test, il y a des personnes adoptant des postures variées, il semble donc logique que le classifieur obtenu avec la base *MIT* soit moins performant que le classifieur obtenu avec la base *INRIA*. En effet, les personnes présentes dans la base d'apprentissage *MIT* sont exclusivement de face ou de dos, contrairement à la base *INRIA*. Cependant, les résultats obtenus avec la base *NICTA* sont étonnamment moins bons. La résolution inférieure des images de cette base est certainement une cause de cette observation. Par la suite, nous utiliserons donc la base *INRIA*.

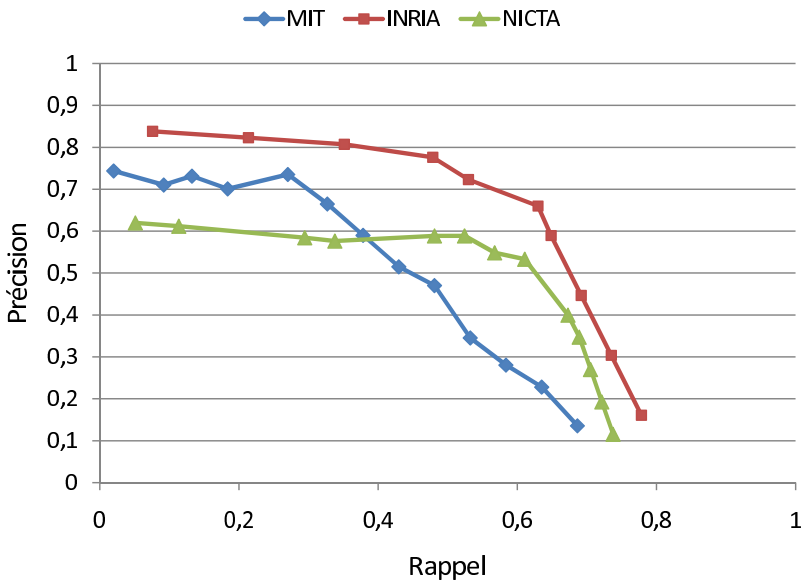


Figure 2.22: Courbes Précision/Rappel. Résultats obtenus avec différentes bases d'apprentissage.

Test 2 : Étude des différentes variantes du boosting Nous comparons désormais quatre variantes de l'algorithme du boosting appelées *Gentle Adaboost*, *Real Adaboost*, *Discret Adaboost* et *Logitboost*. Ces quatre variantes ont été décrites avec précision dans Friedman *et al.* [51]. Toutes ces méthodes ont une complexité identique. *Gentle Adaboost* et *Real Adaboost* sont les deux variantes qui sont les plus utilisées actuellement dans la littérature.

Les classifieurs ont été entraînés sur la base *INRIA* et évalués sur la base *TEST*. Les résultats sont présentés dans la figure 2.23 et le tableau 2.9.

	Gentle Adaboost	*Discret Adaboost*	*Real Adaboost*	*Logitboost*
f-score	**0.65**	0.62	0.59	0.58

Table 2.9: *f-score* des différentes variantes du boosting.

Il est montré que la différence entre ces variantes est relativement faible dans le cas de la détection de personnes. *Gentle Adaboost* est sensiblement meilleure que les autres et *Logitboost* est légèrement moins performante.

Test 3 : Évaluation de la classification par parties Nous présentons ensuite les résultats de détection obtenus avec un classifieur du corps entier et un classifieur de la tête et des épaules. Les classifieurs ont été entraînés sur la base *INRIA* et évalués sur la base *TEST*. Ces résultats seront riches d'enseignement pour notre application

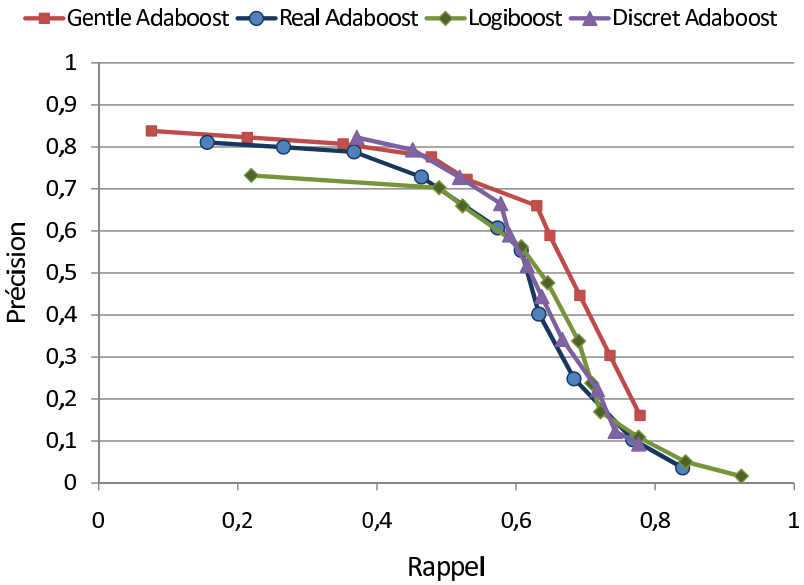


Figure 2.23: Courbes précision/rappel. Résultats obtenus avec différentes variantes du boosting.

puisque, si les performances sont équivalentes, il est alors suffisant d'utiliser un classifieur de la partie supérieure du corps. En effet, en milieu intérieur, le corps entier n'est pas souvent entièrement visible. Les résultats sont présentés dans la figure 2.24 et le tableau 2.9.

	corps entier	partie supérieure
f-score	**0.69**	0.49

Table 2.10: *f-score* des résultats de détection en utilisant un classifieur du corps entier et un classifieur de la partie supérieure du corps.

Sur les courbes Précision/Rappel de la figure 2.24, on voit clairement que les résultats de détection de la partie supérieure sont inférieurs aux résultats obtenus sur le corps entier. Ces résultats étaient attendus, mais la différence observée est très significative. Beaucoup moins de détails sont disponibles en ne considérant que la partie supérieure du corps.

Test 4 : Évaluation du domaine spectral utilisé pour la détection. Finalement, nous présentons les résultats de détection dans le spectre infrarouge et le spectre visible. L'apprentissage du classifieur dans le spectre visible a été réalisé avec la base *INRIA* et l'apprentissage du classifieur dans le spectre infrarouge a été réalisé avec la base *IR*. Pour évaluer les performances de ces deux détecteurs, nous avons construit deux bases composées de 640 images chacune. Nous avons construit

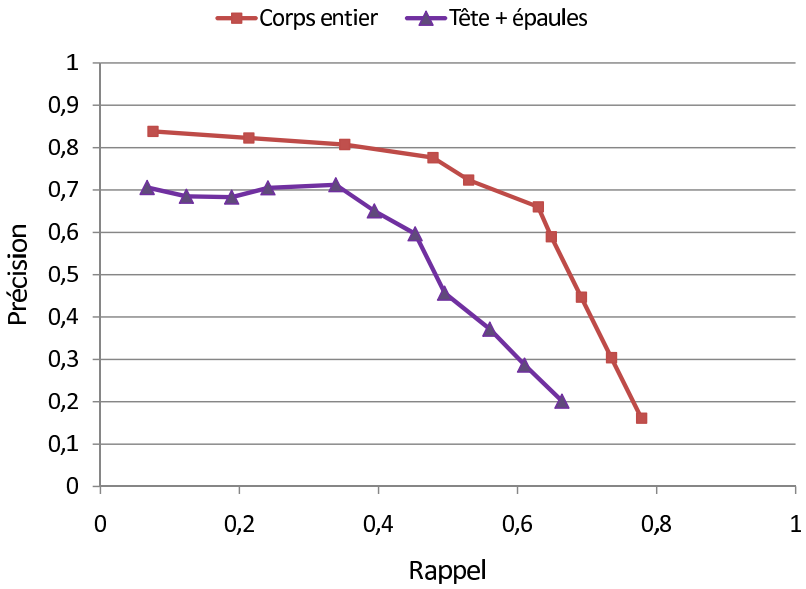


Figure 2.24: Courbes Précision/Rappel. Résultats obtenus avec un classifieur du corps entier et de la tête et des épaules.

une nouvelle base de test afin d'avoir pour chaque scène, une image dans les deux spectres. Cette base sera également utilisée pour valider le système de détection de personnes par stéréovision présenté dans le chapitre 4.3. Les résultats sont présentés dans la figure 2.25 et le tableau 2.11.

	Infrarouge	Visible
f-score	**0.70**	0.56

Table 2.11: *f-score* des résultats dans le spectre infrarouge et le spectre visible.

On peut observer que les performances sont clairement meilleures dans le spectre infrarouge. Le contraste entre les personnes et l'arrière-plan est plus prononcé dans les images infrarouges, il y a donc moins de détections manquées en IR. De plus, l'arrière-plan est en général plus homogène dans les images du spectre infrarouge, il y a donc moins de fausses détections dans ce spectre.

Bien entendu, le prix de la technologie des caméras infrarouge est encore aujourd'hui très dissuasif. Cependant, certaines applications ayant de fortes contraintes sur les performances de détection ou de fortes contraintes sur les conditions d'illumination tout en ayant plus de tolérance au regard du prix peuvent être intéresser par la technologie infrarouge. Par exemple, le secteur de l'automobile utilise dés aujourd'hui la technologie infrarouge pour des systèmes d'aide à la conduite.

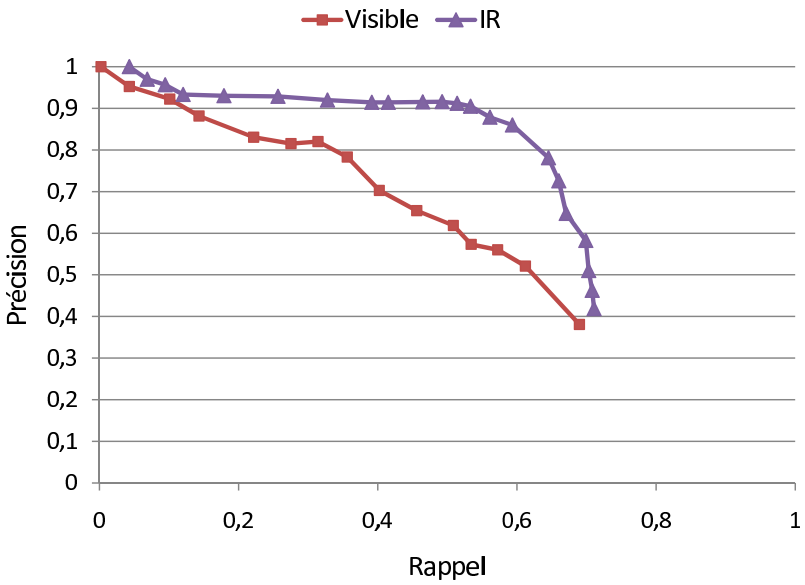


Figure 2.25: Courbes Précision/Rappel. Résultats obtenus dans le spectre visible et le spectre infrarouge.

2.3 Conclusion

Nous avons présenté dans ce chapitre les résultats de deux études comparatives. La première concerne les algorithmes de soustraction de l'arrière-plan. Ces derniers ont été évalués sur une large base de vidéos réelles, semi-synthétiques et synthétiques. Dans un second temps, nous avons présenté des résultats concernant la reconnaissance d'humains dans une image. Deux méthodes de reconnaissance très utilisées dans la littérature ont été comparées. Nous avons ensuite fait varier plusieurs paramètres (taille de la base d'apprentissage *etc.*) afin d'optimiser cette étape.

Les résultats présentés précédemment vont servir à la conception de la méthode de détection de personnes proposée dans le cadre du projet *CAPTHOM*. Tout d'abord, concernant la soustraction de l'arrière-plan, les résultats présentés ici permettent clairement de faire le choix d'une méthode. En effet, il ressort de cette étude que les bénéfices retirés de l'utilisation des méthodes complexes ne sont pas très marqués lorsque l'on travaille sur des vidéos qui ne présentent pas de difficultés particulières. L'environnement dans lequel sera utilisé la caméra est *a priori* relativement simple. Il y aura peu de zones clairement multimodales comme il peut y en avoir dans des environnements extérieurs. Enfin, nous avons également besoin d'une méthode facilement paramétrable et qui ne nécessite pas beaucoup de calcul et d'espace mémoire. Il paraît alors évident qu'une méthode simple est plus adaptée. Nous avons choisi d'utiliser un modèle gaussien de l'arrière-plan. Cette méthode présente de bonnes performances, utilise peu d'espace mémoire et de ressource de calcul et est également facilement paramétrable.

Deuxièmement, concernant la reconnaissance d'humains, nous sommes également désormais capable d'effectuer des choix à propos de la méthode que nous utiliserons. Même si les performances obtenues avec la méthode de Viola et Jones [148] et la méthode de Dalal et Triggs [34] sont légèrement à l'avantage de cette dernière, nous avons choisi d'utiliser la méthode de Viola et Jones [148] pour la reconnaissance, et ce pour deux raisons. Tout d'abord, les performances sont assez proches. Ensuite, cette méthode est particulièrement adaptée aux contraintes matérielles de notre application. L'utilisation des images intégrales permet le calcul très rapide des filtres de Haar et l'architecture en cascade permet de rejeter rapidement les exemples négatifs. Ensuite, en ce basant sur les résultats obtenus ici, nous avons choisi d'utiliser la méthode *Gentle Adaboost* et de réaliser l'apprentissage sur la base *INRIA* en réduisant le contexte autour des personnes sur chaque image. Ensuite, même si d'un point de vue système il paraît suffisant de n'utiliser qu'un détecteur de la partie supérieure du corps, les résultats présentés dans cette partie montrent clairement que les performances attendues sont moins bonnes avec ce détecteur. Nous utiliserons donc plusieurs détecteurs, et combinerons les résultats de détecteurs de la partie supérieure du corps et d'un détecteur du corps entier. Finalement, les performances de détection obtenues dans le spectre infrarouge sont nettement meilleures que dans le spectre visible. Au vue des contraintes économiques du projet *CAPTHOM*, il ne paraît pas raisonnable de baser nos travaux sur le spectre infrarouge, cependant nous présenterons des travaux dans le dernier chapitre de ce manuscrit exploitant les avantages de ce spectre.

Les études présentées dans ce chapitre nous ont donc permis d'établir les grandes lignes des méthodes que nous utiliserons pour la détection de personnes. À partir de ces résultats, nous détaillons dans le chapitre suivant la méthode proposée dans le cadre de cette thèse pour la détection de personnes.

CHAPITRE 3

Détection de personnes

Dans ce chapitre, nous présentons la méthode de détection de personnes proposée pour le projet CAPTHOM. *Cette méthode s'articule autour de trois grandes étapes : la détection de changement, le suivi d'objets mobiles et la classification. La soustraction de l'arrière-plan permet de simplifier les traitements ultérieurs en localisant les régions d'intérêt dans l'image. La mise à jour du modèle de l'arrière-plan est réalisée à trois niveaux afin de prendre en compte les différentes variations possibles de l'environnement. Ensuite, à partir de la liste des composantes connectées détectées, nous établissons un historique de leur déplacement dans le plan image. Le suivi est basé sur la combinaison de l'analyse des composantes connectées et le suivi de points d'intérêt. Chaque région d'intérêt est analysée par plusieurs classifieurs par parties pour déterminer leur nature. Nous construisons alors un indice de confiance sur l'appartenance de cet objet à la classe "humain". Cette méthode a été évaluée sur une large base de vidéos correspondant aux scénarios de référence, établis par les partenaires du projet, auxquels le système doit être capable de répondre.*

Nous détaillons ici le système de détection de personnes proposé pour le projet *CAPTHOM*. Il est basé sur l'analyse du flux vidéo d'une caméra. Cette méthode s'articule autour de trois grandes étapes : la détection de changement, le suivi d'objets mobiles et la classification. Avant de détailler chacune de ces étapes, nous commençons par présenter le principe général du système proposé.

3.1 Principe général du système proposé

La figure 3.1 présente une illustration de l'architecture de l'algorithme proposé.

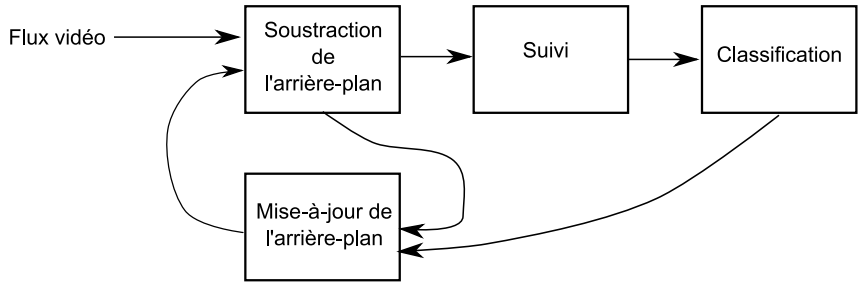


Figure 3.1: Architecture de l'algorithme proposé.

La première étape est donc la soustraction de l'arrière-plan. Cette étape permet de simplifier les traitements ultérieurs en localisant les régions d'intérêt dans l'image. Suite aux résultats de l'étude comparative présentée dans le chapitre précédent, nous avons choisi de modéliser chaque pixel de l'arrière-plan par une distribution gaussienne. Ce modèle présente un très bon compromis entre qualité de détection, temps de calcul et espace mémoire utilisé. La mise à jour est réalisée à trois niveaux. Tout d'abord, chaque pixel est mis à jour par un filtre temporel afin de s'adapter aux variations lentes de l'environnement. Ensuite, une mise à jour au niveau de l'image permet de prendre en compte les variations rapides de l'illumination de la scène. Enfin, nous forçons les objets statiques non-humain (ou leur rémanence) à appartenir à l'arrière-plan très rapidement.

Ensuite, à partir de la liste des composantes connectées détectées par la soustraction de l'arrière-plan, nous souhaitons connaître un historique de leur déplacement dans le plan image. Une composante connectée correspondant potentiellement à une personne, nous souhaitons suivre indépendamment chaque personne présente dans la scène en lui affectant une étiquette consistante dans le temps. Cet historique est très utile puisqu'il nous permet de grandement augmenter les performances du système global. Le suivi est basé sur la combinaison de l'analyse des composantes connectées et le suivi de points d'intérêt. Chaque objet est caractérisé par un ensemble de points qui sont suivis dans chaque image. La position de ces points par rapport aux composantes connectées permet de faire la correspondance entre les objets suivis et les composantes connectées détectées par la soustraction de l'arrière-plan.

Une fois la région d'intérêt détectée et suivie, nous déterminons la nature de cet objet de façon à répondre à la question suivante : est-ce que nous suivons un humain ?

Pour ce faire, nous utilisons la méthode de classification proposée initialement par Viola et Jones [148]. Cette méthode est basée sur les filtres de Haar et adaboost. La nature de l'objet suivi est déterminée en parcourant son voisinage avec plusieurs classifieurs. Nous construisons alors un indice de confiance sur l'appartenance de cet objet à la classe "humain".

3.2 Système d'acquisition

Dans le cadre de cette thèse, nous avons utilisé trois types de matériel d'imagerie dans des domaines spectraux différents : visible, proche infrarouge et infrarouge.

La caméra dans le domaine spectral visible, de longueurs d'ondes comprises entre $400nm$ et $700nm$, présente l'avantage d'être aujourd'hui très répandue (téléphonie mobile, appareil photo numérique...). Les coûts de production ont en conséquence beaucoup chuté ces dernières années (le prix peut être estimé à quelques euros). Cependant, pour obtenir une image de qualité exploitable, il est nécessaire d'avoir un niveau d'illumination minimal de la scène. La vision nocturne n'est pas possible avec cette technologie. De plus, le contraste entre les personnes et la scène n'est pas toujours très prononcé et il y a des difficultés dues aux projections d'ombres et aux reflets.

Pour être capable de voir la nuit avec du matériel bon marché, il est possible d'utiliser le domaine spectral proche infrarouge, de longueurs d'ondes comprises entre $700nm$ et $1400nm$. La technologie utilisée est la même que les caméras dans le domaine visible (par exemple CCD). En effet, les capteurs utilisés pour le visible sont sensibles également (dans une moindre mesure) aux longueurs d'ondes dans le proche infrarouge. Cependant, pour récupérer une image lorsque l'illumination est très faible, il est nécessaire d'apporter une source de lumière. Il est possible d'utiliser une source de lumière émettant à environ $800nm$, donc invisible à l'œil nu, mais visible par la caméra. Les capteurs étant identiques aux caméras standards, les coûts de production sont également très faibles. Puisque le projecteur proche infrarouge n'est composé que de LED, il est donc également bon marché. Cette technologie est aujourd'hui beaucoup utilisée pour la vidéo-protection. Il est également possible de trouver sur le marché des veille-bébés à vision nocturne basés sur cette technologie. Cependant, la consommation moyenne d'un projecteur est de quelques watts, il est donc impossible de respecter les normes environnementales imposées par la directive EUP (Eco-Design Of Energy Using Products) statuant sur la consommation des appareils électriques si le système d'éclairage est allumé en permanence. Il faut alors avoir une approche système en couplant la caméra avec un autre capteur - un capteur pyroélectrique par exemple - qui commanderait le déclenchement de la caméra lorsqu'il détecte un mouvement. L'image obtenue est en niveau de gris, le contraste entre les personnes et l'arrière-plan est incertain. L'utilisation d'un projecteur accentue également fortement la projection d'ombres.

Il est possible d'utiliser le domaine spectral de l'infrarouge lointain ou infrarouge thermique, de longueurs d'ondes comprises entre $3\mu m$ et $1000\mu m$. Les caméras thermiques permettent d'avoir une image de la scène dans la partie basse de cette bande

spectrale. Ces caméras sont insensibles à la lumière ambiante et permettent la vision nocturne sans ajouter de sources de lumière. C'est une technologie qui reste encore aujourd'hui très chère (le prix d'une caméra thermique est de plusieurs milliers d'euros), cependant, on peut constater la multiplication des applications utilisant la vision infrarouge, comme par exemple le secteur automobile pour des applications d'aide à la conduite. Le secteur automobile étant porteur de gros volumes de production, il est raisonnable de penser que les coûts de production de telle caméra vont fortement décroître dans les années à venir. Le corps humain rayonne à environ $10\mu m$, les caméras thermiques sont en général sensibles aux longueurs d'ondes entre $7,5\mu m$ et $13\mu m$, le corps humain est donc totalement détectable avec une caméra infrarouge. Le contraste entre les personnes et l'environnement est en règle générale très nettement marqué. Cependant, il est possible de rencontrer quelques cas particuliers où cette observation n'est pas vérifiée. Par exemple, si une personne porte un manteau à température ambiante, la faible différence de température entre ce manteau et l'environnement ne permettra pas de clairement distinguer cette personne jusqu'à ce que le manteau soit réchauffé par sa chaleur corporelle. Il y a également des problèmes de réflexions sur tous les sols plastiques et les surfaces en verre.

Nous présentons dans le tableau 3.1 un résumé des avantages et inconvénients de ces différentes technologies et dans la figure 3.2 un exemple d'images d'une scène sombre obtenue dans les trois spectres.

	Avantages	**Inconvénients**
Visible	Prix	Niveau d'illumination minimum nécessaire Sensible aux ombres, reflets etc. Contrastes entre les personnes et l'arrière-plan incertains
Proche IR	Prix	Éclairage d'appoint consommant beaucoup d'énergie (contradiction avec les applications visées) Approche système pour déclencher l'éclairage d'appoint nécessaire Projection des ombres accentuée par l'éclairage Contrastes entre les personnes et l'arrière-plan incertains
IR Lointain	Vision nocturne Contraste entre les personnes et l'arrière-plan clairement défini Information sur la température	Prix Réflexions sur les surfaces en verre ou plastique

Table 3.1: Avantages et inconvénients de l'utilisation des différents domaines spectraux pour la détection de personnes.

Figure 3.2: Exemple d'images obtenues en vision nocturne avec une caméra dans le spectre visible, le proche infrarouge (avec éclairage) et l'infrarouge.

3.3 Soustraction de l'arrière-plan

La soustraction de l'arrière-plan permet de simplifier les traitements ultérieurs en localisant les régions d'intérêt dans l'image. À partir d'un modèle de l'environnement et d'une observation, on cherche à détecter ce qui a changé. C'est une étape très importante car les étapes suivantes se baseront sur ce résultat. Pour notre application, les régions d'intérêt sont les régions de l'image où il y a une forte probabilité qu'il y ait une personne. L'image 3.3 présente un exemple dans lequel les régions d'intérêt sont détectées avec une soustraction de l'arrière-plan. On voit clairement dans cet exemple qu'il est *a priori* plus facile de déterminer la nature de l'objet détecté à l'issue de la soustraction de l'arrière-plan que de parcourir toute l'image originale à la recherche d'une forme humaine. Par la suite, le terme "arrière-plan" désigne l'union de toutes les zones statiques de l'image et le terme "avant-plan" correspond aux zones de l'image où un changement a été détecté.

