



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
UNIVERSITE KASDI MERBAH OUARGLA



Faculté des Nouvelles technologies de l'information et de la  
communication  
Département D'électronique

## MEMOIRE MASTER ACADEMIQUE

Domaine : Électronique  
Spécialité : Automatique

Présenté par :

LABABSA Hafsa Imen.

MISSOUN Nour El-imane.

### Thème

# *Détection et classification des obstacles dynamiques*

Soutenu publiquement  
le : 01/06/2016

Devant le jury :

Mr	Chebara	MCB	Président	UKM Ouargla
Mr	Hamza	MAB	Examineur	UKM Ouargla
Mr	Djeddi	MAB	Examineur	UKM Ouargla
Mr	Djalal Djerah	MAA	Encadreur	UKM Ouargla
Mr	Meraoumia Abdallah	MCB	Co-Encadreur	UKM Ouargla

Année Universitaire : 2015 /2016

## *Remerciements*

*Tout d'abord, nous remercions Dieu, notre créateur de nous  
avoir donné la force, la volonté et le courage afin  
d'accomplir ce modeste travail.*

*Nous adressons le grand remerciement à notre encadreur  
qui a proposé le thème de ce mémoire, pour ses conseils du  
début à la fin de ce travail.*

*Nous tenons également à remercier messieurs, mes dames les  
membres de jury pour l'honneur qu'ils nous ont fait en  
acceptant de siéger à notre soutenance.*

*Finalement, nous tenons à exprimer notre profonde  
gratitude à nos familles qui nous ont toujours soutenues et  
à tous ceux qui ont participé à réaliser ce mémoire.*

*Ainsi qu'à l'ensemble des enseignants qui ont contribué  
à notre formation.*

## Sommaire

### **Introduction générale**

Introduction.....	i
Objectif du mémoire.....	ii
Structure de mémoire.....	ii

### **Chapitre I :**

#### **Vision par ordinateur et détection de mouvement**

Introduction.....	01
1. Détection de mouvement.....	02
2. Architecture générale des systèmes de détection et de suivie de mouvement.....	02
3. Système d'acquisition de vidéo .....	02
4. La détection des régions en mouvement .....	03
4.1 Les techniques de détection des régions en mouvement.....	03
4.1.1 Flux optique.....	03
4.1.2 La différence temporelle des images consécutive .....	04
4.1.3 Double différence temporelle des images consécutive.....	05
4.1.4 Soustraction de l'arrière-plan.....	07
4.2 Approche proposée .....	08

4.2.1 Algorithme de soustraction de fond.....	08
4.2.2 Autres manières pour la Création d'image de fond.....	11
4.2.3 La Combinaison des images résultantes par des images de contours.....	13
4.2.3.1 Détection des contours.....	13
4.2.3.2 Détecteur de contour Canny.....	13
5. Localisation et classification des objets en mouvement .....	16
6. Extraction des propriétés des objets en mouvement.....	16
7. Le suivi de mouvement .....	17
Conclusion.....	18

## Chapitre II :

### suivi d'objets et filtre de Kalman

Introduction.....	19
1. Suivi de mouvement.....	20
2. Quelques applications du suivi de mouvement.....	20
3. Notre objectif.....	22
4. Structure générale des algorithmes de suivi.....	22
5. Méthodologie des Algorithmes.....	23
6. Problématique .....	24
7. La solution proposée .....	24
8. Le filtre de kalman.....	24
8.1 Paramètres de filtre de Kalman.....	25

8.2 Principe et fonctionnement de filtre .....	25
8.3 Cas des systèmes non linéaire.....	29
9. Filtre de Kalman pour le suivi des multiples objets .....	30
9.1 Le suivi spatio-temporel des centres d masse.....	31
9.2 Proposition Modèle d'estimation.....	32
9.3 Algorithme proposée.....	33
10. Les avantages et points forts de filtre .....	34
Conclusion.....	35

## Chapitre III:

### Implémentation et discussion des résultats

Introduction.....	36
1. Software et Langage de programmation .....	36
2. Hardware .....	37
3. Objectif de cette étude.....	37
4. Synoptique générale de système .....	37
5. Les étapes à suivre.....	38
6. Choix de Simulink.....	39
7. Le Modèle Simulink proposé pour la détection et suivi des objets en mouvement.....	39
8. Résultats et Analyse Expérimentale.....	42
I. 1 <sup>er</sup> Cas : vidéos enregistrées.....	42
I.1 Détection de mouvement par soustraction de fond.....	43
I.2 Résultats de détection et suivi de mouvement.....	44
I.3 Implantation du filtre de kalman pour l'estimation et le suivi.....	49

I.4 Résultats obtenu de l'implémentation de KF.....	50
I.4.1 Etude de l'influence de quelque paramètre de filtre sur le suivi.....	50
1.4.1.1 Influence de la matrice de la covariance du bruit de mesure ...	50
1.4.1.2 Influence de la matrice de covariance du bruit de processus ....	53
1.4.1.3 Comparaison des résultats précédents.....	55
1.4.1.4 Extractions de la vitesse des véhicules suivis.....	59
1.4.1.5 la trajectoire des véhicules suivis par les quartes rectangles...	60
II. 2 <sup>ème</sup> Cas : En temps réel.....	63
II.1 Résultats de la détection et suivi de mouvement en temps réel.....	63
II.2 Résultats de suivi des centres de masse.....	66
Conclusion.....	69
Conclusion générale.....	iii
Bibliographie.....	vi

# Liste des figures

## *Chapitre 1 : Vision par ordinateur et Détection de mouvement*

<i>Figure 1.1 : Schéma général de système de détection de mouvement.....</i>	<i>2</i>
<i>Figure 1.2: Schéma du système d'acquisition de vidéo.....</i>	<i>3</i>
<i>Figure 1. 3 : Détection par la méthode du flux optique.....</i>	<i>4</i>
<i>Figure 1.4: Un exemple de la différence temporelle, (a) une scène simple avec deux Objets, (b) les régions rouges sont la différence entre deux images consécutives .....</i>	<i>5</i>
<i>Figure 1.5: Schéma fonctionnel de la méthode de différence temporelle.....</i>	<i>5</i>
<i>Figure 1.6: Schéma fonctionnel de la méthode double différence temporelle.....</i>	<i>6</i>
<i>Figure 1.7: Principe de soustraction du fond, (a) image courante, (b) image du fond,(c)image après la soustraction du fond.....</i>	<i>8</i>
<i>Figure 1.8:Schéma fonctionnel de la méthode Canny- Background adaptatif.....</i>	<i>11</i>
<i>Figure 1.9: (a) image en niveau de gris, (b) image après la détection de contour (Canny).....</i>	<i>12</i>
<i>Figure 1.10 : (a) Image originale en couleur. (b) Image en niveaux de gris.....</i>	<i>14</i>
<i>Figure 1.11 : Opération de dilatation .....</i>	<i>15</i>
<i>Figure 1.12: Opération d'érosion .....</i>	<i>15</i>

## Chapitre 2 : Suivi d'objets et Filtre de Kalman

<b>Figure 2.1 :</b> Quelques applications du suivi et de la classification d'objets.....	20
<b>Figure 2.2:</b> exemple des applications robotique utilise le suivie.....	21
<b>Figure 2.3:</b> exemple de Surveillance du trafic routier.....	21
<b>Figure 2.4:</b> reconnaissance d'action dans le domaine sportif.....	22
<b>Figure 2.5:</b> Schéma d'un système typique de surveillance.....	23
<b>Figure 2.6:</b> Boucle de rétroaction récursive du filtre de Kalman.....	25
<b>Figure 2.7:</b> Cycle d'estimation et prédiction par filtre de Kalman.....	28
<b>Figure 2.8:</b> Boucle de prédiction et de correction ... ..	31
<b>Figure 2.9 :</b> Illustration de la détection de l'intersection.....	34

## Chapitre 3: Implémentation et discussion des résultats

<b>Figure 3.1.</b> Synoptique général de la présente étude.....	37
<b>Figure.3.2 :</b> Block Diagramme de Système.....	39
<b>Figure 3.3 :</b> Le Modèle Simulink proposé.....	40
<b>Figure 3.4 :</b> Résultats de la détection de mouvement avec des paramètres n°1.....	45
<b>Figure 3.5 :</b> Résultats de la détection de mouvement avec les paramètres n°2.....	47



## Liste des figures

<b>Figure 3.6 : Résultats de suivi avec les paramètres <math>n^{\circ}2</math>.....</b>	<b>48</b>
<b>Figure 3.7 : Les courbes <math>\hat{X}(t)</math> et <math>X(t)</math> des positions des centres de masse.....</b>	<b>51</b>
<b>Figure 3.8 : Les courbes <math>\hat{Y}(t)</math> et <math>Y(t)</math> des positions des centres de masse.....</b>	<b>51</b>
<b>Figure 3. (9.a) : Résultat de l'erreur de mesure des positions <math>\tilde{X}(t)</math>.....</b>	<b>52</b>
<b>Figure 3. (9.b) : résultat de l'erreur de mesure des positions <math>\tilde{Y}(t)</math>.....</b>	<b>52</b>
<b>Figure 3.(10.a): Les courbes <math>\hat{X}(t)</math> et <math>X(t)</math> des positions des centres de masse.....</b>	<b>53</b>
<b>Figure 3.(10.b): Les courbes <math>\hat{Y}(t)</math> et <math>Y(t)</math> des positions des centres de masse.....</b>	<b>54</b>
<b>Figure 3.11.a : Résultat de l'erreur de mesure de positions <math>\tilde{X}(t)</math>.....</b>	<b>54</b>
<b>Figure 3.(11.b) : Résultat de l'erreur de mesure de position <math>\tilde{Y}(t)</math>.....</b>	<b>55</b>
<b>Figure 3.(12.a) : Comparaison des signaux <math>\tilde{X}(T)</math> dans les deux cas de mesure.....</b>	<b>58</b>
<b>Figure 3.(12.b) : Comparaison des signaux <math>\tilde{Y}(T)</math> dans les deux cas de mesure.....</b>	<b>58</b>
<b>Figure 3.(13.a) : Résultats des mesures de la vitesse <math>V_x(t)</math> réelle et estimée.....</b>	<b>59</b>
<b>Figure 3.(13.b) : la vitesse <math>V_y(t)</math> réelle et estimée mesurée par les 4 rectangles.....</b>	<b>60</b>
<b>Figure 3.(14.a) :Trajectoire des véhicules suivis par le rectangle 1.....</b>	<b>61</b>
<b>Figure 3.(14.b) :Trajectoire des véhicules suivis par le rectangle 2.....</b>	<b>61</b>
<b>Figure 3.(14.c) : Trajectoire des véhicules suivis par le rectangle 3.....</b>	<b>62</b>
<b>Figure 3.(14.d) :Trajectoire des véhicules suivis par le rectangle 4.....</b>	<b>62</b>

<i>Figure 3.15 : Résultats de la détection et suivi des objets mouvant avec les mains.....</i>	<i>64</i>
<i>Figure 3.16 : L'effet sur la détection lorsque l'objet s'approche du camera.....</i>	<i>65</i>
<i>Figure 3.17 : Résultats de la détection et suivi des corps humains.....</i>	<i>66</i>
<i>Figure 3.18 : Quelques trames qui représente le mouvement.....</i>	<i>67</i>
<i>Figure 3.(19.a): La trajectoire obtenue de mouvement verticale de l'objet 1.....</i>	<i>67</i>
<i>Figure 3.(19.b): La trajectoire obtenue de mouvement horizontale de l'objet 2.....</i>	<i>68</i>

# Liste des figures

## *Chapitre 1 : Vision par ordinateur et Détection de mouvement*

<i>Figure 1.1 : Schéma général de système de détection de mouvement.....</i>	<i>2</i>
<i>Figure 1.2: Schéma du système d'acquisition de vidéo.....</i>	<i>3</i>
<i>Figure 1. 3 : Détection par la méthode du flux optique.....</i>	<i>4</i>
<i>Figure 1.4: Un exemple de la différence temporelle, (a) une scène simple avec deux Objets, (b) les régions rouges sont la différence entre deux images consécutive .....</i>	<i>5</i>
<i>Figure 1.5: Schéma fonctionnel de la méthode de différence temporelle.....</i>	<i>5</i>
<i>Figure 1.6: Schéma fonctionnel de la méthode double différence temporelle.....</i>	<i>6</i>
<i>Figure 1.7: Principe de soustraction du fond, (a) image courante, (b) image du fond,(c)image après la soustraction du fond.....</i>	<i>8</i>
<i>Figure 1.8:Schéma fonctionnel de la méthode Canny- Background adaptatif.....</i>	<i>11</i>
<i>Figure 1.9: (a) image en niveau de gris, (b) image après la détection de contour (Canny).....</i>	<i>12</i>
<i>Figure 1.10 : (a) Image originale en couleur. (b) Image en niveaux de gris.....</i>	<i>14</i>
<i>Figure 1.11 : Opération de dilatation .....</i>	<i>15</i>
<i>Figure 1.12: Opération d'érosion .....</i>	<i>15</i>

## Chapitre 2 : Suivi d'objets et Filtre de Kalman

<b>Figure 2.1 :</b> Quelques applications du suivi et de la classification d'objets.....	20
<b>Figure 2.2:</b> exemple des applications robotique utilise le suivie.....	21
<b>Figure 2.3:</b> exemple de Surveillance du trafic routier.....	21
<b>Figure 2.4:</b> reconnaissance d'action dans le domaine sportif.....	22
<b>Figure 2.5:</b> Schéma d'un système typique de surveillance.....	23
<b>Figure 2.6:</b> Boucle de rétroaction récursive du filtre de Kalman.....	25
<b>Figure 2.7:</b> Cycle d'estimation et prédiction par filtre de Kalman.....	28
<b>Figure 2.8:</b> Boucle de prédiction et de correction ... ..	31
<b>Figure 2.9 :</b> Illustration de la détection de l'intersection.....	34

## Chapitre 3: Implémentation et discussion des résultats

<b>Figure 3.1.</b> Synoptique général de la présente étude.....	37
<b>Figure.3.2 :</b> Block Diagramme de Système.....	39
<b>Figure 3.3 :</b> Le Modèle Simulink proposé.....	40
<b>Figure 3.4 :</b> Résultats de la détection de mouvement avec des paramètres n°1.....	45
<b>Figure 3.5 :</b> Résultats de la détection de mouvement avec les paramètres n°2.....	47

## Liste des figures

<b>Figure 3.6 : Résultats de suivi avec les paramètres <math>n^{\circ}2</math>.....</b>	<b>48</b>
<b>Figure 3.7 : Les courbes <math>\hat{X}(t)</math> et <math>X(t)</math> des positions des centres de masse.....</b>	<b>51</b>
<b>Figure 3.8 : Les courbes <math>\hat{Y}(t)</math> et <math>Y(t)</math> des positions des centres de masse.....</b>	<b>51</b>
<b>Figure 3. (9.a) : Résultat de l'erreur de mesure des positions <math>\tilde{X}(t)</math>.....</b>	<b>52</b>
<b>Figure 3. (9.b) : résultat de l'erreur de mesure des positions <math>\tilde{Y}(t)</math>.....</b>	<b>52</b>
<b>Figure 3.(10.a): Les courbes <math>\hat{X}(t)</math> et <math>X(t)</math> des positions des centres de masse.....</b>	<b>53</b>
<b>Figure 3.(10.b): Les courbes <math>\hat{Y}(t)</math> et <math>Y(t)</math> des positions des centres de masse.....</b>	<b>54</b>
<b>Figure 3.11.a : Résultat de l'erreur de mesure de positions <math>\tilde{X}(t)</math>.....</b>	<b>54</b>
<b>Figure 3.(11.b) : Résultat de l'erreur de mesure de position <math>\tilde{Y}(t)</math>.....</b>	<b>55</b>
<b>Figure 3.(12.a) : Comparaison des signaux <math>\tilde{X}(T)</math> dans les deux cas de mesure.....</b>	<b>58</b>
<b>Figure 3.(12.b) : Comparaison des signaux <math>\tilde{Y}(T)</math> dans les deux cas de mesure.....</b>	<b>58</b>
<b>Figure 3.(13.a) : Résultats des mesures de la vitesse <math>V_x(t)</math> réelle et estimée.....</b>	<b>59</b>
<b>Figure 3.(13.b) : la vitesse <math>V_y(t)</math> réelle et estimée mesurée par les 4 rectangles.....</b>	<b>60</b>
<b>Figure 3.(14.a) :Trajectoire des véhicules suivis par le rectangle 1.....</b>	<b>61</b>
<b>Figure 3.(14.b) :Trajectoire des véhicules suivis par le rectangle 2.....</b>	<b>61</b>
<b>Figure 3.(14.c) : Trajectoire des véhicules suivis par le rectangle 3.....</b>	<b>62</b>
<b>Figure 3.(14.d) :Trajectoire des véhicules suivis par le rectangle 4.....</b>	<b>62</b>

<i>Figure 3.15 : Résultats de la détection et suivi des objets mouvant avec les mains.....</i>	<i>64</i>
<i>Figure 3.16 : L'effet sur la détection lorsque l'objet s'approche du camera.....</i>	<i>65</i>
<i>Figure 3.17 : Résultats de la détection et suivi des corps humains.....</i>	<i>66</i>
<i>Figure 3.18 : Quelques trames qui représente le mouvement.....</i>	<i>67</i>
<i>Figure 3.(19.a): La trajectoire obtenue de mouvement verticale de l'objet 1.....</i>	<i>67</i>
<i>Figure 3.(19.b): La trajectoire obtenue de mouvement horizontale de l'objet 2.....</i>	<i>68</i>

## ***LISTE DES TABLEAUX***

### ***Chapitre 1 :***

<b><i>Tableau 1.1 : Comparaison des deux méthodes Simple différence et Double différence.....</i></b>	<b><i>7</i></b>
---	-----------------

### ***Chapitre 3:***

<b><i>Tableau 3.1 : Description des ports utilisé dans notre étude.....</i></b>	<b><i>41</i></b>
---	------------------

<b><i>Tableau3.2 : Résultats de la mesure n°1 des positions réels et estimés.....</i></b>	<b><i>55</i></b>
---	------------------

<b><i>Tableau3.3 : Résultats de la mesure n°01 de l'erreur de la position des 4 rectangles de suivi.....</i></b>	<b><i>56</i></b>
--	------------------

<b><i>Tableau3.4 : Résultats de la mesure n°2 des positions réels et estimés.....</i></b>	<b><i>56</i></b>
---	------------------

<b><i>Tableau3.5 : Résultats de la mesure n°02 de l'erreur de positions.....</i></b>	<b><i>57</i></b>
--	------------------

# Introduction générale



## *Introduction générale*

La vision nous permet de percevoir et d'interpréter le monde qui nous entoure à travers l'œil qui est un dispositif biologique complexe. Le fonctionnement d'une caméra est comparé souvent avec le fonctionnement de l'œil. La vision artificielle a pour but de reproduire certaines fonctionnalités de la vision humaine à travers l'analyse d'images. C'est un problème qui n'est pas facile du fait que l'information de l'environnement (3D) est projetée sous forme d'images (2D) à travers des caméras. Les images représentent donc une scène tridimensionnelle sur un support en deux dimensions.

Par rapport à d'autres données sensorielles acquises par des capteurs exploités pour percevoir l'environnement, le traitement de données visuelles présente une difficulté particulière. Non seulement, comme dans les autres cas, il faut prendre en compte le fait que l'information fournie par la camera, n'est pas parfaite du fait de différentes sources d'erreurs (bruit, imprécision, etc.), mais des traitements spécifiques doivent considérer que cette projection de 3D en 2D entraîne une grande perte d'information [1].

Nous nous intéressons dans ce mémoire, aux traitements nécessaires pour détecter des objets spécifiques dans des images, puis pour suivre ces objets dans des séquences d'images, donc ce qu'il est convenu d'appeler des vidéos. Même si notre contexte applicatif est la robotique, ou la surveillance d'environnements, les algorithmes que nous proposons ne sont pas spécifiques à ce contexte : ils relèvent du domaine général de la segmentation des images, c'est-à-dire des traitements requis pour extraire d'une image, une entité spécifique, caractérisée par divers attributs (forme, couleur, texture, mouvement...).

La segmentation et le suivi d'objets sont des domaines de recherche dans la vision par ordinateur. Une de leurs applications importantes réside dans la robotique où la capacité à segmenter un objet d'intérêt du fond de l'image, d'une manière précise, est cruciale particulièrement dans des images acquises à bord durant le mouvement du robot. Segmenter un objet dans une image est une opération qui consiste à distinguer la région objet de celle du fond suivant un critère défini. Suivre un objet dans une séquence d'images est une opération qui consiste à localiser la région objet au fil du temps dans une vidéo. Plusieurs techniques

peuvent être utilisées afin d'assurer ces opérations [2]. Dans ce mémoire, nous nous sommes intéressés à segmenter et suivre des objets en utilisant la méthode de soustraction de fond qui est basée essentiellement sur la détection des contours en mouvement en raison de sa robustesse et son efficacité à pouvoir segmenter et suivre des objets non rigides.

## ***Objectif du mémoire :***

Dans ce travail, nous nous intéressons au suivi d'objets mobiles dans une scène perçue par une caméra couleur fixe, en se basant sur le filtre de Kalman, ce dernier représente un outil mathématique très important, permet d'améliorer la détection et le suivi des objets en robotique ou en vidéosurveillance.

## ***Structure du mémoire :***

Le présent travail est scindé en trois parties.

- La première partie sera consacrée au traitement de la vidéo et le suivi des objets par vision.
- La deuxième partie, nous aborderons l'utilité et la compréhension de filtre de Kalman et son algorithme.
- La troisième partie sera consacrée aux résultats de simulation.

Enfin, une conclusion générale fera la synthèse de l'étude et travail effectués et donnera la perspective quant à son évolution future.

# Chapitre I :

## Vision par ordinateur et détection de mouvement

## ***Introduction***

La détection d'objet est toujours une tâche importante dans le monde de la technologie, elle joue un rôle important dans la surveillance visuelle automatique, système de suivi et aussi pour la vision robotique pour éviter les obstacles dynamiques et la collision. Les applications de la détection d'objets sont nombreuses. On peut notamment citer le suivi d'objets, l'analyse du comportement, la compression vidéo, la reconstruction 3D...etc.

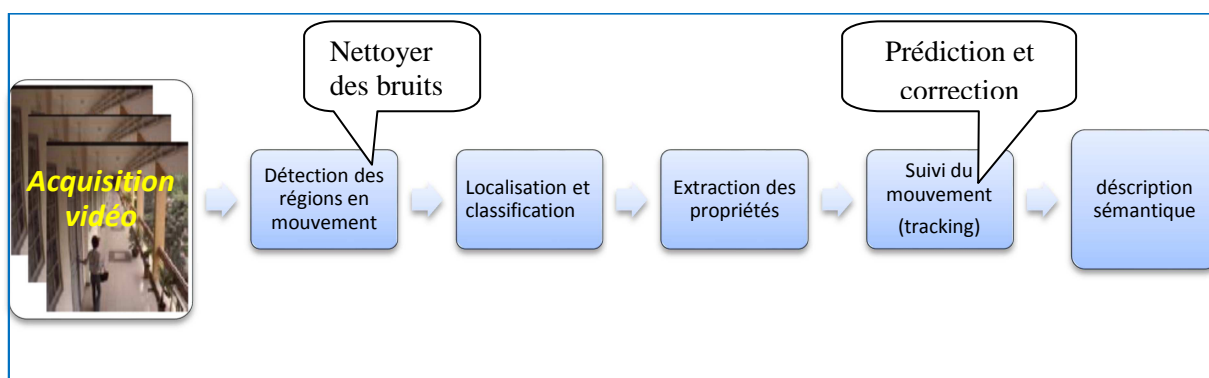
Les techniques de détection du mouvement peuvent être considérées comme un prétraitement permettant de réduire la quantité d'informations à analyser. Un objet en mouvement est détecté si sa position change relativement à celle d'un ensemble d'objets statiques ou s'il est localisé dans une image acquise 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.

Dans ce chapitre, nous nous intéressons à la détection d'objets en mouvement dans une séquence d'image, par conséquent aux méthodes de séparation des objets mobiles du fond de l'image (l'arrière-plan). Dans ce contexte, la méthode la plus utilisée est la méthode de soustraction de fond qui est basée essentiellement sur la détection des contours en mouvement. On présente également dans ce chapitre des autres méthodes : la méthode de flux optique et la méthode de la différence inter images (avec deux ou trois images consécutives). Puis nous précisons notre contexte applicatif à « la détection pour suivi » qui est une technique proposé dans le contexte particulier de la robotique ou de la surveillance.

## 1. Détection de mouvement

La détection d'objets en mouvement est un problème courant et largement étudié en vision par ordinateur dans différentes applications comme la vidéosurveillance et la surveillance du trafic routier. Plus récemment, d'autres champs d'applications ont été explorés et plusieurs solutions ont été proposées. On peut citer entre autres des travaux destinés à des fins pédagogiques ou scientifiques comme la détection et la reconnaissance des animaux, d'autres pour des applications multimédia, des jeux vidéo, et des applications de reproduction des mouvements capturés sur des personnages de synthèse.

