| | |
|---|---|
| **Titre:** Title: | Détection et suivi d'objets en mouvement dans une scène filmée par une carméra fixe |
| **Auteur:** Author: | Rafik Mohamed El-Kamel Bourezak |
| **Date:** | 2006 |
| **Type:** | Mémoire ou thèse / Dissertation or Thesis |
| **Référence:** Citation: | El-Kamel Bourezak, R. M. (2006). Détection et suivi d'objets en mouvement dans une scène filmée par une carméra fixe [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/7865/ |

UNIVERSITÉ DE MONTRÉAL

# DÉTECTION ET SUIVI D'OBJETS EN MOUVEMENT DANS UNE SCÈNE FILMÉE PAR UNE CAMÉRA FIXE

RAFIK MOHAMED EL-KAMEL BOUREZAK
DÉPARTEMENT DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
AOÛT 2006

*Your file Votre référence*
*ISBN: 978-0-494-19284-9*
*Our file Notre référence*
*ISBN: 978-0-494-19284-9*

UNIVERSITÉ DE MONTRÉAL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

DÉTECTION ET SUIVI D'OBJETS EN MOUVEMENT DANS UNE SCÈNE FILMÉE PAR UNE CAMÉRA FIXE

présenté par : BOUREZAK Rafik Mohamed El-Kamel
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de :

Mme. NICOLESCU Gabriela, Doct., président
M. BILODEAU Guillaume-Alexandre, Ph.D., membre et directeur de recherche
Mme. CHERIET Farida, Ph.D., membre

*À ma très chère mémé, décédée le 12 avril 2001.*

*À mes très chers parents, ma mère Safia et mon père Salah, pour tout l'amour et le soutien incommensurable qu'ils m'ont toujours offert durant ma vie.*

*À ma sœur Rym, à mes frères Ahmed Hamza et Chams Eddine, pour leur solidarité, leur soutien et surtout leur patience depuis toujours.*

*À l'Algérie ma terre natale.*

# REMERCIEMENT

Je tiens à exprimer ma gratitude à mon directeur de recherche Guillaume-Alexandre Bilodeau tant au niveau technique qu'au niveau financier. Je le remercie particulièrement pour sa patience, sa disponibilité et ses conseils qui ont favorisés un climat de travail agréable. Son expertise dans le domaine m'a été très bénéfique pour la conduite de ce travail.

Je tiens aussi à remercier la professeure Farida Cheriet pour m'avoir permis de travailler dans son laboratoire de recherche (LIV4D) au début de ma maîtrise. Travailler dans le LIV4D m'a permis non seulement de bien démarrer mon projet, mais aussi d'établir un bon contact avec les membres de son équipe, notamment Reda Ennajih, Pascal Fallavollita, Jérémie Thériault et Philippe Debanné qui ont créé une atmosphère de travail plaisante tout au long du projet. Merci aussi à Thomas Hurtut et Hérvé Lombaert pour avoir prêté leurs talents d'acteur lors de la capture de séquences vidéo.

Finalement, je tiens à remercier tous mes amis à travers le monde pour leur soutien moral et leur éternelle bonne humeur, tout particulièrement Momo de Paris, Bachir Lanez et Salim Dechmi d'Alger, et Oscar & los espagnoles de Madrid!

# RÉSUMÉ

La recherche reliée à la télésurveillance gagne de plus en plus d'intérêt. Car un bon système de télésurveillance pourrait non seulement être utilisé pour sécuriser les lieux en détectant les mouvements suspects, mais pourrait aussi assister les personnes âgées chez elles en détectant les situations dangereuses. Ces systèmes peuvent même être utilisés pour faire une reconstruction 3D dans un environnement virtuel à des fins de jeux vidéo.

Le LITIV (Laboratoire d'Interprétation d'Images et Vidéos) s'intéresse particulièrement à étudier les divers problèmes reliés à la télésurveillance pour proposer de nouvelles méthodes pour contribuer à améliorer les systèmes existants.

L'objectif de ce mémoire est d'étudier et d'implanter des algorithmes pour la détection et le suivi d'objets en mouvement dans une scène filmée par une camera fixe.

À la base, un système de surveillance peut être implanté en trois étapes distinctes. D'abord, détecter les objets en mouvement dans la scène, puis faire le suivi de ces objets et enfin interpréter la relation qui existe entre eux s'il y en a.

Pour la détection des objets en mouvement dans la scène, nous présentons une nouvelle approche basée sur les histogrammes de couleurs et la division itérative de régions intéressantes de l'image. Les histogrammes de couleurs sont invariants à la rotation, la translation, la déformation, ce qui les rend utiles pour détecter les régions où des changements se produisent, indiquant ainsi qu'un mouvement important a été

détecté à cette endroit. Ainsi, ces régions là sont divisées à leur tour pour obtenir plus de précision sur les objets en mouvement.

Pour le suivi, des informations sur les objets en mouvement sont rassemblées pour pouvoir les identifier. La couleur et la texture sont des informations pertinentes pour différencier entre deux objets dans la scène. C'est pour cela que les corrélogrammes sont utilisés. Ils permettent d'étudier la distribution spatiale des couleurs en combinant les informations de la texture et de la couleur. Comme les histogrammes, ils sont invariants à la rotation, la translation, et la déformation. De plus, des études ont prouvé que les corrélogrammes sont plus fiables que les histogrammes pour faire la distinction entres deux images. Ainsi, en utilisant l'intersection des corrélogrammes nous pouvons savoir si deux objets sont les mêmes, si l'un fait partie de l'autre.

Enfin, pour étudier la relation qui existe entre les objets suivis, dans le cadre de ce travail, nous ne cherchons pas à identifier ces objets. Nous nous intéressons plutôt à savoir si des objets ont fusionné ou se sont séparés dans la scène. Pour ce faire la position de chaque objet est utilisée en plus de la couleur et la texture.

Un des problèmes qui revient le plus souvent dans les algorithmes de télésurveillance est relié à l'intensité de la lumière, cela peut être dû aux changements de l'intensité, ou à la présence de différentes sources de lumières. Pour pallier à ce problème, nous travaillons dans l'espace de couleur HSV (Teinte, Saturation et Valeur). Cet espace de couleur permet de contrôler l'intensité de la lumière selon la

quantification choisie et ainsi diminuer l'influence de l'intensité sur les algorithmes pour les rendre plus robustes.

Les algorithmes proposés ont été implantés, et des tests ont été effectués sur différentes séquences vidéo prises de scènes d'intérieur avec diverses conditions d'éclairage. Les tests ont été effectués en montrant la validité des méthodes implantées. Notamment le fait que la méthode de détection est robuste aux distributions multimodales, et permet de varier le niveau de précision pour segmenter les objets détectés. En ce qui concerne le suivi, l'approche adoptée utilise des informations robustes pour décrire les objets, ce qui permet de les suivre et de détecter les situations de fusion et de séparation. Cela dit, il y a des parties qui peuvent être améliorées. Par exemple, un peu de bruit subsiste sur les bords des objets détectés.

Dans les travaux futurs, les algorithmes pourraient être renforcés en intégrant des méthodes de calibrage pour connaître la distance des objets de la caméra et en adaptant la quantification des couleurs dynamiquement selon la situation. Plus encore, on peut rajouter des méthodes pour détecter les parties de l'objet qui risquent d'être obstruées dans des endroits précis de la scène. Enfin, une méthode de détection d'arêtes pourrait être rajoutée pour raffiner les bordures des objets détectés.

# ABSTRACT

The research connected to remote monitoring gains more and more interest. A good system of remote monitoring could not only be used to make public places safe by detecting suspect movements, but could also assist old people at their home by detecting dangerous situations. These systems can also be used in video games to reconstruct the detected objects in a virtual environment.

The LITIV (Laboratoire d'Interprétation d'Images et Vidéos) is particularly interested to study various problems connected to remote monitoring, to propose new methods and thus contribute to improve existing systems.

The objective of this master thesis is to study and establish algorithms for the detection and the tracking of moving objects in a scene filmed by a fixed camera.

Basically, a monitoring system can be established in three distinct stages. Initially, detect the objects moving in the scene, then make the tracking of these objects and finally interpret the relationships which exist between them if there is any.

For the detection of the objects moving in the scene, we present a new approach based on histograms of colors and iterative division of interesting areas of the image. The color histograms are invariants to rotation, translation and scaling, which makes them useful to detect the areas where changes occur, indicating that important movement was detected there. Thus, these areas are subdivided themselves to obtain more precision on the moving objects.

For tracking, information on moving objects is gathered to be able to identify them frame by frame. Color and textures are relevant information to differentiate between two objects in the scene, therefore the correlograms are used. They make it possible to study the spatial distribution of the colors by combining information of texture and color. Like the histograms, they are invariants to rotation, translation, and scaling. Moreover, studies proved that the correlograms are more reliable than the histograms to make the distinction between two images. Thus, by using the intersection of the correlograms we can know if two objects are similar or if one belongs to the other.

Finally, to study the existing relationship between the tracked objects within the framework of this work, we do not seek to identify these objects. We are interested rather in knowing if objects have fused or split in the scene. To do that, the position of each object is used in addition to the color and texture.

One of the problems which generally occur in the algorithms of remote monitoring is related to the light intensity that is due to the changes of the intensity, or the presence of various sources of lights. To fix this problem, we work in the HSI color space (Hue, Saturation and Intensity). This color space makes it possible to control the intensity of the light according to the selected quantification and thus to decrease the influence of intensity on the algorithms to make them more robust.

The proposed algorithms were implemented, and tests were carried out on different indoor video sequences with various lightning conditions. Tests were carried

out showing the validity of the established methods. In particular the method of detection is robust with multimodal distributions, and makes it possible to vary the level of precision to segment the detected objects. Concerning tracking, the adopted approach uses robust information to describe the objects, which makes it possible to follow them and detect the situations of fusion and separation. However, there are parts which can be improved. For example, a little noise remains on borders of the detected objects. This noise is generally due to large reflection of the objects.

In future work, the algorithms could be improved by integrating methods of calibration to know the distance of the objects from the camera and by adapting the quantification from the colors dynamically according to the situation. Moreover, one can add methods to detect the parts of the object which are likely to be occluded in precise places in the scene. Finally, edge detection method could be added to improve the detection of the objects.

# TABLE DES MATIÈRES

# LISTE DES TABLEAUX

# LISTE DES FIGURES

# LISTE DES NOTATIONS ET SYMBOLES

| | |
|---|---|
| $F_i$ | Image de la séquence en cours de traitement |
| $A_i$ | Image représentant l'arrière-plan. |
| $I_{cour}$ | Image courante |
| $I_{ref}$ | Image de référence |
| Th | Seuil |
| H | Histogramme |
| HI | Intersection des histogrammes |
| C | Corrélogrammes |
| CI | Intersection des corrélogrammes |
| $L_1$ | Mesure de distance |
| $D_1$ | Mesure de distance |
| $t_v$ , $t_s$ ,. $t_h$ | Seuils pour le détecteur d'ombrage |
| Obj | Objet |
| Séq. | Séquence |

# LISTE DES ANNEXES

# CHAPITRE 1
# INTRODUCTION

## 1.1 Mise en contexte

De nos jours, la télésurveillance est un important champ de recherche dans le domaine de la vision par ordinateur [1]. Les organisations qui ont besoin d'un système de surveillance peuvent obtenir des caméras de surveillance pour un prix modique. Cela dit, ils ont toujours besoin d'agents qui surveillent en permanence les écrans de surveillance. Et là encore, ces agents ne peuvent pas observer tous les écrans en même temps. Ce qui fragilise ces systèmes de surveillance. Une réponse à ces difficultés est l'utilisation de systèmes de vision artificielle qui analysent continuellement les vidéos et peuvent aviser automatiquement un agent de sécurité lorsqu'un événement suspect est en cours. Cependant, ces technologies robotisées de surveillance sont seulement à leurs premiers balbutiements. De tels systèmes ont le potentiel de faire la surveillance automatique d'un lieu et de faire intervenir les agents de sécurité que lorsqu'une intervention est requise ou qu'un événement suspect est détecté. De tels systèmes rendraient donc plus significatif le rôle des caméras dans la prévention des pertes de vie et des blessures.

Par ailleurs, la télésurveillance n'est pas utile seulement aux organisations. Les particuliers peuvent aussi en bénéficier. Plus précisément les personnes en besoin d'assistance, telles que les personnes âgées ou encore les personnes à mobilité réduite,

qui désirent demeurer chez eux. Ce domaine de surveillance s'appelle la domotique [2]. Un système de surveillance placé dans les maisons de ces personnes peut détecter les situations dangereuses telles qu'une chute ou une immobilité trop longue, pour automatiquement prévenir les services d'urgences.

Cela dit, les algorithmes développés pour la surveillance ne se limitent pas seulement à la sécurité des personnes et des lieux, ils sont aussi utilisés à des fins éducatives et l'amusement des enfants comme dans le cas du système KidsRoom [3]. Dans une chambre d'immersion en 3 dimensions, le système détecte et interprète les mouvements des enfants présents dans la pièce pour intégrer leurs avatars dans l'environnement virtuel où une histoire est racontée et visualisée selon les actions des enfants.

### 1.2 Problématique et travaux antérieurs

La surveillance par vision artificielle est un domaine de recherche en expansion où de nombreux problèmes sont rencontrés. Premièrement, la détection d'un humain [4] dans une séquence vidéo n'est pas solutionnée pour plusieurs situations, particulièrement quand les conditions d'éclairage sont variables. Pour cette raison, la plupart des travaux en détection d'humains s'intéressent à des situations où la caméra est fixe et où les conditions d'éclairage ne changent pas brusquement. Les méthodes de soustraction d'arrière-plans existantes travaillent sur une base de pixel par pixel, ce qui les rend particulièrement sensibles au bruit dû à l'éclairage et au système d'acquisition. Un sommaire de ces méthodes est présenté par Cucchiara et al.[5]. La détection de l'humain

est particulièrement difficile car sa forme, son comportement, sa couleur, et même sa texture varient d'un humain à un autre. Une des méthodes les plus robustes qui existe est celle proposée par Stauffer et Grimson [6]; elle consiste à utiliser une mixture de K Gaussiennes, permettant ainsi de prendre en compte les distributions multimodales telles que l'oscillation de l'eau ou des feuilles d'arbres. Ces oscillations rendent particulièrement difficile la segmentation des objets dans la séquence parce que pour le même pixel de l'image plusieurs variations sont effectuées sans pour autant que cela soit un mouvement intéressant. Toujours est-il que cette méthode travaille sur une base de pixel par pixel ce qui laisse du bruit s'inclure plus facilement dans les résultats.

Deuxièmement, les méthodes utilisées pour faire le suivi des humains/objets détectés sont généralement basées sur des statistiques, des phases d'apprentissage et reposent sur plusieurs hypothèses, ce qui les fragilise. W4 [7] par exemple est un système qui construit un modèle pour prédire la position de l'objet suivi dans la prochaine image de la séquence puis ce modèle prédictif est ajusté selon la nouvelle position réelle de l'objet. La principale contrainte rencontrée est que le comportement des objets en mouvement doit être constant, car des changements de direction brusques perturberaient le système.

Enfin, tandis que plusieurs approches existent pour faire la détection des objets en mouvement et les suivre, l'interprétation des événements dans la scène est rarement adressée. Spécialement en ce qui concerne la prise ou le dépôt d'objets dans la scène ou même la reconnaissance de ces objets pour les classer comme étant dangereux ou non.

W4 [7] localise le transport d'objets en supposant que l'humain a une silhouette symétrique, ce qui lui permet de conclure qu'un objet est transporté quand une déformation de la symétrie est observée. Ainsi, parmi les limitations de cette approche, la prise de vue doit permettre de voir clairement la déformation de la silhouette créée par l'objet transporté. Cela dit, la silhouette de l'humain elle-même est déformable, ce qui diminue la robustesse du système.

Dans ce contexte, la problématique qui nous intéresse est de faire la détection des régions en mouvement dans la séquence et de faire le suivi de ces régions par une caméra fixe selon des conditions d'éclairage variables. Cela présuppose une approche originale où les changements dans l'image doivent être détectés et la région obtenue doit correspondre précisément à l'objet en déplacement sans être altérée par son ombre ou des mouvements d'arrière-plans tels que l'oscillation des feuilles d'arbres ou de l'eau. Ensuite l'objet est suivi tout en interprétant des événements simples tels que la fusion et la séparation de deux objets dans la scène qui permettrait éventuellement de savoir quand un objet est déposé ou pris.

### 1.3 Objectifs

Ce mémoire a quatre objectifs :

1. La détection des objets en mouvement. Proposer un algorithme pour faire la détection des objets en mouvement dans une scène. L'algorithme doit être

robuste aux changements graduels de lumière et au bruit dû à la capture des séquences vidéo.

2. Faire le suivi des objets détectés. Proposer un algorithme pour faire le suivi des objets en combinant les informations de textures et de couleurs pour distinguer entre les objets présents dans la scène.

3. Interpréter la relation entre les objets détectés dans la scène. Développer un algorithme qui interprète les relations qui existent entre les objets, plus spécifiquement un algorithme qui aide à savoir si deux objets se sont séparés ou ont fusionné.

4. Valider les différents algorithmes proposés. Effectuer des tests de validation sur des séquences prises dans des scènes intérieures composées d'humains et d'objets avec des conditions d'éclairage variables. Certaines de ces séquences seront prises de bases de données partagées avec d'autres équipes de recherche pour comparer les performances de nos algorithmes avec les performances de travaux déjà existants.

Ainsi l'approche proposée se divise en trois parties. L'algorithme de la détection travaille sur des blocs de régions pour être moins sensible au bruit dans l'image. L'idée est de diviser itérativement l'image en carrés de tailles similaires. A chaque étape, la distribution de couleur des régions de l'image de référence et l'image courante est

comparée. Si un changement s'est produit, l'algorithme continue la division itérative pour ces régions là.

Pour ce qui est du suivi, comme deux objets distincts sont rarement susceptibles d'avoir la même distribution spatiale de couleur, cette information est utilisée pour suivre et distinguer les objets détectés. Enfin, une analyse d'hypothèses est faite pour faire l'interprétation de la relation entres les objets présents dans la scène, plus précisément pour savoir si une séparation ou une fusion entre deux objets s'est produite.

## 1.4. Contribution

La contribution de ce mémoire est :

1- Le développement d'un nouvel algorithme pour la détection d'objets en mouvement. Cet algorithme utilise une soustraction d'arrière-plan basée sur la division successive des régions intéressantes localisées grâce à la comparaison entre les histogrammes de couleurs des régions de l'image courante et celles de l'image de référence qui est mise à jour régulièrement. Parmi les avantages de la méthode développée, la détection des objets se fait en général sans avoir recours à des prétraitements spéciaux des images. Cet avantage est dû au fait que le traitement se fait par blocs de régions et non pixel par pixel, éliminant ainsi l'impact des pixels bruités sur la segmentation des objets.

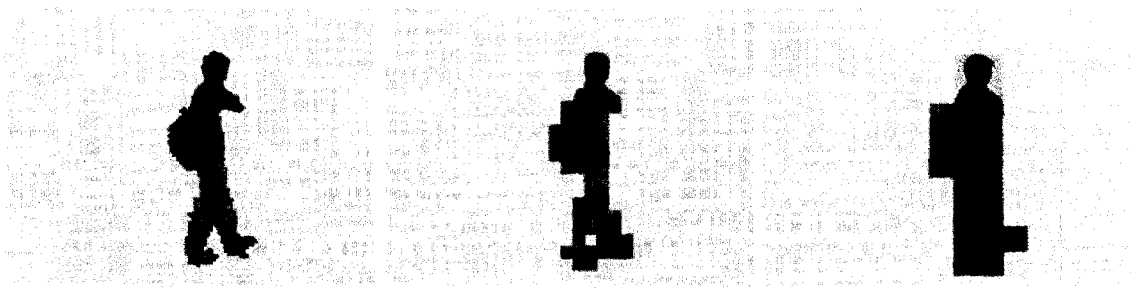
Cette méthode permet aussi de décider quel niveau de précision on veut obtenir pour les objets en mouvement dans la scène (figure 1.1). Un autre avantage est que

l'arrière-plan peut être graduellement mis à jour dans les carrés où aucun mouvement n'est détecté.