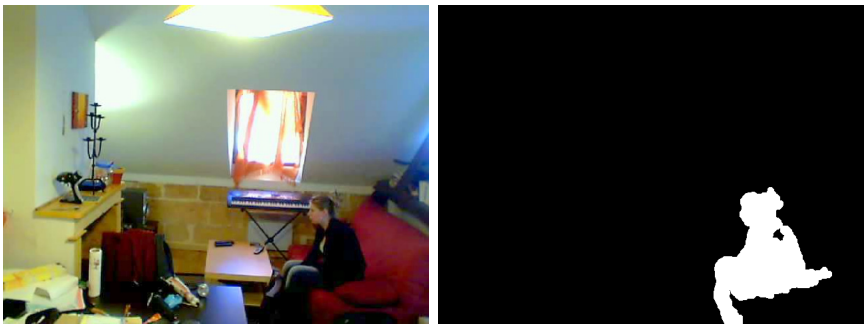


Figure 3.3: Exemple de résultat obtenu par soustraction de l'arrière-plan.

3.3.1 Méthode développée

D'après les résultats présentés lors de l'étude comparative des algorithmes de soustraction de l'arrière-plan dans le chapitre 2, il apparaît clairement que le meilleur compromis entre la qualité de la détection, le temps de calcul et la mémoire utilisée est obtenu avec des méthodes de soustraction de l'arrière-plan simples. Nous avons choisi de modéliser chaque pixel de l'arrière-plan par une densité de probabilité gaussienne [153]. Pour chaque pixel s, le modèle de l'arrière-plan $B_{s,t}$ est composé de la

moyenne $\boldsymbol{\mu}_{s,t} = \{\mu_{r,s,t}, \mu_{v,s,t}, \mu_{b,s,t}\}$ et de la matrice de covariance $\boldsymbol{\Sigma}_{s,t}$. Nous posons l'hypothèse que $\boldsymbol{\Sigma}_{s,t}$ est diagonale :

$$\boldsymbol{\Sigma}_{s,t} = \begin{pmatrix} \sigma^2_{r,s,t} & 0 & 0 \\ 0 & \sigma^2_{v,s,t} & 0 \\ 0 & 0 & \sigma^2_{b,s,t} \end{pmatrix} \tag{3.1}$$

La distance de Mahalanobis est utilisée pour calculer l'écart entre le modèle de l'arrière-plan et l'image courante :

$$d_M(\boldsymbol{I}_{s,t}, B_{s,t}) = (\boldsymbol{I}_{s,t} - \boldsymbol{\mu}_{s,t})\boldsymbol{\Sigma}^{-1}_{s,t}(\boldsymbol{I}_{s,t} - \boldsymbol{\mu}_{s,t})^T. \tag{3.2}$$

La détection est finalement réalisée par un simple seuillage de la distance :

$$\mathcal{X}_{s,t} = \begin{cases} 1 & \text{si } d_M(\boldsymbol{I}_{s,t}, B_{s,t}) > \tau_1 \\ 0 & \text{sinon.} \end{cases} \tag{3.3}$$

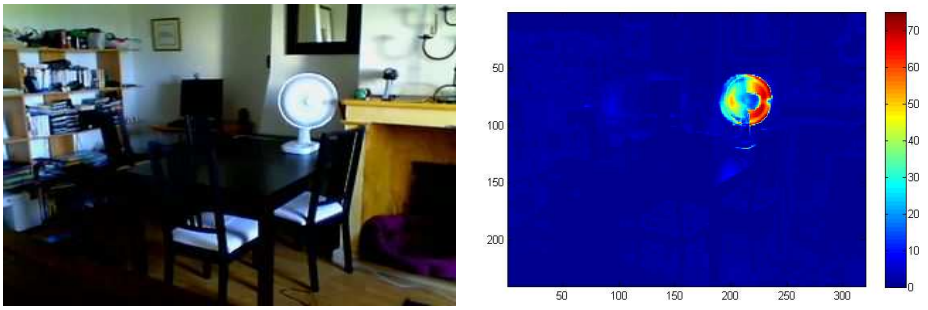
Si $\mathcal{X}_{s,t} = 1$, le pixel s appartient à l'avant-plan (ou région d'intérêt), sinon le pixel s appartient à l'arrière-plan et représente un point d'une zone statique.

En posant l'hypothèse que la matrice de covariance est diagonale, l'espace mémoire utilisé par cette méthode est relativement faible. En effet, le modèle $B_{s,t}$ n'est composé que de 6 valeurs par pixel (3 valeurs pour la moyenne et 3 valeurs correspondant aux termes diagonaux de la matrice de covariance). Ensuite, nous travaillons en milieu intérieur et il apparaît clairement que la probabilité d'avoir un environnement multimodal est alors moins important qu'en milieu extérieur. Il est cependant possible d'observer des zones multimodales dans la vidéo. Dans ce cas, la modélisation par une distribution gaussienne n'est pas la modélisation la plus adaptée mais puisque les valeurs des variances de la matrice de covariance seront grandes pour les zones de l'image en mouvement, la différence entre la moyenne $\boldsymbol{\mu}_{s,t}$ et le pixel $\boldsymbol{I}_{s,t}$ doit être plus importante dans cette zone que dans les zones clairement statiques. En effet, la distance de Mahalanobis est inversement pondérée par la matrice de covariance. Ce faisant, le nombre de fausses détections dans des zones multimodales est diminué.

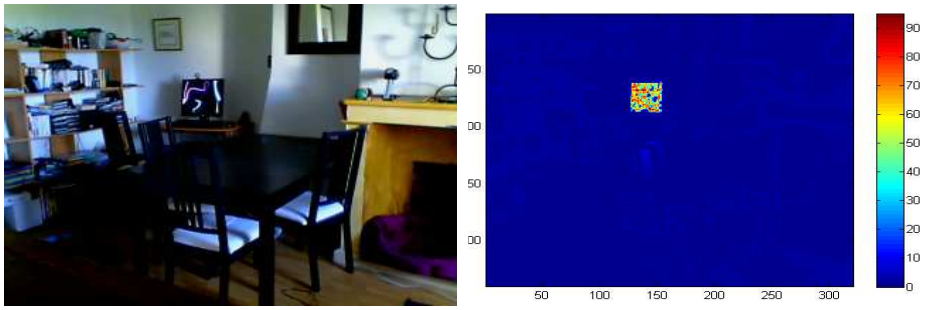
De plus, la répartition du bruit dans l'image n'étant pas *a priori* homogène, cette modélisation permet de pondérer la distance pour s'adapter localement aux spécificités de l'image. Le seuil reste global mais le terme $\boldsymbol{\Sigma}^{-1}_{s,t}$ dépendant directement de la quantité de bruit présent au pixel s pondère localement la distance. La figure 3.4 montre deux exemples de la répartition des valeurs des variances sur deux images. Dans le premier exemple, il y a un ventilateur en fonctionnement et dans le deuxième un écran d'ordinateur en veille en arrière-plan.

3.3.2 Mise à jour du modèle

Puisque la scène n'est jamais totalement statique, il faut permettre au modèle de s'adapter aux différentes variations de l'environnement :



(a)



(b)

Figure 3.4: Exemple de répartition des valeurs des variances sur toute l'image à travers deux exemples. (a) un ventilateur en fonctionnement (b) un écran d'ordinateur en veille.

1. une variation lente, causée par exemple par une variation de la lumière du jour,
2. une variation soudaine et importante due par exemple à l'ajout d'un éclairage d'appoint,
3. l'ajout ou le retrait d'un objet statique.

Pour la variation 1, une variation lente de l'illumination, la moyenne de chaque pixel est mise à jour par un filtre moyenneur :

$$\boldsymbol{\mu}_{s,t+1} = (1-\alpha).\boldsymbol{\mu}_{s,t} + \alpha.\boldsymbol{I}_{s,t} \tag{3.4}$$

les termes diagonaux de la matrice de covariance sont mis à jour par :

$$\begin{cases} \sigma^2_{r,s,t+1} = (1-\alpha)\sigma^2_{r,s,t} + \alpha(I_{r,s,t} - \mu_{r,s,t})^2 \\ \sigma^2_{v,s,t+1} = (1-\alpha)\sigma^2_{v,s,t} + \alpha(I_{v,s,t} - \mu_{v,s,t})^2 \\ \sigma^2_{b,s,t+1} = (1-\alpha)\sigma^2_{b,s,t} + \alpha(I_{b,s,t} - \mu_{b,s,t})^2 \end{cases} \tag{3.5}$$

où $\sigma^2_{r,s,t}$, $\sigma^2_{v,s,t}$ et $\sigma^2_{b,s,t}$ correspondent respectivement aux variances des composantes rouge, verte et bleue du pixel de coordonnée s à l'instant t. Le paramètre α détermine la rapidité à laquelle le modèle va se mettre à jour.

Pour la variation 2, une variation soudaine et importante de l'illumination, nous utilisons la valeur de :

$$\Omega = \frac{\sum_{s \in S} \mathcal{X}(s)}{S} \tag{3.6}$$

où S est le nombre de pixels dans l'image. $\sum_{s \in S} \mathcal{X}(s)$ correspond donc au nombre de pixels appartenant à l'avant-plan. Si la valeur de Ω est supérieure à un seuil prédéfini, la moyenne est réinitialisée avec l'image courante. Ce faisant, le modèle est suffisamment flexible pour s'adapter aux changements d'illumination brusques.

Pour la variation 3, une troisième mise à jour est réalisée au niveau objet. Cette mise à jour est utile puisqu'elle permet de réinitialiser le modèle de l'arrière-plan lorsque des objets statiques sont introduits ou retirés de la scène. Dans le dernier cas, l'objet qui n'est plus présent dans la scène reste dans le modèle de l'arrière-plan (appelé phénomène de rémanence ou fantôme) et dans le premier cas, l'objet introduit, bien que statique, n'est pas dans le modèle. Si l'écart entre le modèle et l'environnement sera comblé lentement avec le filtre moyenneur, pour notre application, nous avons choisi de forcer la mise à jour de telle sorte que le fantôme de l'objet mobile disparaisse rapidement ou que l'objet statique soit rapidement inclus dans le modèle de l'arrière-plan. Grâce aux étapes de suivi et de reconnaissance présentées après, nous sommes capables de déterminer la nature des objets présents et leurs positions dans les images précédentes. Si un objet détecté par la soustraction de l'arrière-plan est statique et considéré comme n'étant pas un humain pendant un nombre prédéfini d'images, la moyenne des pixels correspondant à cet objet est réinitialisée par la valeur de l'image courante.

Donc, en effectuant une mise à jour à trois niveaux, nous sommes capables de gérer les variations les plus courantes de l'environnement. Une mise à jour au niveau des pixels permet de prendre en compte les variations lentes. Une mise à jour globale, au niveau de l'image, permet de prendre en compte les variations brutales où tout le modèle à besoin d'être réinitialisé. Enfin, une mise à jour au niveau objet, force les objets qui ne sont pas des humains à appartenir à l'arrière-plan s'ils restent statiques.

3.3.3 Post-traitement

Les régions de l'avant-plan détectées correspondent idéalement à une région compacte avec des frontières lisses. Les fausses détections sont souvent réparties sur toute l'image et correspondent à de petits amas de pixels isolés. Dans l'étude comparative présentée précédemment, nous avons évalué les performances de plusieurs méthodes de post-traitement. Nous avons observé que leurs performances étaient globalement équivalentes. Nous avons choisi d'utiliser un ensemble d'opérations morphologiques pour supprimer les pixels isolés et combler les trous (faux négatifs) dans l'avant-plan. Nous utilisons un masque de taille 3×3 et nous utilisons une ouverture suivie d'une dilatation [15].

Ensuite, les pixels de l'avant-plan sont regroupés en composantes connectées [10]. Dans le cas idéal, une composante connectée correspond à un objet de la scène.

La figure 3.5 présente une illustration d'un résultat obtenu après une soustraction de l'arrière-plan, un filtrage et un regroupement en composantes connectées (une couleur représente une composante connectée).

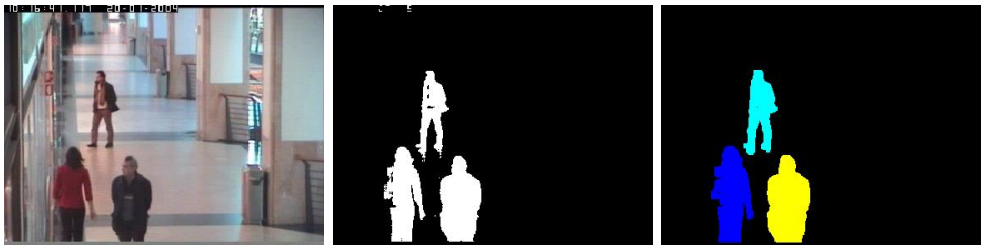


Figure 3.5: De gauche à droite : image originale, résultat de la soustraction de l'arrière-plan et résultat obtenu après post-traitement (une couleur représente une composante connectée).

3.4 Suivi d'objets

Pour la présentation de la méthode de suivi, nous utiliserons les termes "blob" ou "composante connectée" pour désigner des regroupements de pixels de l'avant-plan détectés avec la soustraction de l'avant-plan. Nous utiliserons aussi le terme "objet" pour désigner les entités de la scène (humain ou non).

Dans la section précédente, nous avons expliqué comment nous obtenons, pour chaque image, la liste des blobs présents avec leur position respective. Désormais, nous souhaitons connaître un historique du déplacement de ces blobs dans le plan image. Un blob correspondant potentiellement à un objet, nous souhaitons suivre indépendamment chaque objet présent dans la scène en lui affectant une étiquette consistante dans le temps. Cet historique est très utile puisqu'il nous permet de grandement augmenter les performances du système global en lissant dans le temps les erreurs de classification.

3.4.1 Méthode développée

Comme nous l'avons expliqué précédemment, les contraintes concernant le temps de calcul et l'espace mémoire utilisé sont très importantes dans notre système. Il ne semble donc pas adéquat d'utiliser un modèle complexe des objets suivis.

Ensuite, il est possible d'initialiser le processus de suivi avec le résultat de la détection de personnes (détection réalisée par exemple avec une fenêtre de classification glissante parcourant toute l'image *e.g.* [34]) ou avec le résultat du regroupement en composantes connectées des pixels de l'avant-plan détectés par la soustraction de l'arrière-plan. D'après les résultats de la reconnaissance présentés dans les parties 2.2 et 3.5, les performances d'un détecteur ne semble pas suffisamment fiables pour initialiser le suivi. Il semble alors plus judicieux d'utiliser le résultat de la soustraction

de l'arrière-plan pour le suivi. La représentation choisie est donc la silhouette des objets.

Une méthode de suivi basée directement sur l'avant-plan implique quelques difficultés. Par exemple, lorsque deux objets distincts sont très proches, ils ne forment qu'une seule composante connectée et inversement, un même objet peut être représenté par plusieurs composantes connectées s'il y a une occultation partielle ou s'il y a des trous dans la détection de l'avant-plan. Dans la figure 3.6, nous présentons un exemple où une personne est représentée par plusieurs blobs sur le résultat de la soustraction de l'arrière-plan. L'utilisation du suivi de points d'intérêt permet de gérer la plupart des cas les plus courants.

Figure 3.6: Exemple où une personne est représentée par plusieurs composantes connectées sur le résultat de la soustraction de l'arrière-plan.

À chaque instant t, nous disposons de la liste des blobs présents et de la liste des objets suivis dans les images précédentes. Nous cherchons donc à faire la correspondance entre les deux listes. Nous utilisons la matrice de correspondance $\mathcal{H}_t$ définie par :

$$\mathcal{H}_t = \begin{pmatrix} \beta_{1,1} & \cdots & \beta_{1,N} \\ \vdots & \ddots & \vdots \\ \beta_{M,1} & \cdots & \beta_{M,N} \end{pmatrix} \tag{3.7}$$

où M correspond au nombre d'objets suivis et N au nombre de blobs présents à l'instant t. $\beta_{i,j} = 1$ s'il y a correspondance entre l'objet suivi i et la composante connectée j, sinon $\beta_{i,j} = 0$.

Chaque objet est caractérisé par un ensemble de points d'intérêt. Ces points sont suivis, image par image, et la position de ces points par rapport aux composantes connectées permet de faire la correspondance entre les objets suivis et les blobs détectés.

Le suivi des points d'intérêt est réalisé avec la méthode de Lucas et Kanade [103, 142, 132, 14]. Deux contraintes sont ajoutées à la méthode originale :

1. Un point suivi doit être sur un pixel de l'avant-plan. Dans le cas contraire, le point est supprimé de la liste de points et un nouveau est créé.

2. Lors de la création d'un nouveau point, on impose une contrainte sur la distance avec les autres points de manière à avoir une répartition homogène des points sur tout l'objet.

La correspondance entre les objets suivis et les blobs est réalisée en calculant, pour chaque blob, à quel objet appartiennent les points présents sur ce blob. Soit $\gamma_{i,j}$ le nombre de points appartenant à l'objet i présents sur le blob j.

$$\begin{cases} \beta_{i,j} = 1 & \text{si } \gamma_{i,j} > \tau_2, \\ \beta_{i,j} = 0 & \text{sinon.} \end{cases} \tag{3.8}$$

Le seuil τ_2 dépend directement du nombre de points utilisés pour représenter un objet. En pratique, le seuil τ_2 est fixé à 25% du nombre de points d'intérêt par objet.

Un exemple est donné dans la figure 3.7 où il y a cinq objets suivis (représentés par leurs points d'intérêt) et cinq blobs détectés. En analysant à quel objet appartiennent les points présents sur chaque blob, nous sommes capables de construire la matrice de correspondance $\mathcal{H}_t$. Puisqu'il y a cinq objets et cinq blobs, la matrice $\mathcal{H}_t$ est une matrice 5×5.

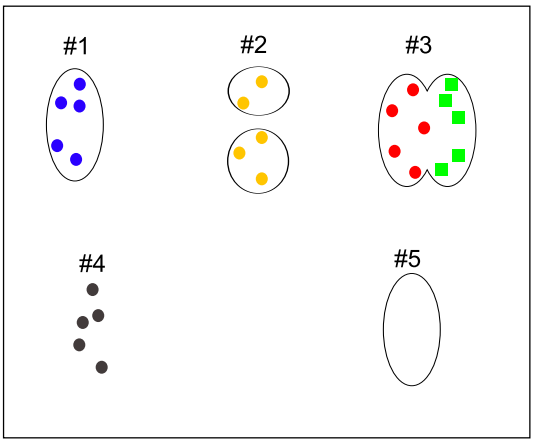


Figure 3.7: Illustration des 5 cas considérés par le suivi. Les points représentent les objets suivis (une couleur par objet) et les ovales les blobs détectés.

$$\mathcal{H}_t = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{3.9}$$

L'illustration figure 3.7 présente les 5 cas considérés par le suivi. Le comportement adopté dans chacun de ces 5 cas est décrit ci-dessous.

Cas n°1 : Correspondance

$$\mathcal{H}_t = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Ce premier cas, le plus simple, apparaît lorsqu'un seul blob correspond à un seul objet. On met alors simplement à jour les coordonnées de l'objet suivi par celles observées à l'instant t.

Cas n°2 : Séparation

$$\mathcal{H}_t = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Ce cas de figure intervient lorsqu'un objet est représenté par plusieurs blobs. Ceci peut être dû à une occultation partielle (cf. figure 3.6) ou lorsque plusieurs objets distincts étaient suffisamment proches dans les images précédentes pour être fusionnés puis s'éloignent les uns des autres sur l'image courante. Pour faire la distinction entre ces deux cas, chaque objet suivi i possède une variable λ_i qui a pour valeur le nombre d'entités de la scène que représente l'objet suivi. $\lambda_i = 1$ signifie que l'objet i ne représente qu'un seul objet. Donc si $\lambda_i > 1$ et l'objet i est représenté par plusieurs blobs, on sépare l'objet i en plusieurs objets. Si $\lambda_i = 1$, on met à jour l'objet i comme étant l'union des blobs correspondant (avec une contrainte sur la distance entre les blobs).

Cas n°3 : Fusion

$$\mathcal{H}_t = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Ce cas de figure intervient lorsqu'une composante connectée représente plusieurs objets. Dans ce cas, les objets concernés ne sont pas immédiatement fusionnés. Les coordonnées des objets auront pour valeurs la boîte englobante du blob concerné mais chaque objet aura ses propres points d'intérêt pendant quelques dizaines d'images. Ceci nous permet de suivre indépendamment les objets de la scène même avec de légères occultations mutuelles. Ensuite, les objets sont supprimés, un nouvel objet k est construit, de nouveaux points d'intérêt sont initialisés et la valeur de λ_k vaut la somme des λ des objets fusionnés.

Cas n°4 : Suppression

$$\mathcal{H}_t = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Dans ce cas de figure, un objet ne correspond à aucun blob. Si c'est le cas pendant plusieurs images consécutives, cet objet est supprimé de la liste.

Cas n°5 : Création

$$\mathcal{H}_t = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Dans ce cas de figure, un blob détecté ne correspond à aucun objet de la liste, alors un nouvel objet est créé, ses points d'intérêt sont initialisés.

3.4.2 Discussion

Si les ressources matérielles sont critiques lors du développement de la partie embarquée, il est alors possible de ne pas utiliser le suivi de points d'intérêt pour construire la matrice de correspondance mais d'analyser simplement le recouvrement des composantes connectées (entre l'instant $t-1$ et t) avec le critère de la compétition Pascal [46] :

$$Pas(A,B) = \frac{card(A \cap B)}{card(A \cup B)} \tag{3.10}$$

où A et B représentent les composantes connectées des objets à l'instant $t-1$ et t. La matrice de correspondance est alors construite en utilisant :

$$\begin{cases} \beta_{i,j} = 1 & \text{si } \ Pas(i,j) > \tau \\ \beta_{i,j} = 0 & \text{sinon.} \end{cases} \tag{3.11}$$

Cependant, si chaque personne suivie est caractérisée par des points d'intérêt, on a vu dans le cas n°3 décrit ci-dessus qu'il est possible de gérer les légères occultations mutuelles. L'étiquette associée à chaque objet suivi sera donc consistante dans le temps. En utilisant le critère de Pascal, il est impossible (sans considérer d'autre modèle de l'objet ou de modèle du déplacement) de conserver l'étiquette individuelle de chaque objet après une occultation mutuelle. De plus, la position des

points d'intérêt nous permet de récupérer des informations qui peuvent être utiles pour la reconnaissance d'activités ou encore la reconnaissance de posture.

Nous présentons dans la figure 3.8 un exemple de résultat du suivi. Dans cet exemple, deux personnes se croisent durant quelques instants. On voit bien sur cet exemple que l'identifiant de chaque personne est consistant dans le temps.

Figure 3.8: Illustration d'un résultat de suivi avec une occultation partielle temporaire. La première ligne correspond aux points d'intérêt suivis (une couleur par objet) et la deuxième ligne correspond au résultat du suivi avec l'identifiant de chaque objet suivi affiché.

3.5 Classification

Une fois la région d'intérêt détectée et suivie, nous souhaitons connaître sa nature, en l'occurrence, si c'est un humain. Comme nous l'avons présenté dans l'état de l'art, la méthode de classification proposée par Viola et Jones [148] semble être clairement adaptée à notre application car elle présente quelques avantages déterminants :

- les performances avancées sont très bonnes [148]. Nous présentons dans la partie 2.2.2.1 une comparaison de cette méthode avec une autre méthode [34] très utilisée pour la détection de personnes et les performances sont assez proches malgré une complexité moindre,
- les filtres de Haar peuvent être calculés très rapidement (seulement quelques opérations) en utilisant les images intégrales,
- l'architecture en cascade permet de rejeter beaucoup d'exemples négatifs avec très peu de traitements,
- le classifieur final est un simple classifieur linéaire.

L'ensemble des filtres de Haar pour une sous-fenêtre de détection représente un nombre considérable de filtres. Papageorgiou *et al.* [120] ont proposé de n'en conserver qu'un nombre limité mais la sélection était manuelle. Dans la méthode proposée

par Viola et Jones, la sélection des descripteurs les plus discriminants se fait de manière automatique. Nous allons présenter plus en détail les filtres de Haar, les images intégrales et la version d'Adaboost utilisée, basée sur la bibliothèque *OpenCV*.