## 2. Architecture générale des systèmes de détection et de suivi de mouvement



*Figure 1.1: Schéma général de système de détection de mouvement (exemple de la vidéosurveillance [3]).*

Ce schéma montre l'architecture générale du système de détection de mouvement dans l'exemple de vidéo surveillance. Chaque application de vidéo surveillance a une exigence différente [3].

## 3. Système d'acquisition de vidéo

Principalement, ce présent projet vise l'élaboration d'un système de détection automatique et suivi simultané des multi-objets en mouvement dans une scène utilisant une caméra couleur fixe ou mobile. Ainsi, vu la particularité de ce projet, les séquences d'images couleurs doivent être acquise dans un environnement extérieur pendant de longues durées et sous différentes conditions climatiques et d'éclairage. Quant au montage physique, il se doit d'être simple et peu coûteux tout en offrant performance, robustesse et efficacité. Cette étape

concerne donc le choix de composantes standards et non spécialisées de l'équipement informatique et des capteurs à utiliser [4].



*Figure 1.2: Schéma du système d'acquisition et traitement de vidéo.*

## 4. La détection des régions en mouvement

La détection des régions en mouvement est l'étape de base de chaque système de vision par ordinateur puisqu'elle fournit un centre d'attention (région d'intérêt) et simplifie le traitement et l'analyse de la scène.

Bien souvent, l'étape de détection fait partie d'une chaîne de traitements dont elle est le premier maillon. Cette chaîne varie d'une application à une autre dépendamment des fins souhaitées. On trouve alors dans la littérature de nombreuses méthodes développées dans le cadre de leurs applications.

### 4.1 Les techniques de détection des régions en mouvement

L'étude bibliographique dans [4] a permis de distinguer trois types de méthodes : celles basées sur la différence ou double différence d'images consécutives, celles basées sur la différence d'image avec une image de référence nommé soustraction de fond et celles basées sur le flux optique (ou nommé parfois flot optique).

#### 4.1.1 Flux optique

L'une des célèbres techniques conçues pour caractériser le mouvement dans une séquence d'images est l'estimation du flux optique. Il s'agit ici d'une représentation de bas niveau qui détermine les vecteurs de vitesse en chaque point de l'image. Autrement dit, associé à chaque point physique élémentaire de l'image, un vecteur de vitesse 2D, comme étant la projection sur le plan, de la vitesse dans l'espace 3D de ce point. Le flux optique

figure (1.3) désigne le champ de vitesses apparent observé entre deux images successives d'une même scène.



*Figure 1.3 : Détection par la méthode du flux optique.*

L'estimation du flux optique n'est généralement qu'une première étape à l'extraction des objets mobiles. Cette estimation est non seulement coûteuse en temps de calcul mais aussi très sensible aux mouvements de forte amplitude. De plus, les estimés sont souvent bruités aux frontières des objets mobiles et délicats à obtenir à l'intérieur de larges régions homogènes [5].

### **4.1.2 La différence temporelle des images**

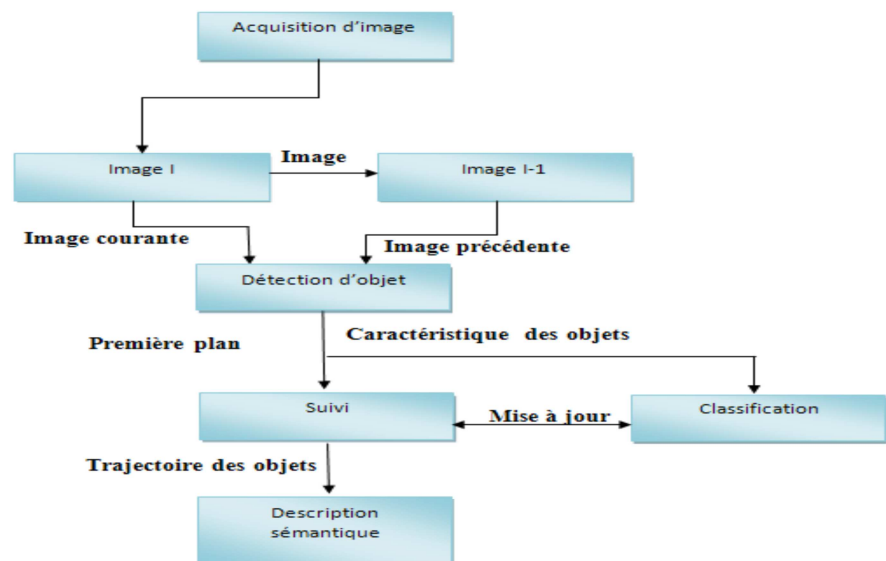
Une seconde classe de méthodes de détection de changements se base sur une différenciation temporelle, entre quelques trames consécutives. Celles-ci s'adaptent aux variations dans le temps de l'environnement. En contrepartie, elles ont tendance à négliger certaines variations liées au mouvement d'objets dans la scène, surtout si ceux-ci se déplacent lentement. Elles produisent souvent des trous dans les objets détectés. Ces techniques nécessitent donc des traitements de lissage, avec des opérateurs morphologiques et le filtrage des trous et formes trop petites. Pour ne conserver que les mouvements significatifs et éliminer les mouvements occasionnels [3].



**Figure 1.4 :** Un exemple de la différence temporelle. (a) une scène simple avec deux objets, (b) les régions rouges sont la différence entre deux images consécutives.

Voici, c'est l'idée principale de cette méthode : Soit  $I_t$  l'image à l'instant  $t$  et  $I_{t-1}$  l'image à l'instant  $t-1$ . L'objet du mouvement se compose des pixels qui satisfont l'équation suivante :

$$\text{Max} (| I_t(x, y) * c - I_{t-1}(x, y) * c |, c=(R, G, B)) \geq \text{seuil} \quad (1.1)$$



**Figure 1.5:** Schéma fonctionnel de la méthode de différence temporelle [3].

### 4.1.3 Double différence temporelle des images consécutive

La méthode, est également basée sur la différence de pixel par pixel des images consécutives dans un flux vidéo. L'opérateur proposé, effectue la différence de deux paires d'images successives au temps ( $I_{t-2}$ ,  $I_{t-1}$ ) et ( $I_{t-1}$ ,  $I_t$ ). Les deux images résultantes sont alors



# Chapitre 1 : vision par ordinateur et détection de mouvement

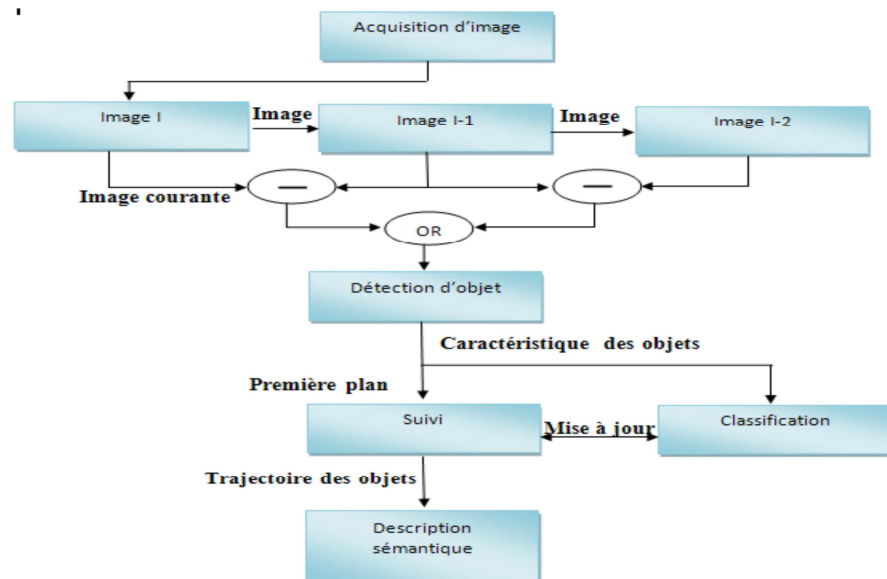
binarités et un ensemble de points appartenant à l'objet mobile est obtenu en effectuant l'intersection des masques binaires correspondants. La figure (1.6) illustre les différentes étapes d'un système de vidéo surveillance basé sur cette technique. Sa formulation est définie comme suit. Soit l'image à l'instant  $t$ , l'image à l'instant  $t-1$  et l'image à l'instant  $t-2$ . L'objet en mouvement se compose des pixels qui satisfont les équations suivantes :

$$I_1(x, y) = \max (| I_t(x, y) * c - I_{t-1}(x, y) * c |, c=(R, G, B)) \geq \text{seuil}$$

$$I_2(x, y) = \max (| I_{t-1}(x, y) * c - I_{t-2}(x, y) * c |, c=(R, G, B)) \geq \text{seuil}$$

$$I_{\text{résultat}}(x, y) = I_1(x, y) \cup I_2(x, y) \quad (1.2)$$

Par ailleurs, le défaut principal de cette méthode est son manque de précision et de robustesse aux forts mouvements pouvant être présents dans la scène. En plus, elle n'offre aucune adaptation face aux bruits intrinsèques de la caméra et des variations de luminosité au cours du temps. Ce qui nuit fortement à la performance des résultats pour les applications extérieures [4, 6].



**Figure 1.6:** Schéma fonctionnel de la méthode double différence temporelle.

Le tableau suivant illustre une comparaison des deux méthodes simple et double différence :

<i>Méthode</i>	<i>Simple Différence</i>	<i>Double Différence</i>
Principe	les occultations identifiées par les déplacements des objets en découvrant et recouvrant d'autres régions de la scène	Différence pixel par pixel d'images consécutives
Avantages	-Simple à mettre en œuvre.  -De bons résultats pour les déplacements lents et texturés.	-Pratique pour les mouvements rapides et l'extraction des régions mobiles suffisamment texturée.
Inconvénients	-Phénomène de délocalisation.  -Apparition de fantômes.	Peu performante pour des mouvements lents

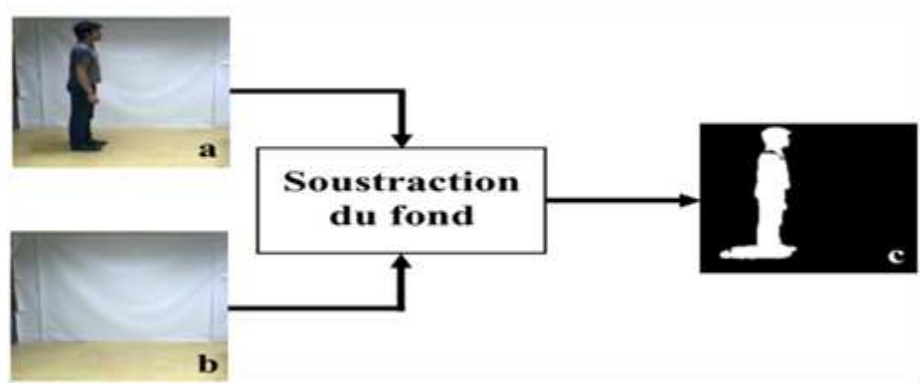
**Tableau 1.1:** Comparaison des deux méthodes Simple différence et Double différence [3].

#### 4.1.4 Soustraction de l'arrière-plan (Background subtraction)

Les méthodes basées sur la différence d'image avec une image de référence sont les plus utilisées car offrant le meilleur compromis entre efficacité et temps de calcul. Le principe consiste à utiliser un modèle pour calculer l'image de référence, à mettre à jour cette image de référence et à calculer la différence avec l'image courante pour obtenir les objets en mouvement à l'aide d'un seuil.

Grâce à sa simplicité théorique ainsi qu'à sa faible complexité algorithmique, la soustraction d'arrière-plan est facile à implémenter et permet une gestion plus flexible des variations lumineuses et des mouvements réguliers d'un ou de plusieurs objets du fond. Par contre dans la pratique, on obtient des fausses détections et des manques des informations à cause des irrégularités de la scène statique observée causées principalement par: les variations lumineuses locales ou globales, lentes ou rapides, le fond peu visible, le mouvement des objets du fond (objets de l'avant-plan ayant la même couleur que le fond).

Un modèle idéal de calcul et de mise à jour d'image de référence doit prendre en compte tous ces problèmes et doit être adapté à l'application. Pour cela, il est nécessaire de développer cette approche en se basant sur d'autres outils mathématiques permettant de prendre en compte l'imprécision et l'incertitude de la classification du pixel en arrière ou avant-plan dans le cas d'une scène quelconque [7, 8].



**Figure 1.7:** Principe de soustraction du fond, (a) image courante, (b) image du fond, (c) image après la soustraction du fond.

## 4.2 Approche proposé

Nous devons d'abord développer une méthode efficace et précise pour détecter des objets en mouvement, Un algorithme classique de détection d'objets en mouvement doit présenter les caractéristiques suivantes [9]:

- l'estimation de la partie fixe (fond).
- l'obtention d'images de différence de trames.
- L'application de la valeur de seuil.
- Avoir une estimation précise de la position et de la vitesse de l'objet.

Nous précisons que dans notre cas la méthode de soustraction de fond, était la technique proposée dans le contexte de la robotique ou de la surveillance.

### 4.2.1 Algorithme de soustraction de fond

L'algorithme utilisé pour la soustraction de l'arrière-plan par modélisation statistique comporte trois étapes importantes : l'initialisation, l'extraction du mouvement (avant-plan) et la mise à jour du modèle [10].

i. **Initialisation** : La première étape consiste à modéliser l'arrière-plan à partir des N premières images ( $N \approx 10$ ) d'une séquence vidéo. Une moyenne d'intensité est donc calculée à partir de ces images pour chaque pixel et pour chacun des canaux (R, G et B). La moyenne d'intensité d'un pixel donné se résume alors à l'équation suivante :

$$u_c(x, y) = \frac{1}{N} \sum_{i=0}^N I_{i, c}(x, y) \quad (1.3)$$

Où  $I_i$  est la  $i$ ème image d'initialisation,  $N$  la quantité d'images utilisées et  $c$  le canal sélectionné [10].

L'étape suivante consiste à **calculer un écart-type** pour chaque pixel (et pour chaque canal) afin d'être utilisé comme seuil de détection. Cette opération nécessite habituellement le stockage des  $N$  premières images. Or, une équation modifiée permet de contourner cette contrainte de façon incrémentale et ainsi réduire la consommation d'espace mémoire. Pour ce faire, deux accumulateurs sont utilisés, soient  $S(x, y)$  pour stocker la somme des intensités des pixels et  $Sc(x, y)$  pour emmagasiner la somme des carrés. Les écarts-types peuvent alors être calculés à l'aide de l'équation (1.4). Par ailleurs, il est intéressant de remarquer que  $S(x, y)$  peut être réutilisé pour le calcul de la moyenne, ce qui évite des opérations supplémentaires et superflues.

$$\sigma_{c(x,y)} = \sqrt{\left(\frac{Sc(x,y)}{N}\right) - \left(\frac{S(x,y)}{N}\right)^2} \quad (1.4)$$

ii. **Extraction de l'avant-plan** : afin d'extraire le mouvement dans une image, le modèle de l'arrière-plan doit tout d'abord lui être soustrait. Chaque pixel, dont la différence en valeur absolue dépasse la valeur  $\alpha \times \sigma$ , est ensuite classifié comme étant un pixel en mouvement. Dans l'expression précédente, la variable représente une certaine fraction de l'écart-type. En pratique, ce paramètre se situe dans l'intervalle [2.0, 4.0] et dépend du niveau d'exclusion désiré. Un masque binaire de mouvement peut alors être généré pour chaque canal à l'aide de l'équation (1.5) :

$$m_c(x, y) = \begin{cases} 1 & \text{si } |I_c(x, y) - u_c(x, y)| > \alpha \times \sigma \\ 0 & \text{autrement} \end{cases} \quad (1.5)$$

Où  $m_c(x, y)$  représente le masque de mouvement pour un canal  $c$  et  $I_c(x, y)$  l'image d'entrée à analyser [10].

L'équation (1.5) représente le calcul du masque de mouvement pour un seul canal. Pour utiliser cet algorithme avec les 3 canaux (RGB) des images utilisées, les masques individuels doivent tout d'abord être générés indépendamment et combinés par la suite à l'aide d'un opérateur OU logique. Par conséquent, si un mouvement est détecté pour un pixel

dans un seul canal, cela sera suffisant pour en modifier l'état. L'équation suivante représente cette combinaison produisant ainsi un masque de mouvement à un seul canal :

$$M(x, y) = m_r(x, y) \cup m_g(x, y) \cup m_b(x, y) \quad (1.6)$$

Une fois cette opération complétée, certaines opérations de morphologie mathématique doivent être appliquées afin d'éliminer le bruit et les fausses détections.

Pour ce faire, 2 érosions et 2 dilations sont appliquées respectivement dans cet ordre sur le masque de mouvement. Finalement, l'image d'entrée est combinée avec le masque pour produire une image à 3 canaux (avant-plan) contenant seulement les pixels représentant du mouvement. Cette opération peut se résumer à l'équation suivante :

$$F(x, y) = M(x, y) * I(x, y) \quad (1.7)$$

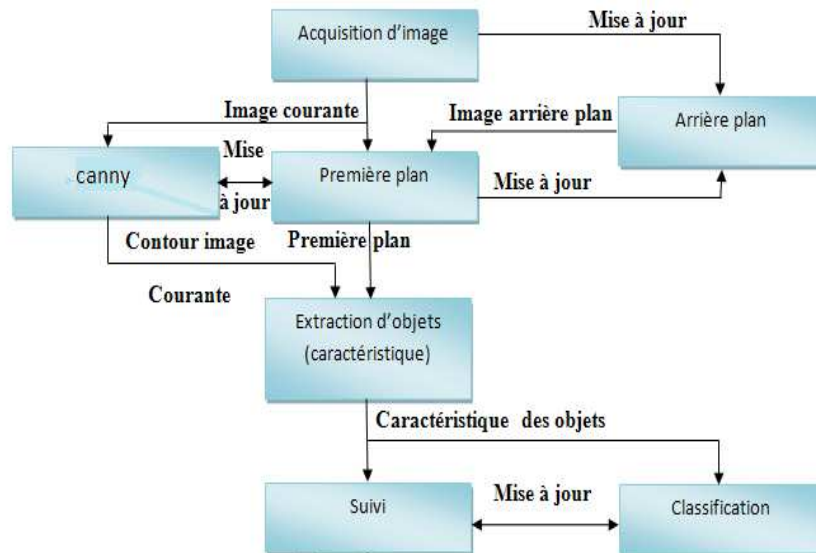
Où  $F(x, y)$  représente l'image d'avant-plan (mouvement) et  $I(x, y)$  l'image d'entrée. Les deux images sont combinées grâce à une multiplication pixel à pixel pour chacun des canaux [10].

iii. **Mise à jour du modèle** : au cours de la période d'acquisition, certaines régions de la scène peuvent subir des modifications d'éclairage, ce qui rend la mise à jour du modèle statistique de l'arrière-plan primordiale. Ainsi, un changement graduel de luminosité (p. ex. : lever du soleil) sera donc intégré au modèle et ne sera pas considéré comme du mouvement. Pour ce faire, l'extraction de l'avant-plan est réalisée avec l'image courante, ce qui génère un masque de mouvement  $M$ . Le modèle de l'arrière-plan est ensuite mis à jour à partir du complément de  $M$ , c'est-à-dire en utilisant tous les pixels qui sont étiquetés comme faisant partie de l'arrière-plan. Les changements brusques dans l'image ne sont donc pas ajoutés au modèle. L'équation (1.8) illustre ce processus de mise à jour :

$$u^c(x, y) = (1 - \eta) * u^c(x, y) + \eta I_c(x, y) * \bar{M}(x, y) \quad (1.8)$$

Où  $u^c(x, y)$  représente un pixel de l'arrière-plan moyen mis à jour et  $\eta$  le taux d'apprentissage. L'expression  $I_c(x, y) * \bar{M}(x, y)$  représente les pixels statiques de l'image courante, c'est-à-dire ceux pour lesquels aucun changement n'est associé.

Afin de ne pas modifier radicalement le modèle d'arrière-plan, seulement une fraction  $\eta$  de l'image temporaire  $I_c(x, y)$   $\hat{M}(x, y)$  est utilisée [10].



**Figure 1.8:** Schéma fonctionnel de la méthode Canny- Background Adaptatif.

En pratique, ce taux d'apprentissage peut prendre des valeurs comprises dans l'intervalle  $[0.05, 0.25]$ . Plus la valeur de ce paramètre est élevée, plus les changements s'intégreront rapidement. Cela revient alors à oublier rapidement le modèle construit lors de la phase d'initialisation. Il est conseillé d'utiliser des valeurs relativement faibles (p. ex. : 0.05).

Finalement, l'écart-type n'est pas ajusté ou mis à jour pendant l'exécution de l'algorithme (c.-à-d. : une fois l'initialisation effectuée) afin de réduire la somme de calculs nécessaire. Certaines expérimentations supplémentaires devraient cependant être réalisées pour vérifier l'utilité et l'impact de cette mise à jour sur les résultats [10].

## 4.2.2 Autres manières pour la Création d'image de fond

Le choix de la manière dépend de l'application. Nous avons proposé trois manières pour créer l'image de fond :

- Le calcul de la moyenne des  $N$  premières images.
- Nous choisissons une image de l'arrière-plan, on tenant compte du non changement de cette image pendant une durée bien déterminée.

- On prend la première image dans le flux vidéo comme image de fond.

### 4.2.3 La Combinaison des images résultantes par des images de contours

L'extraction de mouvement par soustraction d'arrière-plan demeure une technique très efficace, mais qui dépend toutefois des seuils utilisés ainsi que de la qualité des images d'entrées. En effet, le système mis en place n'offre pas suffisamment d'informations que nécessite le traitement (nombre de niveaux de gris dans les images). De plus, les conditions ambiantes lors des acquisitions peuvent aussi altérer les données. D'où la nécessité de faire la combinaison des résultats avec des images de contours.

#### 4.2.3.1 Détection des contours

Pour parvenir à cette fin, une image de contour est extraite à partir de l'image originale. Elle sera par la suite combinée avec le résultat de la segmentation par soustraction d'arrière-plan. D'un point de vue théorique, la détection des contours est l'opération qui consiste à détecter les brusques variations d'intensité dans une image. Pratiquement, elle consiste à faire passer dans l'image des opérateurs assimilables à des calculs de dérivées premières ou secondes. L'extraction des contours consiste alors à rechercher des maxima locaux, ou des lieux de passages par zéro de la dérivée seconde dans les images résultant de ces transformations [4].



**Figure 1.9:** (a) image en niveau de gris, (b) image après la détection de contour (Canny)

En pratique, le calcul de gradient est effectué par la convolution de filtres linéaires. Ces filtres peuvent estimer les dérivées premières, les dérivées secondes, etc., et se focaliser

sur des contours de directions différentes. Il existe ainsi de nombreux filtres, qui ont des fondements mathématiques différents. Cependant, quelle que soit leur justification théorique, leur objectif reste le même : mettre en évidence les pixels qui ont une valeur très différente de leurs voisins. Comme les opérateurs du gradient et du laplacien, les filtres de Prewitt, Sobel, Roberts où basées sur des critères d'optimalité comme le filtre de Canny. Signalons qu'une présence du bruit dans l'image peut rendre la tâche de la détection des contours un peu difficile.

Les différents tests effectués sur les séquences utilisées semblent montrer que l'opérateur de "Canny" est le modèle le plus approprié et donne de meilleurs résultats parce qu'il tient compte du phénomène de bruit qui se superpose à l'image. Il permet de déterminer des contours sur des images plus ou moins bruitées. Un détecteur de contour typique comprend les étapes suivantes [9]:

- suppression de bruit autant que possible.
- application d'un filtre pour améliorer la qualité de contour d'image.
- Il détermine quel contour à pixels doit être jeté comme du bruit.
- On détermine la position exacte d'un contour.

### 4.2.3.2 Détecteur de contour Canny

Le filtre de Canny est l'un des outils de traitement d'image les plus utilisés pour détecter les contours de l'image. Il comprend les étapes suivantes [9]:

#### A. *Conversion vers Niveau de Gris : (grayscale conversion)*

Une image numérique en niveaux de gris est une image dans laquelle la valeur de chaque pixel est un échantillon unique et qu'il ne transporte que l'intensité d'information. Nous trouvons cette conversion sous MATLAB avec l'instruction suivante:

**Gray = rgb2gray(a);** tels que : a= l'image qui subit cette conversion.

Après la lecture du fichier de la vidéo d'entrée, il faut extraire la couleur rouge, le vert et les intensités bleues séparément pour avoir l'histogramme facilement. Nous allons apprendre



donc à transformer une image couleur RGB en une image en niveau de gris. Pour la conversion on utilise le calcul de la luminance. Qui se fait comme suit :

$$\text{Image luminance} = \beta_r \times R + \beta_g \times G + \beta_b \times B$$

Où  $\beta_r$ ,  $\beta_g$  et  $\beta_b$  sont les trois facteurs de pondération pour les trois couleurs R, G et B. Il existe beaucoup de normes différentes pour ces facteurs.  $\beta_r=30\%$  ;  $\beta_g=59\%$  et  $\beta_b=11\%$ .



