

**INSTITUT DE LA FRANCOPHONIE POUR L'INFORMATIQUE
INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE
ET EN AUTOMATIQUE**



**INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE**



INRIA

centre de recherche **SOPHIA ANTIPOLIS - MÉDITERRANÉE**

MEMOIRE DE FIN D'ETUDES

MASTER D'INFORMATIQUE

**REPARATION DES TRAJECTOIRES DE PERSONNES
SUIVIES BASEE SUR LE CLUSTERING DE POINTS**

ETUDIANT : CHAU Duc Phu

PROMOTION : XII

Sous la direction de :

François BREMOND, HDR

HANOI, 10 - 2008

TABLE DES MATIERES

LISTE DES FIGURES ET DE TABLEAU	2
REMERCIEMENTS	3
RESUME.....	4
ABSTRACT	5
CHAPITRE I INTRODUCTION	6
1.2 Motivation et Objectif	7
1.3 Contribution.....	8
1.4 Environnement de Stage.....	8
CHAPITRE II ETAT DE L'ART	10
2.1 Détection et Suivi du Mouvement.....	10
2.1.1 Détection du mouvement.....	10
2.1.2 Suivi du Mouvement	11
2.2 Amélioration de l'Algorithme de Suivi.....	12
2.3 Conclusion.....	14
CHAPITRE III PRESENTATION DU SYSTEME D'INTERPRETATION DE VIDEO.....	15
3.1 Détection du Mouvement et Suivi Trame par Trame	16
3.2 Combinaison de Multiples Caméras.....	16
3.3 Suivi à long terme des individus, groupe de personnes et la foule.....	18
CHAPITRE IV REPARATION DES TRAJECTOIRES PAR CLUSTERING DE POINTS	19
4.1 Notre Approche	19
4.2 Caractéristiques d'une trajectoire	20
4.3 Déterminer les Poids des Caractéristiques par l'Algorithme Génétique et l'Apprentissage.....	22
4.3.1 Mutation et Cross-Over pour l'Algorithme Génétique.....	23
4.3.2 Algorithme génétique	26
4.4 Types de Zones Utilisées dans la Scène	27
4.5 Apprentissage des Zones	28
4.6 Calcul des Triplets de Zones	30
4.7 Réparer les Trajectoires Perdues	32
CHAPITRE V EXPERIMENTATION ET VALIDATION.....	34
CHAPITRE VI CONCLUSION ET PERSPECTIVE	39
REFERENCES.....	41

LISTE DES FIGURES ET DE TABLEAU

Fig 2.1 Illustration de la différence d'images pour détecter le mobile.....	11
Fig 2.2 Le cycle récursif de Kalman.....	12
Fig 3.1 Système d'interprétation de vidéo.....	15
Fig 3.2 Illustration du processus de combinaison de multiples caméras.....	17
Fig 4.1. Illustration d'opérateur Mutation	25
Fig 4.2. Illustration d'opérateur 'Cross-Over'	25
Fig 4.3. Les pas de réalisation d'algorithme génétique	26
Fig 4.4 Description d'une 'entry zone'	28
Fig 4.5 Description d'une 'lost-found zone'	28
Fig 4.6. Résultat de clustering des 8 'lost zones'	29
Fig 4.7 Représentation des clusters par les polygones.	30
Fig 4.8 Illustration de création d'un triplet de zones.....	31
Fig 4.9 Illustration de l'algorithme de réparation des trajectoires.....	33
Fig 5.1 Images capturées avant et après la fusion de la trajectoire à $t = 711$ s.....	37
Fig 5.2 Images capturées avant et après la fusion de la trajectoire à $t = 903$ s.....	38
Tableau 5.1 Statistique les donnés dans deux cas : appliquer et sans appliquer l'algorithme de réparation	36

REMERCIEMENTS

Avant de vous présenter ce rapport, je tiens à remercier tous ceux qui m'ont aidé pendant mon stage et mes études à l'IFI. Je voudrais en particulier remercier :

- M. François Brémond, chercheur de l'équipe Pulsar, INRIA Sophia Antipolis, pour l'encadrement, l'aide, l'encouragement et la sympathie qu'il m'a donnés. Grâce à ses conseils, j'ai pu terminer et compléter mes travaux.

- Mme Monique Thonnat, chef de l'équipe Pulsar, INRIA Sophia Antipolis, qui m'a donné beaucoup de conseils et de bonnes critiques sur mon travail. Grâce à ses remarques, mon travail est plus complet.

- Etienne Corvée, ingénieur de l'équipe Pulsar, INRIA Sophia Antipolis, qui m'a aidé chaleureusement. Sans son aide, le stage n'aurait pas pu atteindre son terme à temps. Je remercie aussi tous les autres membres de l'équipe Pulsar pour leurs encouragements, leurs aides, leur sympathie et l'environnement de travail très chaleureux dans lequel j'ai travaillé.

- LE Mélanie Nguyen My Linh, une amie vietnamienne, qui m'a aidé pendant mon séjour de six mois en France.

Je voudrais aussi adresser mes sincères remerciements à tous les professeurs de l'IFI pour leurs enseignements et les cours intéressants qu'ils m'ont donné pendant ma scolarité à l'IFI. Je n'oublie pas de remercier aussi tous les personnels de l'IFI qui m'ont apporté leur aide.

Depuis mes premiers jours dans cet institut, j'ai reçu beaucoup d'aides, de conseils et d'encouragements de mes amis, en particulier ceux de la promotion 12. Tout cela m'a permis de murir chaque jour. Je les remercie et je ne pourrais jamais oublier les souvenirs gais et tristes que j'ai passé avec eux durant ces deux ans à l'IFI.

Je voudrais aussi remercier aussi les confrères de l'Université Privée Phu Xuan où je suis en train de travailler, qui m'ont donné les meilleures conditions pour que je puisse bien passer ma scolarité à l'IFI.

Enfin, un grand merci à mes parents, mon frère, ma fiancée et les autres membres de ma famille de m'avoir énormément encouragé dans les moments les plus difficiles de ma scolarité à l'IFI.

*Chau Duc Phu
Hue (Vietnam), automne 2008*

RESUME

Cette mémoire présente une méthode pour améliorer n'importe quel algorithme de suivi basé sur l'apprentissage. La méthode proposée est divisée en deux phases : une phase de formation et une phase de test. Au cours de la phase de formation, les caractéristiques importantes des trajectoires sont extraites et qui sont ensuite utilisées pour calculer la valeur de fiabilité pour chaque trajectoire. Un algorithme génétique est utilisé pour déterminer l'importance (poids) de chaque caractéristique. Les positions auxquelles les objets sont souvent perdus et retrouvés et le suivi sont regroupées afin de construire l'ensemble des 'lost zones' et 'found zones' de la scène. Grâce à ces zones, on construit un ensemble de triplets de zones : 'In/Out zone' (zone où un objet peut entrer ou sortir de la scène), 'lost zone' et 'found zone'. Chaque triplet de zones est «l'image» d'un chemin que les objets détectés suivent. Grâce à eux, pendant la phase de test, nous pouvons réparer les trajectoires perdues car ces trois zones déterminent les trajectoires plus fiables. Les avantages de notre approche sur l'état de l'art actuel sont que (i) Cette méthode n'est pas dépendante d'une scène prédéfinie, (ii) nous exploitons la sémantique de la scène et (iii) nous avons proposé une méthode pour filtrer les bruits grâce à ses valeurs de fiabilités. L'algorithme a été expérimenté dans le projet Européen CARETAKER (<http://www.ist-caretaker.org>) avec des vidéos capturées dans les stations de métro.

ABSTRACT

This report presents a method for improving any object tracking algorithm based on machine learning. The proposed method is divided into a training phase and a testing phase. During training phase, important trajectory features are extracted which are then used to calculate a confidence value of trajectory. A genetic algorithm is used to determine the importance (weight) of each feature to compute the confidence of object trajectories. The positions at which objects are usually lost and found are clustered in order to construct the set of ‘lost zones’ and ‘found zones’ in the scene. Using these zones, we construct a triplet set of zones i.e. 3 zones: zone ‘In/Out’ (zone where an object can enter or exit the scene), ‘lost zone’ and ‘found zone’. Each zone triplet defines a route that tracked objects undertake. Thanks to these triplets, during the testing phase, we can repair the erroneous trajectories according to which triplet they are most likely to belong to. The advantage of our approach over the existing state of the art approaches is that (i) this method does not depend on a predefined contextual scene, (ii) we exploit the semantic of the scene and (iii) we have proposed a method to filter out noisy trajectories based on their confidence value. The method was experimented within the CARETAKER project (<http://www.ist-caretaker.org>) with videos captured in subway stations.

CHAPITRE I

INTRODUCTION

1.1 Problématique

Aujourd'hui, on peut trouver facilement des applications de la surveillance vidéo partout : au parking, au musée, à l'hôpital, à la gare ... La surveillance vidéo nous aide à observer beaucoup de positions différentes en même temps et à reconnaître vite les événements anormaux qui se passent dans la scène. Cependant, comment un agent de sécurité peut-il analyser en temps réel des dizaines d'écrans de contrôle avec un risque d'erreur minimal? D'autre part, l'observation des écrans dans un temps long est un travail ennuyeux. La réponse de ces deux problèmes tient en deux mots : vidéo intelligente.

La vidéo intelligente est un nouveau domaine qui étudie ce qui se passe dans une vidéo. Ce terme s'exprime un axe de recherche assez large, qui est appliqué dans des domaines différents : robotique, médecine, par exemple. En particulier, beaucoup de recherches, de travaux dans ce domaine sont déjà implantés dans la surveillance vidéo. La reconnaissance des objets, des activités dans la vidéo..., ce sont des sujets qui sont actuellement intéressés par plusieurs chercheurs. Cette mémoire étudie une des applications les plus importantes de la vidéo intelligente, c'est le suivi du mouvement des objets mobiles dans la vidéo.