3.5.1 Les filtres de Haar et les images intégrales

Les filtres de Haar sont également appelés *Haar wavelet features*, *Haar-like filters* ou *filtres rectangles*. La figure 3.9 présente les 12 filtres de Haar utilisés. Il y a deux types de filtres présentés ici, les premiers d'entre eux sont composés de régions adjacentes de même formes et de même tailles. La valeur du descripteur sera dans ce cas la différence entre la somme des pixels d'une région et la somme des pixels de l'autre région. Dans le cas où le descripteur est composé de trois régions adjacentes, on calcule la somme des pixels des deux régions extérieures que l'on soustrait à la région centrale.

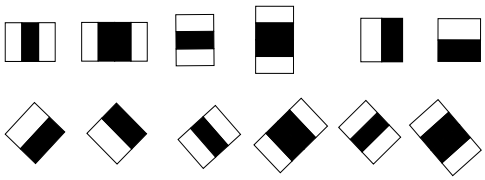


Figure 3.9: Filtres de Haar utilisés

Ces descripteurs peuvent être calculés très rapidement en utilisant les images intégrales. On calcule d'abord l'image intégrale de chaque image, et ensuite, il est possible de calculer la valeur des filtres de Haar à n'importe quelle position avec seulement quelques opérations. L'image intégrale Int, pour le pixel $s = (x, y)$ a pour valeur la somme des pixels au dessus et à gauche de s, autrement dit :

$$Int_s = \sum_{x' \leq x} \sum_{y' \leq y} I(x', y') \tag{3.12}$$

En utilisant les images intégrales, la somme des pixels d'une région se calcule avec seulement 4 valeurs. Si on prend l'exemple de la figure 3.10, la somme des pixels de la région A peut simplement être calculée en effectuant $Int_{s4} - Int_{s2} - Int_{s3} + Int_{s1}$.

En pratique, pour calculer la valeur d'un filtre de Haar composé de 2 rectangles adjacents, on a besoin de récupérer la valeur de 6 pixels de l'image intégrale et 8 pixels dans le cas d'un filtre composé de 3 rectangles.

3.5.2 Adaboost

La combinaison de tous les descripteurs à toutes les positions et tailles possibles fait qu'il existe pour une fenêtre de détection de taille relativement petite un très grand nombre de descripteurs (environ 110000 pour une fenêtre de taille 24×24

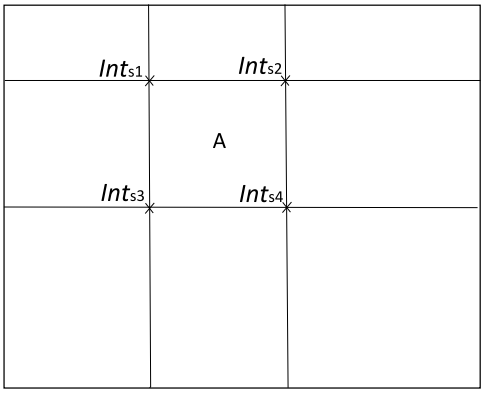


Figure 3.10: Exemple d'utilisation des images intégales. La somme des pixels dans la région A se calcule simplement par $Int_{s4} - Int_{s2} - Int_{s3} + Int_{s1}$.

[97]). *Adaboost* permet de sélectionner automatiquement les descripteurs les plus discriminants.

Le principe du boosting est de combiner des classifieurs faibles pour former un classifieur robuste. Dans la méthode de Viola et Jones [148], le classifieur faible est le seuillage de la valeur d'un descripteur :

$$f_j(x) = \begin{cases} 1 & \text{si } \varpi_j(x) p_j \geq \tau_j p_j \\ 0 & \text{sinon,} \end{cases} \tag{3.13}$$

où $f_j(x)$ représente le $j^{\text{ème}}$ classifieur faible de la fenêtre de détection x et ϖ_j est la valeur du $j^{\text{ème}}$ filtre de Haar, p_j est un terme de parité indiquant le sens de l'inégalité. Il y a donc autant de classifieurs faibles que de descripteurs possibles. La valeur du seuil τ_j est déterminée de façon à ce qu'un nombre minimum d'exemples soient mal classifiés sur la base d'apprentissage. Lors de l'apprentissage, un classifieur robuste (ou classifieur boosté) F est ensuite construit avec une somme pondérée de plusieurs classifieurs faibles. La démarche est décrite dans le tableau 3.2.

3.5.3 La cascade de classifieurs

On a vu précédemment la construction d'un classifieur robuste avec une combinaison linéaire de classifieurs faibles. Viola et Jones [148] ont proposé d'utiliser une cascade de classifieurs boostés (cf. figure 3.11) pour rejeter beaucoup d'exemples négatifs très rapidement. Des classifieurs plus complexes sont ensuite utilisés une fois que la majorité des exemples négatifs a été rejetée. Dans la phase de reconnaissance, une fenêtre d'entrée est donc analysée successivement par chaque classifieur boosté F_i qui peut envoyer la fenêtre au classifieur suivant ou rejeter la fenêtre.

- Soit K le nombre de classifieurs faibles utilisé pour former le classifieur robuste
- Soit J le nombre de descripteurs
- Soit N exemples labellisés $\{(x_1, y_1), \ldots, (x_n, y_n), \ldots, (x_N, y_N)\}$ avec $y_n = \{0, 1\}$ pour respectivement les exemples négatifs et positifs,
- On initialise les poids $w_{1,n} = \frac{1}{2m}, \frac{1}{2l}$ pour $y_n = 0, 1$ respectivement. m et l sont les nombres d'exemples négatifs et positifs.
- **Pour** $k = 1 \ldots K$ **Faire**
  - Normaliser les poids tel que $\sum_{n=1}^{N} w_{k,n} = 1$
  - **Pour** $j = 1 \ldots J$ **Faire**
    - On calcule $f_j(x_n)$ pour tous les exemples x_n.
    - On évalue les erreurs $\varepsilon_j = \sum_{n=1}^{N} w_{k,n} |f_j(x_n) - y_n|$ pour chaque classifieur faible en tenant compte des poids de chaque exemple.
  - **Fin Faire**
  - On choisit le classifieur f_j avec l'erreur ε_j la plus faible
  - On met à jour les poids $w_{k+1,n} = w_{k,n}.\beta_k^{1-e_n}$, où $e_n = 0$ si l'exemple x_n est classifié correctement, $e_n = 1$ autrement, et $\beta_k = \frac{\varepsilon_k}{1-\varepsilon_k}$
- **Fin Faire**

Le classifieur final est $F(x) = \begin{cases} 1 & \text{si } \sum_{k=1}^{K} \alpha_k f_k(x) \geq \frac{1}{2} \sum_{k=1}^{K} \alpha_k \\ 0 & \text{sinon} \end{cases}$

où $\alpha_k = \log \frac{1}{\beta_k}$

Table 3.2: Algorithme Adaboost. Construction d'un classifieur boosté.

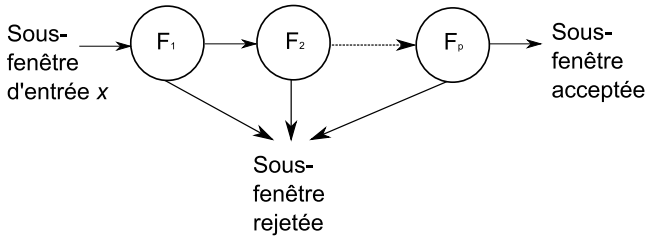


Figure 3.11: Cascade de classifieurs boostés

3.5.4 Fenêtre glissante et fusion de résultats de détection

Comme expliqué précédemment, nous cherchons désormais à connaître la nature de l'objet suivi. Est-ce que nous suivons un humain ? Si la soustraction de l'arrière-plan permettait de détecter et de dissocier avec certitude et précision tous les objets et si le classifieur permettait de déterminer avec certitude la nature de l'objet suivi, il suffirait d'analyser avec le classifieur la boîte englobante détectée par la soustraction de l'arrière-plan. Mais un objet détecté peut représenter plusieurs personnes et le classifieur n'est pas infaillible.

Une région d'intérêt autour de l'objet suivi est donc définie avec une marge d de chaque côté de la boîte englobante détectée. Cette région d'intérêt est parcourue par le classifieur (ou cascade de classifieurs boostés) à différentes positions et échelles. La figure 3.12 présente la boîte englobante de l'objet détecté et la région d'intérêt

parcourue par le classifieur.

De manière pratique, le classifieur parcourt la région d'intérêt avec un décalage de 2 pixels dans la direction horizontale et 2 pixels dans la direction verticale. Comme la taille des personnes présentes n'est *a priori* pas connue et le classifieur à une taille fixe (par exemple 12×36), la région d'intérêt est parcourue plusieurs fois en modifiant son échelle. La taille de la région d'intérêt est divisée par un facteur de 1.2 entre deux échelles.

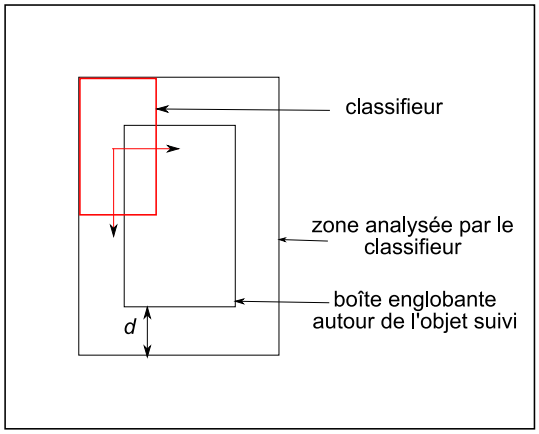


Figure 3.12: Illustration de la région d'intérêt parcourue par le classifieur.

En utilisant une fenêtre de détection glissante à plusieurs échelles et positions, il y a logiquement plusieurs détections se chevauchant pour représenter une seule personne. La figure 3.13 présente un exemple dans lequel plusieurs détections représentent la même personne. Pour fusionner les détections qui se chevauchent, Gu *et al.* [66] utilisent par exemple l'algorithme du *Mean-Shift* [30]. Pour des raisons de rapidité de calcul, nous utilisons simplement la moyenne des détections qui s'intersectent. Le critère de la compétition Pascal [46] est utilisé pour détecter les intersections :

$$Pas(B_i, B_j) = \frac{card(B_i \cap B_j)}{card(B_i \cup B_j)} \tag{3.14}$$

où B_i et B_j représentent les boîtes englobantes des détections i et j. Si $Pas(B_i, B_j) > \tau_3$, les détections i et j sont fusionnées. La nouvelle détection aura pour centroïde, largeur et hauteur la moyenne des centroïdes, largeurs et hauteurs des détections i et j.

Le nombre de détections fusionnées sera utilisé pour déterminer un indice de confiance. Il est également possible d'obtenir le nombre de personnes présentes dans une région d'intérêt à partir du nombre de détections qui n'ont pas été fusionnés.

3.5.5 Classification par parties

Dans le cadre d'une utilisation en milieu intérieur, les occultations partielles sont fréquentes. Il est donc clairement insuffisant de ne rechercher dans la région d'intérêt

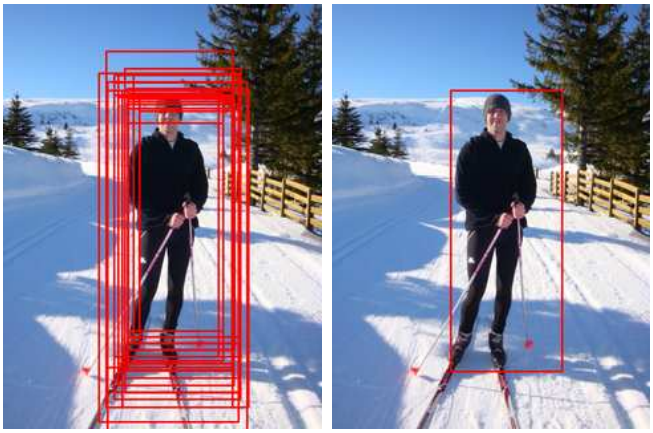


Figure 3.13: Exemple de fusion de résultats de détection..

que des formes similaires au corps humain dans son intégralité. Le haut du corps (tête + épaules par exemple) est souvent la seule partie du corps visible.

D'un point de vue système, il peut donc paraître suffisant de ne rechercher que la partie supérieure du corps. Cependant, nous avons montré dans la partie 2.2.2.2 que la détection sur le corps entier présente de meilleures performances que la détection de la tête et des épaules. Nous avons donc choisi de rechercher les deux parties dans la région d'intérêt.

En pratique, quatre classifieurs sont utilisés :

- le corps entier (indépendamment du point de vue),
- la tête et les épaules (vue de face ou de dos),
- la tête et les épaules (vue de gauche),
- la tête et les épaules (vue de droite).

Il est important d'utiliser plusieurs classifieurs pour la partie supérieure du corps. En effet, nous avons empiriquement remarqué qu'un seul classifieur pour la tête et les épaules, sous tous les points de vue, n'était pas suffisamment robuste. En pratique, le classifieur "vue de gauche" est le symétrique du classifieur "vue de droite". Les résultats de ces différents classifieurs sont utilisés pour la construction de l'indice de confiance. Nous montrons un exemple dans la figure 3.14 où trois des quatres classifieurs sont utilisés, chaque couleur représente le résultat de détection d'un classifieur différent (le rectangle bleu correspond au classifieur tête et épaules vue de droite, le rectangle vert correspond au classifieur tête et épaules vue de gauche et le rectangle rouge correspond au classifieur du corps entier).

En pratique, pour le classifieur du corps entier, nous utilisons un classifieur de taille 12×36 composé d'une cascade de 27 classifieurs boostés et 4457 classifieurs faibles au total. Ce classifieur a été entraîné sur la base proposée par l'INRIA pour laquelle nous avons réduit la taille du contexte autour de chaque personne. Les classifieurs de la partie supérieure du corps ont été entraînés par *ST Imaging* sur leur propre base d'apprentissage. Ils sont de taille 20×20. Le détecteur vue de face est composé de 23 étages et 1549 classifieurs faibles au total, le classifieur vue de profil est composé de 22 étages et de 1109 classifieurs faibles au total.

Figure 3.14: Exemple de détection où trois classifieurs ont été utilisés. Chaque couleur représente le résultat de détection d'un classifieur différent - le rectangle bleu correspond au classifieur tête et épaules vue de droite, le rectangle vert correspond au classifieur tête et épaules vue de gauche et le rectangle rouge correspond au classifieur du corps entier.

3.5.6 Indice de confiance

Nous avons vu précédemment, dans la partie 3.5.4, qu'une détection était une superposition de plusieurs détections. Il est donc possible de construire un indice de confiance $\mathcal{Y}_{i,t}^k$, correspondant à la fiabilité de la réponse du classifieur k, à l'instant t et pour l'objet i. Cette indice dépend du nombre de détections $\gamma_{i,t}^k$:

$$\mathcal{Y}_{i,t}^k = \min(1, \frac{\gamma_{i,t}^k}{\nu}) \tag{3.15}$$

où ν est le nombre minimal de détections, déterminé empiriquement, pour que l'indice de confiance du classifieur k soit maximal, c'est à dire $\mathcal{Y}_{i,t}^k = 1$. Pour une personne, il y a quatre indices de confiance $\mathcal{Y}_{i,t}^1$, $\mathcal{Y}_{i,t}^2$, $\mathcal{Y}_{i,t}^3$ et $\mathcal{Y}_{i,t}^4$ correspondant à la fiabilité de la détection du classifieur du corps entier, de la partie supérieure du corps vue de face, de droite et de gauche.

Puisqu'il est possible de suivre indépendamment chaque objet de la scène, il est possible d'affecter une étiquette consistante dans le temps à tous les objets suivis. On construit alors un indice de confiance $\mathcal{I}_{i,t}$ qui dépend de l'indice de confiance de ce même objet i à l'instant précédent et aux indices de confiance des détecteurs à l'instant courant :

$$\mathcal{I}_{i,t} = \min(1, \mathcal{I}_{i,t-1} + \alpha_1(\mathcal{Y}_{i,t}^1 - \mathcal{I}_{i,t-1}) + \alpha_2(\mathcal{Y}_{i,t}^{2'} - \mathcal{I}_{i,t-1}) - \alpha_3\mathcal{I}_{i,t-1}) \tag{3.16}$$

où $\mathcal{Y}_{i,t}^{2'} = \max(\mathcal{Y}_{i,t}^2, \mathcal{Y}_{i,t}^3, \mathcal{Y}_{i,t}^4)$.

Cet indice de confiance varie entre 0 et 1 ; 1 étant l'indice de confiance maximal. α_1, α_2 et α_3 sont trois seuils permettant de contrôler la rapidité de prise en compte

des nouveaux indices. On imposera $\alpha_1 = 0$ s'il n'y a aucune détection du corps entier à cause d'une occultation partielle et $\alpha_3 = 0$ s'il y a au moins une détection.

Finalement, un simple seuillage de cet indice nous permet de déterminer la nature de l'objet i :

$$\mathcal{I}_{i,t} \underset{non-humain}{\overset{humain}{\gtrless}} \tau_4. \tag{3.17}$$

3.6 Validation du système proposé

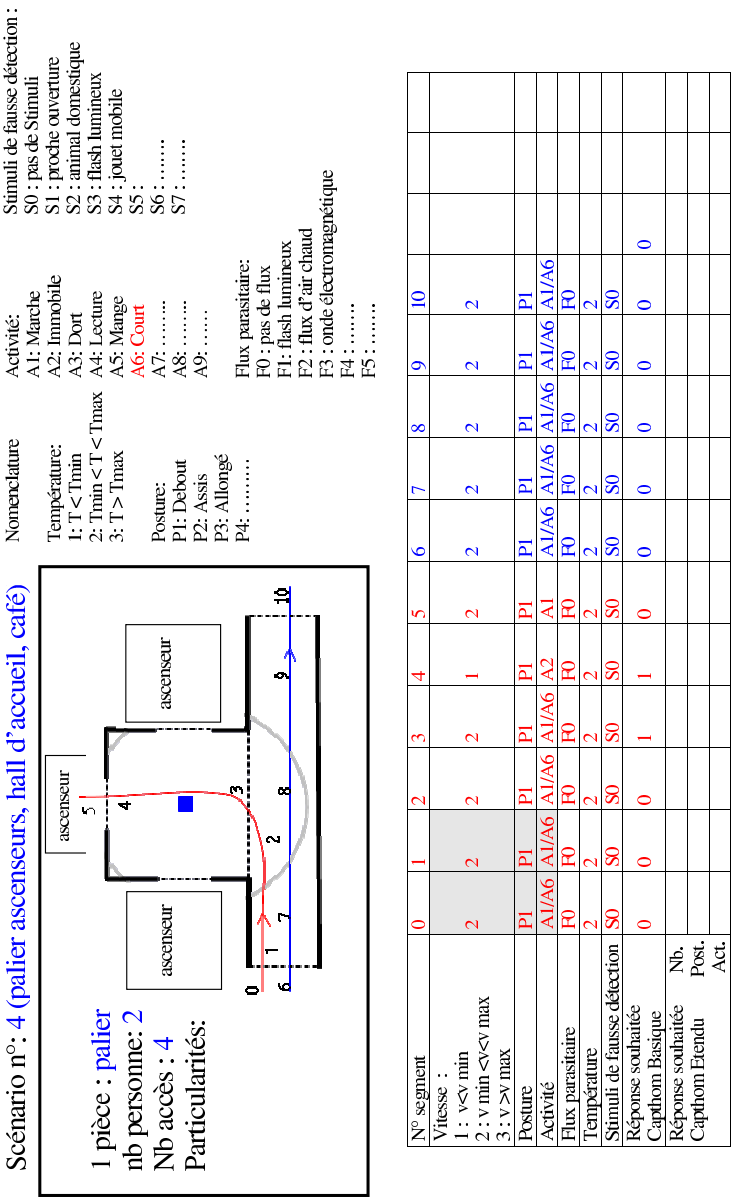
Nous avons proposé ci-dessus une méthode pour détecter des personnes dans une séquence d'images. Nous présentons désormais des résultats expérimentaux validant cette méthode.

3.6.1 Protocole expérimental

Afin d'évaluer le système proposé, les partenaires du consortium ont exprimé leurs besoins spécifiques à travers un ensemble de scénarios de référence auxquels le capteur, en l'occurrence l'algorithme proposé ici, doit répondre adéquatement. Un exemple de scénario, avec le formalisme utilisé, est présenté dans la figure 3.15. Le formalisme et la synthèse de tous les scénarios ont été proposés par Pierre David [36], doctorant au sein du projet *CAPTHOM*, dans le cadre de ses travaux de thèse. Il a ainsi proposé trois classes de scénarios à partir desquelles nous avons construit la base d'évaluation. Les trois classes sont les suivantes :

1. les scénarios d'usage normal d'occupation d'une pièce. Dans ces scénarios, on peut trouver une ou plusieurs personnes statiques ou en mouvement, assises ou debout. Nous avons fait 14 vidéos dans 9 endroits différents (bureaux, salles de réunion, couloirs et salles à manger).
2. Les scénarios d'activités inhabituelles (chutes lentes ou rapides, agitation). Nous avons ici réalisé 7 vidéos.
3. Les scénarios rassemblant tous les stimuli de fausses détections recensés par les partenaires du consortium (variation de l'illumination, objets en mouvement *etc.*). Nous avons ici réalisé 8 vidéos.

Nous avons donc réalisé 29 vidéos dans 10 endroits différents. Les films ont une résolution de 320×240 et ont une qualité "moyenne" puisqu'ils ont été acquis avec une simple webcam.



N° segment	0	1	2	3	4	5	6	7	8	9	10							
Vitesse : 1 : v<v min 2 : v min <v<v max 3 : v >v max	2	2	2	2	1	2	2	2	2	2	2							
Posture	P1	P1	P1	P1	P1	P1	P1	P1	P1	P1	P1							
Activité	A1/A6	A1/A6	A1/A6	A1/A6	A2	A1	A1/A6	A1/A6	A1/A6	A1/A6	A1/A6							
Flux parasitaire	F0	F0	F0	F0	F0	F0	F0	F0	F0	F0	F0							
Température	2	2	2	2	2	2	2	2	2	2	2							
Stimuli de fausse détection	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0	S0							
Réponse souhaitée Capthom Basique	0	0	0	1	1	0	0	0	0	0	0							
Réponse souhaitée Capthom Etendu	Nb.																	
	Post.																	
	Act.																	

Figure 3.15: Exemple de scénario établi par les partenaires du projet *CAPTHOM*.

3.6.2 Évaluation de la détection de présence

Dans le cadre de cette première évaluation, nous nous intéressons à la capacité de l'algorithme proposé de fournir une information sur la présence ou l'absence de personne dans l'angle de vue de la caméra. L'algorithme présenté ci-dessus est comparé avec :

- **IRP** : un détecteur de présence du commerce (technologie infrarouge passif),
- **Haar-Boost** : le système de détection de Viola et Jones [148] avec une fenêtre

glissante parcourant chaque image sans information *a priori*,
- **Haar-Boost + S-AP** : variation de la méthode *Haar-Boost* où l'espace de recherche est réduit avec une soustraction de l'arrière-plan (*S-AP*).

Les performances obtenues avec les trois méthodes de vision sont bien entendu extrêmement dépendantes du facteur d'échelle présenté dans 3.5.4. Les algorithmes ont été utilisés avec différentes valeurs de ce facteur d'échelle, et seul les meilleurs résultats pour chaque méthode sont présentés ici.

Les résultats sont présentés à l'aide de la valeur maximale du *f-score* et de la matrice de confusion :

	+	-
+	a	b
-	c	d

Table 3.3: Exemple de matrice de confusion.

où

- a correspond au nombre de bonnes détections (vrais positifs),
- b correspond au nombre de détections manquées (faux négatifs),
- c correspond au nombre de fausses détections (faux positifs),
- d correspond au nombre de vrais négatifs.

Les résultats sont présentés en pourcentage :

	+	-
+	$a/(a+b)$	$b/(a+b)$
-	$c/(c+d)$	$d/(c+d)$

Table 3.4: Présentation des résultats avec des pourcentages dans la matrice de confusion.

Le placement, le paramétrage, l'acquisition et le dépouillement des résultats de détection du détecteur *IRP* ont été réalisés par Antoine Belconde, doctorant au sein du projet *CAPTHOM*.

Test 1 : Usage normal Les premiers résultats présentés ci-dessous correspondent aux scénarios d'usage normal d'une pièce. Les résultats sont présentés dans les tableaux 3.5 et 3.6.