2- L'utilisation des corrélogrammes dans l'espace HSV pour suivre les objets en mouvement et faire le raffinement du suivi pour les cas de fusion et séparation de régions en mouvement.

Ce mémoire est organisé de la façon suivante :

Dans le chapitre 2, un aperçu des techniques classiques et récentes pour faire la détection de mouvement et le suivi est présenté. D'autres notions requises au traitement d'images telles que la couleur et la texture ainsi que les structures utilisées sont décrites. Dans le chapitre 3, les algorithmes proposés pour faire la détection et le suivi sont présentés en détails. Dans le chapitre 4, les résultats expérimentaux sont présentés incluant une étude comparative avec d'autres travaux existants. Enfin, dans la conclusion une récapitulation sur les contributions de ce mémoire est faite et une perspective sur les travaux futurs est présentée.



A) B) C)

**Figure 1.1.** Détection à différents niveaux de précision.

# CHAPITRE 2

# REVUE DE LA LITTÉRATURE

Beaucoup de développement a été fait dans le domaine de la télésurveillance se basant sur des raisonnements différents. L'objectif principal d'un système de télésurveillance est la détection des objets en mouvement dans une scène. Généralement ces systèmes vont plus loin en faisant le suivi et l'interprétation des mouvements.

Dans ce chapitre nous allons traiter les travaux reliés à la détection d'objets en mouvement dans une scène d'une part, et d'autre part ceux qui traitent le problème du suivi des objets détectés. Cela dit, nous allons nous limiter aux travaux qui traitent des scènes filmées par des caméras fixes.

Pour faire la détection et le suivi, il est nécessaire d'utiliser certaines informations sur la séquence d'images et les objets détectés. Les informations utiles et les structures utilisées pour les analyser sont présentées dans ce chapitre.

## 2.1 Détection des objets en mouvement :

L'algorithme consiste à détecter des objets en mouvement dans une scène filmée par une caméra fixe. L'algorithme de détection doit être robuste et capable de s'adapter aux conditions variables d'une scène à une autre telle que la luminosité. Il est important à cette étape d'avoir une bonne détection pour pouvoir faire le suivi et l'interprétation correctement.

Il y a deux étapes importantes dans la détection d'objets en mouvement, l'ordre diffère d'un algorithme à un autre :

1- La maintenance de l'arrière-plan. Elle consiste à mettre à jour périodiquement une image modèle considérée comme l'arrière-plan, pour pouvoir détecter les changements qui se sont produits dans la scène par la suite. L'arrière-plan constitue la partie statique de la scène qui ne doit pas être détectée comme en mouvement par l'algorithme de détection. La partie qui doit être détectée constitue l'avant-plan.

2- La détection des changements dans la scène. Elle consiste à comparer l'image considérée comme arrière-plan avec une nouvelle image de la séquence pour voir si des changements se sont produits dans la scène.

La difficulté réside surtout dans la première étape, la deuxième étant généralement effectuée en faisant une soustraction entre deux images, l'image courante et celle

représentant l'arrière-plan, puis en comparant la différence de chaque pixel avec un seuil fixé selon l'approche utilisée.

Différentes approches ont été développées pour faire la détection des objets en mouvement, la plus part se basant sur l'étude de l'image pixel par pixel.

Une approche naïve serait de faire la différence entre l'image courante de la séquence $F_i$, et une image statique $F_0$ qui représente l'arrière-plan et vérifier si la différence est à l'intérieure d'un seuil Th dont la valeur est fixé expérimentalement :

- Si $|F_i(x,y) - F_0(x,y)| < Th$
  - Donc $F_i(x,y)$ appartient à l'arrière plan.
- Sinon
  - $F_i(x,y)$ appartient à l'avant plan.

Mais cette méthode comporte de multiples problèmes liés aux :

- Changements dans l'intensité de la lumière. C.-à-d. changement de l'intensité des pixels de l'image de la séquence au cours du temps. Comme l'arrière-plan n'est pas mis à jour avec le changement graduel de la lumière, les résultats sont complètement erronés.

- Mouvements constants dans l'image telle que l'oscillation de feuilles d'arbres ou le mouvement de l'eau. Ces mouvements considérés comme inintéressants changent constamment la valeur des pixels dans l'image à la même position. Donc avec cette méthode, ces mouvements sont détectés et faussent les résultats attendus.

Une autre méthode consiste à faire la soustraction de chaque paire d'images adjacentes ($F_i, F_{i-1}$) de la séquence [8].

- Si $|F_i(x,y) - F_{i-1}(x,y)| < Th$
  - Donc $F_i(x,y)$ appartient à l'arrière-plan.
- Sinon
  - $F_i(x,y)$ appartient à l'avant-plan.

Encore là, cette méthode comporte des lacunes importantes. Elle ne résout pas le deuxième problème énoncé précédemment et dépend de plusieurs paramètres tels que la vitesse des objets en mouvement pour pouvoir reconnaître les pixels qui ont bougé dans l'image.

Pour résoudre les problèmes reliés à ces méthodes de bases concernant la maintenance de l'arrière-plan, plusieurs algorithmes ont été développés.

**Méthode de la moyenne et de la médiane.** Cette méthode consiste à faire la médiane [9] ou la moyenne sur les *n* images précédentes pour chaque pixel de l'arrière-plan. Ceci permet de tenir compte des changements graduels de la lumière qui s'effectuent sur les *n* dernières images.

Le problème majeur de cette méthode est qu'elle a besoin de beaucoup de mémoire : n*T, où *T* représente la taille de l'image. Puisque chaque pixel des n dernières images doit être maintenu en mémoire.

Pour ce qui est du temps d'exécution, la méthode de la médiane prend beaucoup plus de temps puisqu'elle a besoin de trier *T* tableaux de *n* valeurs dans l'image courante avant d'effectuer le rafraîchissent.

**Valeurs minimale et maximale.** Cette méthode est utilisée par le système W4 [7]. Chaque pixel de l'arrière-plan pour un certain nombre d'images, est décrit avec trois valeurs : la valeur minimale (M), la valeur maximale (N), et la valeur maximale de la différence entre deux images successives (D). Puis deux images sont générées, la première constituant la différence entre l'image courante et l'image M, la deuxième constituant la différence entre l'image courante et l'image N. Enfin, pour chaque pixel de l'image, le pixel est classifié comme appartenant à l'avant-plan, si ses deux valeurs dans les deux nouvelles images sont plus grandes qu'un seuil construit à partir de sa valeur dans l'image D.

Un problème majeur rencontré par cette méthode est que le bruit peut s'introduire sur un pixel et modifier la valeur maximale ou minimale pour certains pixels de l'image dans l'arrière-plan et ainsi diminuer la précision de l'algorithme. Ce qui pousse à faire plusieurs prétraitements.

**Méthode de la Moyenne courante.** Pour résoudre les problèmes de la méthode précédente on pourrait calculer la moyenne au fur et à mesure qu'on passe à une nouvelle image (équation 2.1) [10].

Pour tous les pixels de l'image courante $F_{i+1}$ et l'image précédente $F_i$ on applique le filtre adaptative suivant

$$A_{i+1}(x, y) = \alpha \times A_i(x, y) + (1 - \alpha) \times F_{i+1}(x, y) \qquad (2.1)$$

α, représentant le taux d'apprentissage est fixée entre 0 et 1. Plus on se rapproche de 1, plus on donne d'importance à la moyenne précédente $A_i(x, y)$ et vice-versa. Cette valeur est fixée expérimentalement.

L'avantage principal de cette méthode par rapport à la première, est qu'elle ne consomme pas beaucoup de mémoire puisqu'on a besoin de garder juste les pixels de l'image précédente pour faire le rafraichissement de l'arrière-plan.

Comme la mise à jour s'effectue sur tous les pixels, le problème rencontré est le fait que les pixels où des mouvements sont détectés sont aussi intégrés dans l'équation pour calculer la nouvelle valeur du pixel dans l'image courante.

**Méthode de sélectivité.** [5] Dans cette méthode on utilise aussi l'équation 2.1, sauf qu'à chaque image, chaque pixel est classifié comme appartenant à l'arrière-plan ou non. Si le pixel appartient à l'avant plan, il n'est pas utilisé pour mettre à jour l'arrière-plan. Ainsi, il y a moins d'erreurs dans la moyenne.

L'équation 2.1 est transformée comme suit :

- Si $F_i$ (x, y) appartient à l'arrière plan
  - $A_{i+1}(x, y) = \alpha \times A_i(x, y) + (1 - \alpha) \times F_{i+1}(x, y)$
- Sinon
  - $A_{i+1}(x, y) = A_i(x, y)$ (2.2)

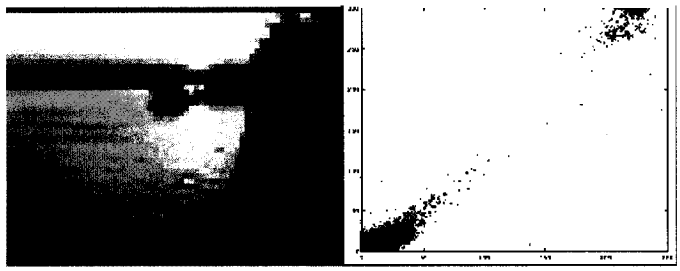
Pour raffiner cette méthode et la rendre plus robuste Heikkila et Silven [11], ont proposé de rajouter les contraintes suivantes.

1- Si un pixel n'appartient pas à l'arrière-plan pour un nombre fixe d'images alors l'arrière-plan est rafraîchi pour ce pixel; de cette manière $A_{i+1}$ (x, y) = $A_i$ (x, y). Ceci est pour palier au changement soudain d'illumination et l'apparence d'objets qui s'immobilisent pour une longue durée dans l'image.

2- Si un pixel change d'état fréquemment alors il est considéré comme un pixel de l'arrière-plan en permanence. Ceci permet de ne pas prendre en considération la réflexion de l'eau et les feuilles d'arbres qui oscillent au vent.

**Méthode Gaussienne.** Chaque pixel est modélisé séparément par une distribution Gaussienne en calculant sa valeur moyenne $\mu$ et sa covariance $\sum$ pour chaque nouvelle image. Puis chaque pixel de l'image courante est comparé avec la valeur moyenne du pixel correspondant de l'image représentant l'arrière-plan. Le pixel est considéré comme appartenant à l'arrière plan s'il est à n écart-types de la moyenne de la gaussienne du pixel appartenant à l'arrière-plan [12].

Cette méthode ne fonctionne pas pour les distributions multimodales résultant par exemple du mouvement continu des feuilles d'arbre ou l'oscillation de l'eau (Figure 2.1).



A) B)

**Figure 2.1 [6]:** A) Image représentant l'oscillation de l'eau.
B) Représente la distribution multimodale de l'eau.

Pour résoudre ce problème Staufer et Grimson [6] ont proposé d'utiliser une méthode de K (un entier variant de 3 à 5) distributions Gaussiennes. Ainsi, la distribution multimodale est prise en compte. Le poids de chaque gaussienne est proportionnel au temps que sa distribution soit trouvée dans la scène. Les couleurs les plus probables de l'arrière-plan sont celles qui restent présentes le plus longtemps. Une soustraction entre l'arrière-plan et l'image courante est effectuée pour chaque pixel de l'image :

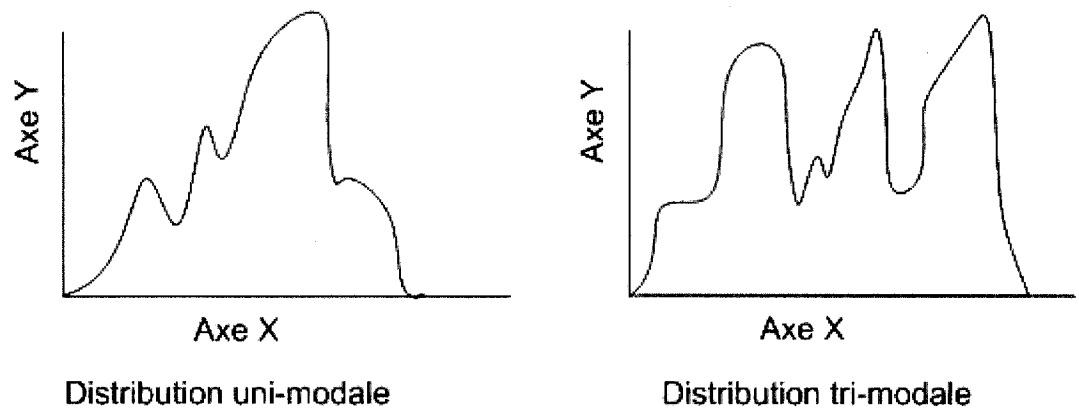



**Figure 2.2** : Histogrammes d'une distribution uni-modale et une multimodale.

- Si la valeur de la soustraction est plus de 2.5 écart-types plus loin de chacune des B distributions, alors on marque comme avant-plan ce pixel de l'image et on remplace la distribution la moins probable dans l'arrière-plan par une distribution ayant la valeur du pixel courant comme moyenne, une covariance élevée et un poids très petit.

- Sinon, les paramètres de la première distribution gaussienne qui correspond à la valeur du pixel sont mis à jour.

Une étude comparative détaillée incluant d'autres méthodes de soustraction d'arrière-plan est présentée pas Karaman et al. [13].

Parce que ces méthodes font un traitement pixel par pixel, elles sont sensibles aux bruits dans l'image. Ce problème provient principalement des caméras. En effet, même si la scène filmée possède une luminosité constante, il n'en est pas de même pour les valeurs

des pixels capturés qui parfois sont complètement fausses. Les capteurs de caméra sont affectés en particulier lorsqu'il fait sombre.

Pour palier aux problèmes rencontrés par ces approches locales, une approche proposée par Matsuyama et al. [14] serait de segmenter chaque image par régions, et comparer chaque région avec la région lui correspondant dans l'image suivante en utilisant la corrélation entre la couleur, pour déterminer si des changements se sont produits.

Comme la taille des blocs est fixe, on n'obtient pas une bonne segmentation des objets détectés et des parties des objets peuvent être perdues parce qu'elles ne sont pas assez grandes pour être détectées dans les blocs voisins.

Une méthode robuste a été introduite par Heikkila et Pietikäinen [15]. Ils ont introduit la méthode des Patrons Binaires Locaux pour modéliser l'arrière-plan et le maintenir à jour. Cette approche consiste à modéliser l'arrière-plan en utilisant l'opérateur de texture des patrons binaires locaux. Il consiste à créer pour chaque pixel de l'image un nombre binaire contenant n chiffres $c_1\ c_2\ \ldots.\ c_7\ c_n$, ce nombre est construit à partir des n voisins; $c_i = 0$ si la valeur du voisin i est plus petite que celle du pixel central sinon $c_i = 1$. Pour déterminer les voisins, une distance est requise; pour une valeur de la distance variant de 1 à m, le nombre de voisins varie de 8 (voisins directe du pixel) à m X 8. Ainsi, les voisins se trouvent à une distance fixée par l'utilisateur. Puis un histogramme de ces nombres binaires est calculé pour représenter la texture de l'image.

Heikkila et Pietikäinen [15] construisent l'arrière-plan en utilisant pour chaque pixel un vecteur incluant les K derniers histogrammes des K dernières images de la séquence, avec chaque histogramme ayant un poids donné. Ainsi, pour détecter les changements dans la nouvelle image, l'histogramme de chaque voisinage est comparé avec le vecteur des histogrammes correspondant dans l'image de références. Ils ont prouvé que cette méthode est très robuste et performe mieux dans la plus part des cas que les méthodes connues.

Même si cette méthode s'étend au voisinage des pixels, elle reste dépendante de la valeur de chaque pixel central du voisinage. Cependant, elle a prouvé, l'utilité de ne pas se limiter à l'étude de chaque pixel indépendamment des autres pour faire la détection.

Donc développer un algorithme se basant sur la couleur des pixels, qui ne se limite pas à l'étude de l'image pixel par pixel, mais plutôt par régions s'avère efficace. Il faut trouver une manière pour ne pas perdre les extrémités des objets et inclure un minimum de bruit à la détection.

**2.2 Suivi des objets en mouvement :**

Le suivi des objets détectés au cours du temps implique la reconnaissance des objets d'une image à une autre en utilisant différents types d'informations pour les décrire, tels que la couleur, la texture, la forme ou leurs contours.

En général, il y a deux types d'approches utilisées :

### 2.2.1 Approches prédictives :

De nombreux algorithmes de suivi sont basés sur des probabilités et des hypothèses. Ils construisent un modèle de prédiction pour ensuite pouvoir suivre l'objet, interpréter ses mouvements et les classifier. Ainsi, ils facilitent le suivi en trouvant l'endroit potentiel où se trouve l'objet dans la nouvelle image et dans le cas où l'objet n'est pas trouvé la prédiction peut être utilisée pour contrer la détection ratée. Parmi les approches populaires utilisées :

**Le modèle de Markov.** C'est un nombre fini d'états reliés par des transitions construites de probabilités (Figure 2.3). Chaque état génère une observation/résultat selon la distribution de probabilité qui lui est associée. Modèle Markovien, veut dire que l'état courant dépend seulement de l'état précédent. Donc, il faut calculer les probabilités des transitions pour déterminer le nouvel état (celui qui est le plus probable). En général les algorithmes de suivi qui se basent sur ce modèle, font un apprentissage à chaque nouvelle fenêtre de la séquence pour interpréter les mouvements [16, 17] en cours ou prédire la prochaine position. Ils se servent de l'état précédent pour déterminer l'état courant.

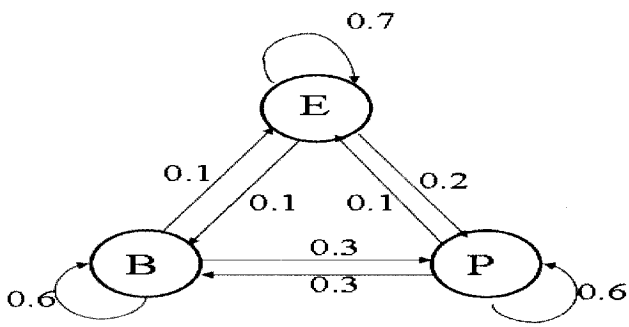



**Figure 2.3 :** Un modèle de Markov

La figure 2.3 est un exemple d'un modèle de markov, où les états représente le temps qu'il fait : ensoleillé (E), pluvieux (P), brumeux (B) avec une probabilité de transition entre eux.

**Décalage moyen (Mean Shift).** C'est un algorithme qui opère sur les distributions de probabilité en recherchant le maximum local dans un domaine donné. Il a été utilisé dans la segmentation d'images pour la première fois en 1997 [18] puis en 2000 pour faire le suivi [19]. Dans le cas du suivi, l'image est représentée comme une distribution de probabilité par des histogrammes. La position de l'objet à suivre est recherchée dans les différentes images, à partir du coefficient de Bhattacharyya, qui est un coefficient dérivé de l'erreur bayésienne, pour mesurer la similarité entre les distributions de couleurs d'un modèle de l'objet et de certaines régions de l'image. L'information sur la forme de l'objet suivi est mise à jour pour chaque changement majeur qui s'effectue au cours du temps [20].

Une variante du « Mean shift » est le « Camshift ». Elle est plus robuste car elle rafraîchit régulièrement les informations sur l'objet suivi sans attendre un changement majeur sur les informations, diminuant ainsi le risque de le perdre [21].

**Filtre de Kalman.** C'est un filtre récursif. Il est généralement utilisé pour faire le suivi des objets en mouvement en rafraichissant régulièrement les informations connues telles que la vitesse et la position. Aussi, c'est un modèle prédictif qui fait une prédiction qui est comparée avec les résultats réels pour modifier les paramètres pour améliorer la prochaine prédiction. En suivant l'objet au cours du temps, les informations sont filtrées du bruit qu'elles contiennent pour ainsi obtenir une bonne prédiction de la future position de l'objet en mouvement [22, 23, 24]

Le problème principal rencontré par cette méthode est qu'il faut que les changements soient constants dans le temps. Un changement brutal de direction ou de forme de l'objet suivi perturberait l'algorithme.