(a)



(b)

**Figure 1.10:** (a) Image originale en couleur. (b) Image en niveaux de gris.

### **B. Réduction du bruit**

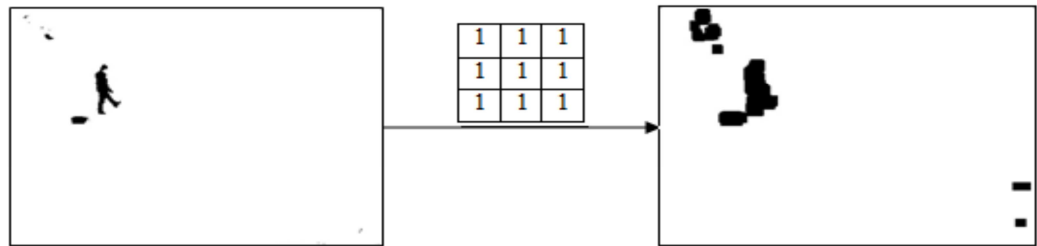
Une fois ces opérations complétées, l'image de contours est combinée par un "OU" logique avec le masque binaire obtenu par soustraction d'arrière-plan. Cette tâche permet de produire une image complète (avant-plan) contenant seulement les pixels représentant du mouvement.

Finalement, certaines opérations de morphologie mathématique doivent être appliquées afin d'éliminer le bruit et les fausses détections. Dans une opération morphologique, La valeur de chaque pixel dans l'image de sortie est basée sur une comparaison du pixel correspondant dans l'image d'entrée avec ses voisins.

**Les opérations morphologiques** les plus élémentaires sont la dilatation et l'érosion :

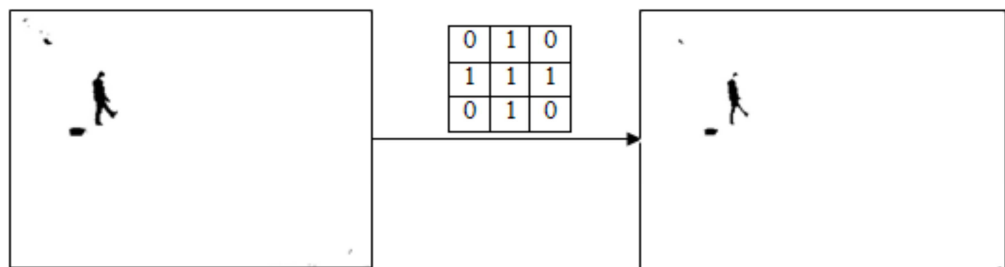
- a) Dilatation** : est une opération qui "pousse" et "épaissit" un objet dans une image binaire, Par exemple une image binaire simple A contenant un objet, l'exemple utilise un objet carrés

de 3 par 3 élément structurant. Après avoir effectué l'opération de dilatation, une image binaire est donnée par:



**Figure1.11:** Opération de dilatation.

**b) Erosion :** Erosion "rétrécit" ou "Mince" un objet dans une image binaire. Opération de l'érosion est tout à fait opposée à l'opération de dilatation. Après avoir effectué l'opération érosion sur l'image binaire:



**Figure1.12:** Opération d'érosion.

En pratique, 2 érosions et 2 dilatations doivent être appliquées respectivement sur le masque binaire de mouvement, afin d'éliminer le bruit et les fausses détections.

### **C. Calcule de gradient:**

Pour faire simple, l'opérateur calcule le gradient de l'intensité de chaque pixel. Ceci indique la direction de la plus forte variation du clair au sombre, ainsi que le taux de changement dans cette direction. On connaît alors les points de changement soudain de luminosité, correspondant probablement à des bords, ainsi que l'orientation de ces bords.

En termes mathématiques, le gradient d'une fonction de deux variables (ici l'intensité en fonction des coordonnées de l'image) est un vecteur de dimension 2 dont les coordonnées sont les dérivées selon les directions horizontale et verticale. En chaque point, le gradient

pointe dans la direction du plus fort changement d'intensité, et sa longueur représente le taux de variation dans cette direction. Le gradient dans une zone d'intensité constante est donc nul. Au niveau d'un contour, le gradient traverse le contour, des intensités les plus sombres aux intensités les plus claires [9, 4].

### 5. Localisation et classification des objets en mouvement

Consiste à catégoriser les objets dans les classes prédéfinies. Deux importantes approches coexistent: celles qui sont basées sur la forme et celles basées sur le mouvement. Les méthodes basées sur la forme utilisent l'information spatiale 2D des objets, tandis que les méthodes basées sur le mouvement (notre cas) emploient les dispositifs de suivi temporel des objets [11].

### 6. Extraction des propriétés des objets en mouvement

Avant d'entamer l'étape suivie des objets, il est nécessaire de faire une analyse des composantes connexes. C'est une procédure également appelée "capture des connexités" ou "étiquetage" des pixels. La technique consiste à regrouper les pixels voisins dans un ensemble appelé "composante connexe". Chaque ensemble est disjoint des autres et peut ensuite être aisément isolé. La connexité "4" est distinguée de la connexité "8" suivant que le critère de voisinage comprend les 4 ou les 8 voisins d'un pixel [4].

Une fois cette analyse achevée, la prochaine étape consiste à chercher et sélectionner les informations pertinentes qui se trouvent noyées dans la masse de l'information brute acquise. Cette tâche peut être réalisée à l'aide de l'extraction des "primitives" ou «caractéristiques». Ces dernières représentent les informations élémentaires aux applications de reconnaissance et du suivi du mouvement. Il y a 4 propriétés que l'on s'intéresse [4]:

➤ La taille de l'objet : On conservera la position haut-gauche et bas-droite du rectangle qui enveloppe l'objet. Parce que l'on n'a pas besoin des propriétés de l'objet que l'on s'intéresse pas, dans la partie 3.2, on a extrait seulement cette propriété pour déterminer la région de l'objet. Si les objets mouvants sont dans la région de contrôle, on continue extraire les autres propriétés de l'objet.

➤ L'histogramme de l'objet : Pour le calcul simple, on va réduire cet histogramme de 256 à M couleurs (environ 16 ou 24 couleurs). Pour diminuer le temps de calcul, on n'extrait

que cette propriété quand l'objet apparaît la première fois, ou dans le cas, ou le changement de la taille de l'objet est grand.

➤ Le centre de l'objet : Soit  $C_m(x_c, y_c)$  le centre de l'objet  $m$ ,  $N$  le nombre de pixel de l'objet  $m$  :  $X_C = \sum_{i=0}^N X_i/N$  ;  $Y_C = \sum_{i=0}^N Y_i/N$ .

➤ La vitesse : qui se compose  $dx$  et  $dy$ , les distances (sur l'axe  $X$  et l'axe  $Y$ ) entre le centre de l'objet à l'instant  $t$  et le centre de l'objet à l'instant  $t-1$ . Si l'objet apparaît la première fois,  $dx$  et  $dy$  sont égaux zéro.

### 7. Le suivi du mouvement : (motion Tracking)

L'objectif du suivi est de localiser comment les objets de premier plan se déplacent au fil du temps en reliant les instances du même objet dans des frames successives. Cela est essentiel pour plusieurs applications telles que la sécurité et la surveillance, la compression de vidéo, traitement médical et de chambres élégantes, le trafic la gestion et les jeux informatiques interactifs.

Cette étape permet non seulement l'identification temporelle des objets détectés mais pourraient produire également des informations telles que la trajectoire, la vitesse et la direction des objets à suivre. Et c'est ce que nous allons aborder dans le chapitre suivant.

### *Conclusion*

Dans ce chapitre, différentes techniques de détection de mouvement ont été présentées et analysées. Un choix justifié de la méthode de soustraction de l'arrière-plan a été fait pour la réalisation de cette partie du projet. L'algorithme proposé est une amélioration de la technique de base, notamment par rapport à l'utilisation des seuils et des gains automatiques permettant l'adaptation dynamique du système avec les conditions non contrôlées de l'environnement extérieur.

L'utilisation complémentaire des images de contours dans l'algorithme a été un atout majeur pour remédier au désagrément causé par la faible qualité des images utilisées. Pour une meilleure performance, d'autres optimisations ont été proposées pour accélérer les traitements.

Finalement, la solution proposée procure de très bons résultats dans des applications de vidéosurveillance grâce à une invariance accrue aux ombres et aux conditions d'éclairage et au climat extérieur.



# Chapitre II :

## Suivi d'objets et filtre de Kalman

### ***Introduction***

Le suivi d'objets dans des séquences d'images vidéo est l'un des sujets les plus importants dans de nombreux champs de la vision par ordinateur, tels que la vidéosurveillance, l'intelligence artificielle, la détection des obstacles dans la navigation des robots, aussi dans les applications médicales et biologiques...Etc. Ces dernières années, un certain nombre de succès dans les systèmes de suivi d'un seul objet sont apparus, mais en présence de plusieurs objets, le problème est celui du suivi de multiples objets où les objectifs et les observations doivent être assortis de trame en trame dans une séquence vidéo.

Dans ce chapitre, nous nous intéressons plus particulièrement à la phase de suivi. La problématique que nous proposons consiste un suivi simultané de multiples objets en mouvement, de forme et de texture quelconques, soumis à un mouvement quelconque dans des séquences d'images couleurs.

Un bref aperçu des algorithmes de suivi et leurs applications sera donc présenté. Nous présenterons ensuite l'algorithme de suivi avec filtre kalman pour établir le modèle de l'objet en mouvement, en utilisant les informations de l'objet en cours de mouvement pour prédire leur position, de sorte que nous pouvons réduire le temps de la recherche de l'objet pour atteindre rapidement le suivie.



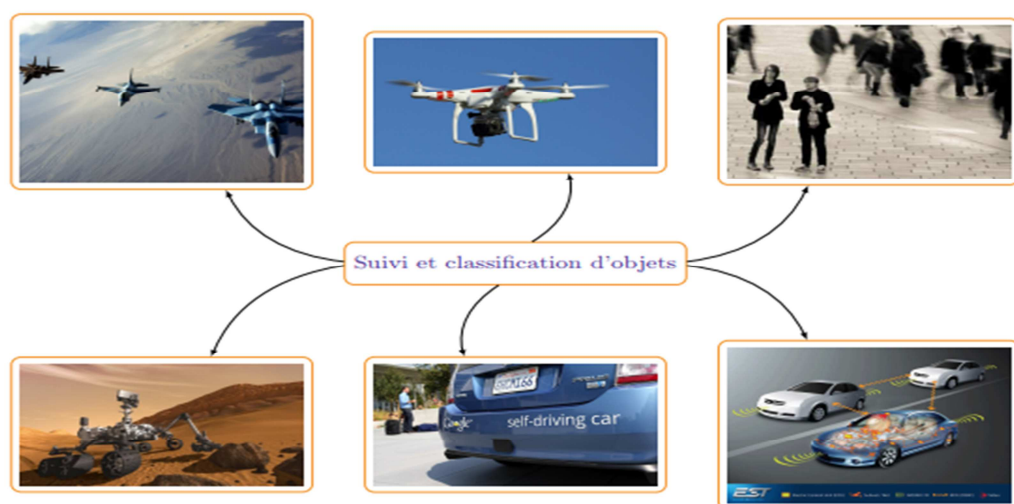
## 1. Suivi de mouvement

Comme décrit dans le chapitre précédent, l'étape de segmentation de mouvement sert à identifier les régions qui correspondent aux objets d'avant-plan dans une image donnée. Typiquement, le résultat est une image binaire dont chaque pixel est marqué en tant que fond ou avant-plan. Des régions sont alors groupées ensemble et étiquetées en utilisant les composants connexes. Afin de simplifier la tâche au processus de suivi, les régions de petite taille sont souvent éliminées au moyen d'un filtrage morphologique [4].

Durant l'étape d'appariement, les données de sorties, sont combinées pour fournir une correspondance temporelle à travers les trames de la séquence, et donc identifier la progression des objets à travers la scène. En réalité, l'étape d'appariement joue un double rôle [4]:

- le premier consiste à déterminer le moment d'entrée d'un objet en scène pour initialiser son suivi, et également déterminer moment de sa sortie du champ de vision.
- La seconde doit établir la correspondance entre les régions nouvellement détectées dans l'étape de segmentation avec les objets actuellement suivis, pour être capables de faire le suivie simultanée des multiples objets en mouvement.

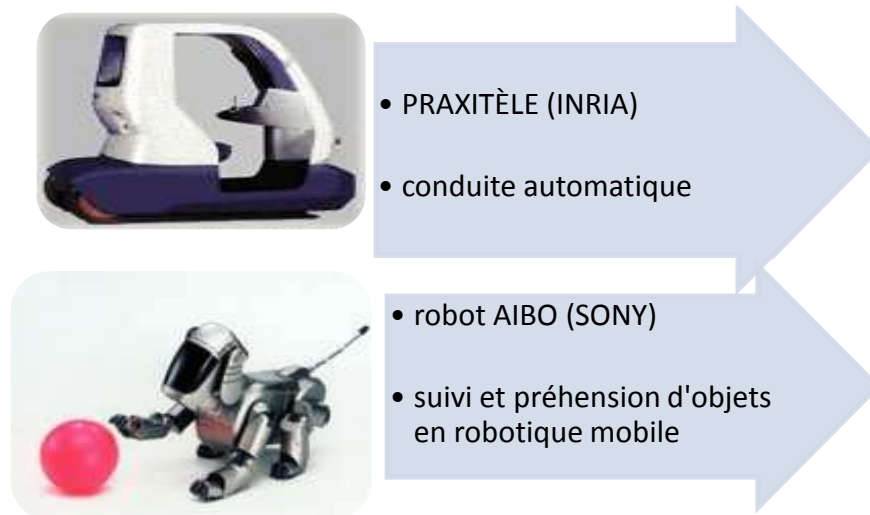
## 2. Quelques applications du suivi de mouvement



*Figure 2.1 : Quelques applications du suivi et de la classification d'objets.*

L'intérêt donné au suivi de mouvement dans les systèmes intelligents de surveillance s'est beaucoup développé ces dernières années, probablement grâce aux succès des grands efforts de recherche et aux améliorations phénoménales du matériel utilisé. Les algorithmes ainsi développés peuvent avoir une large gamme de domaine d'exploitation.

### A. La robotique



*Figure 2.2: exemple des applications robotique utilise le suivie [12].*

### B. Surveillance du trafic routier

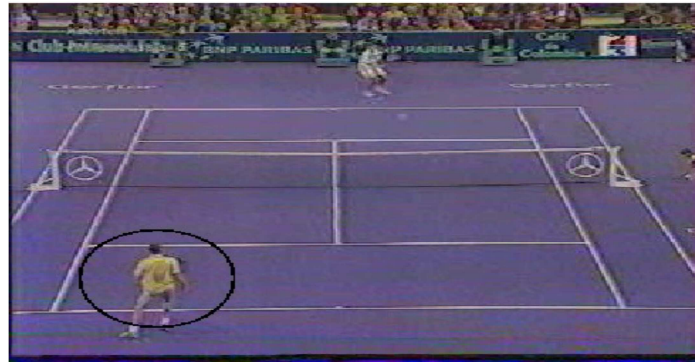
On trouve également un intérêt particulier au suivi visuel dans le domaine de l'aide à la conduite et de la surveillance du trafic routier. Une fois un tel système embarqué chez des patrouilleurs, il est capable de suivre et reconnaître les plaques minéralogiques des véhicules dans le but d'identifier ou repérer ceux qui font l'objet d'un avis de recherche enregistrés dans le serveur central de la police.



*Figure 2.3: exemple de Surveillance du trafic routier [12]*

### *C. MultiMedia*

Exemple reconnaissance d'action dans un match de tennis pour l'indexation vidéo (INRIA).



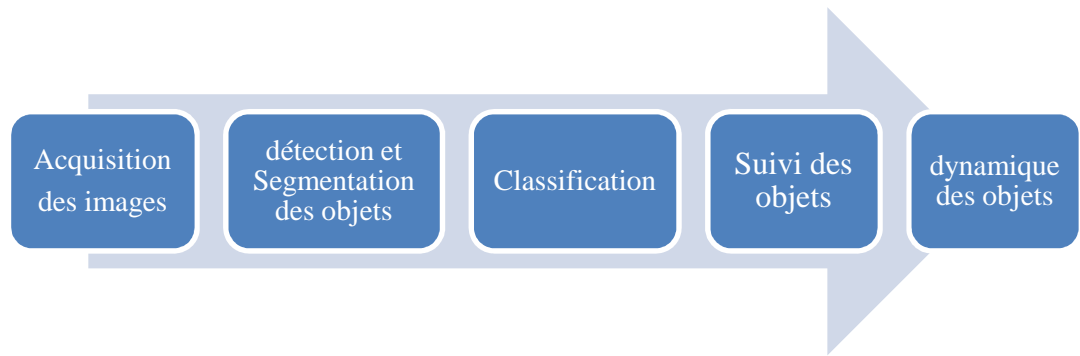
*Figure 2.4: reconnaissance d'action dans le domaine sportif.*

### **3. Notre Objectif**

Nous présenterons plus loin dans ce chapitre la méthodologie empruntée pour la réalisation d'un module de suivi spatio-temporel simultané de personnes et de véhicules dans un environnement de vidéosurveillance par vision d'une seule caméra couleur.

### **4. Structure générale des algorithmes de suivi**

On a présenté dans la section précédente un aperçu global sur les utilisations du suivi de mouvement dans divers domaines d'intérêt. Ceci nous indique clairement l'existence d'une variété de techniques qui opèrent différemment selon leurs contextes et objectifs. En revanche, puisque le thème central est commun entre ces méthodes, une architecture globale d'un traitement typique peut être schématisée dans la figure (2.5):



*Figure 2.5: Schéma d'un système typique de surveillance.tiré de [4].*

La conception d'un système de suivi est fortement sensible au type et à la qualité des données traitées. Par exemple, l'image d'entrée peut être couleur ou en tons de gris et avoir différents niveaux de résolution. Le capteur utilisé peut être statique (caméra stationnaire) ou mobile (caméra embarquée), sans oublier de mentionner l'influence des facteurs intérieurs et extérieurs reliés à l'acquisition des données (bruit du capteur et les conditions d'éclairage). La flexibilité et la complexité algorithmique sont aussi contraintes par le matériel de calcul utilisé, et c'est souvent une question d'un bon compromis entre la complexité et la rapidité [4].

### 5. Méthodologie des Algorithmes

Regardons maintenant l'aspect purement algorithmique. Après avoir calculé une segmentation spatiale et/ou temporelle initiale, le suivi de mouvement d'un instant  $t$  vers un instant  $t+1$  peut généralement être composé au minimum de trois étapes [4]:

**a. Prédiction** du mouvement qui relie la segmentation de l'image à l'instant  $t$  à celle de l'image à l'instant  $t+1$ .

**b. Ajustement** de la segmentation prédite, où on ajuste les frontières des régions de la segmentation sur les frontières réelles.

**c. Estimation** du mouvement des régions suivies, où on calcule le mouvement effectif des régions suivies entre les images aux instants  $t$  et  $t+1$ .

### 6. Problématique

Dans les cas pratiques, des difficultés peuvent surgir quand un seul objet se sépare en deux ou plusieurs régions (ou "blobs") à cause d'une faible segmentation. De même, pour le cas de deux objets qui se déplacent étroitement ensemble. Ils peuvent fusionner pour former un simple blob. L'algorithme de suivi devrait alors pouvoir reconnaître de tels cas et réagir convenablement, avec la capacité d'apparier n'importe quel nombre d'objets avec n'importe quel nombre de blobs.

En plus, un bon algorithme de suivi devrait pouvoir aussi traiter des occultations sous ses diverses formes. Ceci pourrait inclure des occultations entre les objets de l'avant-plan, ou entre ces derniers avec ceux de l'arrière-plan. Le processus peut alors être extrêmement complexe, et cela accentue les incertitudes lors de la mesure des positions des objets.

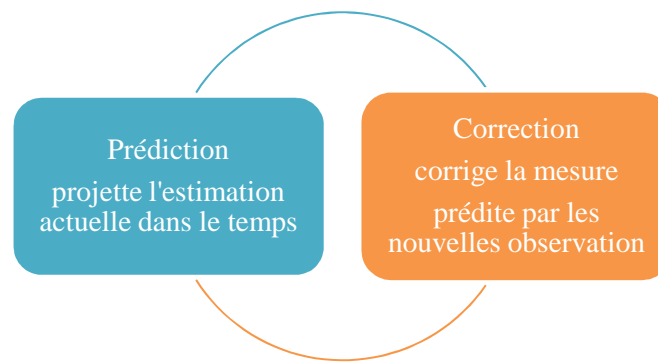
### 7. La solution proposée

Pour faire face à ces problèmes, beaucoup de solutions proposées dans la littérature modélisent ces imprécisions sous forme d'un bruit, et emploient un certain filtrage linéaire, habituellement sous forme d'un filtre de "Kalman".

### 8. Le filtre de Kalman

Le filtre de Kalman est un algorithme récursif optimal, qui suppose que les bruits de système et de mesure sont blancs et gaussiens [13]. Il est optimal parce qu'il utilise toutes les données disponibles, indépendamment de la précision, pour fournir l'estimée de la variable désirée avec une minimisation statistique de l'erreur. À la différence des autres filtres, le filtre de Kalman opère d'une manière récursive où les données précédentes n'ont pas besoin d'être stockées et retraitées à chaque étape, ce qui allège relativement la complexité algorithmique.

À chaque intervalle, le filtre prévoit d'abord l'état actuel en utilisant des données précédentes et corrige ensuite cette évaluation en utilisant les nouvelles données dans une boucle de rétroaction récursive à deux étapes: prédiction et correction. Le cycle de Kalman est illustré dans la figure (2.6).



**Figure 2.6:** Boucle de rétroaction récursive du filtre de Kalman. [4]

Par la prédiction, le filtre de Kalman effectue une projection temporelle des variables d'un moment  $t$  vers un autre moment  $t + 1$  par une estimation de l'état prédit avec la valeur de covariance de son erreur. Les résultats sont ensuite améliorés lors de la correction où un nouvel état et sa covariance d'erreur sont évalués grâce aux nouvelles mesures prises à l'instant  $(t + 1)$  [4].

### 8.1 Paramètres de filtre de Kalman

Mathématiquement, le filtre de Kalman est un algorithme d'estimation statistique développé par Rudolf Emil Kalman à la fin des années 50. Il est utilisé pour prédire et corriger l'état courant d'un système dynamique linéaire dont l'état est caractérisé par un vecteur aléatoire, dont on connaît à l'instant  $t$ , la valeur moyenne est une matrice de covariance avec un modèle de bruit additionnel. La prédiction utilise des mesures linéaires perturbées aussi par du bruit. Dans le temps, les perturbations localisées sur les mesures sont représentées par du bruit gaussien avec une matrice de covariance connue à chaque temps de mesure [15].

### 8.2 Principe et fonctionnement de filtre

Le fonctionnement du filtre de Kalman peut se diviser en deux étapes [4]:

- i. **Une première étape** de prédiction de l'estimation selon le modèle du système. Pour ce faire, le filtre de Kalman reprend l'estimation précédente des paramètres et de l'erreur et

prédit les nouveaux paramètres et la nouvelle erreur en fonction de la modélisation du système.

- ii. **La seconde étape** va faire la mise à jour de cette prédiction grâce aux nouvelles mesures. Ces mesures (par définition bruitées) vont permettre d'obtenir une estimation des paramètres et de l'erreur à partir de la prédiction faite. Si jamais le modèle comporte des erreurs, cette étape de mise à jour permettra de les rectifier.

Dans la suite nous présenterons les principales équations du filtre du Kalman ainsi que le modèle d'état que nous utilisons. Dans les équations suivantes, nous utilisons le sous-script  $k$  pour indiquer la valeur d'une donnée (état, mesure. . .) au temps discret  $t_k$ .

D'abord, nous définissons la fonction d'évolution du vecteur d'état  $x$  du temps  $t_{k-1}$  vers le temps  $t_k$  comme un système linéaire gaussien :

### 1) Équation de processus :

$$X_k = AX_{k-1} + G_k W_{k-1} \quad (\text{Équation de processus}) \quad (1)$$

Où  $X_k$  dépend directement de la dimension  $D$  des données, donc de façon générale  $X_k \in R^{D \times 1}$ . La matrice  $A$  représente le modèle de mouvement de dimension  $R^{D \times D}$ .  $G_k$  Est une matrice connue de dimensions  $R^{D \times D_v}$ , parfois utilisée pour regrouper des coefficients de pondération appliqués sur le processus de bruit de dynamique gaussien  $W_k$  de moyenne zéro, décrit par :

$$W_k \sim N(0, Q)$$

Avec matrice de covariance  $G_k Q G'_k$  et  $Q \in R^{D_v \times D_v}$ . Si la matrice  $G_k$  n'apparaît pas comme coefficient dans l'équation d'état, la matrice de covariance de  $W_k$  est simplement  $Q$  [14].

