

UNIVERSITÉ MOHAMMED SEDDIK BENYAHIA  
JIJEL  
FACULTÉ DE SCIENCES EXACTES ET D'INFORMATIQUE



**MEMOIRE DE MASTER**

Présentée pour l'obtention du diplôme de :

**MASTER**

En **INFORMATIQUE**

**Option : INFORMATIQUE LEGALE ET MULTIMEDIA**

Par :

**BOUDAB Ilhem , BELHOUCHEAT Yousra**

Thème

**Détection et suivi des objets en mouvement  
dans les séquences vidéo**

Encadré par :

**Dr. MEHROUK Zahir**

ANNEE UNIVERSITAIRE 2021-2022

# *Remerciements*

Avant tout nous remercions ALLAH le tout puissant pour son aide et le courage qui nous a donné pour surmonter toutes les difficultés durant nos études.

Notre gratitude, nos sincères remerciements et nos salutations à notre encadrant, M. MAHROUK Zahir, pour tous ses conseils avisés, et le temps qu'il nous a consacré, et pour son orientation constante vers un esprit purement scientifique.

Nous adressons nos sincères remerciements aux membres du jury pour avoir accepté d'examiner et d'évaluer notre travail.

Un grand merci à tous les enseignants du département informatique qui ont été impliqués d'une manière ou d'une autre dans la formation en master et en licence.

Nous adressons également nos remerciements à notre famille, nos amis et tous ceux qui ont contribué de près ou de loin à la cristallisation de ces souvenirs, ainsi qu'à la réussite de cette belle année universitaire...

# *Dedicace*

*En témoignage d'amour et d'affection, je dédie ce travail avec  
une grande fierté*

*À mes parents qui ont été d'un dévouement exemplaire et  
d'un réconfort inestimable.*

*À mes frères, mes sœurs et toute ma famille en  
reconnaissance de leurs encouragements.*

*A tous mes amis pour leur sympathie, leur amitié et leur  
solidarité envers moi. J'espère que vous acceptez mes hautes  
salutations et considérations.*

*Que Dieu puisse vous protéger.*



# *Dedicace*

*Je rends grâce à Dieu de m'avoir donné le courage , la volonté et la conscience d'avoir pu terminer mes études.*

*Je dédie ce mémoire :*

*À mes parents qui m'ont soutenu et encouragé jusqu'ou bout et a qui je dois tout l'amour et le respect, j'espère que Dieu les bénira .*

*À mes chère frères et soeurs .*

*À tous mes amis et ceux qui m'aiment et m'aident, sans oublier toutes les profs de la promo.*

*Salutations et considérations.*

*Que Dieu puisse vous protège.*



**YOUSRA**

# Résumé

Au cours des dernières décennies, la détection et le suivi d'objets en mouvement ont attiré beaucoup d'attention en raison de leurs diverses applications dans la vie humaine, en particulier la vidéosurveillance et la robotique. Le suivi d'objet est l'estimation de l'emplacement d'un objet et de sa trajectoire dans le temps dans une séquence vidéo.

Bien que de nombreux algorithmes de détection et de suivi aient été développés ces dernières années pour le résoudre, il reste un problème non résolu en raison du grand nombre de facteurs environnementaux.

Dans cette mémoire, nous nous intéressons particulièrement à l'étude de l'algorithme de détection GMM (Modèle de mixture gaussien) et comment il a été amélioré par l'entropie 2D. Le but est d'obtenir une méthode de détection rapide et efficace, résistante au bruit et à l'éclairage soudain. Comme pour le suivi de l'objet en mouvement, nous avons proposé l'étude du filtre de Kalman. Ces algorithmes ont été appliqués à la base de données ... et à un certain nombre de problèmes pratiques pour voir s'ils nous conduisent à la solution attendue.

**Mots-clés** : *Vidéosurveillance, détection, suivi d'objet en mouvement.*

# Abstract

Over the past decades, detection and tracking of moving objects have attracted a lot of attention due to their various applications in human life, especially CCTV and robotics. Object tracking is the estimation of an object's location and trajectory over time in a video sequence.

Although many detection and tracking algorithms have been developed in recent years to solve it, it remains an unsolved problem due to the large number of environmental factors.

In this memory, we are particularly interested in the study of the detection algorithm GMM (Gaussian Mixture Model) and how it has been improved by 2D lontology. The goal is to obtain a fast and effective detection method, resistant to noise and sudden lighting. As for the tracking of the moving object, we proposed the study of the Kalman filter. These algorithms have been applied to the database... and to a number of practical problems to see if they lead us to the expected solution.

**Keywords :** *video surveillance, detection, tracking of moving objects*

# TABLE DES MATIÈRES

<b>Table des Matières</b>	<b>i</b>
<b>Table des figures</b>	<b>iv</b>
<b>Liste des tableaux</b>	<b>vi</b>
<b>Liste des acronymes</b>	<b>vii</b>
<b>Introduction Générale</b>	<b>1</b>
<b>1 Vidéosurveillance</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Définition de la vidéosurveillance . . . . .	4
1.3 Classifications des systèmes de vidéosurveillance . . . . .	4
1.3.1 Classification suivant le niveau d'automatisation . . . . .	4
1.3.1.1 Systèmes de surveillance manuels . . . . .	4
1.3.1.2 Systèmes de surveillance semi-autonomes . . . . .	4
1.3.1.3 Systèmes de vidéosurveillance autonomes (Systèmes de vi- déosurveillance intelligents) . . . . .	5
1.3.2 Classification suivant l'architecture réseau . . . . .	5
1.3.2.1 Architecture centralisée . . . . .	5
1.3.2.2 Architecture distribuée . . . . .	5
1.3.3 Classification suivant le domaine d'application . . . . .	5
1.3.3.1 Protection et confidentialité . . . . .	6
1.3.3.2 Analyse d'objets . . . . .	6
1.3.3.3 Reconnaissance de l'objet . . . . .	6
1.3.3.4 Surveillance du trafic . . . . .	6
1.3.3.5 Détection d'événements anormaux . . . . .	6
1.3.4 Classification suivant le niveau de l'application . . . . .	6
1.3.4.1 Premier niveau . . . . .	7
1.3.4.2 Second niveau . . . . .	7
1.3.4.3 Troisième niveau . . . . .	7
1.3.4.4 Quatrième niveau . . . . .	7

---

1.3.5	Structure d'un système de vidéosurveillance . . . . .	8
1.4	Avantages . . . . .	8
1.5	Inconvénients . . . . .	9
1.6	Conclusion . . . . .	9
<b>2</b>	<b>Détection et suivi des objet en mouvement</b>	<b>10</b>
2.1	Introduction . . . . .	10
2.2	Représentation d'une vidéo . . . . .	11
2.3	Nature de la scène . . . . .	12
2.3.1	Séquences à caméra fixe (Scène statique) . . . . .	12
2.3.2	Séquences à caméra mobile (Scène dynamique) . . . . .	12
2.4	Détection des objets en mouvement . . . . .	12
2.4.1	Méthodes pour distinguer et détecter les objets en mouvement . . .	12
2.4.1.1	Méthodes basées sur la modélisation de l'arrière-plan . .	12
2.4.1.2	Méthodes basée sur la mesure de mouvement . . . . .	16
2.4.1.3	Méthodes intelligentes . . . . .	17
2.4.2	Etude comparative entre les différentes méthodes de détection des objets en mouvement . . . . .	22
2.5	Suivi des objet en mouvement . . . . .	23
2.5.1	Défis du suivi d'objet . . . . .	23
2.5.2	Caractéristiques visuelles pour le suivi d'objets . . . . .	24
2.5.2.1	Caractéristique de couleur . . . . .	24
2.5.2.2	Caractéristique de flux optique . . . . .	24
2.5.2.3	Caractéristique de texture . . . . .	25
2.5.2.4	Caractéristique de gradient . . . . .	25
2.5.3	Méthodes de suivi d'objets . . . . .	25
2.5.3.1	Suivi de points . . . . .	26
2.5.3.2	Suivi de Noyau . . . . .	28
2.5.3.3	Suivi de silhouette . . . . .	30
2.6	Conclusion . . . . .	32
<b>3</b>	<b>Contribuer à la détection et au suivi d'objets en mouvement</b>	<b>33</b>
3.1	Introduction . . . . .	33
3.2	Schéma général de travail . . . . .	34
3.3	Détection d'objets en mouvement . . . . .	34
3.3.1	Modèle de mixture gaussien (GMM) . . . . .	35
3.3.1.1	Modélisation une mixture de gaussiennes . . . . .	35
3.3.1.2	Initialisation du modèle . . . . .	36
3.3.1.3	Mise à jour des paramètres . . . . .	36
3.3.2	Construction de l'image de référence . . . . .	36
3.3.3	GMM amélioré . . . . .	37
3.3.3.1	Entropie 2-D de l'image . . . . .	37
3.3.3.2	Amélioration GMM par l'entropie 2D . . . . .	37
3.3.4	Suivi des objets détectés . . . . .	40
3.3.4.1	Filtre de Kalman . . . . .	40
3.3.4.2	Suivi de mouvement par filtre de Kalman . . . . .	40

3.3.4.3	Principe du filtre de Kalman . . . . .	41
3.3.4.4	Calcul de l'estimateur de Kalman . . . . .	41
3.3.4.5	Prédiction du vecteur d'état et des mesure . . . . .	42
3.3.4.6	Correction de l'état . . . . .	42
3.4	conclusion . . . . .	43
<b>4</b>	<b>Résultats et discussions</b>	<b>44</b>
4.1	Introduction . . . . .	44
4.2	Environnement de travail . . . . .	45
4.2.1	Langage de programmation python . . . . .	45
4.2.2	Caractéristique de la plateforme . . . . .	45
4.2.3	Dataset . . . . .	45
4.3	Interfaces et résultats . . . . .	46
4.3.1	Interface de l'application . . . . .	46
4.3.2	Etapes d'exécution de l'application . . . . .	48
4.3.3	Test et résultats . . . . .	52
4.4	Conclusion . . . . .	54
	<b>Conclusion générale</b>	<b>55</b>
	<b>Bibliographie</b>	<b>vii</b>

## TABLE DES FIGURES

1.1	Vidéosurveillance intelligente . . . . .	4
1.2	Visualisation centralisée du système . . . . .	5
1.3	Visualisation du système distribué . . . . .	6
1.4	Les fonctions exécutées dans les systèmes de vidéosurveillance avec leurs niveaux d'applications correspondants. . . . .	7
1.5	Système de vidéosurveillance . . . . .	8
2.1	Représentation compacte d'une vidéo. t représente l'axe temporel. . . . .	11
2.2	Exemple d'une séquence d'images . . . . .	11
2.3	Processus de détection du mouvement par la mise à jour de l'arrière-plan .	13
2.4	Détection de mouvement par différence temporelle . . . . .	16
2.5	: Flux optique. (a et b) Région de la bouche sur deux images consécutives. (c) Champ de vecteur de vitesses. . . . .	17
2.6	Principe de SVM . . . . .	19
2.7	Schéma du perceptron . . . . .	21
2.8	Schéma du perceptron multi couches . . . . .	22
2.9	Quelques difficultés de suivi d'objets . . . . .	24
2.10	les méthodes de suivi d'objets. . . . .	26
2.11	Différentes contraintes de mouvement. (a) la proximité, (b) vitesse maxi- male (r désigne le rayon), (c) changement de vitesse du centre,(d) contraintes de rigidité. désigne la position de l'objet à l'image t - 2, désigne la position de l'objet à l'image t - 1, et enfin × désigne la position de l'objet à l'image t.	27
2.12	Suivi des caractéristiques en utilisant le suivi KLT . . . . .	29
3.1	Schéma général du travail . . . . .	34
3.2	Organigramme a été amélioré de la méthode de modèle de mixture gaussien	39
3.3	Le cycle du filtre de Kalman . . . . .	41
4.1	Vidéos de la base de données LASIESTA . . . . .	46
4.2	Interface de l'application . . . . .	46
4.3	Interface principale de l'application . . . . .	47
4.4	Interface principale de l'application . . . . .	48
4.5	Résultat de détection et suivi en utilisant le GMM par vidéo . . . . .	49

4.6	Résultat de détection et suivi en utilisant le GMM amélioré par vidéo . . .	50
4.7	Résultat de détection et suivi par frames . . . . .	51
4.8	Les valeurs statistique de GMM . . . . .	52
4.9	Les valeurs statistique de GMM amélioré . . . . .	52
4.10	Résultats de quelques vidéos de la base de données LASIESTA . . . . .	53
4.11	Résultats de détection et suivi des vidéos filmées . . . . .	54

# LISTE DES TABLEAUX

2.1	Comparaison entre différentes méthodes de détection des objets en mouvement . . . . .	23
-----	---	----

## LISTES DES ACRONYMES

<b>RGB</b>	Red,Green,Blue.
<b>SVM</b>	Support Vector Machine.
<b>HOG</b>	histogram of oriented gradients.
<b>KPPV</b>	K plus proches voisins .
<b>KLT</b>	anade-Lucas-Tomasi.
<b>ACP</b>	Analyse en Composantes principales.
<b>GMM</b>	Gaussian Mixture model .

# INTRODUCTION GÉNÉRALE

La vision par ordinateur est une branche de l'intelligence artificielle et est la base de tout système de vision artificielle qui vise à permettre à une machine d'analyser, de traiter et de comprendre un certain nombre d'images capturées par le système d'acquisition (caméras, etc).

De nos jours, la détection et le suivi d'objets sont devenus l'un des sujets de recherche les plus importants et les plus difficiles en vision par ordinateur et ont été appliqués avec succès dans un large éventail d'applications du monde réel, telles que la vidéosurveillance et la robotique. La vidéosurveillance est répandue dans de nombreux secteurs d'activité (banques, transports...) ou lieux de vie (villes, immeubles de bureaux, équipements collectifs...). La détection et le suivi d'objets en mouvement en vidéo est un domaine de recherche important en vision par ordinateur et en traitement d'image, où son importance réside dans : l'amélioration de la sécurité, la facilitation de la gestion vidéo et la création de nouvelles applications basées sur l'interaction homme-ordinateur.

La détection d'objets en mouvement est la détection d'un ensemble de zones d'intérêt en mouvement dans une scène. Le suivi tente de détecter l'objet cible dans chaque image de la séquence vidéo, puis fait correspondre l'objet détecté au moment actuel avec les objets détectés dans les images précédentes afin d'établir le chemin de l'objet. Le suivi d'objets reste une tâche complexe et difficile en raison de nombreux défis tels que les distorsions de couleur, le bruit, la petite taille des objets, le contraste d'éclairage, etc. Par conséquent, de nombreux algorithmes ont été proposés pour suivre les objets en mouvement, mais il n'existe pas de méthode de suivi unique qui puisse être appliquée avec succès à toutes les vidéos.

## Problématiques et objectifs

Ce modèle générique fonctionne dans des conditions idéales, mais échoue lamentablement dans la vie réelle. Les principaux défis à relever par ce modèle sont les suivants :

- Occultation.
- Le changement graduel de l'éclairage.

- Le changement soudain de l'éclairage.
- Ombres faussement détectés comme des objets.
- Objet qui reste fixe pendant un certain temps est traité comme arrière-plan.

Dans ce projet, nous nous intéressons à des problématiques de la vidéo-surveillance, à savoir la segmentation d'objets, la détection du mouvement dans un contexte contrôlé. L'objectif du projet est de réaliser un algorithme de détection et de suivi d'objets mobiles dans une scène.

## Organisation du mémoire

Ce mémoire est divisé en quatre chapitres, qui sont organisés comme suit :

**Chapite 1 :** Il est consacré à l'étude le système de vidéo-surveillance et son architecture pour mieux comprendre les concepts de base de la vidéosurveillance.

**Chapite 2 :** Nous commençons ce chapitre en mentionnant les méthodes de détection de mouvement les plus récentes et les plus efficaces, puis nous mentionnons les méthodes de suivi.

**Chapite 3 :** Dans ce chapitre, nous allons présenter notre contribution pour le système de détection et de suivi des objets en séquence vidéo.

**Chapite 4 :** Dans ce dernier chapitre, nous avons détailler les différentes configurations nécessaires à la création du projet (langage de programmation, DataSet utilisée) et présentant l'ensemble des résultats expérimentaux obtenus pour la détection et le suivi d'objets en mouvement.

# CHAPITRE 1

## VIDÉOSURVEILLANCE

### 1.1 Introduction

De nos jours, la vidéosurveillance constitue l'une des solutions de sécurité les plus anciennes et répandues, jouent un rôle de plus en plus important dans la surveillance à distance des personnes, des biens et des sites publics et privés, ce qui a permis le développement de systèmes de vidéosurveillance.

Le premier système de vidéosurveillance a été conçu en Allemagne en 1942, par Siemens AG à des fins de contrôle de missiles . Depuis lors, les systèmes de vidéosurveillance se sont considérablement développés, se développant à la fois techniquement et économiquement. la vidéosurveillance est devenue un maillon essentiel des politiques gouvernementales de sécurité. Cette évolution répond au besoin de sécurité de tous les citoyens face à l'augmentation de la délinquance et de la criminalité. Dans ce chapitre, nous allons donner une définition d'un système de vidéosurveillance, les différents domaines d'utilisations, sa structure générale, puis quelques avantages et inconvénients.

## 1.2 Définition de la vidéosurveillance

La vidéosurveillance est un système de caméras qui sont placées dans un endroit à surveiller. Ces caméras sont connectées à un système informatique qui permet le traitement et l'analyse des données entrantes.

### — Vidéosurveillance intelligente

La vidéosurveillance intelligente est une technologie qui permet à un logiciel d'identifier automatiquement certains objets, comportements ou situations dans une séquence vidéo.

Les systèmes de vidéosurveillance intelligente utilisent des algorithmes mathématiques pour détecter des objets en mouvements dans l'image et filtrer les mouvements non pertinents.



FIGURE 1.1 – Vidéosurveillance intelligente

## 1.3 Classifications des systèmes de vidéosurveillance

Les systèmes de surveillance peuvent être classés selon divers critères.

### 1.3.1 Classification suivant le niveau d'automatisation

Les systèmes de surveillance sont classés en trois types, en exploitant des techniques avancées d'analyse vidéo et d'intelligence artificielle [1].

#### 1.3.1.1 Systèmes de surveillance manuels

Il se caractérise par la présence du opérateur humain dont la tâche est de surveiller directement les écrans ou d'utiliser les enregistrements de suite.

#### 1.3.1.2 Systèmes de surveillance semi-autonomes

Il combine le traitement vidéo avec l'intervention humaine, c'est-à-dire l'enregistrement de mouvements inattendus et leur envoi pour analyse par un expert humain.

### 1.3.1.3 Systèmes de vidéosurveillance autonomes (Systèmes de vidéosurveillance intelligents)

il Analyse intelligemment les données vidéo sans intervention humaine, en obtenant les caractéristiques suivantes :

- l'utilisation sans le contrôle humain
- la prédiction des événements, des comportements, des mouvements
- le suivi, le contrôle et l'alerte en cas d'activités imprévues

### 1.3.2 Classification suivant l'architecture réseau

Il existe deux types [1] :

#### 1.3.2.1 Architecture centralisée

Dans une architecture centralisée, tous les traitements sont effectués dans la même station de contrôle (Figure 1.2). L'encodage, l'enregistrement, le visionnement et l'analyse des flux vidéo nécessitent une grande puissance de calcul.