Le filtre de Kalman est souvent utilisé en combinaison avec d'autres modèles probabilistes pour rendre plus robuste le modèle de suivi et d'interprétation des mouvements. Il a été combiné notamment avec le modèle de Markov [25] et le Mean shift [26].

**Les réseaux de neurones.** C'est un modèle composé de réseaux de nœuds interconnectés par des liaisons affectées de poids dont la conception est très schématiquement inspirée du fonctionnement de vrais neurones humains. Les réseaux de

neurones sont surtout utilisés pour faire l'interprétation des mouvements produits dans la scène. Sacchi et al. [27] ont proposé un système de surveillance basé sur les réseaux de neurones pour faire la détection des actes de vandalismes et des attaques de personnes pour les classifier. Une phase d'apprentissage est requise pour reconnaître les actions à classifier.

La limitation principale de ces méthodes est qu'on suppose des mouvements facilement prédictibles. Elles ne sont pas très fiables car dans les cas où des changements brusques de direction des objets suivis se produisent, les prédictions sont souvent fausses.

### 2.2.2 Approches par apparences :

Cette approche ne nécessite pas de phase d'apprentissage, et ne fait aucune prédiction, l'idée est de faire une comparaison au cours du temps entre des points ou des régions se basant sur différentes informations les décrivant, telles que la couleur [28], la texture [29]et la forme. Voici deux méthodes populaires qui adoptent cette approche.

**Corrélation de points (Cross-correlation).** Dans un premier temps cette méthode consiste à trouver les points intéressants à suivre dans l'image de référence. Puis les points correspondants sont identifiés dans l'image courante. On prend le point intéressant de l'image de référence et ses voisins pour rechercher la meilleure correspondance à ce groupe de points dans l'image courante en se basant sur la couleur. Enfin, les vecteurs de mouvement sont définis entre la première position du point et sa

nouvelle position [30]. On peut aussi sélectionner une région dans l'image de référence [31] et la rechercher dans une image donnée. Pour retrouver la meilleure correspondance, une convolution de la région recherchée est faite sur les points de l'image de recherche (Figure 2.4) en appliquant l'équation 2.3, puis en prenant le point qui donne le meilleur score c'est-à-dire la plus grande valeur; ce point représente le centre de la nouvelle position de la région recherché.

$$G[x, y] = \sum_{i=-w/2}^{w/2} \sum_{j=-h/2}^{h/2} F[x + i, y + j] \times R[i, j] \qquad (2.3)$$

Où R représente la région recherchée, F l'image de recherche et G la matrice contenant les scores.

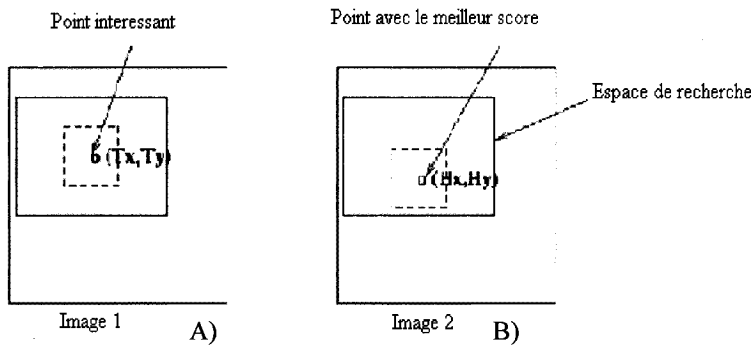
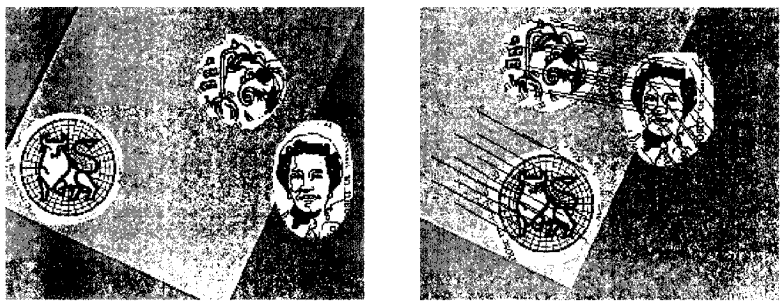



**Figure 2.4 [30] :** A) Image originale, B) Image de recherche

Si le nombre de points intéressants est grand, cette méthode prend beaucoup de temps à l'exécution puisque pour chaque point elle fait une convolution du filtre sur la nouvelle image pour rechercher la correspondance. Un autre problème rencontré par cette méthode est qu'elle ne fonctionne pas pour des objets déformables ou quand le point de vue est changé.

La figure 2.4 [8] montre les vecteurs de mouvement pour trois objets en mouvement.



A) B)

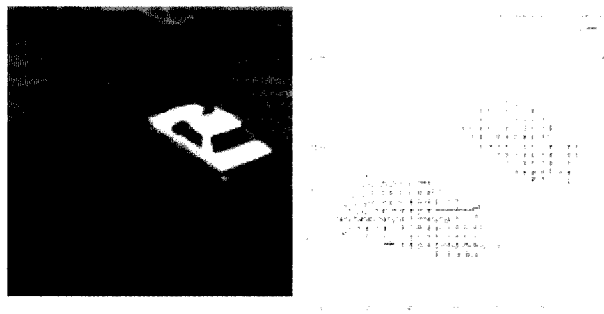
**Figure 1.5** [30] :A) Image originale

B) Image avec vecteur de déplacements.

**Flux optique.** Un vecteur de vélocité est calculé pour chaque pixel entre deux images en supposant que son intensité ne change pas et que son déplacement est petit. Ce vecteur représente le champ qui décrit la vitesse et la direction du déplacement produit dans l'image par les objets en mouvement. Il est estimé en analysant la relation entre les variations temporelles de l'intensité de l'image ou la distribution de fréquence spatio-temporelle dans le domaine de fourrier [32].

Ainsi, le flux optique permet de combiner la détection et le suivi des objets en mouvement en utilisant les intensités des pixels pour retourner l'information sur leurs vitesses et leurs directions de mouvement.

Par contre, les algorithmes de flux optique ne sont pas utilisés pour la surveillance en temps réel par des caméras fixes, car ils ne prennent pas en compte toutes les informations pertinentes sur les objets suivis pour en faire la distinction, et ils sont complexes à implanter.



A) B)

**Figure 2.6** : A) Image originale

B) Image avec vecteur de flux optiques.

**Par intersection de région.** Fuentes et al. [33] ont proposé un algorithme qui fait la correspondance entre les objets de l'image courante et ceux de l'image précédente (et vice versa) seulement en comparant les positions de leurs rectangles englobants. Puis deux matrices sont construites contenant les informations identifiant les rectangles englobants qui se sont croisés d'une image à l'autre. Enfin, une interprétation des évènements est faite à partir de l'analyse de ces matrices se basant sur différentes hypothèses.

Cette méthode performe bien pour faire le suivi d'une image à une autre d'un objet. Cela dit, la principale lacune de cette méthode est qu'elle n'utilise pas d'informations robustes pour décrire les objets suivis. Ainsi, elle ne peut faire le suivi s'il n'y a pas d'intersection entre les rectangles englobants.

Donc, l'approche par apparences est particulièrement intéressante pour faire une comparaison entre les objets détectés d'une image à une autre de la séquence. Cela dit il faut utiliser des informations plus robustes sur les objets à suivre telles que la couleur et la texture. En plus, il faut trouver un moyen pour faire la comparaison sans que ça soit couteux en termes de complexité et s'assurer que des changements de point de vue ou de luminosité n'affectent pas le suivi des objets.

Par ce que nous nous intéressons à l'apparence des objets, les informations sur leur couleur et texture sont particulièrement pertinentes.

## 2.3 Approches par apparences: couleurs

La perception des couleurs est très importante pour l'humain. Il utilise l'information de la couleur, entre autres pour distinguer entre les objets, places, et le temps de la journée. De nos jours, l'utilisation de couleurs par les machines est devenue courante avec les caméras couleurs, les télévisions couleurs et les logiciels qui traitent les images couleurs. Ce qui permet aux machines d'utiliser l'information de la couleur pour les mêmes raisons que l'humain. Plus encore, les machines peuvent voir plus que les

humains puisque le champ spectral de l'humain se limite entre 400 nm et 700 nm alors qu'elles peuvent utiliser aussi, entre autres, l'infrarouge et les rayons X.

Un moyen efficace et populaire pour étudier la couleur dans des images est l'histogramme de couleur [34]. L'utilisation de l'histogramme implique différentes étapes, la quantification de l'espace de couleur choisi en différents intervalles, mettre chaque pixel de l'image dans l'intervalle correspondant, et calculer la distance entre deux histogrammes pour comparer leur similarité.

### 2.3.1 Histogrammes de couleurs :

L'histogramme de couleurs compte le nombre de fois qu'un pixel d'une couleur spécifique est présent dans une image. Il est généralement utilisé dans la recherche d'image par le contenu et la reconnaissance d'objets [30, 31]. Sa force réside dans le fait qu'il est invariant à la translation, la rotation et la déformation de l'objet étudié [32,33].

Soit I l'image de largeur W et longueur L, le vecteur H qui représente l'histogramme est défini comme suit :

$$H(i)=\left|\left\{(x,y)\in N^2, x\prec W, y\prec L \,\middle|\, I(x,y)=i\right\}\right| \qquad (2.4)$$

### 2.3.2 Comparaison des Histogrammes :

Pour savoir si deux images ont la même distribution de couleurs, il suffit de comparer leurs histogrammes de couleurs. Il existe différentes mesures de distance pour mesurer la similarité entre deux histogrammes. Le plus grand est la distance, le moins similaires

sont les histogrammes. Celle qui est la plus utilisée pour la comparaison entre les histogrammes est la distance $L_1$, elle est simple à implanter et robuste à la fois. Par contre, elle ne permet pas de connaître la distribution des différences entre les deux histogrammes, ce qui dans notre cas ne pause pas de problème majeur puisque nous ne cherchons pas à savoir où se trouve l'erreur.

La formule suivante est utilisée pour calculer la distance $L_1$:

$$L_1 = \sum_i H_a(i) - H_b(i) \tag{2.5}$$

Huang et al. [39] ont introduit une nouvelle mesure $D_1$ dérivée de $L_1$ qu'ils ont prouvé être plus fiable.

Par exemple, Si on considère deux paires d'images $I_1$, $I_2$ et $\grave{I}_1$, $\grave{I}_2$ dont la valeur des histogrammes pour une couleur donnée a sont $h_{I1}(a) = 1000$, $h_{I2}(a)=1050$, $h_{\grave{I}1}(a)= 100$, $h_{\grave{I}2}(a)=150$. Même si la différence entre chaque paire d'histogrammes est 50, la différence est nettement plus significative pour la deuxième paire d'images.

Donc, il faudrait donner plus d'importance à la différence $|H_a(i)-H_b(i)|$ à l'équation 2.5, quand $H_a(i)+H_b(i)$ est petite et vice-versa. Donc la formule pour calculer la distance devient :

$$D_1 = \sum_i \frac{|H_a(i) - H_b(i)|}{1 + H_a(i) + H_b(i)} \tag{2.6}$$

L'addition du 1 dans le dénominateur est pour éviter la division par 0.

### 2.3.3 Intersection des histogrammes :

Pour savoir si la distribution de couleur d'une image est incluse dans la distribution de couleurs d'une autre image, Swain et Ballard [40] ont introduit l'intersection des histogrammes.

Pour deux objets donnés I, et M dont les histogrammes sont $H_i$, $H_m$. L'histogramme HI représentant leur intersection est :

$$HI(i) = \min(H_i(i), H_m(i)) \tag{2.7}$$

Maintenant pour savoir si les couleurs de l'image M sont contenus dans I, il suffit de calculer la distance entre les deux histogrammes HI et $H_m$. Si la distance est approximativement nulle alors l'objet M fait partie de l'objet I.

**Exemple :**

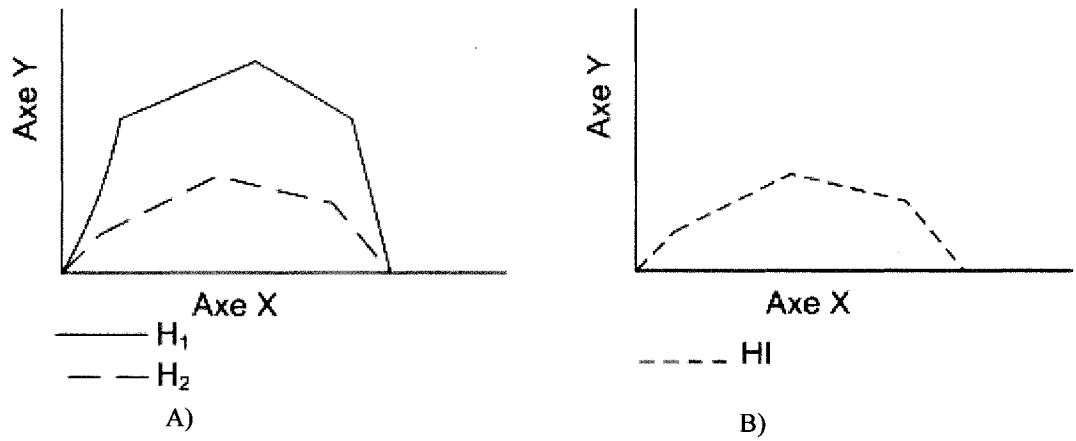



**Figure 2.7** : Représentation de l'intersection des histogrammes.

La Figure 2.6.A représente les histogrammes $H_1$, $H_2$ pour deux images différentes. La Figure 2.6.B représente l'intersection de ces deux histogrammes HI. Nous remarquons que HI est similaire à $H_2$, ceci est dû au fait que $H_2$ est inclus dans $H_1$ ce qui veut dire que la distribution des couleurs de l'image représentée par $H_2$ fait partie de la distribution des couleurs de l'image représentée par $H_1$.

Avant de construire l'histogramme de couleur il faut choisir dans quel espace de couleur travailler. Un descriptif des espaces de couleurs les plus utilisés est présenté à l'annexe I.

## 2.4 Approches par apparences: texture

La texture est une autre information qui peut être utilisée pour caractériser une région ou un objet. Elle représente une répétition spatiale d'un même motif dans différentes directions de l'espace. Dans notre cas nous allons utiliser cette information pour caractériser les objets détectés.

Il existe différentes méthodes populaires pour étudier la texture. Rosenfeld et al. [41] ont introduit le concept de la densité d'arêtes qui consiste à mesurer le nombre d'arêtes par unité de surface. Les textures fines tendent à avoir une densité plus élevée d'arêtes que les textures plus grossières. Cependant, dans notre application les objets étudiés n'ont pas forcément des textures générant beaucoup d'arêtes et cette méthode ne tient pas compte des différences de couleur des textures.

Une autre méthode pour décrire la texture est la banque de filtres [42] qui consiste à faire la convolution de l'image avec différents filtres, chacun permettant d'extraire une propriété différente. Cela dit, cette méthode consomme beaucoup de temps à l'exécution et les filtres utilisés doivent être adaptés selon les textures à étudier. Cette méthode ne tient pas compte des différences de couleur des textures.

Une structure efficace pour notre application qui permet de décrire la texture est le corrélogramme qui a été introduit par Huang et al. [39].

### 2.4.1 Corrélogrammes :

Contrairement à l'histogramme de couleurs qui capture seulement la distribution des couleurs dans l'image, le corrélogramme nous donne l'information sur la distribution spatiale de ces couleurs. Pour comprendre le corrélogramme, il faut d'abord expliquer la matrice de cooccurrence introduite par Haralick et al.[43] et qui en est l'origine.

Une matrice de cooccurrence Cv donne la relation spatiale entre les niveaux de gris dans l'image. C'est une matrice à deux dimensions où $C_v$ (i,j) indique le nombre de fois que le niveau de gris i co-occurre avec le niveau de gris j selon un vecteur de distance donné $V(d_x, d_y)$, où $d_x$ et $d_y$ représentent le déplacement en lignes et colonnes respectivement.

Soit I l'image de largeur W et de longueur L, la matrice de co-occurrence est définie comme suit :

$$C(i,j) = \left| \left\{ (x,y) \in N^2, x \prec W, y \prec L \mid I(x,y) = i \wedge I(x+dx, y+dy) = j \right\} \right| \qquad (2\ 8)$$

Dans l'exemple ci-dessous, trois matrices de cooccurrences sont construites pour trois vecteurs de distances différents.

La matrice de cooccurrence est utilisée pour les images de niveaux de gris. Pour les images couleurs c'est le corrélogramme qui est utilisé. Pour le corrélogramme, i et j dans l'équation 2.7 représente des couleurs au lieu de niveau de gris.

Il a été prouvé que les corrélogrammes sont plus efficaces que les matrices de cooccurrences pour la recherche des images par le contenu [44], car ils tiennent compte aussi de la couleur.

**Exemple :**

| | Vecteur de distance C[dx,dy] | Résultats |
|---|---|---|
| Image originale<br>1 1 0 0<br>1 1 0 0<br>0 0 2 2<br>0 0 2 2 | i<br>j<br>$C_{[1,2]}$ | 0 0 2<br>2 0 2<br>0 0 0 |
| | i<br>j<br>$C_{[2,2]}$ | 0 0 0<br>0 0 4<br>0 0 0 |
| | i<br>j<br>$C_{[2,3]}$ | 0 0 0<br>0 0 2<br>0 0 0 |

### 2.4.2 Comparaison des corrélogrammes :

Pour savoir si deux images/objets ont la même texture, il suffit de comparer leurs corrélogrammes de couleurs. A noter, que les mêmes mesures de distances que pour les histogrammes s'appliquent pour les corrélogrammes.

Huang et al. [39] ont proposé l'utilisation de la mesure $D_1$ proposée à l'équation 2.6 mais étendue pour des matrices à deux dimensions :

$$D1 = \sum_i \sum_j \frac{|C_a(i,j) - C_b(i,j)|}{1 + C_a(i,j) + C_b(i,j)} \qquad (2.9)$$

### 2.4.3 Intersection des corrélogrammes :

Pour savoir si la texture d'une image fait partie de la texture d'une autre image, Huang et al. [39] ont proposé d'étendre l'intersection des histogrammes à l'intersection des corrélogrammes.

Pour deux objets donnés I, et M dont les corrélogrammes sont $C_i$ $C_m$. Le corrélogrammes CI représentant leur l'intersection est :

$$CI(i,j) = \min(C_m(i,j), C_i(i,j)) \qquad (2.10)$$

Maintenant pour savoir si la texture de M est contenue dans I, il suffit de calculer la distance $D_1$ entre les deux corrélogrammes CI et $C_m$. Si la distance est approximativement nulle alors la texture de l'objet M fait partie de l'objet I.

## 2.5 Discussion

Il est clair que dans la télésurveillance, il reste beaucoup de développement à faire, notamment pour les algorithmes de détection où des approches plus générales ne se limitant pas à l'étude de l'image pixel par pixel, comme la méthode proposée par Heikkila et Pietikäinen [15], devraient être développés pour rendre la détection plus robuste au bruit dans l'image. Nos tests sur les méthodes pixel par pixel ont montré que celles-ci ne sont pas très robustes. Pour cette raison, nous avons choisi de développer une approche qui tient compte du voisinage des pixels.

Pour ce qui est du suivi, les approches se basant sur l'apparence des objets ont prouvé leur efficacité pour identifier les objets d'une image à une autre dans les cas, comme ceux qui nous intéressent, où le mouvement est difficilement prédictible. Pour cette raison, nous optons pour l'utilisation de méthodes par apparences, c'est-à-dire les histogrammes et les corrélogrammes.

# CHAPITRE 3
# MÉTHODOLOGIE

L'approche proposée (figure 3.1) consiste dans un premier temps à faire la détection des objets en mouvement en étudiant la distribution des couleurs dans l'image par blocs de régions pour déterminer où des changements se sont effectués et en répétant cette procédure en divisant les blocs itérativement. Les histogrammes de couleurs sont utilisés pour étudier les distributions de couleurs.