Il existe des recherches qui focalisent sur ce domaine [1, 3, 16, 18]. Quelques approches sont proposées mais il y a des problèmes ne sont pas encore résolus. Un algorithme peut seulement être appliqué dans quelques scènes fixes. La sémantique de la scène est un élément important qu'on

peut exploiter pour mieux suivre les mouvements, cependant cet élément n'obtient pas encore beaucoup d'intérêts de recherches. Quelques chercheurs font des efforts pour modéliser la scène [11,15], toutefois les scènes considérées sont encore assez simples. Dans le processus du suivi, l'apparition des bruits n'est pas évitable à cause de la complexité de la scène, la relativité de l'algorithme utilisé... alors il est aussi nécessaire de filtrer ces trajectoires avant de commencer la modélisation. Sinon, l'utilisation des bruits pour la construction des scènes peut entraîner les résultats imprécis.

1.2 Motivation et Objectif

Le suivi du mouvement donne toujours les résultats incomplets. Il existe les cas que le système n'arrive pas à suivre le mouvement ou prend des décisions imprécises. Notre travail focalise sur la réparation on-line des trajectoires perdues pendant le processus du suivi.

Dans notre approche, nous utilisons l'apprentissage afin d'extraire automatiquement la sémantique de la scène. Nous proposons également une méthode pour calculer la valeur de fiabilité des trajectoires. Cette valeur est utilisée pour filtrer les bruits avant le processus d'apprentissage. Seules les trajectoires typiques sont utilisées dans la phase de formation et pour modéliser la scène. Pour pouvoir modéliser les scènes complexes ou les scènes où il n'y a pas clairement les routes, l'objet que nous modélisons dans cette recherche, n'est ni un chemin, ni une route mais une zone. Grâce à la modélisation de la scène, on peut détecter et réparer les trajectoires incomplètes. Pour vérifier l'efficacité de l'algorithme, chaque trajectoire est assignée à une valeur de fiabilité. L'expression de cette valeur est une fonction des caractéristiques qui estiment la vérité des trajectoires. Plus une

trajectoire est complète et plus sa valeur de fiabilité est grande. En calculant la somme des valeurs de fiabilité des trajectoires avant et après l'intégration de l'algorithme pour la réparation, on peut évaluer l'effet de l'algorithme.

1.3 Contribution

Le stage obtient quelques résultats qui sont intégrés dans la plateforme de l'équipe. Tout d'abord, on a construit un ensemble des caractéristiques qui estime la vérité d'une trajectoire. Grâce à cet ensemble, on peut déterminer une trajectoire, qui peut être un bruit ou non. On a trouvé une nouvelle approche pour modéliser une scène. Notre méthode est générale, l'objet étudié pour la modélisation n'est pas des chemins et des routes comme les recherches précédentes, mais des zones. On arrive aussi à détecter l'apparition anormale des objets dans la scène et à réparer les trajectoires qui sont perdues durant le suivi.

1.4 Environnement de Stage

Le stage se déroule dans l'équipe PULSAR (**P**erception **U**nderstanding **L**earning **S**ystem **A**ctivity **R**ecognition, <http://www-sop.inria.fr/pulsar>) de l'INRIA Sophia Antipolis. PULSAR se focalise sur la reconnaissance des activités. Plus précisément, PULSAR s'intéresse à l'interprétation sémantique et en temps réel des scènes dynamiques observées par des capteurs. Ainsi, PULSAR étudie des activités spatio-temporelles à long terme effectuées par des êtres humains, des animaux ou des véhicules. L'objectif de ce stage est d'améliorer l'algorithme de suivi des mouvements dans des vidéos qui a été développé dans la plateforme VSUP de l'équipe. L'algorithme proposé a été expérimenté sur les vidéos

du projet CARETAKER (www.ist-caretaker.org) qui surveillent les stations de métro.

Le reste du document est organisé de la manière suivante. Le chapitre 2 introduit d'abord un état de l'art des travaux concernés. Dans le chapitre 3, une description du système global est présentée. Le chapitre 4 décrit le contenu principal de ce rapport. Les étapes de l'algorithme proposé qui comprennent la phase d'apprentissage et la phase de test, seront présentées et expliquées. Le chapitre 5 fournit les résultats de la phase d'expérimentation et de la validation qui permettent d'illustrer l'intérêt de la méthode présentée. Le dernier chapitre présentera la conclusion, ainsi que les propositions pour améliorer notre algorithme pour mieux réparer les trajectoires.

CHAPITRE II

ETAT DE L'ART

2.1 Détection et Suivi du Mouvement

2.1.1 Détection du mouvement

Pour pouvoir suivre les mouvements dans la vidéo, la première étape est de détecter les mouvements. Cette étape joue un rôle très important dans le système de vidéo surveillance car le résultat de cette étape va influencer toutes les étapes suivantes. Dans cette partie, on va étudier deux méthodes pour détecter le mouvement. L'idée principale de ces deux méthodes est de comparer la différence pixel par pixel entre deux images. Les régions qui ont la grande différence sont les régions de l'objet mobile.

Le premier est basé sur le calcul d'un gradient temporel : une mesure de vraisemblance du mouvement est fournie par le changement instantané calculé entre deux trames consécutives [4, 9, 17]. Ces méthodes sont naturellement adaptables aux changements d'environnements, mais sont aussi dépendantes de la vitesse et de la taille des objets en mouvement. Ce défaut peut être minimisé en utilisant des combinaisons de filtres spatio-temporelles mais au prix d'une complexité croissante.

Le second est basé sur des techniques de différence au fond [7, 12, 14] qui utilise une image de référence (le fond), représentant les éléments stationnaires de la scène (voir la figure 2.1). Ici la mesure de vraisemblance du mouvement est la différence entre la trame courante et le fond. Ces méthodes sont moins dépendantes de la vitesse et de la taille des objets. Cependant, l'adaptation aux environnements dynamiques est une tâche bien

plus ardue, pénalisant la détection des mouvements de faible amplitude (des objets petits, très lents ou très peu contrastés).

Dans [10] l'auteur a présenté plusieurs méthodes pour créer l'image de fond :

- Calculer de moyenne de N premières images.
- Choisir l'image qui a peu de changement de scène ou qu'il n'y a pas d'objet mobile pour l'image de fond.
- Choisir la première image dans la séquence de vidéo.

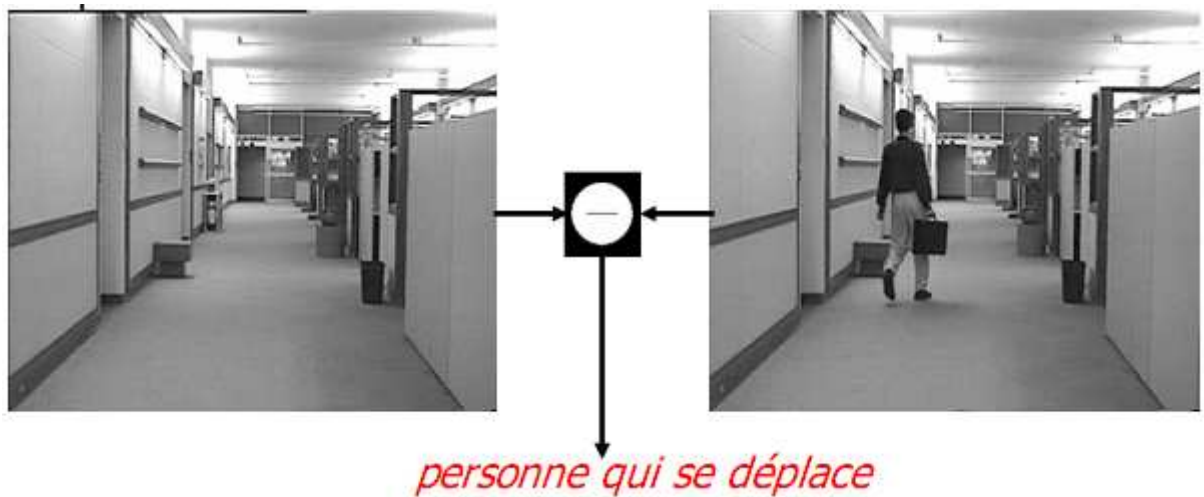


Fig 2.1 Illustration de la différence d'images pour détecter le mobile.
Source [2]

2.1.2 Suivi du Mouvement

Dans le suivi du mouvement, le filtre de Kalman est une méthode de prédiction intéressée par plusieurs chercheurs [1, 2, 6, 18]. Un filtre de Kalman est essentiellement un ensemble d'équations récursives, elles sont utilisées ensemble pour aider à modéliser et évaluer très précisément le mouvement d'un système dynamique linéaire. Le filtre de Kalman est appelé 'discret' parce que l'algorithme est discret dans le temps. En utilisant une multitude de différents vecteurs et des matrices, le filtre de

Kalman est capable de construire une représentation interne du système et grâce à cette représentation, on peut estimer l'état futur du système. Les filtres sont modélisés en utilisant une chaîne de Markov. Dans le filtrage de Kalman, les équations décrivent l'état du système à temps successifs, grâce à ces équations on peut prédire l'état futur en utilisant l'état actuel sans dépendre des états antérieurs dans les calculs.

Dans le processus du suivi, le filtre Kalman 'regarde' un objet quand il se déplace, c'est à dire il prend les informations sur l'état de l'objet à ce moment-là. Puis il utilise ces informations pour prédire où se trouve l'objet dans la trame prochaine. La position de l'objet est corrigée ensuite en considérant la prédiction et aussi l'observation (voir la figure 2.2).