FIGURE 1.2 – Visualisation centralisée du système [1]

#### 1.3.2.2 Architecture distribuée

Dans une architecture distribuée les traitements sont répartis dans les différents nœuds du système de vidéosurveillance (Figure 1.3). Ainsi, les calculs nécessaires à l'analyse peuvent être faits sur des caméras intelligentes dotées de processeurs, ou dans les encodeurs.

### 1.3.3 Classification suivant le domaine d'application

Les applications de vidéosurveillance peuvent être divisées en cinq catégories en fonction de leurs objectifs [2].

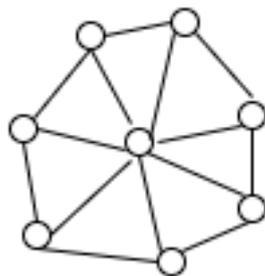


FIGURE 1.3 – Visualisation du système distribué  
[1]

### 1.3.3.1 Protection et confidentialité

Il est largement utilisé pour protéger les personnes, les lieux, la surveillance à domicile [3,4] et aussi par les gouvernements pour la sécurité intérieure [5], et la sécurité des lieux publics (musée, aéroport, gare, banque, etc. .) [6] . Il est également utilisé pour surveiller et contrôler les personnes âgées [7,8], etc.

### 1.3.3.2 Analyse d'objets

peut être utilisée pour découvrir les parcours des personnes grâce au traçage [9,10]. ou dans la surveillance d'environnements complexes comme la surveillance des activités des travailleurs dans les usines [11] ou encore dans la navigation autonome [12].

### 1.3.3.3 Reconnaissance de l'objet

Les applications de vidéosurveillance détectent l'identité d'objets en mouvement (piétons ou véhicules par exemple) en détectant des éléments distinctifs tels que : la reconnaissance faciale [13,14], ou la classification des véhicules en mouvement [15].

### 1.3.3.4 Surveillance du trafic

Des systèmes avancés facilitent la gestion, la sécurité et l'analyse du trafic sur les réseaux routiers tels que l'estimation des débits de véhicules, le contrôle de la vitesse des véhicules et le calcul de la densité du trafic sur les autoroutes [16], le trafic aérien [17] et le trafic maritime [18].

### 1.3.3.5 Détection d'événements anormaux

Ces systèmes surveillent les environnements, détectent les événements anormaux et alertent dans certains cas, tels que : détection d'incendie [19], détection d'accident [20], crime [21], détection de fraude [22], etc.

## 1.3.4 Classification suivant le niveau de l'application

les systèmes de vidéosurveillance peuvent être regroupés selon les tâches accomplies en quatre catégories, comme illustré dans (Figure 1.4).

### 1.3.4.1 Premier niveau

Cette catégorie comprend les applications de bas niveau des systèmes de vidéosurveillance, fonctionnant uniquement avec la détection d'objets. Dans la plupart des systèmes, les caméras utilisées sont supposées fixes.

### 1.3.4.2 Second niveau

Consiste à suivre l'objet après sa détection, visant à détecter la trajectoire d'un objet en mouvement dans la scène.

### 1.3.4.3 Troisième niveau

Ces applications indiquent la classification et / ou les métadonnées. Il fonctionne pour classer les éléments détectés par le système de vidéosurveillance en général à différentes catégories : homme, voiture, animal, etc [23].

### 1.3.4.4 Quatrième niveau

Représenté dans l'analyse et la compréhension du comportement et est la tâche la plus élevée utilisée dans les applications de vidéosurveillance. C'est une étape essentielle dans laquelle les informations des fonctions de niveaux inférieurs sont combinées et interprétées par une description sémantique de haut niveau .

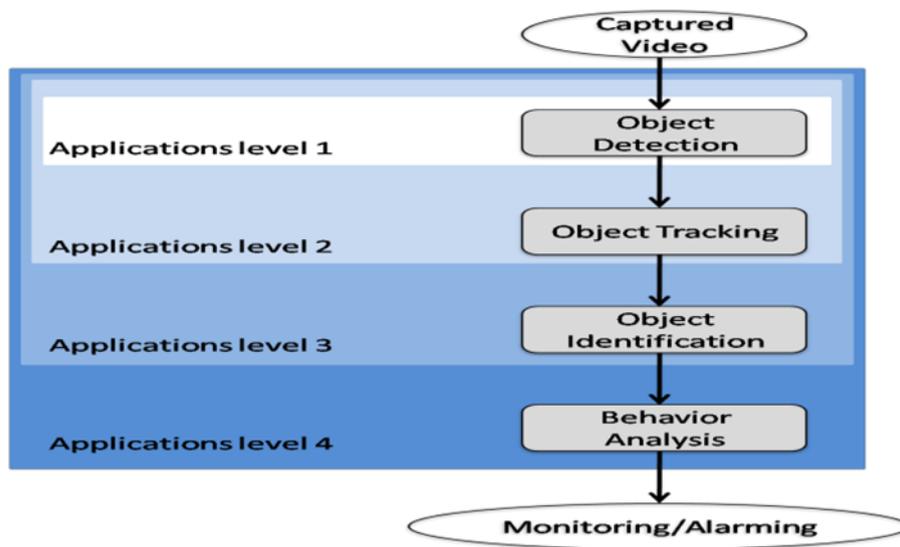


FIGURE 1.4 – Les fonctions exécutées dans les systèmes de vidéosurveillance avec leurs niveaux d'applications correspondants.

[24]

### 1.3.5 Structure d'un système de vidéosurveillance

Un système de vidéosurveillance est composé de 3 modules comme indiqué sur (Figure 1.5) :

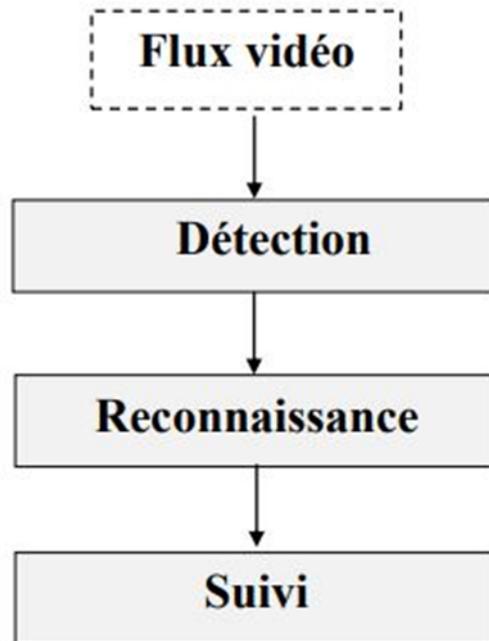


FIGURE 1.5 – Système de vidéosurveillance  
[25]

- Détection : La détection est la première étape d'un système de vidéosurveillance, elle a pour but d'identifier les objets en mouvement et les individus présents dans la séquence vidéo pour une analyse ultérieure.
- Reconnaissance : Il Permet à reconnaître d'un objet spécifique parmi un ensemble d'objets dans une scène.
- Suivi : Un processus dont le but de déterminer la position de chaque primitive ou région dans l'image à chaque instant.

## 1.4 Avantages

- Éléments de preuves pour poursuivre en justice les auteurs de crimes ou de délits.
- La vidéosurveillance est une mesure de sécurité fantastique qui peut aider à protéger les entreprise contre les voleurs, les intrus ou les vandales.
- installation du système de surveillance vidéo aide surveiller n'importe quoi par les enregistrements vidéo et audio. [26]
- Aider à développer au lieu de travail et à améliorer la productivité.
- suivre en temps réel ou en différé les opérations et activités.

## 1.5 Inconvénients

- Difficulté d'installation de certains systèmes de surveillance car cela nécessite des paramètres supplémentaires.
- Le coût.
- Si vous avez opté pour un système de caméras à alimentation électrique, vous pourriez vous retrouver sans surveillance en cas de panne.

## 1.6 Conclusion

Dans ce chapitre, nous avons défini qu'est ce qu'une vidéosurveillance, structure d'un système de vidéosurveillance , ensuite Classifications des systèmes de vidéosurveillance et on a présenté quelques avantages et inconvénients de la vidéosurveillance.

Il existe plusieurs méthodes de détection et suivi de mouvement dans la vidéo surveillance, qui seront l'objet du prochain chapitre.

## CHAPITRE 2

# DÉTECTION ET SUIVI DES OBJET EN MOUVEMENT

### 2.1 Introduction

Ces dernières années, le problème de la détection et du suivi d'objets est devenu l'un des problèmes les plus étudiés. Ils sont des tâches importantes et difficiles dans de nombreuses applications de vision par ordinateur telles que la robotique, la vidéosurveillance [27, 28]. La détection d'objet consiste à localiser l'objet dans chacune des trames d'une séquence vidéo. Le suivi d'objet est le processus de localisation spatiotemporelle d'un objet en mouvement au cours d'une séquence vidéo. Chaque méthode de suivi d'objet nécessite un mécanisme de détection d'objet, soit dans chaque trame ou lorsque l'objet apparaît d'abord dans la vidéo.

Dans ce chapitre, nous allons citer les différentes méthodes existes pour détecter un objet en mouvement, puis nous présenterons les méthodes de suivi des objets

## 2.2 Représentation d'une vidéo

Une vidéo peut être considérée comme une séquence d'images (Figure 2.1) dont chacune contient une vue statique des scènes qui s'y déroulent.

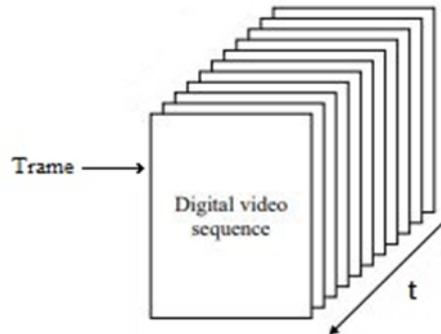


FIGURE 2.1 – Représentation compacte d'une vidéo.  $t$  représente l'axe temporel.

**Une séquence d'images :** est une succession d'images, ou trames, bidimensionnelles qui montrent l'évolution temporelle d'une scène. Sont prises dans des intervalles réguliers et très courts (30 images par seconde). Les trames sont des images à deux dimensions  $X = (x, y)$  et l'axe temporel  $t$  constitue la troisième dimension.

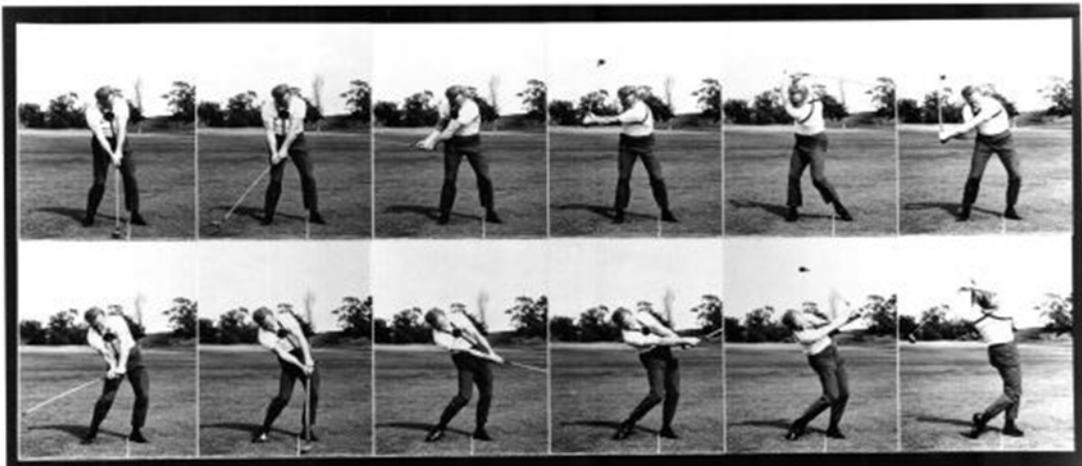


FIGURE 2.2 – Exemple d'une séquence d'images  
[29]

## 2.3 Nature de la scène

Il existe deux types [29] :

### 2.3.1 Séquences à caméra fixe (Scène statique)

Les séquences vidéo générées par une caméra fixe sont relativement stables, c'est-à-dire que les changements temporels visibles dans l'image correspondent généralement aux objets d'intérêt. La grande majorité des systèmes tels que les systèmes de vidéosurveillance reposent sur ce type de caméra par rapport à la scène filmée.

### 2.3.2 Séquences à caméra mobile (Scène dynamique)

La caméra en mouvement fait perdre à la scène sa stabilité donc on ne peut pas supposer une scène statique et il est très difficile de modéliser les différences qui existent. Le mouvement dominant de la scène appliqué à tous les pixels est considéré comme étant le mouvement de la caméra et les pixels ne suivant pas ce mouvement global forment les objets en mouvement.

## 2.4 Détection des objets en mouvement

La détection du mouvement constitue la première étape d'un système de vidéosurveillance, elle permet de détecter les objets mobiles sur la scène. C'est une étape critique et difficile car, elle doit être robuste aux variations de la luminosité de la scène et la présence des ombres. Il existe différentes méthodes de détection du mouvement que l'on peut donner certaines d'entre elles par la suite.

### 2.4.1 Méthodes pour distinguer et détecter les objets en mouvement

Les méthodes de détection du mouvement peuvent être considérées comme un prétraitement permettant de réduire la quantité d'informations à analyser. Un objet en mouvement est détecté si sa position change relativement à celle d'un ensemble d'objets statiques ou s'il est localisé dans une image acquise à l'instant  $t$  de la séquence à une position différente de celle qu'il occupait dans l'image précédente. Le mouvement d'un objet dans la séquence d'images peut donc être mis en évidence en comparant le contenu d'au moins deux images issues de la séquence. Il existe différentes méthodes de détection du mouvement :

#### 2.4.1.1 Méthodes basées sur la modélisation de l'arrière-plan

L'image de l'arrière-plan d'une scène contient les éléments statiques de cette scène, c'est la représentation de l'environnement dans lequel les objets évoluent. Cette image est considérée comme une image de référence qui se soustrait de chaque image de la séquence afin d'extraire les objets en mouvement. Plusieurs méthodes de détection de mouvement

par soustraction d'arrière-plan ont été proposées. Elles présentent pour la plupart un modèle de l'arrière-plan qui est mis à jour.

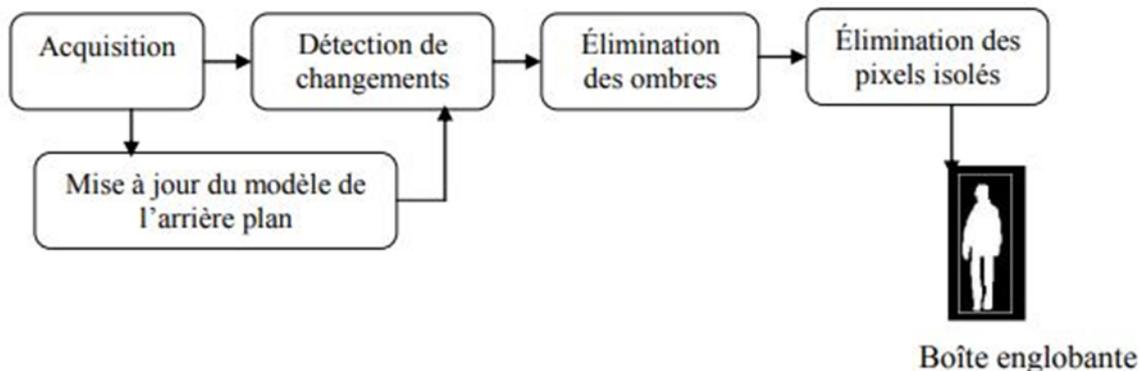


FIGURE 2.3 – Processus de détection du mouvement par la mise à jour de l'arrière-plan [30]

Cette méthode dans sa version simple, consiste à faire la différence des intensités des pixels  $I(x,y,t)$  d'une image  $I$  prise à l'instant  $t$ , par rapport aux intensités des pixels  $B(x,y)$  d'une image de référence d'arrière-plan  $B$

si  $|I(x,y,t) - B(x,y)| > \tau$  alors le pixel  $(x,y)$  de  $I$  à l'instant  $t$  est en mouvement

sinon le pixel  $(x,y)$  appartient à l'arrière-plan

Avec  $\tau$  un seuil fixé manuellement.

L'arrière-plan est non uniforme à travers la séquence d'images, à cause du bruit du capteur et la présence de petits mouvements dans l'arrière-plan ainsi que des changements graduels de la luminosité. Pour pallier à cette non-uniformité, plusieurs méthodes de modélisation de l'arrière-plan ont été proposées.

**2.4.1.1.1 Méthode de la moyenne temporelle :** Cette méthode consiste à calculer un modèle d'arrière-plan qui est continuellement mis à jour. Initialement on calcule l'image moyenne temporelle à partir de  $M$  premières images de la séquence. Cette image moyenne sera considérée comme l'arrière-plan initial de la scène :

$$B(x, y, M) = \frac{1}{M} \sum I(x, y, t) \quad (2.1)$$

où  $I(x, y, t)$  : est la valeur du pixel  $(x, y)$ , sur l'image  $I$  au temps  $t$  et  $B(x, y, t)$  : est la valeur du pixel  $(x, y)$ , sur l'image arrière-plan au temps  $t$ .

L'arrière-plan est continuellement mis à jour pour les pixels appartenant à l'arrière-plan dans l'image courante. On considère que chaque image de la séquence contribue à la mise à jour de l'arrière-plan :

$$B(x, y, t) = \frac{(t-1)B(x, y, t-1) + I(x, y, t)}{t} \quad (2.2)$$

On considère qu'il y a du mouvement si :  $| I(x, y, t) - B(x, y, t) | > \tau$   
 Où  $\tau$  est un seuil fixé manuellement.

**2.4.1.1.2 Méthode de distribution gaussienne simple :** Cette méthode est plus adaptée à la détection d'objets en mouvement lorsque l'arrière-plan contient des instabilités qui proviennent des changements dans l'illumination de la scène ou des mouvements d'objets non stationnaires [31]. Cette méthode prend en considération le bruit du capteur qui rend l'arrière-plan non uniforme, ainsi que les changements graduels de la luminosité de la scène.