Dans un deuxième temps, une approche par apparences est utilisée pour le suivi. Un modèle pour chaque objet détecté est créé. Ce modèle contient l'information sur la couleur, la texture et la taille. Ces informations sont intégrées dans un arbre de décision qui permet de faire la comparaison entre deux objets donnés pour savoir s'ils sont similaires ou si l'un fait partie de l'autre. La texture est étudiée en utilisant les corrélogrammes.

Enfin, une interprétation des actions réalisées dans la scène est effectuée en se basant sur l'analyse d'hypothèses en utilisant les résultats du suivi jumelés aux mouvements des centroïdes des objets détectés. A ce stade le système se limite à interpréter les cas où des objets fusionnent ou se séparent à un moment donné dans l'image.

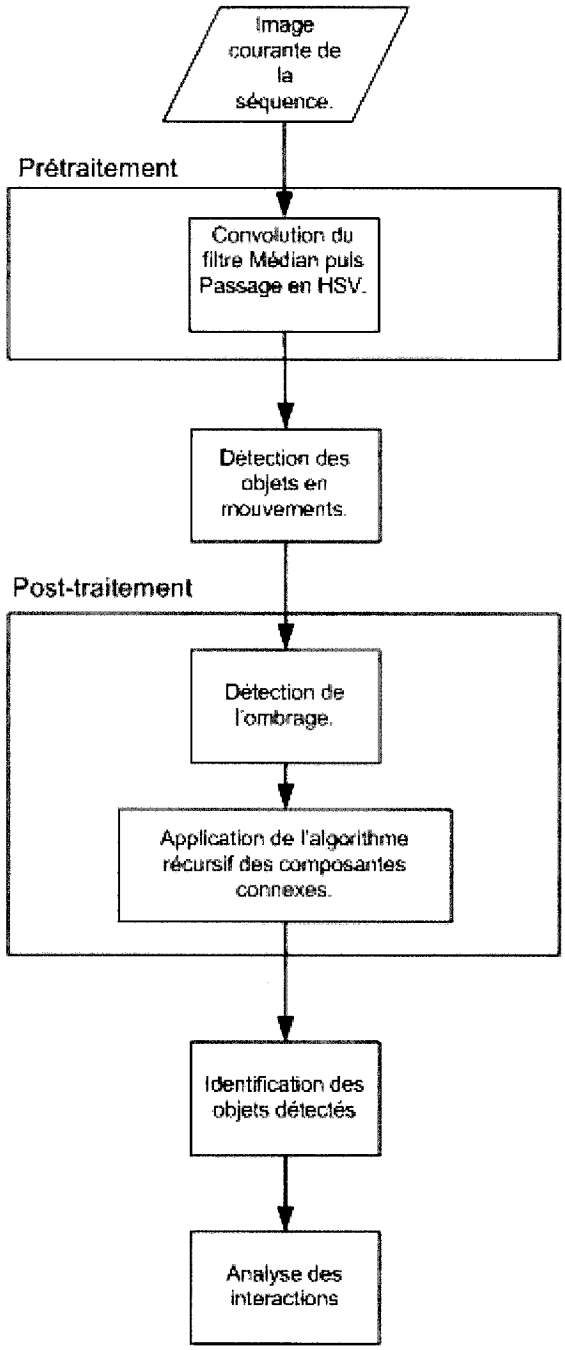



**Figure 3.1** : Schéma représentant la méthodologie suivi.

### 3.1 Prétraitement

Avant de traiter l'image nous avons besoin de réduire le bruit dû généralement à l'acquisition. Pour ce faire nous utilisons le filtre médian. Il a pour but de supprimer les bruits impulsionnels dus à la capture dans une image. Par bruit impulsionnel, on désigne des points qui sont présents dans l'image et qui sont très différents de leurs voisins alors qu'ils devraient avoir la même valeur. Ce filtre est particulièrement utile parce qu'il nettoie le bruit sur les surfaces uniformes en préservant les bordures des objets [45].

La *médiane* est une mesure statistique. Pour un tableau de réel T contenant n valeurs, la médiane est la valeur du tableau ordonné qui se trouve à la position n/2 si n est paire et à (n/2)-1 si n est impaire.

Le filtre consiste à prendre le point que l'on considère avec ses voisins, ensuite on trie tous les points en fonction de leurs valeurs et on prend le point médian comme point résultant du filtrage. Pour une image RGB, une convolution de ce filtre est faite sur toute l'image pour les trois composantes du pixel.

Le filtre médian qui est utilisé prend en compte les voisins directs du pixel considéré. C'est-à-dire que le filtrage se fait par groupe de 9 valeurs. Ceci permet de nettoyer les pixels bruités sans trop altérer les informations de l'image telle que la texture.

Après l'application du filtre médian, l'espace de couleur est converti de RGB vers HSV pour la suite du traitement de la séquence. La raison principale de l'utilité de

HSV est qu''il nous permet de contrôler l'importance de la lumière dans la scène. Une quantification des trois composantes H, S et V en intervalles est faite de telle sorte à donner moins d'importance à V (Intensité) et à donner plus d'importance à H (Teinte) qui contient l'information sur les couleurs dans la scène, ainsi plus d'importance est donnée à l'information pertinente sur la couleur.

Au départ, les intervalles pour HSV résultant de la transformation de l'espace RGB faite par OpenCV (voir annexe II), sont [0, 180] pour H, [0, 255] pour S et [0, 255] pour V. Après avoir effectué la quantification, les intervalles deviennent [0, a] pour H, [0, b] pour S et [0, c] pour V, où a, b et c sont des entiers à fixer selon la quantification voulue.

Cette quantification rend l'algorithme plus robuste en diminuant sa dépendance à l'intensité de la lumière dans la scène tout en accélérant son exécution par l'utilisation d'intervalles avec des tailles réduites (255 X 255 X 180) à (a X b X c) [46]. La méthode utilisée pour quantifier l'espace de couleur est la suivante :

Soit a, b, c les trois valeurs recherchées et H, S, et V les valeurs à quantifier et soit tmp un entier utilisé pour les calculs.

tmp = quotient de la division de H par (180 / a)

a = tmp X (180/H)

tmp = quotient de la division de S par (255 / b)

b = tmp X (255/S)

tmp = quotient de la division de V par (255 / c)

c = tmp X (255/V)

### 3.2 Détection d'objets en mouvement dans une scène

La plupart des approches existantes étudient l'image pixel par pixel. Notre approche consiste à l'étudier par blocs de régions pour que le bruit qui subsiste au prétraitement ou les petits mouvements inintéressants, soient filtrés automatiquement à la détection. En prenant un bloc de pixels, les quelques pixels bruités ayant été négligés par le filtre médian du bloc auront moins d'importance lors de la comparaison avec un autre bloc de pixels car le changement apporté par ce bruit est négligeable relativement à la taille du bloc, et les oscillations des feuilles d'arbres ou de l'eau seront filtrées automatiquement puisqu'en oscillant dans le même bloc il n'y aura pas de changement significatif de la couleur dans le bloc de région.

L'idée est de diviser itérativement l'image en régions de taille similaires. Nous avons choisi de mettre les régions en forme de carrés pour faciliter le traitement. A chaque étape, l'histogramme de couleur de l'image de référence et de l'image courante est généré pour chaque région carrée puis normalisé pour que le changement de l'intensité de lumière d'une scène à une autre n'affecte pas le seuil fixé. La taille de

l'histogramme dépend de la quantification choisie. Puis comme indiqué à la section 2.3.2, la distance $D_1$ est utilisée pour calculer la distance entre les deux histogrammes et ainsi permettre de faire la comparaison entre chaque paire de régions de $H_{ref}$ et $H_{cour}$.

Si $D_1$ est plus grand qu'un seuil spécifié, alors les deux régions sont différentes et ainsi on considère qu'il y a un mouvement dans cette région. Le seuil est fixé comme un pourcentage de la taille du carré selon le niveau de changement qu'on veut détecter.

Expérimentalement ces valeurs ont été fixées ainsi:

$Th_{i+1} = Th_i + 0.10$ avec $Th_0 = 0.10$.

À la première division de l'image en blocs de tailles NxN le seuil est $Th_0$. Puis, pour chaque division i, on augmente le seuil $Th_i$. En augmentant le seuil, on est plus exigeant et on cherche à faire de moins en moins de faux positifs par subdivision puisqu'on considère seulement les plus grands changements à l'intérieure d'un carré.

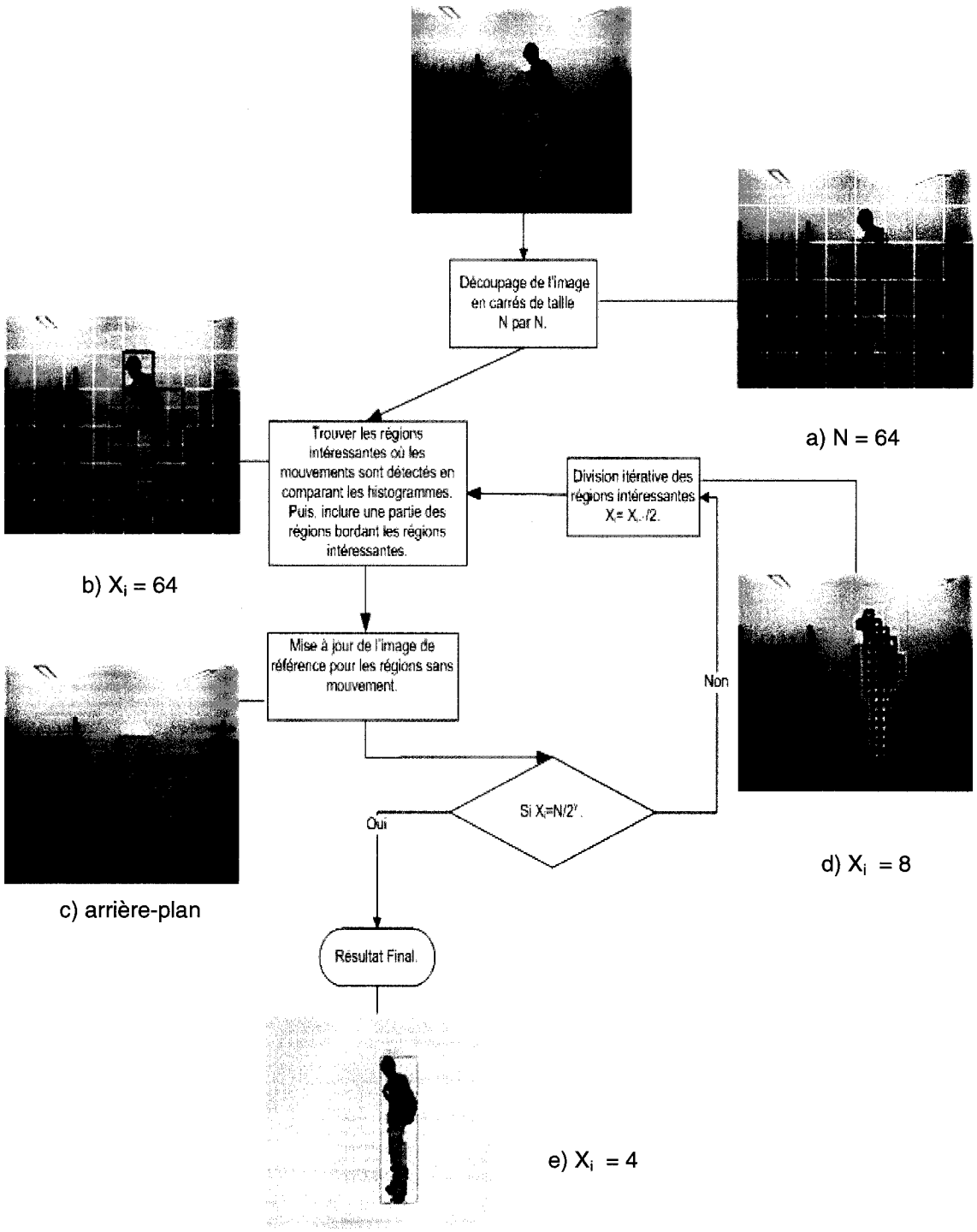



**Figure 3.2** : Schéma représentant l'algorithme de détection d'objets en mouvement.

Les étapes de la détection d'objets en mouvement présentées dans le schéma (figure 3.2) sont détaillées ci-dessous :

Dans un premier temps, l'image est coupée en carrés de taille N par N; la valeur de N dépend de la taille de l'objet qu'on veut traquer. Le plus grand est l'objet relativement à l'image, le plus grand est la valeur de N et vice-versa. (Figure 3.2.a). L'histogramme de chaque région de l'image de référence et de l'image courante est généré en utilisant la quantification présentée précédemment. Puis la distance $D_1$ est calculée entre chaque paire d'histogrammes. Si selon $D_1$ et le seuil $Th_i$ les régions carrées sont similaires, c'est-à-dire $D_1(H_{ref}(i), H_{cour}(i))< Th_i$, alors aucun mouvement n'est détecté dans cette région là. Sinon on considère qu'un mouvement s'est effectué et on marque cette région comme une région qui doit être divisée plus tard.

Dans un deuxième temps, les régions identifiées comme intéressantes (carrés foncés à la figure 3.3.a) sont divisées en quatre petits carrés (carrés à l'intérieur des carrés foncés). C'est-à-dire $X_i= X_{i-1}/2$ avec $X_0=N$. Pour avoir une segmentation plus précise des objets en mouvement. Les extrémités des objets risquent d'être perdues car elles ne sont pas assez significatives pour être détectées dans les blocs voisins. Pour préserver ces petites extrémités (Le pied de la personne dans la Figure 3.3.a) nous divisons en quatre aussi les régions voisines extérieures puis les voisins selon un voisinage de huit sont inclus pour l'étape suivante (carré en dehors des carrés foncés dans la Figure 3.3.a).

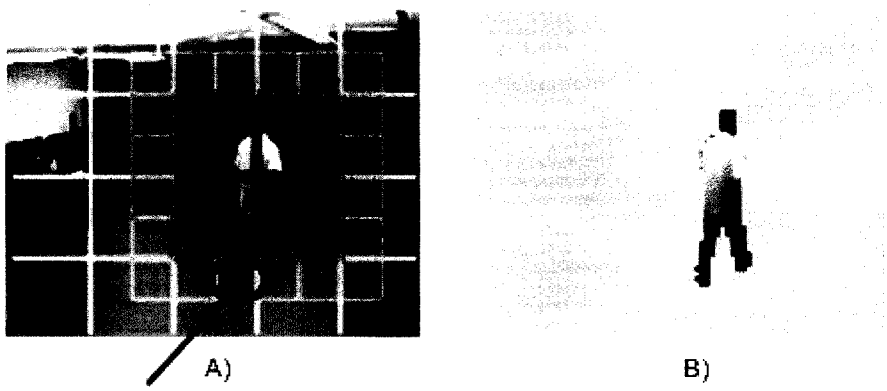



**Figure 3.3** : Extrémité qui déborde des régions intéressantes

L'image de référence est mise à jour avec le contenu des régions où aucun mouvement n'est détecté. Quand un objet bouge dans la scène, pour un carré donné, l'objet prend de plus en plus d'importance en termes de surface par rapport à la surface du carré. Donc, tant qu'il occupe une surface trop faible à l'intérieur d'un carré, il ne sera pas détecté. Ainsi pour ne pas mettre des parties d'objets dans l'image de référence, nous appliquons un détecteur d'ombrage (voir section 3.3.1) sur tout le reste de l'image, les pixels considérés comme de l'ombre (C'est-à-dire, comme non objets) sont mises à jour. Pour le reste des pixels, ils appartiennent soit à des parties d'objets ou bien c'est simplement du bruit dans l'image dont la position vari d'une image à une autre et qui ne doit pas être inclus dans notre mise à jour d'arrière-plan. Grâce à cette opération, dans les scènes où un changement graduel de lumière s'effectue, l'algorithme ne sera pas perturbé entre un moment où la lumière est intense et un moment où elle est faible puisque ces changements de lumière sont pris en considération continuellement dans le temps. Cela devrait permettre au système de performer avec fiabilité dans des scènes extérieures à supposer qu'un changement brutal de luminosité ne se produise pas.

Dans la troisième phase, on vérifie si on a atteint le nombre de divisions voulues pour $X_i=N/2^y$ où *y* est le seuil fixé selon le degré de précision recherché, sinon on répète l'étape précédente, La figure 3.2.d illustre l'itération finale de l'algorithme pour N =64 et y =16. L'algorithme est exécuté sur l'image jusqu'à ce que la taille des carrés soit égale à 4X4. Le résultat final de l'algorithme est présenté à la figure 3.2.e L'arrière-plan est mis en gris.

Comparé aux algorithmes standards de soustraction d'arrière-plans, l'algorithme proposé ne nécessite pas de modèle statistique. Il n'a pas besoin de phase d'apprentissage pour s'adapter à chaque scène où il est appliqué. Une fois la taille des carrés N et $Th_0$ fixés selon la taille des objets à détecter, il peut être appliqué pour détecter les objets en mouvement.

### 3.3 Post-Traitement

Après la segmentation des objets détectés, certaines régions sont faussement détectées comme appartenant à l'avant-plan. Cela est dû dans la plupart des cas à l'ombre des objets en mouvement.

#### 3.3.1 Détection de l'ombre :

De nombreux travaux ont été faits pour traiter ce problème de l'ombrage. Nous avons décidé d'utiliser une variante de l'algorithme proposé par Cucchiara et al. [47]. Cet algorithme est simple à implanter et efficace. Il est appliqué sous l'espace HSV à l'image originale avant la quantification et consiste à itérer sur chaque pixel des objets

détectés par l'algorithme de détection et décider s'il appartient à l'ombre d'un objet ou non en utilisant certains critères de comparaisons par rapport à l'image de référence. La méthode considère que la teinte (H) et la saturation (S) varient dans un intervalle limité, et que la luminosité (V) courante devrait être un pourcentage de la luminosité de l'image de référence compris dans un intervalle limité aussi. Si ce pixel appartient à l'ombrage alors il est considéré comme appartenant à l'arrière-plan. Dans notre cas on fixe l'intervalle relativement aux valeurs maximales entre l'image de référence et l'image courante de chaque composante en pourcentage, ce qui rend l'algorithme plus général pour fonctionner à différentes intensités de lumière.

L'équation suivante représente le test effectué pour chaque pixel de l'image; si le pixel est de l'ombre alors il est mis à 1 (blanc), sinon il est mis à 0 (gris):

$$F_{cour}(p)=\begin{cases}1 & \text{si } \dfrac{(V_{cour}-V_{réf})}{\max(V_{cour},V_{réf})}\le t_v \wedge \dfrac{(S_{cour}-S_{réf})}{\max(S_{cour},S_{réf})}\le t_s \wedge \dfrac{(H_{cour}-H_{réf})}{\max(H_{cour},H_{réf})}\le t_h \\ 0 & \text{sinon}\end{cases} \tag{3.1}$$

Avec $t_v$ , $t_s$ ,. $t_h$ $\in[0,1]$ et qui sont des paramètres fixés expérimentalement. Une fois fixés ces paramètres sont les mêmes pour toutes les régions de l'image.

### 3.3.2 Algorithme récursif des composantes connexes:

Après avoir séparé l'ombrage des objets, il faut les étiqueter. L'algorithme récursif des composantes connexes (recursive connected component) [48] est appliqué sur l'image, il consiste à marquer chaque groupe de pixel connecté comme étant un objet indépendant

en utilisant un voisinage de huit.

À noter que des faux objets positifs de petites tailles peuvent être détectés. Aussi, après l'étiquetage, les régions de tailles négligeables sont tout simplement ignorées dans la suite du traitement en les enlevant de la liste des objets en mouvement, les autres objets sont considérés chacun comme étant un objet indépendant en mouvement dans la scène. La taille des objets à négliger est à fixer par l'utilisateur selon la scène et la taille des objets à détecter. Enfin, chaque objet détecté est mis dans un rectangle englobant (Bounding box) qui délimite ses frontières.