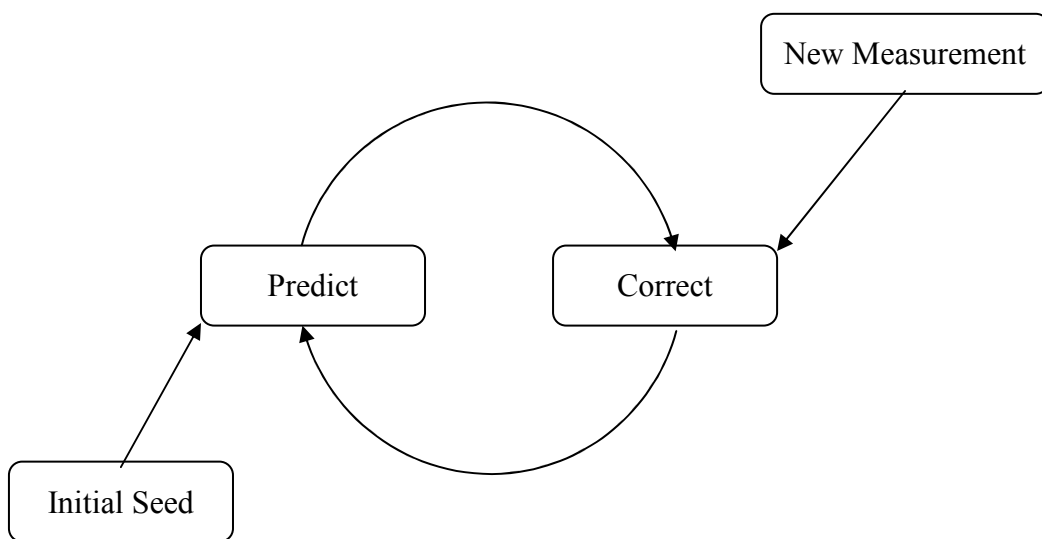


Figure 2.2 Le cycle récursif de Kalman
(Source [13])

2.2 Amélioration de l'Algorithme de Suivi

Il existe peu de recherches visant à la réparation et l'amélioration de la piste d'objets dans des vidéos [1, 16, 18, 19]. Dans [18] les auteurs proposent d'utiliser de filtre de Kalman combiné avec des techniques

d'optimisation pour les données d'association afin de filtrer et de mieux gérer les occlusions et les mouvements non-linéaire. L'ensemble des caractéristiques de la trajectoire courante est obtenu en optimisant la somme des distances de Mahalanobis entre les prévisions et les mesures correspondantes. Dans [16], une méthode pour suivre les objets en mouvement est présentée, en utilisant des 'particle filters' pour estimer l'état des objets, en se basant sur la probabilité d'association des filtres de données. Les auteurs dans [19] proposent d'utiliser un modèle pour gérer les caractéristiques suivies. Chaque état de caractéristique de trajectoire est assigné à une valeur de fiabilité. Cette valeur aide à distinguer entre l'occlusion et la disparition d'un objet. Toutefois, la valeur de fiabilité est calculée par des caractéristiques simples (nombre de fois où un objet est correctement suivi) et ne peut pas détecter les bruits. Tous ces travaux énumérés ci-dessus ont obtenu des premiers résultats mais dans les cas précédents, on ne considère pas encore des situations d'occlusions complexes et longues.

Pour résoudre le problème d'occlusion, des recherches [11, 15] ont focalisé sur la modélisation de la scène afin d'améliorer l'algorithme de suivi. L'idée principale consiste à exploiter des informations intéressantes de la scène telles que: les positions, les directions des chemins, les zones sensibles dans la scène où le système peut perdre la piste des objets avec une haute probabilité, les zones où des objets mobiles apparaissent et disparaissent souvent... Ces éléments peuvent aider le système à prédire mieux la trajectoire des objets. Il existe deux méthodes pour modéliser une scène, soit en utilisant les techniques d'apprentissage, soit à la main. Avec l'apprentissage automatique, le coût de la modélisation est faible, mais l'algorithme de modélisation doit assurer la qualité et la précision du

modèle construit. Par exemple, les auteurs dans [15] ont présenté une méthode pour modéliser les chemins de la scène basée sur les trajectoires détectées. Un graphe est construit afin de représenter la structure de routes modélisées dans la scène. La construction des chemins est effectuée automatiquement par les techniques d'apprentissage non supervisé, basées sur le clustering des trajectoires. Toutefois, cette méthode ne peut être appliquée que dans des scènes simples où il existe clairement des routes. Les critères pour évaluer les bruits se basent dans la plupart des cas sur la durée des trajectoires. Fernyhough et al [11] utilisent le même modèle pour apprendre automatiquement les chemins des objets grâce à l'accumulation des traces des objets suivis. Cette méthode est non supervisée et d'auto-initialisation. Toutefois, cette méthode exige les trajectoires complètes, elle ne gère pas les occlusions et les résultats sont dépendants de la forme et de la taille des objets, car ils sont détectés sur le plan 2D.

2.3 Conclusion

Dans ce chapitre, on a présenté les travaux qui concernent le suivi et l'amélioration du suivi du mouvement dans la vidéo. Dans notre approche, nous voudrions exploiter la sémantique de la scène pour réparer le suivi des objets mobiles. Les recherches antérieures pour cette approche ne donnent pas beaucoup de résultats clairs, il nous manque une méthode générale pour modéliser tous les types de scène dans le but de suivi du mouvement. Dans ce rapport, nous allons présenter une méthode pour réparer les trajectoires perdues suivi à l'aide de la modélisation de la scène. Nous visions à une méthode générale qui peut modéliser plusieurs types de scène et qui n'exige pas beaucoup de connaissances préalables de la scène.

CHAPITRE III

PRESENTATION DU SYSTEME

D'INTERPRETATION DE VIDEO

Le module Vision qui est le noyau du système, comprend trois tâches (voir la figure 3.1). Tout d'abord, un détecteur du mouvement (Motion Detector) et un 'frame to frame tracker' créent un graphe d'objets mobiles pour chaque caméra calibrée. Deuxièmement, un mécanisme de combinaison est réalisé pour combiner les graphes calculés par chaque caméra en un graphe global. Troisièmement, ce graphe global est utilisé pour le suivi des individus, des groupes de personnes et les foules dans la scène (pendant centaines de trames).

Ici, nous utilisons les modèles de 3D de la scène, chaque modèle est défini pour chaque caméra, considéré comme les connaissances préalables du contexte de la scène observée. Plus clairement, nous définissons les positions 3D et les dimensions 3D des objets statiques dans la scène (par exemple un banc, un distributeur automatique de billets) et les zones d'intérêt (par exemple une zone d'entrée). Les attributs sémantiques (par exemple fragile) peuvent être associés aux objets ou zones d'intérêt pour être utilisés dans le processus de reconnaissance des comportements.

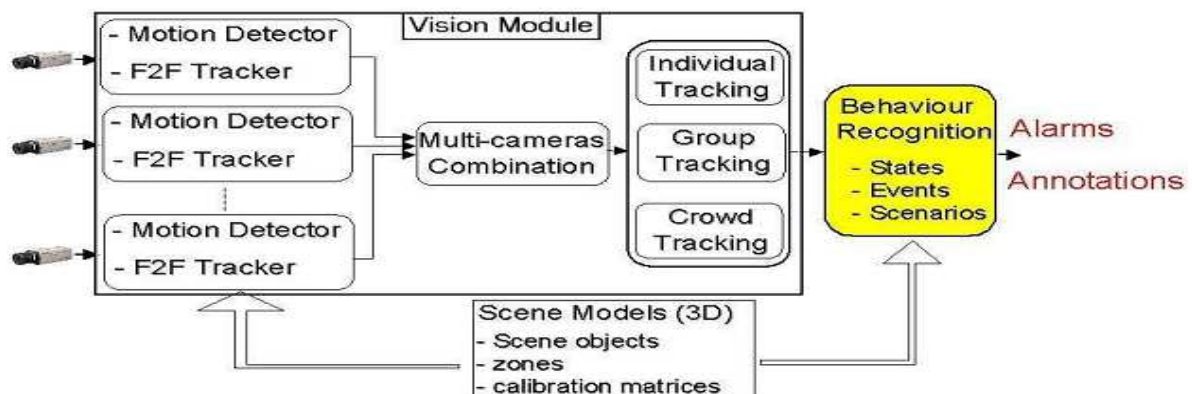


Fig 3.1 Système d'interprétation de vidéo
Source [8]

3.1 Détection du Mouvement et Suivi Trame par Trame

L'objectif du Détecteur du mouvement est de détecter les régions de la scène en mouvement à chaque trame et de les classier dans une liste d'objets mobiles avec des étiquettes correspondantes à leur type, 'PERSON' par exemple. Cette classification est basée sur leur taille 3D. Cette tâche peut être divisée en trois sous-tâches: détection d'objets mobiles, extraction des caractéristiques et classification des objets mobiles. Une liste des objets mobiles est obtenue à chaque trame. Chaque objet mobile est décrit par des paramètres numériques 3D (centre de gravité, position, hauteur, largeur...) et est classifié à une classe sémantique (PERSON, OCCLUDED PERSON, GROUP, CROWD, METRO TRAIN, SCENE OBJECT, NOISE or UNKNOWN).

L'objectif de 'frame to frame tracker' (F2F Tracker) est de lier trame par trame la liste des objets mobiles calculés par le détecteur du mouvement. La sortie de cette tâche est un graphe d'objets mobiles. Ce graphe fournit toutes les trajectoires possibles qu'un objet mobile peut suivre. Le lien d'un objet mobile entre les trames est considéré sur trois critères: la similitude entre leurs classes sémantiques, leurs distance 2D (dans l'image) et leur distance 3D (dans le monde réel).

3.2 Combinaison de Multiples Caméras

Afin d'exploiter les avantages des caméras calibrés surveillant la même scène (la vision de ces caméras se superposent), nous combinons tous les graphes des objets mobiles calculés par le F2F Tracker pour chaque caméra à un graphe global qui est appelé Graphe Combiné (Combined Graph). En conséquence, les caractéristiques (les positions 3D

et les dimensions) des objets mobiles calculées dans le graphe combiné donnent une meilleure estimation des positions et des dimensions des objets réels dans la scène.