### 2) Equation de mesure :

Ensuite, les mesures  $Z$  au temps  $t_k$ , sont directement obtenues à partir de l'état donné par 1, mais elles sont perturbées par du bruit  $V_k$  :

$$Z_k = HX_k + V_k \quad (\text{équation de mesure}) \quad (2)$$

Où  $H$  est la matrice de transformation de l'état vers l'espace des mesures, de dimension  $R^{Dz \times Dx}$ . Cette matrice est connue, mais peut-être variable dans le temps. Le vecteur  $V_k$  est un bruit gaussien de dimension  $R^{Dz \times 1}$  donné par :

$$V_k \sim N(0, R)$$

Avec la covariance  $R$  de dimension  $R^{Dz \times Dz}$ . Les deux processus de bruit gaussien (pour la dynamique et pour les mesures) ainsi que l'état initial du système, sont supposés mutuellement indépendants, donc les termes de covariance croisée seront nuls, ce qui facilite ainsi l'estimation de l'état [14].

Jusqu'ici nous avons présenté les équations du vecteur d'état  $X$  et du vecteur de mesures  $Z$  avec leurs respectives fonctions de bruit additionnelles. Le modèle de dynamique et le modèle d'observation sont exploités pour la prédiction et la correction de l'état du système, mises en œuvre en utilisant les équations du filtre de Kalman.

### 3) *Prédiction du vecteur d'état et des mesures :*

Soit  $\hat{X}_k^-$  le vecteur d'état estimé et  $P_k^-$  sa matrice de covariance correspondante au temps  $t_{k-1}$ , dont le deuxième argument  $k-1$  indique le caractère conditionnel aux informations des mesures  $Z_{k-1}$  obtenues à l'instant  $t-1$ .

La prédiction de l'état et sa matrice de covariance au temps  $t_k$  sont données par le modèle dynamique :

$$\hat{X}_k^- = A\hat{X}_k + W_k \quad (3)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (4)$$

Où la matrice  $P$  est de dimension  $R^{Dx \times Dz}$  et le deuxième terme dans l'équation (4) correspond à la covariance du vecteur de perturbation de l'état. Ensuite, la prédiction des mesures  $\hat{Z}_k$  faites à l'instant  $t_k$  depuis cet état prédit [14].



## 4) Correction de l'état :

Pour la mise à jour du vecteur d'état estimé  $\hat{X}_k$  et sa covariance  $P$ , il est nécessaire d'obtenir le gain du filtre  $K_k$  qui dépend directement des matrices de covariance obtenues pour l'état prédit et pour l'innovation [15].

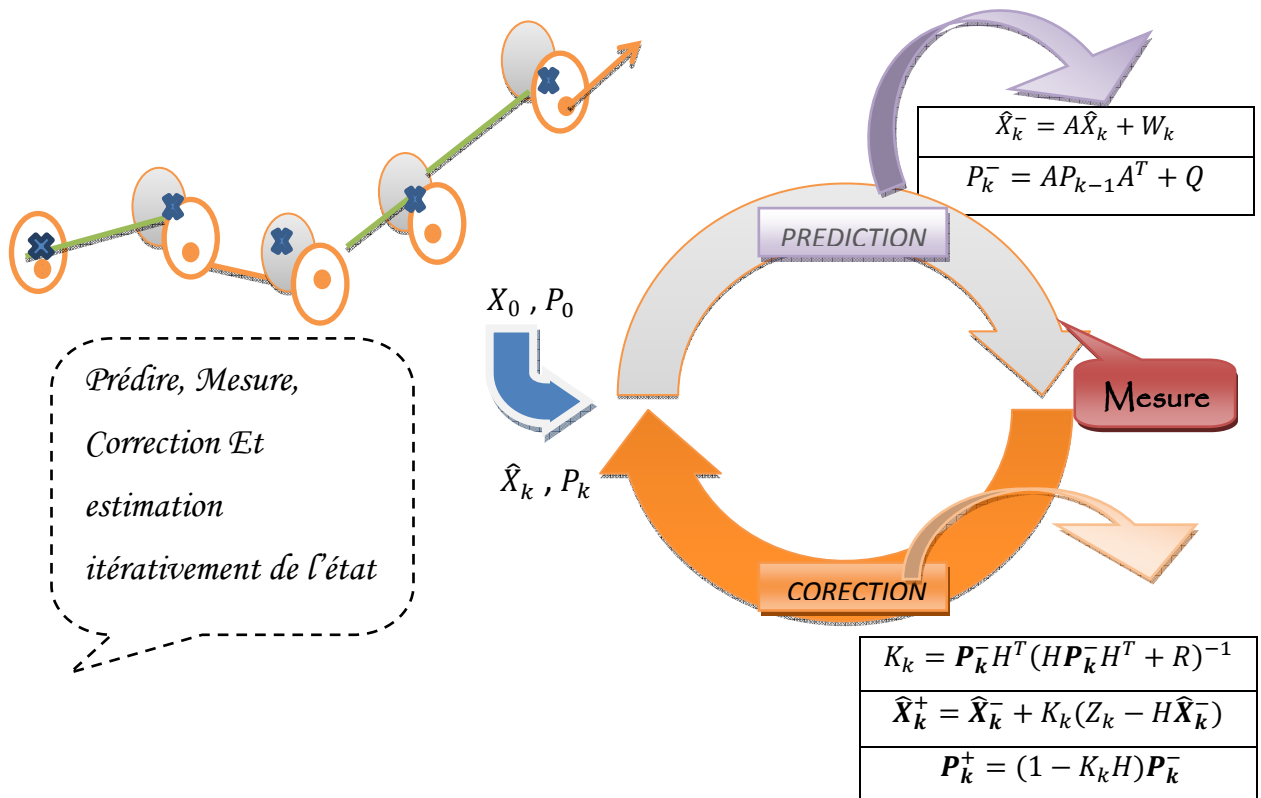
Ainsi, les équations (3) et (4) régissent la correction de l'état a posteriori, tenant compte de l'observation faite à l'instant courant [15]:

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (\text{Gain du filtre}) \quad (5)$$

$$\hat{X}_k^+ = \hat{X}_k^- + K_k (Z_k - H \hat{X}_k^-) \quad (\text{Vecteur d'état estimé}) \quad (6)$$

$$P_k^+ = (1 - K_k H) P_k^- \quad (\text{Covariance de l'état}) \quad (7)$$

On peut le résumer dans le schéma suivant :



**Figure 2.7:** Cycle d'estimation et prédiction par filtre de Kalman.

### 8.3 Cas des systèmes non linéaires

Enfin, cette forme du filtre de Kalman peut seulement modéliser les systèmes linéaires. Bien qu'il existe une version étendue du filtre capable de traiter les systèmes non linéaires, souvent seulement la forme de base généralisée est utilisée pour la modélisation des objets suivis, car elle est toujours considérée comme suffisante [16].

Tout d'abord, les équations d'état et les équations liant l'état précédent à l'instant suivant qui étaient linéaires dans le cas du filtre de Kalman classique sont maintenant non linéaire. Il est donc impossible de l'écrire sous forme matricielle. On remplace donc les équations [17] :

$$\begin{cases} \hat{X}_k^+ = A \cdot \hat{X}_k \\ Y = H \cdot X + B \end{cases} \text{ Par } \begin{cases} \hat{X}_k^+ = f(\hat{X}_k) \\ Y = h(X, B) \end{cases}$$

On est donc obligé d'appliquer ces équations non linéaires pour le calcul de la prédiction et la mise à jour du vecteur d'état (correction). Les équations de Kalman deviennent donc :

La phase de prédiction :

$$\hat{X}_k^+ = f(\hat{X}_k) \tag{a.1}$$

$$P_k^+ = A \cdot P_k \cdot A^T + Q \tag{a.2}$$

La phase de mise à jour (correction) :

$$K_{k+1} = P_k^+ H_{k+1}^T (R_{k+1} + H_{k+1} P_k^+ H_{k+1}^T)^{-1} \tag{a.3}$$

$$P_{k+1} = (1 - K_{k+1} H_{k+1}) \cdot P_k^+ \tag{a.4}$$

$$\hat{X}_{k+1}^+ = \hat{X}_k^+ + K_{k+1} (y_{k+1} - h(\hat{X}_k^+, 0)) \tag{a.5}$$

On remarque que pour mettre à jour le vecteur d'état, on utilise l'équation d'état non linéaire avec un bruit nul (second paramètre de la fonction h à 0).

Il reste un problème en ce qui concerne le calcul de la covariance de l'erreur et du gain de Kalman.

En effet, on utilise toujours les matrices d'observation et de transition  $H$  et  $A$ . Pour pouvoir utiliser ces formules, il faut donc que l'on linéarise localement les fonctions  $h$  et  $f$ . On obtient donc les matrices d'observation et de transition en prenant les matrices des dérivées partielles des équations non linéaires.

$$H_k = \left. \frac{\partial h}{\partial X} \right|_{\hat{X}_k^+}$$

$$A_k = \left. \frac{\partial f}{\partial X} \right|_{\hat{X}_{k-1}}$$

Avec ces Jacobéennes, il est donc possible d'appliquer le filtre de Kalman tel que défini ci-dessus. Il suffit de recalculer les matrices aux dérivées partielles à chaque nouvel échantillon à traiter et d'utiliser ces matrices dans les équations [16].

Par contre, on se rend bien compte que l'on linéarise localement les équations afin d'appliquer le filtre de Kalman. Cette linéarisation est locale, ce qui entraîne donc une convergence locale. Du filtre de Kalman étendu. Ce filtre ne garantit donc pas une convergence globale (à l'inverse du filtre de Kalman classique). La stabilité d'un KF est donc plus difficile à garantir et dépend souvent de sa bonne initialisation.

Dans la suite, nous détaillons les valeurs utilisées lors de l'exécution du filtre de Kalman, pour faire le suivi des multi-objets dynamiques le long d'une séquence d'images.

### 9. Filtre de Kalman pour le suivi des multiples objets

L'algorithme de suivi développé pour la réalisation de cette partie du projet est du type «suivre pour suivre» est basé sur le suivi spatio-temporel des coordonnées de centres de gravité des objets par des boîtes englobantes. Les objectifs visés pour cette étape sont les suivants [4]:

- ✓ Déterminer le moment d'entrée d'un nouvel objet dans le champ de vision du système, et initialiser le modèle de mouvement pour le suivi.

- ✓ Déterminer la correspondance entre les régions d'avant-plan détectées par soustraction de l'arrière-plan et les objets actuellement suivis.
- ✓ Établir la trajectoire complète de chaque objet durant toute sa présence en scène.

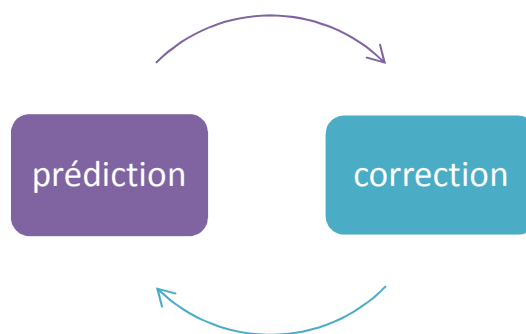
Il est donc important à ce stade, de concevoir un algorithme adéquat permettant un compromis entre la performance et la rapidité des résultats.

### 9.1 Le suivi spatio-temporel des centres de masse

Comme indiqué dans cette section, cette partie du système se base sur le suivi spatio-temporel des centres de masse. Le choix de cette coordonnée médiane est justifié par sa robustesse à l'évaluation des positions des objets, et parce qu'elle n'est pas affectée par les grands mouvements des extrémités [4].

Une forme classique d'un algorithme de suivi a été utilisée, elle repose sur deux étapes distinctes (figure 2.8):

- Une phase de prédiction, durant laquelle une ou plusieurs hypothèses sur la position de l'objet dans l'image ou sur sa localisation par rapport à la caméra sont formulées.
- Une phase de correction ou d'exploration d'un voisinage encadrant la prédiction, durant laquelle la position de l'objet est optimisée.



**Figure 2.8:** Boucle de prédiction et de correction.

### 9.2 Proposition modèle d'estimation

Filtre Kalman utilisé pour le suivi est définie par termes de ses états, modèle de mouvement, et les équations matricielles de mesure.  $x_k$  Est un vecteur d'état de huit dimensions du système, qui peut être exprimée comme suite:

$$x_k = [x_{0,k}, y_{0,k}, v_{x,k}, v_{y,k}]^T$$

Où  $x_{0,k}, y_{0,k}$  représentent les cordonnées de centre de gravité horizontal et vertical,  $l_k, h_k$  représentent une demi-largeur et demi-hauteur de la fenêtre de suivi (rectangle),  $v_{x,k}, v_{y,k}$ , représente la vitesse respectivement [14]. Le vecteur de mesure du système adoptif est le suivant :

$$z_k = [x_{0,k}, y_{0,k}]^T$$

Dans ce qui suit, A est la matrice de transition et H matrice de mesure de notre système de suivi avec le processus gaussien  $W_k$  et la mesure  $V_k$  [14]. Ces valeurs de bruit sont entièrement dépendants du système qui est suivi et ajustés de manière empirique.

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La matrice d'observation H peut être décrite comme suit:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Après l'équation d'état et l'équation de mesure de modèle de mouvement sont définis, dans la trame suivante, le filtre de Kalman peut être utilisée pour estimer l'emplacement et la taille de l'objet dans un petit gamme, et d'acquérir des trajectoires d'objets en mouvement.

### 9.3 Algorithme proposée

**i. Initialisation du suivi [4]:** Durant l'initialisation, toutes les régions de l'avant-plan détectées sont considérées commettant des blobs. Ces derniers sont suivis séparément pendant plusieurs trames (notre système emploie 5 images consécutives) avant d'être pris en charge par le module suivi. Ceci nous permet de :

- Déceler les fausses détections issues d'une segmentation de faible qualité, ou d'autres causes liées à l'utilisation d'un capteur.
- Initialisation des vecteurs de caractéristiques, où les vitesses de déplacement sont mesurées afin de permettre l'enclenchement du processus de prédiction.

**ii. Phase de prédiction :** La prédiction des paramètres de mouvement correspond à la première étape de l'algorithme de suivi. Son rôle est d'essayer de fournir une estimation de la prochaine évolution spatiale de la région pendant son mouvement. Autrement dit, elle effectue une projection temporelle des coordonnées des centres de masse  $(X_t, Y_t)$  d'un moment temporel  $t$  vers un état  $(X_{t+1}, Y_{t+1})$  du moment temporel  $t + 1$ . Pour cette fonction, notre modèle de mouvement procède par une extrapolation linéaire utilisant les couples de vitesses  $(V_{x_t}, V_{y_t})$ , comme indiqué par l'équation suivante [4]:

$$Y_{t+1} = Y_t + V_{y_t}$$

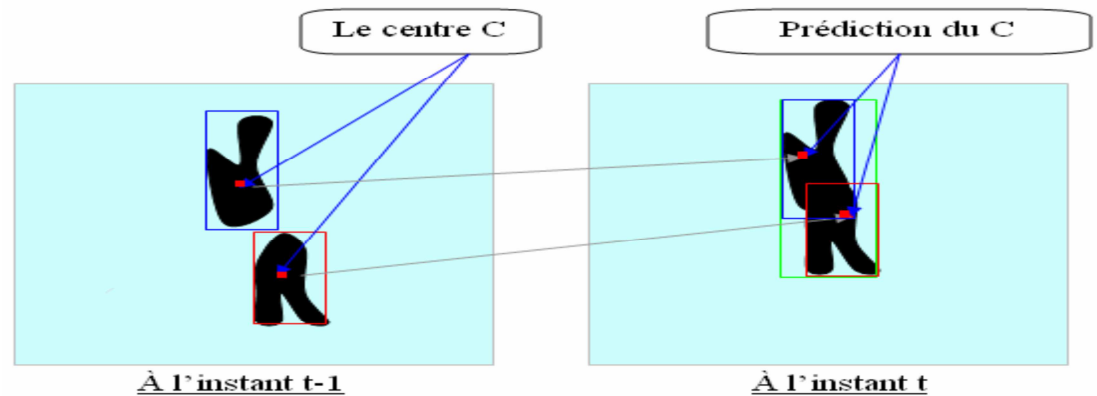
$$X_{t+1} = X_t + V_{x_t}$$

Où  $V_{x_t}$  et  $V_{y_t}$  sont les valeurs respectives des vitesses de déplacement en X et en Y des centres de masse estimées avec la trame précédente  $t - 1$  par l'équation suivante [4]:

$$V_{x_t} = X_t - X_{t-1}$$

$$V_{y_t} = Y_t - Y_{t-1}$$

**iii. Phase de correction :** Une fois la prédiction réalisée, l'algorithme emploie la nouvelle mesure pour effectuer une mise en correspondance basée sur le recouvrement des boîtes englobantes mesurées autour de la zone de prédiction (figure 2.9):



**Figure 2.9 :** Illustration de la détection de l'intersection [18].

Le critère de recouvrement représente le pourcentage de l'intersection entre la boîte englobante mesurée et celle prédite pour le même instant.

Sa valeur minimale dépend de la fréquence d'acquisition des images ainsi que la vitesse des différents mouvements présents dans la scène. Pour ce projet, le seuil minimal de recouvrement a été fixé à 75%. Ce qui signifie qu'avec une valeur supérieure à ce seuil, la candidature de la région (ou blob) correspondante est retenue par l'algorithme [18].

### 10. Les avantages et points forts de filtre

La force de ce filtre est sa capacité de prédiction des paramètres et de rectification des erreurs, non seulement des capteurs mais aussi du modèle lui-même.

En effet, pour appliquer un filtre de Kalman, il faut avant tout modéliser le système pour lequel on veut estimer les paramètres, de manière linéaire. (Des variantes du filtre de Kalman existent pour la prise en compte du modèle non linéaire).

Dans une méthode d'estimation classique (par exemple, la méthode des moindres carrés), une simple erreur dans la modélisation du système entraîne inévitablement une erreur au niveau de l'estimation. La force du filtre de Kalman est d'intégrer un terme d'imprécision sur le modèle lui-même, ce qui lui permet de donner des estimations correctes malgré les erreurs de modélisation (pour peu que les erreurs restent raisonnables).

Un autre point fort du filtre de Kalman (mais que l'on retrouve aussi dans la méthode des moindres carrés par exemple) est sa capacité à déterminer l'erreur moyenne de son estimation.

En effet, l'outil Kalman fournit un vecteur contenant les paramètres estimés, mais aussi une matrice de covariance de l'erreur. Cette matrice nous renseigne donc sur la précision de l'estimation, ce qui peut être utile dans de nombreuses applications. Un autre atout du filtre de Kalman est que la convergence de cette erreur est garantie.

### ***Conclusion***

D'après ce qu'on a vu précédemment nous pouvons conclure, que le filtre de Kalman est un ensemble d'équations permettant d'obtenir le filtrage du point de vue de l'erreur faite sur l'estimation de l'état du système. Le filtre de Kalman est un outil mathématique puissant et particulièrement utile dans le monde la localisation et navigation en robotique mobile, il permet en effet d'estimer l'état du robot mobile, en fonction de son état précédent, des commandes appliquées et de mesures bruitées.

Le chapitre suivant sera consacré aux résultats et à l'application de filtre de Kalman dans le domaine de poursuite des objets mobiles.



# Chapitre III :

## Implémentation, résultats et discussion

### ***Introduction***

Dans ce chapitre, les résultats des méthodes de détection et de suivi des objets multiples en mouvement seront présentés.

Nous décrivons aussi toutes les mesures d'évaluation et les variations des paramètres pour chaque étape des méthodes proposées, afin d'atteindre les paramètres optimale pour notre système.

#### **1. Software et Langage de programmation**

##### **- MATLAB**

C'est un logiciel utilisé pour calculer des fonctions numériques, opère sur des matrices et des matrices entières.

Tous les travaux liés au traitement de l'image sont d'abord effectuée en utilisant MATLAB. Le traitement des images et de vidéo est toute forme de traitement du signal pour lequel l'entrée est une image numérique. La sortie de traitement d'image peut être soit une image ou des paramètres liés à l'image. Contrairement à d'autres logiciels MATLAB traite pixel par de pixel le calcul sur l'image d'entrée selon l'algorithme.

##### **- SIMULINK**

Est un utilitaire de MATLAB utilisé pour représenter un processus sous la forme de blocs. Un schéma de principe est une représentation du processus consistant en entrées, le système et ses sorties. Plutôt que d'écrire des codes comme cela se fait pour MATLAB, les blocs sont simplement placés et connectés à rendre le système complet. En bref, il fournit une interface utilisateur graphique (GUI) pour la construction de blocs, effectuer des simulations et enfin l'analyse des résultats.

### 2. Hardware :

Ordinateur portable Toshiba satellite dans les caractéristiques suivantes :

- *CPU Intel core i3-3217U @ 1.80 GHz ;*
- *RAM 4 Go ;*
- *Toshiba webcam HD.*

### 3. Objectif

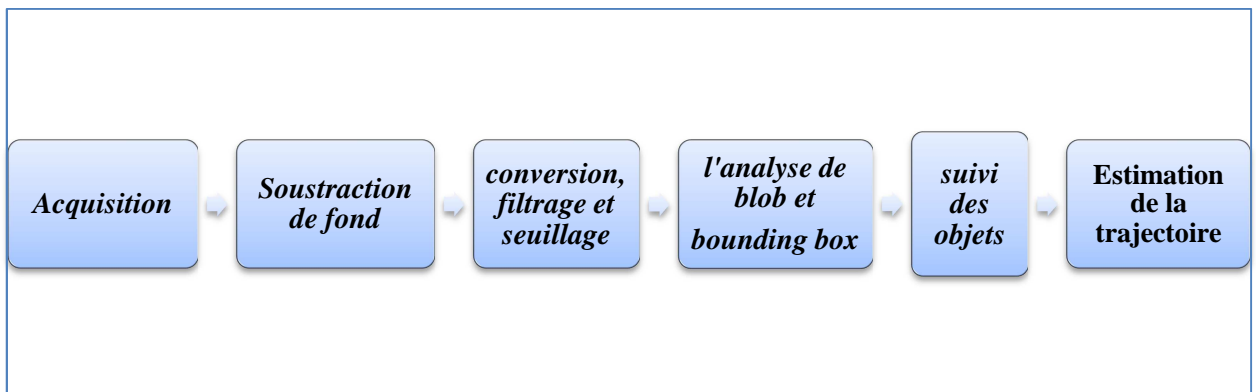
L'objectif de notre travail est d'appliquer le filtre de Kalman dans le suivi des objets multiples en mouvement. Dans les deux cas possibles de l'implémentation de ces algorithmes:

1<sup>ère</sup> cas : Une base de donnée qui existe déjà (vidéo enregistrées).

2<sup>ème</sup> cas : Des scènes captées en temps réel par des webcams.

### 4. Synoptique générale de système :

Afin de pouvoir implémenter ces méthodes et résoudre certains problèmes évoqués auparavant qui touchent le suivi et l'estimation de la dynamique des objets ou bien des obstacles en mouvement, on opte pour la solution résumée par le synoptique ci-dessous :



*Figure 3.1. Synoptique général de la présente étude.*

### 5. Les étapes à suivre

Les étapes suivant montrent comment nous pouvons implémenter une méthode de détection et suivi des objets multiples en mouvement sur SIMULINK en se servant de la

Toolbox acquisition d'image et computer vision, qui sont utilisée essentiellement pour concevoir un tel système robuste et performant. Afin d'atteindre cet objectif, nous devons suivre les étapes suivantes :

#### Etape 1 : Acquisition

L'acquisition des données se fait selon deux cas

- fichiers vidéo qui existent déjà (vidéos enregistrées).
- vidéos captées directement (temps réel) via une caméra.

#### Etape 2 : Soustraction de fond

L'application détecte un mouvement dans la scène en effectuant une comparaison pixel à pixel dans des paires de trames d'images entrantes.

#### Etape 3 : conversion, filtrage et seuillage

Des opérations morphologiques et des filtrages nécessaires sont appliqués au masque de premier plan pour éliminer le bruit résultant.

#### Etape 4 : l'analyse de blob et bonding box

Enfin, l'analyse de blob détecte des groupes de pixels connectés, qui sont susceptibles de correspondre à des objets en mouvement.