### 3.4 Suivi des objets détectés :

Maintenant que les objets en mouvement sont détectés, nous essayons de les retrouver dans l'image suivante. Ainsi, nous allons suivre les objets détectés. Pour ce faire nous avons besoin d'informations fiables pour les décrire. Parce que les objets en question sont généralement déformables comme les humains par exemple, on a besoin d'informations qui ne soient pas trop altérées par la déformation des objets d'une image à l'autre. Ce qui écarte l'utilisation des moments d'inertie par exemple. Cela dit ces informations peuvent être intégrées dans des travaux futurs pour reconnaitre des objets rigides transportés ou déposés.

La distribution de couleurs est une bonne caractéristique pour distinguer deux objets puisqu'elle est invariante à la déformation, la translation et à la rotation [32,33]. Cela dit, deux objets différents ont souvent la même distribution de couleur. C'est pour

cela que nous incluons la modélisation de la texture. Nous avons choisi la modélisation de la texture comme la deuxième composante parce qu'elle nous donne plus d'information pour faire la distinction entre les objets.

Ainsi, pour le suivi un arbre de décision est utilisé pour comparer deux objets donnés, un détecté dans l'image précédente et l'autre détecté à l'image courante qui se trouvent dans la même zone de l'image; cette zone est fixée par une distance entre les deux objets. Dans un premier temps leurs distributions de couleurs sont comparées en utilisant les histogrammes, puis leurs distributions spatiales de couleurs sont comparées en utilisant les corrélogrammes, ici on suppose que la couleur ne change pas brusquement d'une image à une autre. Pour faire les comparaisons, les intersections entres les histogrammes et entre les corrélogrammes sont calculées (voir section 2.3.3 et 2.4.3). Et enfin les tailles des objets sont comparées.

Les corrélogrammes sont plus complexes à calculer que les histogrammes, c'est pour cela que les histogrammes sont calculés en premier. Dans le cas où les objets n'ont pas la même couleur, il n'est pas nécessaire de comparer leurs textures.
La figure 3.4 illustre l'arbre de décision.

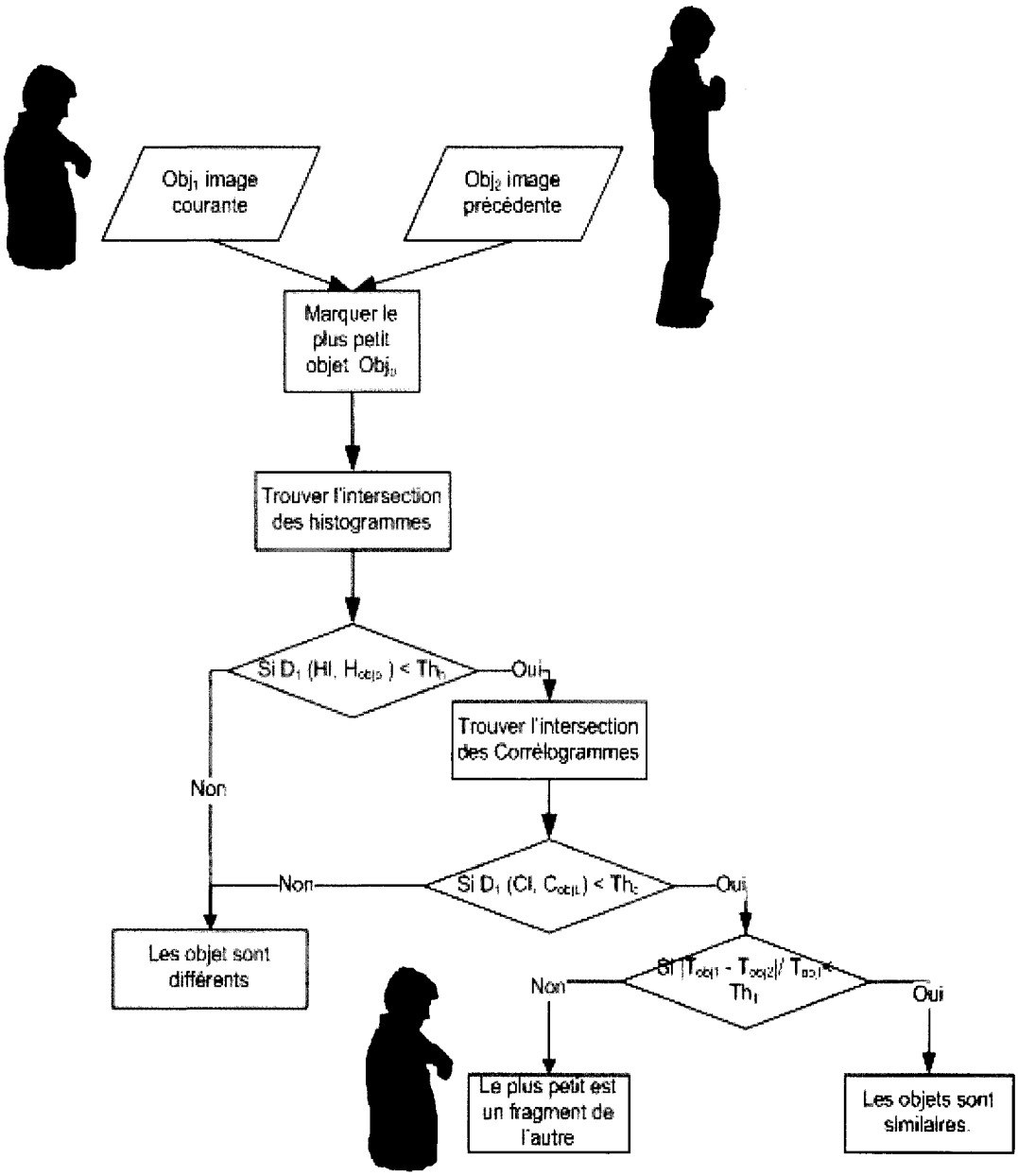



**Figure 3.4** : Schéma de l'algorithme de suivi.

Voici la description détaillée de l'algorithme :

Pour chaque objet détecté dans la fenêtre courante,

Dans un premier temps l'histogramme de couleur de l'objet de l'image courante est calculé. Soit le plus petit objet $Obj_p$ et l'autre $Obj_o$.

Puis, l'intersection des histogrammes HI (voir section 2.3.3) de couleur pour les deux objets est calculée. Si la distance $D_1$ entre $H_p$ et HI est plus grand qu'un seuil fixé, c.-à-d. D1 (HI, $H_p$) < $S_h$, alors $Obj_p$ et $Obj_o$ sont différents. On reprend le processus, en prenant un autre objet de l'image précédente. Sinon, les corrélogrammes des deux objets et l'intersection des corrélogrammes CI des deux objets sont calculées.

Si la distance D1 entre CI et Cp est plus grande qu'un seuil fixé, c.-à-d. D1 (CI, Cp) < $S_c$, alors les objets sont différents On reprend le processus, en prenant un autre objet de l'image précédente. Sinon, on compare les tailles des deux objets, $|T_{obj2} - T_{obj1}|/ T_{Max} < S_t$, nous supposons que la taille de l'objet ne change pas plus que 15% entre deux image adjacentes, cette supposition nous permet de distinguer entre la situation ou deux objets sont les mêmes et celle où l'un fait partie de l'autre, car dans les deux cas $D_1$ (CI, Cp) sera très petite. Ainsi, si un objet est nettement plus petit que l'autre alors il est considéré comme son fragment pouvant être résultant de l'occlusion d'une partie de l'objet, d'une fusion de deux objets ou de la séparation entre les deux objets comme on va le voir dans la section suivante, sinon les deux objets sont considérés comme étant similaires.

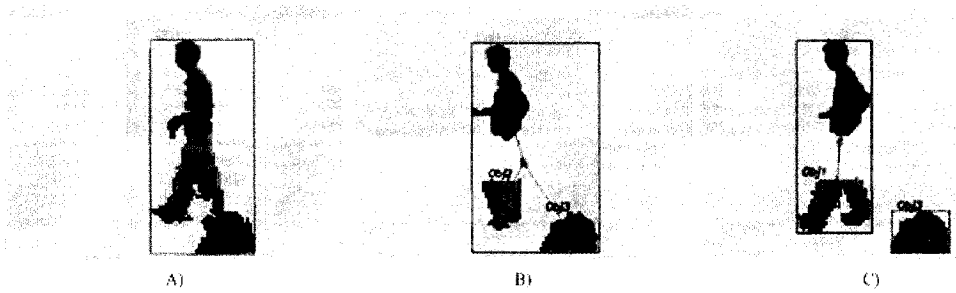


**Figure 3.5** : un objet est déposé dans la scène

À noter que les seuils pour comparer les distances $D_1$ entre les histogrammes et les corrélogrammes ont été fixés expérimentalement. Comme ces structures sont normalisées, le changement de scène ou d'éclairage n'altère pas la comparaison.

### 3.5 Raffinement du suivi

Dans cette phase, on essaye de raffiner les résultats concernant les objets suivis. On introduit une nouvelle information sur ces objets, qui est leur centroïde. Ainsi, quand un objet est fragmenté en plusieurs morceaux dans une partie de la séquence alors l'analyse de texture et de couleur jumelée avec l'information sur le centroïde, permettront de déterminer si les fragments appartiennent au même objet. Ceci, permettra éventuellement entre autres de savoir si un objet a été laissé par l'humain ou la séparation est due à une occlusion ou une mauvaise détection.

Ainsi, quand un objet est fragmenté à un moment donné à cause d'une séparation due à une occlusion ou un dépôt, l'algorithme de suivi est capable de nous dire que les fragments appartiennent au même objet présent précédemment dans la scène. Cela est

possible grâce aux intersections des histogrammes et des corrélogrammes qui nous permettent de savoir si la couleur ou la texture du fragment est présente dans l'objet de l'image précédente. De la même manière l'algorithme détecte aussi quand des objets présents dans les images précédentes fusionnent pour former un seul fragment.

Donc, après avoir appliqué l'algorithme de suivi, trois situations différentes sont possibles:

1- Des fragments des objets ayant été présents précédemment sont identifiés, sans pour autant être regroupés.

2- Des objets présents précédemment sont retrouvés dans un même objet de l'image courante.

3- L'objet présent dans l'image précédente est retrouvé tel quel dans la nouvelle image.

Dans la première situation, les fragments d'un même objet sont groupés ensemble, et le centroïde du groupe de fragment est calculé. Puis la distance entre le centroïde de chaque fragment et le centroïde du groupe est calculée. Si la distance est plus grande qu'un seuil fixé alors nous concluons que ce fragment correspond à la séparation de deux objets, sinon le fragment fait toujours partie de l'objet et la séparation est peut être due à une occlusion partielle. À noter que le seuil de comparaison pour la distance est une mesure fixée par l'utilisateur.

Dans la deuxième situation si un des fragments de l'objet a été indépendant pour

une longue période dans les images précédentes, puis pour une certaine période a fusionné avec le reste des fragments alors nous considérons qu'il y a fusion d'objets.

Dans la troisième situation les informations sur l'objet sont simplement mises à jour. C'est-à-dire que sa texture, sa couleur et sa taille sont représentées par ses nouveaux corrélogrammes, histogrammes et nombre de pixels le composant.

# CHAPITRE 4

# RÉSULTATS ET DISCUSSION

Dans les chapitres précédents nous avons décrit les algorithmes pour faire la détection et le suivi des objets en mouvement développés dans le cadre de ce mémoire. Dans ce chapitre nous présentons quelques expérimentations qui valident chaque algorithme. L'application est implémentée en Visual C++, et utilise quelques méthodes de traitement d'images d'une librairie de code source libre OpenCV [49]. L'entrée à l'application est une séquence vidéo, et la sortie est une séquence vidéo contenant les objets segmentés et les rectangles englobants du suivi. Les tests ont été effectués sur un ordinateur 3.2 Ghz Intel Xeon(tm) à partir de séquences vidéo prises avec des caméras couleurs CCD.

Ce chapitre est organisé comme suit. La section 4.1 décrit les séquences utilisées pour valider les algorithmes développés, puis la section 4.2 présente les résultats de l'algorithme de détections des objets en mouvement. Enfin, la section 4.3 présente les résultats de l'algorithme de suivi.

## 4.1 Description des séquences utilisées :

Il est difficile de se comparer aux autres travaux existants puisque la plupart des chercheurs utilisent leurs propres séquences vidéo. Cela dit, il existe quelques bases de données dont Wallflower[51] et PETS [50], contenant des séquences vidéo utilisées par certaines équipes de recherche. Nous les avons utilisées pour comparer nos résultats. En plus de cela, nous avons utilisé des séquences vidéo maison prises dans le laboratoire du LITIV et dans l'atrium du Pavillon MacKay-Lassonde à Polytechnique (avec des caméras couleurs CCD Sony DFW-SX910 d'une résolution de 1280x960).

### 4.1.1 Séquences vidéo utilisées pour valider la détection :

En ce qui concerne les séquences Wallflower, elles ont toutes une résolution de 192X128, voici leur description :

*Objets bougés (séq. 1)* : Une personne marche dans une chambre, téléphone, puis quitte la chambre laissant le téléphone et la chaise dans une position différente. On évalue l'algorithme à l'image 50 après que la personne quitte la chambre.

*Heures de la journée (séq.2)* : Représente une chambre noire qui s'éclaircit graduellement. Une fois la chambre bien éclaircie, une personne entre dans la scène pour s'assoir sur le divan.

*Oscillation d'arbres (séq.3)*:Une personne rentre dans la scène illuminée où des arbres sont en mouvement.

*Camouflage (séq.4)* : Un moniteur est sur un bureau avec des barres d'interférences qui se produisent sur le moniteur. Une personne rentre dans la scène et cache le moniteur.

*Endroit publique (séq.5) :* La séquence consiste en une vue sur une cafétéria prise sur une longue période. Le mouvement est constant et chaque image de la séquence contient des personnes.

*Ouverture d'avant-plan (séq.6)* : Vue de l'arrière d'une personne qui est endormie sur son bureau. Elle se réveille est commence à bouger.

En ce qui concerne les séquences prises avec les caméras du laboratoire LITIV :

*Atrium* : Trois personnes marchent dans un atrium contenant différentes sources de lumières. La séquence contient 100 images. Pour accélérer le traitement, cette séquence est traitée aussi avec une résolution de 192X128.

### 4.1.2 Séquences vidéo utilisées pour valider le suivi :

En ce qui concerne la séquence PETS :

*À l'extérieur* : Séquence vidéo extraite de la base de donné PETS 2001 dataset 2 (caméra 2), elle représente une voiture qui roule sur une route en plein jour (100 images avec une résolution de 192X128)

En ce qui concerne les séquences prises avec les caméras du laboratoire LITIV :

*Dépôt de sac* : Une personne marche dans une salle contenant différente sources de lumières, dépose un sac puis revient le prendre.

*Dépôt de sac avec occlusion* : Un personne qui porte un pantalon avec la même couleur que le mur marche dans une salle contenant différentes sources de lumières, dépose un sac, puis revient le prendre.

*Croisement entre personnes* : Deux personnes marchent dans une salle contenant différentes sources de lumières, et se croisent à plusieurs reprises.

## 4.2 Validation de l'algorithme de détection :

L'algorithme de détection est principalement basé sur la couleur tout comme les algorithmes de détection les plus populaires. Pour une région donnée, il est possible de distinguer entre la partie de l'objet en mouvement et l'arrière-plan seulement s'il y a une différence de couleur significative.

### 4.2.1 Méthodologie :

Les images sont extraites dynamiquement des séquences et sont traitées une par une. La taille de ces images varie selon la vidéo étudiée comme indiqué à la section 4. Les paramètres utilisés pour chaque étape de l'algorithme ont été évalués et déterminés expérimentalement pour garder ceux qui donnent les meilleurs résultats. Sauf quand spécifié autrement, pour la détection, la taille des carrés au départ, N est fixé à 32 et $Th_0$ qui représente le premier seuil de détection, est fixé à 0.10.

L'évaluation qualitative est basée sur une interprétation visuelle, tandis que, l'évaluation quantitative est faite en termes de faux négatifs (le nombre de pixels

appartenant à l'avant-plan qui ont été mal classés) et les faux positifs (le nombre de pixels appartenant à l'arrière-plan qui ont été détectés comme appartenant à l'avant-plan). La segmentation idéale de l'avant-plan est faite manuellement pour une image sélectionnée de la séquence. Le nombre de faux positifs et de faux négatifs est évalué en comparant la segmentation idéale de l'avant-plan avec celle trouvée par notre algorithme. Pour chaque image, le nombre de carrés de tailles 4X4 représentant les faux négatifs et les faux positifs est divisé par le nombre total de carrés dans l'image. La valeur moyenne pour chaque séquence est mise dans le tableau. La fréquence des objets perdus (faux objets positifs) ainsi que des faux objets détectés (faux objets négatifs) est présentée aussi. Pour mieux voir l'impact du changement des paramètres N et $th_0$ sur la détection, les séquences utilisées ont une résolution de 512X384 pour la première expérience, alors que pour la seconde où notre méthode est comparée à celles d'autres travaux, la taille standard de 192x128 est utilisée.

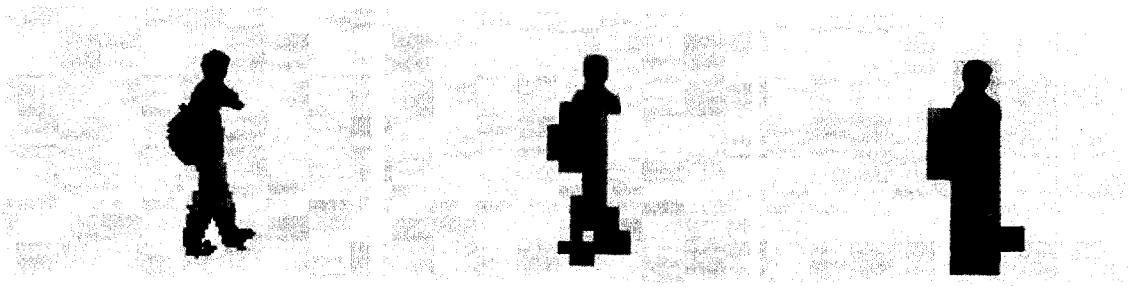
Dans cette deuxième expérience basée sur des résultats présentés pour le projet Wallflower [51], les résultats de différents algorithmes de détection appliqués à des séquences vidéo sont présentés. Chaque séquence adresse un problème spécifique de l'arrière-plan. Nous avons comparé notre méthode avec les résultats des algorithmes présentés à la section 2.1 et ceux de Heikkilä et Pietikäinen [15]. Des résultats qualitatifs et des résultats quantitatifs sont présentés. On présente aussi le cumulatif des faux négatifs/positifs de chaque algorithme représenté.

Pour la quantification, nous avons utilisé 54 (6X3X3) intervalles HSV, 6 pour H, 3 pour S et 3 pour V. Cette quantification permet d'avoir les informations essentielles sur la scène tout en minimisant le nombre de données à traiter.

### 4.2.2 Tests expérimentaux :

#### 4.2.2.1 Évaluations qualitatives :

Commençons tout d'abord par une discussion sur les avantages de l'algorithme de détection proposé. La figure 4.1 montre un avantage de l'algorithme de détection. Si on a juste besoin de détecter le mouvement dans une image sans avoir besoin de détecter les contours de l'objet avec précision (comme montré à la figure 4.1A où y= 4, y étant le nombre de divisions effectuées pour obtenir le résultat final), nous pouvons arrêter l'algorithme pour des valeurs de N grandes comme montré à la figure 4.1B et 4.1C, où y est mis à 2 et 1 respectivement. En plus, l'algorithme est plus rapide si on s'arrête à une grande échelle.



A) B) C)

**Figure 4.1** : Résultat de l'algorithme de détection
A) à Xi=4, B) à Xi=16, C) à Xi=32

La figure 4.2 est tirée de la séquence *atrium*. Dans cette figure, il est intéressant de voir le résultat de l'algorithme de détection d'ombrage sur les images de la deuxième ligne; l'ombre étant représenté en blanc. Il nous permet d'éliminer le bruit dans les bordures des objets segmentés.