Le bruit du capteur est considéré comme un bruit gaussien de paramètres estimés

$\sigma_{cam}^2 = (\sigma_{rcam}^2, \sigma_{gcam}^2, \sigma_{bcam}^2)$ , ( $\sigma_{cam}^2$  :le bruit de caméra,  $\sigma_{rcam}^2$  :le bruit de caméra selon le canal rouge,  $\sigma_{gcam}^2$  :le bruit de caméra selon le canal vert,  $\sigma_{bcam}^2$  :le bruit de caméra selon le canal bleu), selon les 3 canaux RGB. De plus, chaque pixel de l'image de référence (c-à-d arrière-plan) est modélisé sous forme d'une distribution gaussienne afin de tenir compte du bruit et l'image de référence est mise à jour afin de tenir compte des changements graduels de la luminosité. En considérant une trame de vidéo à l'instant t, l'estimation de l'arrière-plan par cette méthode s'effectue en calculant de façon récursive la moyenne  $\mu_t$  et la covariance  $\sigma_t$  des intensités des pixels sur un nombre n de trames précédentes par l'équation suivant :

$$\mu_{t+1} = \alpha\mu_t + (1 - \alpha)X_{t+1} \quad (2.3)$$

$$\sigma_{t+1}^2 = \alpha(\sigma_t^2 + (\mu_{t+1} - \mu_t)^2) + (1 - \alpha)(X_{t+1} - \mu_{t+1})^2 \quad (2.4)$$

Dans cette équation,

$X_{(t+1)}$  est l'intensité courante du pixel dans l'espace de couleur

$\mu_{(t+1)}$  est la moyenne précédente,

$\alpha$  est un paramètre de contrôle du taux d'adaptation ( $0 < \alpha < 1$ )

Les différentes valeurs que prend un pixel (x,y) dans le temps constituent un processus

$X : X = \{X_1, \dots, X_T\} = \{I(x, y, t) : I \leq t \leq T\}$

Ce processus est modélisé pour chaque pixel par une distribution gaussienne de moyenne  $\mu_t = (\mu_{rt}^2, \mu_{gt}^2, \mu_{bt}^2)$  pour les 3 canaux RGB de variance  $\sigma_t^2 = (\sigma_{rt}^2, \sigma_{gt}^2, \sigma_{bt}^2)$  :

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} e^{-1/2(X_t - \mu)^T \Sigma^{-1} (X_t - \mu)} \quad (2.5)$$

où n : nombre de canaux (généralement n=3 pour RGB) et

$\Sigma = \sigma I$  (où I est la matrice identité).

Un pixel est classé en mouvement s'il satisfait à au moins une des conditions suivantes :

$$\begin{aligned} |X_{rt} - \mu_{rt}| &> 3 \max(\sigma_{rt}, \sigma_{rcam}) \\ |X_{gt} - \mu_{gt}| &> 3 \max(\sigma_{gt}, \sigma_{gcam}) \\ |X_{bt} - \mu_{bt}| &> 3 \max(\sigma_{bt}, \sigma_{bcam}) \end{aligned}$$

Les valeurs  $\sigma_{cam}$  sont initialisées à une petite valeur correspondant à un faible bruit qui peut être dû à des vibrations de la caméra (qui est fixe). La valeur 3 est choisie, car les valeurs d'une gaussienne sont centrées autour de sa moyenne et sont dans un intervalle  $[-3\sigma, 3\sigma]$  ( $\sigma$  étant l'écart type de cette gaussienne).

**2.4.1.1.3 Modélisation de l'arrière-plan par mixture de Gaussiennes :** Une autre approche améliorée a été proposée par Stauffer and Grimson [32] qui consiste à modéliser l'arrière-plan en appliquant un mélange de distributions gaussiennes. Cette approche est plus robuste aux bruits et aux petits mouvements de l'arrière-plan (par exemple : feuillage d'un arbre) et qu'il y a des changements graduels de la luminosité. Chaque pixel de l'image de référence est modélisé sous forme d'un mélange de distributions gaussiennes afin de tenir compte du bruit et des changements de l'arrière-plan. L'image de référence est mise à jour afin de tenir compte des changements de luminosité. Les différentes valeurs que prend un pixel (x,y) dans le temps constituent un processus

$X$  :  $X = \{X_i, \dots, X_T\} = \{I(x, y, t) : i \leq t \leq T\}$  Ce processus est modélisé par un mélange de K distributions gaussiennes. Ainsi, la probabilité d'avoir une intensité  $X_t$  est donnée par :

$$P(X_t) = \sum_{i=1}^K \omega_{i,t} \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (2.6)$$

$\omega_{i,t}$  est le poids de la  $i^{\text{ème}}$  distribution gaussienne du mélange de distributions à l'instant t ; cela représente la proportion des données utilisées dans le calcul de la gaussienne.

$\mu_{i,t}$  est la moyenne de la  $i^{\text{ème}}$  distribution gaussienne du mélange de distributions à l'instant t et  $\Sigma_{i,t}$  est sa matrice de covariance avec  $\Sigma_{i,t} = |\sigma_i^2|$  ( est la matrice identité).  $\eta$  Représente la fonction de densité de probabilité d'une gaussienne i :

$$\eta(X_t, \mu_{i,t}, \Sigma_{i,t}) = \frac{1}{(2\pi)^{n/2} |\Sigma_{i,t}|^{1/2}} \exp^{-\frac{1}{2}(X_t - \mu_{i,t})^T \Sigma_{i,t}^{-1} (X_t - \mu_{i,t})} \quad (2.7)$$

où :

- n : nombre de canaux (généralement n=3 pour RGB)
- $\mu_{i,t} = (\mu_{i,rt}, \mu_{i,gt}, \mu_{i,bt})$  est la moyenne de la gaussienne i à l'instant t, pour les 3 canaux RGB,
- $\sigma_t^2 = (\sigma_{rt}^2, \sigma_{gt}^2, \sigma_{bt}^2)$  sa variance,
- $\Sigma_{i,t} = \sigma_{i,t} \cdot I$  est la matrice de covariance de la gaussienne i à l'instant t , avec I la matrice identité.

Un pixel  $X_t$  provient d'un objet en mouvement si :  $|X_t - \mu_{i,t}| > 2\sigma_{i,t}$  pour les S distributions Avec  $S = \underset{b}{\operatorname{argmin}} (\sum_{K=1}^b \omega_K > \rho)$  où  $\rho$  est la proportion minimum de données choisies dans la modélisation de l'arrière-plan.

### 2.4.1.2 Méthodes basée sur la mesure de mouvement

il y a plusieurs méthodes basée sur la mesure de mouvement ,parmis ces méthodes :

**2.4.1.2.1 Méthode différentielle :** Le mouvement dans une image, peut se définir comme un changement de la valeur de l'intensité d'au moins un pixel entre deux images consécutives dans une séquence vidéo. La différence de deux images successives devrait permettre de détecter un mouvement entre les deux images.

Collins et al [33] proposent la différence temporelle entre deux images successives.  $I(x, y, t)$  la valeur de l'intensité du pixel  $(x, y)$  à l'instant  $t$ ,  $\tau(x, y, t)$  la valeur du seuil assigné au pixel  $(x, y)$  à l'instant  $t$  qui permet de détecter un changement. Un pixel est classé en mouvement si :

$$|I(x, y, t) - I(x, y, t - 1)| \geq \tau(x, y, z) \quad (2.8)$$

Cette méthodes ne fonctionne pas dans plusieurs cas, pour différentes raisons, telles que : la présence de bruit du capteur et les changements de luminosité de la scène qui modifient les intensités des pixels, elle ne permet pas de détecter le mouvement dans les zones uniformes intérieures à l'objet (Figure 2.4)

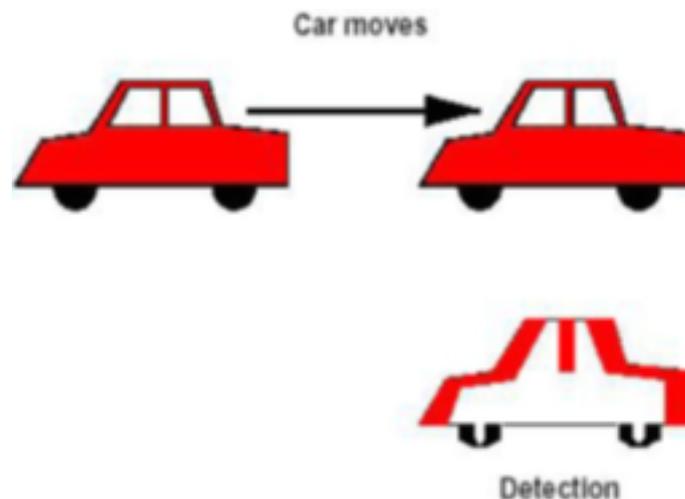


FIGURE 2.4 – Détection de mouvement par différence temporelle [33]

**2.4.1.2.2 Flux optique :** Le flux optique est un champ de vecteur des vitesses apparentes des objets de la scène sur le plan image (Figure 2.5). Il permet de déterminer le mouvement de chaque pixel appartenant à une image de la séquence d'images.



FIGURE 2.5 – : Flux optique. (a et b) Région de la bouche sur deux images consécutives. (c) Champ de vecteur de vitesses.

[34]

L'estimation du flux optique est basée sur l'hypothèse de l'intensité constante c-à-d que l'intensité observée d'un objet ne change pas lorsque celui-ci est en mouvement. Cette hypothèse est valable pour les objets à modèle de réflectance lambertienne avec un faible mouvement. Cette méthode suppose donc, que le mouvement est instantané et faible d'une image à l'autre et que les changements observés entre deux images sont uniquement causés par le mouvement. [34]

Cette caractéristique induit une sensibilité du flux optique par rapport au bruit dans l'image en plus du temps machine que la complexité de son calcul induit. De ce fait, il est peu utilisé en vidéosurveillance . [35]

### 2.4.1.3 Méthodes intelligentes

**2.4.1.3.1 Détection par apprentissage supervisé :** La détection d'objets peut s'effectuer par l'apprentissage automatique de différentes vues de l'objet contenues dans un ensemble d'exemples au moyen d'un mécanisme d'apprentissage supervisé. L'apprentissage des différentes vues de l'objet renonce à l'exigence de stocker un ensemble complet de modèles. Étant donné un ensemble d'exemples d'apprentissage, les méthodes d'apprentissage supervisé génèrent une fonction qui fait correspondre les entrées aux sorties désirées. les techniques les plus courantes sont :

1. Boosting adaptatif :

Freund et Schapire [36] a proposé la méthode Adaboost (Adaptative Boosting) , qui est basée sur le principe du booster (boosting). Adaboost est une méthode itérative de combinaison qui combine de nombreux classificateurs de base en un seul classificateur très efficace. Cette méthode est basée sur le principe de sélection itérative de classificateurs faibles sur la base d'exemples d'apprentissage.

L'algorithme de Adaboost peut être décrit en trois étapes :

- la distribution de poids pour l'ensemble d'apprentissage : Donnez à chaque échantillon d'apprentissage un poids initial de  $1/n$  où  $n$  est le nombre d'échantillons.

- Formation des classificateurs faibles : Si un échantillon particulier est classé avec précision, son poids sera réduit lors de la construction du prochain ensemble d'apprentissage. Au contraire, si un échantillon particulier n'est pas classé avec précision, son poids augmente. L'ensemble d'échantillons avec les poids mis à jour est ensuite utilisé pour former le classificateur suivant. L'ensemble du processus de formation se fait à plusieurs reprises de cette manière.
- Fusion de classificateurs faibles obtenus par entraînement en un classificateur fort.

Dans le contexte de la détection d'objet, en 2001, Viola et Jones [37] ont appliqué Adaboost dans la détection de visages. Puis en 2003, ils ont amélioré cette méthode [38] pour la détection de piéton, combinant les informations de mouvement et d'apparence.

### 2. Machines à vecteurs supports :

Les machines à vecteurs supports sont l'une des techniques d'apprentissage supervisée les plus utilisées à ce jour, qui traitent des problèmes non linéaires séparables. est utilisée pour séparer les données en classes en trouvant l'hyperplan marginal maximal qui sépare une classe des autres. L'idée principale est de trouver l'hyperplan optimal qui sépare les données en deux classes (Figure 3.2), en utilisant le principe de marge maximale : considérons des points d'apprentissage appartenant aux classes  $-1$  et  $+1$ . La marge de l'hyperplan, qui est maximisée, est définie comme la distance entre l'hyperplan et les points de données les plus proches de celui-ci. Ce sont ces points qui sont appelés "vecteurs supports". Le problème de trouver les vecteurs supports est résolu en le formulant comme un problème d'optimisation quadratique.

Dans le contexte de la détection d'objets, plusieurs méthodes ont été proposées dans la littérature " Papageorgiou et al" [39] ont utilisé SVM pour détecter les piétons et les visages dans les images. Les caractéristiques utilisées pour discriminer les classes sont extraites en appliquant des ondelettes Haar. Et il y a d'autres travaux [40, 41] ont utilisé les caractéristiques de HOG (les descripteurs d'histogramme de gradient orienté) [42] pour discriminer les classes.

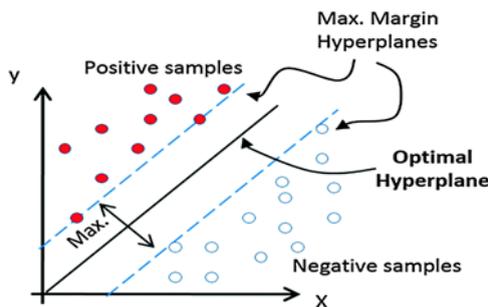


FIGURE 2.6 – Principe de SVM  
[43]

3. Méthode des K Plus Proches Voisins(*kppv*) :

KPPV est un algorithme d'apprentissage non paramétrique supervisé basé sur des instances [44]. La classification d'un nouvel objet est basée sur sa similitude avec ses plus proches voisins qui sont eux-mêmes membres de classes prédéfinies avec une étiquette donnée. [45]

La similarité entre le nouvel objet et les objets dans une classe des classes s'obtient en calculant la distance qui les sépare. phase d'apprentissage sur ces classes étiquetées permet de déterminer le paramètre K.

On dispose d'un échantillon de données constitué de paire de valeurs

$(U_1, w_1), (U_2, w_2), \dots, (U_i, w_i)$ , avec  $i = 1, \dots, n$   $U_i$  représente le descripteur de rang  $i$ , et  $w_i$  sa classe. Supposons que la distance métrique qui mesure la distance entre deux valeurs quelconques  $U_1$  et  $U_2$  est  $d(U_1, U_2)$ . Pour chaque descripteur inconnu  $U$ , l'algorithme K Nearest Neighbors calcule la valeur  $\sigma_i = d(U_i, U)$  puis affecte les indices  $\{a_1, \dots, a_k\}$  aux plus petites valeurs de  $K\sigma_i$  telle que :

$$\begin{cases} \sigma_{a_i} \leq \sigma_j \text{ pour tout } j \notin a_1, \dots, a_k, i \in 1, \dots, k \\ \sigma_{a_1} \leq \sigma_{a_2} \leq \dots \leq \sigma_{a_k} \end{cases}$$

$k_j$  le nombre de descripteurs connus qui sont les plus proches voisins de  $U$  et dont la classe est  $w_i$  par l'équation :

$$k_j = \sum_{i=1}^k I(w_{a_i} = j) \tag{2.9}$$

La fonction  $I(a = b)$  prend la valeur 1 si  $a = b$  et 0 sinon . La règle de décision pour la méthode KPPV consiste en une association avec  $U$  La désignation la plus courante parmi les  $k_j$  adjacents.

- Le nombre  $k$  est choisi entre une valeur assez grande pour réduire la sensibilité aux erreurs de reconnaissance et une valeur assez faible qui permet un effet sur le descripteur.

A la classe appropriée, on peut donner des valeurs de  $k$  supérieures ou égales à 1. Si la reconnaissance a lieu entre deux classes , Tout en évitant les valeurs individuelles qui rendent une décision impossible.

4. Méthode naïve de Bayes pour la classification :

La classification naïve bayésienne est un type de classification bayésienne probabiliste simple basée sur le théorème de Bayes avec une forte indépendance des hypothèses [44]. peuvent être entraînés efficacement dans un contexte d'apprentissage supervisé, le classifieur naïf de Bayes appartenant à la famille des classifieurs linéaires. Le théorème de Bayes qui s'exprime sous la forme suivante :

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)} \quad (2.10)$$

$P(A|B)$  : indique la probabilité a posteriori d'observer l'événement A sachant que l'événement B est réalisé.  $P(A)$  et  $P(B)$  sont les probabilités a priori des événements A et B.

D'après cette l'équation s'exprime :

$$p(w_k|U) = \frac{p(U|w_k) p(w_k)}{p(U)} \quad (2.11)$$

Tel que :

$p(w_k|U)$  : la probabilité a posteriori qu'un des cripteur  $U = (u_1, u_2, \dots, u_p)$  appartienne à une classe  $w_k$

- la règle de décision de Bayes consiste à choisir d'affecter l'individu à la classe dont la probabilité a posteriori est la plus grande. On démontre que cette décision minimise le risque d'erreur de classification c à dire un descripteur U appartiendra à une classe  $w_i$  si :

$$P(w_i|U) > p(w_j|U) \text{ avec } 1 \leq j \leq z, \text{ et } j \neq i$$

où  $z$  : le nombre de classes

- En fait, la méthode bayésienne naïve repose sur l'hypothèse forte que les éléments du descripteur  $U = (u_1, u_2, \dots, u_p)$  est conditionnellement indépendant de la classe  $w_k$  :

$$p(U|w_k) = \prod_{i=1}^n p(u_i|w_k) \quad (2.12)$$

- La règle de décision par la méthode naïve de Bayes se définit par l'équation suivante , et On peut estimer  $p(w_k)$  en prenant le rapport entre le nombre d'éléments appartenant à la classe  $w_k$  et le nombre total d'objets.

$$p(w_k|U) = \arg_{w_k} \max \prod_{i=1}^n p(u_i|w_k) p(w_k) \quad (2.13)$$

5. Réseaux de Neurones [46] :

Les réseaux de neurones artificiels sont construits à partir du modèle biologique des neurones. Leurs fondements proviennent des premiers travaux sur les réseaux de neurones qui a construit un modèle simplifié de neurones biologiques appelé le neurone formel . Ce dernier peut être représenté comme un automate à  $\eta$  ntrées et une sortie.

- Perceptron c'est le premier modèle de réseau de neurones complet (Figure 2.7). dans ce modèle ,une fonction de combinaison associe les entrées du neurone ,puis compare ce résultat à une valeur seuil , via une fonction d'activation.