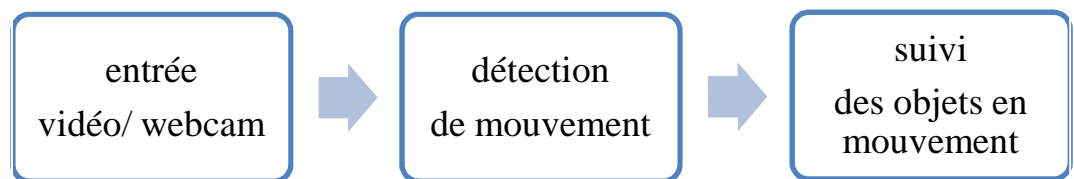
#### Etape 5 : Le suivie des objets multiples avec le filtre de kalman

Après l'analyse de blob et bounding box on peut alors faire le suivie ou le tracking de ces objets détecter. Le mouvement de chaque voie est estimé par un filtre de Kalman. Le filtre est utilisé pour prédire l'emplacement de la piste dans chaque trame, et de déterminer la probabilité de chaque détection étant attribué à chaque piste.

### 6. Choix de Simulink

Le choix de Simulink est pour :

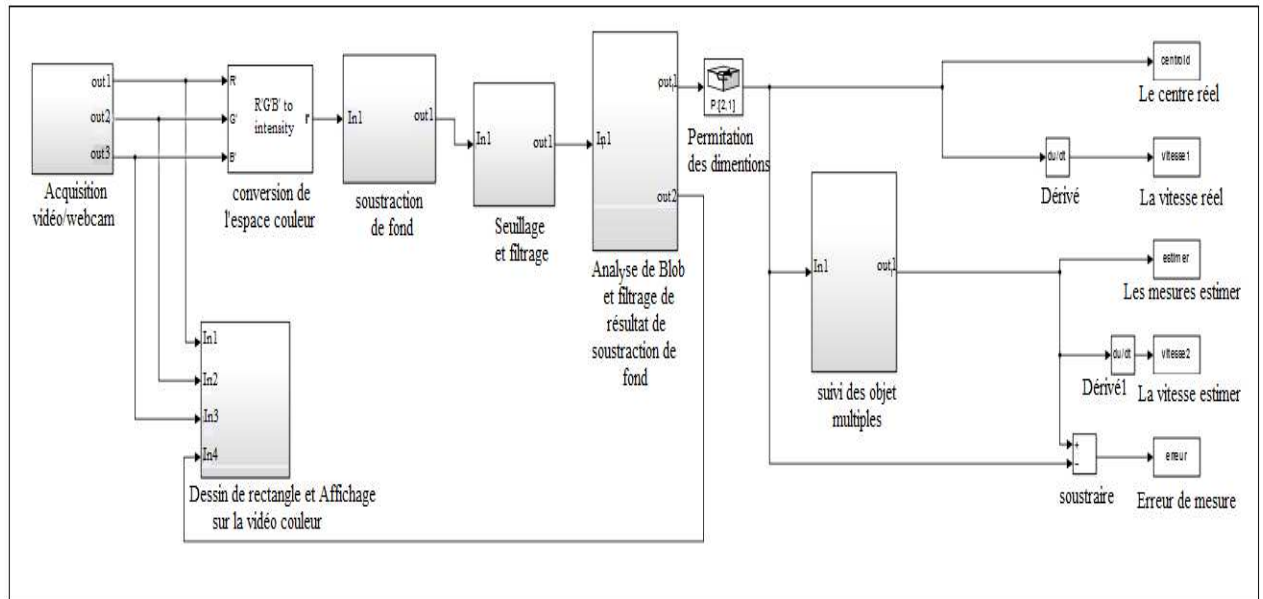
1. Les blocks Simulink de traitement d'image et de vidéo permettent d'effectuer des simulations rapides même pour des systèmes de détection en temps réel.
2. Il peut créer des spécifications exécutables pour communiquer le système en aval de conception d'équipe et de fournir une référence en or pour la vérification tout au long du processus de conception.
3. La quantité de travail ne varie pas avec la complexité ou la durée de la performance au même degré que lors de l'utilisation des techniques traditionnelles. Le but de cette partie est de mettre en œuvre une méthodologie efficace pour détecter et suivre des objets multiples en mouvement à l'intérieur des vidéos. La figure 3.2 montre le schéma synoptique général du système :



**Figure 3.2 :** Block Diagramme de Système.

### 7. Le Modèle Simulink proposé pour la détection et suivi des objets en mouvement :

Le modèle Simulink représenté sur la figure (3.3) réalisé sur MATLAB 2015a à l'aide du navigateur de la bibliothèque SIMULINK qui contient des ensembles des blocs de traitement d'image et de vidéo.



**Figure 3.3 : Le Modèle Simulink proposé.**

### Description de modèle et des blocs

Pour la détection et le suivi des objets en mouvement le modèle proposé contient plusieurs blocs de traitement qui commence par la partie acquisition des données cette dernière se constituent elle-même de deux blocs d'acquisitions différentes :

- fromVideoDevice : avec ce bloc nous permet d'acquérir les données de flux vidéo provenant des dispositifs d'acquisition d'image, telle la caméra. Donc le bloc est pour l'acquisition en temps réel.
- FromMultimedia File : ce bloc lit les échantillons des images de vidéo enregistré.

Après l'acquisition à partir de l'un des deux blocs précédents, les trames passent par le prétraitement. En prétraitement les trames en couleurs sont converties en échelle de gris, avec le bloc de conversion de l'espace de couleur «color space conversion». Donc nous spécifions la conversion de RGB à intensité.

Ensuite, les trames converties passent par l'opération de soustraction de fond pour créer une image de référence pour chaque image courante. Afin d'obtenir les pixels en mouvement.

Puis c'est au tour d'un filtrage médian et un seuillage avec leurs blocs correspondants « median filtre », « Autothreshold » afin de créer un masque binaire sur les images (noire et blanc), pour soustraire les objets de premier plan et de l'arrière-plan et pour éliminer le bruit résultant.

Ensuite, nous avons utilisé le bloc « blob analyse » pour calculer les statistiques des régions marquées dans les images binaires. Les informations sorties de ce bloc sont : le centre de masse, boîte englobant, matrice de l'étiquette, et le nombre de blob. Les ports utilisés dans notre cas sont :

Port	Entrée / sortie
BW	-Un vecteur ou une matrice qui représente une image binaire.
Centroid	-[M-2] sous-matrice de coordonnées de centre de masse, où M représente le nombre de blobs.
BBox	-matrice M-en-4 de [x y largeur hauteur] coordonnées de boîte englobant, où M représente le nombre de blobs et [x y] représente le coin supérieur gauche de la zone de délimitation.

**Tableau 3.1 :** description des ports utilisé dans notre étude [19].

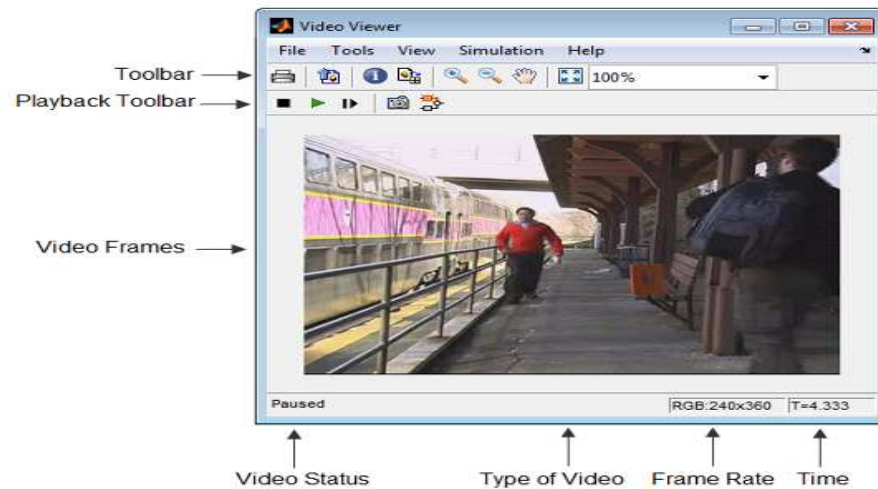
Après, place à un cadre où rectangle qui est géré par le bloc « DrawShape ».

Enfin, les objets détecter et tout ce traitement en parallèle effectué par les blocs Simulink vont nous fournir des informations utiles (centroid) qui vont servir eux-mêmes comme entrées au bloc du « filtre de kalman » pour qu'il puisse estimer les positions, les vitesses et le déplacement des centres de masse des objets suivis autrement dit la trajectoire parcourue par ces derniers.

La visualisation et l'affichage des résultats sont marqués par les deux blocs :

- VideoViewer : ce bloc nous permet de visualiser les résultats des images binaires de soustraction et les résultats de suivi sur la vidéo en couleur. Le bloc fournit des

commandes de simulation pour le jeu, pause, et l'étape lors de l'exécution du modèle. Le bloc fournit également des outils d'analyse de la région de pixel.



- To workspace : Pendant la simulation, le bloc écrit des données dans une mémoire-tampon interne. Dans notre cas les centres de masse. Lorsque la simulation est terminée ou interrompue, que les données sont écrites dans l'espace du travail. Ce dernier nous permet de les avoir afin de les réutilisées ultérieurement.

## 8. Résultats et Analyse Expérimentale

### Aperçu du système

Pour détecter et suivre l'apparence physique des objets mobiles tels que les personnes et les véhicules et l'identifier en scène dynamique, on tout d'abord localiser la position, estimer le mouvement de ces blobs et de suivre leurs mouvements entre la trame de référence et les trames consécutives dans la vidéo. L'application se reproduise sur deux types des séquences vidéo : enregistrer ou capter en temps réel.

#### *1. 1<sup>er</sup> Cas : vidéos enregistrées.*

Nous utilisons des séquences de la circulation des véhicules afin de vérifier la méthode de détection et le suivi basé sur le filtre de Kalman. Pour cela, nous avons choisi des séquences d'images de vidéo enregistrées, la vidéo représente le mouvement des véhicules dans une bande de circulation leurs caractéristiques et comme suit :

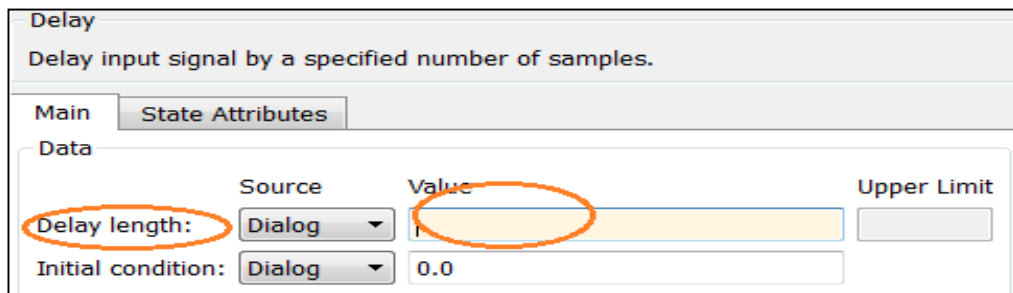


- Nom et type de vidéo: 'viptraffic.avi'.<sup>1</sup>
- Longueur (s) : 00:00:10 ;
- Largeur de trame : 160 ;
- Hauteur de trame : 120 ;
- Fréquence d'images : 15 trames/s ;
- Débit de données : 2048 Kbits/s ;
- Format de la vidéo: 'RGB24'.

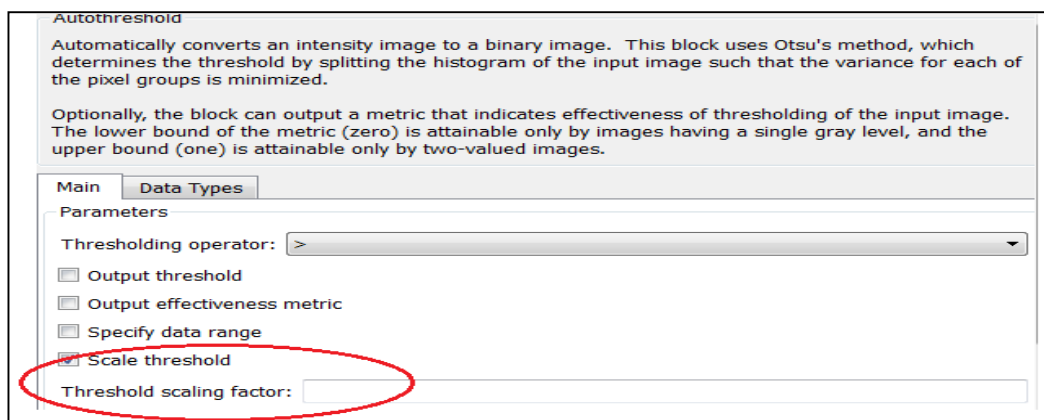
### I.1 Détection de mouvement par soustraction de fond

L'implémentation SIMULINK de l'algorithme de segmentation de mouvement par soustraction d'arrière-plan, basée sur le choix des quatre paramètres suivants :

- 1) longueur de retard (DL): C'est le délai de décalage entre l'image de référence et l'image courante.

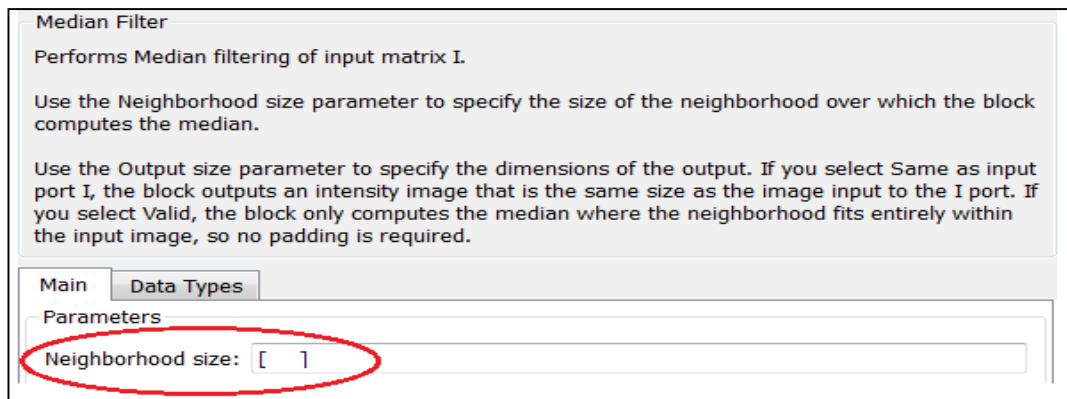


- 2) facteur d'échelle de seuil(ThS): représente le coefficient de seuil de séparation des objets de l'arrière-plan.

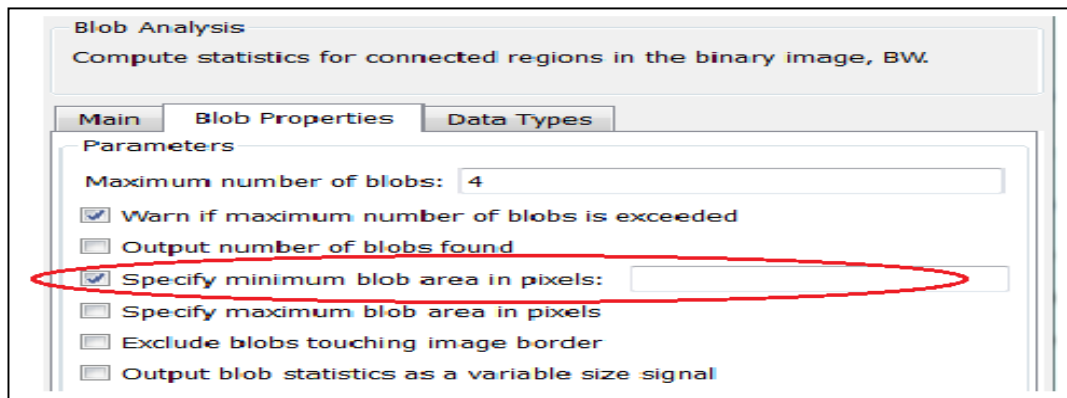


<sup>1</sup> MATLAB\MATLAB Production Server\R2015a\toolbox\vision\visiondata\viptraffic.avi

### 3) le filtre médian (Ns):



### 4) Le nombre minimal de pixels spécifiée dans la zone de blob (Minarea):



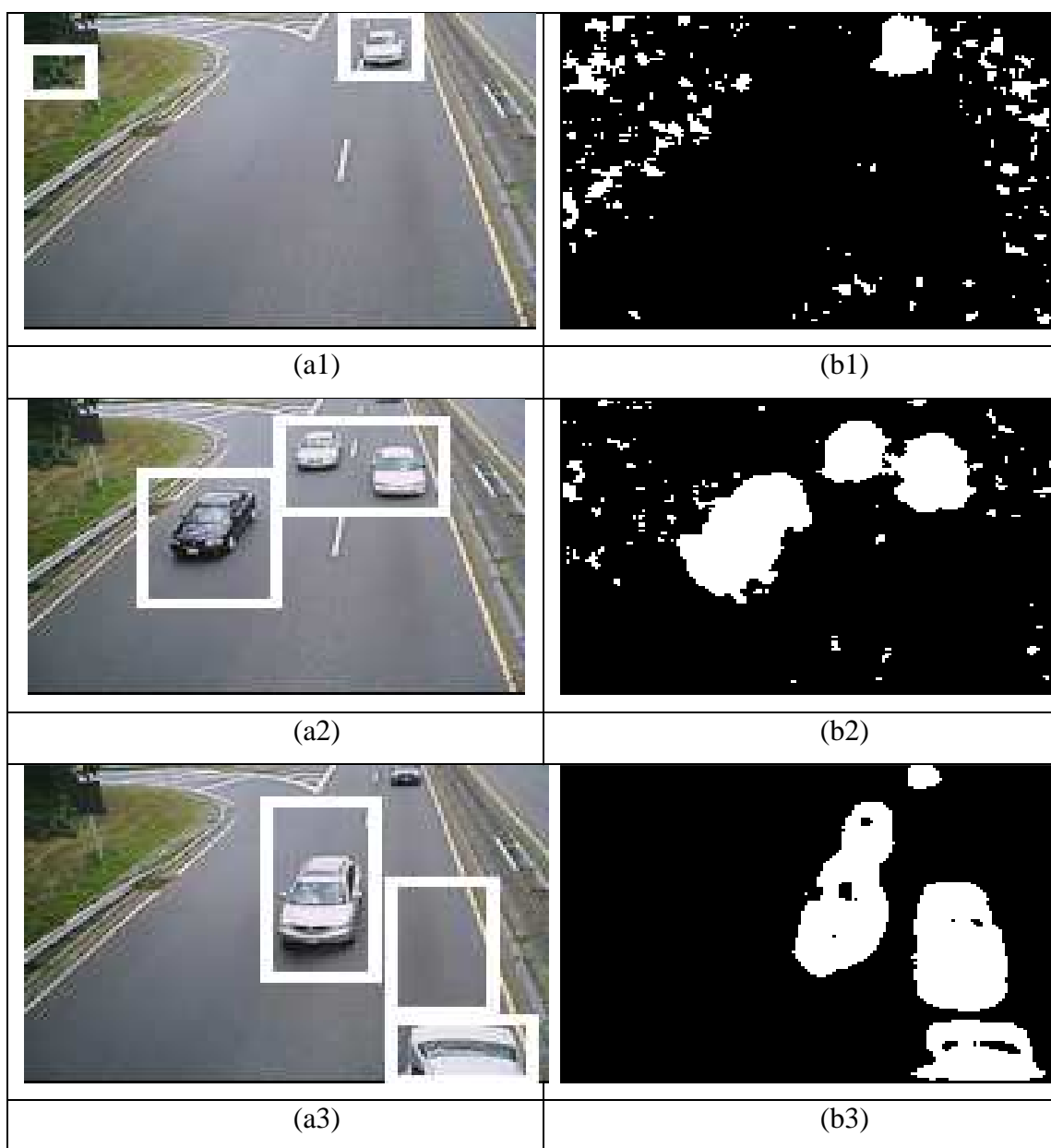
## I.2 Résultats de détection et suivi de mouvement

Le but de cette méthode de détection par soustraction de fond est de séparer l'objet en mouvement de leurs environnements afin d'extraire des informations utiles pour le traitement et la poursuite, telles que la position, la vitesse, et la trajectoire...etc.

Nous allons faire, dans ce qui suit une étude des résultats de détection des véhicules en mouvement dans des différentes variations de quatre paramètres du système. On peut observer un changement significatif dans la réaction du système dans chaque variation de quelque paramètre. Les images de la figure.3.4 sont captées sous les plages des paramètres suivants:

Paramètres n°1	DL	NS	ThS	Minarea
Les plages	$Z^{-n} \geq Z^{-3}$	[4 4]	$S \geq 0.5$	$M > 150$

Les images ci-dessous représentent les résultats de détection des véhicules :



**Figure 3.4 :** résultats de la détection de mouvement avec des paramètres n°1.

Discussion des résultats concernant les paramètres n°01 :

Dans les images (a1 et b1) de la figure (3.4) on remarque :

- des fausses détections ou une absence de détection bien que le mouvement existe. à cause du mauvais réglage du seuil qui peut conduire à un taux élevé de bruit (ex. le seuil choisi dans l'image :  $ThS=1$ ).

Aussi on trouve dans l'image (a2 et b2) de la figure (3.4):

- Un assemblage des deux ou plusieurs véhicules au même blob, à cause de faible choix de la surface minimum des blobs qui peut faire un grand chevauchement entre les objets proches. Donc le résultat est une fusion des deux objets.

Dans les images (a3 et b3) deux problèmes existés :

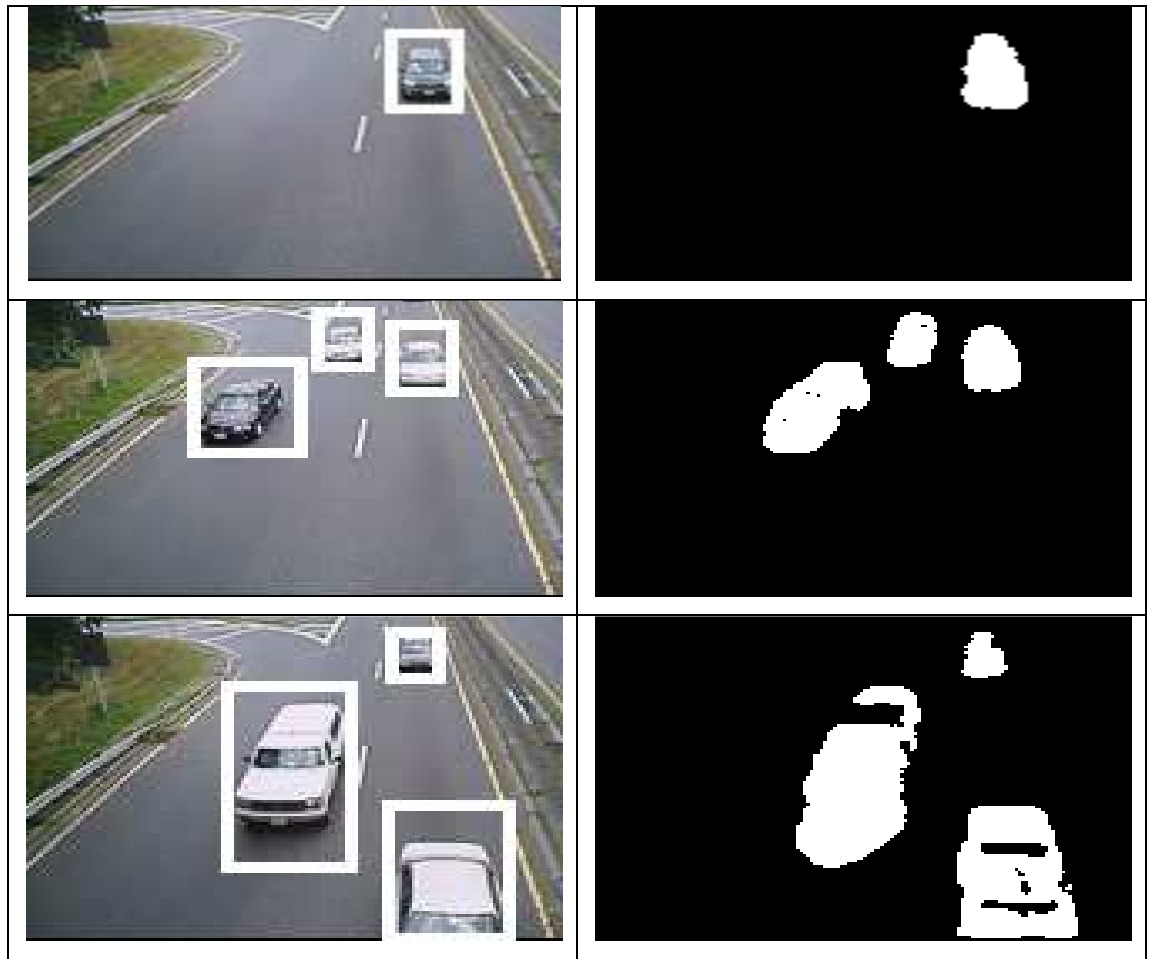
- une absence de détections quoique le mouvement du véhicule existe, donc une absence d'informations dans la détection des objets en mouvement. Ça résultait au mal choix de la zone minimum surtout le choix des seuils élevés de la surface détectée (ex. Dans ce cas :  $Minarea=500$ ).
- aussi quand on fera un grand délai de décalage dans les images (ex. dans notre cas:  $DL = Z^{-5}$ ), des faux rectangles sont apparu autour des objets détectés.