Avec ce premier test, nous pouvons remarquer que les résultats de notre méthode sont totalement satisfaisants dans ce contexte. En effet, le nombre de faux positifs

	+	-
+	0.24	0.76
-	0.47	0.53

IRP

	+	-
+	0.82	0.18
-	0.60	0.40

Haar-Boost

	+	-
+	0.81	0.19
-	0.09	0.91

Haar-Boost + S-AP

	+	-
+	0.98	0.02
-	0.16	0.84

Méthode proposée

Table 3.5: Matrices de confusion obtenues sur les scénarios d'usage normal d'une pièce.

	IRP	Haar-Boost	Haar-Boost + S-AP	Méthode proposée
F-score	0.35	0.86	0.89	**0.98**

Table 3.6: F-score correspondant aux résultats obtenus sur les scénarios d'usage normal d'une pièce.

relativement élevé vient du faible nombre d'exemples négatifs dans cette base. Par exemple, sur une vidéo contenant une centaine d'images annotées, il peut y avoir seulement 4 ou 5 images où aucune personne n'est présente, une fausse détection sur une seule de ces images augmente donc de façon non représentative le pourcentage de fausses détections. La troisième base de vidéos contient quant à elle beaucoup d'images où personne n'est présent, les résultats globaux se compenseront donc. Le *f-score* de 0.98 est un très bon résultat.

Les résultats obtenus avec le détecteur *IRP* ne sont pas satisfaisants. En effet, celui-ci est un détecteur de mouvement et donc le nombre de faux négatifs est très élevé. Pour compenser cela, une temporisation est généralement utilisée. Seulement, le nombre de faux négatifs diminue lorsque cette temporisation est longue mais le nombre de faux positifs augmente alors.

Test 2 : Événements anormaux Nous présentons maintenant les résultats obtenus avec la deuxième base de scénarios correspondant aux événements anormaux (chutes etc). Les résultats sont présentés dans les tableaux 3.7 et 3.8.

	+	-
+	0.17	0.83
-	0.21	0.79

IRP

	+	-
+	0.48	0.52
-	0.17	0.83

Haar-Boost

	+	-
+	0.75	0.25
-	0.01	0.99

Haar-Boost + S-AP

	+	-
+	0.99	0.01
-	0.07	0.93

Méthode proposée

Table 3.7: Matrices de confusion obtenues sur les scénarios d'événements anormaux.

Plusieurs remarques peuvent être tirées de ces résultats :

	IRP	Haar-Boost	Haar-Boost + S-AP	Méthode proposée
F-score	0.27	0.64	0.85	**0.99**

Table 3.8: F-score correspondant aux résultats obtenus sur les scénarios d'événements anormaux.

- tout d'abord, les résultats obtenus avec la méthode proposée sont très satisfaisants,
- ensuite, les différences entre les méthodes sont ici clairement identifiables. Dans cette base, il y a 4 scénarios dans lesquels une personne chute et reste quelques instants couchée au sol. Les classifieurs utilisés ne sont pas capable de reconnaître une personne couchée. Mais grâce au suivi, notre méthode est capable de reconnaître une personne couchée.
- Les résultats obtenus avec le détecteur *IRP* sont encore une fois bien en dessous de ceux obtenus par les autres méthodes. Dans les scénarios impliquant des chutes, les personnes restent statiques et ne peuvent donc pas être détectées par cette technologie.

Test 3 : Stimuli de fausses détections Nous présentons ensuite les résultats d'un troisième test correspondant aux scénarios de stimuli de fausses détections (variations d'illumination, *etc.*). Les résultats sont présentés dans les tableaux 3.9 et 3.10.

	+	-
+	0.40	0.60
-	0.31	0.69

IRP

	+	-
+	0.83	0.18
-	0.60	0.40

Haar-Boost

	+	-
+	0.83	0.18
-	0.01	0.99

Haar-Boost + S-AP

	+	-
+	0.89	0.11
-	0.01	0.99

Méthode proposée

Table 3.9: Matrices de confusion obtenues sur les scénarios de stimuli de fausses détections.

	IRP	Haar-Boost	Haar-Boost + S-AP	Méthode proposée
F-score	0.5	0.3	0.88	**0.92**

Table 3.10: F-score correspondant aux résultats obtenus sur les scénarios de stimuli de fausses détections.

Dans cette base, il y a peu d'exemples où une personne est présente, ceci peut expliquer le nombre très important de fausses détections (en pourcentage) de *Haar-Boost*. Les résultats obtenus avec la méthode proposée sont à nouveau satisfaisants. Grâce aux différentes façons de mettre à jour le modèle de l'arrière-plan, nous sommes

capables de gérer les différentes variations d'illumination et l'étape de reconnaissance permet de ne pas détecter les objets mobiles qui ne sont pas des humains.

Test 4 : toute la base Finalement, nous présentons les résultats obtenus sur l'ensemble des scénarios. Les résultats sont présentés dans les tableaux 3.11 et 3.12.

	+	-
+	0.40	0.60
-	0.31	0.69

IRP

	+	-
+	0.77	0.23
-	0.58	0.42

Haar-Boost

	+	-
+	0.77	0.23
-	0.03	0.97

Haar-Boost + S-AP

	+	-
+	0.97	0.03
-	0.03	0.97

Méthode proposée

Table 3.11: Matrices de confusion obtenues sur l'ensemble des scénarios.

	IRP	Haar-Boost	Haar-Boost + S-AP	Méthode proposée
F-score	0.50	0.67	0.83	**0.97**

Table 3.12: F-score correspondant aux résultats obtenus sur l'ensemble des scénarios.

Les résultats présentés dans ces deux tableaux permettent de formuler quelques remarques. Tout d'abord, il apparaît clairement ici que le détecteur *IRP* ne présente pas des performances suffisantes. Son principal défaut est qu'il ne peut détecter que les variations de température et donc que les mouvements des personnes. Dans beaucoup de scénarios, il y a donc beaucoup de détections manquées (faux négatifs). Ensuite, avec *Haar-Boost*, même si les résultats sont sensiblement meilleurs, les performances obtenues n'apportent pas une rupture avec les détecteurs du commerce. Le nombre de faux négatifs et de faux positifs est trop élevé. Avec *Haar-Boost + S-AP*, le nombre de faux positifs est très nettement diminué. En effet, avec cette méthode, la soustraction de l'arrière-plan permet de réduire l'espace de recherche du classifieur et donc il y a logiquement moins de fausses détections. Enfin, avec la méthode proposée qui utilise également le suivi pour avoir un historique des déplacements des personnes, nous sommes capables de "lisser" dans le temps les détections et donc de détecter une personne même si à un instant donné le classifieur ne peut pas reconnaître la personne (personne couchée, contraste insuffisant, *etc.*).

Nous avons ici démontré les avantages apportés par l'utilisation de la soustraction de l'arrière-plan et du suivi. Les performances de la méthode proposée sont bonnes puisque cette méthode présente, pour un taux de détection de 97%, un taux de fausses détections d'environ 3%.

3.6.3 Évaluation globale

L'évaluation présentée ci-dessus ne considère que l'information de présence ou d'absence. Nous présentons ici, les résultats de l'évaluation en considérant le nombre

de personnes détectées, leur position dans l'image et la précision de leur localisation. La méthode utilisée est issue des travaux de Hemery *et al.* [72]. Elle est composée de quatre étapes, à savoir :

1. calcul de la correspondance entre la vérité terrain et les résultats de détection,
2. évaluation de la localisation,
3. compensation de la sous-détection et de la sur-détection,
4. calcul du score global.

La première étape est donc le calcul de la correspondance entre les objets de la vérité terrain et les résultats de détection. À l'instar de l'algorithme de suivi présenté ci-dessus, nous construisons une matrice de correspondance $\mathcal{H}$, définie par :

$$\mathcal{H} = \begin{pmatrix} \beta_{1,1} & \cdots & \beta_{1,N} \\ \vdots & \ddots & \vdots \\ \beta_{M,1} & \cdots & \beta_{M,N} \end{pmatrix} \tag{3.18}$$

où le nombre de lignes M correspond au nombre d'objets dans la vérité terrain, et le nombre de colonnes N correspond au nombre de personnes détectées. Chaque élément de la matrice de correspondance $\beta_{i,j}$ représente la valeur du recouvrement entre l'objet i de la vérité terrain A_i et l'objet j de la détection B_j définie par :

$$\beta_{i,j} = \frac{card(A_i \cap B_j)}{card(A_i \cup B_j)} \tag{3.19}$$

Nous considérons ici les boîtes englobantes des détections. La matrice de correspondance $\mathcal{H}$ pour les résultats de détection présentés dans l'image 3.16 est :

$$\mathcal{H} = \begin{pmatrix} 0 & 0.87 & 0 & 0 \\ 0 & 0 & 0.75 & 0 \\ 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0.6 \end{pmatrix} \tag{3.20}$$

Il est possible qu'une détection corresponde à plusieurs objets dans la vérité terrain. En effet, puisque l'algorithme que nous nous proposons d'évaluer ici est très largement basé sur la soustraction de l'arrière-plan, ce cas de figure est relativement fréquent dès lors que les deux personnes sont très proches dans l'image.

La correspondance est ensuite réalisée en effectuant simplement un seuillage de chaque élément de la matrice. Pour l'exemple de la figure 3.16, nous obtenons la matrice de correspondance suivante :

$$\mathcal{H} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.21}$$

Figure 3.16: Exemple de résultat de détection.

Ensuite, si $\beta_{i,j} = 1$, nous évaluons la précision de la localisation avec la mesure de Martin [106] entre l'objet i de la vérité terrain A_i et l'objet j de la détection B_j. En effet, il a été montré dans les travaux d'Hemery *et al.* [73] que cette métrique était l'une des plus fiables pour l'évaluation de la localisation. La mesure de Martin est définie par :

$$Mar(A_i, B_j) = 1 - \frac{1}{card(I)} \min\left(\frac{card(A_i \setminus B_j)}{card(A_i)}, \frac{card(B_j \setminus A_i)}{card(B_j)}\right) \tag{3.22}$$

Finalement, nous calculons la moyenne des valeurs de la mesure de Martin pour tous les $\beta_{i,j} = 1$ en considérant un 0 pour chaque sous-détection et chaque sur-détection. Dans l'exemple ci-dessus, la valeur du score global se calcule par :

$$S = \frac{0.87 + 0.75 + 0.6}{5} = 0.44. \tag{3.23}$$

Nous présentons dans le tableau 3.13 les moyennes des scores globaux obtenus sur chacune des bases de scénarios présentées dans la partie précédente.

	Haar-Boost	Haar-Boost + S-AP	Méthode proposée
Base 1	0.44	0.63	**0.67**
Base 2	0.47	0.53	**0.96**
Base 3	0.88	**0.93**	**0.93**
Base 1 + 2 + 3	0.51	0.7	**0.77**

Table 3.13: Résultats de l'évaluation globale.

Les performances obtenues avec notre méthode sont encore une fois supérieures aux résultats des deux autres méthodes. Cependant, les différences entre *Haar-Boost + S-AP* et la méthode proposée sont moins nettes que lors de l'évaluation précédente.

En effet, dans notre approche, ce sont les composantes connectées obtenues avec la soustraction de l'arrière-plan qui nous permettent de compter le nombre d'objets dans une image. Cependant, lorsque deux personnes sont proches dans une image, une seule composante connectée les représentera dans l'image. Dans le calcul du score global, notre méthode sera fortement pénalisée car il y aura une sous-détection et une mauvaise localisation, alors que les deux personnes sont détectées mais représentées par une seule composante connectée. Malgré cela, les performances de notre méthode restent meilleures que les deux autres méthodes.

3.6.4 Utilisation des ressources matérielles

Nous présentons, dans la figure 3.17, la répartition de l'utilisation du *CPU* entre les différents modules de l'algorithme présenté ci-dessus. Bien entendu, cette répartition est variable et dépend de la taille de la région d'intérêt analysée par les classifieurs. Si aucune région d'intérêt n'est détectée, la totalité du *CPU* sera utilisée par la soustraction de l'arrière-plan et les diverses opérations (acquisition, initialisation de variables *etc.*). Les résultats présentés dans la figure 3.17 représentent la répartition de l'utilisation du *CPU* lorsqu'il y a au moins une région d'intérêt à analyser. Les pourcentages obtenus sont une moyenne sur l'ensemble des vidéos (tailles et nombre des régions d'intérêt différents).

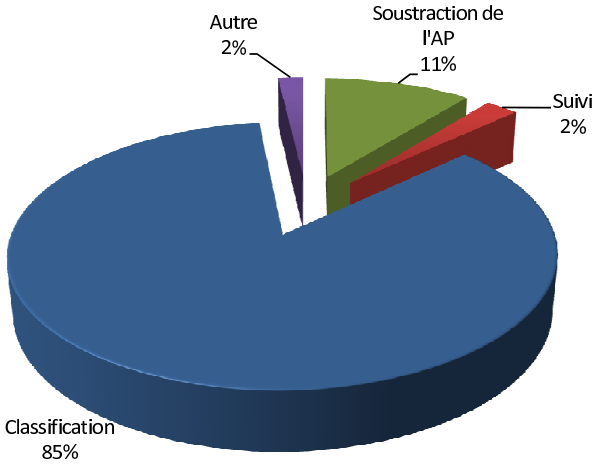


Figure 3.17: Répartition d'utilisation du CPU entre les différents modules de l'algorithme.

On peut remarquer que la grande majorité du temps de calcul est utilisée par le classifieur. En effet, nous utilisons finalement 4 classifieurs parcourant les régions d'intérêt. La soustraction de l'arrière-plan, qui effectue des traitements sur tous les pixels de l'image représente 11% des opérations et le suivi n'utilise que 2%. Cette répartition de l'utilisation des ressources est intéressante puisqu'elle permet de clairement identifier la reconnaissance comme offrant le plus de possibilités pour diminuer le temps d'exécution associé à chaque image. Une analyse dégradée devra, si besoin, permettre de trouver le nombre de classifieurs associé au compromis optimal entre

temps d'exécution et performance de détection. On peut noter que le temps d'exécution moyen, obtenu sur l'ensemble des vidéos, est d'environ 10 images par seconde sur un ordinateur portable standard (2 GHz).

3.6.5 Étude qualitative des résultats et discussion

Nous présentons désormais dans les figures 3.18 à 3.22 des exemples de résultats. Ces exemples montrent des situations où l'algorithme a un comportement attendu.

Les résultats quantitatifs ont été présentés dans les parties 3.6.2 et 3.6.3. On peut remarquer que la soustraction de l'arrière-plan nous permet de détecter les personnes à des distances assez importantes. Par exemple, dans la deuxième image de la figure 3.18, une personne se trouvant à environ 20 mètres a été détectée dans une image de résolution seulement 320×240.

Figure 3.18: Exemple de résultat illustrant une scène de couloir avec une ou plusieurs personnes se déplaçant.

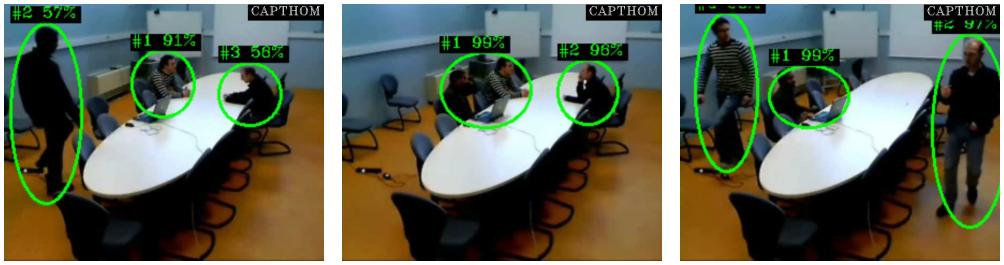


Figure 3.19: Exemple de résultat illustrant une scène de réunion avec des occlusions partielles.

Cependant, malgré les bons résultats quantitatifs présentés dans les parties 3.6.2 et 3.6.3, il reste principalement trois causes d'erreur avec la méthode proposée ci-dessus. Une erreur possible est lorsqu'une personne n'est pas détectée. Grâce au suivi

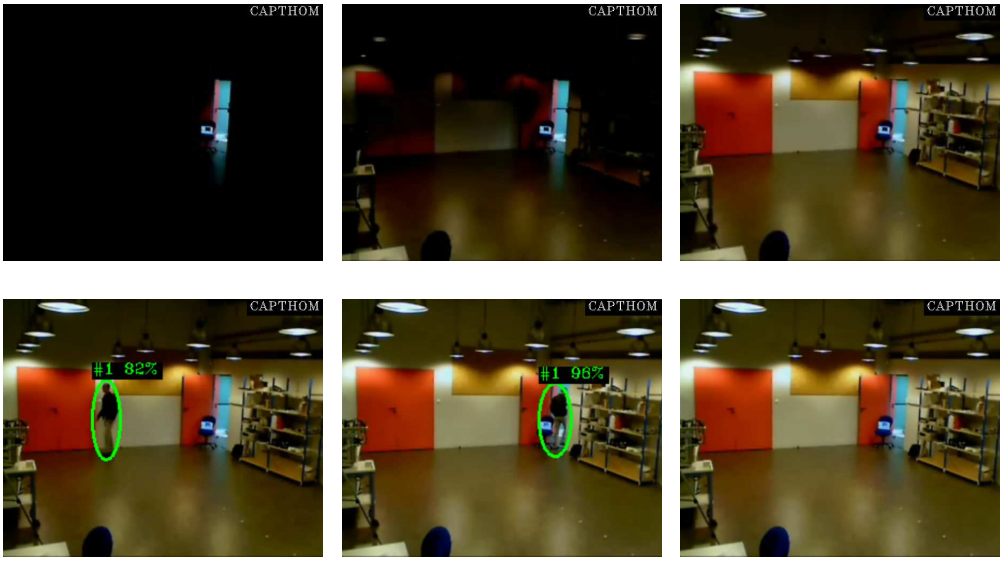


Figure 3.20: Exemple de résultat avec un changement brusque de l'illumination.

Figure 3.21: Exemple de résultat illustrant une scène de bureau avec occultation partielle.

des objets, ce cas est relativement rare puisque les détections sont lissées dans le temps avec l'indice de confiance. Cependant, il est possible qu'une personne reste statique dans une configuration où le classifieur ne la reconnaît pas. Les causes de cette détection manquée peuvent être que le contraste entre la personne et l'arrière-plan qui n'est pas très marqué, ou que la personne adopte une pose inhabituelle.

Ensuite, lorsqu'une personne reste statique longtemps, elle est petit à petit incluse dans le modèle de l'arrière-plan. Si le paramètre de rapidité de mise à jour de l'arrière-plan est convenablement fixé, ce n'est pas un problème. Cependant, lorsque la personne se déplacera à nouveau, on verra apparaître son "fantôme" à l'endroit où elle était stationnée. Ce fantôme sera inclus dans le modèle de l'arrière-plan grâce à la mise à jour au niveau "objet" du modèle cependant, le temps de latence entre le départ de la personne et la mise à jour du modèle de l'arrière-plan peut entraîner une fausse détection.

La troisième cause d'erreur possible vient directement d'une erreur lors de la soustraction de l'arrière-plan. Il peut se produire une fausse détection due à une variation de la scène (illumination *etc.*) ou une détection manquée ou partielle (généralement causée par une différence entre la personne et l'arrière-plan trop faible).

Figure 3.22: Exemple de résultat illustrant des variations brusques de luminosité.

La gestion des différents seuils et paramètres utilisés par cet algorithme est très important et peut être délicat. Il y a trois types de paramètres.

- Il y a tout d'abord les paramètres qui peuvent être fixés à une valeur par défaut qui sera valable pour tous les environnements. Par exemple, le seuil τ_2 de l'équation 3.8, utilisé pour le suivi des points d'intérêt, peut être fixé à 25% du nombre de points d'intérêt par objet. Le seuil τ_3, utilisé pour fusionner les résultats de détection peut être fixé à $0,7$.
- Ensuite, il y a les seuils qui dépendent directement des performances des différents classifieurs (les paramètres ν de l'équation 3.15, α_1, α_2 et α_3 de l'équation 3.16 et τ_4 de l'équation 3.17). Les classifieurs étant génériques, ces paramètres peuvent être fixés une fois pour tous les environnements. Mais il est également possible, pour répondre aux contraintes d'un environnement spécifique, d'ajuster ces paramètres afin de trouver le compromis entre fausses détections et détections manquées le plus adapté. Le paramètre α de l'équation 3.4 utilisé pour la mise à jour de l'arrière-plan, pourra également être fixé à une valeur par défaut mais les performances du système seront meilleures si ce paramètre est modifié en fonction de l'environnement. Par exemple, dans un bureau où les personnes restent statiques longtemps, ce seuil devra être très faible et inversement, dans un couloir, ce seuil pourra avoir une valeur élevée.
- Il y a enfin le seuil τ_1, utilisé dans l'équation 3.3 pour la soustraction de l'arrière-

plan, dont la valeur choisie est très importante car elle aura une influence très grande sur le fonctionnement du système global. La valeur de ce seuil devra, si possible, être ajustée lors de chaque installation du *CAPTHOM*.

Au final, même si la soustraction de l'arrière-plan nous permet de grandement augmenter les performances globales du système, celle-ci reste la cause d'une grande partie des erreurs. Dans de futurs travaux, il pourrait être intéressant de disposer d'un détecteur suffisamment robuste pour ne plus utiliser la soustraction de l'arrière-plan. Dans le cas présent, le classifieur est générique et s'adapte à tous les environnements. Dans le cas d'utilisation de *CAPTHOM*, le capteur étant statique, il est imaginable d'inclure une procédure d'auto-adaptation du classifieur par rapport à l'environnement.

Une autre piste intéressante pour la suite des travaux serait de travailler sur un descripteur qui ne se base pas sur les images intégrales tout en étant aussi rapide à calculer que les filtres de Haar. En effet, les images intégrales sont beaucoup plus gourmandes en mémoire que les images intensités. On peut imaginer par exemple un descripteur calculant des différences d'intensité entre des paires de pixels et non plus des régions.

3.7 Conclusion

Nous avons présenté ci-dessus la méthode de détection de personnes dans des séquences d'images proposée dans le cadre du projet *CAPTHOM*. Cette méthode a été largement évaluée en se basant sur un ensemble de scénarios établi à partir des spécifications des partenaires du consortium. Les résultats obtenus permettent d'avoir un taux de détection de 97% pour un taux de fausses détections et de détections manquées de 3%. Ces résultats sont bons grâce à la soustraction de l'arrière-plan qui permet de réduire l'espace de recherche du classifieur et donc de réduire les fausses détections. Ensuite, les différents niveaux de mises à jour du modèle de l'arrière-plan permettent de gérer la plupart des cas les plus courants de variation d'illumination, d'introduction ou de retrait d'objets... Enfin, le suivi des objets dans le plan image nous permet de construire un indice de confiance sur la reconnaissance qui évolue dans le temps et permet donc de limiter les détections manquées ponctuelles. Par exemple, si une personne adopte à un moment une posture inhabituelle qui ne peut être reconnue par aucun des classifieurs, grâce à cet indice de confiance, nous sommes tout de même capables de la détecter.

S'il est important d'avoir une information sur la présence d'une personne dans un environnement, la vision permet d'obtenir également d'autres informations utiles au système de gestion technique du bâtiment. Nous présentons dans le chapitre suivant plusieurs exemples d'applications s'intégrant dans le projet *CAPTHOM*. En effet, pour des applications de maintien à domicile ou de régulation du chauffage, il peut être intéressant de connaître la position dans l'espace et l'activité des personnes détectées. Il peut également être intéressant de pouvoir détecter automatiquement une situation anormale (chute *etc.*).

CHAPITRE 4

Applications de CAPTHOM

Ce chapitre présente quelques applications possibles d'un capteur de présence humaine et propose des méthodes basées sur l'analyse vidéo pour répondre à des besoins spécifiques. Nous présentons dans un premier temps une mesure de l'activité des personnes pouvant par exemple être utile pour adapter le chauffage dans un bâtiment. Ensuite, nous présentons un système de stéréovision multicapteurs basé sur une caméra infrarouge et une caméra dans le spectre visible. Cette combinaison de deux technologies permet de limiter le nombre de fausses détections inhérentes à l'utilisation de chacune des technologies. L'utilisation d'un système de stéréovision nous permet également de localiser dans l'espace les personnes détectées. Enfin, nous présentons une méthode pour modéliser et détecter les événements anormaux dans une vidéo en se basant sur des statistiques bas-niveau concernant les co-occurrences de pixels de l'avant-plan. Cette méthode de détection est générique et peut s'appliquer à la détection de plusieurs types d'activités anormales : détection d'intrusion (sécurité des biens), détection de chute (sécurité des personnes) etc.