Pour calculer le graphe global, nous combinons à chaque trame les objets mobiles nouveaux détectés par deux caméras en utilisant une matrice combinée et un ensemble de règles (voir l'illustration sur la figure 3.2). La matrice combinée donne les correspondances entre les objets mobiles détectés par deux caméras en utilisant les critères de position 3D et de taille 3D. Dans le cas d'aucune ambiguïté entre les objets mobiles détectés par les deux caméras, on fusionne les objets mobiles en calculant la valeur moyenne des caractéristiques 3D. En cas d'ambiguïtés, un ensemble de règles est utilisé pour sélectionner ou éliminer l'objet mobile détecté par un des deux caméras.

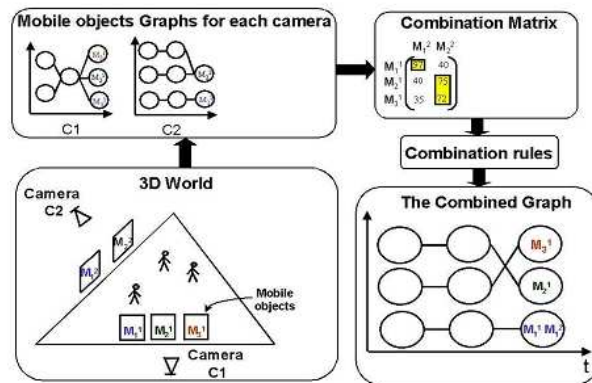


Fig 3.2 : This figure illustrates the multiple cameras combination process. Three persons are evolving in the scene. Camera C1 detects three mobile objects whereas camera C2 detects only two mobile objects. The combination matrix enables to determine (a) a high correspondence between the mobile object M11 of C1 and the mobile object M12 of C2; these two mobile objects are fused together in the combined graph, and (b) an ambiguous correspondence between the two mobile objects M21 and M31 of C1 and the mobile object M22 of C2; the two mobile objects M21 and M31 detected by C1 are selected in the combined graph.

Source [8]

3.3 Suivi à long terme des individus, groupe de personnes et la foule

L'objectif ici est de suivre sur une longue période de temps les individus, les groupes de personnes ou de la foule afin de permettre aux scénarios concernés à ces trois différents types d'acteurs d'être reconnu. Par exemple, lorsque l'on veut détecter un groupe de personnes (au moins deux personnes) qui est en train de bloquer une zone de sortie, c'est mieux logique de raisonner avec le 'Group Tracker', car il fournit plus précisément la location 3D du groupe de personnes dans la scène.

Le 'Individual Tracker' réalise le suivi de chaque personne pendant que le 'group tracker' réalise le suivi de toutes les personnes dans le même groupe. Tous les deux 'trackers' effectuent les analyses temporelles. Le 'Individual tracker' calcule et sélectionne les trajectoires des objets mobiles qui sont correspondantes à une personne réelle à l'aide du modèle explicite 'trajectoire d'une personne'. De la même façon, le Groupe Tracker calcule et sélectionne les trajectoires des objets mobiles qui peuvent correspondre aux personnes dans un groupe réel grâce à un modèle explicite 'trajectoires des personnes dans un groupe'.

'Individual Trackers' et 'Group Trackers' sont exécutés en parallèle. Lorsque la densité des objets mobiles détectés devient trop élevé (en général, si les objets mobiles superposent plus de 2/3 de l'image), nous arrêtons ces deux 'trackers' parce que dans un tel cas, ils ne peuvent pas donner des résultats fiables. En ce moment-là, on déclenche 'Crow Tracker' qui est en fait l'extension des modèles de trajectoires dans le 'Group Tracker' qui permet une grande densité des personnes détectées appartenant au même groupe.

CHAPITRE IV

REPARATION DES TRAJECTOIRES

PAR CLUSTERING DE POINTS

Ce chapitre présente le contenu principal du rapport. L'objectif de notre travail focalise sur l'amélioration de l'algorithme de suivi du système. Notre approche est de réparer les trajectoires que le système perd le suivi grâce au processus de l'apprentissage.

4.1 Notre Approche

Pour résoudre les problèmes posés par les travaux antérieurs, dans notre approche, nous proposons à exploiter la sémantique de la scène grâce à l'apprentissage supervisé. La phase de modélisation de la scène qui ne doit pas exiger beaucoup de connaissances préalables de la scène et doit être réalisée automatique. Nous cherchons une méthode qui peut modéliser plusieurs types de scènes. Pour cela, au lieu de construire et représenter précisément les routes ou les chemins dans la scène, nous proposons une méthode plus générale. La scène ici est modélisée par les triplets de zones. Chaque triplet de zones peut représenter plusieurs routes différentes de la scène. Le filtre des trajectoires est aussi considéré pour éliminer les bruits. C'est pour cela que seules les trajectoires typiques sont utilisées pour apprendre la scène. Le résultat de la phase de modélisation est donc meilleur. Lorsque la modélisation de la scène est finie, on peut détecter la perte du suivi et réparer les trajectoires incomplètes.

4.2 Caractéristiques d'une trajectoire

Ici, nous utilisons les trajectoires obtenues par un algorithme pour le suivi du mouvement comme les entrées [3]. Nous voudrions filtrer les bruits, seules les trajectoires complètes sont utilisées pour la phase d'apprentissage. Pour cela, nous définissons 'la valeur de fiabilité' d'une trajectoire. Plus la valeur de fiabilité de la trajectoire est grande et plus cette trajectoire est complète. Cette valeur est calculée grâce aux caractéristiques extraites des trajectoires. Nous proposons de définir 9 caractéristiques:

1. Caractéristique 'entry zone' est activée quand l'objet entre dans la scène par une zone d'entrée, zone autour de la porte par exemple.
2. Caractéristique 'exit zone' est activée quand l'objet sort de la scène par une zone de sortie.

Ici, les zones d'entrées et zones de sortie sont déjà déterminées à la main avant le processus de l'apprentissage. Ce sont les connaissances préalables.

3. Time: la durée de la trajectoire.
4. Length: la longueur spatiale de la trajectoire.
5. Nombre de fois que l'objet mobile est classifié dans la classe 'Personne'. Cette classification se base sur la dimension 3D et un modèle d'objet 'Personne' 3D prédéfini. Ce nombre est directement proportionnel à sa valeur de fiabilité.
6. Nombre de fois que le système perd le suivi de cette trajectoire.
7. Nombre de voisins d'objet mobile à quatre instants spéciaux. Ici, nous comptons le nombre d'objets mobiles proches de l'objet considéré quand il est (1) détecté pour la première fois, (2) perdu, (3) trouvé (si déjà perdu) et (4) lorsque la trajectoire se termine. Cette caractéristique est utilisée pour

évaluer le potentiel d'erreur lors de la détection d'un objet. Plus ce nombre est grand, plus la fiabilité de la trajectoire est petite.

8. Nombre de fois que l'objet mobile change de sa taille. Ce changement se base sur un seuil de variation dimensionnel prédéfinie. Le changement de taille d'un objet mobile qui est trop grande, pénalisera la valeur de fiabilité de sa trajectoire.

9. Nombre de fois que l'objet mobile change de directions spatiales. Dans les stations de métro, les personnes se déplacent normalement tout droit pour aller à ses arrivées. Lorsque cette caractéristique est élevée, la valeur de fiabilité de trajectoire est faible.

Ces neuf caractéristiques définies ci-dessus sont utilisées pour caractériser la fiabilité d'une trajectoire détectée. Comme ces neuf caractéristiques sont multimodales, l'unité de 'time' est temporelle, l'unité de 'length' est spatiale, celle de 5 à 9 est le nombre de fois, une phase de normalisation pour ces caractéristiques est nécessaire. La valeur de fiabilité d'une trajectoire est seulement calculée par les valeurs des caractéristiques après avoir normalisé. Les valeurs des caractéristiques 5 et 8 sont normalisées par la durée de la trajectoire correspondante. Les caractéristiques 1 et 2 sont booléens: 0 (non activé) ou 1 (activé). La valeur des autres caractéristiques (3, 4, 6, 7 et 9) sont normalisées comme suit:

$$f_i = \frac{FV_i - \mu_i}{\sigma_i} . \quad (1)$$

où,

f_i : la nouvelle valeur de caractéristique i après normalisation

FV_i : la valeur de caractéristique i où $i = \{3,4,6,7,9\}$

μ_i : la valeur moyenne de caractéristique i pour toutes les trajectoires traitées pendant la phase d'apprentissage

σ_i : la variance de caractéristique i pour toutes les trajectoires traitées pendant la phase d'apprentissage

La valeur de fiabilité d'une trajectoire est calculée par la formule suivante:

$$CV = \sum_{i=1}^5 (w_i * f_i) + \sum_{i=6}^9 (w_i * (1 - f_i)) \quad (2)$$

où,

CV (*confidence value*) : la valeur de fiabilité de trajectoire considérée

w_i (*weight*) : le poids (importance) de la caractéristique i (voir la section 4.2 pour détails)

f_i : la valeur de caractéristique i , après normalisation

Les 5 premières caractéristiques sont les caractéristiques directement proportionnelles à la valeur de fiabilité, les dernières caractéristiques sont inversement proportionnelles à la valeur de fiabilité.

4.3 Déterminer les Poids des Caractéristiques par l'Algorithme Génétique et l'Apprentissage

On a au total neuf caractéristiques pour qualifier la fiabilité d'une trajectoire. Clairement, les caractéristiques différentes ont des niveaux d'influences différents sur la trajectoire. On qualifie l'influence d'une caractéristique par la notion 'poids'. Le poids d'une caractéristique représente l'importance de la caractéristique dans sa contribution pour la valeur de fiabilité de la trajectoire. Le problème posé est de déterminer les poids des neuf caractéristiques de la trajectoire. Dans notre approche, nous

utilisons l'apprentissage supervisé combiné avec l'algorithme génétique pour résoudre ce problème.