Les différents tests avec les variations des différents paramètres sur le modèle avec ce type de vidéo prouver notre choix des paramètres suivants :

Paramètre n°2	DL	NS	ThS	Minarea
Les plages choisis	$Z^{-1} \leq Z^{-n} \leq Z^{-2}$	[4 4]	$0.3 < Ths < 0.5$	$90 < M < 150$

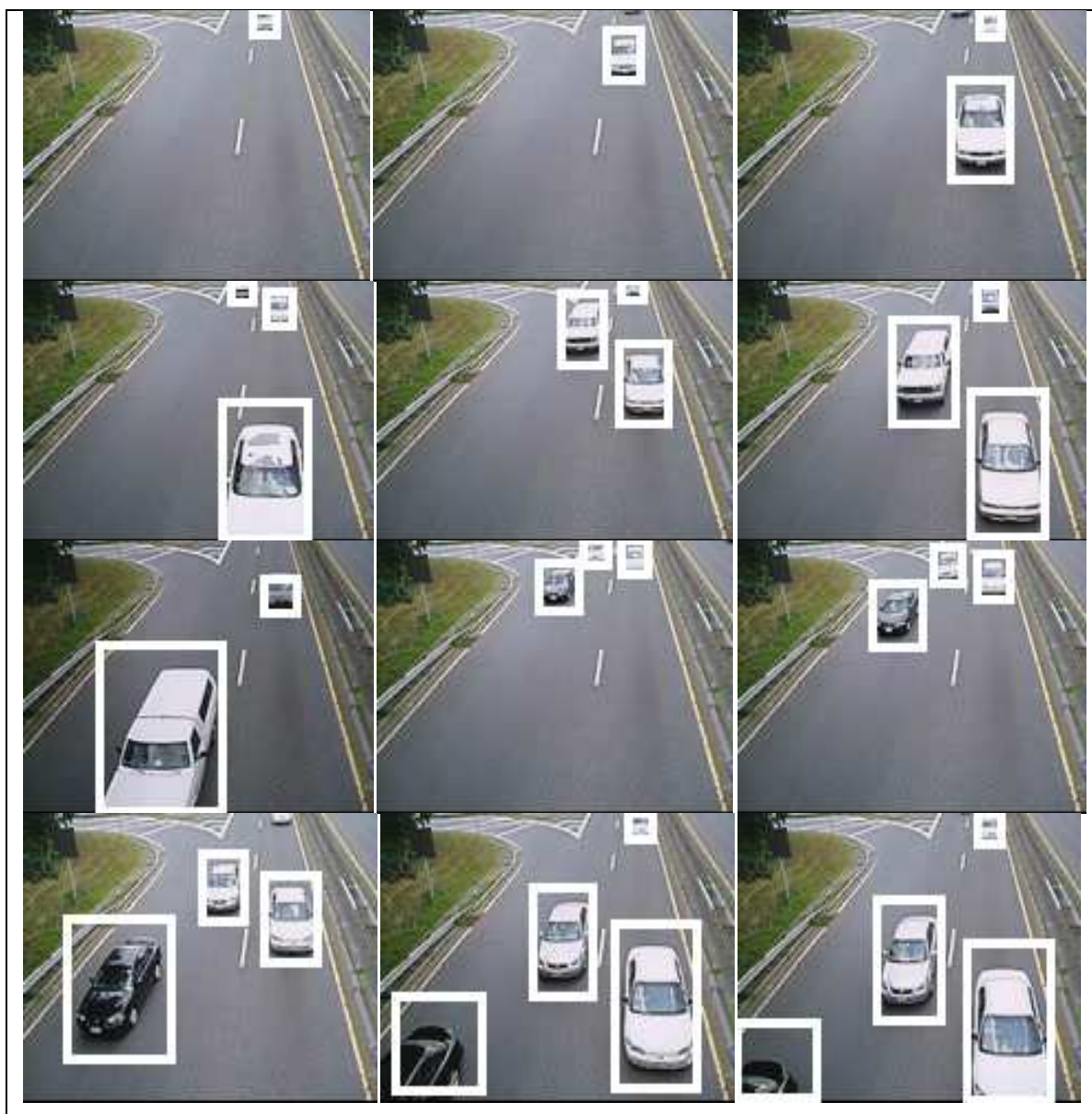
Remarque : Le choix de ces paramètres est relié directement avec les propriétés de la vidéo, beaucoup plus a la fréquence des trames.

Les résultats suivants représentent l'influence du bon choix des paramètres sur la détection et la soustraction des images binaires :



**Figure 3.5 :** *résultats de la détection de mouvement avec les paramètres n°2.*

La figure 3.6 montre quelques frames de la vidéo de suivi que nous avons obtenue. Nous remarquons que nous avons pu suivre tous les véhicules qui sont en mouvement dans la scène surveillée.



*Figure 3.6 : résultats de suivi avec les paramètres n°2.*

Discussion des résultats concernant les paramètres n°02 :

Le choix de ces derniers paramètres dans la simulation fournit des bons résultats de détection et de suivi des véhicules. Aussi, les différents tests sur le modèle semblent prouver son invariance face aux conditions non contrôlées de l'environnement ainsi qu'une rapidité d'exécution convenable pour la contrainte de la variation de vitesse des véhicules.

### I.3 Implantation du filtre de kalman pour l'estimation et le suivi

Le filtre de Kalman sera employé dans cette partie pour soutenir l'étape de détection et pour faire l'estimation des positions des centres de masses, estimation de la vitesse et aussi l'extraction de la trajectoire des véhicules suivis.

Nous avons fixé dans l'étude suivante maximums quatre objets en mouvement qui doit être suivis par Kalman dans chaque trame des séquences vidéo. Les paramètres qui suivent, illustrent les différents résultats obtenus sur le même système de surveillance de trafic utilisant cette approche. Nous avons intéressé par l'étude de la position et la vitesse, donc nous avons fixé les quatre paramètres comme suit :

- matrice de transition d'état A.

$$A = [1 \ 0 \ 0 \ 0; 0 \ 1 \ 0 \ 0; dt \ 0 \ 1 \ 0; 0 \ dt \ 0 \ 1].$$

- matrice de mesure H.

$$H = [1 \ 0 \ 0 \ 0; 0 \ 1 \ 0 \ 0].$$

Conditions initiales :

- la condition initiale pour l'état estimé.

$$\hat{X}_0 = \text{Zeros}([4, 1]).$$

- La condition initiale pour la covariance d'erreur estimée

$$\hat{P}_0 = 10 * \text{eye}(4).$$

Notons que les performances et les résultats du filtre récursif de Kalman avec estimation quadratique linéaire s'appuient essentiellement sur la variation des deux paramètres suivants:

$$(\text{SigmaQ}, \text{SigmaR}) \in (\mathbb{R}^+)^2.$$

- ✓ La matrice de covariance de bruit de processus :  $Q = \text{sigmaQ} * \text{eye}(4)$ .
- ✓ La matrice de covariance de bruit de mesure :  $R = \text{sigmaR} * \text{eye}(4)$ .

### I.4 Résultats obtenu de l'implémentation de KF

Après la détection des objets en mouvement obtenu par un ensemble de traitement en parallèle et un réglage (tubings) d'un ensemble de paramètres ce qui nous a donnés finalement des résultats tant souhaités.

Maintenant à l'aide de filtres de Kalman on va estimer la position et la vitesse de chaque objet détecté dans les quatre rectangles, pour le but de réduire et minimiser l'erreur de bruit de mesure. Dans l'étude suivante nous avons choisi de discuter les résultats des deux cas de mesures différentes dans les paramètres Q et R de filtre. Afin de clarifier l'effet de chaque paramètre sur le processus de suivi.

#### I.4.1 Etude de l'influence de quelque paramètre de filtre sur le suivi

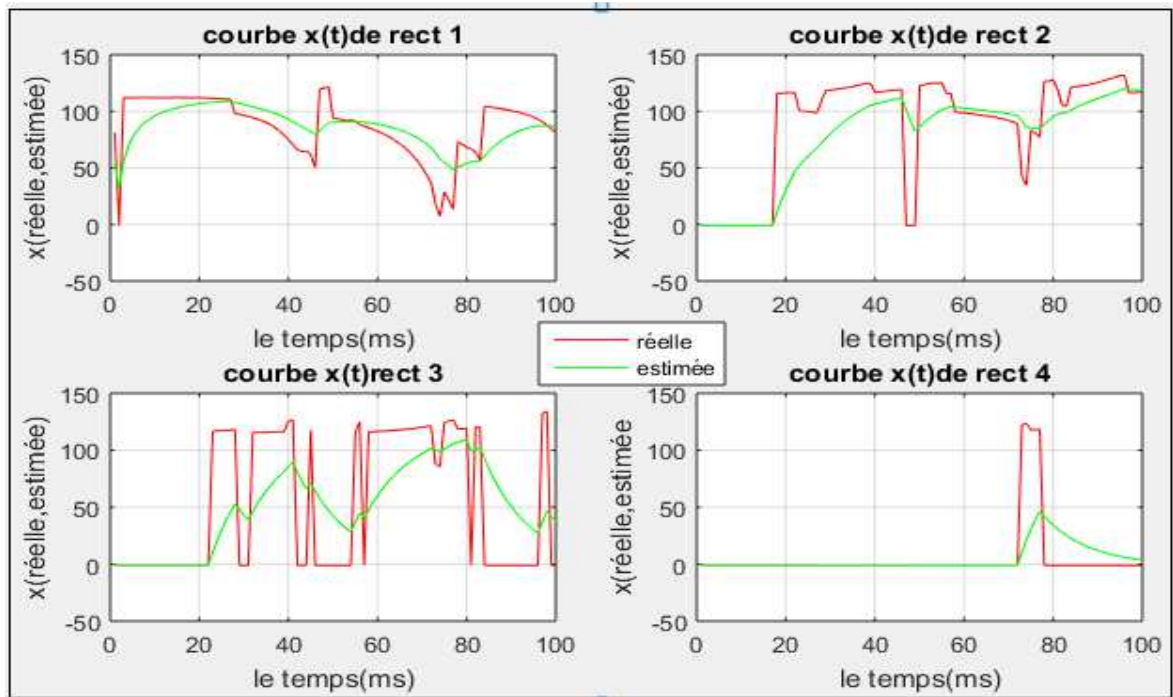
La qualité du filtrage peut dépendre de la nature du signal à traiter et de certains paramètres de filtre. Nous allons, dans ce qui suit étudier l'influence des valeurs de la matrice de covariance de bruit de processus « Q » et la matrice de covariance de bruit de mesure « R ».

##### I.4.1.1 Influence de la matrice de la covariance de bruit de mesure :

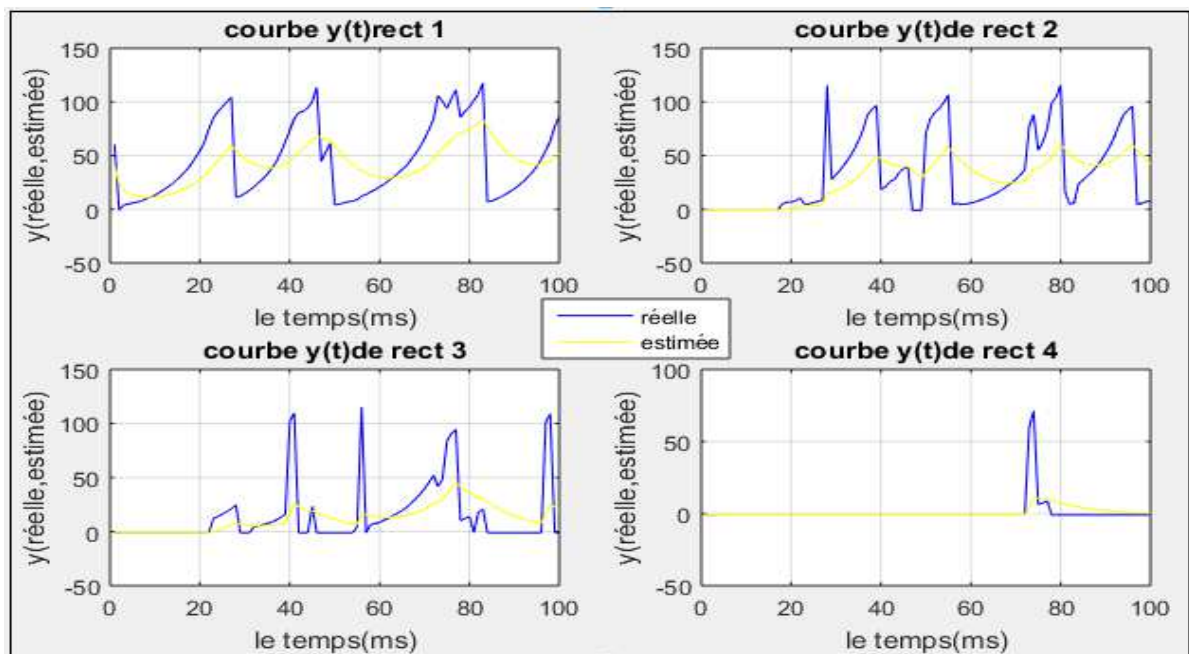
Nous faisons tester différentes valeurs de la matrice de covariance de bruit de mesure, tel que en augmentant la valeur  $\sigma_R$  jusqu'à 5 et fixer la valeur de  $\sigma_Q$  de la matrice Q à 0.05 (Q est par défaut). On obtient alors les figures suivantes :

Les deux figures ci-dessous représentées les courbes des signaux de mesure des positions  $X(t)$  et  $Y(t)$  des centres de masse et les signaux  $\hat{X}(t)$  et  $\hat{Y}(t)$  donné par le filtre de kalman.



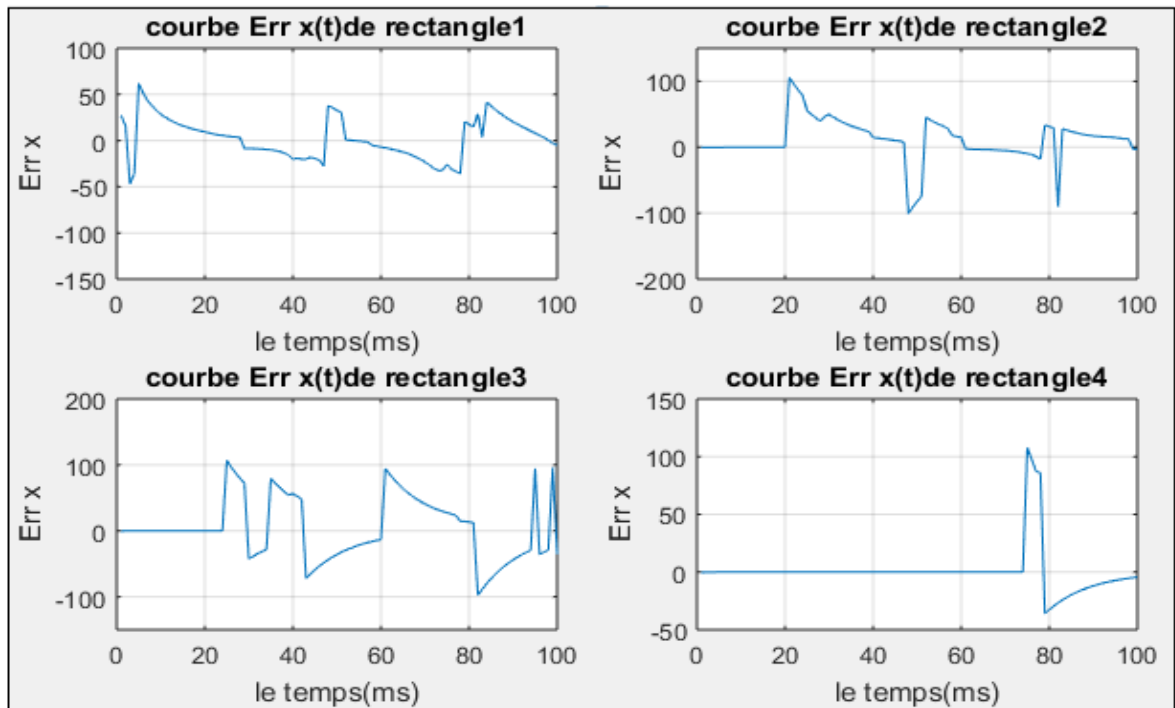


**Figure 3.7 :** Les courbes  $\hat{X}(t)$  et  $X(t)$  des positions des centres de masse.

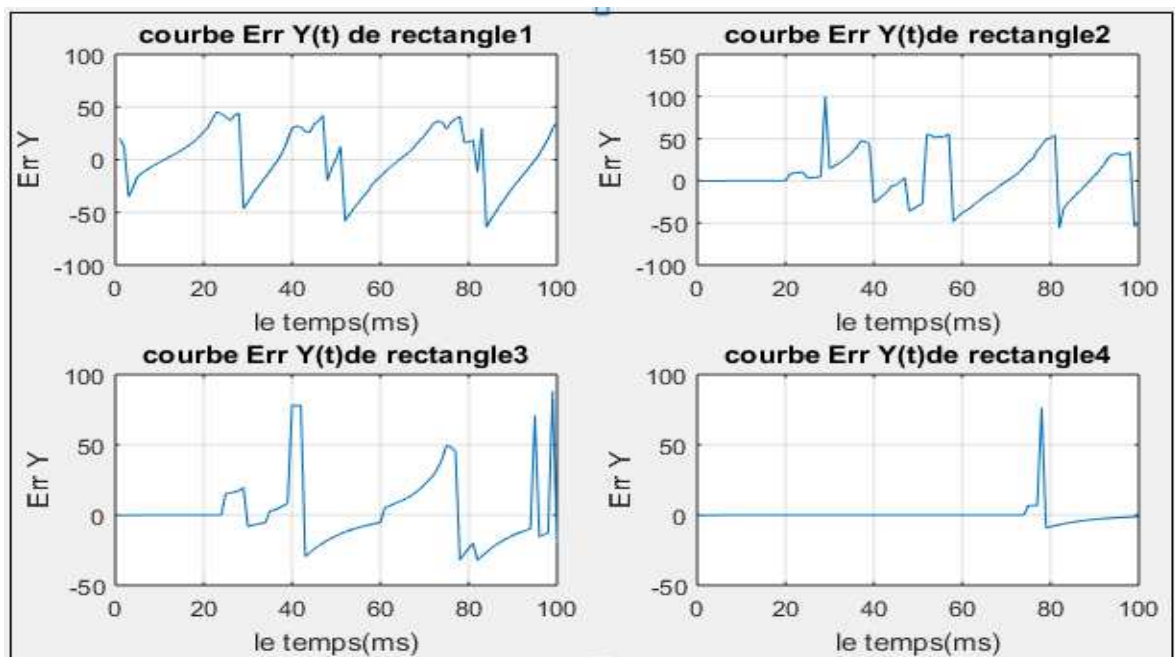


**Figure 3.8 :** Les courbes  $\hat{Y}(t)$  et  $Y(t)$  des positions des centres de masse.

Donc les erreurs de mesure  $\tilde{X}(t)$  et  $\tilde{Y}(t)$  obtenues des quatre objets suivis sont représentées sur les deux figures suivantes:



**Figure 3. (9.a) :** résultat de l'erreur de mesure des positions  $\tilde{X}(t)$ .



**Figure 3. (9.b) :** résultat de l'erreur de mesure des positions  $\tilde{Y}(t)$ .

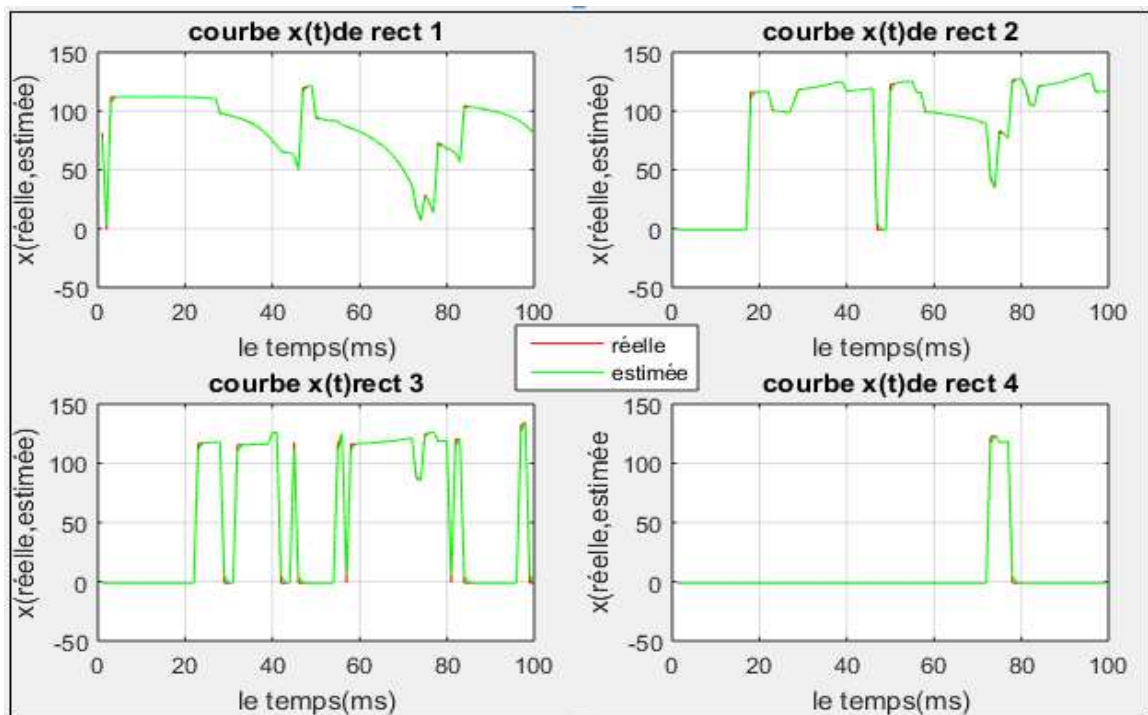
Nous remarquons qu'après avoir fixé la valeur de  $\sigma_Q$  de covariance de bruit de processus à 0.05 et augmenté la valeur de  $\sigma_R$  de la covariance de bruit de mesure à 5, la

qualité des signaux  $\hat{X}(t)$  et  $\hat{Y}(t)$  sortie par le filtre est encore bruitée .donc on ne peut pas dire que la qualité de l'estimation est satisfaite.

### I.4.1.2 Influence de la matrice de covariance du bruit de processus :

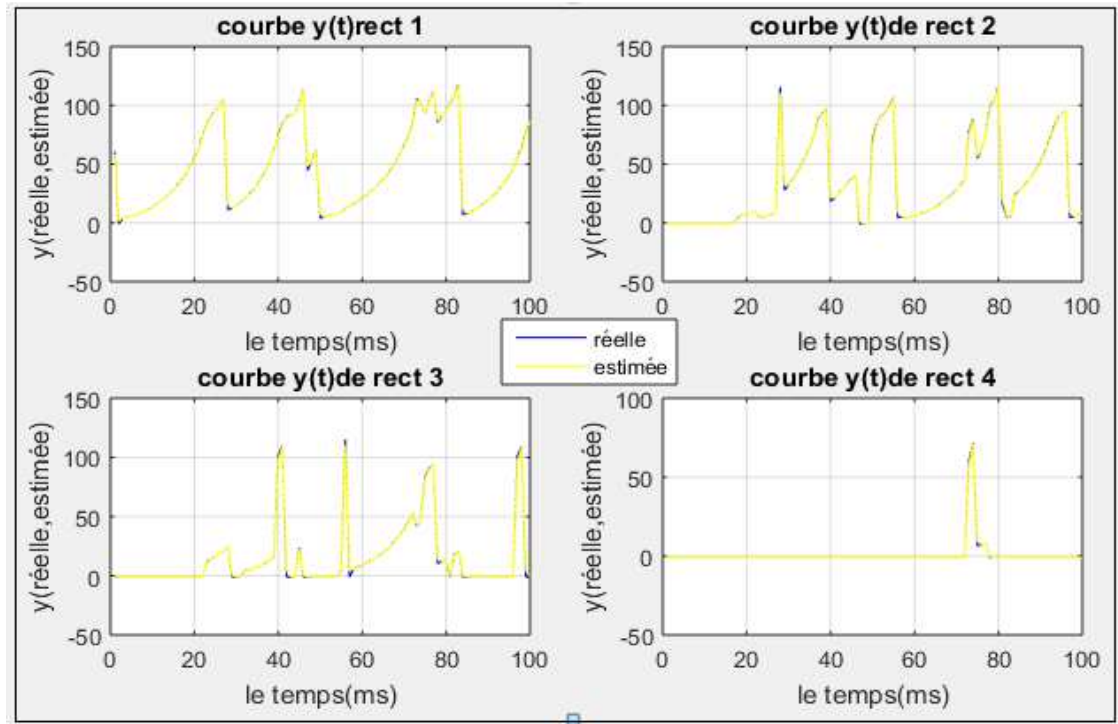
Nous faisons varier la valeur de la matrice de covariance de bruit de processus, en augmentant leur valeur de coefficient sigmaQ jusqu'à 40 et fixait la valeur de sigmaR à 2. On obtient alors les résultats dans les figures suivantes :

La première figure représente les signaux de mesure des positions  $X(t)$  et  $\hat{X}(t)$  des centres de masse des véhicules suivis.