4.1 Introduction

L'information de présence ou d'absence d'une personne est une information cruciale pour beaucoup d'applications. Nous avons présenté dans le chapitre précédent une méthode pour obtenir cette information à partir d'une séquence d'images. Cependant, un capteur autonome avec une caméra et une unité de traitement permet d'obtenir des informations de plus haut-niveau et donc d'étendre le champ des applications possibles. Nous présentons ci-dessous quelques unes de ces applications.

La régulation automatique du chauffage est une première application pour laquelle l'information de présence ou d'absence de personne dans la pièce est importante mais n'est pas forcément suffisante. En effet, on peut aisément concevoir que l'apport calorifique nécessaire à une régulation automatique de la température d'une pièce sera différent en fonction du nombre et de l'activité des personnes détectées. Il est donc intéressant de pouvoir caractériser cette activité. Cette caractérisation n'implique pas forcément une compréhension au niveau sémantique de l'activité des personnes mais plus simplement une mesure de l'activité ou de l'agitation.

Le maintien à domicile est un deuxième exemple pour lequel l'information de présence n'est pas toujours suffisante. Même si la détection de présence pour la gestion automatique de l'éclairage est très importante dans beaucoup de situations de maintien à domicile (personnes à mobilité réduite *etc.*), il peut être intéressant de connaître la position de la personne dans l'espace et sa posture. La combinaison de ces deux informations peut par permet de détecter des situations anormales. Une personne étendue sur son lit est une situation ordinaire alors qu'une personne étendue par terre peut certainement correspondre à une situation d'alerte.

Il existe plusieurs types de solutions technologiques pour la détection de chutes. Le premier correspond à l'ensemble des technologies passives où la personne porte un détecteur (*e.g.* un bracelet) sensible aux chocs, aux mouvements brusques ou capable de détecter une période d'inactivité (par exemple le $iLife^{TM}$ *Fall Detection Sensor* [3]). Ce composant peut être relié à un réseau par connexion sans fil pour déclencher des alertes. Il existe ensuite les méthodes dites "actives", pour lesquelles la personne porte un appareil spécifique et doit déclencher elle-même un signal d'alerte. La vision peut être également utilisée pour détecter des situations de détresse, comme la chute, de manière totalement automatique, sans utiliser d'appareil spécifique mais en intégrant un module supplémentaire au *CAPTHOM*.

Dans le cadre du maintien à domicile, il peut aussi être utile de connaître l'identité des personnes présentes. En effet, il peut être intéressant de pouvoir faire la distinction entre le personnel soignant et les patients. De même, pour éviter que des personnes désorientées se retrouvent dans des endroits où elles ne sont pas supposées être, on a besoin de connaître leur identité. Les caméras, avec les techniques de biométrie, sont une possibilité d'identification à distance et sans contact.

La sécurité et la protection des biens et des personnes sont également deux applications pour lesquelles un capteur autonome de type *CAPTHOM* peut être utilisé. Dans ce contexte, le *CAPTHOM* doit être capable de renseigner sur l'identité des personnes présentes mais également de détecter des situations dangereuses suite par exemple à la détection d'intrusion ou d'un colis abandonné dans un bâtiment.

Nous présentons dans la suite de ce chapitre quelques extensions possibles du *CAPTHOM* défini au chapitre précédent, permettant d'accéder à des informations de plus haut-niveau. Il s'agit d'une méthode pour la caractérisation de l'activité, d'un système de stéréovision multicapteurs et d'une méthode générique de détection d'événements anormaux.

4.2 Caractérisation de l'activité

Comme nous l'avons expliqué ci-dessus, dans un objectif de réguler automatiquement le chauffage, il peut être intéressant de quantifier l'activité des personnes présentes dans une pièce. Pour cela, nous proposons un critère très simple à calculer qui peut être aisément implémenté dans un matériel embarqué.

4.2.1 Méthode

Le critère proposé est basé sur le rapport entre le nombre de pixels en mouvement et le nombre de pixels de l'avant-plan. Puisque nous disposons déjà de l'avant-plan, calculé pour la détection de personne, il suffit alors de calculer une détection de mouvement.

Soit $\mathcal{A}_t(s)$, le résultat de la détection de mouvement défini par :

$$\mathcal{A}_t(s) = \begin{cases} 1 & \text{si } d_2(\boldsymbol{I}_{s,t}, \boldsymbol{I}_{s,t-\eta}) > \tau_{\mathcal{A}} \\ 0 & \text{sinon,} \end{cases} \tag{4.1}$$

où $d_2(\boldsymbol{I}_{s,t}, \boldsymbol{I}_{s,t-\eta})$ représente la distance euclidienne entre le pixel s à l'instant t et ce même pixel à l'instant $t-\eta$, $\tau_{\mathcal{A}}$ est un seuil. $\mathcal{A}_t(s) = 1$ signifie que le pixel s est sur une zone en mouvement.

Soit $\mathcal{X}_t(s)$, le masque de l'avant-plan défini par :

$$\mathcal{X}_t(s) = \begin{cases} 1 & \text{si } d_M(\boldsymbol{I}_{s,t}, B_{s,t}) > \tau_{\mathcal{X}} \\ 0 & \text{sinon,} \end{cases} \tag{4.2}$$

où $d_M(\boldsymbol{I}_{s,t}, B_{s,t})$ représente la distance de Mahalanobis entre le pixel s à l'instant t et le modèle de l'arrière-plan à l'instant t, composé de la moyenne et de la matrice de covariance, $\tau_{\mathcal{X}}$ est un seuil. $\mathcal{X}_t(s) = 1$ signifie que le pixel s représente l'avant-plan.

La figure 4.1 présente un exemple d'images du mouvement et d'images de l'avant-plan.

(a) (b) (c)

Figure 4.1: Exemples d'images utilisées pour la mesure de l'activité (a) image d'entrée (b) image du mouvement (c) image de l'avant-plan.

La mesure de l'activité, à l'instant t, est réalisée en calculant le rapport entre le nombre de pixels en mouvement et le nombre de pixels de l'avant-plan :

$$\kappa_t = \frac{\sum_{s \in S} \mathcal{A}_t(s)}{\sum_{s \in S} \mathcal{X}_t(s)} \tag{4.3}$$

où S correspond au nombre de pixels de l'image. Un filtre temporel peut être utilisé pour lisser dans le temps la mesure de l'activité κ_t :

$$\kappa_t = \alpha . \frac{\sum_{s \in S} \mathcal{A}_t(s)}{\sum_{s \in S} \mathcal{X}_t(s)} + (1 - \alpha) . \kappa_{t-1} \tag{4.4}$$

où α est un coefficient fixé empiriquement.

4.2.2 Validation

Protocole : Afin de valider cette mesure, nous avons utilisé ce critère sur deux groupes de vidéos (extraites de l'ensemble des vidéos décrites dans la partie 3.6.1). Le *groupe 1* rassemble les vidéos où l'activité physique est faible (scène de bureau, réunion *etc.*), nous avons utilisé 21 vidéos dans ce cas. Le *groupe 2* rassemble toutes les vidéos dans lesquelles il y a une activité plus importante (des zones de passage dans un couloir ou des personnes agitées), nous utilisons ici 6 vidéos.

Résultats expérimentaux : Nous présentons dans le tableau 4.1 la moyenne de la mesure de l'activité $\overline{\kappa}$ sur ces deux groupes de vidéos.

	groupe 1	*groupe 2*
$\overline{\kappa}$	0.08	0.23

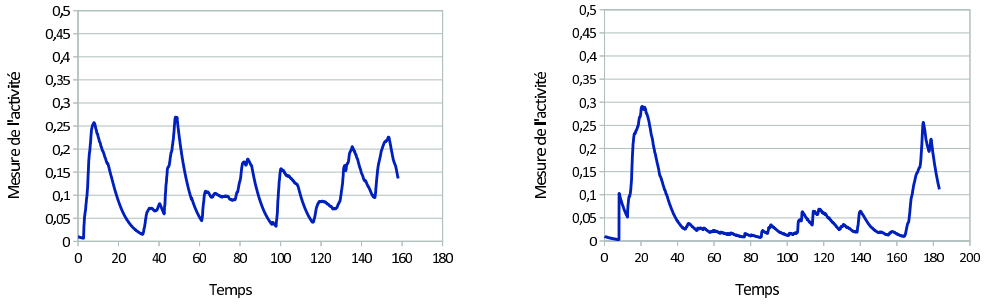
Table 4.1: Moyenne de la mesure d'activité sur les deux groupes de vidéos.

Le tableau 4.2 présente les résultats obtenus lorsque la mesure de l'activité est seuillée pour classifier le contenu de la scène en deux classes "active" ou "calme". Nous pouvons alors observer que malgré la simplicité de cette mesure, il est possible d'obtenir, avec un taux de confiance assez élevé, une mesure de l'activité globale des personnes présentes. Nous obtenons 91% de réponse calme lorsque le contenu de la scène est effectivement calme et 100% de bonne réponse concernant une scène où il y a de l'agitation. Bien entendu, ces résultats sont à relativiser par rapport à la subjectivité de la notion d'activité.

	calme	active
calme	0.91	0.09
active	0.00	1.00

Table 4.2: Matrice de confusion sur la classification du contenu d'une scène entre "calme" et "active".

Nous présentons dans la figure 4.2 l'évolution dans le temps de la mesure de l'activité dans deux zones présentant des activités distinctes. Dans le premier cas de figure illustrant une zone de passage, chaque pic de la mesure κ correspond au passage d'une personne. Dans le second cas de figure correspondant à une pièce où se déroule une réunion de travail, on peut observer deux pics de la valeur de κ correspondant à l'arrivée et au départ des personnes.



(a) (b)

(c) (d)

Figure 4.2: Évolution dans le temps de la mesure de l'activité physique dans le cas d'une zone de passage fréquent (a) et (c) et dans le cas d'une zone calme de réunion de travail (b) et (d).

4.2.3 Discussion

Le critère de l'équation 4.3 permet donc simplement de quantifier l'activité des personnes présentes dans une pièce. Même si les informations obtenus sont simples, elles sont utiles pour la régulation automatiquement du chauffage. Il y a principalement deux avantages à cette mesure de l'activité :

1. cette mesure est très simple à calculer, d'autant plus que le masque de l'avant-plan a déjà été calculé lors de la détection de personnes,
2. cette mesure ne dépend ni du point de vue d'observation de la caméra ni de la distance entre les personnes présentes et la caméra. En effet, le terme $\sum_{s \in S} \mathcal{X}_t(s)$ permet de normaliser l'activité mesurée avec $\sum_{s \in S} \mathcal{A}_t(s)$, en fonction du nombre de personnes et de leur distance par rapport à la caméra.

4.3 Détection de personnes par stéréovision multicapteurs

Dans cette partie, nous proposons un système de stéréovision multicapteurs permettant d'augmenter les performances du système de détection de personnes et de les localiser dans l'espace. Cette information peut être utile dans le cadre d'une application de maintien à domicile ou de domotique.

Les résultats présentés dans cette partie ont été obtenus avec un système de stéréovision basé sur une caméra dans le spectre visible et une caméra dans le spectre infrarouge. Il est évidemment possible d'utiliser les techniques présentées ici avec un système de stéréovision utilisant deux caméras dans le même spectre mais nous avons remarqué dans la partie 2.2.2.2 que les fausses détections dans ces deux spectres sont dues à des causes différentes. Dans le spectre visible, il y a des problèmes avec certains objets ayant des formes particulières, avec les ombres projetées *etc.* Dans le spectre infrarouge, la majorité des fausses détections est due aux réflexions sur les sols plastiques. La figure 4.3 montre deux exemples de réflexions sur des sols plastiques dans une image infrarouge. Le système de stéréovision multicapteurs (caméra infrarouge et visible) permet, en utilisant les propriétés géométriques d'un système de stéréovision, de combiner les résultats de détection dans les deux spectres et donc de diminuer le nombre de fausses détections inhérentes à l'utilisation d'un seul type de caméra. Une caméra infrarouge et une caméra visible sont donc montées en un système de stéréovision, les personnes sont détectées dans les deux spectres indépendamment puis les résultats sont fusionnés avec la géométrie épipolaire du système de stéréovision. Les personnes détectées sont ensuite localisées dans l'espace.

4.3.1 Présentation du système

Connaissant la géométrie épipolaire d'un système de stéréovision, il est possible de faire correspondre chaque point de l'image d'une caméra avec sa droite épipolaire dans

Figure 4.3: Exemples de réflexions sur des sols plastiques dans des images acquises par une caméra thermique.

l'image de l'autre caméra. Cette correspondance peut être réalisée avec la matrice fondamentale [15] ou bien en utilisant les paramètres intrinsèques et extrinsèques du système de stéréovision. Nous utiliserons la deuxième approche car nous disposons des paramètres intrinsèques et extrinsèques nécessaires à la localisation dans l'espace.

Soit R_{C1}, R_{C2} et R_A les repères de la caméra 1, la caméra 2 et le repère absolu. Nous notons R_C le repère de la caméra 1 ou de la caméra 2. Les paramètres extrinsèques permettent d'exprimer les coordonnées homogènes d'un point de l'espace dans le repère de la caméra par :

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}_{R_c} = M_{CA} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}_{R_A} . \tag{4.5}$$

Avec les paramètres intrinsèques, on exprime un point du repère caméra $[X_c, Y_c, Z_c, 1]^T$ en un point sur l'image $[u, v, 1]^T$:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_c} \begin{pmatrix} F_x & \gamma & u_0 & 0 \\ 0 & F_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}_{R_c} , \tag{4.6}$$

où $f = (F_x, F_y)$ est la distance focale de la caméra, $[u_0, v_0]$ sont les coordonnées du centre optique et γ est un paramètre de distorsion. En utilisant les équations 4.5 et 4.6, il est possible de projeter un point de l'espace $3D$, $A = [X, Y, Z]^T_{R_A}$ en coordonnées $[u, v]$ dans l'image de l'une des deux caméras.

Les paramètres intrinsèques et extrinsèques ont été déterminés avec une mire de dimensions connues et les outils proposés par Bouguet [14]. Les motifs de la mire devant être visibles dans les deux spectres, nous avons utilisé une mire atypique comportant des zones chaudes et froides. Pour obtenir des zones de températures différentes, nous avons peint certaines zones et présenté la mire à une source de chaleur, les zones peintes chauffant plus vite que les zones brutes, nous sommes

Figure 4.4: Mire utilisée pour la calibration vue dans les deux spectres visible et infrarouge.

capables de quadriller la mire avec des zones de différentes couleurs et températures. La figure 4.4 présente une vue de la mire dans les deux spectres.

Une fois les paramètres de la calibration obtenus, il est possible d'obtenir les matrices de projection M_{C1C2} et M_{C2C1} qui permettent de passer du repère de la caméra 1 au repère de la caméra 2 et inversement. Si nous faisons l'hypothèse que $\gamma = 0$, d'après l'équation 4.6, on peut exprimer $[X_c, Y_c, Z_c]^T$ en fonction de $[u, v]$:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} (u - u_0)Z_c/F_x \\ (v - v_0)Z_c/F_y \\ Z_c \end{bmatrix}, \tag{4.7}$$

où Z_c est la distance entre la caméra C et le point $3D$. Ainsi, un point $p = [u, v]$ d'une image peut être projeté en une droite d_p dans le repère caméra R_c avec l'expression 4.7 où Z_c est inconnu. Ensuite, d_p est projeté dans le repère de l'autre caméra avec les matrices de transformations M_{C1C2} ou M_{C2C1} puis finalement dans la seconde image avec l'équation 4.6. Cette correspondance est utilisée pour fusionner les résultats de détection obtenus dans chaque spectre.

4.3.2 Fusion des résultats de détection dans chaque spectre

En utilisant la méthode décrite ci-dessus, il est possible d'exprimer un point de l'image d'une caméra en une droite dans l'image de la seconde caméra avec les paramètres du système de stéréovision. Cette correspondance est utilisée pour fusionner les détections réalisées indépendamment dans chaque spectre. Le principe de la fusion des résultats de détection est expliqué à travers l'exemple illustré dans la figure 4.5. Soit M détections dans l'image du spectre visible et N détections dans le spectre infrarouge. Soit A_i^{vis} et B_i^{vis} les points supérieurs gauches et inférieurs droits de la i^{eme} détection dans le spectre visible et dA_i^{vis} et dB_i^{vis}, leurs droites épipolaires respectives dans l'image du spectre infrarouge. Dans cette méthode, une détection $i \in [1, M]$ dans l'image du spectre visible n'est conservée que si :

$$\exists j \in [1, N] \text{ tel que } dist(A_j^{IR}, dA_i^{vis}) < \tau \text{ et } dist(B_j^{IR}, dB_i^{vis}) < \tau. \tag{4.8}$$

Où $dist(A, dA)$ représente la distance entre le point A et sa projection orthogonale sur la droite dA. Si cette condition n'est pas réalisée, la détection est supprimée. Dans la figure 4.5, il y a deux détections dans le spectre visible et seulement une dans le spectre infrarouge. La détection 1 dans l'image visible sera conservée. Bien entendu, ce processus est utilisé dans les deux sens. C'est-à-dire que les détections du spectre infrarouge seront également confirmées par les détections dans le spectre visible.

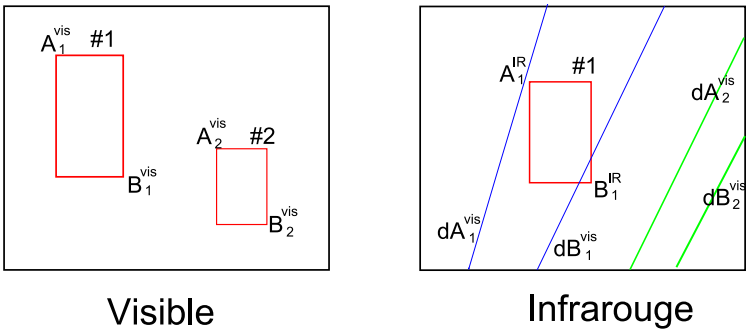


Figure 4.5: Exemple de fusion des détections en utilisant la géométrie épipolaire. La projection des points caractéristiques d'une détection dans l'image 1 doit se faire à une distance inférieure à τ d'une détection dans l'autre image.

Il est possible d'obtenir un nombre de détections différent dans les images de chaque spectre résultant de quelques cas particuliers. Un exemple est présenté dans la figure 4.6. Dans ce cas, il est alors possible de répéter le processus de fusion pour lever les cas ambigus. Par exemple, dans l'illustration de la figure 4.6, la deuxième détection dans l'image visible ne se reprojetterait pas à proximité d'une détection dans l'infrarouge si le processus de fusion était réitéré.

4.3.3 Validation

Protocole : Nous présentons dans un premier temps les résultats de l'évaluation de la fusion des détections. Cette méthode a été évaluée sur une base composée de 640 images dans chaque spectre. Nous avons utilisé la méthode de Viola et Jones [148] puis chaque résultat de détection a été confirmé ou infirmer par la méthode présentée ci-dessus. Les résultats sont présentés en utilisant les courbes Précision/Rappel.

Dans un second temps, nous présentons un exemple de localisation dans l'espace des personnes détectées.

Résultats expérimentaux : Les courbes précision/rappel de l'évaluation de la fusion des détections sont présentées dans la figure 4.7. Les courbes *IR* et *Visible* sont les résultats obtenus à partir des images du spectre infrarouge et visible sans utiliser la fusion. Les courbes *Visible amélioré* et *IR amélioré* présentent les résultats de détection après fusion. On peut observer que les performances ont été nettement améliorées.

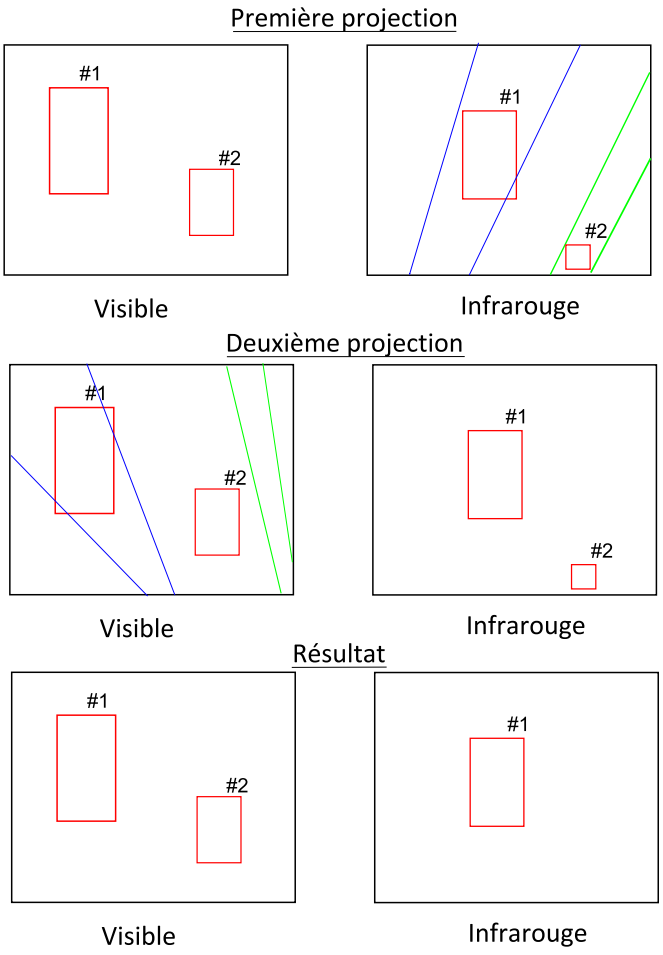


Figure 4.6: Exemple d'un cas particulier de fusion des détections, le nombre de détections dans chaque image est différent après optimisation. La réitération du processus de fusion permet de lever cette ambiguïté.

Dans la figure 4.8, nous présentons un exemple de résultat dans lequel les fausses détections dans les deux spectres ont été supprimées. Sur la première ligne, on voit les détections obtenues sans utiliser la méthode de fusion. Des fausses détections sont présentes dans chaque image des deux spectres. Sur la deuxième ligne, on voit les résultats de détection obtenus avec la fusion où les fausses détections ont été supprimées. On peut également remarquer que la procédure de fusion est très rapide puisqu'elle ne requiert que deux projections par détection.

À partir des paramètres du système de stéréovision calibré, nous pouvons également localiser dans l'espace les personnes détectées. Dans la figure 4.9, on peut voir un exemple de localisation. Le rectangle bleu correspond à la zone dans laquelle se trouvait la personne (cf. figure 4.10) et les croix rouges correspondent aux posi-

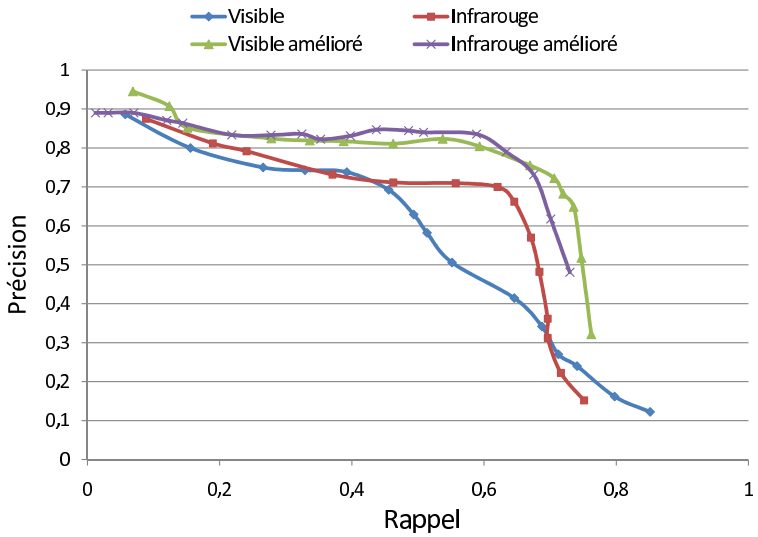


Figure 4.7: Courbes Précision/Rappel obtenues sur les images infrarouge et visible avec et sans utiliser la fusion des détections.