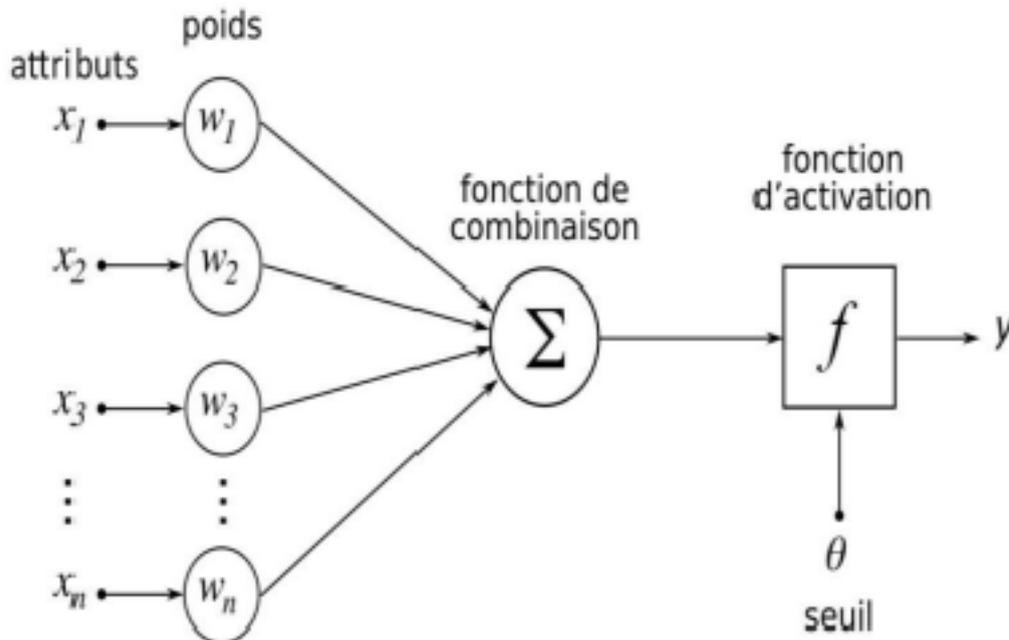


FIGURE 2.7 – Schéma du perceptron [46]

On peut représenter la sortie du réseau par l'équation suivante (avec  $f$  la fonction d'activation) :

$$y = f \left( \sum_{i=1}^n (w_i x_i) - \theta \right) \quad (2.14)$$

La fonction d'activation renvoie une valeur binaire , les paramètres importants de ce modèle sont les coefficient synaptiques(poids) , le seuil , et la façon de les ajuster lors de l'apprentissage .

Il faut choisir un mécanisme permettant de calculer ces paramètres et de les faire converger vers une valeur assurant une classification aussi proche que possible de l'optimale , lors de la phase d'apprentissage .

- le modèle du perceptron multicouche [46] est une amélioration de modèle perceptron .il reprend le principe du perceptron , en ajoutant plusieurs vecteurs de poids appelées couches cachées (Figure 2.9) .

Les neurones d'une couche sont reliés aux neurones des couches adjacentes par des liaisons pondérées . Ainsi , le poids de chacune de ces liaisons est l'élément clef du fonctionnement du réseau .

On peut considérer que la fonction d'activation est la même à chaque couche , mais qu'elle varie d'une couche à l'autre.

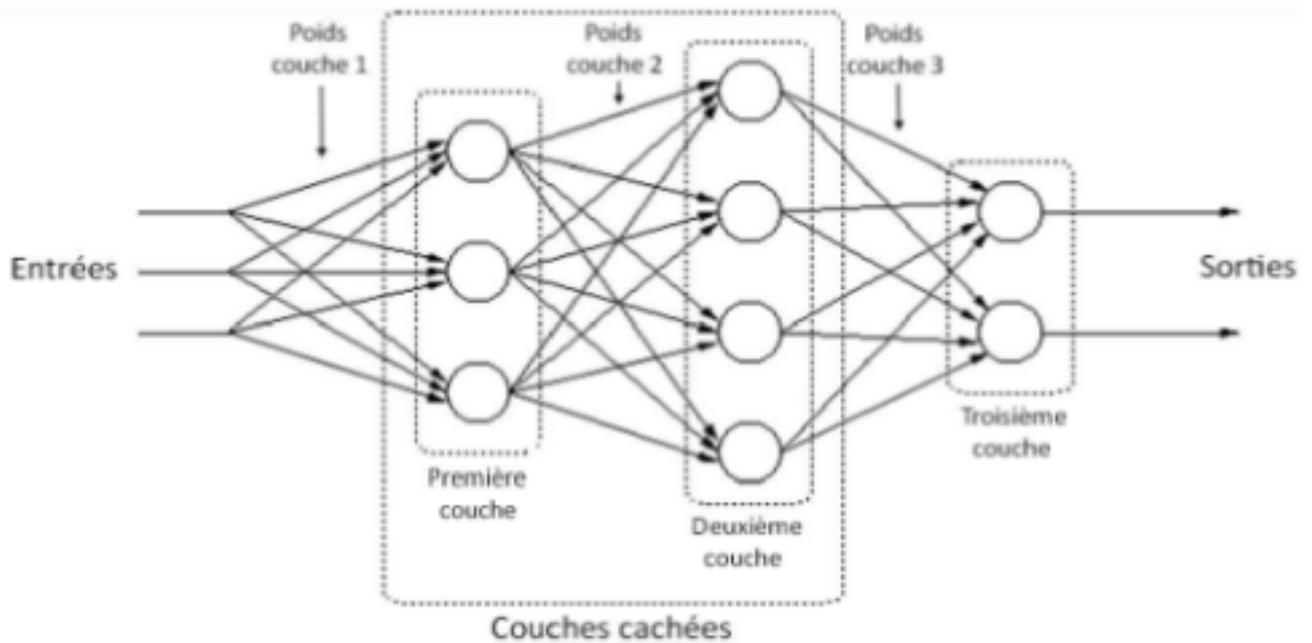


FIGURE 2.8 – Schéma du perceptron multi couches [46]

**2.4.1.3.2 Détection par apprentissage non supervisé :** l'apprentissage non supervisé désigne la situation d'apprentissage automatique où les données ne sont pas étiquetées, la tâche de classification non-supervisée a pour but d'organiser un ensemble d'objets sans connaissance a priori d'une classification, afin de simplifier, ou structurer les données par l'extraction ou la reconnaissance des "formes" récurrentes dans ces données [47].

## 2.4.2 Etude comparative entre les différentes méthodes de détection des objets en mouvement

Il n'existe pas d'unité de mesure précise et définie pour comparer les performances des différentes méthodes de détection d'objet en mouvement. Dans ce chapitre notre étude comparative est basée sur les critères suivants :

- Le type d'arrière plan (statique, dynamique) ;
- Complexité (+++ bonne performance, ++ performance moyenne, + faible performance) ;
- Application de méthode (Temps réel, Retardement) ;
- Efficacité (+++ bonne performance, ++ performance moyenne, + faible performance).

Le tableau 2.1 présente les résultats comparatifs entre les quatre méthodes définies précédemment.

Méthodes		Arrière plan	Complexité	Application de méthode	Efficacité
Modélisation de l'arrière-plan	La moyenne temporelle	Statique	+	Temps réel	+
	Gaussienne simple	Statique	+	Temps réel	+
	GMM	Statique	+	Temps réel	++
Mesure de mouvement	Défférentielle	Statique	++	Temps réel	++
	Flot optique	Statique	+++	Retardement	++
Intelligent	Apprentissage supervisé	Dynamique	++	Retardement	+++
	Apprentissage non supervisé	Dynamique	+++	Retardement	+++

TABLE 2.1 – Comparaison entre différentes méthodes de détection des objets en mouvement

## 2.5 Suivi des objet en mouvement

Après la détection des objets en mouvement, leurs déplacements sont suivis tout au long de la séquence vidéo. Le suivi est l'estimation de la trajectoire d'un objet dans le plan image comme il se déplace dans la scène. Cette tâche requiert localiser chaque objet à partir d'une image à l'autre. Le suivi peut être fait en 2D, à partir d'une seule caméra, ou 3D, en combinant deux vues ayant une relation géométrique connue.

### 2.5.1 Défis du suivi d'objet

Bien que le suivi d'objets soit étudié depuis plusieurs décennies et que des progrès significatifs aient été réalisés ces dernières années [48, 49], il reste un problème très difficile [50]. En effet, de nombreux facteurs affectent les performances de l'algorithme de suivi d'objet. Parmi ces facteurs, nous mentionnons les suivants :

- Changements d'illumination
- Occultations partielles ou totales
- Similarité de couleur entre l'objet cible et le fond
- Mouvement de caméra
- Rotation dans le plan de l'image
- Objets non-rigides et/ou articulés
- Objet de petite taille
- Objet en mouvement rapide
- Présence de bruit dans les images
- Nécessité d'un suivi en temps réel.



FIGURE 2.9 – Quelques difficultés de suivi d’objets [43]

## 2.5.2 Caractéristiques visuelles pour le suivi d’objets

La sélection des bonnes fonctionnalités joue un rôle essentiel dans le suivi. En général les fonctionnalités qui permettent de mieux distinguer plusieurs objets et entre un objet et un arrière-plan sont plus efficaces pour le suivi d’objet. Les détails des caractéristiques visuelles les plus connues et les plus utilisées sont les suivantes :

### 2.5.2.1 Caractéristique de couleur

Les couleurs sont les repères les plus visibles pour l’œil humain, elles sont donc les plus couramment utilisées pour représenter des objets. En traitement d’image, l’espace colorimétrique RVB (rouge, vert, bleu) est généralement utilisé pour représenter la couleur. Cependant, l’espace RVB n’est pas un espace de couleur perceptuellement uniforme, c’est-à-dire que les différences entre les couleurs dans l’espace RVB ne correspondent pas aux différences de couleur perçues par les humains [51]. La transformation de l’information de l’espace RGB à un espace couleur différent a pour objectif de séparer l’information dans les canaux pertinents.

### 2.5.2.2 Caractéristique de flux optique

Le flux optique est un champ dense de vecteurs de déplacement décrivant le mouvement de chaque pixel dans une région d’image. En supposant une luminosité constante pour le même pixel entre des images successives, le flux optique est calculé à partir de l’extraction d’un champ de vitesse dense à partir d’une séquence d’images. Le flux optique est couramment utilisé comme caractéristique dans les applications de segmentation spatio-temporelle et de suivi basées sur le mouvement [52].

### 2.5.2.3 Caractéristique de texture

La texture est une mesure du changement de densité de surface qui détermine des propriétés telles que le lissé et l'uniformité. Par rapport à la couleur, la texture nécessite une étape de traitement pour créer les descripteurs. Il existe de nombreux descripteurs de tissus. L'un des tissus les plus étudiés est le filtre de Gabor [53].

### 2.5.2.4 Caractéristique de gradient

les caractéristiques de gradient ont été prouvées utiles dans la détection humaine, il existe deux catégories de ces caractéristiques [54].

la catégorie principale des méthodes basées sur l'information de gradient consiste à utiliser la forme et le contour pour représenter les objets, tels que le corps humain. Cette information est extraite à partir de l'analyse spatiale de l'intensité lumineuse de l'image. Une propriété importante du gradient est sa sensibilité plus faible aux changements d'illumination comparée aux caractéristiques couleurs.

## 2.5.3 Méthodes de suivi d'objets

Plusieurs méthodes de suivi d'objets ont été proposées, la différence entre ces méthodes réside dans le choix de la représentation et de la forme des objets, les caractéristiques de l'image utilisée, la nature du mouvement estimé, etc. Dans [55], les algorithmes de suivi d'objets sont classés en 4 catégories : algorithmes basés sur des régions, basés sur des contours actifs, basés sur des caractéristiques et basés sur un modèle.

Dans [56], les algorithmes de suivi sont classés selon la représentation de l'objet cible. Sa classification comprend trois classes de méthodes : elle suivi par points, le suivi par noyau et le suivi par silhouette (Figure 2.10). Il existe d'autres classifications de suivi d'objets [57] et [58], ici nous discuterons de la classification des algorithmes de suivi proposés dans [56]. Parce que c'est un travail qui décrit les méthodologies de suivi des objets modernes de manière très claire et compréhensible.

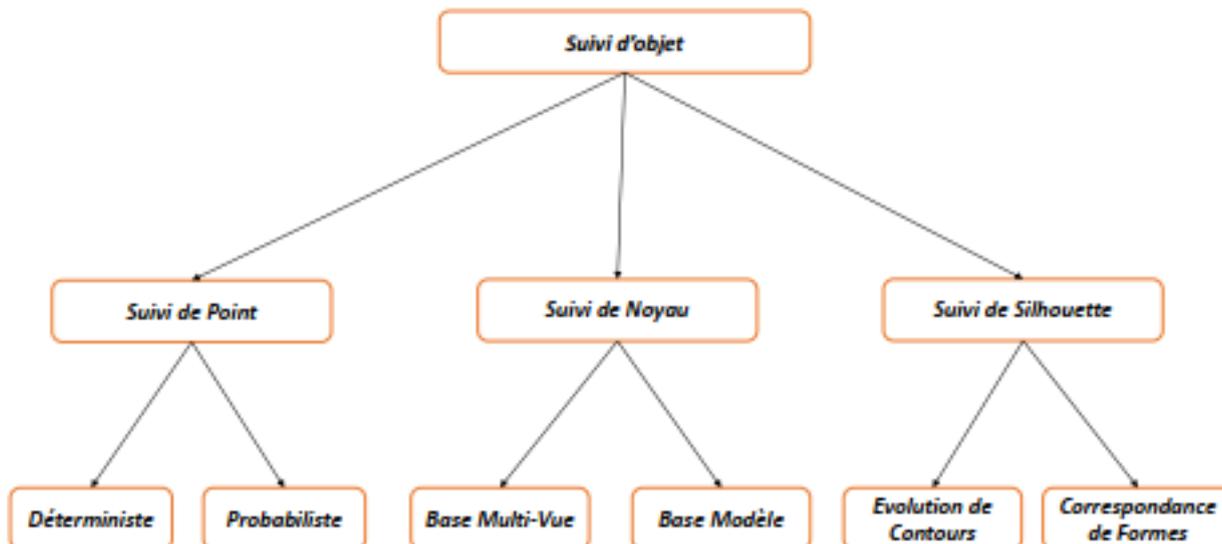


FIGURE 2.10 – les méthodes de suivi d’objets.

### 2.5.3.1 Suivi de points

Le suivi d’objet peut être formulé comme un problème de mise en correspondance d’objets représentés par des points d’une trame sur l’autre. La correspondance de point est un problème complexe, particulièrement en présence de bruit dans les images, les occultations partielles ou totales, les erreurs de détection de points, etc. Globalement, les méthodes de correspondance de points peuvent être divisées en deux grandes catégories :

#### 1. Méthodes déterministes

Dans les méthodes déterministes, la traçabilité est réalisée en réduisant le coût d’association entre les objets de la trame précédente et chaque objet unique de la trame courante. Le coût est un ensemble de contraintes, parmi lesquelles on trouve :

- La proximité suppose que l’emplacement de l’objet ne change pas sensiblement d’une image à l’autre (Figure 2.11(a)).
- vitesse maximale définit une limite supérieure pour la vitesse de l’objet et limite les correspondances possibles au voisinage circulaire de l’objet (Figure 2.11(b)).
- Une faible variation de la vitesse (mouvement régulier) suppose que la direction et la vitesse de l’objet ne changent pas radicalement (Figure 2.11(c)).
- La rigidité suppose que les objets du monde 3D sont rigides. Par conséquent, la distance entre deux points de l’objet réel reste inchangée ( Figure 2.11(d)).

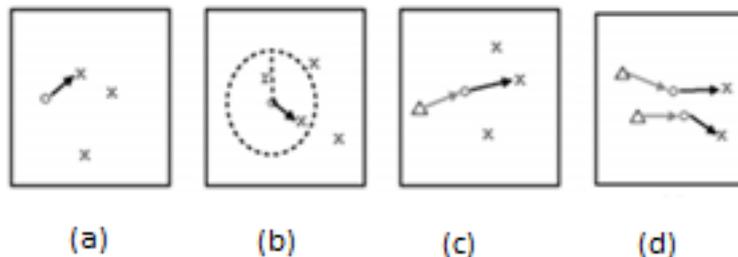


FIGURE 2.11 – Différentes contraintes de mouvement. (a) la proximité, (b) vitesse maximale ( $r$  désigne le rayon), (c) changement de vitesse du centre, (d) contraintes de rigidité.  $\Delta$  désigne la position de l'objet à l'image  $t - 2$ ,  $\circ$  désigne la position de l'objet à l'image  $t - 1$ , et enfin  $\times$  désigne la position de l'objet à l'image  $t$ .

ces contraintes ne sont pas spécifiques aux méthodes déterministes et qu'elles peuvent également être utilisées dans le cadre du suivi de points par des méthodes statistiques.

Pour résoudre le problème d'appariement il existe plusieurs algorithmes, Sethi et Jain [59] résolvent le problème de mise en correspondance en utilisant l'approche gloutonne (greedy) basée sur les contraintes de proximité et de rigidité. L'algorithme gloutonne considère deux trames consécutives, et est initialisé par une recherche des plus proches voisins. Les correspondances sont échangées itérativement pour minimiser le coût. Il existe d'autres solutions au problème d'appariement [60,61] [62]

## 2. Méthodes probabilistes

Les méthodes de correspondance statistique résolvent des problèmes de suivi tels que le bruit causé par l'obtention de données à partir de capteurs vidéo, la possibilité de mouvements d'objets soumis à des perturbations aléatoires, par exemple lors de la manœuvre de véhicules. En tenant compte des incertitudes de modèle et de mesure lors de l'estimation de l'état de l'objet. Les méthodes de correspondance statistique utilisent l'approche de l'espace d'état pour modéliser les propriétés de l'objet telles que la position, la vitesse et l'accélération. Les mesures consistent généralement en la position de l'objet dans l'image, qui est obtenue par un mécanisme de détection. Le filtre de Kalman et le filtre à particules parmi les méthodes d'estimation de l'état dans le contexte du suivi des points [63].

**Filtre de Kalman** est utile pour suivre des types distincts d'objets en mouvement [64]. Il a été initialement créé par Rudolf Kalman à la NASA. Un filtre de Kalman est utilisé pour estimer l'état d'un système linéaire où l'état est supposé être distribué par une gaussienne [63]. Il s'agit d'un estimateur optimal de processus aléatoires. Le filtre de Kalman est un filtre récursif à deux étapes. Les deux phases sont prédire et mettre à jour [65]. Dans la phase de prédiction, l'emplacement actuel de l'objet en mouvement est estimé ou prédit sur la base de l'observation précédente

effectuée. A titre d'illustration, si un objet se déplace avec une répartition cohérente, l'emplacement actuel de l'objet à  $X_t$  peut être basé sur son emplacement précédent,  $X_{t-1}$ . Dans la mise à jour phase la capacité de l'emplacement actuel de l'objet est mise en commun avec l'emplacement prédit et acquiert la projection a posteriori position actuelle de l'objet [66].

Le filtre de Kalman a été largement utilisé dans les algorithmes de suivi d'objets. Parmi l'utilisation de filtre kalman : Broida et Chellappa. [67] ont utilisé le filtre de Kalman pour suivre les points dans des images bruyantes. Dans le suivi d'objet par caméra stéréo, Beymer et Konolige. [68] utilisent le filtre de Kalman pour prédire la position et la vitesse de l'objet dans les dimensions x-z.

Malgré la popularité du filtre de Kalman, cet algorithme a des limites parmi lesquelles, filtre de Kalman est l'hypothèse que l'état les variables sont normalement distribuées (gaussiennes). Ainsi, le filtre de Kalman donnera une mauvaise estimation des variables d'état qui ne suivent pas la distribution gaussienne [63]

**Cette limitation peut être surmontée en utilisant le filtrage particulaire** [69]. Le filtre à particules est une généralisation du filtrage de Kalman dans lequel la variable d'état n'est plus décrite par une gaussienne, un filtre bayésien technique d'échantillonnage séquentiel qui est une approche récursive de la distribution à l'aide d'un ensemble fini d'essais pondérés. Il se compose également de deux phases : prédiction et mise à jour comme le filtrage de Kalman. Il est appliqué dans des domaines en développement tels que la vision par ordinateur communal et appliqué à la problématique du suivi. Arnaud et al. [70] ont proposé d'appliquer le filtrage particulaire qui permet une bonne qualité de suivi, même pour des variables d'état de distribution variable.