Tout d'abord, on prend les caractéristiques de 300 premières trajectoires dans une vidéo. A chaque trajectoire, on assigne la valeur 'Ground Truth' après l'avoir observée. La notion 'Ground Truth' indique la valeur globale de fiabilité d'une trajectoire et est assignée à la main. Cette valeur est dans l'intervalle $[0,1]$. Nous définissons quatre niveaux pour classer les trajectoires dans quatre classes :

- trajectoires complètes : $\text{Ground Truth} \geq 0.8$
- trajectoires incomplètes : ne commencent pas ou ne se terminent pas dans une zone d'entrée/sortie, $0.5 \leq \text{Ground Truth} < 0.8$
- trajectoires peu fiables : ne commencent pas et ne se terminent pas dans une zone d'entrée/sortie, $0.2 \leq \text{Ground Truth} < 0.5$
- bruits : ne sont pas la trajectoire d'une personne, $\text{Ground Truth} < 0.2$

Le problème réside dans l'apprentissage des poids des caractéristiques pour optimiser la correspondance entre la valeur de 'Ground Truth' et celle de fiabilité de la même trajectoire calculée par les poids. Il existe des méthodes qui sont utilisées pour apprendre les poids, recuit simulé par exemple. Dans notre approche, nous choisissons l'algorithme génétique parce qu'il économise le coût temporel (on a totalement jusqu'à neuf caractéristiques) et il est relativement efficace.

4.3.1 Mutation et Cross-Over pour l'Algorithme Génétique

Dans l'algorithme génétique, il y a trois notions importantes qu'il faut aborder. Ce sont 'individu' et deux opérateurs 'Mutation', 'Cross-Over'. En général, un individu est une proposition d'une solution pour le problème considéré. Le terme 'Mutation' exprime une transformation sur

lui-même, alors que celui ‘Cross-Over’ s’explique par un échange des valeurs entre les deux individus. A chaque contexte, ces trois termes représentent des notions sémantiques différentes.

Pour ce problème, on appelle un individu un ensemble de neuf valeurs représentant les poids de neuf caractéristiques de la trajectoire apprise. Correspondant à chaque individu et chaque trajectoire, on peut calculer une valeur de fiabilité.

Fitness de chaque individu représente la somme des différences entre la valeur de ‘Ground Truth’ de chaque trajectoire et la valeur de fiabilité calculée par cet individu (de même trajectoire). Plus clairement, la valeur de fitness est calculée par la formule suivante :

$$\phi = \sum_{i=1}^{300} |GT(i) - CV(i)| \quad (3)$$

où,

ϕ : fitness de l’individu considéré

$GT(i)$: Valeur de ‘Ground Truth’ de la trajectoire i

$CV(i)$: Valeur de fiabilité de la trajectoire i calculée par les poids (individu) considéré

Plus le fitness de l’individu est petite, plus l’individu est bon. Afin d’explorer les distributions différentes des individus, nous définissons deux opérateurs ‘Mutation’ et ‘Cross-Over’ comme suit. La mutation se passe avec une probabilité de 30%. Dès qu’un individu est sélectionné pour la mutation, une position dans l’ensemble de poids des caractéristiques est choisie au hasard. Toutes les valeurs de cet individu à partir de cette position jusqu’à la fin seront modifiées par de nouvelles valeurs aléatoires

(voir la figure 4.1).

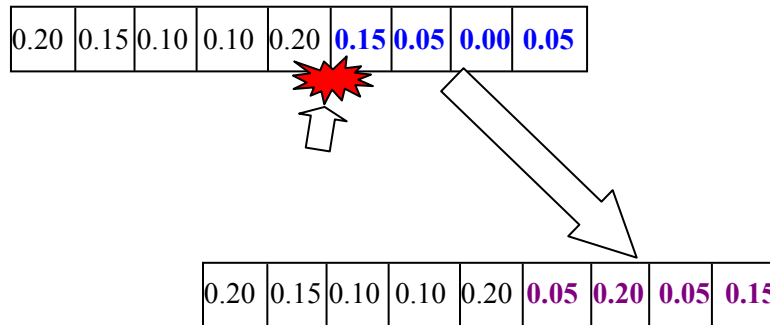


Fig 4.1. Illustration d'opérateur Mutation

La transformation Cross-Over est effectuée pour chaque individu avec une probabilité de 80%. Deux individus sont sélectionnés pour échanger leur poids. Une position de caractéristique est choisie au hasard pour tous les deux individus. Toutes les valeurs de poids de ces deux individus de cette position jusqu'à la fin seront échangées (voir la figure 4.2).

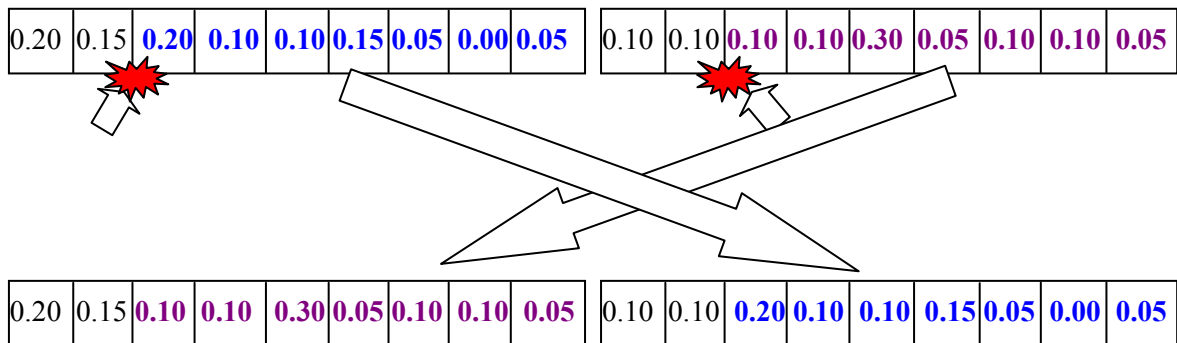


Fig 4.2. Illustration d'opérateur 'Cross-Over'

Après les transformations des opérateurs telles que mutation ou cross-over, la somme des poids d'un individu peut être plus ou moins égale à 1. Une phase de normalisation est nécessaire pour assurer toujours cette somme égale 1. Pour faire cela, chaque poids sera remplacé par son pourcentage par rapport aux nouvelles valeurs de poids.

$$P_i = \frac{P_{GN_i}}{\sum_{i=1}^9 P_{GN_i}} \quad (4)$$

où :

P_i : le nouveau poids de caractéristique i après avoir normalisé

P_{GN_i} : le poids de caractéristique i obtenu après avoir transformé Mutation ou Cross-Over

4.3.2 Algorithme génétique

On crée la première génération qui se compose de 5000 individus. Pour chaque génération, on réalise les transformations cross-over et mutation en ordre avec les probabilités déjà définies pour créer les nouvelles générations. A chaque fois qu'une nouvelle génération est créée, le meilleur individu de cette génération sera comparé au meilleur individu obtenu dans les générations précédentes. L'individu qui a le fitness le moins important, sera pris pour continuer l'algorithme (voir la figure 4.3). Ce processus est réalisé jusqu'au fitness de meilleur individu ou le nombre de génération créée inférieurs à un seuil donné. Alors, cet individu est choisi la solution du problème.

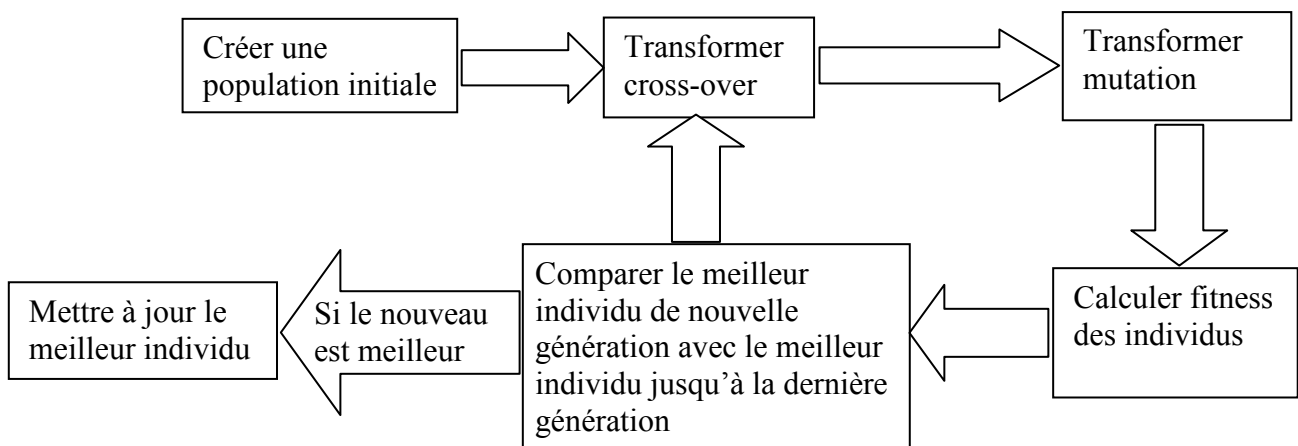


Fig 4.3. Les pas de réalisation d'algorithme génétique

4.4 Types de Zones Utilisées dans la Scène

Dans notre recherche, nous voudrions exploiter la sémantique de la scène. D'après ce qui a été présenté précédemment, il y a des recherches qui focalisent sur la modélisation de la scène [11,15]. La plupart de ces recherches utilisent des techniques différentes pour trouver les routes et les chemins dans la scène. Ces éléments seront utilisés pour détecter les activités anormales... Cependant, il y a des scènes qui n'existent pas clairement les routes, la station de métro par exemple. Dans ce cas là, l'approche ci-dessus ne marche pas. Pour résoudre ce problème, au lieu de modéliser les chemins, nous proposons de modéliser les zones spéciales de la scène. Voilà les définitions des types de zones que nous nous intéressons :

- 'entry zone' (zone d'entrée) : zone où l'objet peut entrer dans la scène.
- 'exit zone' (zone de sortie) : zone où l'objet peut quitter de la scène.
- IO zone (zone d'entrée/sortie): zone où l'objet peut entrer et également quitter de la scène.
- 'lost zone' : zone qui n'est pas un 'exit zone' ou un 'IO zone', où on perd souvent le suivi de l'objet.
- 'found zone' : zone, qui n'est pas une 'entry zone' ou une 'IO zone', où on retrouve souvent le suivi de l'objet.
- 'lost-found zone' : zone a tous les deux caractéristiques 'lost' et 'found'.