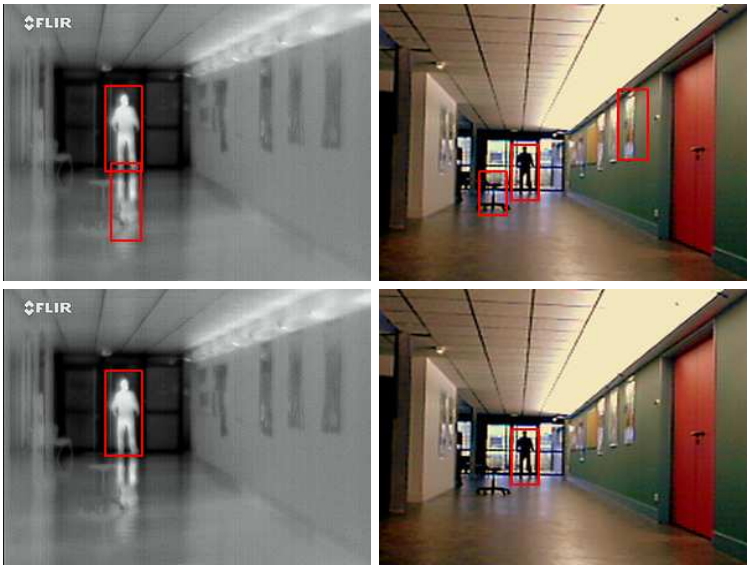


Figure 4.8: Première ligne : exemples de détections sans utiliser la méthode de fusion. Deuxième ligne : Exemples de détections avec la stéréovision.

tions estimées. Pour localiser la personne détectée, nous avons simplement récupéré les coordonnées du centroïde de la détection dans les deux spectres puis calculé ses coordonnées $3D$ par stéréotriangulation.

Cette méthode ne permet pas de localiser avec précision la position des personnes détectées. En effet, l'incertitude associée à la position des détections dans chaque image est assez importante (jusqu'à 10 pixels) et l'erreur de localisation se propage lors de la stéréotriangulation. Cependant, les résultats de la localisation présentés ici permettent de connaître les zones dans lesquelles se trouvent les personnes détectées.

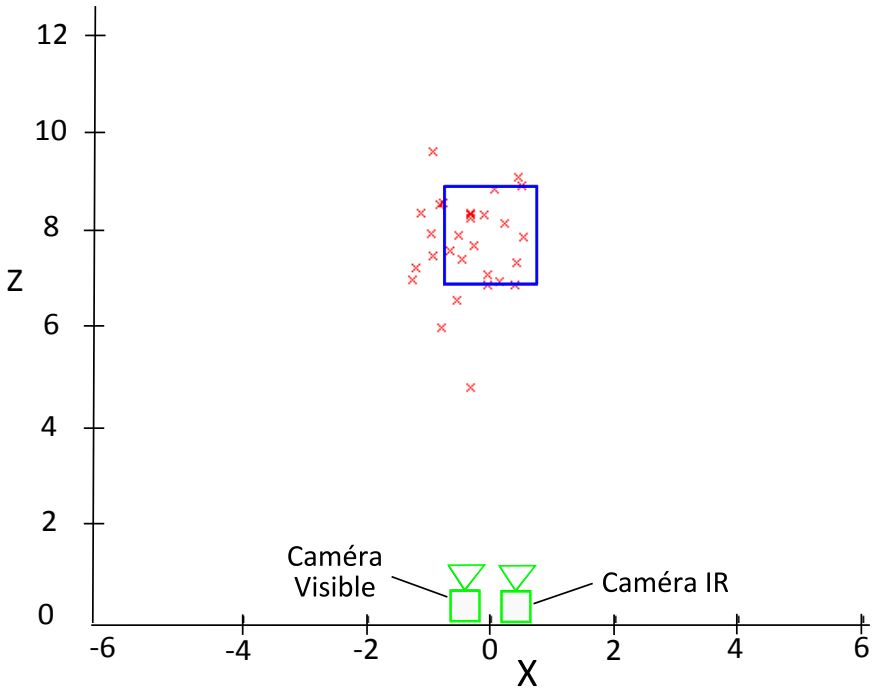


Figure 4.9: Illustration de la localisation d'une personne détectée. Le rectangle bleu correspond à la zone dans laquelle se trouve la personne et les croix rouges correspondent aux positions estimées. Les axes X et Z sont en mètres.

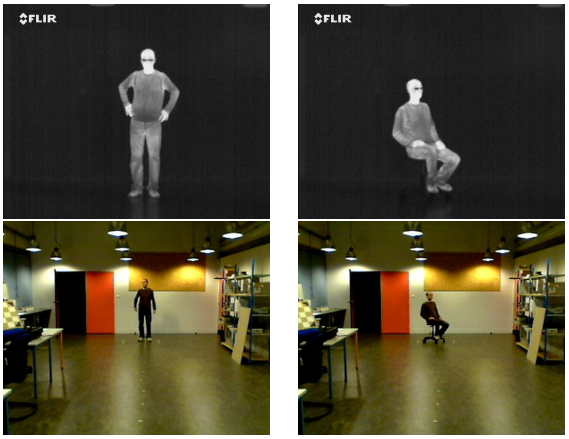


Figure 4.10: Exemple d'images utilisées pour la localisation.

Nous pouvons noter que cette information est suffisante pour des applications de maintien à domicile où une localisation plus précise ne serait pas utile.

4.3.4 Discussion

Nous avons présenté ici un système de stéréovision multicapteurs basé sur une caméra infrarouge et une caméra visible permettant d'augmenter les performances

de la détection de personnes et également de localiser les personnes détectées dans l'espace.

On peut tout d'abord noter que les performances de détection ont été nettement améliorées en utilisant la géométrie épipolaire du système de stéréovision. Ensuite, la localisation dans l'espace nous fournit une information suffisante pour une application de maintien à domicile. En effet, il semble suffisant de pouvoir déterminer si la personne détectée se trouve sur son lit ou dans une autre zone d'intérêt, une localisation plus précise est inutile.

4.4 Détection d'événements anormaux

Afin d'analyser l'activité des objets mobiles d'un environnement, beaucoup de méthodes se basent sur une architecture en cascade où les objets sont tout d'abord détectés, suivis puis classifiés [76, 77, 84, 86]. Ensuite, le chemin parcouru par les différents objets détectés est analysé puis confronté à un modèle pour détecter les situations suspectes.

Nous proposons ici de modéliser l'activité et de détecter les événements anormaux en analysant les pixels de l'avant-plan obtenus avec la soustraction de l'arrière-plan. Cette méthode se base sur l'observation que les pixels de l'avant-plan sont spatialement et temporellement dépendants. En effet, un objet en mouvement détecté au pixel $s1$ à l'instant $t1$, sera détecté au pixel $s2$ à l'instant $t2$. À partir de cette observation, nous proposons de construire un modèle contenant les statistiques de co-occurrence spatio-temporelles des pixels de l'avant-plan. Ces statistiques seront ensuite utilisées pour la détection d'événements anormaux.

Il existe dans la littérature quelques méthodes dans lesquelles les événements anormaux sont détectés à partir d'informations bas-niveau (*e.g.* [7, 83]). Cependant, dans ces méthodes, chaque pixel est traité indépendamment de son voisinage. Ces méthodes se basent sur l'hypothèse qu'une séquence temporelle des valeurs de l'avant-plan, pour un pixel donné, permet d'avoir suffisamment d'information pour modéliser l'activité habituelle. Et donc, le modèle ainsi obtenu permet de détecter les situations anormales ou peu fréquentes. Par exemple, il est possible d'apprendre la séquence de pixels de l'avant-plan correspondant à l'activité d'un pixel sur une route ou dans un couloir. Dans ce dernier cas, une période d'activité correspond au passage d'une personne. Une séquence temporelle des pixels de l'avant-plan générée par un colis abandonné ou une personne étendue à terre suite à un malaise différera alors de la séquence apprise. Les corrélations spatiales entre les pixels de l'avant-plan ne sont pas considérées dans ce cas. Il y a cependant des cas où la distribution temporelle n'est pas suffisante et où la distribution spatiale des pixels de l'avant-plan est nécessaire pour détecter certaines activités. Nous présentons ici un modèle incluant les statistiques des distributions spatiales et temporelles des pixels de l'avant-plan.

La méthode décrite ci-dessous est générique et peut être utilisée pour la détection de plusieurs types d'activités anormales. Nous présenterons quelques résultats expérimentaux dans le cadre de la détection de colis abandonnés (pouvant s'apparenter

à la détection de chutes puisque le caractère statique d'une personne étendue à terre sera alors anormal) mais également de changements anormaux de trajectoire.

4.4.1 Modèle et apprentissage du comportement normal

Nous nous proposons ici de définir un modèle encapsulant les fréquences de co-occurrence des pixels de l'avant-plan dans un même volume spatio-temporel. Nous commençons tout d'abord par définir les grandeurs utilisées dans le reste de cette partie.

Nous définissons le voisinage spatio-temporel $\mathcal{M}_u$ d'un pixel $u = (s1, t)$, $s1$ étant la position spatiale et t la position temporelle. Ce voisinage est un sous-volume de la séquence vidéo S, à savoir $\mathcal{M}_u \subset \mathcal{S}$ centré sur $u \in \mathcal{S}$ et de taille $Q \times R \times T$, $Q < Q_0$, $R < R_0$ et $T < T_0$ où Q_0 et R_0 représentent la taille de chaque image de la séquence vidéo et T_0 le nombre d'images de la séquence d'apprentissage. Ces paramètres sont illustrés dans la figure 4.11.

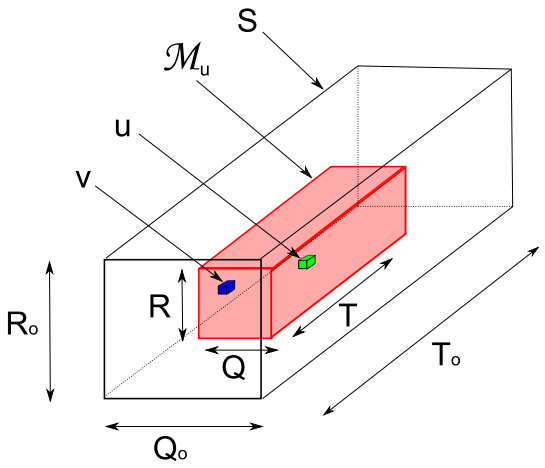


Figure 4.11: Vidéo S avec le voisinage spatio-temporelle $\mathcal{M}_u$ de u.

Soit $v = (s2, t + \tau) \in \mathcal{M}_u$, v est dans le voisinage de u si et seulement si $s2$ est dans le voisinage spatial de $s1$ et $\tau \in [-T/2, T/2]$. Nous considérons par la suite que les deux pixels u et v co-occurrent lorsque leurs pixels de l'avant-plan sont tous les deux actifs, c'est-à-dire lorsque $\mathcal{X}_u = 1$ et $\mathcal{X}_v = 1$.

La matrice de co-occurrence, notée $\mathcal{C}$, est construite par :

$$\mathcal{C}_{uv} = \frac{\beta_{uv}}{T_0 - T} \sum_{t=T/2}^{T_0 - T/2} \delta(\mathcal{X}_u, \mathcal{X}_v) \tag{4.9}$$

où T_0 est le nombre total d'images dans la séquence d'apprentissage, β_{uv} est une constante dépendant de la distance entre u et v et $\delta(\mathcal{X}_u, \mathcal{X}_v) = \mathcal{X}_u.\mathcal{X}_v$.

Le nombre de paires de pixels à considérer étant très important, de manière pratique, nous ne considérons que les paires de pixels avec un ou plusieurs pixel(s)

clé, réparti(s) de manière clairsemée sur l'image. Autrement dit, pour un ou plusieurs pixels-clé, nous construisons la matrice de co-occurrence avec leur voisinage spatio-temporel.

En choisissant avec attention la taille $Q \times R \times T$ de la matrice de co-occurrence, on encapsule implicitement les informations concernant la taille, la direction et la vitesse des objets passant devant ce pixel-clé. Nous obtenons finalement une distribution de probabilité en 3D non-paramétrique. Chaque élément de cette matrice correspond à une probabilité de co-occurrence d'un pixel avec le pixel clé.

4.4.2 Détection d'événements anormaux

Nous considérons qu'un événement est anormal s'il a une faible probabilité d'occurrence. Soit $\mathcal{S}$ une séquence vidéo de taille $Q_0 \times R_0 \times T_{test}$, $\mathcal{M}_u$ un voisinage spatio-temporel du pixel $u = (s, t)$ et $\mathcal{O}_u$ les valeurs des pixels de l'avant-plan du voisinage de u, appelé par la suite "trace" $\mathcal{O}_u$.

Nous souhaitons désormais, pour chaque instant t, déterminer la normalité de la trace $\mathcal{O}_u$, appelée $p(\mathcal{O}_u)$. Le test est réalisé comme suit :

$$p(\mathcal{O}_u) = \frac{\sum_{v \in \mathcal{M}_u} \mathcal{C}_{uv} \delta(\mathcal{X}_u, \mathcal{X}_v)}{\sum_{v \in \mathcal{M}_u} \mathcal{X}_v} \underset{anormal}{\overset{normal}{\gtrless}} \tau. \tag{4.10}$$

La figure 4.12 illustre une scène de vidéo-surveillance dans laquelle les piétons se déplacent de gauche à droite et de droite à gauche. La matrice de co-occurrence, représentée dans la figure 4.12(b) modélise effectivement ces deux activités distinctes car on peut observer que le modèle présente une forme de croix. Dans la figure 4.12(c), on peut voir la trace laissée par une personne abandonnant un colis, la trace laissée par cet événement est sensiblement différente du modèle et donc cette activité sera détectée comme anormale.

Nous présentons ensuite dans la figure 4.13 un autre exemple extrait d'une vidéo de ETISEO [113]. Dans cet exemple, les personnes se déplacent en partant du coin inférieur droit de l'image pour aller vers le coin supérieur-gauche et inversement. On constate bien que la matrice de co-occurrence, figure 4.13(b), est bimodale également. Ensuite, nous présentons trois exemples de traces laissées par trois activités différentes. Dans les figures 4.13(c) et 4.13(d) on peut observer les traces de deux personnes se déplaçant dans le couloir en suivant des directions opposées. Dans la figure 4.13(e), on peut voir la trace laissée par une personne abandonnant un colis (les axes x et t ont été inversés pour mieux observer cette trace).

Finalement, nous présentons dans la figure 4.14 un exemple extrait d'une vidéo de PETS [141]. Dans cet exemple, on peut observer que la matrice de co-occurrence 4.14(b) est également bimodale mais qu'un mode est beaucoup plus important que l'autre. Ensuite, nous présentons deux exemples de trace laissées par une activité normale 4.14(c) et une activité anormale 4.14(d).

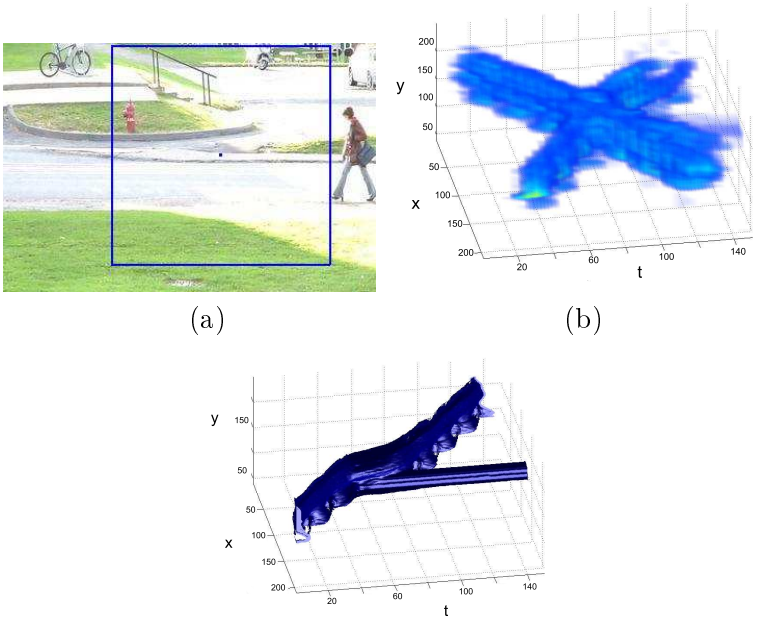


Figure 4.12: (a) Image extraite de la vidéo considérée, le rectangle bleu représente une coupe de la matrice de co-occurrence, (b) matrice de co-occurrence modélisant le déplacement des piétons de gauche à droite et de droite à gauche (c) trace laissée par une personne abandonnant un colis.

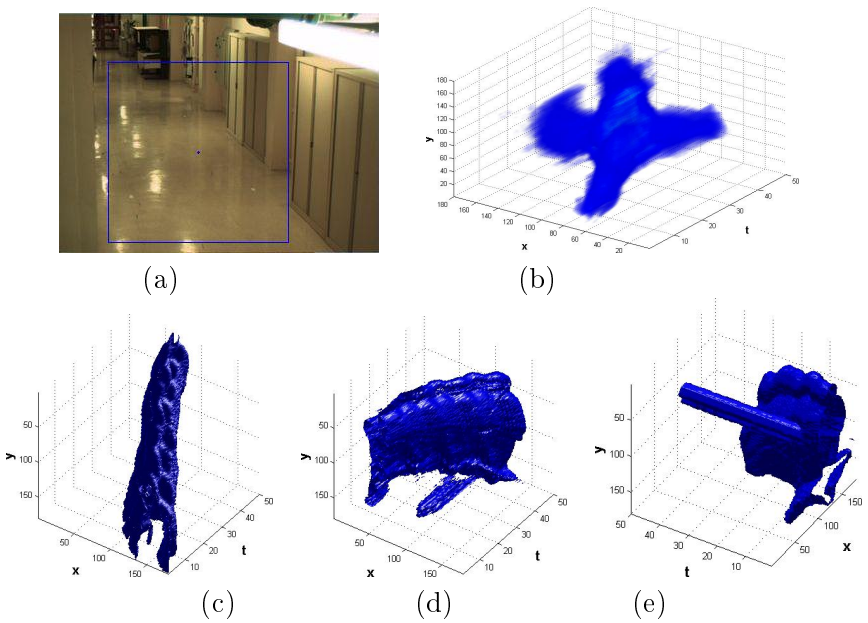


Figure 4.13: (a) Image extraite d'une séquence de vidéo-surveillance de ETISEO [113] (b) matrice de co-occurrence bimodale (c) et (d) traces de deux événements normaux (e) trace laissée par une personne abandonnant un colis.

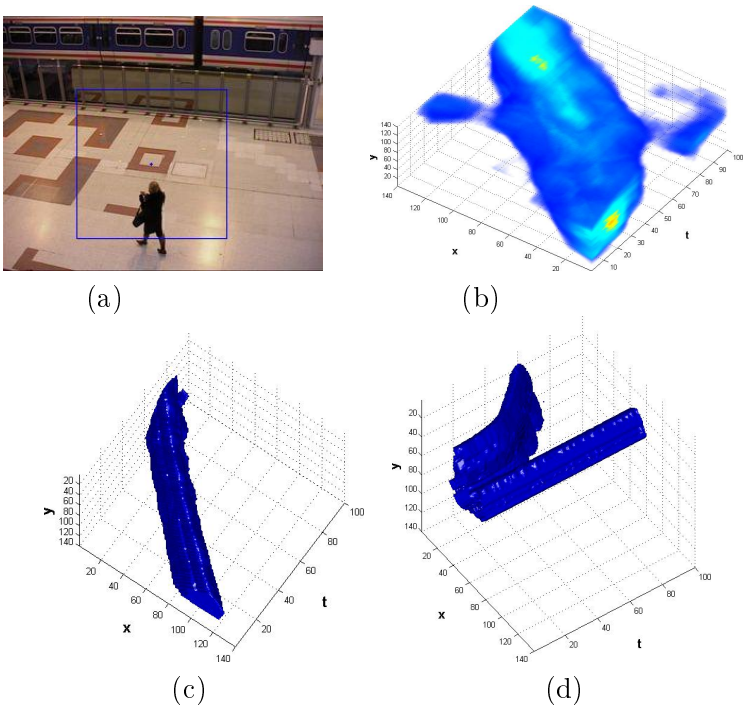


Figure 4.14: (a) Image extraite d'une vidéo de vidéo-surveillance PETS [141] (b) matrice de co-occurrence bimodale avec un axe plus important que l'autre (c) trace laissée par le déplacement d'une personne (d) trace laissée par une personne abandonnant un colis.

4.4.3 Gestion de plusieurs objets mobiles

Cependant, en considérant un voisinage $\mathcal{M}_u$ de grande taille, il est possible que $\mathcal{O}_u$ contienne les traces de plusieurs objets alors qu'il peut être plus informatif d'analyser l'activité de chaque objet indépendamment. Afin de ne sélectionner dans $\mathcal{O}_u$ que les pixels de l'avant-plan correspondant à l'objet d'intérêt (celui passant sur le pixel-clé), nous ne considérons que les pixels de l'avant-plan qui, non seulement co-occurrent avec u, mais sont également reliés entre eux (composantes connectées en 3D).

Si un objet passe suffisamment proche d'un événement détecté comme anormal, leurs traces ne formeront alors qu'une seule trace dans $\mathcal{O}_u$ et par l'équation 4.10, nous évaluerons la normalité de l'union de ces deux traces. L'activité de l'objet en mouvement sera détectée comme étant anormale par le simple fait de passer à proximité de la trace d'un objet qui a une activité anormale ou alors l'activité anormale sera détectée normale à cause de la trace de l'activité normale. Ce cas peut être illustré par l'exemple d'un colis abandonné. Le colis abandonné sera détecté comme anormal puisque sa trace différera grandement du modèle appris. Chaque personne passant devant ce colis abandonné verra sa trace rejoindre la trace du colis abandonné et sera détecté comme anormal ou alors le colis ne sera plus détecté comme anormal. Or il est plus informatif d'analyser séparément les traces de ces deux événements. Nous cherchons donc à analyser l'activité de la nouvelle observation sans considérer la trace de l'objet ayant déjà été détecté comme anormal. Une manière simple pour savoir

si l'observation courante est composée de l'union de l'observation détectée anormale à l'instant $t-1$ et un nouvel objet dont la trace lui est connectée est de calculer le rapport entre l'intersection et l'union de l'observation à l'instant $t-1$ détectée comme anormale et l'observation à l'instant t. Donc si la trace $\mathcal{O}_{s,t-1}$ est détectée comme anormale, nous calculons le rapport suivant :

$$\varepsilon = \sum_{v \in \mathcal{M}_u} \frac{(O_{s,t}(v) \ \wedge \ \mathcal{O}_{s,t-1}(v))}{(O_{s,t}(v) \ \vee \ \mathcal{O}_{s,t-1}(v))} \tag{4.11}$$

Nous rappelons que $u = (s,t)$. La valeur ε, permet de tester si la trace spatio-temporelle observée est composée de l'union de la trace de l'événement anormal et d'une nouvelle observation (si $\varepsilon < \tau$) ou seulement une mise à jour de $\mathcal{O}_{s,t-1}$ (si $\varepsilon > \tau$). Dans le premier cas, il est possible d'effectuer le test de normalité (cf. équation 4.10) sur $\mathcal{O}'_{s,t} = \mathcal{O}_{s,t} - \mathcal{O}_{s,t-1}$, où $\mathcal{O}'_{s,t}$ représente la trace spatio-temporelle de cette nouvelle observation.