On remarque que parfois, comme à l'image 97, l'ombre de la personne à ses pieds change radicalement la couleur du sol, ce qui fait qu'il échappe à la détection et est inclus dans la segmentation de l'objet en mouvement. À l'image 130 la troisième personne est perdue de la détection, ceci est dû au fait qu'elle se trouve à une position où elle est coupée sur deux régions, et chaque partie est trop petite pour être détectée dans la région où elle se trouve pour les paramètres plus exigeants utilisés ($th_0 = 0.15$ et N= 32). Une manière de régler ce problème est de prédire la position de l'objet grâce à sa position et sa vitesse ou en intégrant une des méthodes prédictives présentées à la section 2.2.1, et faire appel à cette méthode dans les cas où la détection par apparences échoue, puis diminuer dynamiquement $th_0$ pour la région où l'objet devrait se trouver.

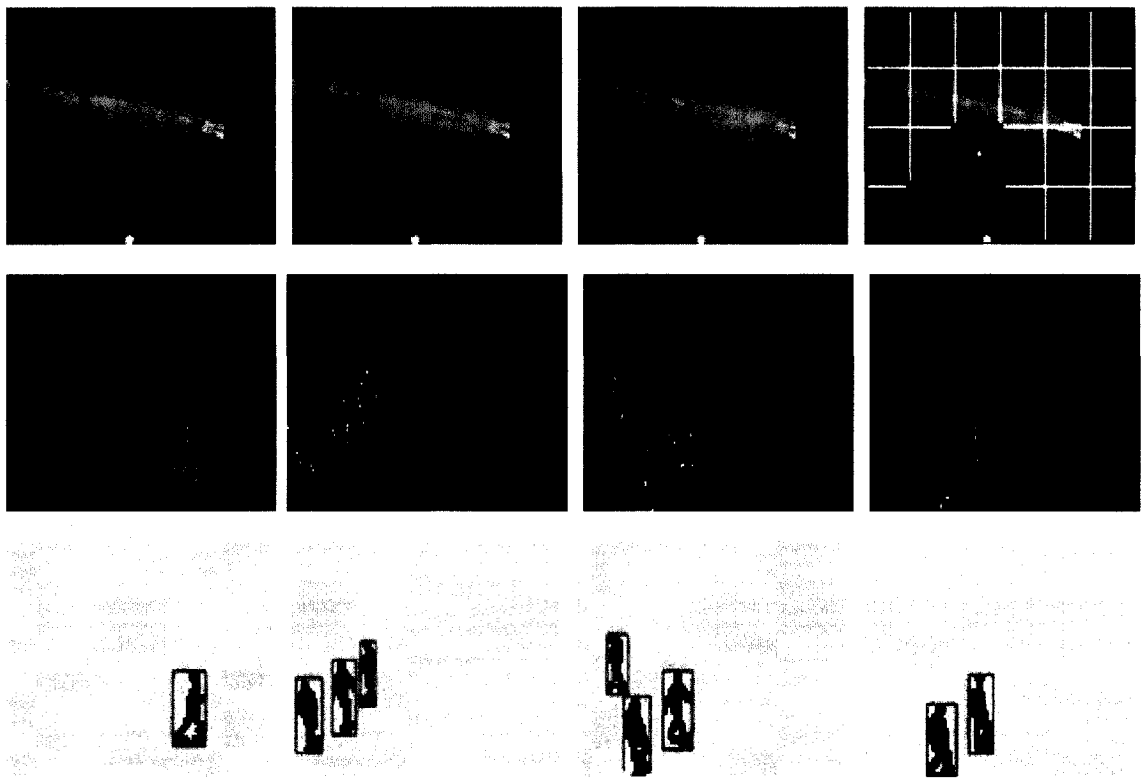


Image 97 image 117 image127 image130

**Figure 4.2** : personnes circulantes dans un atrium

La figure 4.3a montre les résultats de l'algorithme de détection sans post-traitement. Nous pouvons voir que du bruit apparaît près des régions segmentées. Ce bruit est dû à la présence de l'ombre autour de l'objet. Ceci est causé en partie par le fait que le sol a une grande réflectivité et des réflexions se trouvent dans les régions où du mouvement a été détecté. Un résultat plus précis peut être obtenu en appliquant le post-traitement (le détecteur d'ombre) comme montré à la figure 4.3B.

A) B) C)

**Figure 4.3** : A) Avant post-traitement, B) Après post-traitement
C) Image originale

La figure 4.4 représente une comparaison visuelle de notre algorithme avec d'autres algorithmes présentés à la section 2.1. Globalement, notre algorithme performe mieux que les autres algorithmes en faisant moins d'erreurs de détection, ce qui est dû au traitement de l'image par blocs, qui permet de détecter que les mouvements importants tout en étant robuste au bruit. Une discussion est faite plus bas sur les changements de paramètres pour les séquences *oscillation d'arbres* et *heures de journée.* Pour la séquence *ouverture d'avant-plan*, l'erreur de détection est principalement due au fait que la personne fait déjà partie de l'arrière-plan. Comme le pull a une couleur uniforme, un mouvement du pull dans la même région ne permet pas de détecter ce mouvement. Une manière de contrer ce problème serait d'intégrer une méthode de détection d'arête pour détecter aussi les bordures des objets. Pour ce qui est de la séquence *objets bougés*, la chaise et le téléphone ne devraient pas être détectés (voir segmentation idéal), car ce sont des objets appartenant à l'arrière-plan qui ont fait des mouvements non significatifs. Ces mouvements ne posent pas de problème à notre

méthode par ce qu'ils ont bougé dans le même bloc de l'image ne produisant pas de changement dans la couleur de ce bloc, ce qui fait que le mouvement n'est pas détecté.

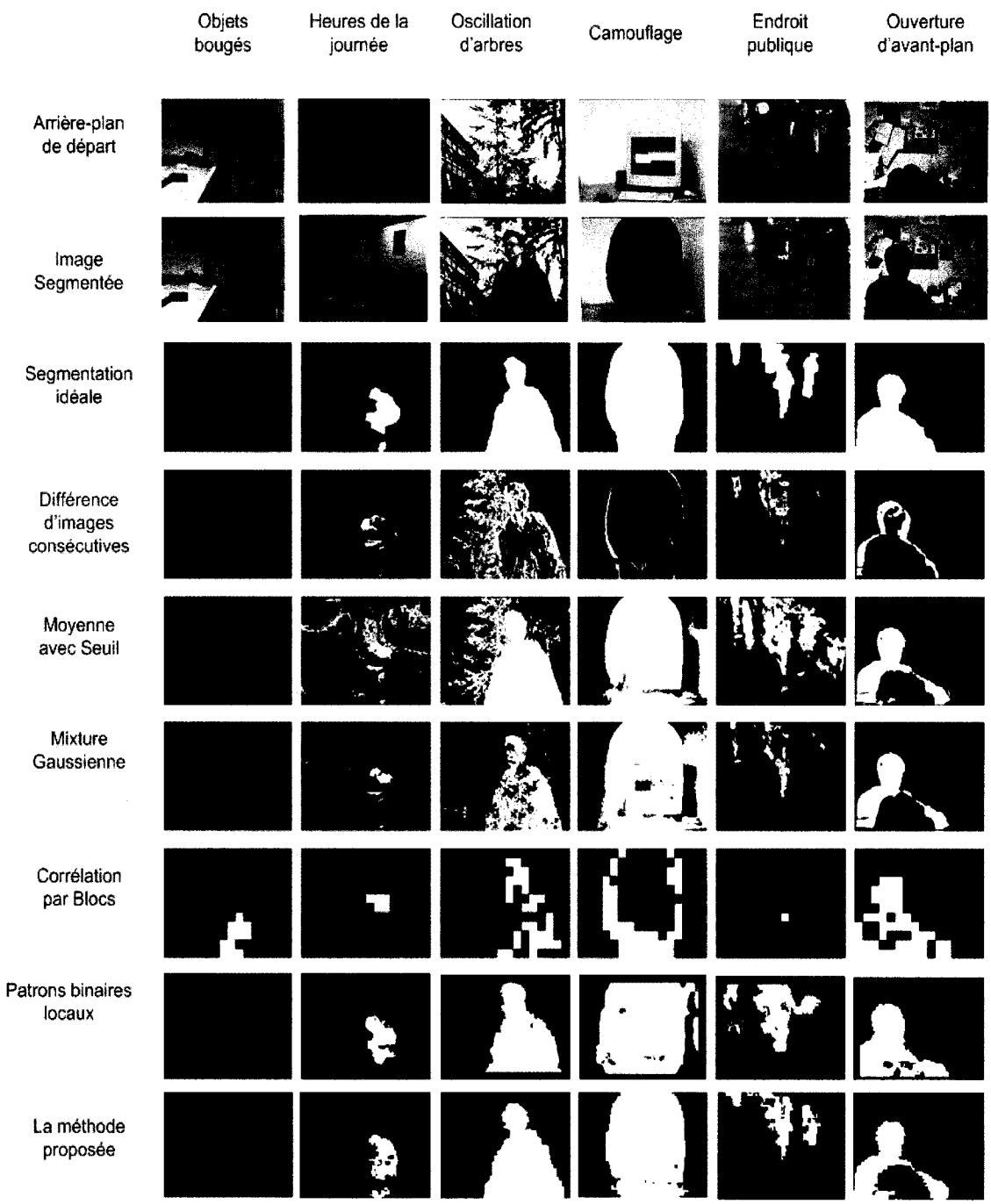



**Figure 4.4** : Résultats visuels des algorithmes de détection appliqués aux séquences Wallflower.

À la figure 4.5 l'algorithme de détection a été testé sur la séquence *Mouvement d'arbres* décrite plus haut. Comme la taille de la personne est grande relativement à la taille de la fenêtre, N est fixé à 64. Pour minimiser l'importance des mouvements d'arbres en arrière-plan $Th_0$ est fixé à 0.15. La première ligne montre les images de la séquence originale, et la deuxième les images résultantes de l'algorithme de détection.

Nous remarquons que les arbres en mouvement ne sont pas détectés car ils ne produisent pas des changements globaux dans la couleur des carrés de taille N. Par contre, du bruit est intégré dans les bordures de l'humain puisque les mouvements des arbres se sont introduits dans la subdivision des bordures, et qu'à petites échelles ces mouvements prennent de l'importance. Pour régler ce problème, on pourrait intégrer une méthode de détection d'arêtes pour mieux détecter les contours des objets et en enlever le bruit.

Image 9 image 14 image18 image19

**Figure 4.5** : Arbres en mouvement

La figure 4.6 représente la séquence *Heures de la journée*. Comme le changement de la lumière est graduel, l'algorithme de détection n'est pas altéré entre deux images adjacentes puisque la couleur ne change pas radicalement. Dans les régions où aucun mouvement n'est détecté la mise à jour de l'arrière-plan est lancée et ainsi la nouvelle intensité de la lumière est mise à jour dans l'arrière-plan.

On remarque aussi que du bruit est inclus dans la segmentation de l'objet détecté, ceci est dû au fait que la couleur de l'objet est presque similaire à celle du divan dans les petits blocs de détection. De plus, le pied est perdu à l'image 1883 parce qu'il est trop petit pour être détecté dans le bloc le contenant, car il n'y a pas un changement de couleur suffisamment important pas rapport à la taille du bloc.

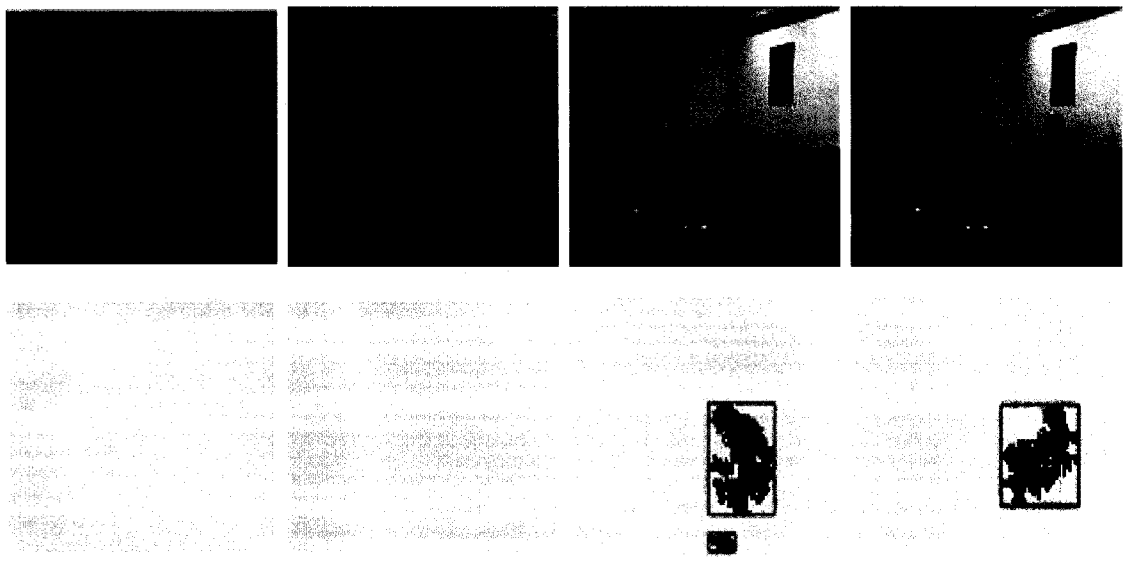


Image 160 image 420 image1850 image1883

**Figure 4.6** : changement graduel de la lumière

#### 4.2.2.2 Évaluations quantitatives :

Le tableau 4.1 présente les résultats quantitatifs pour l'algorithme de détection appliqué aux séquences vidéo des figures 4.10 et 4.2. La section 4.2.1 présente la méthodologie suivie pour effectuer l'évaluation.

Comme on peut le constater, si les objets à détecter sont trop petits pour être détectés relativement à N et $Th_0$ est trop grand, les régions en mouvement sont mal détectées et parfois des objets en mouvement sont complètement perdus parce que leur mouvement ne produit pas assez de changements pour être détectés dans les régions où ils se trouvent. A l'inverse quand $Th_0$ est trop petit, le nombre de régions de faux positifs augmente, créant de faux objets positifs dans certaines images. Ainsi, quand N est grand relativement aux objets en mouvement, de bons résultats sont obtenus pour des $Th_0$ petits et vice versa. On remarque aussi que les temps d'exécution relativement au

**Tableau 4.1** : Évaluation quantitative de l'algorithme de détection.

| Séquence | N | $Th_0$ | Faux positives pixels | Faux négatifs pixels | Faux positifs objets | Faux négatifs objets | Temps d'exécution total |
|---|---|---|---|---|---|---|---|
| Figure 4.2 (160 image) | 64 | 0.1 | 0.20% | 0.017% | 0% | 0% | 13 sec |
| | | 0.2 | 0.10% | 0.23% | 0% | 27% | 12 sec |
| | 32 | 0.2 | 0.18% | 0.017% | 0% | 0% | 15 sec |
| | | 0.3 | 0.031% | 0.14% | 0% | 19% | 14 sec |
| Figure 4.10 (100 image) | 64 | 0.2 | 0.14% | 0.04% | 0% | 0% | 20 sec |
| | | 0.3 | 0.07% | 0.15% | 0% | 1% | 19 sec |
| | 32 | 0.1 | 0.35% | 0.015 | 30% | 0% | 30 sec |
| | | 0.2 | 0.13% | 0.017% | 0% | 0% | 16 sec |

nombre d'images montrent que l'algorithme est relativement rapide à l'exécution, même s'il est non temps réel sous son implantation présente (3-12 images par secondes).

Le tableau 4.2 contient l'évaluation quantitative des résultats montrés à la figure 4.4 la méthodologie pour faire cette évaluation est expliquée à la section 4.2.1. On remarque que le nombre de faux positifs est moins élevé comparativement aux autres algorithmes. Cela est dû principalement au fait que l'approche globale (utilisation de régions de pixels) adoptée fait en sorte qu'il y ait moins de fausses détections introduites dans les résultats. Le nombre de faux négatifs lui aussi est moins élevé que pour les autres algorithmes, cela prouve globalement que notre algorithme détecte mieux les mouvements dans l'image. La figure 4.7 est la représentation graphique du tableau pour les erreurs totales.

**Tableau 4.2** : Résultats quantitatifs des algorithmes présentés à la figure 4.4

| Algorithmes | Faux | séq.1 | séq.2 | séq.3 | séq.4 | séq.5 | séq.6 | erreurs totales |
|---|---|---|---|---|---|---|---|---|
| différence d'images | négatifs | 0 | 1165 | 3509 | 990 | 1881 | 3884 | 11429 |
| | positifs | 0 | 193 | 3280 | 170 | 294 | 470 | 4407 |
| Moyenne avec seuil | négatifs | 0 | 873 | 17 | 194 | 415 | 2210 | 3709 |
| | positifs | 0 | 1720 | 3268 | 1638 | 2821 | 608 | 10055 |
| Mixture Gaussienne | négatifs | 0 | 1008 | 1323 | 398 | 1874 | 2442 | 7045 |
| | positifs | 0 | 20 | 341 | 3098 | 217 | 530 | 4206 |
| Corrélation par blocs | négatifs | 0 | 1030 | 3323 | 6103 | 2638 | 1172 | 14266 |
| | positifs | 1200 | 135 | 448 | 567 | 35 | 1230 | 3615 |
| Notre méthode | négatifs | 0 | 267 | 27 | 121 | 790 | 1940 | 3145 |
| | positifs | 0 | 180 | 411 | 246 | 555 | 562 | 1954 |

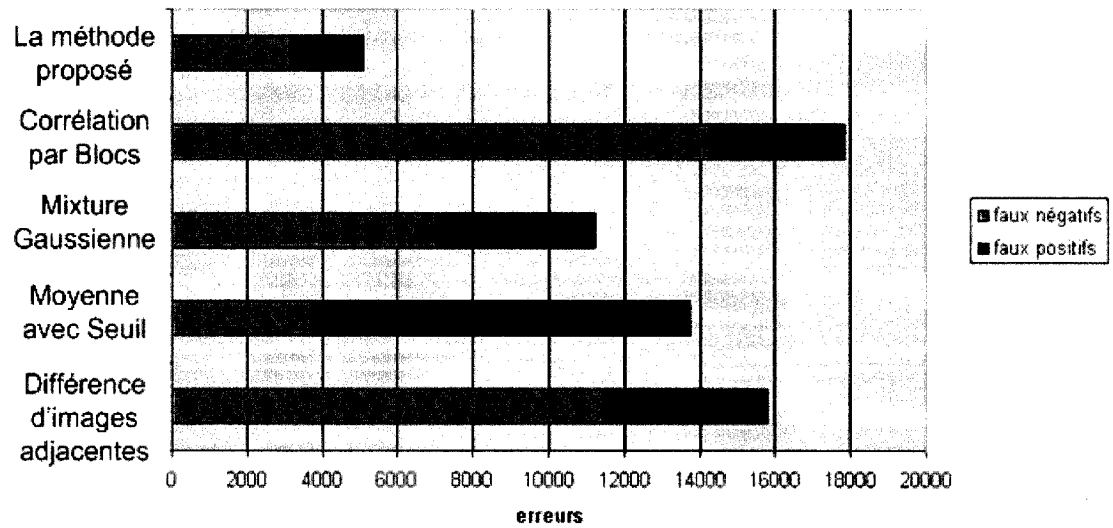



**Figure 4.7** : Performances globales

Ainsi, l'étude de l'image par régions permet notamment de contrer le problème de distributions multimodales comme dans le cas de *l'oscillation d'arbres* puisque les oscillations s'effectuent dans une même région, ce qui ne change pas la couleur globale de la région. Un autre problème corrigé est celui du changement graduel de la lumière (*Heures de la journée)* puisqu'une mise à jour est effectuée régulièrement pour les pixels des régions où aucun mouvement n'est détecté. Enfin, des objets appartenant à l'arrière-plan qui ont bougé dans la même région (*objets bougés*) ne posent pas de problèmes encore une fois parce que la couleur n'a pas changé dans les régions où les mouvements se sont effectués.

À noter que les résultats quantitatifs pour la méthode des patrons binaires [15] ne sont pas disponibles, c'est pour cela qu'ils n'ont pas été ajoutés au tableau.4.2.