Dans les deux figures 4.4 et 4.5, on peut voir deux illustrations de la description des différents types de zone.

```

<Zone ident="9" name = "ZoneIOLeftTop" plane_name =
"ground">
  <Properties_list>
    <Property name = "In_out_zone:Entry"/>
  </Properties_list>
  <Outline_list>
    <Point x="-830.0" y="-350.0" z = "0"/>
    <Point x="-300.0" y="-350.0" z = "0"/>
    <Point x="-300.0" y="-100.0" z = "0"/>
    <Point x="-830.0" y="-100.0" z = "0"/>
  </Outline_list>
</Zone>

```

Fig 4.4 Description d'une 'entry zone'

```

<Zone ident = "2" name = "ZoneLearning0" plane_name =
"ground">
  <Properties_list>
    <Property name = "Lost_found_zone:Yes"/>
  </Properties_list>
  <Outline_list>
    <Point x="-2046.000000" y = "12.000000" z="0" />
    <Point x="-2046.000000" y = "778.000000" z="0" />
    <Point x="-1402.000000" y = "778.000000" z="0" />
    <Point x="-1402.000000" y = "12.000000" z="0" />
  </Outline_list>
</Zone>

```

Fig 4.5 Description d'une 'lost-found zone'

Les balises <Properties_list> </Properties_list> sont utilisées pour décrire les caractéristiques de la zone, lost ou found zone par exemple.

Les balises <Outline_list> </Outline_list> sont utilisées pour décrire les bords de la zone.

4.5 Apprentissage des Zones

D'après ce qui a été présenté dans la section précédente, on définit plusieurs de types de zones différents. Chaque type a une méthode de construction spécifique. Les 'entry zone', 'exit zone' et 'IO zone' sont construits manuellement. Par contre, les 'lost zone', 'found zone' et 'lost-found zone' sont construits automatiquement en appliquant le processus de

l'apprentissage. Pour déterminer ces zones, on lance le processus de suivi et on récupère les données. Les coordonnées où le système perd ou retrouve le suivi des personnes, sont marquées. Les positions où on perd le suivi sont groupées en 'lost zone'. Par contre les positions où on retrouve le suivi, sont groupées en 'found zone'. Un processus de clustering basé sur les positions 3D des personnes dans la scène est réalisé. Dans notre approche, nous choisissons l'algorithme Kmean pour le clustering.

Les résultats du processus 'clustering' des 'lost zones' sont affichés dans la figure 4.6.

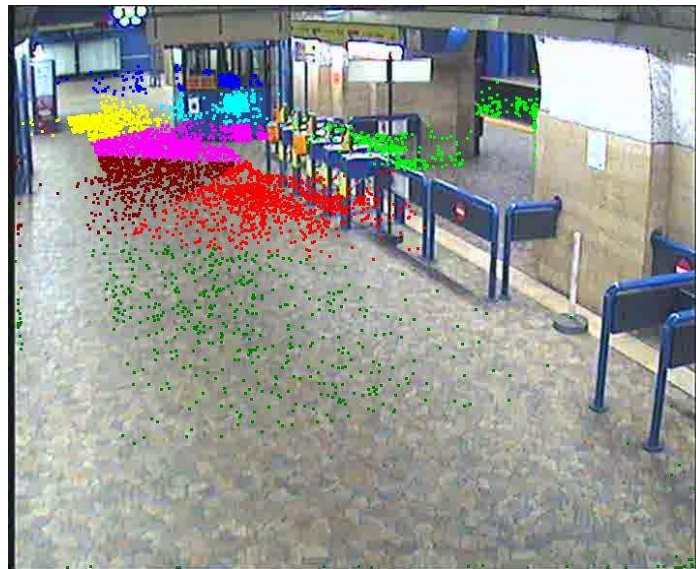


Fig 4.6. Résultat de clustering des 8 'lost zones'.
Chaque couleur représente chaque cluster.

Après le 'clustering', on va déterminer les bords de chaque cluster et utiliser les syntaxes définies (voir les illustrations dans les figures 4.4 et 4.5) pour définir les zones trouvées.

Dans la figure 4.7, les bords des 'lost zones' sont colorés en jaune, les bords rouges représentent les 'entry zones', 'exit zones' et 'IO zones'.



Fig 4.7 Représentation des clusters par les polygones.
Les polygones jaunes représentent les bords des 'lost zones' qui
sont le résultat de l'apprentissage

4.6 Calcul des Triplets de Zones

Un triplet de zones est une structure pour aider le système à réparer les trajectoires perdues le suivi. Le principe du processus de réparation des trajectoires est d'utiliser les trajectoires complètes pour réparer les trajectoires similaires mais incomplètes. Les triplets de zones sont construits pour représenter les chemins dans la scène qui sont créés par les trajectoires complètes. Une trajectoire complète est une trajectoire dont la valeur de fiabilité est supérieure à un seuil donné, et qui commence à une 'start zone' ou 'IO zone'.

Un triplet de zones est un ensemble de 5 valeurs (start Tzone, lost Tzone, found Tzone, temps minimal, temps maximal). La 'start Tzone' est une 'entry zone' ou 'IO zone' où les trajectoires commencent. La 'lost Tzone' est la première 'lost zone' ou la première 'lost-found zone' où les trajectoires complètes passent. La 'found Tzone' est la première 'found zone' ou 'lost-found zone' où les trajectoires complètes passent juste après

avoir passé la 'lost Tzone'. La 'Found Tzone' doit seulement être cherché quand l'objet a quitté de son 'lost Tzone' (voir la figure 4.8). Le 'temps minimal' est assigné à la différence entre l'instant où l'objet entre dans la 'found Tzone' et celui où l'objet quitte de 'lost Tzone'. Le 'temps maximal' est égal à la différence entre l'instant où l'objet quitte de la 'found Tzone', et celui où l'objet entre dans la 'lost Tzone'.

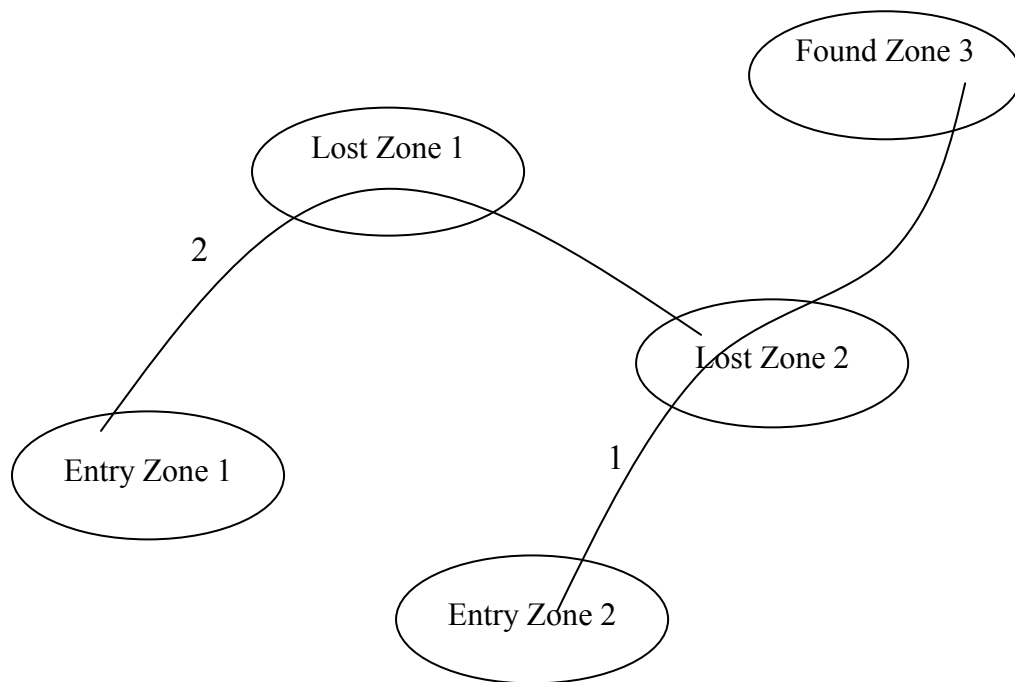


Fig 4.8 Illustration de création d'un triplet de zones.

Trajectoire 1 qui est une trajectoire complète, passe successivement par trois zones 'entry', 'lost' et 'found' zone. Alors on crée un triplet de zone <entry zone 2, lost zone 2, found zone 3, Mi1, Ma1>. Par contre la troisième zone où la trajectoire 2 passe, n'est pas une 'found' zone, on ne peut pas créer un triplet de zone avec cette trajectoire.

Seules les trajectoires complètes, qui passent successivement le 'start Tzone', 'lost Tzone' et 'found Tzone', sont utilisées pour construire un triplet de zones. Tous les triplets de zones sont classés par le nombre des trajectoires passant par eux-mêmes.

Un triplet de zones qui possède un nombre de trajectoires important, sera classé avec rang de priorité élevé. Si un triplet de zones est passé par

plusieurs trajectoires, le temps minimum et le temps maximum sont calculés comme la valeur moyenne de temps minimum et de temps maximum de ces trajectoires. La section suivante présentera comment les triplets seront utilisés pour réparer les trajectoires perdues durant le suivi.