**Filtre de Bayes** nécessite une méthode d'association, qu'il utilisée pour le suivi multi-objets. Le but des méthode d'associations : relier les observations fournies par le détecteur aux trajectoires des objets (pistes). Parmi les méthodes statistiques d'association : PDAF est utilisé par Chang et Aggarwal [71] pour effectuer une reconstruction de structure 3D à partir d'une séquence vidéo, Rasmussen et Hager [72] utilisent un filtre JPDAF contraint pour suivre les régions. La principale limite de l'algorithme JPDAF est son incapacité à gérer nouveaux objets entrant dans le champ de vision ou objets déjà suivis sortant du champ de vision. La technique MH-Test développée par Reid et al [73] pour remédier cette limitation. Cette technique a la capacité de créer de nouvelles pistes pour les objets entrant dans le champs de vision et terminer les pistes pour les objets sortant.

### 2.5.3.2 Suivi de Noyau

Le suivi de noyau s'effectue en calculant le mouvement de l'objet, qui est représenté par une forme géométrique (rectangle ou ellipse), d'une image sur l'autre. Le mouvement de l'objet est généralement sous la forme de mouvement paramétrique (translation, rotation, affine, etc). Il existe un grand nombre de méthodes de suivi de noyau. Celles-ci diffèrent en termes de la représentation de l'apparence utilisée, le nombre d'objets suivis, et la

méthode utilisée pour estimer le mouvement de l'objet. Où, selon la représentation de l'apparence utilisée, on distingue deux sous-catégories :

### 1. Méthodes basées sur des Templates ou densité de probabilité

Les méthodes basées sur des Templates ou densité de probabilité ont été largement utilisés en raison de la simplicité relative et de leur faible coût de calcul. L'approche la plus simple pour suivre un objet dans ce paradigme est mise en correspondance de Templates (Template Matching) [74, 75]. Il prend le contenu de l'image au sein de la boîte englobante dans la première trame comme une Template initial, Le suivi est ensuite effectué pour trouver la partie de l'image la plus similaire au Template initial, par des fonctions de similarités. Les informations utilisées sont l'intensité ou les composantes couleurs de l'image, ce qui rend ces méthodes sensibles aux changements d'illumination.

L'inconvénient majeur de mise en correspondance de Templates est la lenteur de la recherche exhaustive, surtout lorsque la taille du Template et/ou la zone de recherche est grande.

Il existe plusieurs méthodes dans cette approche parmi lesquelles nous avons le tracker KLT (Kanade-Lucas-Tomasi). [76, 77] c'est la méthode la plus efficace basée sur la mise en correspondance de Templates. Il s'agit de trouver les correspondances affines transformées entre deux trames successives au moyen de dérivés spatio-temporels. La nouvelle localisation de la cible est déterminée en mettant en correspondance sa position dans la trame précédente à la localisation dans la trame actuelle en utilisant la transformation affine estimée (Figure 2.12).



FIGURE 2.12 – Suivi des caractéristiques en utilisant le suivi KLT

### 2. Méthodes basées sur une représentation multi-vues de l'objet

Les objets peuvent apparaître différents dans différentes vues, et si la vue de l'objet change radicalement pendant le suivi, le modèle d'apparence peut ne plus être valide et la piste d'objet peut être perdue. Pour surmonter ce problème, les méthodes basées sur une représentation multi-vues de l'objet construisent leur modèle par apprentissage d'un ensemble de vues différentes de ce dernier [78]. deux catégories de ces méthodes :

**Méthodes basées sur les modèles génératifs :** Le suivi utilisant le modèle génératif se base sur l'apprentissage d'un modèle pour représenter l'objet cible, puis l'utilise pour trouver la région la plus similaire dans les trames suivantes.

Black et Jepson [79] ont proposé une approche basée sur le sous-espace, c'est-à-dire l'espace propre, pour calculer la transformation affine de l'image courante de l'objet vers l'image reconstruit à l'aide de vecteurs propres. Tout d'abord, une représentation de sous-espace de l'apparence d'un objet est construite à l'aide de l'analyse en composantes principales (ACP), le vecteurs propres est utilisée pour calculer la transformation par minimisation de la différence entre l'image d'entrée et l'image reconstruite. le suivi est effectué en estimant les paramètres affines de manière itérative jusqu'à ce que la différence entre l'image d'entrée et l'image projetée soit minimisée.

**Méthodes basées sur les modèles discriminatifs :** les modèles discriminatifs transforme le problème de suivi en un problème de classification binaire pour distinguer l'objet cible de l'arrière-plan [80].ils décrivent l'objet par rapport à l'arrière-plan.ils se concentrent sur la découverte de caractéristiques très informatives pour le suivi des objets visuels,et Ils visent à maximiser la séparation entre les régions objet et non-objet de manière discriminante.

Avidan [81] a utilisé un classificateur Support Vector Machine (SVM) pour le suivi SVM est un schéma de classification général qui, étant donné un ensemble de valeurs positives et négatives, exemples d'entraînement atifs, trouve le meilleur hyperplan séparant les deux classes . Pour les trackers basés sur SVM, le positif les exemples sont constitués des images de l'objet à suivre, et les exemples négatifs se composent de toutes les objets qui ne doivent pas être suivies ( appartiennent d'arrière-plan). La méthode de suivi d'Avidan, maximise le score de classification SVM sur les régions d'image afin d'estimer la positionnement de l'objet, au lieu de minimiser la différence d'intensité d'un modèle par rapport aux régions d'image. Un avantage de cette approche est que la connaissance des objets d'arrière-plan.

### 2.5.3.3 Suivi de silhouette

La représentation d'un objet par une silhouette permet de tenir compte précisément de la forme de l'objet. Surtout les objets aux formes complexes qui ne peuvent pas être bien décrites par des formes géométriques simples par exemple, les mains, la tête et les épaules .Les méthodes basées sur la silhouette fournissent une description de forme précise pour ces objets, en utilisant l'information encodée à l'intérieur de la région objet . Le but des méthodes de suivi fondées sur l'utilisation de silhouettes est d'estimer la silhouette des objets d'intérêt pour chaque image de la vidéo .On distingue deux catégories de ces méthodes :

## 1. Méthodes de correspondance de formes

Les méthodes de correspondance de forme recherchent la silhouette de l'objet et son modèle associé dans l'image actuelle, puis effectuent la recherche en calculant la similarité de l'objet avec le modèle créé à partir de la silhouette d'objet supposée basée sur l'image précédente. Ici, la silhouette est supposée se traduire uniquement de l'image actuelle à la suivante .

En 1993, Huttenlocher et al. [82] ont effectué une correspondance de forme en utilisant une représentation basée sur les arêtes. La distance de Hausdorff est utilisée pour construire une surface de corrélation pour définir le minimum comme la nouvelle position de l'objet. L'échelle de Hausdorff est donnée par une relation mathématique permettant de comparer deux ensembles de points  $A = \{a_1, a_2, \dots, a_m\}$  et  $B = \{b_1, b_2, \dots, b_n\}$  en termes de leurs membres les moins similaires

$$H(A, B) = \max \{h(A, B), h(B, A)\} \quad (2.15)$$

où  $h(a, b) = \inf_{c \in A} \sup_{d \in B} \|a - c\|$  est la norme de choix.

la distance de Hausdorff mesure les arêtes les plus incompatibles. Pour cette raison, la méthode met l'accent sur les parties de la carte des bords qui ne sont pas radicalement affectées par le mouvement de l'objet.

## 2. Méthodes d'évolution du contour

Les méthodes de suivi de contour [83], font progresser de manière itérative un contour primaire dans la trame précédente vers sa nouvelle position dans la trame actuelle, Cette progression de contour nécessite une certaine quantité de l'objet dans la superposition de l'image actuelle avec la région de l'objet dans l'image précédente. Le suivi par l'évolution d'un contour peut être effectué en utilisant deux approches : La première approche utilise des modèles d'espace d'état qui modélisaient la forme et le mouvement du contour. La deuxième approche fait directement évoluer le contour en minimisant l'énergie du contour en utilisant des techniques de minimisation directe telles que la descente de gradient. L'avantage le plus important du suivi des silhouettes est de fournir une flexibilité pour gérer une grande variété de formes d'objets [84].

**Suivi par Modèle d'espace d'état :** L'état de l'objet est défini en terme de la forme et des paramètres de mouvement du contour. L'état à chaque fois instant est mis à jour et ça maximisée la probabilité a posteriori du contour. cette probabilité dépend de l'état antérieur et de la probabilité actuelle qui est définie en termes de distance du contour aux bords observés.

Terzopoulos et Szeliski [85] définissent l'état de l'objet par la dynamique des points de contrôle. qui est modélisée en termes de modèle de ressort, Le nouvel état (paramètres de ressort) du contour est prédit à l'aide du filtre de Kalman. L'étape de mis à

jour utilise les observations d'image qui sont définies en termes de gradients d'image.

Isard et Blake [86] ont défini l'état de l'objet en termes de paramètres de forme de spline et de paramètres de mouvement affine pour définir l'état de l'objet, L'état est mis à jour à l'aide d'un filtre à particules. Afin d'obtenir des échantillons initiaux pour le filtre, ils calculent les variables d'état à partir des contours extraits dans des trames consécutives lors d'une phase d'apprentissage.

### **Suivi par Minimisation directe de la fonctionnelle d'énergie du contour :**

Dans le contexte de l'évolution des contours, il existe une analogie entre les méthodes de segmentation et les méthodes de suivi des contours de cette catégorie, cette analogie est que les deux sont minimisent l'énergie fonctionnelle soit par des méthodes gloutonnes(greedy) soit par descente de gradient. L'énergie de contour est définie en termes d'informations temporelles sous la forme soit du gradient temporel (flux optique) [87, 88] ou des statistiques d'apparence générées à partir de l'objet et des régions d'arrière-plan [89, 90].

Le suivi des contours à l'aide de gradients d'image temporels est motivé par les travaux approfondis sur le calcul du flux optique [63], Bertalmio et al. [91] utilisent cette contrainte pour faire évoluer le contour dans des trames consécutives en utilisant la représentation par courbes de niveaux, ils ont utilisé deux fonctionnelles énergétiques, une pour le suivi des contours, Et , et une autre pour le morphing d'intensité (minimise les changements d'intensité d'une trame à la suivante).

Mansouri utilise également la contrainte de flux optique pour le suivi de contour. il a appliqué la contrainte de flot optique pour calculer le vecteur de flot pour chaque pixel dans la région d'objet complète. Une fois les vecteurs de flux calculés, l'énergie de contour, qui est basée sur la contrainte de constance de luminosité, est évaluée. Ce processus est effectué de manière itérative jusqu'à ce que l'énergie soit minimisée.

## 2.6 Conclusion

Dans ce chapitre, nous avons présenté un état de l'art sur la détection d'objet, qui représente le fondement de tout algorithme de suivi. Chaque méthode de suivi nécessite un mécanisme de détection d'objet, soit sur toutes les trames, soit lorsque l'objet apparaît pour la première fois dans la vidéo et aussi nous avons fait une comparaison entre les méthodes de détection d'objet. Ensuite, Nous avons parlé sur les Caractéristiques visuelles pour le suivi d'objets et nous avons présenté un état de l'art sur le suivi d'objet visant à montrer la diversité des approches proposées dans ce domaine. La différence entre ces approches réside en partie dans le choix de la représentation des objets, des caractéristiques de l'image utilisées et de la nature du mouvement estimé. Ce choix dépend de l'application ainsi que de la vidéo traitée. Caractéristiques visuelles pour le suivi d'objets.

## CHAPITRE 3

# CONTRIBUER À LA DÉTECTION ET AU SUIVI D'OBJETS EN MOUVEMENT

### 3.1 Introduction

Ce chapitre est divisé en deux parties principales, la première partie est consacrée à expliquer l'algorithme de détection d'objets en mouvement, en utilisant la méthode la plus efficace par rapport aux autres méthodes, qui est la méthode du modèle de mélange gaussien qui permet de détecter des objets en mouvement en divisant un scène dans l'arrière-plan pour regrouper les objets statiques, et le premier plan qui identifie les objets en mouvement.

Nous avons appliqué une amélioration à cette méthode en l'intégrant avec un algorithme d'entropie 2D pour être plus adaptable au contraste lumineux, ce qui est un défi majeur dans la procédure de détection d'objets en mouvement vidéo.

La deuxième partie est consacrée à expliquer la méthode de suivi des objets détectés, où nous avons choisi le filtre de Kalman en raison des résultats satisfaisants qu'il fournit.

## 3.2 Schéma général de travail

Ce travail se déroule en trois étapes principales, comme le montre le schéma suivant (Figure 3.1) :

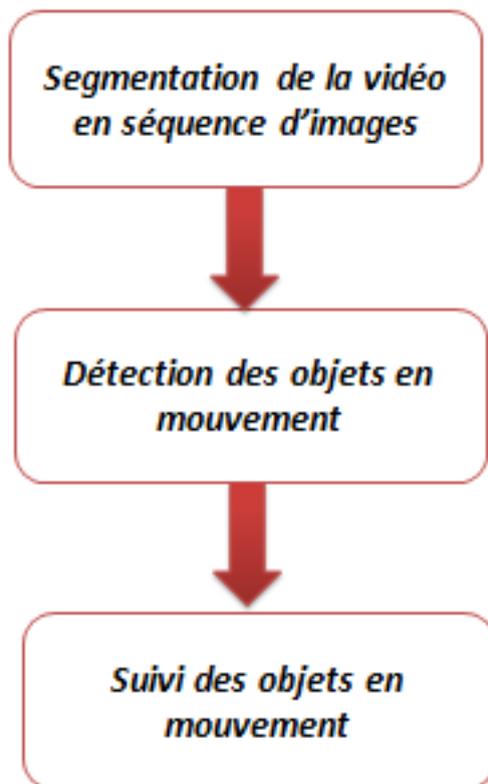


FIGURE 3.1 – Schéma général du travail

## 3.3 Détection d'objets en mouvement

Avant de commencer la détection d'objet il faut d'abord segmenter la vidéo en images séparées appelés « frame » le travail va se faire sur ses dernière.

La mixture gaussien est une méthode de modélisation d'arrière-plan largement utilisée pour détecter des objets en mouvement à partir d'une caméra fixe. Nous avons utilisé cette méthode car elle est plus résistante au bruit, aux petits mouvements en arrière-plan et aux changements brusques de lumière.

Cependant, ce genre de méthodes est souvent sensibles à la présence de lumière brusque, on a donc utilisé une amélioration de GMM basée sur l'entropie 2D visant la mise à jour lente du fond dans le modèle de mélange gaussien en éclairage brusque, modifiant ainsi le taux d'apprentissage afin d'accélérer la modélisation du fond. cette amélioration nous permet d'avoir une estimation d'arrière-plan plus précis dans des scènes complexes et améliorer la précision de la détection des cibles.

### 3.3.1 Modèle de mixture gaussien (GMM)

Modèle de mixture gaussien La modélisation d'arrière-plan gaussien est fournie par C.stauffer et WEL Grimson [92,93], cette méthode est plus adaptée à la détection d'objets en mouvement lorsque l'arrière plan contient des instabilités qui proviennent des changements dans l'illumination de la scène ou des mouvements d'objets non stationnaires.. La modélisation gaussienne du fond [94] consiste à modéliser jusqu'à chaque pixel du fond, et ses paramètres seront mis à jour au fur et à mesure du temps. Une cible mobile est détectée si la distribution de pixels de la cible coïncide avec la modélisation de fond gaussien.

#### 3.3.1.1 Modélisation une mixture de gaussiennes

Dans ce modèle [92],  $K$  ( $3 < K < 5$  en général) pièces de modélisation gaussienne seront établies pour chaque pixel. Pour la fonction de densité de probabilité de la valeur d'échantillonnage  $x_t$  au temps  $T$ , elle peut être désignée par une somme de densité de probabilité pondérée de  $K$  éléments de la fonction de distribution gaussienne multidimensionnelle :

$$P(x_t) = \sum_{i=1}^K W_{i,t} \eta_{i,t}(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (3.1)$$

Avec les paramètres des gaussiennes :

- $K$  est le nombre de modélisation gaussienne,
- $W_{i,t}$  est le poids du  $i^{\text{ème}}$  Distribution gaussienne au temps  $t$ ,
- $\mu_{i,t}$  est la valeur moyenne de la  $i^{\text{ème}}$  Distribution gaussienne au temps  $t$ ,
- $\Sigma_{i,t}$  est la matrice de covariation du  $i^{\text{ème}}$  Distribution gaussienne au temps  $t$ .  
 $\Sigma_{i,t} = \sigma_{i,t}^2 I$ , pour chaque pixel  $\sigma_{i,t}^2$  est la variation de la  $i^{\text{ème}}$  distribution gaussienne au temps  $t$  pour la séquence d'images de chaque pixel,
- $\eta(X_t, \mu_{i,t}, \Sigma_{i,t})$  est la fonction de densité de probabilité de la  $i^{\text{ème}}$  distribution gaussienne au temps  $t$ . Où :

$$\eta(X_t, \mu_{i,t}, \Sigma_{i,t}) = \frac{1}{(2\pi)^{n/2} |\Sigma_{i,t}|^{1/2}} \cdot \exp\left(-\frac{1}{2} (X_t - \mu_{i,t})^T \Sigma_{i,t}^{-1} (X_t - \mu_{i,t})\right) \quad (3.2)$$

Parmi les méthodes utilisées dans le calcul des paramètres GMM la méthode classique EM (Expectation Maximisation). Le principe de cette méthode est de calculer les paramètres inconnus (la moyenne, la variance et les poids des gaussiennes), des donnée observée et incomplètes (l'histogramme temporel de chaque pixel), l'idée de ces algorithmes consiste à déterminer un poids pour chaque mélange gaussienne (pour chaque pixel) où le poids de gaussienne est conforme aux nombres des images et de la valeur que prend le pixel.