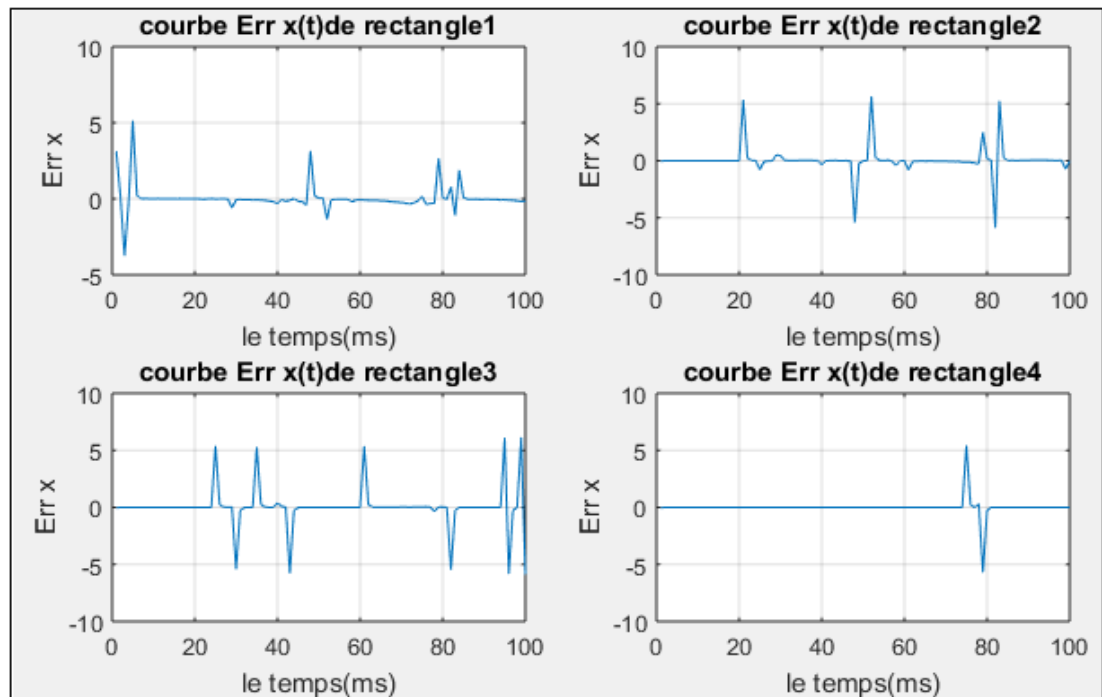
**Figure 3.(10.a):** les courbes  $\hat{X}(t)$  et  $X(t)$  des positions des centres de masse.

La figure qui suit, nous donne les signaux  $Y(t)$  et  $\hat{Y}(t)$  avec un coefficient de covariance de bruit de mesure égale à 40 et sigma R=2 :

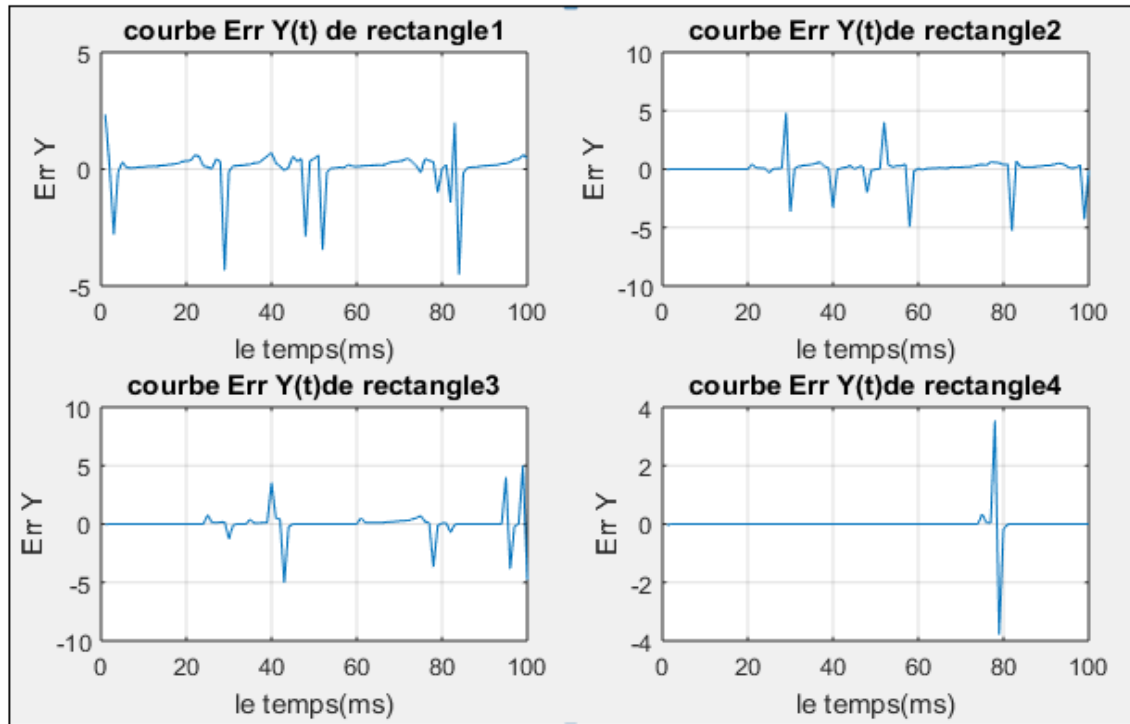


**Figure 3.(10.b):** Les courbes  $\hat{Y}(t)$  et  $Y(t)$  des positions des centres de masse.

Donc les courbes des erreurs de position  $\tilde{X}(t)$  et  $\tilde{Y}(t)$  obtenue sont représentées sur les deux figures suivantes:



**Figure 3.(11.a) :** résultat de l'erreur de mesure de positions  $\tilde{X}(t)$ .



**Figure 3.(11.b) : résultat de l'erreur de mesure de position  $\tilde{Y}(t)$ .**

Discussion des résultats:

Après avoir augmenté la valeur de covariance de bruit de processus  $Q$  à 40 est diminué la valeur de covariance de bruit de mesure  $R$  à 2. Nous constatons que le signal estimé par le filtre de kalman convergé vers le signal des positions réel des objets suivis. Donc on peut dire que la qualité de l'estimation a augmenté.

### I.4.1.3 Comparaison des résultats précédents

Nous allons faire, dans ce qui suit une comparaison des deux mesures précédentes, tel que :

- Mesure n°01 :  $Q=0.05 \cdot \text{eye}(4)$  ;  $R=5 \cdot \text{eye}(4)$ .
- Mesure n°02 :  $Q=40 \cdot \text{eye}(4)$  ;  $R=2 \cdot \text{eye}(4)$ .

Les tableaux ci-dessus représentent des échantillons de mesures des signaux  $X(t)$  et  $\hat{X}(t)$  des positions des centres de masse des véhicules suivis, pendant des instants choisis  $T$  (ms):

## Chapitre III: implémentation, résultats et discussion

Rect n°= numéro de la rectangle suivi.

### Mesure n°01

Bruit de processus: $Q=0.05*\text{eye}(4)$  Bruit de mesure $R=5*\text{eye}(2)$  T (ms)	X				$\hat{X}$			
	Rect 1	Rect 2	Rect 3	Rect 4	Rect 1	Rect 2	Rect 3	Rect 4
6	112.03	-1	-1	-1	60.55	-0.92	-0.92	-0.92
10	12.00	-1	-1	-1	83.42	-0.96	-0.96	-0.96
15	112.10	-1	-1	-1	96.14	-0.977	-0.97	-0.97
20	112.03	-1	-1	-1	102.72	-0.98	-0.98	-0.98
25	111.08	99.95	117.26	-1	106.16	44.40	10.40	-0.99
48	119.92	-1	-1	-1	81.82	99.55	42.71	-0.99
81	69.86	127.98	119.46	-1	54.88	99.01	107.04	28.47

**Tableau 3.2 :** résultats de la mesure n°1 des positions réels et estimés.

Donc l'erreur de mesure  $\tilde{X}(T)$  obtenue est représentée sur le tableau suivant:

$\tilde{X} =  x - \hat{x} $				
T (ms)	Rect 1	Rect 2	Rect 3	Rect 4
6	51.47	0.07	0.07	0.07
10	28.57	0.03	0.03	0.03
15	15.96	0.02	0.02	0.02
20	9.31	0.01	0.01	0.01
25	4.91	55.55	106.85	0.00
48	38.09	100.55	43.79	0.00
81	14.98	28.97	12.39	29.47

**Tableau 3.3 :** résultats de la mesure n°01 de l'erreur de la position des 4 rectangles de suivi.

## Mesure n°02

Bruit de processus $Q=40 \cdot \text{eye}(4)$ )	Les mesures réelles				Les mesures estimées			
	$X$				$\hat{X}$			
Bruit de mesure $R=2 \cdot \text{eye}(2)$ T (ms)	Rect 1	Rect 2	Rect 3	rect 4	Rect 1	Rect2	Rect3	Rect 4
6	6.81	-1	-1	-1	6.75	-1	-1	-1
10	12.88	-1	-1	-1	12.71	-1	-1	-1
15	26.99	-1	-1	-1	26.82	-1	-1	-1
20	54.72	-1	-1	-1	54.37	-1	-1	-1
25	90.57	6.12	117.26	-1	90.48	6.44	111.87	-1
48	49.70	-1	-1	-1	52.58	0.99	-1	-1
81	94.31	115.52	119.43	-1	94.12	115.09	119.41	-0.98

**Tableau 3.4 :** résultats de la mesure n°2 des positions réelles et estimées.

Donc l'erreur de mesure  $\tilde{X}(T)$  obtenue est représentée sur le tableau suivant:

$\tilde{X} =  x - \hat{x} $ T (ms)	Rect 1	Rect 2	Rect 3	Rect 4
6	0.06	0.00	0.00	0
10	0.09	0.00	0.00	0
15	0.16	0.00	0.00	0
20	0.35	0.00	0.00	0
25	0.09	0.32	5.38	0
48	2.88	1.99	0	0
81	0.18	0.43	0.01	0.01

**Tableau 3.5 :** résultats de la mesure n°02 de l'erreur de positions.

Donc les figures suivantes on représente le résultat de comparaison entre l'erreur de mesure  $\tilde{X}(T)$ , des deux cas 1 et 2 précédents:

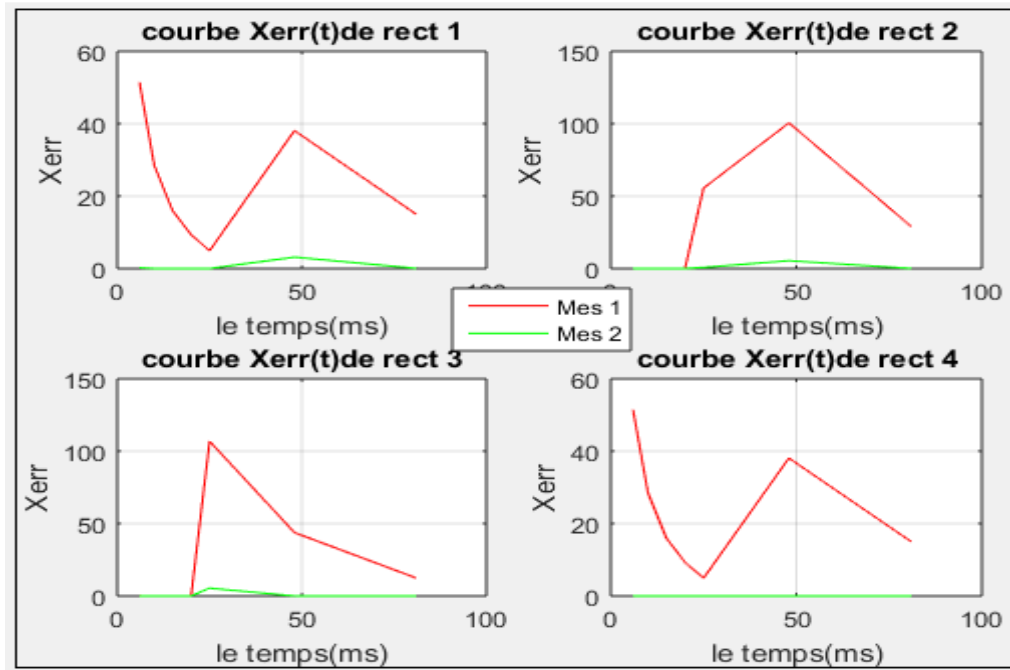


Figure 3.(12.a) : Comparaison des signaux  $\tilde{X}(T)$  dans les deux cas de mesure.

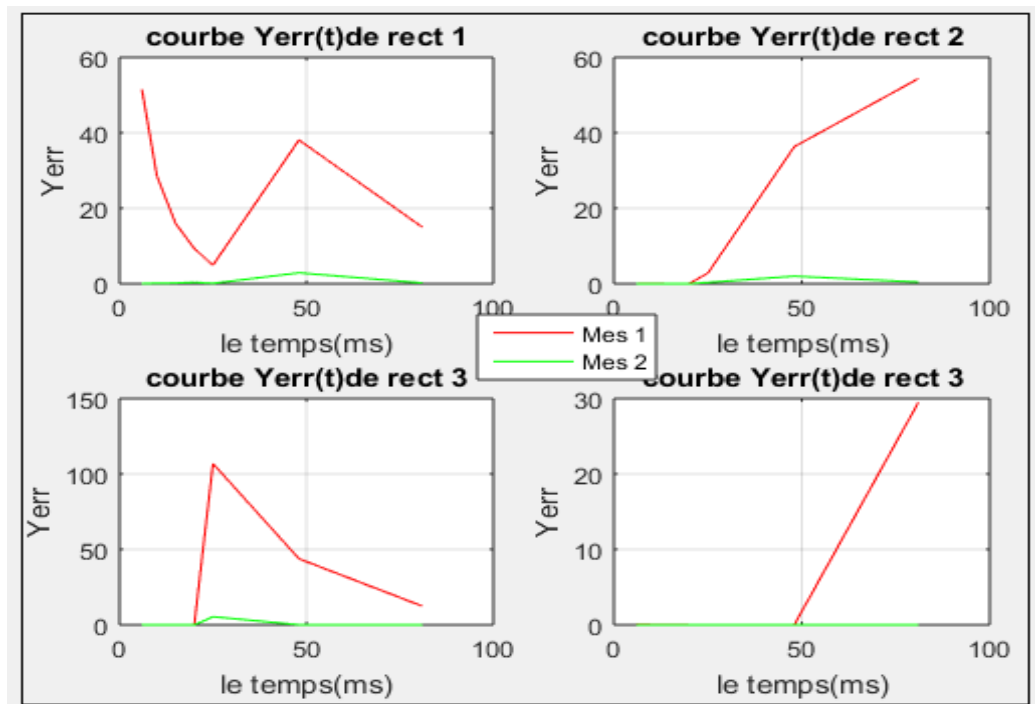


Figure 3.(12.b) : Comparaison des signaux  $\tilde{Y}(T)$  dans les deux cas de mesure.



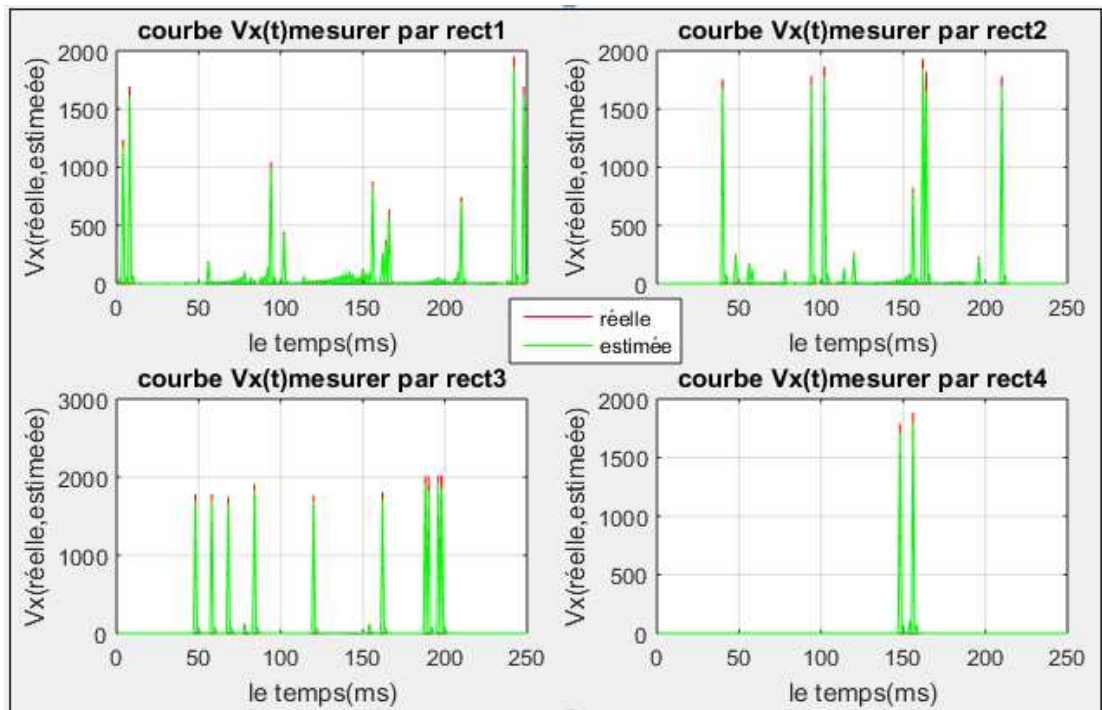
Comme conclusion on peut dire que :

- On remarque que l'influence de ces derniers est très importante pour l'obtention de meilleurs résultats.
- On remarque aussi qu'à travers une comparaison entre les courbes des erreurs de mesures, que le deuxième cas (mesure n°02) est considéré comme le meilleur en matière de taux d'erreur (erreur tend vers 0). Donc le système est considéré comme un très bon suiveur avec les paramètres de deuxième cas.

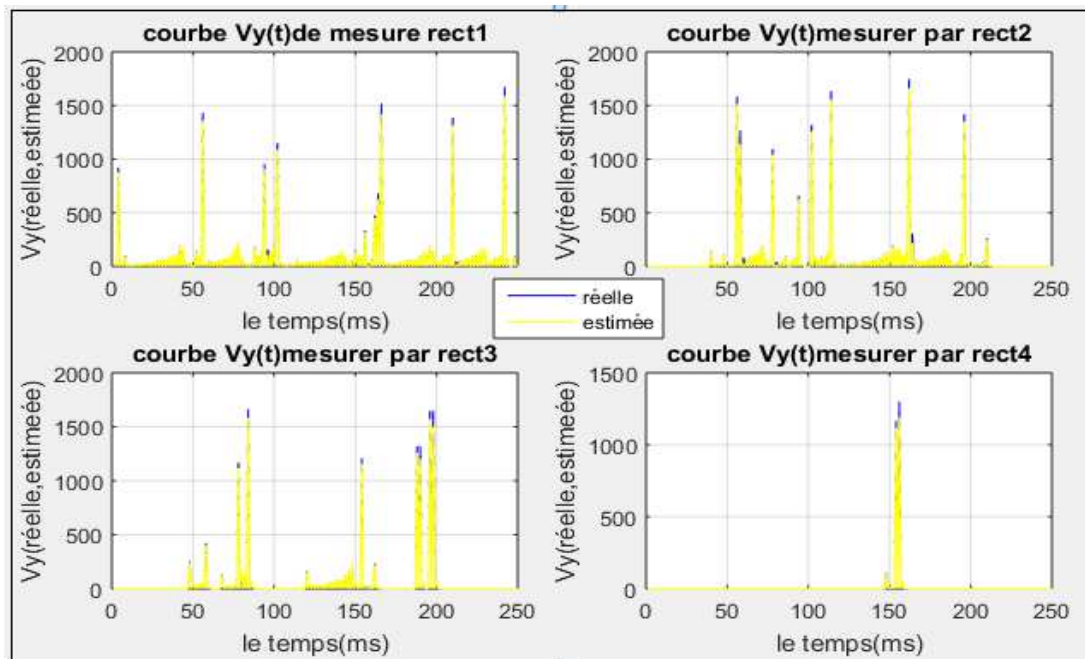
Pour assurer le bon fonctionnement du filtre, le bon choix des paramètres de la covariance des bruits de mesure et de processus sont donc nécessaires.

### I.4.1.4 Calcul de la vitesse des véhicules suivis

Maintenant, nous allons faire l'extraction des courbes de l'évolution de la vitesse réelle et estimée par kalman en fonction du temps. Donc les figures suivantes représentent ces courbes des véhicules suivis:



**Figure 3.(13.a) :** résultats des mesures de la vitesse  $V_x(t)$  réelle et estimée.



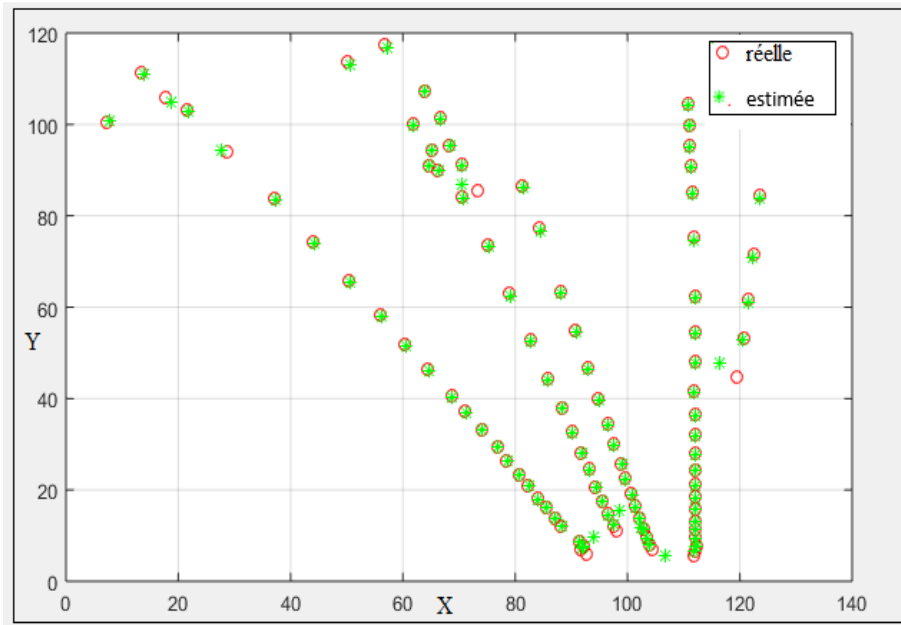
**Figure 3.(13.b) :** la vitesse  $V_y(t)$  réelle et estimée mesurée par les 4 rectangles.

- Discussion des résultats de la vitesse

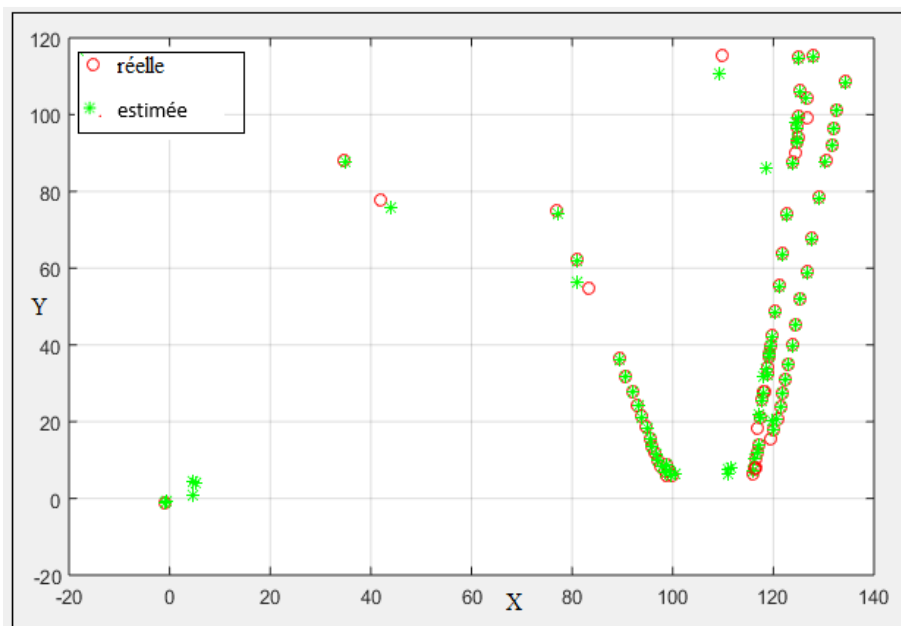
Avec les résultats de suivi comme indiqué dans les figures précédentes, nous pouvons constater que la méthode utilisée nous a permis d'avoir des bonnes valeurs de la vitesse que ce soit réelle ou bien estimée. Ce qui résulte d'avoir un bon suivi ainsi qu'une très bonne estimation, même avec des véhicules qui se déplacent rapidement.