4.7 Réparer les Trajectoires Perdues

La mission du processus de la réparation des trajectoires est de détecter la perte du suivi des trajectoires, de chercher et de fusionner les parties perdues de ces trajectoires. Les trajectoires qui sont fusionnées pour la réparation, doivent assurer la compatibilité de la direction. Cette compatibilité est traînée grâce à l'apprentissage des trajectoires complètes. A la fin de la section précédente, on a obtenu des triplets de zones. Quand le système détecte un objet qui apparaît dans une location anormale (qui n'est pas un 'entry zone' ou un 'IO zone'), on va vérifier s'il apparaît dans une 'found zone' ou dans une 'lost-found' zone ZF. Si c'est vrai, on va parcourir dans tous les triplets définis pour chercher le triplet avec 'found Tzone' ZF, $\langle ZS, ZL, ZF, Mi, Ma \rangle$ par exemple. S'il y a plusieurs triplets trouvés, le système va choisir le triplet dont la priorité est la plus haute. Le système va chercher ensuite la trajectoire perdue qui commence à ZS et perd à ZL ; l'intervalle temporelle entre l'instant où l'objet est perdu le suivi jusqu'à l'instant actuel est supérieur à Mi et inférieur à Ma. Une fois qu'une trajectoire est trouvée, le système va fusionner cette trajectoire avec l'objet qui vient d'être détecté.

La figure 4.9 est une illustration de l'algorithme de réparation. La personne P est détectée dans la 'found zone 3'. Il y a deux triplets $\langle \text{entry zone 1, lost zone 1, found zone 3, } Mi1, Ma1 \rangle$ et $\langle \text{entry zone 2, lost zone 2, found zone 3, } Mi2, Ma2 \rangle$ dans la base de données. On a deux trajectoires 1

et 2 qui ont commencé respectivement à 'entry zone 1' et 'entry zone 2' et qui ont été perdues le suivi à 'lost zone 1' et 'lost zone 2'. On suppose que la priorité du deuxième triplet est plus importante que la première. Alors le système va choisir la deuxième trajectoire qui va être la trajectoire perdue de la personne P.

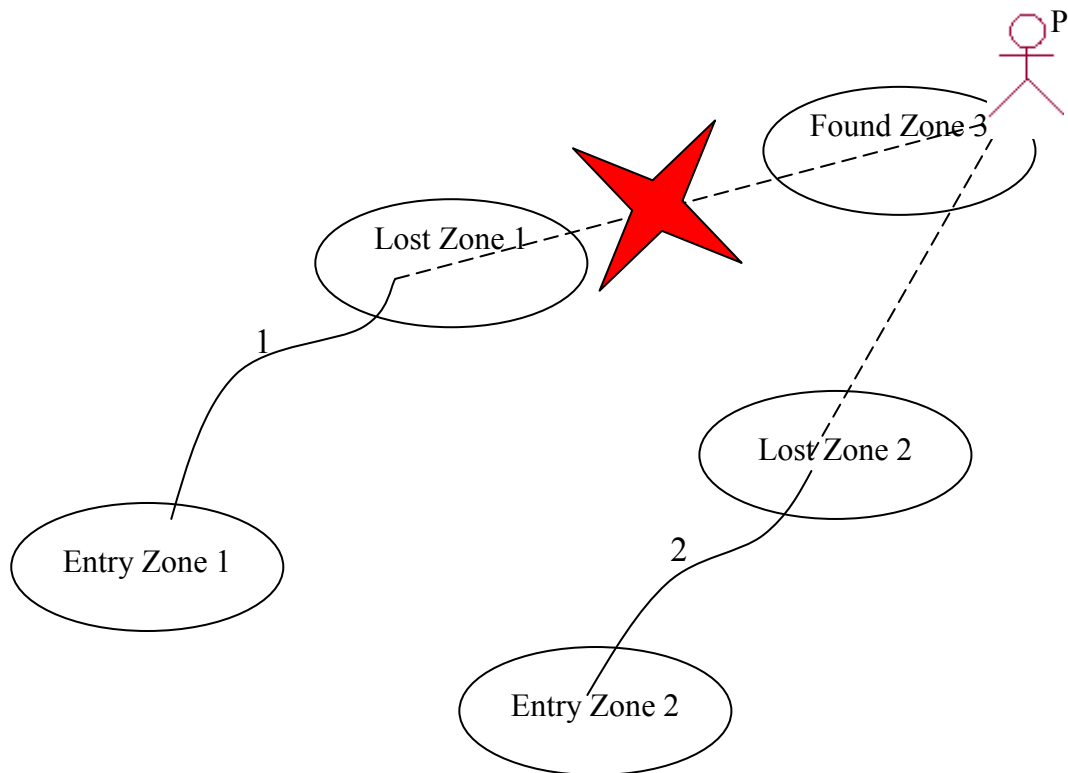


Fig 4.9 Illustration de l'algorithme de réparation des trajectoires. Personne P apparaît à 'found zone 3'. Il y a deux triplets <entry zone 1, lost zone 1, found zone 3, Mi1, Ma1> et <entry zone 2, lost zone 2, found zone 3, Mi2, Ma2>, deux trajectoires 1 et 2 perdues le suivi à 'lost zone 1' et 'lost zone 2'. Le deuxième triplet a la priorité plus haute. Alors la fusion entre la deuxième trajectoire et la personne P est réalisée.

CHAPITRE V

EXPERIMENTATION ET VALIDATION

La phase d'expérimentation est testée dans les vidéos du projet Européen CARETAKER (<http://www.ist-caretaker.org>). Le but de ce projet est de suivre les mouvements et de reconnaître les activités des personnes dans les stations de métro. Ce type de vidéo est vraiment intéressant parce que le mouvement des personnes est divers. Il n'y a pas clairement les chemins et les personnes peuvent se déplacer partout. C'est pour cela que l'exploitation de la sémantique de la scène est très difficile et exige les techniques de l'apprentissage. Dans la phase d'apprentissage, nous avons réalisé l'algorithme 'clustering' dans une vidéo d'une durée de cinq heures. Seules 23 trajectoires ont été choisies pour construire les triplets et 12 triplets sont trouvés. Dans la phase de test, le système a détecté 340 cas qui peuvent être réparés par l'algorithme proposé. C'est-à-dire 680 trajectoires sont déjà fusionnées après la réparation. Pour évaluer l'efficacité de l'algorithme, nous avons calculé la valeur de fiabilité avant et après la fusion des trajectoires. Dans 340 cas détectés, il y a jusqu'à 337 cas où la valeur de fiabilité augmente. Cependant, il y a quelques cas qui montrent que le système réalise une fusion imprécise. L'erreur se passe quand le système perd le suivi de plusieurs trajectoires en même temps mais dans les triplets différents. Dans ce cas, l'objet perdu va être associé avec la trajectoire perdue qui se trouve dans le triplet ayant la priorité la plus élevée. Cela peut entraîner parfois une fusion incorrecte (voir la figure 4.8 et on suppose que si la trajectoire 1 est la trajectoire dans le passé de la personne P, alors le système a mal réparé dans ce cas).

Les trajectoires dont la valeur de fiabilité est inférieure à un seuil donné, sont les bruits. Le système arrive à filtrer 4550 cas sur 8394 trajectoires, équivalents à 54.2%, qui sont les bruits. Après la réparation, ce nombre est 4481. Les bruits sont définis comme les trajectoires des objets mobiles qui ne sont pas des personnes (voir la section 4.3). Toutefois, la phase de d'expérimentation a montré que ce sont la durée et la longueur d'une trajectoire qui possèdent les poids importants pour l'expression de la valeur de fiabilité. En conséquence, les trajectoires dont la longueur est courte, même des personnes qui sont également classifiées dans la classe 'bruits'. C'est pour cela que, non seulement les trajectoires incomplètes sont réparées, mais aussi les bruits ont une capacité d'être réparés et seront classifiés dans la classe 'trajectoires incomplètes' plus tard.

Dans le tableau 5.1 on voit un résumé des données dans les deux cas : appliquer et sans appliquer l'algorithme de réparation des trajectoires. Après avoir réparé 340 cas, le nombre des trajectoires diminue clairement 340. Dans la colonne à droite ('Appliquer la réparation'), on peut constater qu'il y a une augmentation de nombre des trajectoires complètes et une diminution de nombre des trajectoires incomplètes et de bruits. Après la réparation, le nombre des trajectoires complètes augmente 37, le nombre des bruits diminue 69. Le changement le plus important entre les deux cas réside dans la classe 'Trajectoires incomplètes'. Dans cette classe, on voit une diminution jusqu'à 308 trajectoires. Cela s'explique par plusieurs trajectoires incomplètes qui ont été réparées, toutefois les trajectoires de résultats restent eux-mêmes dans leur classe. L'augmentation de la valeur de fiabilité dans ces cas ne suffit pas pour qu'elles soient classifiées dans la classe 'trajectoires complètes'.