### 3.3.1.2 Initialisation du modèle

Tout d'abord, les paramètres sont initialisés avec  $w_k = w_0$ ,  $\mu_k = \mu_0$  et  $\sigma_k = \sigma_0$ . S'il y a correspondance (le pixel  $X$  appartient à  $i^{\text{ème}}$  distribution gaussian si la différence entre l'intensité du pixel et la moyenne de la  $i^{\text{ème}}$  distribution gaussian ( $\mu_k$ ) inférieure à  $2.5 \sigma_k$  de  $i^{\text{ème}}$  distribution gaussian sinon n'appartient pas), c'est-à-dire :

$$\|X - \mu_k\| < 2.5 \sigma_k \quad (3.3)$$

### 3.3.1.3 Mise à jour des paramètres

Si un pixel ne peut correspondre à aucune distribution gaussienne existante, une nouvelle distribution gaussienne sera construite pour remplacer la distribution existante dont le poids est le plus petit. Les paramètres mis à jour du modèle de mélange de gaussian sont représentés par l'expression :

$$w_{i,t} = (1 - \alpha) w_{i,t-1} + \alpha \quad (3.4)$$

$$\mu_{i,t} = (1 - \beta) \mu_{i,t-1} + \beta x_t \quad (3.5)$$

$$\sigma_{i,t}^2 = (1 - \beta) \sigma_{i,t-1}^2 + \beta \| (x - \mu_{i,t}) \|^2 \quad (3.6)$$

Où  $\alpha$  ( $0 < \alpha < 1$ ) est le taux de mise à jour,  $\beta$  est le taux d'apprentissage et est défini comme suit :

$$\beta = \alpha \eta (X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (3.7)$$

## 3.3.2 Construction de l'image de référence

Après avoir calculé les paramètres de chaque pixel de l'image actuelle, on continue à chercher la référence gaussienne pour chaque pixel. Il existe un critère permettant de choisir la référence gaussienne :

**Poids  $w$  :** vu que le fond existe dans la plus part des images, un poids fort décrit un pixel appartenant au fond.

**Variance :** vu que la valeur du fond (l'arrière-plan) est presque fixe, sa variance est alors minimale.

**Poids/variance :** il est le critère le plus utilisé.

### 3.3.3 GMM amélioré

En raison des lacunes des systèmes GMM conventionnels et Dans le but de mettre à jour le fond lent dans un modèle de mélange gaussien en illumination soudaine, Entropie 2D est utilisé pour juger de la surtension lumineuse, modifiant ainsi le taux d'apprentissage pour accélérer la modélisation du fond.

#### 3.3.3.1 Entropie 2-D de l'image

Dans une image grise [95], le niveau de gris est divisé en valeurs  $m$  et le niveau de gris moyen est également divisé en même valeurs  $M$ . À chaque pixel, la valeur moyenne au niveau gris du voisinage est calculée.

Cela forme une paire : le niveau de gris pixel et la moyenne du voisinage, chaque paire appartient à un espace à 2 dimensions.

Le nombre total de l'espace est  $m*m$  et le nombre total de pixels à tester est  $N*N$ . Le nombre total d'occurrence  $f(i, j)$  d'une paire  $(i, j)$  divisé par le nombre total de pixels  $N*N$ , définit la fonction de masse de probabilité d'articulation  $p_{ij}$ , à savoir :

$$P_{ij} = f(i, j) / N * N \quad (3.8)$$

où :  $i, j = 0, \dots, m-1$ .

Il reflète la distribution d'échelle de gris d'une position de pixel et ses caractéristiques de distribution gris pixel environnante. l'entropie 2-D de l'image est définie comme :

$$H = - \sum_{i=0}^{255} \sum_{j=0}^{255} p_{ij} \ln p_{ij} \quad (3.9)$$

#### 3.3.3.2 Amélioration GMM par l'entropie 2D

Selon la théorie de l'entropie [95], la valeur d'entropie de l'image sombre est plus petite et l'image plus claire a une valeur d'entropie plus grande. Nous calculons l'entropie 2-D de l'image pour juger si l'éclairage change. Les étapes de l'algorithme amélioré sont les suivantes :

1. Un certain nombre d'images de la vidéo sont sélectionnées comme ensemble d'apprentissage et les images restantes sont utilisées comme ensemble de test. Nous utilisons l'algorithme original du modèle de mélange de gaussien pour modéliser la région d'arrière-plan, et le premier plan et l'arrière-plan sont détectés dans l'ensemble d'apprentissage. La région antérieure est enregistrée comme  $A_f$ , et la région d'arrière-plan est indiquée comme  $A_b$ .
2. Avec la frame  $t$ , nous calculons la différence d'entropie 2-D entre la frame  $t$  et la frame  $t-1$  dans  $A_f$ . Lorsque la valeur est supérieure à  $T_c$ , nous pensons que la mutation légère se produit, et notons  $A_f$  comme  $T_c$ , sinon comme  $A_s$ .

$$\left\{ \begin{array}{l} H(X_{t-1}) = -\sum_{i=0}^{255} \sum_{j=0}^{255} p_{ij}^{t-1} \ln p_{ij}^{t-1} \\ H(X_t) = -\sum_{i=0}^{255} \sum_{j=0}^{255} p_{ij}^t \ln p_{ij}^t \\ |H(X_t) - H(X_{t-1})| \geq T_c, X_t \in A_c \\ |H(X_t) - H(X_{t-1})| < T_c, X_t \in A_s \end{array} \right. \quad (3.10)$$

où :

- $H(X_t)$  représente l'entropie 2-D de la trame courante  $t$ ,
- $H(X_{t-1})$  représente l'entropie 2-D de la trame précédente  $t-1$ ,
- $T_c$  est le seuil et  $T_c = 0,05$ .

3. Dans cet algorithme, le taux d'apprentissage de la région d'arrière-plan reçoit un taux d'apprentissage fixe, et le taux d'apprentissage de la région de premier plan est obtenu par l'expression suivante :

$$\left\{ \begin{array}{l} \alpha_t \in \min \{ \alpha_{t-1} \cdot (1 + \exp(-\lambda \cdot |H(X_t) - H(X_{t-1})| / T_c)), 1 \}, X_t \in A_c \\ \alpha_t \in \min \{ 0, \alpha_{t-1} \cdot (1 + \exp(-\lambda \cdot |H(X_t) - H(X_{t-1})| / T_c)) \}, X_t \in A_s \end{array} \right. \quad (3.11)$$

où :

- $\alpha_t$  est le taux d'apprentissage de la trame courante.
- $\lambda$  est le paramètre de contrôle.

La méthode utilisée pour détecter les objets en mouvement est représentée par l'organigramme illustré sur (Figure 3.2)

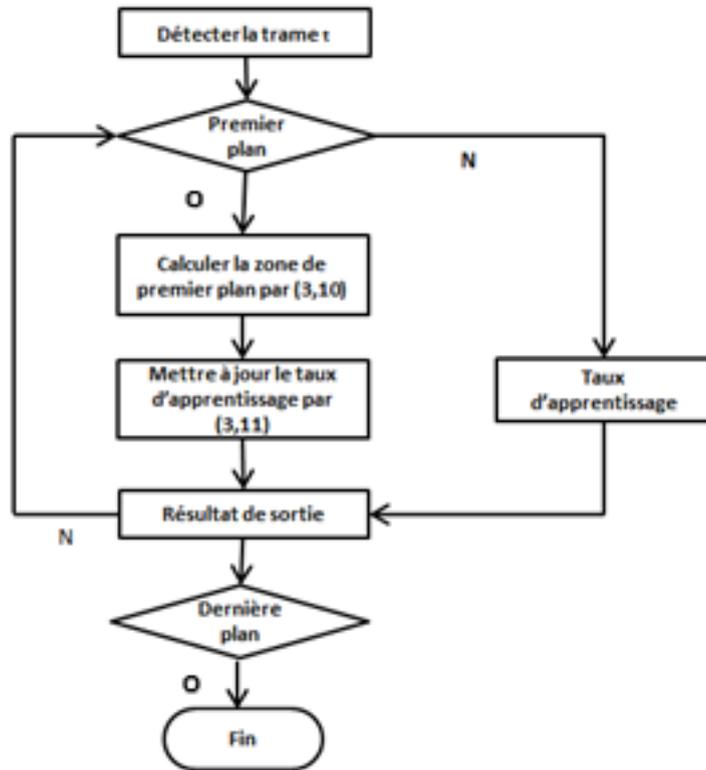


FIGURE 3.2 – Organigramme a été amélioré de la méthode de modèle de mixture gaussien [95]

### 3.3.4 Suivi des objets détectés

Les filtres stochastiques proposent des solutions pour estimer l'état d'un système conditionnellement à l'ensemble des mesures passées. Parmi ces filtres, on propos dans ce travail d'utiliser le filtre de Kalman.

Les deux idées qui motivent l'utilisation d'un filtre de Kalman sur les paramètres du mouvement sont :

- Accumuler les mesures des paramètres de mouvement d'une région au cours du temps pour affiner les estimations.
- Obtenir une bonne prédiction du mouvement nécessaire pour prédire la segmentation avant ajustement et pour initialiser l'estimation de mouvement.

#### 3.3.4.1 Filtre de Kalman

Le filtre de Kalman a été présenté en 1960, Il s'agit d'un estimateur optimal de processus aléatoires. Il permet l'estimation de variables d'état d'un système (position, vitesse) et il fournit aussi la variance de l'erreur de l'estimation.

Son appellation « filtre » vient du fait qu'il permet, entre autre, d'estimer un signal bruité (bruit blanc), en séparant le signal de son bruit.

Le filtre de Kalman s'applique aux signaux non stationnaires, il résout le problème de filtrage linéaire et se présente sous la forme d'un ensemble d'équations différentielles ou récurrentes plus faciles à résoudre sur ordinateur numérique en présence de condition initiales et d'entrées déterministe.

Le filtre de Kalman est une approche statistique de l'ingestion de données qui réduit l'erreur entre l'état vrai et l'état filtré en combinant les observations avec les informations fournies par le modèle.

#### 3.3.4.2 Suivi de mouvement par filtre de Kalman

Dans le suivi du mouvement, le filtre de Kalman est une méthode de prédiction intéressant plusieurs chercheurs [96, 97]. Un filtre de Kalman est essentiellement un ensemble d'équations récursives, qui sont utilisées ensemble pour évaluer le mouvement dans un système dynamique linéaire et une modélisation très précise. Le filtre de Kalman est appelé 'discret' parce que l'algorithme est discret dans le temps. En utilisant une multitude de différents vecteurs et de matrices, le filtre de Kalman est capable de construire une représentation interne du système et grâce à cette représentation, on peut estimer l'état futur du système. Dans le filtrage de Kalman, les équations décrivent l'état du système à des moments successifs, avec ces équations, l'état futur peut être prédit en utilisant l'état actuel sans compter sur les états passés dans les calculs.

Dans le processus du suivi, le filtre de Kalman 'regarde' un objet quand il se déplace, c'est à dire il prend les informations sur l'état de l'objet à ce moment-là. Puis il utilise ces informations pour prédire où se trouve l'objet dans la prochaine trame. Ensuite, grâce à la prédiction ainsi qu'à l'observation, la position de l'objet est corrigée.

### 3.3.4.3 Principe du filtre de Kalman

Basé sur le principe du filtre de Kalman qui consiste à donner une estimation de processus en utilisant un système à retour sous la forme de mesures (bruités), On distingue deux types d'équations [98] :

- Les équations de prédiction (extrapolation) : les équations d'extrapolation sont utilisées pour obtenir des estimations à priori de la prochaine étape en publiant les estimations de l'état actuel et les covariances d'erreur.
- Les équations de mise à jour (correction) : Les équations de correction insèrent une nouvelle mesure avec une estimation préalable afin d'obtenir une estimation ultérieure améliorée, c'est-à-dire les équations responsables du retour (Figure 4.4) .

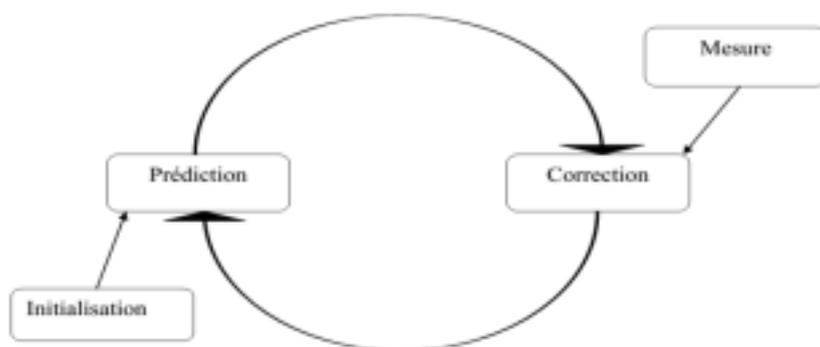


FIGURE 3.3 – Le cycle du filtre de Kalman [98]

### 3.3.4.4 Calcul de l'estimateur de Kalman

On considère un système dynamique linéaire à temps discret défini par un vecteur d'équations aux différences, entaché d'un bruit blanc gaussien. Plus précisément, il existe un système de temps discret et son état à l'instant  $n$  est donné par le vecteur  $x_n$ . L'état de la prochaine étape  $n+1$  est donné par 3.12 :

$$x_{n+1} = F_{n+1,n} x_n + w_{n+1} \quad (3.12)$$

Où :

- $F_{n+1,n}$  est la matrice de transition d'état caractérisant l'évolution du système.
- $w_{n+1}$  est un bruit blanc gaussien à moyenne nulle et de matrice de covariance  $Q_{n+1}$ .

L'équation de mesure est donnée par :

$$z_{n+1} = H_{n+1} x_{n+1} + v_{n+1} \quad (3.13)$$

Où :

- $z_{n+1}$  est le vecteur de mesure.
- $H_{n+1}$  est la matrice de transformation de l'état vers l'espace des mesures.
- Le bruit de mesure est modélisé par un vecteur  $v_{n+1}$  sa distribution est aussi une gaussienne de moyenne nulle et de covariance  $R_{n+1}$ .

Dans l'équation 3.13, le vecteur de mesure  $z_{n+1}$  ne dépend que du vecteur d'état actuel  $x_{n+1}$  et le vecteur de bruit  $v_{n+1}$  est indépendant du bruit  $w_{n+1}$ . Le filtre de Kalman calcule l'erreur d'estimation quadratique moyenne minimale de l'état  $x_k$  étant donné les mesures  $z_1, \dots, \dots, z_k$ .

### 3.3.4.5 Prédiction du vecteur d'état et des mesure

Une prédiction de l'état futur des variables que l'on cherche à estimer est tout d'abord réalisée, à partir de l'état précédent. La prédiction de l'état et sa matrice de covariance au temps  $t_n$  sont données par le modèle dynamique :

$$\hat{x}_n^- = F_{n,n-1} \hat{x}_{n-1} \quad (3.14)$$

$$P_n^- = F_{n,n-1} P_{n-1} F_{n,n-1}^T + Q_n \quad (3.15)$$

Où la matrice de covariance  $P$  est de dimensions  $R^{D_x \times D_x}$  deuxième terme dans l'équation correspond à la covariance du vecteur de perturbation de l'état.

### 3.3.4.6 Correction de l'état

Pour la mise à jour du vecteur d'état estimé  $\hat{x}$  et sa covariance  $P$ , il est nécessaire d'obtenir le gain du filtre  $G_n$  qui dépend directement des matrices de covariance obtenues pour l'état prédit.

$$G_n = P_n^- H_n^T [H_n P_n^- H_n^T + R]^{-1} \quad (3.16)$$

$$\hat{x}_n = \hat{x}_n^- + G_n (z_n - H_n \hat{x}_n^-) \quad (3.17)$$

$$P_n = (I - G_n H_n) P_n^- \quad (3.18)$$

Dans cette étape de correction, les observations de l'instant courant sont utilisées pour corriger l'état prédit dans le but d'obtenir une estimation plus précise.

Le filtre de Kalman est appliqué en temps réel, parce que ce filtre ne nécessite pas toutes les données passées pour produire une estimation à l'instant courant, c-à-d il ne nécessite donc pas de mise en mémoire et de retraitement des données.

On résume les étapes de suivi par filtre de Kalman dans l'algorithme 1 suivant :

---

**Algorithm 1** Filtre de Kalman

---

1: Initialisation :  $\hat{x}_0$  la position initiale de l'objet.

$$\hat{x}_0 = E[x_0]$$

$$P_0 = [(x_0 - E[x_0])(x_0 - E[x_0])^T]$$

2: Prédiction :

$$\hat{x}_n^- = F_{n,n-1} \hat{x}_{n-1}$$

$$P_n^- = F_{n,n-1} P_{n-1} F_{n,n-1}^T + Q_n$$

3: Correction :

$$G_n = P_n^- H_n^T [H_n P_n^- H_n^T + R]^{-1}$$

$$\hat{x}_n = \hat{x}_n^- + G_n (z_n - H_n \hat{x}_n^-)$$

$$P_n = (I - G_n H_n) P_n^-$$

Retourner à l'étape de prédiction pour l'itération suivante.

---

## 3.4 conclusion

Dans ce chapitre, nous avons détaillé les différents principes et étapes de la méthode de détection proposée (GMM) et les étapes que nous avons suivies pour la développer afin d'obtenir des résultats plus précis. Nous avons également expliqué le filtre de Kalman utilisé pour suivre les objets en mouvement.

Dans le prochain chapitre, nous allons présenter les résultats expérimentaux pour la détection et le suivi les objets en mouvement dans une séquence vidéo.

## CHAPITRE 4

# RÉSULTATS ET DISCUSSIONS

### 4.1 Introduction

Dans le chapitre précédent, nous avons présenté notre contribution proposée pour le système de détection et suivi d'objet en mouvement. Dans ce chapitre, nous allons présenter la mise en œuvre de notre application, tel que le langage de programmation, les bases de données utilisées et les outils du développement . Ensuite, nous présenterons quelques résultats concernant la détection et le suivi.

## 4.2 Environnement de travail

### 4.2.1 Langage de programmation python

Notre application a été développée sous Python 2.9.

Nous avons choisi Python pour les avantages suivants :

- python fonctionne sur différentes plateformes (Windows, Mac, Linux) ;
- Il a une syntaxe simple similaire à la langue anglaise ;
- Ce langage peut être traité de manière procédurale, de manière orientée objet ou de manière fonctionnelle.

### 4.2.2 Caractéristique de la plateforme

Dans ce travail, On a utilisé un ordinateur ayant les caractéristiques suivant :

- Processeur : Intel(R) Cor (TM) i3-3217 U.
- Une RAM de 4Go.
- Un disque dur 465.76 Go.
- Résolution de l'écran : 1366 X 768.

### 4.2.3 Dataset

Pour évaluer la méthode proposée on l'applique sur certain vidéo de la base LA-SIESTA (Labeled and Annotated Sequences for Integral Evaluation of SegmenTation Algorithms) [99] qui consiste en de nombreuses séquences en direct internes et externes organisées en différentes classes, chacune couvrant un défi spécifique dans les stratégies de détection d'objets en mouvement. Il contient des séquences enregistrées avec des caméras fixes et mobiles, et fournit également des informations sur les objets en mouvement qui restent temporairement immobiles.

Chaque séquence contient les informations suivantes (où "Id" est l'identifiant de la séquence) :

- Vidéo originale : Un dossier nommé "Id" contenant un fichier BMP 24bpp pour chaque image de la vidéo originale.
- Images étiquetées : le dossier nommé "Id-GT" contient un fichier BMP 24png avec la ou les vérités de base correspondant à chaque image de la vidéo originale. Considérant que cette séquence de base de données contient un maximum de trois objets animés

Nous avons essayé de travailler avec des vidéos de différents types tels que une vidéo contenant une personne (Figure 4.1 vidéo(a)) , et une vidéo multi-personnes (Figure 4.1 vidéo(b)), et une vidéo des choses rigide (Figure 4.1 vidéo(c)),et nous avons également travaillé sur d'autres vidéos que nous avons filmées à l'universités.