## 4.3 Validation de l'algorithme de suivi

L'algorithme de suivi utilise les objets segmentés par l'algorithme de détection et étudie leur texture, couleur et taille pour pouvoir les suivre d'une image à une autre et savoir si une fusion entre plusieurs objets ou une fragmentation d'un objet se produit dans la séquence. La précision de l'algorithme de détection dépend de la précision de l'algorithme de suivi.

### 4.3.1 Méthodologie :

Quelques résultats qualitatifs obtenus sur des séquences vidéo sont présentés. Pour l'étude quantitative les textures des objets détectés (voir figure 4.8) sont comparées en calculant la distance entre l'intersection de leur corrélogrammes et le corrélogramme du plus petit des deux objets; le résultat est présenté au tableau 4.3. Ceci permet de confirmer l'utilité des corrélogrammes pour distinguer entre les textures.

Les valeurs données au vecteur de distance utilisé pour les corrélogrammes sont $D = \{1, 3, 5, 7, 9, 13\}$. Les corrélogrammes et les histogrammes ont été normalisés en utilisant la méthode proposée par Huang et al. [39]. Des résultats qualitatifs sont présentés dans les figures qui suivent, et des résultats quantitatifs sont présentés plus loin dans un tableau pour montrer la variation de la distance entre les corrélogrammes de deux objets similaires ou complètement différents.

### 4.3.2 Tests expérimentaux :

#### 4.3.2.1 Évaluation quantitative :

La figure 4.8 représente un exemple des objets que l'algorithme de détection segmente dans une séquence vidéo donnée, pour que l'algorithme de suivi puisse faire le suivi.

Le tableau 4.3 confirme l'utilité de l'intersection des corrélogrammes pour vérifier si un objet contient la texture d'un autre objet. Quand la valeur de la distance se rapproche de 0 alors les deux objets contiennent la même texture, cela veut dire que soit un objet fait partie de l'autre (paires d'objets (E, F), (A, C)) ou que les deux objets sont les mêmes (paire d'objets (A, B)). Sinon, plus la valeur s'éloigne de 0 plus les textures sont différentes comme on peut le vérifier avec les paires d'objets (C, D) et (E, G).

**Tableau 4.3** : Distance entre les corrélogrammes.

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| A | 0 | 0.052 | 0.04 | 0.24 | 0.26 | 0.15 | 0.49 |
| B | 0.052 | 0 | 0.0020 | 0.24 | 0.27 | 0.14 | 0.50 |
| C | 0.04 | 0.0020 | 0 | 0.27 | 0.26 | 0.15 | 0.27 |
| D | 0.24 | 0.24 | 0.27 | 0 | 0.27 | 0.11 | 0.46 |
| E | 0.26 | 0.27 | 0.26 | 0.27 | 0 | 0.027 | 0.42 |
| F | 0.15 | 0.14 | 0.15 | 0.11 | 0.027 | 0 | 0.12 |
| G | 0.49 | 0.50 | 0.27 | 0.46 | 0.42 | 0.12 | 0 |

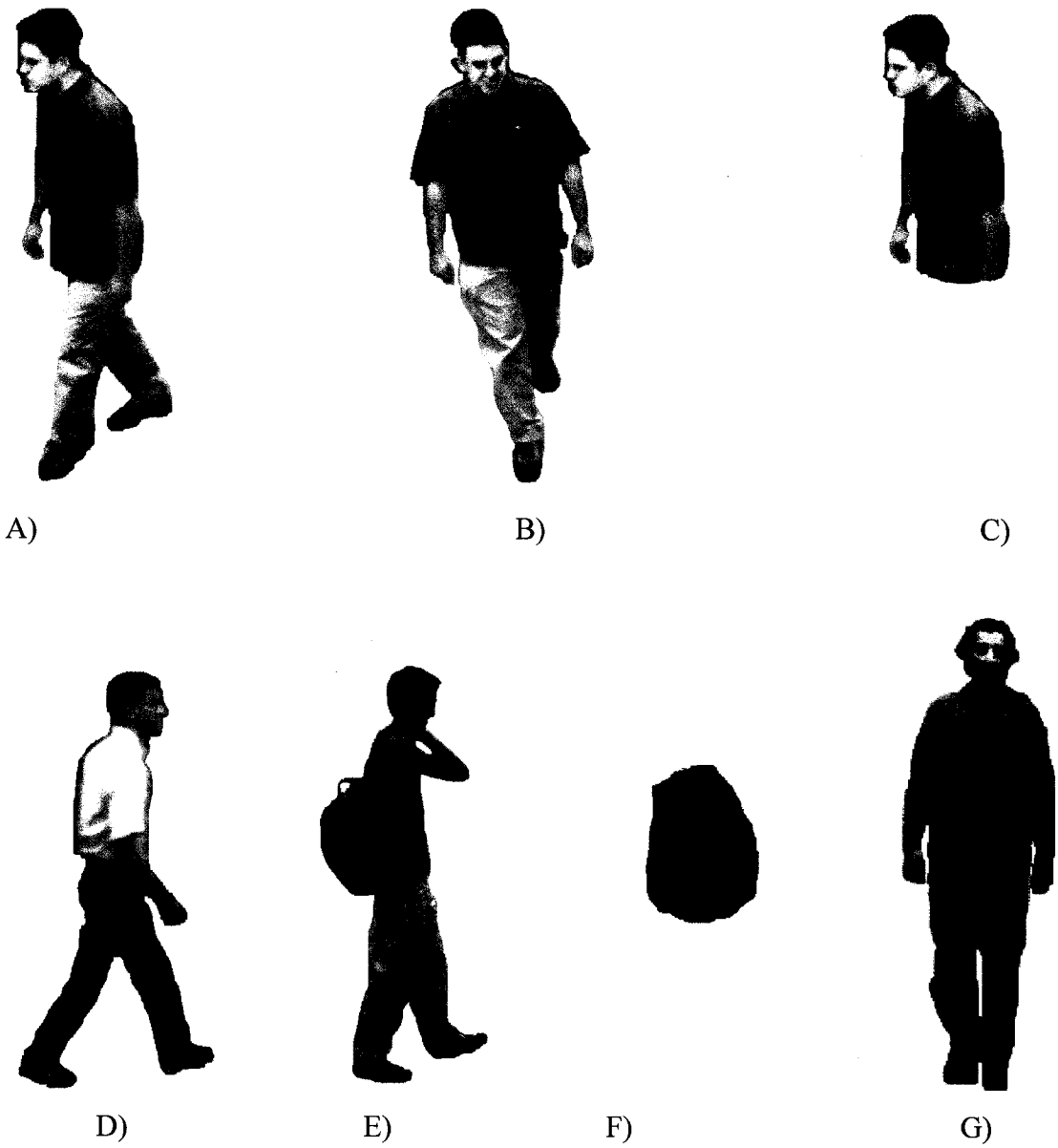



**Figure 4.8** : Objets à comparer pour la similarité.

### 4.3.2.2 Évaluations qualitatives :

Dans la séquence présentée à la figure 4.9, représentant la séquence *Dépôt de sac avec occlusion,* l'objet suivi est montré dans le rectangle englobant. A la figure 4.9B la couleur et la texture des jambes de la personne sont de la même couleur que l'arrière-plan pour la même position (figure 4.9C). Comme c'est le cas pour plusieurs algorithmes de détection, cette partie n'est pas détectée. Mais, comme expliqué aux sections 3.4 et 3.5, l'algorithme de suivi peut déterminer que les deux objets appartiennent au même objet en se basant sur l'analyse des histogrammes et des corrélogrammes des images précédentes. L'intersection entre la couleur et la texture de la partie représentant les pieds à la figure 4.9B et la personne au complet à l'image précédente est faite, puis la distance entre les intersections des histogrammes et l'histogramme des pieds sont calculés, si la couleur s'avère être la même alors la distance entre l'intersection des corrélogrammes et le corrélogramme des pieds sont calculés, comme la valeur se rapproche de zéro alors on considère que la texture fait partie de la personne et ainsi on déduit que les pieds appartiennent à la même personne.

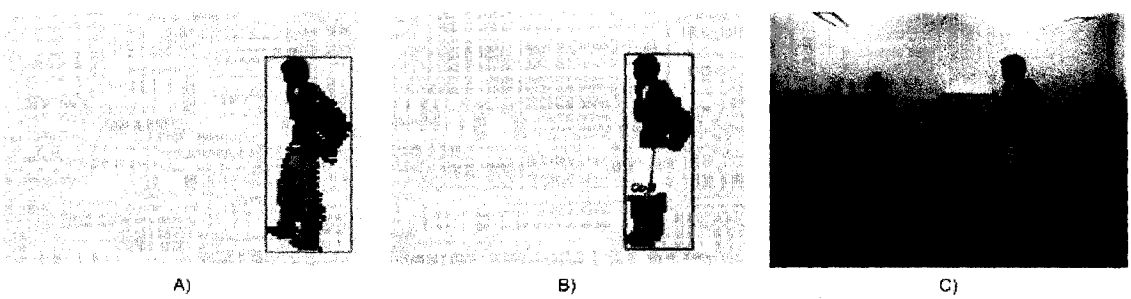



**Figure 4.9** : A) Détection d'un objet dans la scène. B) Suivi de l'objet. C) Image originale.

La figure 4.10A présente l'image où le sac est déposé par la personne. Le sac fait encore une partie de la personne. Un cercle représente son centroïde. L'algorithme de suivi continue de grouper les fragments des objets, comme montré à la figure 4.10B, en comparant les informations de tous les fragments présents dans l'image courante avec les dernières informations connues de l'objet les constituant à la figure 4.10B. Les distances entre les centroïdes du groupe sont calculées; puisque les fragments sont près du centroïde du groupe, nous considérons que les fragments font toujours partie du même objet, c'est pour cela qu'ils apparaissent sous le même rectangle englobant. A la figure 4.10C, le sac se retrouve loin du centroïde du groupe représentant la personne, donc il est considéré comme séparé et il est mis dans un rectangle englobant séparé.

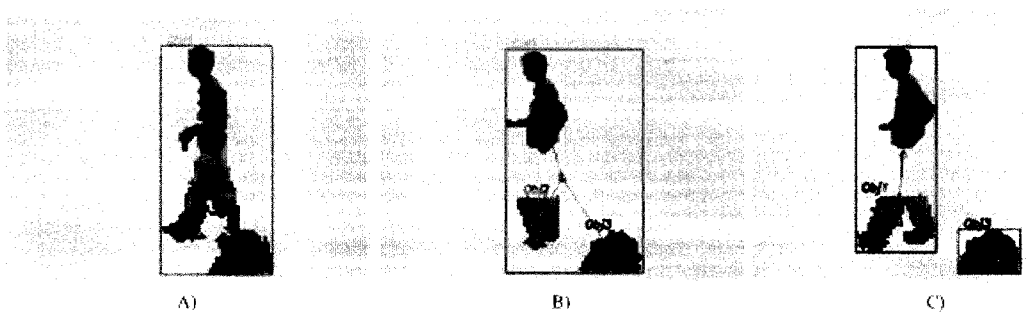
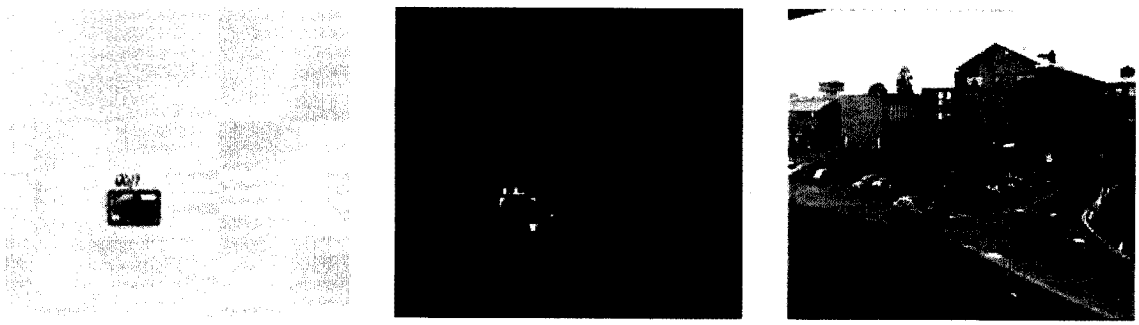



**Figure 4.10** : A) Le sac est toujours connecté à la personne B) Les fragments groupés D) Le sac a été déposé

La figure 4.11 représente l'image 362 de la base de données PETS 2001 dataset 2 (caméra 2) vidéo séquence, elle représente une voiture qui roule sur une route en plein jour. Dans cette séquence l'objet (la voiture) est petit relativement à l'image alors N est mis à 32 et Th0 reste à 0.10. La voiture est détectée et suivie correctement puisque sa

couleur et sa texture est assez claire. Quand Th0 est incrémenté à 0.30, pour trois images de la séquence la voiture n'est pas détectée; ceci est dû au fait que la voiture se trouve aux coins de 4 carrés et se trouve ainsi coupée en quatre, chaque partie étant trop petite pour être détectée dans des carrés différents puisqu'elle ne produit pas un changement assez important dans la couleur de la région. A noter que pour le reste de la séquence PETS, il y a des passants marchant sur la route. Ils sont trop petits pour être détectés pour des valeurs grandes de N et si l'algorithme est appliqué directement pour des valeurs petites de N il perd son avantage puisque le bruit ne sera pas bien filtré pour ces petites régions.



A) B) C)

**Figure 4.11** : image 362 de Pets dataset1, 2001, camera2
A) Détection de la voiture en mouvement B) Suivi de la voiture.

La figure 4.12 représente la séquence *sac déposé*, le sac est déposé à l'image 60, mais comme les deux objets sont connectés ils sont segmentés comme un seul objet. À l'image 73, ils sont séparés mais l'algorithme de suivi les relie toujours car ils ne sont pas assez distants l'un de l'autre, ce qui veut dire qu'ils sont peut être encore

le même objet et que la séparation est due à une occlusion. Pour les même raisons citées pour la figure 4.10, l'algorithme arrive à dire que les deux objets étaient unis précédemment grâce à l'intersection des histogrammes et des corrélogrammes comme expliqué aux sections 3.4 et 3.5. À l'image 140, le sac est repris par la personne et donc les deux objets sont segmentés comme un seul objet, l'algorithme arrive à dire que le nouveau objet est formé de deux objets (sac et personne) précédemment séparés dans la scène. Ceci grâce encore aux intersections des corrélogrammes et des histogrammes.

Donc l'algorithme de suivi peut être utilisé pour identifier les cas de dépôt et de prise d'objets.

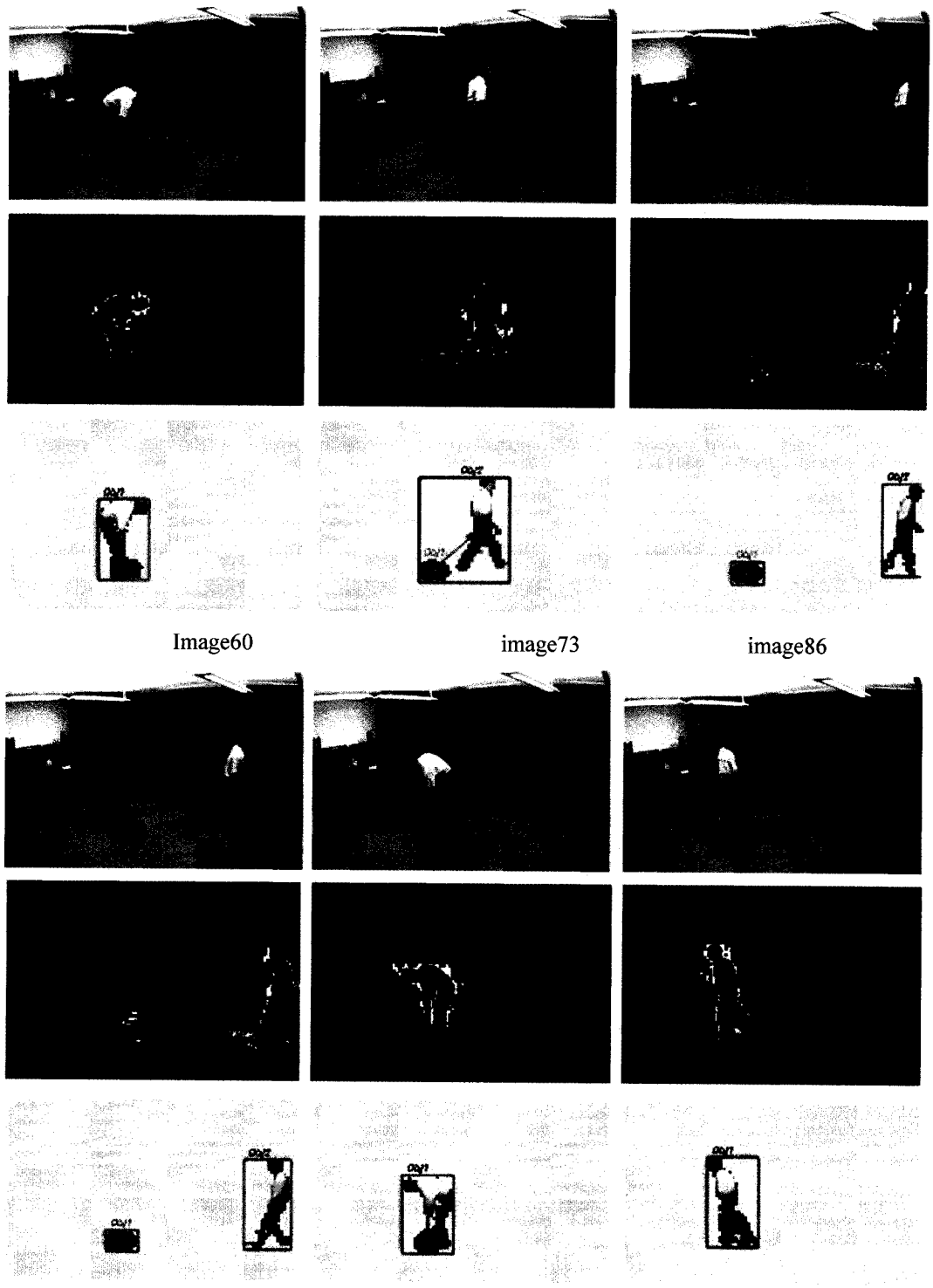



**Figure 4.12** : Démonstration de l'algorithme de suivi (1)

La figure 4.13 représente la séquence *deux personnes qui se croisent,* à l'image 71 l'algorithme nous informe que les deux objets en mouvement étaient liés précédemment. Ceci est possible parce qu'à l'image 68 représentant leur image avant leur séparation, il y a assez d'information sur leur couleur et leur texture pour pouvoir identifier les deux objets, encore une fois, grâce aux intersections des histogrammes et des corrélogrammes. À l'image 77 ils sont assez éloignés pour considérer qu'ils sont séparés.

Ainsi, l'algorithme de suivi peut aussi être utilisé pour détecter les cas de croisement entre les deux objets. En identifiant les objets avant et après le croisement la correspondance est facilement faite.

image 68 image 71 image 77

**Figure 4.13** : Démonstration de l'algorithme de suivi (2)

# CONCLUSION

Dans ce mémoire de maîtrise nous avons présenté un algorithme pour faire la détection des objets en mouvement dans une scène filmée par une caméra fixe, et un autre pour faire leur suivi.

L'algorithme de détection présente une approche globale. Il consiste à faire une division successive des régions de l'image où du mouvement est détecté. Le mouvement est détecté grâce à la comparaison des distributions de couleurs de régions en utilisant les histogrammes de couleurs. Ces structures sont normalisées, ce qui permet à l'algorithme de performer en présence de différents niveaux d'illumination. Une mesure de distance est ensuite utilisée pour en faire la comparaison et éventuellement catégoriser les pixels en tant que pixel d'arrière-plan ou d'avant-plan.