	Sans appliquer la réparation		Appliquer la réparation	
	Nombre	Pourcentage (%)	Nombre	Pourcentage (%)
Trajectoires complètes	758	9.0	795	9.9
Trajectoires incomplètes	3086	36.8	2778	34.5
Bruits	4550	54.2	4481	55.6
Total	8394	100	8054	100

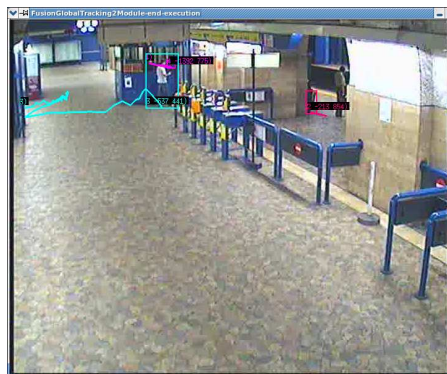
Tableau 5.1 Statistique les donnés dans deux cas : appliquer et sans appliquer l’algorithme de réparation

Maintenant, on va voir quelques résultats de démonstration. Dans la figure 5.1, l’image a) est capturée avant que le système perde la trajectoire d’ID = 3 (couleur verte). L’image b) est capturée après 2s. La personne est détectée, mais le système n’arrive pas à reconnaître que c’est la même personne dans la trame il y a 2 secondes (nouveau id assigné à 5, trajectoire est colorée en jaune). Toutefois dans l’image c), grâce à l’algorithme proposé, le système est capable de réparer la perte de la trajectoire. On arrive à reconnaître et fusionner la partie perdue de la trajectoire. Le mouvement de la personne est suivi complètement.



a)

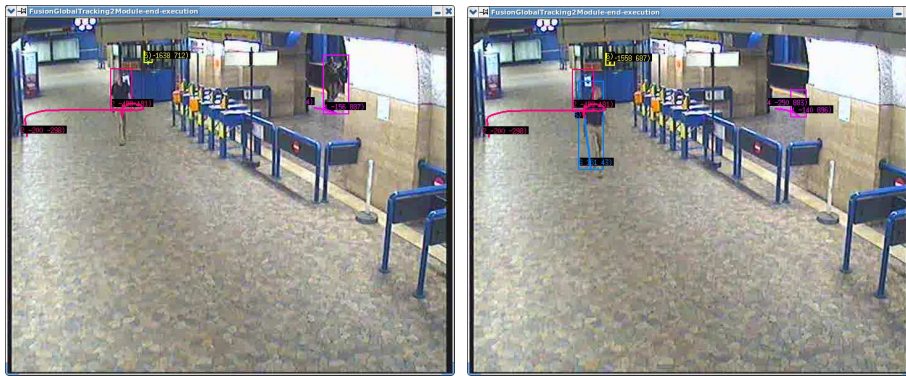
b)



c)

- Fig 5.1 Images capturées avant et après la fusion de la trajectoire à $t = 711$ s
- a) $t = 709$
 - b) $t = 711$, pas d'appliquer l'algorithme de réparation. La trajectoire de personne est coupée.
 - c) $t = 711$, appliquer l'algorithme de réparation. La trajectoire de personne est complète.

On peut voir un autre exemple dans la figure 5.2. Dans cet exemple, le système peut suivre avec succès la trajectoire de la personne avec $id = 2$ (couleur rose).



a)

b)



c)

Fig 5.2 Images capturées avant et après la fusion de la trajectoire à $t = 903s$
 a) $t = 901$
 b) $t = 903$, pas d'appliquer l'algorithme de réparation. La trajectoire de personne est coupée.
 c) $t = 903$, appliquer l'algorithme de réparation. La trajectoire de personne est

CHAPITRE VI

CONCLUSION ET PERSPECTIVE

Ce rapport présente une méthode pour réparer les trajectoires des objets dans les vidéos. Le principe de cette approche est d'utiliser l'apprentissage supervisé pour exploiter la sémantique de la scène. Les zones comme 'IO zone', 'entry zone' et 'exit zone' sont définies manuellement pour indiquer où un objet mobile peut entrer/sortir, entrer ou sortir de la scène respectivement. Les zones définies comme 'lost zone', 'found zone' et 'lost-found zone' sont construites par le processus de clustering des positions 3D où le système perd, retrouve et perd/retrouve respectivement les trajectoires. Neuf caractéristiques de la trajectoire sont extraites et sont utilisées pour calculer la valeur de fiabilité de chaque trajectoire. Un algorithme génétique est également proposé afin de déterminer l'importance de chaque caractéristique. La valeur de fiabilité est utilisée pour filtrer les bruits. Les meilleures trajectoires sont utilisées pour construire les triplets de zones. Un triplet de zones est une 'représentation' d'un chemin dans la scène où le suivi des trajectoires peut être perdu. Les triplets de zones sont très compatibles dans les scènes complexes où il n'y a pas clairement de chemins et de routes, comme la station de métro testée dans la phase d'expérimentation par exemple. De plus, grâce aux triplets, le système peut détecter l'apparition anormale des objets dans la scène, la perte du suivi des trajectoires, et la réparer.

Toutefois, il existe encore quelques limites à cette approche et il faut améliorer pour réparer mieux les pertes de suivi. Bien que nous ayons défini plusieurs éléments pour caractériser la trajectoire et pour calculer la valeur de fiabilité d'une trajectoire, il faut ajouter de plus les

caractéristiques pour mieux évaluer la fiabilité d'une trajectoire dans plus de situations possibles.

La modélisation de la scène dépend strictement de la vidéo d'entrée pour l'apprentissage. Une vidéo qui a assez de trajectoires typiques, va bien assurer le résultat de la modélisation. En revanche, si l'on n'a pas assez de trajectoires typiques, la scène ne peut pas être bien apprise et modélisée.

Les résultats obtenus ont montré que la fiabilité de plusieurs trajectoires a été estimée assez petite en raison des conditions de suivi (il existe beaucoup de scène qui sont bondés). Toutefois, bien que l'utilisation des triplets de zones arrive à réparer certaines trajectoires, on ne peut pas réparer tous les cas détectés à cause de la complexité des activités des personnes dans la scène et d'une grande quantité de trajectoires courtes et incomplètes à fusionner. Les résultats montrent également que la valeur de fiabilité est assez fiable.

Cet algorithme doit être testé avec d'autres vidéos, et dans différents scénarios. Bien que le travail ait certains points faibles, cette méthode ouvre une nouvelle approche pour améliorer tous les algorithmes de suivi et ces premiers résultats font l'ouverture d'études intéressantes et prometteuses dans le futur. Par exemple, on peut tout d'abord augmenter le nombre de zones définies dans le triplet à quatre ou cinq, c'est-à-dire 'le chemin' modélisé serait plus long, la capacité de fusion serait donc plus flexible et plus précise. Deuxièmement, la première zone du triplet n'est pas obligatoirement une 'entry zone' ou 'IO zone', mais elle pourrait être n'importe quel type de zone. Le type de première zone tel qu'a proposé dans cet algorithme a limité l'espace de la recherche des trajectoires pour la fusion et, par conséquent, a limité la flexibilité de la réparation.

REFERENCES

- [1] A. Almeida, J. Almeida, and R. Araujo, “Real-time tracking of multiple moving objects using particle filters and probabilistic data association,” *Automatika*, vol. 46, no. 1-2, pp. 39–48, 2005.
- [2] Alain Boucher, cours Vision par ordinateur, 2007
- [3] A. Avanzi, Francois Bremond, Christophe Tornieri and Monique Thonnat, Design and Assessment of an Intelligent Activity Monitoring Platform, in *EURASIP Journal on Applied Signal Processing*, special issue in "Advances in Intelligent Vision Systems: Methods and Applications", 2005:14, pp.2359-2374.
- [4] P. Bouthemy and P. Lalande. Recovery of moving objects masks in an image sequence using local spatiotemporal contextual information. *Optical Engineering*, 32(6) :1205–1212, June 1993.
- [5] Francois Brémond, Scene Understanding: perception, multi-sensor fusion, spatio-temporal reasoning and activity recognition, HDR Université de Nice-Sophia Antipolis, July 2007.
- [6] E. Brookner, *Tracking and Kalman Filtering Made Easy*, John Wiley & Sons, 1998.
- [7] S-C.S. Cheung and C. Kamath. Robust techniques for background subtraction in urban traffic video. In *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Nice, France, 2003.
- [8] F. Cupillard, A. Avanzi, F. Brémond and M. Thonnat, Video Understanding for Metro Surveillance. The IEEE ICNSC 2004 in the special session on Intelligent Transportation Systems, Taiwan, March 2004.

- [9] Yiğithan Dedeoğlu. Moving Object detection, Tracking and Classification for Smart Video Surveillance, thesis to obtain the degree of Master of Science, August, 2004.
- [10] Cao Tien Dung, Vidéo Surveillance, Rapport final du Travail d'Intérêt Personnel Encadré, 2007.
- [11] Fernyhough, J H, Cohn, A G & Hogg, D C Generation of semantic regions from image sequences in: Buxton, B & Cipolla, R (editors) ECCV'96, pp.475-478. Springer-Verlag. 1996.
- [12] R. Jain and H.H. Nagel. On the analysis of accumulative difference pictures from image sequences of real world scenes. In IEEE trans. on PAMI, volume 206(18), pages 206–214, 1979.
- [13] RC Johnson, A Brief Summarization of the Kalman Filter
- [14] K.P. Karmann and A. Von Brandt. Time-varying image processing and moving object recognition. Elsevier, 1990.
- [15] D. Makris, T. Ellis, Learning semantic scene models from observing activity in visual surveillance, IEEE Transactions on Systems, Man and Cybernetics, Part B 35 (3) (2005) 397–408.
- [16] S. Maskell, N. Gordon, M. Rollason and D. Salmond, Efficient Multitarget Tracking using Particle Filters, Journal in Image and Vision Computing, 21(10): 931-939, September 2003.
- [17] Burkay Birant Örtten, thesis to obtain the degree of Master of Science, Moving Object Identification and Event Recognition in Video Surveillance system, July 2005.

- [18] Raquel R. Pinho, João Manuel R. S. Tavares, Miguel V. Correia, An Efficient and Robust Tracking System using Kalman Filter, VIPSI-2006 VENICE, 2006.
- [19] Raquel R. Pinho¹, João Manuel R. S. Tavares and Miguel V. Correia, A Movement Tracking Management Model with Kalman Filtering, Global Optimization Techniques and Mahalanobis Distance, Lecture Series on Computer and Computational Sciences Volume 1, 2005, pp. 1-3.
- [20] Din-Chang Tseng, Chun-Wei Lin, and Chi-Ming Ling, “Motion object detection and tracking based on adaptive background subtraction,” in Proc. of the 18th IPPR Conf. on Computer Vision, Graphics and Image Processing, Taipei, Taiwan, Aug.21-23, 2005, CD.