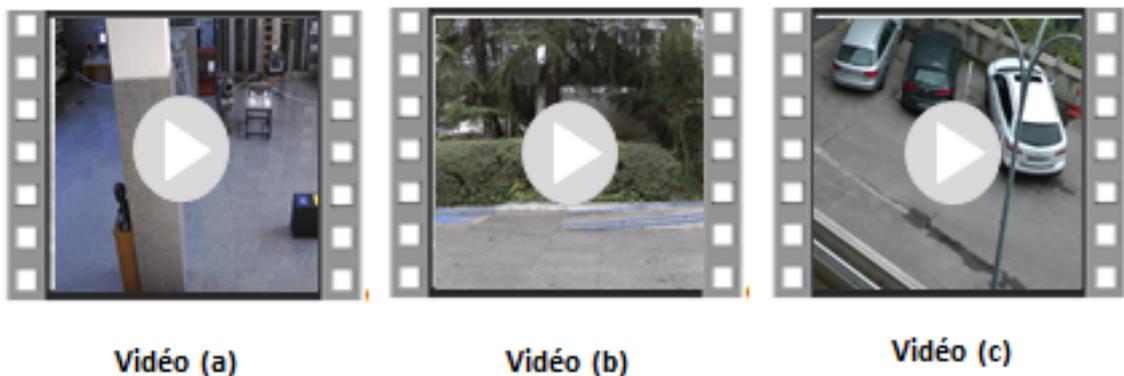


FIGURE 4.1 – Vidéos de la base de données LASIESTA

### 4.3 Interfaces et résultats

Dans cette partie, nous présenterons l'interface de notre application et quelques résultats de la phase de détection et de suivi les objets en mouvement.

#### 4.3.1 Interface de l'application

L'interface qu'on utilisé est simple ,elle compose deux fenêtre :

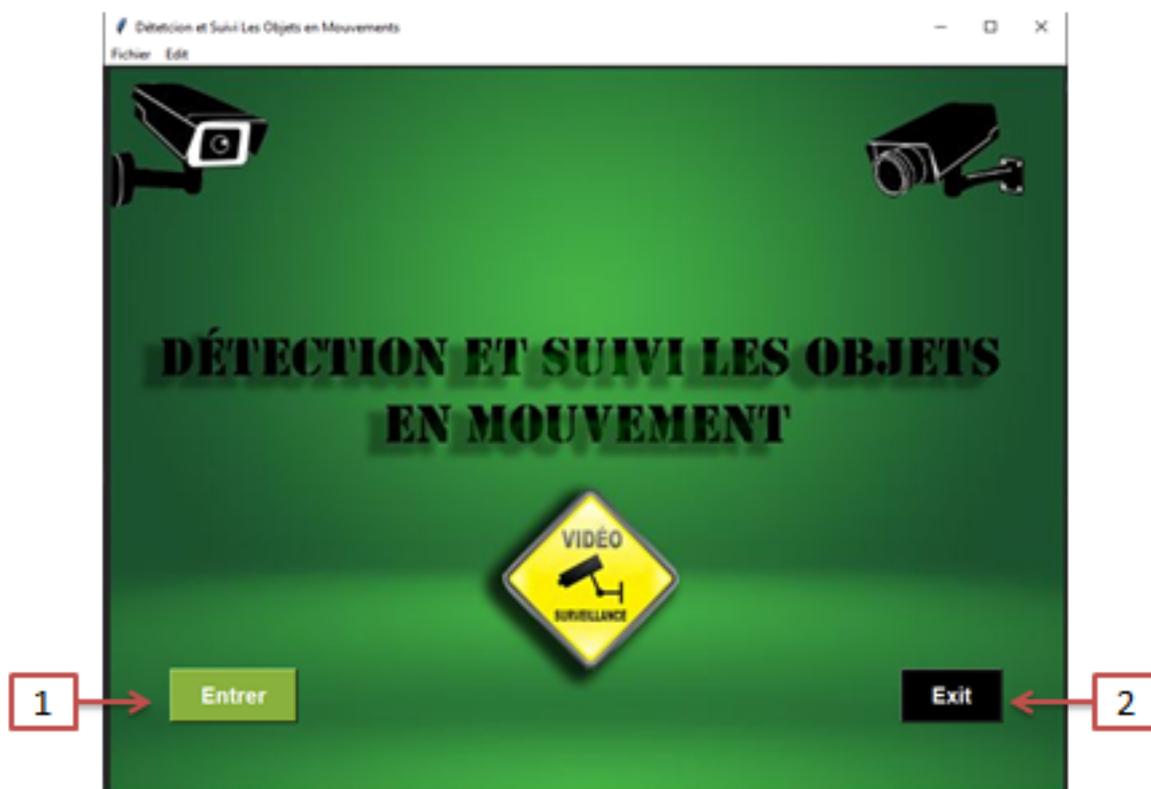


FIGURE 4.2 – Interface de l'application

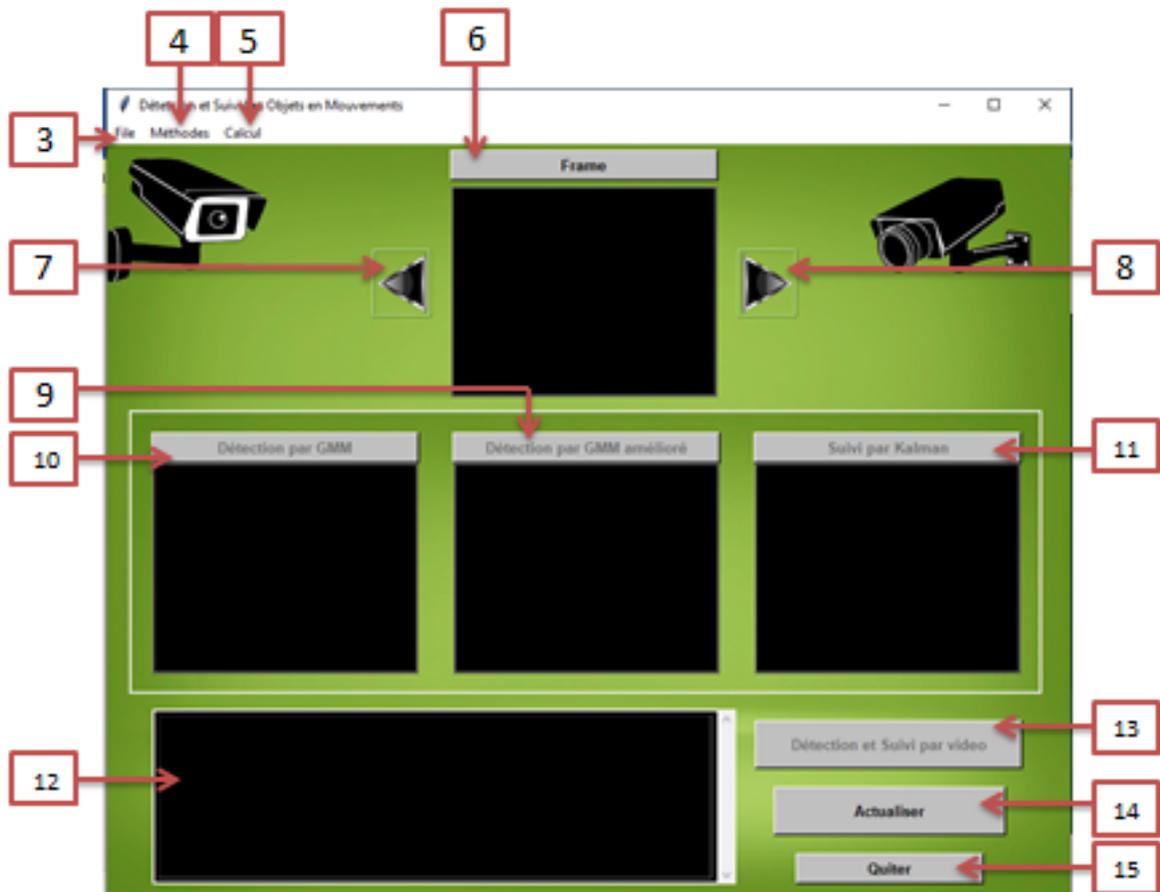


FIGURE 4.3 – Interface principale de l'application

1. Entrer.
2. Exit.
3. Charger la vidéo choisie.
4. Choisir la méthode utilisée.
5. Calculer les valeurs statistique.
6. Fenêtre de frame d'origine.
7. Frame précédent.
8. frame suivant.
9. Fenêtre de détection par GMM.
10. Fenêtre de détection par GMM amélioré.
11. Fenêtre de suivi.
12. Afficher les informations et les valeurs statistiques.
13. Détection et suivi par vidéo.
14. Actualiser.
15. Quitter.

### 4.3.2 Etapes d'exécution de l'application

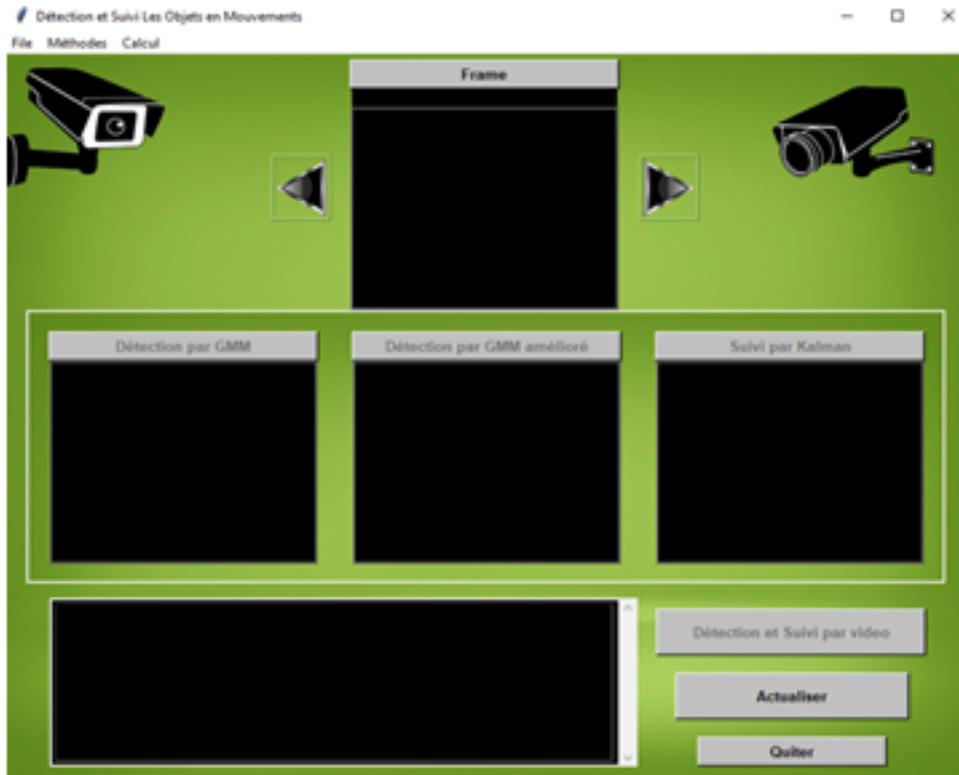
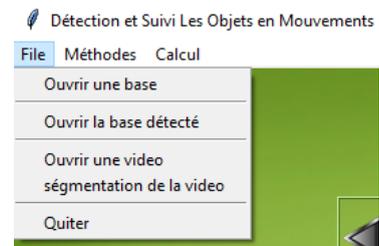


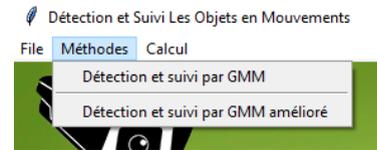
FIGURE 4.4 – Interface principale de l'application

1. Tout d'abord, l'utilisateur de notre application doit sélectionner la source de la vidéo, soit à partir de dataset, soit à partir d'un fichier vidéo sur l'ordinateur. Si une vidéo est sélectionnée à partir d'un fichier vidéo sur l'ordinateur, la segmentation vidéo sera effectuée en premier.
  - Ouvrir une base : pour ouvrir la base des frames originales de la vidéo.
  - Ouvrir la base détecté : pour ouvrir la base des frames détectées de la vidéo pour calculer les valeurs statistiques.
  - Ouvrir une vidéo : Pour accéder aux vidéo à partir d'un fichier vidéo sur l'ordinateur.
  - Ségmentation de la vidéo : segmenter la vidéo en frames



2. L'étape suivante est détection et suivi d'objet en mouvement.

- Détection et suivi par GMM : pour lancer l'exécution de la détection à l'aide de GMM et le suivi.
- Détection et suivi par GMM amélioré : pour lancer l'exécution de la détection à l'aide de GMM amélioré et le suivi.



Ici, nous montrons le résultat de la détection et le suivi en utilisant le GMM traditionnel et aussi nous montrons le résultat de la détection et suivi en utilisant notre méthode(GMM amélioré).

Les résultat de notre application pour cette étape est représenté dans (Figures 4.5, 4.6).

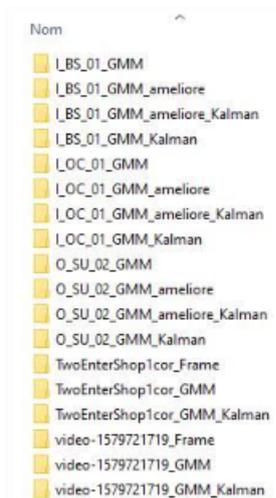


FIGURE 4.5 – Résultat de détection et suivi en utilisant le GMM par vidéo



FIGURE 4.6 – Résultat de détection et suivi en utilisant le GMM amélioré par vidéo

Les résultats peuvent être vérifiés en enregistrant les résultats précédemment affichés et en les affichant frame par frame(Figure 4.7) .



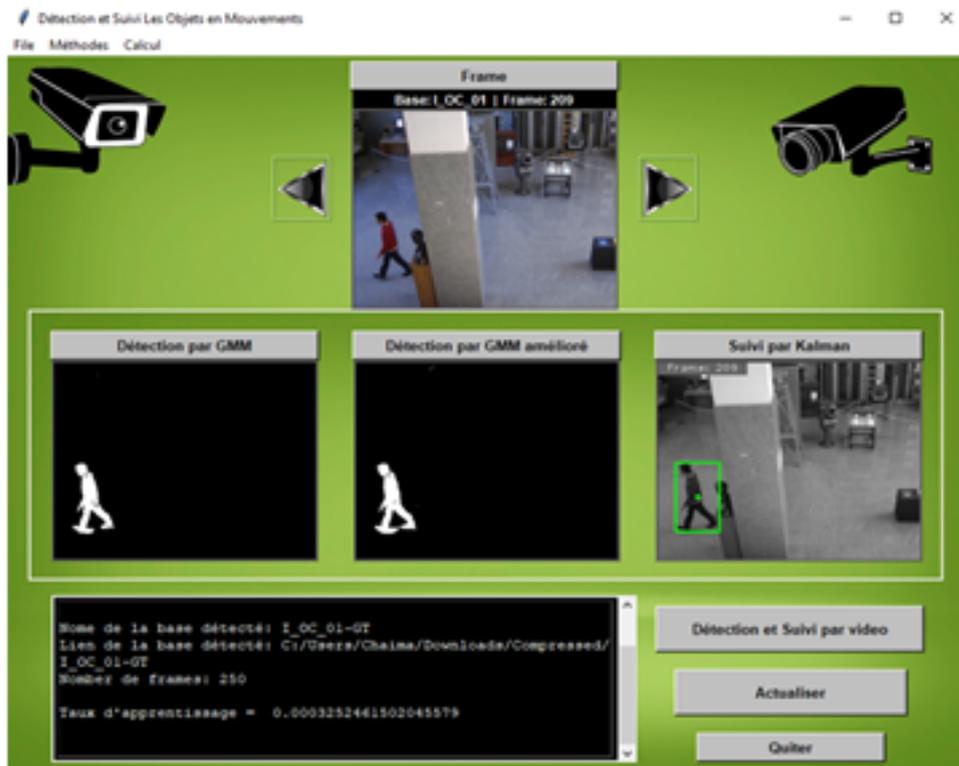
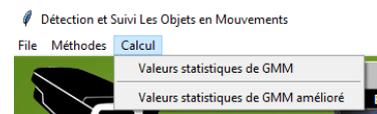


FIGURE 4.7 – Résultat de détection et suivi par frames

3. La différence entre les résultats de la détection à l'aide de GMM et GMM améliorés peut être observée soit par inspection visuelle, soit en calculant les valeurs statistiques.

- Valeurs statistiques de GMM : Pour calculé les vrai positif, négatif, faux positif et faux négatif entre les frames détectés à l'aide de GMM et les frames détectés de la base de donnée choisie.
- Valeurs statistiques de GMM amélioré : Pour calculé les vrai positif, vrai négatif, faux positif et faux négatif entre les frames détectés à l'aide de GMM amélioré et les frames détectés de la base de donnée choisie.



Les résultats de cette étape est représenté dans (Figures 4.8, 4.11).

```
Les valeurs statistiques de la methode GMM:
-----
Vrai Negatif = 85.81424400252531
Faux Negatif = 14.185755997474722
Vrai Positif = 100.0
Faux Positif = 0.0
```

FIGURE 4.8 – Les valeurs statistique de GMM

```
Les valeurs statistiques de la methode GMM amélioré:
-----
Vrai Negatif = 96.33284801136348
Faux Negatif = 3.667151988636363
Vrai Positif = 100.0
Faux Positif = 0.0
```

FIGURE 4.9 – Les valeurs statistique de GMM amélioré

Où :

- Vrai negatif : Le pourcentage des pixels qui devraient être en arrière-plan et nous l' avons trouvés en arrière-plan.
- Faux negatif : Le pourcentage des pixels qui devraient être en arrière-plan et nous l'avons trouvé au avant-plan.
- Vrai positif : Le pourcentage des pixels qui devraient être au avant-plan et nous l'avons trouvés au avant-plan.
- Faux positif : Le pourcentage des pixels qui devraient être au avant-plan et nous l'avons trouvés en arrière-plan.

Les résultats de la détection sont bons lorsque le pourcentage de VP et VN élevé.

### 4.3.3 Test et résultats

Dans cette partie, nous avons présenté les résultats de plusieurs tests de détection et de suivi à l'aide de notre application.

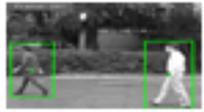
	<b>I_OC_01</b> Frame 208	<b>O_RA_02</b> Frame 241	<b>O_RA_01</b> Frame 944
<b>Frame</b>			
<b>GMM</b>			
<b>Valeurs statistique</b>	VN =85 FN = 14 VP = 100 FP = 0	VN =89 FN = 10 VP = 98 FP = 2	VN =84 FN = 15 VP = 92 FP = 8
<b>GMM amélioré</b>			
<b>Valeurs statistique</b>	VN =95 FN = 4 VP = 100 FP = 0	VN =98 FN = 1 VP = 97 FP = 3	VN =90 FN = 10 VP = 93 FP = 7
<b>Suivi par kALMAN</b>			

FIGURE 4.10 – Résultats de quelques vidéos de la base de données LASIESTA

	<b>VID_01</b> Frame 193	<b>VID_02</b> Frame 171
<b>frame</b>		
<b>GMM</b>		
<b>GMM amélioré</b>		
<b>Suivi par KALMAN</b>		

FIGURE 4.11 – Résultats de détection et suivi des vidéos filmées

## 4.4 Conclusion

Dans ce chapitre, les résultats de détection et de suivi qui ont été menés sur de nombreuses vidéos ont été présentés et les résultats étaient satisfaisants, mais il y a quelques défauts, tels que la vitesse d'exécution et ils sera étudié dans les futurs travaux.