Dans les tests expérimentaux effectués, l'algorithme de détection a été comparé avec d'autres algorithmes. Les performances achevées ont permis de déduire que l'approche globale développée permet de pallier aux problèmes classiques rencontrés par les algorithmes de détection. Notamment, les distributions multimodales telles que l'oscillation des arbres ou de l'eau, et les mouvements sans importance s'effectuant dans une même région tel que le mouvement d'une chaise, ne posent pas de problème. Le changement graduel de la lumière lui non plus ne pose pas de problème puisque ces changements graduels ne sont pas détectés et l'arrière-plan est mis à jour régulièrement. On remarque aussi, que globalement le nombre de faux négatifs et faux positifs est

moins élevé que pour les autres algorithme cela est principalement dû à l'étude de l'image par région qui permet de filtrer le maximum de bruit tout en détectant les mouvements importants.

L'algorithme de suivi utilise une approche par apparence. Les objets détectés par l'algorithme de suivi sont définis par leur couleur en utilisant les histogrammes de couleurs, leur texture en utilisant les corrélogrammes, ainsi que leurs tailles. Un arbre de décision est construit en utilisant ces informations. Cet arbre permet de savoir si deux objets sont similaires, ou si l'un fait partie de l'autre. On peut alors traiter les cas de fusion ou de séparation entre les objets.

Dans les tests expérimentaux effectués, l'approche par apparences proposée combinant dans l'espace HSV, la couleur, la texture et la taille des objets suivi a permis d'obtenir de bons résultats notamment grâce à l'utilisation des corrélogrammes qui est particulièrement intéressante, car ces structures permettent de combiner les informations sur la couleur et sa distribution spatiale, ce qui donne une bonne description des objets détectés et permet de les distinguer entre eux en utilisant les intersections de ces structures. L'intersection nous a aussi permis d'identifier les objets fragmentés, et les cas de fusion et de séparation entre les objets.

Les contributions de ce travail sont:

1. Dans la phase de détection, l'utilisation d'une approche globale pour faire la détection des objets en mouvement. Cette approche consiste à faire la détection par régions puis subdiviser les régions où du mouvement est détecté, jusqu'à une échelle spécifiée.

2. Dans la phase du suivi, l'utilisation d'un arbre de décision qui intègre les corrélogrammes pour décrire la texture des objets détectés et permettre d'identifier les cas de fusion et de séparation grâce à l'intersection de ces structures.

Pour ce qui est des travaux futurs, il y a plusieurs points qui pourraient être améliorés où rajoutés. Les expériences ont montré que dans certains cas, notamment pour la séquence *arbre en mouvement,* les paramètres ont dû être changés pour obtenir de meilleurs résultats. Pour les bordures des objets détectés, l'algorithme introduit quelques erreurs, donc parmi les améliorations possibles :

1. Les seuils utilisés devraient s'adapter dynamiquement selon la situation où le système se trouve. Notamment, le seuil concernant la distance de séparation entre deux objets pour le suivi, le calibrage pourrait être utilisée pour connaître la distance des objets de la caméra et entre les deux objets pour décider d'une distance de séparation minimale.
2. La distance utilisée pour comparer les histogrammes et les corrélogrammes pourrait être changée pour une distance telle que la distance « earth mover » [38] qui tient compte de la distribution des

erreurs. Elle permettrait de savoir où la différence de couleur entre deux objets donnés se situe et déduire si cette différence n'est pas due à une occlusion. Par contre, c'est une méthode plus lente que le calcul de la distance L1

3. Une méthode pour prédire les endroits où des occlusions pour les objets suivis sont probables pourrait être implémentée. Comme l'image et les objets sont divisés en régions, une correspondance entre les régions de l'image et celles de l'objet ayant la même texture ou couleur pourrait être faite dès l'entrée de l'objet dans la scène.
4. Des méthodes de prédiction et d'interprétation telles que la chaîne de Markov peuvent éventuellement être rajoutées pour renforcer les résultats du suivi et permettre l'interprétation des interactions entre les objets.
5. Une méthode de détection d'arêtes pourrait être utilisée pour raffiner les bordures des objets détectés.
6. Finalement, un algorithme d'interprétation pourrait être implanté en utilisant les résultats de ce travail pour déterminer les actions en cours dans la scène, comme par exemple, identifier la prise et le dépôt d'objets, ou bien l'entrée ou la sortie d'un objet dans la scène.

# RÉFÉRENCES


[1] W. Hu, T. Tan, L. Wang and S. Maybank, *A Survey On Visual Surveillance Of Object Motion And Behaviors*, IEEE Transactions on Systems, Man and Cybernetics, Part C, Aug., 2004. pp 334—352, Vol. 34, Num. 3.

[2] R. Cucchiara, A. Prati, R. Vezzani, *Making the home safer and more secure through visual surveillance*, Proceedings of Symposium on Automatic detection of abnormal human behaviour using video processing of Measuring Behaviour, Wageningen, The Netherlands, 2005.

[3] A. F. Bobick, S. S. Intille, J. W. Davis, Freedom Baird, Claudio S. Pinhanez, Lee W. Campbell, Yuri A. Ivanov, Arjan Schütte, Andrew D. Wilson, *The KidsRoom: A Perceptually-Based Interactive and Immersive Story Environment*, 1999. Presence 8(4): 369-393 .

[4] Wei Niu, Jiao Long, Dan Han and Yuan-Fang Wang, "Human Activity Detection and Recognition for Video Surveillance", ICME 2004.

[5] R.Cucchiara, C.Grana, M.Piccardi, A.Prati; *Detecting Moving Objects, Ghosts, and Shadows in Video Streams*, 2003. IEEE Trans. Pattern Anal. Mach. Intell. 25(10): 1337-1342.

[6] C. Staufer and W. E. L. Grimson, *Adaptive background mixture models for real-time tracking*, Computer Vision and Pattern Recognition Fort Collins, Colorado, Jun 1999. pp. 246-252.

[7] I.Haritaoglu, D. Harwood, L.S. Davis, *W4:Real-Time Surveillance of People and Their Activities*, IEEE Trans. on Pattern Analysis and Machine Intelligence, 2000. Vol. 22, No.8.

[8] L. G. Shapiro, G..C. Stockman, Sec. 9.2. *Image Subtraction*, 2000. Computer Vision, pp. 253-254. Prentice Hall.

[9] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, *Detecting Moving Objects, Ghosts, and Shadows in Video Streams*, 2003. IEEE Trans. Pattern Anal. Mach. Intell. 25(10): 1337-1342.

[10] Z. Szlávik, L. Havasi, T. Szirányi, *Estimation of common groundplane based on co-motion statistics*, ICIAR, 2004. Lecture Notes in Computer Science, Vol.LNCS 3211, pp.347-353.

[11] J. Heikkila and O. Silven, *A real-time system for monitoring of cyclists and pedestrians*, Second IEEE Workshop on Visual Surveillance Fort Collins, Colorado, Jun 1999. pp. 74-81.

[12] A. R. Francois and G. G. Medioni, *Adaptive color background modeling for real-time segmentation of video streams*, Las Vegas, NA, in Proceedings of the International Conference on Imaging Science, Systems, and Technology, 1999. pp. 227–232.

[13] M. Karaman, L. Goldmann, Da Yu, and Thomas Sikora, *Comparison of Static Background Segmentation Methods*. Visual Communications and Image Processing (VCIP '05), Beijing, China, July 12-15, 2005.

[14] T. Matsuyama, T. Ohya, and H. Habe, *Background Subtraction for Non-Stationary Scenes,* Department of Electronics and Communications, Graduate School of Engineering, Kyoto University: Sakyo, Kyoto, Japan, 1999.

[15] M. Heikkilä M. Pietikäinen, *A Texture-Based Method for Modeling the Background and Detecting Moving Objects*, IEEE pp. 657-662.

[16] H. Kang, D. Kim, *Real-time multiple people tracking using competitive condensation*. Pattern Recognition ,2005. 38(7): 1045-1058.

[17] V. Nair and J.J. Clark, *Automated visual surveillance using hidden markov models*, In International Conference on Vision Interface.2002. pages 88–93.

[18] D. Comaniciu et P. Meer, *Robust analysis of feature spaces: Color image segmentation*. CVPR'97, Juin 1997.pages 750–755.

[19] D. Comaniciu, V. Ramesh, P. Meer, *Real-Time Tracking of Non-Rigid Objects using Mean Shift*, BEST PAPER AWARD, IEEE Conf. Computer Vision and Pattern Recognition (CVPR'00), Hilton Head Island, South Carolina, Juin 1997. Vol. 2, 142-149.

[20] R. T. Collins; L. Yanxi; M. Leordeanu, *Online selection of discriminative tracking features*, Pattern Analysis and Machine Intelligence, IEEE Transactions on Volume 27, Issue 10, Oct. 2005. Page(s):1631 – 1643.

[21] J. G. Allen, R. Y. D. Xu, and J. S. Jin, *Object Tracking Using CamShift Algorithm and Multiple Quantized Feature Spaces*, presented at Workshop on Visual Information Processing, Conferences in Research and Practice in Information Technology, Sydney, Australia, 2003.

[22] R. Bodor, B. Jackson, N. Papanikolopoulos. *Vision-Based Human Tracking and Activity Recognition.* Proc. of the 11th Mediterranean Conf. on Control and Automation, 2003. p. 18-20.

[23] W. Niu, L. Jiao, D. Han, and Y. Wang. *Real-Time Multi-Person Tracking in Video Surveillance,* Proceedings of the Pacific Rim Multimedia Conference, Singapore, 2003.

[24] R. Bowden, P. KaewTraKulPong, *Towards Automated Wide Area Visual Surveillance: Tracking Objects Between Spatially Separated*, Uncalibrated Views. In IEE Proc. Vision, Image and Signal Processing, April 05. Vol 152, issue 02, pp213-224.

[25] G. Rigoll, H. Breit, F. Wallhoff, *Robust tracking of persons in real-world scenarios using a statistical computer vision approach.* 2004, Image Vision Comput. 22(7): 571-582

[26] N. Song Peng, J. Yang, Z. Liu, *Mean shift blob tracking with kernel histogram filtering and hypothesis testing*, 2005 Pattern Recognition Letters 26(5): 605-614.

[27] C. Sacchi, C. S. Regazzoni, G. Vernazza: A Neural Network-Based Image Processing System for Detection of Vandal Acts in Unmanned Railway Environments. ICIAP 2001: 529-534

[28] A. Koschan, S. Kang, J. Paik, B. Abidi., M. Abidi, *Color active shape models for tracking non-rigid objects*. Pattern Recognition Letters 24, 2003. pp. 1751--1765

[29] A. Jepson, D. Fleet, and T. El-Maraghi. *Robust online appearance models for visual tracking*. IEEE, In Proc. of Int. Conf. on Computer Vision and Pattern Recognition, 2001. volume I, pages 415–422.

[30] L. G. Shapiro, G..C. Stockman. Sec. 9.3.1. *Using Point Correspondence*, 2000. Computer Vision, pp. 256-260. Prentice Hall.

[31] R. C. Gonzalez, R. E. Woods, *Digital Image Processing* (seconde edition), Reading, Massachusetts: Addison-Wesley, 2002.

[32] A. Verri, S. Uras, and E. DeMicheli, *Motion segmentation from optical flow*, Proceedings of the Fifth Alvey Vision Conference, 1989. page 209214.

[33] L. M. Fuentes and S. A. Velastin, *People tracking in surveillance applications*, In 2nd IEEE International Workshop on Performance Evaluation on Tracking and Surveillance, PETS 2001.

[34] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta. and R. Jain, *Content-Based Image Retrieval at the End of the Early Years*, IEEE Transactions on pattern analysis and machine intelligence, december 2000. Vol.22, NO.12.

[35] R. Schettini, G. Ciocca, S. Zuffi, *A Survey on methods for colour image indexing and retrieval in image databases*,Color Imaging Science: Digital Media, (R. Luo, L. MacDonald eds.), J. Wiley, 2001.

[36] E. Saykol, U. Gudukbay, O. Ulusoy, *A Histogram-Based Approach for Object-Based Query-by-Shape-and-Color in Multimedia Databases*, Bilkent University Computer Engineering Dept. *Technical Report BU-CE-0201*, 2002

[37] E. Saykol, U. Gudukbay, O. Ulusoy, *A Histogram-Based Approach for Object-Based Query-by-Shape-and-Color in Image and Video Databases*, Image and Vision Computing, Vol. 23, No. 13, , November 2005. pp. 1170-1180..

[38] F. Serratosa, A. Sanfeliu, *Signatures versus histograms: Definitions, distances and algorithms*, Pattern Recognition 39, 2006. p. 921–934

[39] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, R. Zabih, Image indexing Using Color Correlograms, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico (1997).

[40] M. J. Swain and D. H. Ballard, *Color indexing*, International Journal of Computer Vision, vol. 7, no. 1, November 1991. pp. 11--32.

[41] A. Rosenfeld and M. Thurston, *Edge and curve detection for visual scene analysis*, IEEE Trans. Comput., 1971. vol. 20, pp. 562--569.

[42] D. A. Forsyth, J. Ponce, Chap. 9 *Texture,* 2003. Computer vision a modern approach, pp.189-196. Prentice Hall.

[43] R.M. Haralick, K. Shanmugam, I. Dinstein, *Textural Features for Image Classification*", IEEE Trans. On Systems, Man, and Cybernetics, Vol. SMC-3, No. 6, November 1973. pp. 610-621.

[44] E. L. van den Broek and E. M. van Rikxoort, *Evaluation of color representation for texture analysis*, in Proceedings of the Sixteenth Belgium-Netherlands Arti_cial Intelligence Conference, R. Verbrugge, N. Taatgen, and L. R. B. Schomaker, eds., 2004. pp. 35-42.

[45] L. G. Shapiro, G. C. Stockman, Sec. 5.5 *Median Filtering*, 2000. Computer Vision, pp. 10-11. Prentice Hall.

[46] T. Ojala, M. Rautiainen, E. Matinmikko & M. Aittola; *Semantic image retrieval with HSV correlograms*, Proc. 12th Scandinavian Conference on Image Analysis, Bergen, Norway, 2001. pp. 621-627.

[47] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti, *Improving Shadow Suppression in Moving Object Detection with HSV Color Information*, Proc. IEEE Int'l Conf. Intelligent Transportation Systems, Aug. 2001. pp. 334-339.

[48] L. G. Shapiro, G. C. Stockman, Sec. 3.4, *Connected Components Labeling; Algorithm 3.2,* 2000. Computer Vision, pp. 56-59. Prentice Hall.

[49] Open Source Computer Vision Library Project, website http://www.intel.com/technology/computing/opencv/index.htm

[50] PETS Dadatbase, ftp://pets.rdg.ac.uk/

[51] K. Toyoma, J. Krumm; B. Brumitt; and B. Meyers, *Wallflower: Principles and practice of background maintenance*. In International Conference on Computer Vision, 1999. pp. 255-261.

## ANNEXE I Espace de couleurs

L'espace RGB (Rouge, Vert, Bleu) est le plus utilisé en vision par ordinateur. Chaque axe a la même importance et doit être quantifié avec la même précision. Cette espace est un système additif puisqu'on rajoute des composantes à la couleur noir (0, 0, 0) pour arriver au maximum qui est le blanc (255, 255, 255) et ainsi obtenir une variété de couleur, plus précisément 16 millions codes de couleurs différents.

Voici une représentation visuelle de cet espace de couleur :

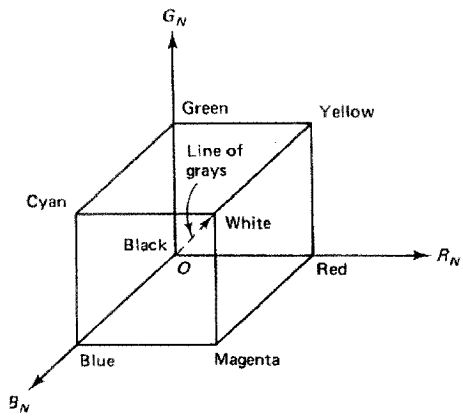



**Figure I.1** : Représentation de l'espace RGB

**RGB normalisé :** Cet espace est la normalisation de l'espace RGB, elle permet de diminuer l'effet de l'intensité de la lumière et donc être utilisée à différents niveaux de luminosité.

L'intensité I = (R+G+B)/3

Rouge normalisé r = R/ (R+G+B)

Vert normalisé g = G/ (R+G+B)

Bleu normalisé b =B/ (R+G+B)

Ainsi r + g + b =1 en mettant b= 1- r- g. cet espace peut être modélisé en 2 dimensions comme à la figure suivante.

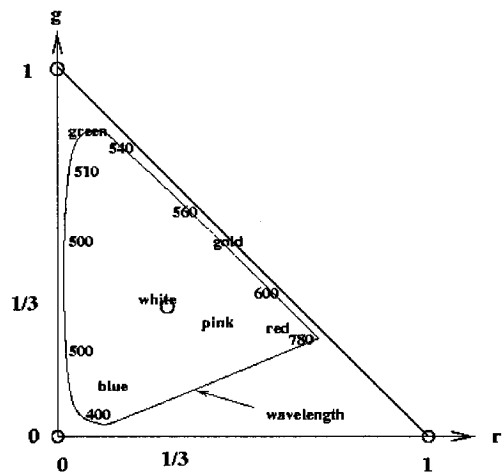



**Figure I.2** : Représentation de l'espace rgb normalisé

**YIQ :** Cet espace de couleurs est utilisé par les télévisions. La luminance est représentée par Y, qui à elle seule est utilisée pour les télévisions noir et blanc. La chromaticité est représentée par I, Q.

Une transformation simple de l'espace RGB vers YIQ est :

Y = 0.30 R + 0.59 G + 0.11 B.

$I = 0.60\ R - 0.28\ G - 0.32\ B.$

$Q = 0.21\ R + 0.52\ G + 0.31\ B.$

**HSV :** Cet espace de couleur est plus proche de la vision humaine que l'espace RGB. Il sépare l'intensité de la couleur V, de la chromaticité H et S. Ceci procure un contrôle sur l'intensité de la lumière présente dans la scène en donnant moins d'importance à V.

La teinte (H) est définie par un angle compris entre 0 et 2л relativement à l'axe du rouge, avec du rouge pur à l'angle 0, du vert pur à l'angle 2л/3 et du bleu pur à l'angle 4л/3.

La saturation (S) définie la pureté de la couleur. Elle est représentée par un réel qui varie de 0 à 1. Plus la saturation d'une couleur baisse, plus le gris sera présent et plus la couleur semblera plus pâle.

L'intensité (V) représente l'intensité de la couleur. Elle est représentée par un entier qui varie entre 0 et 255. Elle est utilisée pour représenter une image HSI en noir et blanc.

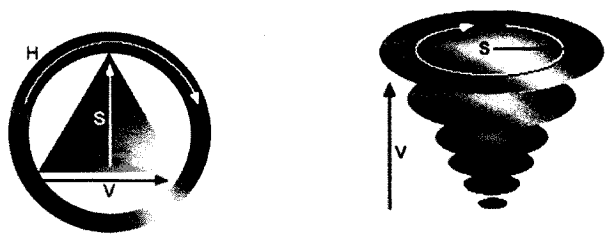



**Figure I.3** : Représentation de l'espace HSV

De nombreux travaux ont montré sa supériorité sur l'espace RGB pour faire la comparaison entre deux images de différentes tailles [3, 42]. A noter que dans notre cas, la comparaison se fait entre des régions de l'image en forme de carré pour l'algorithme de détection, et entre des objets de tailles différentes pour l'algorithme de suivi

## ANNEXE II Conversion en HSV

La conversion consiste à prendre chaque pixel de l'image en RGB et d'effectuer la conversion en HSV en appliquant la transformation d'OpenCV suivante :

$V = \max(R,G,B)$

$S = (V-\min(R,G,B))/V \quad Si\ V \neq 0,\ 0$

$\qquad\qquad\qquad\qquad\qquad\qquad sinon \qquad (G - B)X60/S,\ Si\ V=R$

$H = 180+(B - R)X60/S,\ Si\ V=G$

$\quad 240+(R - G)*60/S,\ Si\ V=B$

Si $H<0$ alors $H = H+360$

À la sortie on aura $0 \leq V \leq 1$, $0 \leq S \leq 1$, $0 \leq H \leq 360$.

Enfin, les valeurs sont converties pour être entre 0 et 255 :

$V = V*255,\ S = S*255,\ H = H/2.$