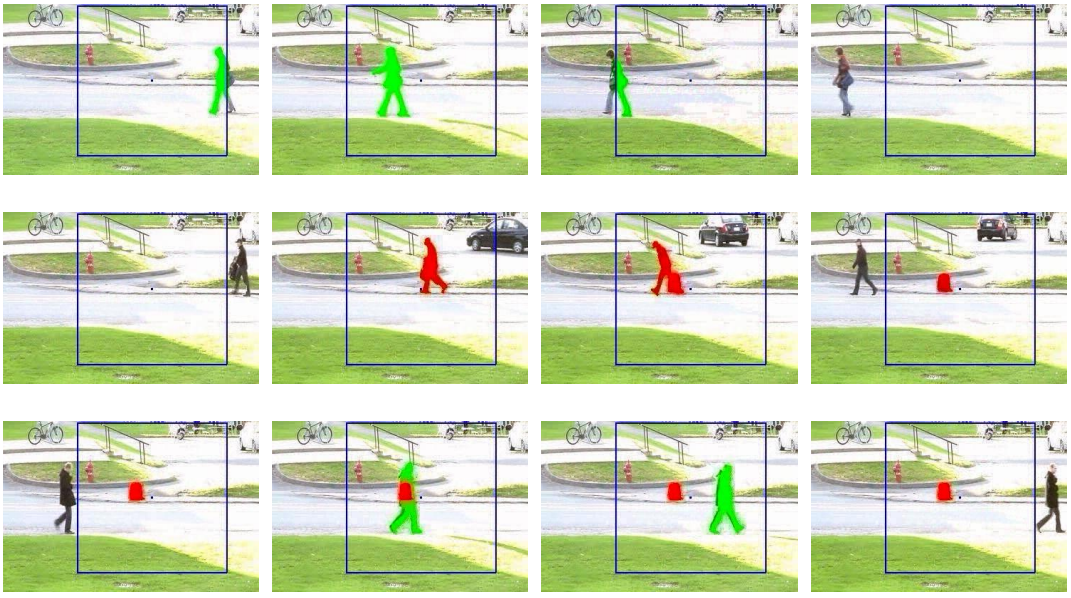
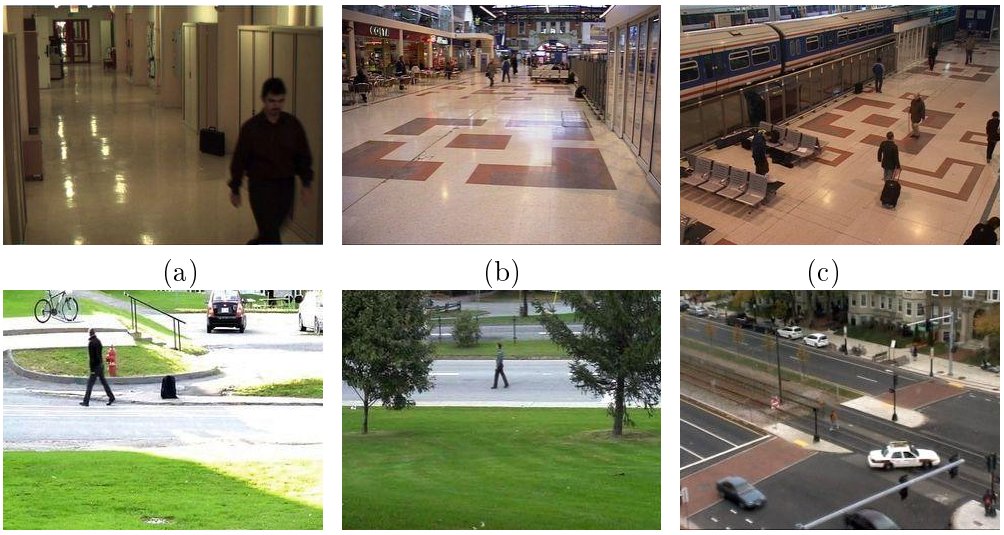


Figure 4.15: Exemple de gestion de multiples objets. La première ligne présente une personne se déplaçant normalement, la deuxième une personne abandonnant un objet et la troisième ligne une personne se déplaçant normalement devant le colis abandonné. Les événements normaux sont marqués en vert et les événements anormaux en rouge, on peut observer que dans l'exemple de la troisième ligne, la personne et le colis ne forme qu'une seule trace, par la méthode décrite ici, nous sommes en mesure de distinguer ces deux événements.

4.4.4 Validation

Protocole Afin d'évaluer la méthode présentée ci-dessus, nous utilisons une base de 6 vidéos contenant chacune au moins un événement anormal. Comme cette mé-

thode est générique et permet d'apprendre une grande variété d'activités, nous avons utilisé plusieurs types de vidéos dans la base d'évaluation. Nous avons créé certaines de ces vidéos, les autres sont extraites de PETS2006 [141] et ETISEO [113]. Nous avons tout d'abord utilisé des séquences de vidéo-surveillance dans lesquelles les événements anormaux sont des colis abandonnés. Ensuite, nous avons utilisé des vidéos dans lesquelles les personnes (ou d'autres objets mobiles) suivent des trajectoires spécifiques et les événements anormaux sont alors des personnes empruntant une trajectoire différente. La figure 4.16 présente quelques illustrations extraites de la base d'évaluation.



(a) (b) (c)

(d) (e) (f)

Figure 4.16: Illustration des vidéos utilisées dans la base d'évaluation. Dans les exemples (a) (b) (c) et (d), les événements anormaux sont des colis abandonnés. Dans les exemples (e) et (f) les événements anormaux sont des objets mobiles empruntant une trajectoire anormale (personne marchant à contre sens ou voiture effectuant un demi-tour).

La méthode présentée ici a été comparée avec la méthode de Chen *et al.* [25]. La méthode développée par Chen *et al.* [25] repose sur une architecture en cascade où les objets sont détectés avec une soustraction de l'arrière-plan, suivis et leurs trajectoires sont analysées pour déterminer leur normalité. Une trajectoire est représentée par un ensemble de vecteurs descripteurs (position, vitesse etc.) accumulés dans le temps. Si un objet détecté possède des descripteurs rarement ou jamais observés auparavant, celui-ci est considéré comme anormal.

Résultats expérimentaux Le tableau 4.3 présente les résultats de la méthode de Chen *et al.* [25] et ceux de notre méthode sur la base de vidéos décrite ci-dessus. On peut voir que tous les événements anormaux ont été détectés avec notre méthode. Cependant, la matrice de co-occurrence modélisant implicitement la taille des objets qui passent usuellemment devant le pixel clé, un objet significativement plus gros que les objets de la séquence d'apprentissage sera considéré comme anormal. Les erreurs

observées avec notre méthode sont dues à cette raison. Les erreurs observées avec la méthode de Chen *et al.* [25] sont dues à des erreurs de suivi.

	Normal	Anormal
Normal	86.9	13.1
Anormal	7.1	92.9

(a)

	Normal	Anormal
Normal	93.1	6.9
Anormal	0	100

(b)

Table 4.3: Matrices de confusion obtenues à partir de 6 vidéos. (a) Méthode de Chen *et al.* [25] basée sur le suivi d'objet (b) méthode proposée.

Le temps d'exécution de la méthode proposée est très court. En effet, avec une implémentation C++, nous sommes capables d'atteindre une exécution en temps réel (environ 19 images par seconde pour une matrice de co-occurrence de $210 \times 210 \times 150$). Bien évidemment, plus on utilise de pixels-clé et plus la taille de la matrice de co-occurrence est grande, plus le temps de détection augmente.

4.4.5 Discussion

Nous avons présenté ci-dessus une méthode pour détecter des événements anormaux dans une séquence d'images. Celle-ci se base sur une analyse des statistiques de co-occurrence des pixels de l'avant-plan dans un volume spatio-temporel. Cette méthode est générique et permet donc de modéliser une grande variété d'activité. De plus, cette méthode ne se base pas sur l'analyse du chemin obtenu après un suivi des objets mobiles, nous supprimons ainsi le risque de propagation des erreurs.

Malgré les bons résultats présentés ci-dessus, il faut tout de même remarquer que cette méthode ne peut être utilisée que dans le cadre d'environnement présentant une activité relativement structurée. En effet, dans le cas d'environnement totalement non-structuré, la matrice de co-occurrence présente des valeurs homogènes et ne permet pas de détecter des événements anormaux. Ensuite, un deuxième inconvénient de cette méthode est l'espace mémoire nécessaire pour sauvegarder la matrice de co-occurrence. Il est toutefois possible d'utiliser les méthodes classiques de réduction de dimension de type analyse en composantes principales ou analyse discriminante linéaire.

4.5 Conclusion

Nous avons présenté dans ce chapitre quelques exemples d'applications dans lesquelles un capteur de détection de présence utilisant une caméra pourrait être utile. Ensuite, une méthode pour caractériser l'activité des personnes présentes dans une pièce a été présentée. Cette information, très simple à calculer, sera utile pour les systèmes de régulation du chauffage. Ensuite, nous avons présenté un système de stéréovision multicapteurs utilisant une caméra infrarouge et une caméra visible. L'utilisation de ces deux technologies permet de diminuer le nombre de fausses détections

liées à l'utilisation de chacune d'elle en fusionnant les résultats de détection de chaque spectre. Nous avons également présenté des résultats expérimentaux portant sur la localisation dans l'espace des personnes détectées. Cette localisation pourra fournir des informations utiles dans le cadre du maintien à domicile. Finalement, une méthode de détection d'événements anormaux a été présentée. Celle-ci se base sur les statistiques de co-occurrence des pixels de l'avant-plan dans un même voisinage spatio-temporel.

Les méthodes de caractérisation de l'activité et de localisation des personnes présentées dans ce chapitre utilisent tout ou partie des résultats de la méthode de détection de personnes présentée dans le chapitre 3. En effet, ce sont souvent les résultats de l'étape de détection et de suivi d'objets mobiles qui permettent de récupérer des informations plus haut-niveau ou de faciliter leur obtention. Par exemple, il est beaucoup plus facile de localiser les visages des personnes présentes dans une pièce pour les identifier à partir de leur position dans l'image. Ensuite, pour reconnaître l'activité des personnes (comme une chute par exemple) il est souvent nécessaire d'être capable de les suivre sur plusieurs images consécutives afin de construire une représentation spatio-temporelle de l'action. Les méthodes présentées dans ce chapitre sont donc la suite de la méthode de détection du chapitre 3.

Cependant, lorsque cela est possible, il peut être intéressant de s'affranchir de l'architecture en cascade pour éviter les risques de propagation des erreurs en utilisant des méthodes basées sur des caractéristiques bas-niveau de la vidéo, comme la méthode de détection d'événements anormaux présentée dans ce chapitre.

Conclusion Générale

Cette thèse présente les travaux menés dans le cadre du projet *CAPTHOM* du pôle de compétitivité $S2E2$ (Sciences et Systèmes de l'Énergie Électrique) de la région Centre. L'objectif de ce projet est le développement d'un capteur de détection de la présence humaine dans des bâtiments résidentiels ou tertiaires. Dans ce contexte, cette thèse présente des solutions algorithmiques pour détecter des personnes dans des séquences d'images.

Après un état de l'art sur les méthodes de la littérature utilisées pour l'analyse de séquence d'images en vidéo-surveillance, nous avons présenté une étude comparative des méthodes de soustraction de l'arrière-plan et des méthodes de reconnaissance d'humains dans une image. Ces deux études comparatives ont permis de définir les outils que nous utilisons pour le système de détection. Il ressort de la première étude sur les algorithme de soustraction de l'arrière-plan que les bénéfices retirés de l'utilisation de méthodes complexes ne sont pas très marqués lorsque l'on travaille sur des vidéos ne présentant pas de difficultés particulières. Cette observation est utile pour beaucoup d'applications, comme *CAPTHOM*, travaillant en milieu intérieur et qui ont des ressources matérielles limitées. Dans la seconde étude portant sur la reconnaissance d'humains dans des images, nous avons comparé deux méthodes très utilisées aujourd'hui et analysé l'influence des différents paramètres sur les performances de la méthode de Viola et Jones [148]. Cette étude a permis de définir de façon claire les paramètres utilisés pour le système proposé dans le cadre de *CAPTHOM*.

Nous avons ensuite développé le système de détection de personnes proposé dans le cadre du projet *CAPTHOM*. Cette méthode s'articule autour de trois grandes étapes : la détection de changement, le suivi d'objets mobiles et la classification. La soustraction de l'arrière-plan permet de simplifier les traitements ultérieurs en localisant les régions d'intérêt dans l'image. La mise à jour du modèle de l'arrière-plan est réalisée à trois niveaux afin de prendre en compte les différentes variations possibles de l'environnement. Ensuite, à partir de la liste des composantes connectées détectées, nous établissons un historique de leurs déplacements dans le plan image. Le suivi est basé sur la combinaison de l'analyse des composantes connectées et le suivi

de points d'intérêt. Chaque région d'intérêt est analysée par plusieurs classifieurs par parties pour déterminer leur nature. Nous construisons alors un indice de confiance sur l'appartenance de cet objet à la classe "humain". Cette méthode a été évaluée sur une large base de vidéos correspondant aux scénarios de référence, établis par les partenaires du projet, auxquels le système doit être capable de répondre.

Le système de détection de personnes proposé répond au cahier des charges de *CAPTHOM*. En effet, cet algorithme est capable de détecter des personnes avec un taux de détection d'environ 97% pour 3% de fausses détections. Il est robuste aux différentes postures, aux occultations partielles, aux variations de l'environnement, aux stimuli de fausses détections (objets mobiles) et est capable de détecter les personnes à plus de 15 mètres. Ensuite, la complexité algorithmique a été considérée avec attention tout au long du processus de conception de l'algorithme. Même si à l'heure actuelle, la preuve d'embarquabilité de ce système n'a pas été formellement apportée, nous avons participé à l'élaboration d'un prototype avec la société ST Imaging, experte dans les systèmes de vision sur matériel embarqué. Enfin, le composant *CAPTHOM* devant envoyer ses informations aux autres éléments du système de gestion techniques du bâtiment, nous avons également développé le module de communication du système présenté, en partenariat avec la société Agilicom, en se basant sur le protocole *Modbus*.

Cette méthode présente cependant quelques limitations intrinsèques. Tout d'abord, la soustraction de l'arrière-plan permet d'obtenir de bons résultats globaux mais, paradoxalement, est la cause d'une très grande partie des 3% des erreurs observées. Même si nous avons présenté ici une méthode pour prendre en compte les variations de l'environnement lors de la mise à jour du modèle, celle-ci reste délicate et des fausses détections peuvent être observées. De plus, afin de présenter de bonnes performances, les différents paramètres doivent être sélectionnés avec attention. Ensuite, l'architecture en cascade de la méthode proposée nécessite une bonne détection de l'avant-plan.

Nous avons ensuite présenté quelques applications possibles d'un capteur de présence humaine basé sur une caméra et proposé des méthodes permettant d'obtenir des informations de plus haut-niveau. Un critère de caractérisation de l'activité permettant, très simplement, d'avoir une information sur l'activité ou le degré d'agitation des personnes a tout d'abord été présenté. Cette information est utile pour des applications de régulation du chauffage. Ensuite, nous avons présenté un système de stéréovision multicapteurs basé sur une caméra infrarouge et une caméra dans le spectre visible. La combinaison de ces deux technologies permet de limiter le nombre de fausses détections inhérentes à l'utilisation de chacune indépendamment. En utilisant les paramètres du système de stéréovision, il est alors possible de localiser dans l'espace les personnes détectées. Cette information est utile dans le cadre du maintien à domicile. Une méthode pour modéliser l'activité d'une scène et détecter les événements anormaux dans une séquence d'images a ensuite été proposée. Celle-ci ne se base que sur des statistiques bas-niveau : les co-occurrences de pixels de l'avant-plan. Dans l'état de l'art, les méthodes utilisent généralement une architecture en cascade où les objets sont détectés, suivis puis leur chemin est analysé pour modéliser l'activité normale de ces objets mobiles. En utilisant des statistiques bas-niveau, il est possible de s'affranchir des risques de propagation des erreurs. Cette méthode de

détection étant générique, elle peut s'appliquer à la détection de plusieurs types d'activités anormales : détection de colis abandonnés, détection de chutes et également surveillance de flux routier *etc.*

Les perspectives de ces travaux sont de deux types. Tout d'abord, il y a des possibilités d'amélioration de la méthode de détection présentée ici. Comme expliqué précédemment, la soustraction de l'arrière-plan permet d'obtenir des résultats globaux très satisfaisants mais rend le paramétrage de l'application délicate et implique quelques fausses détections. Le choix d'utiliser la soustraction de l'arrière-plan a été réalisé pour compenser les performances du classifieur jugées pour l'instant insuffisantes pour une application industrielle. Il est cependant possible de mettre à profit le fait que la caméra est statique afin de spécialiser le classifieur. À l'heure actuelle, le classifieur a appris à reconnaître toutes les personnes (formes, postures, points de vues *etc*) sur tous les arrière-plans possibles. Il est possible de simplifier le problème de détection en utilisant les informations de son environnement pour établir le modèle discriminant entre les personnes et l'arrière-plan. Une procédure de mise à jour du modèle pourrait permettre d'inclure automatiquement de nouveaux exemples négatifs dans le modèle. En effet, seuls les exemples négatifs (l'arrière-plan) est spécifique à une scène. Si le classifieur ainsi adapté présente de bonnes performances de détection, quelques règles pragmatiques pourront également permettre de ne pas parcourir toute l'image à chaque instant. L'initialisation du suivi pourra être également réalisée à partir du résultat de détection et non plus à partir de la soustraction de l'arrière-plan.

Ensuite, même si le calcul des filtres de Haar est très rapide grâce aux images intégrales, celles-ci demandent beaucoup d'espace mémoire. Il serait intéressant de travailler à la conception d'un descripteur qui serait inspiré des filtres de Haar mais ne serait pas basé sur les images intégrales. On peut alors imaginer un descripteur composé d'un ensemble de différences des valeurs de paires ou de triplets de pixels. L'utilisation des pixels et non plus de régions comme dans les filtres de Haar rendra ce descripteur plus sensible au bruit mais l'utilisation d'un grand nombre de paires de pixels peut réduire cet effet indésiré.

Dans un deuxième temps, les perspectives de ces travaux portent sur l'utilisation du résultat du système de détection de personnes afin de construire d'autres méthodes recherchant des informations plus haut-niveau. En effet, la localisation et le suivi de personnes dans une séquence d'images permettent de construire des modèles spatio-temporels utiles à la reconnaissance d'activités. Il sera alors possible de travailler à la détection de chutes ou d'activités plus complexes (reconnaissance des activités quotidiennes ou interactions entre les personnes). Des travaux, utilisant le résultat de notre algorithme de détection, sont d'ores et déjà entrepris dans le cadre du projet CAPTHOM par Damien Brulin, doctorant au sein du projet *CAPTHOM*, pour la localisation dans l'espace des personnes détectées et la détection de chutes [20].

Finalement, afin d'avoir un système toujours plus informatif sur l'environnement, il pourrait également être intéressant de travailler sur le développement de méthodes de reconnaissance de visages. Nous avons actuellement commencé des travaux dans ce sens. Nous pensons que cette nouvelle information, couplée à la reconnaissance d'activité, pourra permettre de fournir des informations véritablement pertinentes

pour la conception de nouveaux produits visant l'aide au maintien à domicile.

Liste des publications de l'auteur

Journal international

1. Y. Benezeth, B. Emile, H. Laurent, C. Rosenberger, "Vision-based system for human detection and tracking in indoor environment", International Journal of Social Robotics, special issue on people detection and tracking (article accepté).

Conférences internationales

1. Y. Benezeth, P.M. Jodoin, V. Saligrama, C. Rosenberger, "Abnormal events detection based on spatio-temporal co-occurences", international conference on Computer Vision and Pattern Recognition (CVPR), pages 2458–2465, 2009.
2. Y. Benezeth, B. Emile, H. Laurent, C. Rosenberger, "A general framework for a robust human detection in images sequences", International Conference on Image and Graphics (ICIG), 2009.
3. Y. Benezeth, P.M. Jodoin, B. Emile, H. Laurent, C. Rosenberger, "Review and evaluation of commonly-implemented background subtraction algorithms", International Conference on Pattern Recognition (ICPR), 2008.
4. Y. Benezeth, B. Emile, H. Laurent, C. Rosenberger, "A real time human detection system based on far infrared vision", International Conference on Image and Signal Processing (ICISP), pages 76–84, 2008.
5. Y. Benezeth, B. Emile, C. Rosenberger, "Comparative study on foreground detection algorithms for human detection", International Conference on Image and Graphics (ICIG), pages 661–666, 2007.

Conférences nationales

1. Y. Benezeth, B. Emile, H. Laurent, C. Rosenberger, "Détection de la présence humaine et caractérisation de l'activité", GRETSI 2009.
2. Y. Benezeth, B. Emile, H. Laurent, A. Hafiane, "Reconnaissance par blocs : une alternative aux descripteurs locaux", GRETSI 2009.

3. Y. Benezeth, B. Emile, H. Laurent, C. Rosenberger, "Détection de la présence humaine par vision infrarouge : Application à la gestion de l'énergie électrique dans l'habitat", Bourges, Colloque Capteurs, 2008.

Autres communications

1. "Avancement des Travaux de Recherche", communication orale devant le consortium du projet *CAPTHOM*, le 7/05/09.
2. "Détection d'humains avec un système de stéréovision multi-capteurs ", communication orale à la réunion SCATI du GDR ISIS sur les systèmes de vision, le 02/04/09.
3. Participation à un stand lors du Colloque Capteurs 2009 à Bourges le 18 et 19/03/2009. Présentation d'un poster et d'un démonstrateur.
4. "Détection de la présence humaine : applications dans le spectre infrarouge et visible", communication orale à la réunion du GDR ISIS sur la vidéo surveillance intelligente, le 17/12/08.
5. "La détection de la présence humaine par vision. Avancement de la thèse", communication orale devant l'équipe ISS de l'Institut PRISME, le 12/06/08.
6. A. Belconde, Y. Benezeth, D. Brulin, P. David, E.A. Fall, "La recherche dans *CAPTHOM*", poster, colloque Capteurs 2008.
7. A. Belconde, Y. Benezeth, D. Brulin, P. David, E.A. Fall, "*CAPTHOM* : vers une évolution des détecteurs de présence humaine", poster, colloque Capteurs 2008.
8. "Détection de la présence humaine par vision infrarouge", communication orale devant le consortium du projet *CAPTHOM*, le 17/01/08.
9. "La détection de la présence humaine par vision", communication orale devant le consortium du projet *CAPTHOM*, le 6/03/07.

Table des figures

Liste des tableaux

Bibliographie

[1] http ://dparks.wikidot.com.

[2] http ://imagelab.ing.unimore.it/vssn06.

[3] http ://www.falldetection.com/ilifefds.asp.

[4] Efficacité énergétique des bâtiments 2007-2008, rapport du ministère de l'écologie du développement et de l'amménagement durable, 2007.

[5] T. Aach and A. Kaup. Bayesian algorithms for adaptive change detection in image sequences using markov random fields. *Signal Processing : Image Communication*, 7 :147–160, 1995.

[6] Y. Abramson. *AdaBoost/GA et filtrage particulaire : La vision par ordinateur au service de la sécurité routière*. PhD thesis, Centre de robotique, Ecole Nationale Supérieure des Mines de Paris, 2006.

[7] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz. Robust realtime unusual event detection using multiple fixed-location monitors. *Pattern Analysis and Machine Intelligence*, 30 :555–560, 2008.

[8] J. K. Aggarwal and Q. Cai. Human motion analysis : a review. *Computer Vision and Image Understanding*, 73 :90–102, 1999.

[9] V. Ayala-Ramirez, C. Parra, and M. Devy. Active tracking based on hausdorff matching. *International Conference on Pattern Recognition*, 4 :706–709, 2000.

[10] D.H. Ballard and C.M. Brown. *Computer Vision*. Prentice Hall Professional Technical Reference, 1982.

[11] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Surf : Speeded up robust features. *Computer Vision and Image Understanding*, 110(3) :346–359, 2008.

[12] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. *International Conference on Computer Vision*, pages 454 – 461, 2001.

[13] A.F. Bobick and J.W. Davis. The recognition of human movement using temporal templates. *Pattern Analysis and Machine Intelligence*, 23 :257–267, 2001.

[14] J.Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker : Description of the algorithm. Technical report, Intel Corporation, Microprocessor Research Labs, 1999.

[15] A.C. Bovik. *Handbook of Image and Video Processing*. Academic Press, Inc. Orlando, FL, USA, 2005.

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

[17] A. Branca, M. Leo, G. Attolico, A. Distante, and E. Immagini. People detection in dynamic images. *International Joint Conference on Neural Networks*, 3 :2428–2432, 2002.

[18] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. *international conference on Computer Vision and Pattern Recognition*, pages 994–999, 1997.

[19] L. Brown, A. Senior, Y. Tian, J. Vonnel, A Hampapur, C. Shu, H. Merkl, and M. Lu. Performance evaluation of surveillance systems under varying conditions. *Performance Evaluation of Tracking Systems Workshop*, pages 1–8, 2005.

[20] D. Brulin and E. Courtial G. Allibert. Visual receding horizon estimation for human presence detection. *People Detection and Tracking Workshop of the International Conference on Robotics and Automation Workshop*, 2009.

[21] H.H. Bui, S. Venkatesh, and G. West. Policy recognition in the abstract hidden markov model. *Journal of Artificial Intelligence Research*, 17 :451–499, 2002.

[22] A. Calway, W. Mayol-Cuevas, M. Pupilli, D. Chekhlov, and A. Gee. Real-time camera tracking using particle filtering. *British Machine Vision Conference*, pages 519–528, 2005.

[23] T.H. Chalidabhongse, K. Kim, D. Harwood, and L. Davis. A perturbation method for evaluating background subtraction algorithms. *international Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2003.

[24] H. Chang, Haizhou, T. Yamashita, L. Shihong, and M. Kawade. Incremental learning of boosted face detector. *International Conference on Computer Vision*, pages 1–8, 2007.