# CONCLUSION GÉNÉRALE

## Conclusion

Dans ce travail, on s'est intéressé à deux des problèmes majeurs d'analyse de mouvement dans une séquence vidéo ; la détection et le suivi d'objets en mouvement dans des scènes simples (caméra fixe). Dans la première partie de ce mémoire on a présenté une étude de système de vidéo-surveillance et son architecture, et dans la deuxième partie, nous avons présenté les différentes méthodes de détection et de suivi d'objets en mouvement, avec une étude comparative de ces méthodes, puis nous avons détaillé les principes et les différentes étapes de la méthode de détection proposée dans la troisième partie en plus d'expliquer la méthode de suivi des objets détectés, où nous avons choisi une méthode de détection et de suivi des objets en mouvement en général, qu'il s'agisse de personnes ou d'objets solides tels que voitures. . . Dans le processus de détection de mouvement, nous avons utilisé le modèle de mélange gaussien et mis en œuvre une amélioration de cette méthode en l'incorporant dans l'algorithme d'entropie 2D, puis en procédant au filtre KALMAN pour suivre les objets détectés. La quatrième partie conclut la mémoire en présentant l'ensemble des résultats expérimentaux obtenus pour la détection et suivi d'objets en mouvement.

## Perspectives

En perspectives, on propose :

- L'application des méthodes de détection et de suivi d'objets des multi- scènes (plusieurs camera).
- L'application des méthodes de détection et de suivi sur les séquences à fonds dynamiques (caméras mobiles).

## BIBLIOGRAPHIE

- [1] Valerie Gouaillier and A Fleurant. Intelligent video surveillance : Promises and challenges. *Technological and commercial intelligence report, CRIM and Technôpole Defence and Security*, 456 :468, 2009.
- [2] Fereshteh Falah Chamasemani, Lilly Suriani Affendey, et al. Systematic review and classification on video surveillance systems. *International Journal of Information Technology and Computer Science (IJITCS)*, 5(7) :87, 2013.
- [3] Ying-Wen Bai, Zi-Li Xie, and Zong-Han Li. Design and implementation of a home embedded surveillance system with ultra-low alert power. *IEEE Transactions on Consumer Electronics*, 57(1) :153–159, 2011.
- [4] Marijn JH Loomans, Cornelis J Koeleman, and Peter HN De With. Low-complexity wavelet-based scalable image & video coding for home-use surveillance. *IEEE Transactions on Consumer Electronics*, 57(2) :507–515, 2011.
- [5] Teddy Ko. A survey on behavior analysis in video surveillance for homeland security applications. In *2008 37th IEEE Applied Imagery Pattern Recognition Workshop*, pages 1–8. IEEE, 2008.
- [6] Kosmas Dimitropoulos, Theodoros Semertzidis, and Nikolaos Grammalidis. Video and signal based surveillance for airport applications. In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 170–175. IEEE, 2009.
- [7] Jia-Luen Chua, Yoong Choon Chang, and Wee Keong Lim. A simple vision-based fall detection technique for indoor video surveillance. *Signal, Image and Video Processing*, 9(3) :623–633, 2015.
- [8] Hao Yang, Jianguang Lou, Hongzan Sun, Weiming Hu, and Tieniu Tan. Efficient and robust vehicle localization. In *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*, volume 2, pages 355–358. IEEE, 2001.
- [9] Abderrahmane Ezzahout and Rachid Oulad Haj Thami. Conception and development of a video surveillance system for detecting, tracking and profile analysis of a person. In *2013 3rd international symposium ISKO-Maghreb*, pages 1–5. IEEE, 2013.
- [10] Peng Zhang, Yanning Zhang, Tony Thomas, and Sabu Emmanuel. Moving people tracking with detection by latent semantic analysis for visual surveillance applications. *Multimedia tools and applications*, 68(3) :991–1021, 2014.

- [11] Lei Wang and Fuhai Li. The design of real-time monitoring system for enterprises in complex environments. In *2012 International Conference on Systems and Informatics (ICSAI2012)*, pages 306–314. IEEE, 2012.
- [12] Pablo Negri. Estimating the queue length at street intersections by using a movement feature space approach. *IET Image Processing*, 8(7) :406–416, 2014.
- [13] Le An, Bir Bhanu, and Songfan Yang. Unified face representation for individual recognition in surveillance videos. In *Wide Area Surveillance*, pages 123–136. Springer, 2014.
- [14] Le An, Mehran Kafai, and Bir Bhanu. Face recognition in multi-camera surveillance videos using dynamic bayesian network. In *2012 Sixth International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–6. IEEE, 2012.
- [15] Demet Duman and Gözde Bozdağlı Akar. Moving vehicle classification. In *2013 21st Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE, 2013.
- [16] Hua-Tsung Chen, Li-Wu Tsai, Hui-Zhen Gu, Suh-Yin Lee, and Bao-Shuh P Lin. Traffic congestion classification for nighttime surveillance videos. In *2012 IEEE International Conference on Multimedia and Expo Workshops*, pages 169–174. IEEE, 2012.
- [17] Xiling Luo, Yanxiong Wu, Yan Huang, and Jun Zhang. Vehicle flow detection in real-time airborne traffic surveillance system. *Transactions of the Institute of Measurement and Control*, 33(7) :880–897, 2011.
- [18] Max Krüger, Jürgen Ziegler, and Kathrin Heller. A generic bayesian network for identification and assessment of objects in maritime surveillance. In *2012 15th International Conference on Information Fusion*, pages 2309–2316. IEEE, 2012.
- [19] CL Lai, JC Yang, and YH Chen. A real time video processing based surveillance system for early fire and flood detection. In *2007 IEEE Instrumentation & Measurement Technology Conference IMTC 2007*, pages 1–6. IEEE, 2007.
- [20] Kimin Yun, Hawook Jeong, Kwang Moo Yi, Soo Wan Kim, and Jin Young Choi. Motion interaction field for accident detection in traffic surveillance video. In *2014 22nd International Conference on Pattern Recognition*, pages 3062–3067. IEEE, 2014.
- [21] Dalia Coppi, Simone Calderara, and Rita Cucchiara. Iterative active querying for surveillance data retrieval in crime detection and forensics. In *4th International Conference on Imaging for Crime Detection and Prevention 2011 (ICDP 2011)*, pages 1–6. IET, 2011.
- [22] Le Lv, Dongbin Zhao, and Zhijiang Fan. Cheating behavior detection based-on pictorial structure model. In *Proceedings of the 33rd Chinese Control Conference*, pages 7274–7279. IEEE, 2014.
- [23] Julfa Tuty and Bailing Zhang. Simultaneous object tracking and classification for traffic surveillance. In *Proceedings of International Conference on Computer Science and Information Technology*, pages 749–755. Springer, 2014.
- [24] Amal Ben Hamida. *Vers une nouvelle architecture de videosurveillance basée sur la scalabilité orientée vers l'application*. PhD thesis, Université de Bordeaux, 2016.

- [25] Jean-Baptiste Milési. *DE LA VIDEOSURVEILLANCE A LA VIDEOPROTECTION : Evolution du régime juridique et mutation de la technique de vidéosurveillance depuis la loi du 21 janvier 1995*. PhD thesis, dissertation, Faculté de droit, Université de Strasbourg, 2011.
- [26] Djamila Mokhtari. Détection des chutes par calcul homographique. 2012.
- [27] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking : A survey acm computing survey. *CSUR*, 38(4) :13, 2002.
- [28] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking : A benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2411–2418, 2013.
- [29] Yedjour Hayat. *Détection de contours et suivi d’objet dans une séquence d’images par les réseaux de neurones impulsionsnels*. PhD thesis, Université mohamed boudiaf des sciences et de la technologie d’oran, 2010.
- [30] Mohamed Dahmane. Système de vidéosurveillance et de monitoring. 2004.
- [31] Jacinto C Nascimento and Jorge S Marques. Performance evaluation of object detection algorithms for video surveillance. *IEEE Transactions on Multimedia*, 8(4) :761–774, 2006.
- [32] Philippe Loic Marie Bouttefroy, Abdesselam Bouzerdoum, Son Lam Phung, and Azeddine Beghdadi. On the analysis of background subtraction techniques using gaussian mixture models. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4042–4045. IEEE, 2010.
- [33] Robert T Collins, Alan J Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yanghai Tsin, David Tolliver, Nobuyoshi Enomoto, Osamu Hasegawa, Peter Burt, et al. A system for video surveillance and monitoring. *VSAM final report*, 2000(1-68) :1, 2000.
- [34] V Nicolas. Suivi d’objets en mouvement dans une séquence vidéo. *These de Doctorat, Université Paris Descartes*, 2007.
- [35] Robert T Collins, Alan J Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yanghai Tsin, David Tolliver, Nobuyoshi Enomoto, Osamu Hasegawa, Peter Burt, et al. A system for video surveillance and monitoring. *VSAM final report*, 2000(1-68) :1, 2000.
- [36] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1) :119–139, 1997.
- [37] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. Ieee, 2001.
- [38] Paul Viola, Michael Jones, and Daniel Snow. Detecting pedestrians using patterns of motion and appearance. 2003.
- [39] Constantine P Papageorgiou, Michael Oren, and Tomaso Poggio. A general framework for object detection. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pages 555–562. IEEE, 1998.

- [40] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.
- [41] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9) :1627–1645, 2010.
- [42] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. Ensemble of exemplar-svm for object detection and beyond. In *2011 International conference on computer vision*, pages 89–96. IEEE, 2011.
- [43] Saadia MEDOUAKH. *Détection et suivi d'objets*. PhD thesis, Université Mohamed Khider-Biskra, 2019.
- [44] Peter E Hart, David G Stork, and Richard O Duda. *Pattern classification*. Wiley Hoboken, 2000.
- [45] Yelakan Berenger Ouattara, Tokpa Arsene Koba, Geneviève Baudoin, Jean-Marc Laheurte, et al. Knn and svm classification for chainsaw identification in the forest areas. *International journal of advanced computer science and applications (IJACSA)*, 10(12), 2019.
- [46] Mokhtar Tafar. Initialisation a l'apprentissage automatique. pages 42–44, 2019.
- [47] Jerome Y Lettvin, Humberto R Maturana, Warren S McCulloch, and Walter H Pitts. What the frog's eye tells the frog's brain. *Proceedings of the IRE*, 47(11) :1940–1951, 1959.
- [48] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *IEEE Transactions on pattern analysis and machine intelligence*, 25(5) :564–577, 2003.
- [49] Michael J Black and Allan D Jepson. Eigenttracking : Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26(1) :63–84, 1998.
- [50] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking : A benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2411–2418, 2013.
- [51] George Paschos. Perceptually uniform color spaces for color texture analysis : an empirical evaluation. *IEEE transactions on Image Processing*, 10(6) :932–937, 2001.
- [52] Deqing Sun, Stefan Roth, and Michael J Black. Secrets of optical flow estimation and their principles. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 2432–2439. IEEE, 2010.
- [53] Bangalore S Manjunath and Wei-Ying Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on pattern analysis and machine intelligence*, 18(8) :837–842, 1996.
- [54] Koen Van De Sande, Theo Gevers, and Cees Snoek. Evaluating color descriptors for object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 32(9) :1582–1596, 2009.

- [55] Weiming Hu, Tieniu Tan, Liang Wang, and Steve Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 34(3) :334–352, 2004.
- [56] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking : A survey. *Acm computing surveys (CSUR)*, 38(4) :13–es, 2006.
- [57] Shengfeng He, Qingxiong Yang, Rynson WH Lau, Jiang Wang, and Ming-Hsuan Yang. Visual tracking via locality sensitive histograms. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2427–2434, 2013.
- [58] Xi Li, Weiming Hu, Chunhua Shen, Zhongfei Zhang, Anthony Dick, and Anton Van Den Hengel. A survey of appearance models in visual object tracking. *ACM transactions on Intelligent Systems and Technology (TIST)*, 4(4) :1–48, 2013.
- [59] Ramesh Jain and H-H Nagel. On the analysis of accumulative difference pictures from image sequences of real world scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2) :206–214, 1979.
- [60] Cor J Veenman, Marcel JT Reinders, and Eric Backer. Resolving motion correspondence for densely moving points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1) :54–72, 2001.
- [61] Khurram Shafique and Mubarak Shah. A noniterative greedy algorithm for multiframe point correspondence. *IEEE transactions on pattern analysis and machine intelligence*, 27(1) :51–65, 2005.
- [62] Paul Scovanner and Marshall F Tappen. Learning pedestrian dynamics from the real world. In *2009 IEEE 12th International Conference on Computer Vision*, pages 381–388. IEEE, 2009.
- [63] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking : A survey. *Acm computing surveys (CSUR)*, 38(4) :13–es, 2006.
- [64] Gandham Sindhuja, Renuka Devi, et al. A survey on detection and tracking of objects in video sequence. *International Journal of Engineering Research and General Science*, 3(2) :418–426, 2015.
- [65] S Bhuvaneswari and TS Subashini. Tracking manually selected object in videos using color histogram matching. *Journal of Theoretical & Applied Information Technology*, 67(3), 2014.
- [66] Kinjal A Joshi and Darshak G Thakore. A survey on moving object detection and tracking in video surveillance system. *International journal of soft computing and engineering*, 2(3) :44–48, 2012.
- [67] Ted J Broida and Rama Chellappa. Estimation of object motion parameters from noisy images. *IEEE transactions on pattern analysis and machine intelligence*, (1) :90–99, 1986.
- [68] David Beymer and Kurt Konolige. Real-time tracking of multiple people using continuous detection. In *IEEE Frame Rate Workshop*, pages 1–8. Citeseer, 1999.
- [69] Genshiro Kitagawa. Non-gaussian state—space modeling of nonstationary time series. *Journal of the American statistical association*, 82(400) :1032–1041, 1987.

- [70] Elise Arnaud, Etienne Memin, and Bruno Cernuschi-Frias. Conditional filters for image sequence-based tracking-application to point tracking. *IEEE Transactions on image processing*, 14(1) :63–79, 2004.
- [71] Y-L Chang and JK Aggarwal. 3d structure reconstruction from an ego motion sequence using statistical estimation and detection theory. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 268–273. IEEE, 1991.
- [72] Christopher Rasmussen and Gregory D. Hager. Probabilistic data association methods for tracking complex visual objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6) :560–576, 2001.
- [73] Donald Reid. An algorithm for tracking multiple targets. *IEEE transactions on Automatic Control*, 24(6) :843–854, 1979.
- [74] Stan Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Proceedings. 1998 IEEE Computer Society conference on computer vision and pattern recognition (Cat. No. 98CB36231)*, pages 232–237. IEEE, 1998.
- [75] Haim Schweitzer, James Wesley Bell, and Feng Wu. Very fast template matching. In *European Conference on Computer Vision*, pages 358–372. Springer, 2002.
- [76] Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felsberg, Luka Cehovin, Gustavo Fernandez, Tomas Vojir, Gustav Hager, Georg Nebehay, and Roman Pflugfelder. The visual object tracking vot2015 challenge results. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 1–23, 2015.
- [77] Carlo Tomasi and Takeo Kanade. Detection and tracking of point. *Int J Comput Vis*, 9 :137–154, 1991.
- [78] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking : A survey. *Acm computing surveys (CSUR)*, 38(4) :13–es, 2006.
- [79] Michael J Black and Allan D Jepson. Eigentracking : Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26(1) :63–84, 1998.
- [80] Yang Hua. *Towards robust visual object tracking : proposal selection and occlusion reasoning*. PhD thesis, Université Grenoble Alpes, 2016.
- [81] Shai Avidan. Support vector tracking. *IEEE transactions on pattern analysis and machine intelligence*, 26(8) :1064–1072, 2004.
- [82] Daniel P Huttenlocher, Jae J Noh, and William J Rucklidge. Tracking non-rigid objects in complex scenes. In *1993 (4th) International Conference on Computer Vision*, pages 93–101. IEEE, 1993.
- [83] Min Huang, Gang Chen, Guo-feng Yang, and Rui Cao. An algorithm of the target detection and tracking of the video. *Procedia Engineering*, 29 :2567–2571, 2012.
- [84] Zezhi Chen and Tim Ellis. A self-adaptive gaussian mixture model. *Computer Vision and Image Understanding*, 122 :35–46, 2014.
- [85] Demetri Terzopoulos and Richard Szeliski. Tracking with kalman snakes. *Active vision*, 20 :3–20, 1992.
- [86] Michael Isard and Andrew Blake. Condensation—conditional density propagation for visual tracking. *International journal of computer vision*, 29(1) :5–28, 1998.

- [87] A-R Mansouri. Region tracking via level set pdes without motion computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7) :947–961, 2002.
- [88] Daniel Cremers and Christoph Schnörr. Statistical shape knowledge in variational motion segmentation. *Image and Vision Computing*, 21(1) :77–86, 2003.
- [89] Alper Yilmaz, Xin Li, and Mubarak Shah. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Transactions on pattern analysis and machine intelligence*, 26(11) :1531–1536, 2004.
- [90] Remi Ronfard. Region-based strategies for active contour models. *International journal of computer vision*, 13(2) :229–251, 1994.
- [91] Marcelo Bertalmio, Guillermo Sapiro, and Gregory Randall. Morphing active contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7) :733–737, 2000.
- [92] Xiaomeng Wang, Dequn Zhao, Guangmin Sun, Xingwang Liu, and Yanli Wu. Target detection algorithm based on improved gaussian mixture model. In *2015 2nd International Conference on Electrical, Computer Engineering and Electronics*, pages 846–850. Atlantis Press, 2015.
- [93] CI Patel and Ripal Patel. Gaussian mixture model based moving object detection from video sequence. In *Proceedings of the International Conference & Workshop on Emerging Trends in Technology*, pages 698–702, 2011.
- [94] Bo Jiao, Liaoliao Yan, and Wei Li. A fast convergent gaussian mixture model for moving object detection. *Journal of image and graphics*, 13 :2139–2143, 2008.
- [95] Gang Feng. Moving objects detection algorithm of improved gmm based on 2-d entropy. In *2nd International Forum on Management, Education and Information Technology Application (IFMEITA 2017)*, pages 411–415. Atlantis Press, 2018.
- [96] António Almeida, Jorge Almeida, and Rui Araújo. Real-time tracking of multiple moving objects using particle filters and probabilistic data association. *Automatika : časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije*, 46(1-2) :39–48, 2005.
- [97] Eli Brookner. Tracking and kalman filtering made easy. 1998.
- [98] Prasad Kalane and P Loni. Target tracking using kalman filter. *International Journal of Science & Technology*, 2(2), 2012.
- [99] [https://www.gti.ssr.upm.es/data/lasiesta\\_database](https://www.gti.ssr.upm.es/data/lasiesta_database).