[25] T. Chen, H. Haussecker, A. Bovyrin, R. Belenov, K. Rodyushkin, A. Kuranov, and V. Eruhimov. Computer vision workload analysis : case study of video surveillance systems. *Intel. Technology Journal*, 9 :109–118, 2005.

[26] A. Choksuriwong, H. Laurent, and B. Emile. Comparison of invariant descriptors for object recognition. *International Conference on Image Processing*, pages 377–380, 2005.

[27] C.-W. Chong, P. Raveendran, and R. Mukundan. Mean shift : A comparative analysis of algorithms for fast computation of zernike moment. *Pattern Recognition*, 36 :731–742, 2003.

[28] H. Chouaib, O.R. Terrades, S. Tabbone, F. Cloppet, and N. Vincent. Feature selection combining genetic algorithm and adaboost classifiers. *International Conference on Pattern Recognition*, 2008.

[29] R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, and O. Hasegawa. A system for video surveillance and monitoring. Technical report, Robotics Institute, Carnegie Mellon University, 2000.

[30] D. Comaniciu and P. Meer. Mean shift : A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence*, 24 :603–619, 2002.

[31] D. Comaniciu, V. Ramesh, and P.Meer. Real-time tracking of non-rigid objects using mean shift. *international conference on Computer Vision and Pattern Recognition*, 2 :142–149, 2000.

[32] R. Cutler and M. Turk. View-based interpretation of real-time optical flow for gesture recognition. *International Conference on Automatic Face and Gesture Recognition*, pages 416–421, 1998.

[33] N. Dalal. *Finding people in images and videos.* PhD thesis, Institut National Polytechnique de Grenoble / INRIA Rhône-Alpes, 2006.

[34] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *international conference on Computer Vision and Pattern Recognition*, 1 :886–893, 2005.

[35] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. *European Conference on Computer Vision*, 2 :428–441, 2006.

[36] P. David. *Contribution à l'analyse de sûreté de fonctionnement des systèmes complexes en phase de conception : application à l'évaluation des missions d'un réseau de capteurs de présence humaine.* PhD thesis, Université d'Orléans - Institut Prisme, 2009.

[37] J. Davis and M. Keck. A two-stage approach to person detection in thermal imagery. *Workshop on Applications of Computer Vision*, 2005.

[38] J. Davis and M. Shah. Gesture recognition. Technical report, University of Central Florida, 1993.

[39] J. Davis and V. Sharma. Background-subtraction using contour-based fusion of thermal and visible imagery. *Computer Vision and Image Understanding*, pages 162–182, 2007.

[40] J.W. Davis and A.F. Bobick. The representation and recognition of action using temporal templates. *international conference on Computer Vision and Pattern Recognition*, pages 928–934, 1997.

[41] S. Denman, V. Chandrana, and S. Sridharan. An adaptive optical flow technique for person tracking systems. *Pattern Recognition Letter*, 28(10) :1232–1239, 2007.

[42] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection : A benchmark. *international conference on Computer Vision and Pattern Recognition*, pages 304–311, 2009.

[43] A.A. Efros, A.C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. *International Conference on Computer Vision*, 2 :726–733, 2003.

[44] A Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. *European Conference on Computer Vision*, pages 751–767, 2000.

[45] A. Ess, B. Leibe, K. Schindler, , and L. van Gool. A mobile vision system for robust multi-person tracking. *international conference on Computer Vision and pattern Recognition*, 2008.

[46] M. Everingham, A. Zisserman, C. Williams, L. Van Gool, M. Allan, C. Bishop, O. Chapelle, N. Dalal, T. Deselaers, and G. Dorko. The 2005 pascal visual object classes challenge. *First PASCAL Challenge Workshop*, 2005.

[47] M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool. The 2005 pascal visual object classes challenge. *Machine Learning Challenges*, pages 117–176, 2006.

[48] S.A. Fahmy, P.Y.K. Cheung, and W. Luk. Hardware acceleration of hidden markov model decoding for person detection. *international conference on Design, Automation and Test in Europe*, 3 :8–13, 2005.

[49] M.E. Farmer, Rein-Lien Hsu, and A.K. Jain. Interacting multiple model kalman filters for robust high speed human motion tracking. *International Conference on Pattern Recognition*, 2 :20–23, 2002.

[50] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. *international conference on Computer Vision and Pattern Recognition*, 2008.

[51] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression : a statistical view of boosting. *Annals of Statistics*, 28, 1998.

[52] W. Förstner. A feature-based correspondence algorithm for image matching. *International Archives of Photogrammetry and Remote Sensing*, 26 :150–166, 1986.

[53] W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. *Intercommision Conference on Fast Processing of Photogrammetric Data*, 50 :281–305, 1987.

[54] J. Gall and V. Lempitsky. Class-specific hough forests for object detection. *international conference of Computer Vision and Pattern Recognition*, pages 1022–1029, 2009.

[55] D. M. Gavrila. The visual analysis of human movement : a survey. *Computer Vision and Image Understanding*, 73 :82–98, 1999.

[56] D.M. Gavrila, J. Giebel, and S. Munder. Vison-based pedestrian detection : The protector system. *Intelligent Vehicles Symposium*, pages 13–18, 2004.

[57] D.M. Gavrila and V. Philomin. Real-time object detection for "smart" vehicles. *International Conference on Computer Vision*, 1 :87–93, 1999.

[58] D. Gerónimo, A.D. Sappa, A. López, and D. Ponsa. Adaptive image sampling and windows classification for on-board pedestrian detection. *International Conference on Computer Vision Systems*, 2007.

[59] N. Ghanem, D. DeMenthon, D. Doermann, and L. Davis. Representation and recognition of events in surveillance video using petri nets. *Computer Vision and Pattern Recognition Workshop*, pages 112–120, 2004.

[60] S. Ghidary, Y. Nakata, T. Takamori, and M. Hattori. Human detection and localization at indoor environment by homerobot. *International Conference on Systems, Man, and Cybernetics*, 2 :1360–1365, 2000.

[61] F. Ghorbel. A complete invariant description for gray-level images by the harmonic analysis approach. *Pattern Recognition Letters*, 15 :1043–1051, 1994.

[62] V. Girondel, A. Caplier, and L. Bonnaud. Real time tracking of multiple persons by kalman filtering and face pursuit for multimedia applications. *Symposium on Image Analysis and Interpretation*, pages 201 – 205, 2004.

[63] J.F. Gobeau. Détecteurs de mouvement à infrarouge passif (détecteurs irp). *Colloque Capteurs*, 2008.

[64] H. Grabner and H. Bischof. On-line boosting and vision. *international conference on Computer Vision and Pattern Recognition*, pages 260–267, 2006.

[65] A. Gritai, Y. Sheikh, and M. Shah. On the use of anthropometry in the invariant analysis of human actions. *International Conference on Pattern Recognition*, 2 :923–926, 2004.

[66] C. Gu, J. J. Lim, P. Arbelàez, and J. Malik. Recognition using regions. *international conference on Computer Vision and Pattern Recognition*, pages 1030–1037, 2009.

[67] T. Harada, T. Sato, and T. Mori. Human motion tracking system based on skeleton and surface integration model using pressure sensors distribution bed. *Workshop on Human Motion*, pages 99–106, 2000.

[68] I. Haritaoglu, D. Harwood, and L.S. Davis. W 4 : real-time surveillance of people and their activities. *Pattern Analysis and Machine Intelligence*, 22 :809–830, 2000.

[69] C. Harris and M. Stephens. A combined corner and edge detector. *Alvey Vision Conference*, 15 :147–151, 1998.

[70] M. Harville, G. Gordon, and J. Woodfill. Foreground segmentation using adaptive mixture models in color and depth. *Workshop on detection and recognition of events in video*, pages 3–11, 2001.

[71] Qiang He and C. Debrunner. Individual recognition from periodic activity using hidden markov models. *Human Motion Workshop*, pages 47–52, 2000.

[72] B. Hemery, H. Laurent, and C. Rosenberger. Evaluation metric for image understanding. *International Conference on Image Processing*, 2009.

[73] B. Hemery, H. Laurent, C. Rosenberger, and B. Emile. Evaluation protocol for localization metrics application to a comparative study. *International Conference on Image and Signal Processing*, pages 273–280, 2008.

[74] S. Hongeng, F. Bremond, and R. Nevatia. Bayesian framework for video surveillance application. *International Conference on Pattern Recognition*, 1 :164–170, 2000.

[75] M. K. Hu. Visual pattern recognition by moment invariants. *Information Theory*, 8 :179–187, 1962.

[76] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *Systems, Man, and Cybernetics, Part C : Applications and Reviews,*, 34(3) :334–352, 2004.

[77] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank. A system for learning statistical motion patterns. *Pattern Analysis and Machine Intelligence*, 28(9) :1450–1464, 2006.

[78] E. Hunter, J. Schlenzig, and R. Jain. Posture estimation in reduced-model gesture input systems. *International Workshop on Automatic Face and Gesture Recognition*, pages 290–295, 1995.

[79] S. Huwer and H. Niemann. Adaptive change detection for real-time surveillance applications. *Workshop on Visual Surveillance*, pages 37–43, 2000.

[80] S.S. Intille and A.F. Bobick. Closed-world tracking. *International Conference on Computer Vision*, pages 672–678, 1995.

[81] D.-S. Jang, S.-W. Jang, and H.-I. Choi. 2d human body tracking with structural kalman filter. *Pattern Recognition*, 35(10) :2041–2049, 2002.

[82] O. Javed and M. Shah. Tracking and object classification for automated surveillance. *European Conference on Computer Vision*, pages 343–357, 2002.

[83] P.-M. Jodoin, J. Konrad, and V. Saligrama. Modeling background activity for behavior subtraction. *International Conference on Distributed Smart Cameras*, 2008.

[84] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. *Imaging and Vision Computing*, 14 :609–615, 1996.

[85] M. Jones and D. Snow. Pedestrian detection using boosted features over many frames. *International Conference on Pattern Recognition*, pages 1–4, 2008.

[86] I. Junejo, O. Javed, and M. Shah. Multi feature path modeling for video surveillance. *International Conference on Pattern Recognition*, pages 716–719, 2004.

[87] G. Junxia, D. Xiaoqing, W. Shengjin, and W. Youshou. Full body tracking-based human action recognition. *International Conference on Pattern Recognition*, pages 1–4, 2008.

[88] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. *Workshop on Advanced Video-based Surveillance Systems conference*, 2001.

[89] R. Kehl, M. Bray, and L.Van Gool. Full body tracking from multiple views using stochastic sampling. *interantional conference on Computer Vision and Pattern Recognition*, 2 :129–136, 2005.

[90] A. Khotanzad and Y. Hua Hong. Invariant image recognition by zernike moments. *Pattern Analysis and Machine Intelligence*, 12(5) :489–497, 1990.

[91] H.-J. Kim and KK.-M. Lee. Silhouette-based human motion estimation for movement education of young children. *International Conference on Hybrid Information Technology*, 2 :673–678, 2006.

[92] K. Kim, T.H. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11 :172–185, 2005.

[93] P. Kumar, K. Sengupta, and A. Lee. A comparative study of different color spaces for foreground and shadow detection for traffic monitoring system. *international Conference on Intelligent Transportation Systems*, 2002.

[94] Y. Kuno, T. Watanabe, Y. Shimosakoda, and S. Nakagawa. Automated detection of human for visual surveillance system. *International Conference on Pattern Recognition*, pages 865–869, 1996.

[95] D.S. Lee. Effective gaussian mixture learning for video background subtraction. *Pattern Analysis and Machine Intelligence*, 27 :827–832, 2005.

[96] R. Li, Y. Chen, and X. Zhang. Fast robust eigen-background updating for foreground detection. *International Conference on Image Processing*, pages 1833–1836, 2006.

[97] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. *International Conference on Image Processing*, 1 :900–903, 2002.

[98] Z. Lin, L.S. Davis, D. Doermann, and D. DeMenthon. Hierarchical part-template matching for human detection and segmentation. *International Conference on Computer Vision*, pages 1–8, 2007.

[99] A.J. Lipton, H. Fujiyoshi, and R.S. Patil. Moving target classification and tracking from real-time video. *Applications of Computer Vision Workshop*, pages 8–14, 1998.

[100] G. Liu, X. Tang, J. Huang, J. Liu, and D. Sun. Hierarchical model-based human motion tracking via unscented kalman filter. *International Conference on Computer Vision*, pages 1–8, 2007.

[101] D.G. Lowe. Object recognition from local scale-invarient features. *International Conference on Computer Vision*, 2 :1150–1157, 1999.

[102] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60 :91–110, 2004.

[103] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[104] F. Lv and R. Nevatia. Single view human action recognition using key pose matching and viterbi path searching. *international conference on Computer Vision and Pattern Recognitio*, pages 1–8, 2007.

[105] A. Madabhushi and J.K. Aggarwal. A bayesian approach to human activity recognition. *Visual Surveillance Workshop*, pages 25–32, 1999.

[106] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *International Conference on Computer Vision*, 2 :416–423, 2001.

[107] M. Meuter, D. Muller, S. Muller-Schneiders, U. Iurgel, S. Park, and A. Kummert. Pedestrian tracking from a moving host using corner points. *Advances in Visual Computing*, pages 367–376, 2007.

[108] D. Meyer, J. Denzler, and H. Niemann. Model based extraction of articulated objects in image sequences for gait analysis. *International Conference on Image Processing*, 3 :78–81, 1997.

[109] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. *European Conference on Computer Vision*, 3021 :69–82, 2004.

[110] T.B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, 104 :90–126, 2006.

[111] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *Pattern Analysis and Machine Intelligence*, 23(4) :349–361, 2001.

[112] R.M. Neal and G.E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. *Learning in Graphical Models*, pages 355–368, 1998.

[113] A.T. Nghiem, F. Bremond, M. Thonnat, and V. Valentin. Etiseo, performance evaluation for video surveillance systems. *international conference on Advanced Video and Signal Based Surveillance*, pages 476–481, 2007.

[114] N.T. Nguyen, S. Venkatesh, G. West, H.H. Bui, and A. Perth. Multiple camera coordination in a surveillance system. *Acta Automatica Sinica*, 29(3) :408–422, 2003.

[115] K. Nickels and S. Hutchinson. Model-based tracking of complex articulated objects. *Robotics and Automation*, 17 :28–36, 2001.

[116] N.M. Oliver, B. Rosario, and A.P. Pentland. A bayesian computer vision system for modeling human interactions. *Pattern Analysis and Machine Intelligence*, 22 :831–843, 2000.

[117] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. *international conference on Computer Vision and Pattern Recognition*, pages 193 –199, 1997.

[118] G. Overett, L. Petersson, N. Brewer, L. Andersson, and N. Pettersson. A new pedestrian dataset for supervised learning. *In Intelligent Vehicles Symposium*, pages 373–378, 2008.

[119] S. Panahi, S. Sheikhi, S. Hadadan, and N. Gheissari. Evaluation of background subtraction methods. *international conference on Digital Image Computing : Techniques and Applications*, pages 357–364, 2008.

[120] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38 :15–33, 2000.

[121] V. Parameswaran and R. Chellappa. View invariance for human action recognition. *International Journal of Computer Vision*, 66(1) :83–101, 2006.

[122] P. Perez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. *European Conference on Computer Vision*, pages 661–675, 2002.

[123] PETS'2001. 2nd international workshop on performance evaluation of tracking and surveillance. 2001.

[124] M.-T. Pham and T.-J. Cham. Online learning asymmetric boosted classifiers for object detection. *international conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[125] M. Piccardi and T. Jan. Mean-shift background image modelling. *International Conference on Image Processing*, 5 :3399–3402, 2004.

[126] K. Rangarajan and M. Shah. Establishing motion correspondence. *international conference on Computer Vision and Pattern Recognition*, pages 103–108, 1991.

[127] C. Rao, A. Gritai, M.Shah, and T. Syeda-Mahmood. View-invariant alignment and matching of video sequences. *International Conference on Computer Vision*, 2 :939–945, 2003.

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

[129] J. Rymel, J. Renno, D. Greenhill, J. Orwell, and G.A. Jones. Adaptive eigen-backgrounds for object detection. *International Conference on Image Processing*, pages 1847–1850, 2004.

[130] K. Shafique and M. Shah. A non-iterative greedy algorithm for multi-frame point correspondence. *Pattern Analysis and Machine Intelligence*, pages 51–65, 2005.

[131] Y. Shan, F. Yang, and R. Wang. Color space selection for moving shadow elimination. *International Conference on Image and Graphics*, pages 496–501, 2007.

[132] J. Shi and C. Tomasi. Good features to track. *international conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[133] J. Shin, S. Kim, S. Kang, S.-W. Lee, J. Paik, B. Abidi, and M. Abidi. Optical flow-based real-time object tracking using non-prior training active feature mode. *Real-Time Imaging*, 11 :204–218, 2005.

[134] H. Sidenbladh. Detecting human motion with support vector machines. *Conference on Computer Vision and Pattern Recognition*, 2 :188–191, 2004.

[135] N.T. Siebel and S. Maybank. Fusion of multiple tracking algorithms for robust people tracking. *European Conference on Computer Vision*, pages 373–387, 2002.

[136] X. Song and R. Nevatia. Combined face-body tracking in indoor environment. *International Conference on Pattern Recognition*, 4 :159–162, 2004.

[137] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. *international conference on Computer Vision and Pattern Recognition*, 2, 1999.

[138] T. Syeda-Mahmood, A. Vasilescu, and S. Sethi. Recognizing action events from multiple viewpoints. *Detection and Recognition of Events in Video Workshop*, pages 64–72, 2001.

[139] C.Y. Tang, Z. Chen, and Y.P. Hung. Automatic detection and tracking of human heads using an active stereo vision system. *International Journal of Pattern Recognition and Artificial Intelligence*, 14(2) :137–166, 2000.

[140] M. Teague. Image analysis via the general theory of moments. *Journal on Optical Society of America*, 70 :920–930, 1980.

[141] D. Thirde, L. Li, and J. Ferryman. An overview of the pets 2006 dataset. *international Workshop on Performance Evaluation of Tracking and Surveillance*, pages 47–50, 2006.

[142] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical report, Carnegie Mellon University, 1991.

[143] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower : principles and practice of background maintenance. *International Conference on Computer Vision*, 1 :255–261, 1999.

[144] Y. Tsuduki and H. Fujiyoshi. A method for visualizing pedestrian traffic flow using sift feature point tracking. *Pacific Rim Symposium on Advances in Image and Video Technology*, 5414 :25–36, 2009.

[145] S. Ullman, E. Sali, and M. Vidal-Naquet. A fragment-based approach to object representation and classification. *International Workshop on Visual Form*, pages 85–102, 2001.

[146] A. Utsumi and N. Tetsutani. Human detection using geometrical pixel value structures. *International Conference on Automatic Face and Gesture Recognition*, pages 34–39, 2002.

[147] C. J. Veenman, M. J. T. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *Pattern Analysis and Machine Intelligence*, 23 :54–72, 2001.

[148] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *international conference on Computer Vision and Pattern Recognition*, 1 :511–518, 2001.

[149] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63 :153–161, 2005.

[150] D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3d exemplar. *International Conference on Computer Vision*, 5 :1–7, 2007.

[151] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2) :249–257, 2006.

[152] C. Wojek, S. Walk, and B. Schiele. Multi-cue onboard pedestrian detection. *international conference on Computer Vision and Pattern Recognition*, pages 794–801, 2009.

[153] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder : Real-time tracking of the human body. *Pattern Analysis and Machine Intelligence*, 1997.

[154] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. *International Conference on Computer Vision*, pages 90–97, 2005.

[155] B. Wu and R. Nevatia. Tracking of multiple, partially occluded humans based on static body part detection. *international Conference on Computer Vision and Pattern Recognition*, 1 :951–958, 2006.

[156] B. Wu and R. Nevatia. Improving part based object detection by unsupervised, online boosting. *international conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[157] S. Xiang, F. Nie, Y. Song, and C. Zhang. Contour graph based human tracking and action sequence recognition. *Pattern Recognition*, 41(12) :3653–3664, 2008.

[158] M. Yasuno, S. Ryousuke, N. Yasuda, and M. Aoki. Pedestrian detection and tracking in far infrared images. *Computer Vision and Pattern Recognition Workshop*, pages 182–187, 2005.

[159] A. Yilmaz, O. Javed, and M. Shah. Object tracking : A survey. *ACM Computing surveys*, 38(4), 2006.

[160] A. Yilmaza and M. Shah. Matching actions in presence of camera motion. *Computer Vision and Image Understanding*, 104(2) :221–231, 2006.

[161] S.M. Yoon and H. Kim. Real-time multiple people detection using skin color, motion and appearance information. *International Workshop on Robot and Human Interactive Communication*, pages 331–334, 2004.

[162] Q. Zhao, J. Kang, H. Tao, and W. Hua. Part-based human tracking in a multiple cues fusion framework. *International Conference on Pattern Recognition*, 1 :450–455, 2006.

[163] J. Zhong and S. Sclaroff. Segmenting foreground objects from a dynamic textured background via a robust kalman filter. *International Conference on Computer Vision*, 1 :44–50, 2003.

[164] Q. Zhou and J. Aggarwal. Tracking and classifying moving objects from video. *Performance Evaluation of Tracking Systems Workshop*, 2001.

[165] Q. Zhu, S. Avidan, M. Ye, and K.T. Cheng. Fast human detection using a cascade of histograms of oriented gradients. *international conference on Computer Vision and Pattern Recognition*, 2 :1491–1498, 2006.

[166] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. *International Conference on Pattern Recognition*, 2004.

[167] Z. Zivkovic and B.Krose. An em-like algorithm for color-histogram-based object tracking. *international conference on Computer Vision and Pattern Recognition*, pages 798–803, 2004.

[168] Z. Zivkovic and F. van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters*, 27 :773–780, 2006.

Yannick BENEZETH

Détection de la présence humaine par vision

Les travaux présentés dans ce manuscrit traitent de la détection de personnes dans des séquences d'images et de l'analyse de leur activité. Ces travaux ont été menés au sein de l'institut PRISME dans le cadre du projet CAPTHOM du pôle de compétitivité $S2E2$.
Après un état de l'art sur l'analyse de séquences d'images pour l'interprétation automatique de scènes et une étude comparative de modules de vidéo-surveillance, nous présentons la méthode de détection de personnes proposée dans le cadre du projet CAPTHOM. Celle-ci s'articule autour de trois étapes : la détection de changement, le suivi d'objets mobiles et la classification. Chacune de ces étapes est décrite dans ce manuscrit. Ce système a été évalué sur une large base de vidéos correspondant à des scénarios de cas d'usage de CAPTHOM établis par les partenaires du projet.
Ensuite, nous présentons des méthodes permettant d'obtenir, à partir du flux vidéo d'une ou deux caméras, d'autres informations de plus haut-niveau sur l'activité des personnes détectées. Nous présentons tout d'abord une mesure permettant de quantifier leur activité. Ensuite, un système de stéréovision multi-capteurs combinant une caméra infrarouge et une caméra visible est utilisé pour augmenter les performances du système de détection mais aussi pour permettre la localisation dans l'espace des personnes et donc accéder à une cartographie de leurs déplacements. Finalement, une méthode de détection d'événements anormaux, basée sur des statistiques de distributions spatiales et temporelles des pixels de l'avant-plan est détaillée. Les méthodes proposées offrent un panel de solutions performantes sur l'extraction d'informations haut-niveau à partir de séquences d'images.
Mots clés : Analyse vidéo, détection de personnes, soustraction de l'arrière-plan, suivi, classification.

Human detection using computer vision

The work presented in this manuscript deals with people detection and activity analysis in images sequences. This work has been done in the PRISME institut within the framework of the *CAPTHOM* project of the French Cluster $S2E2$.
After a state of the art on video analysis and a comparative study of several video surveillance tools, we present the people detection method proposed within the framework of the CAPTHOM project. This method is based on three steps : change detection, mobile objects tracking and classification. Each steps is described in this thesis. The system was assessed on a wide videos dataset.
Then, we present methods used to obtain other high-level information concerning the activity of detected persons. A criterion for characterizing their activity is presented. Then, a multi-sensors stereovision system combining an infrared and a daylight camera is used to increase performances of the people detection system but also to localize persons in the 3D space and so build the moving cartography. Finally, an abnormal events detection method based on statistics about spatio-temporal foreground pixel distribution is presented. These proposed methods offer robust and efficient solutions on high-level information extraction from images sequences.
Key words : video analysis, people detection, background subtraction, tracking, classification.

Institut PRISME
École Nationale Supérieure
d'Ingénieurs de Bourges
88 boulevard Lahitolle
18020 Bourges Cedex