### I.4.1.5 la trajectoire des véhicules suivis par les quatre rectangles

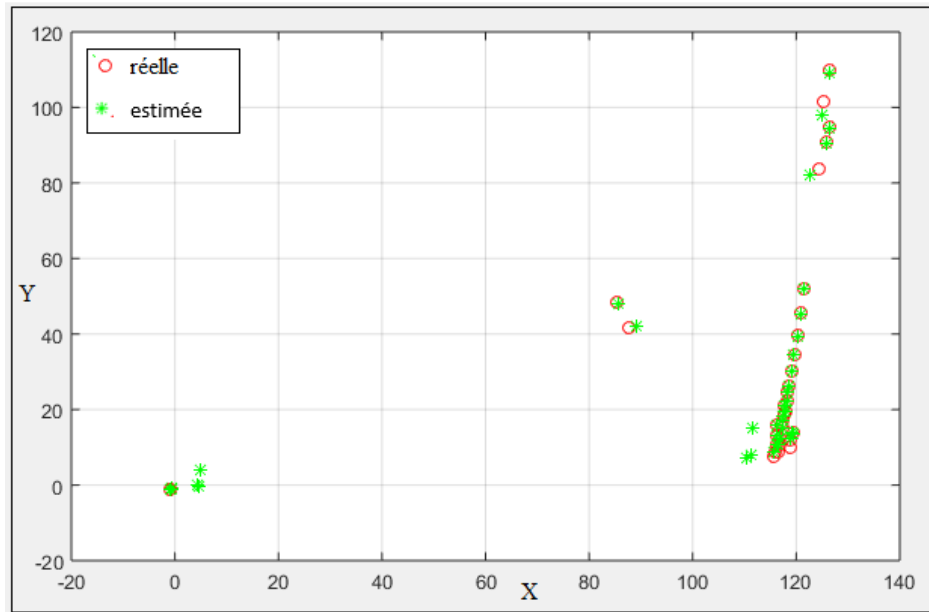
Nous utilisons la trajectoire des points des centres des masses des véhicules suivis afin de vérifier la méthode de l'estimation et le suivi basé sur le filtre de Kalman.



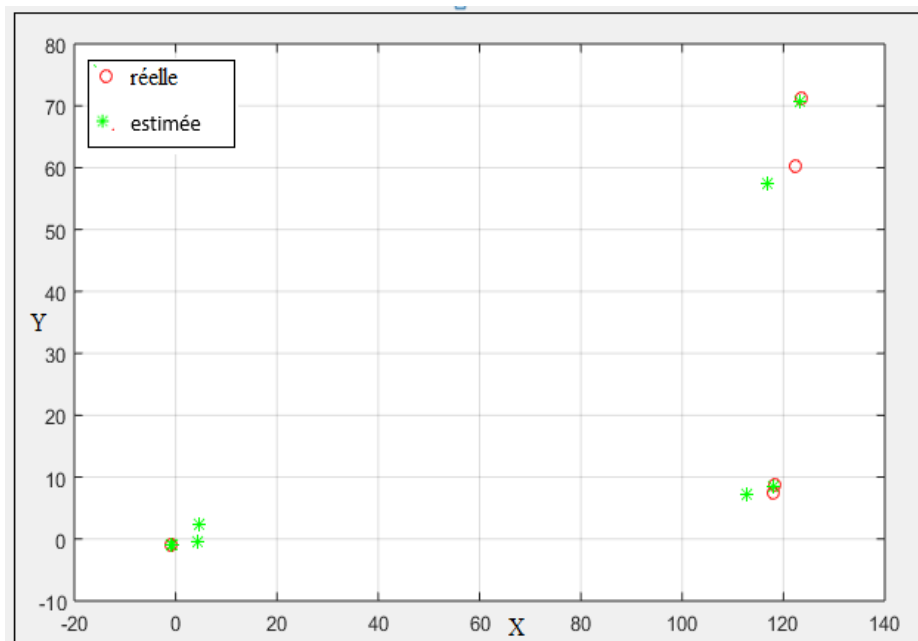
**Figure 3.(14.a) :**Trajectoire des véhicules suivis par le rectangle 1



**Figure 3.(14.b) :**Trajectoire des véhicules suivis par le rectangle 2



**Figure 3.(14.c) :** Trajectoire des véhicules suivis par le rectangle 3



**Figure 3.(14.d) :** Trajectoire des véhicules suivis par le rectangle 2.

### - Discussion des résultats des trajectoires

Quand une voiture entre dans la scène, elle doit être considérée comme un nouvel objet à suivre, donc le système lui affecte un nouveau rectangle pour qu'il puisse le suivre. Les résultats de la trajectoire des véhicules suivis par les 4 rectangles, montrent que la méthode de

suivie avec KF est capable de gérer et estimer correctement le maximums des points de déplacement des nouveaux objets entrées ou sorties de la scène à contrôler.

### ***II. 2<sup>ème</sup> Cas : En temps réel***

Maintenant place à des scènes en temps réel donc il s'agit d'une détection des objets en mouvement dans une scène capturée par la caméra web de l'ordinateur portable, la résolution de la vidéo est 320\*240. Pour faire tester la robustesse et la performance de notre système, nous avons choisi d'étudier deux exemples:

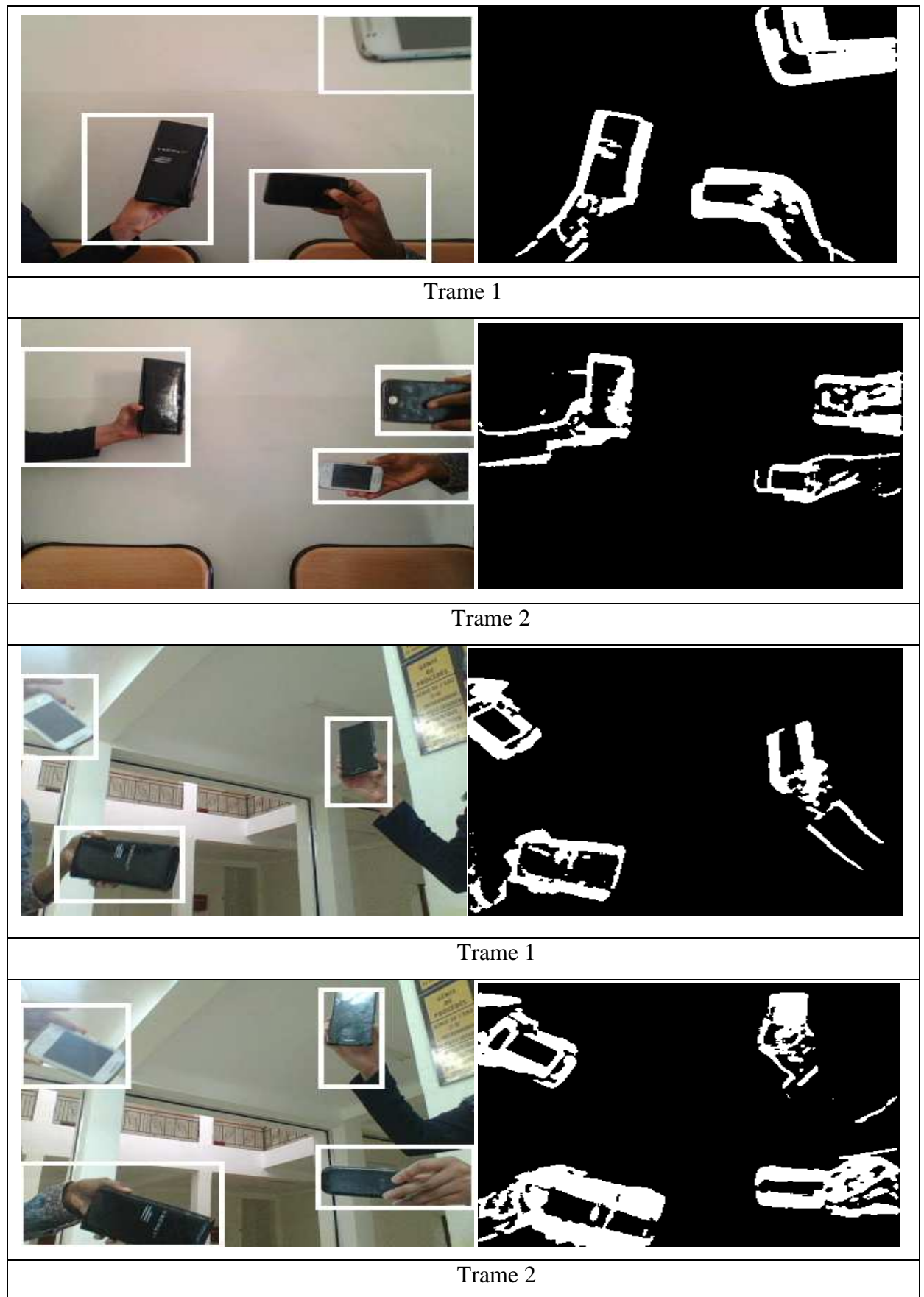
- exemple 1 : quatre objets (ex : portable, souris...etc.) mouvants avec les mains à l'intérieur des deux scènes différentes (dans l'arrière-plan et la luminosité). Ensuite, on va les suivre à l'aide d'un cadre.
- exemple 2 : suivi du corps humain en mouvement à l'intérieur.

#### **II.1. Résultats de la détection et suivi de mouvement en temps réel**

Afin d'évaluer les performances ainsi que les temps d'exécution de cet algorithme, nous testons plusieurs images capturées dans des scènes différentes. Les exemples suivants illustrent l'impact des différents cas existe dans cette approche sur l'efficacité de la technique.

##### **Exemple 1 : Des objets mouvant avec les mains**

Dans les figures ci-dessous quatre objets bien espacés les uns des autres, sans collision entre eux.



**Figure 3.15 :** Résultats de la détection et suivi des objets mouvant avec les mains.

Nous avons constaté des bons résultats de détection et bien séparation entre les 4 objets en mouvement, avec un très bon résultat de soustraction de l'image binaire.

- Ensuite, nous allons essayer de faire l'un de ces objets plus proche à la caméra que les autres, et nous le mouvons en face. Le résultat est représenté dans la figure suivante.



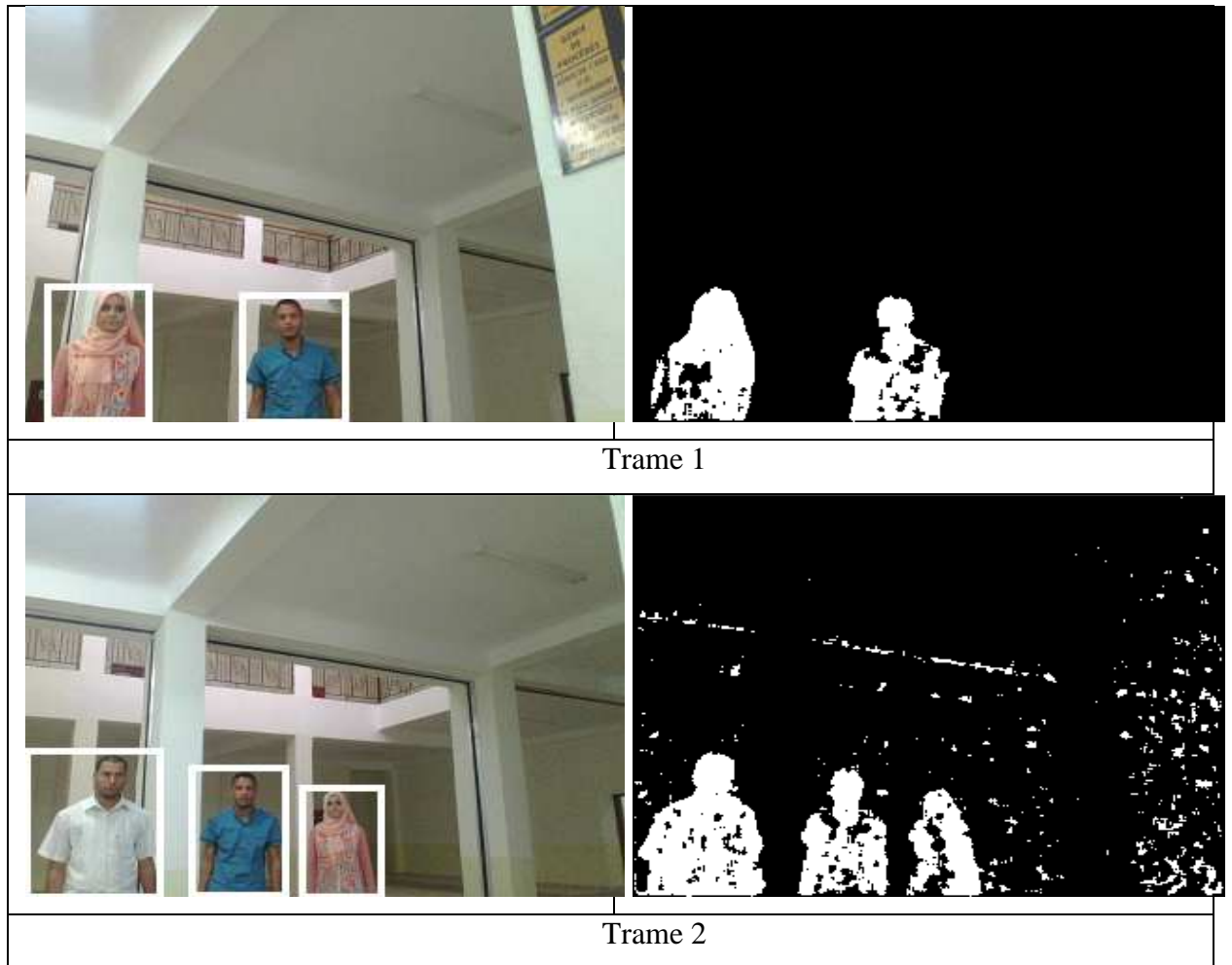
*Figure 3.16 : l'effet sur la détection lorsque l'objet s'approche du camera.*

On remarque que la qualité de détection est diminuée. Cependant, l'élément proche représente l'élément dominant dans la scène, même s'il est lent par rapport aux autres.

Donc notre système connaît des limites, comme par exemple dans le cas où les objets sont très proches l'un de l'autre (occlusion totale), on les suit comme étant un seul.

Exemple 02 : Suivi du corps humain en mouvement à l'intérieur (indoor).

Il est difficile de suivre le déplacement des corps humains. Cependant, nous avons fait quelques mesures pour le suivi des personnes.



**Figure 3.17 :** Résultats de la détection et suivi des corps humains.

Après les résultats des deux exemples, nous pouvons dire que :

- le système de détection de mouvement peut réagir face aux conditions non contrôlées de luminosité, la détection à l'intérieur de l'environnement ainsi qu'une rapidité d'exécution convenable pour la contrainte de la variation de la vitesse des objets.

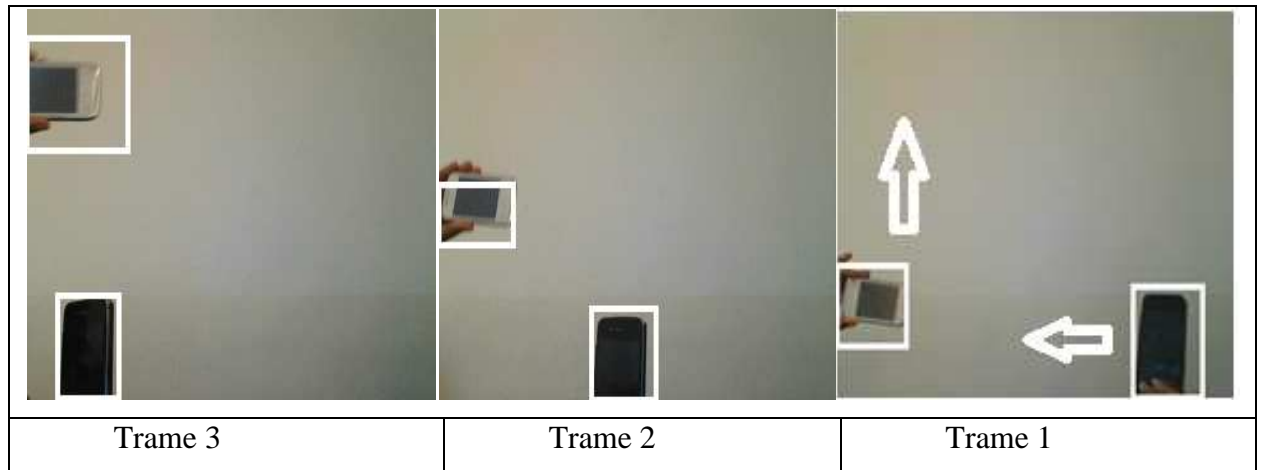
Certainement, notre système connaît des limites, donc les résultats de détection peuvent être influencés par des facteurs tels que :

- l'occlusion et surtout l'occlusion totale des objets, le problème de l'objet dominant ou l'objet plus proche, le mouvement très lent, l'homogénéité de la couleur des objets avec la couleur de l'arrière-plan et même la qualité de la caméra web utilisée.



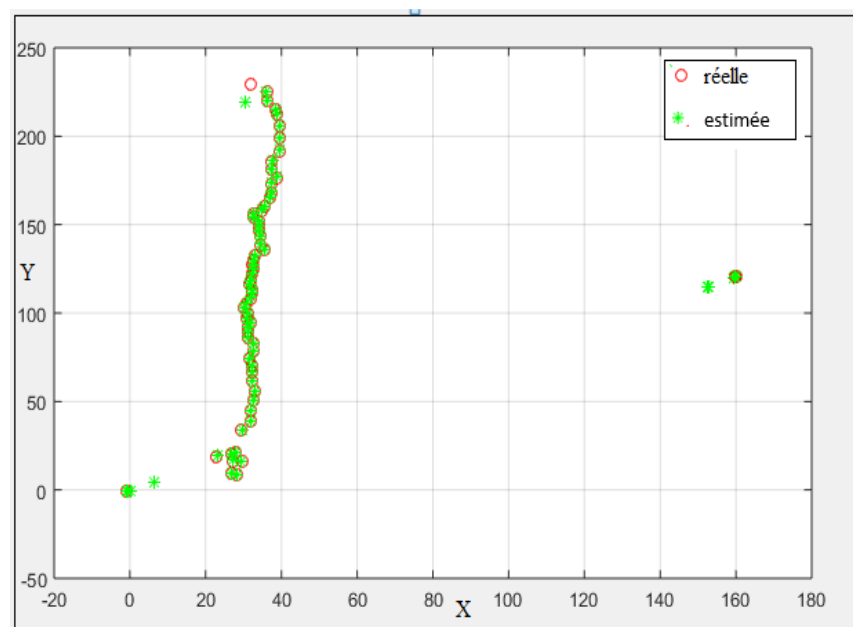
### II.2. Résultats de suivi des centres de masse

Maintenant on va faire déplacer deux objets pendant une durée de 10s. Ensuite, nous utilisons la trajectoire des points des centres des masses des deux objets afin de vérifier l'efficacité de la méthode de l'estimation et le suivi basé sur le filtre de Kalman.

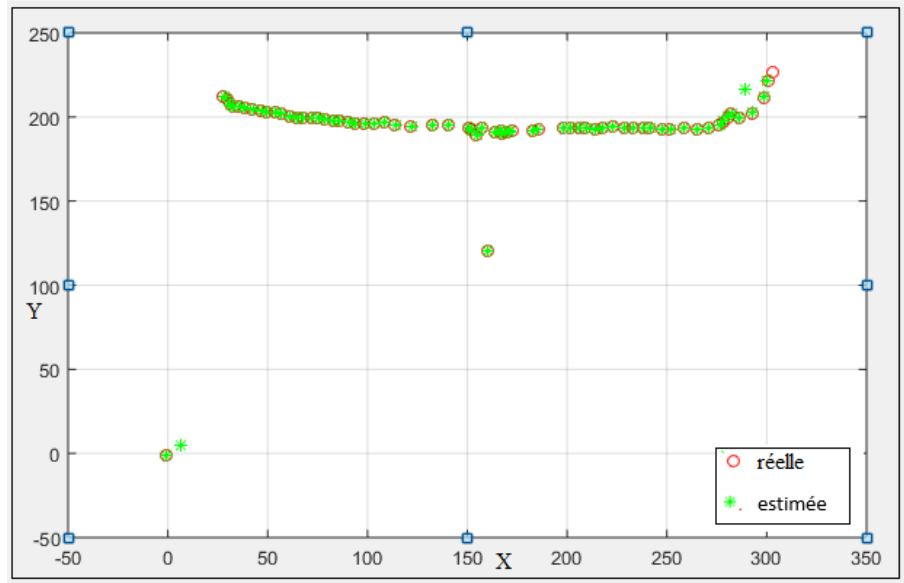


*Figure 3.18 : quelques trames qui représente le mouvement.*

Nous obtenons les résultats suivant :



*Figure 3.(19.a): La trajectoire obtenue de mouvement verticale de l'objet 1.*



**Figure 3.(19.b):** La trajectoire obtenue de mouvement horizontale de l'objet 2.

On a remarqué la superposition du maximum des points des centres de masse estimées par le filtre de kalman avec les points des positions réelles des deux objets aussi que la forme de la trajectoire est suivie la forme réelle de mouvement. Donc on peut dire que le filtre a bien estimé et bien suivi le mouvement des deux objets.

### *Conclusion*

Dans ce chapitre, nous avons abordés le problème de suivi de plusieurs objets en mouvement dans une séquence vidéo acquise à partir deux type de source : des vidéos enregistrées où bien des scènes captées en temps réel à partir une caméra web où USB.

L'objectif principal de notre travail est de réaliser une application qui assure la détection et le suivie des objets multiples en mouvement avec une caméra couleur fixe.

Nous décrivons aussi toutes les mesures d'évaluation et les variations des paramètres pour chaque étape des méthodes proposées, afin d'atteindre les paramètres optimale pour notre système. L'application développée est basé sur la soustraction de fond pour la détection de mouvement et le filtre de kalman pour l'estimation et le suivi de la position des centres de masse, la vitesse et même la trajectoire des objets suivis. Aussi nous remarquons que la performance du filtre est influencée par les valeurs des covariances des bruits de mesure et de processus.

On a obtenu des bons résultats de détection, d'estimation et de suivie simultanée des plusieurs objets en mouvement, sauf dans les cas des mouvements très lent où de l'occlusion des objets dans la scène. Aussi le temps de réponse est acceptable pour une application en temps réel.

# Conclusion générale

### *Conclusion générale*

L'objectif principal de ce travail est la conception d'un système qui permet de détecter et de suivre des objets dans un environnement dynamique, bien que beaucoup de travaux dans ce domaine aient été réalisés, le critère crucial à prendre en considération concerne la détection et l'estimation de la dynamique des objets ou biens des obstacles en mouvement c'est en temps réel. Le problème que nous nous sommes posé est de concevoir un système performant et robuste en tenant de compte toujours le facteur du temps.

Le développement ou bien la conception de ce système a été fondé sur deux étapes. En premier lieu conventionnel, il s'agit de proposer des techniques bien robustes pour la détection des objets en mouvement. En second, mettre en place un outil très performant pour l'estimation de la trajectoire des objets, le choix de l'application de filtre de kalman s'est imposé.

Finalement on peut dire qu'après plusieurs tests effectués sur le système que ce dernier peut être utile dans pas mal des domaines, exemple dans la robotique mobile (navigation, évitement d'obstacles) ou bien dans le domaine de la surveillance.

### *Perspectives*

En perspectives, on propose :

- De chercher une méthode unique pour la détection et le suivi d'objets mobiles pour les deux types de mouvement (lent et rapide).
- L'application des méthodes de détection et de suivi sur les séquences à fonds dynamiques (caméras mobiles).

# BIBLIOGRAPHIES

- [1]. N.KADACHE, "Analyse du mouvement et suivi de cibles mobiles dans les séquences vidéo", Thèse de doctorat, Université Elhadj Lakhder, Batna, aout 2008.
- [2]. M. Gelgon, "Segmentation spatio-temporelle et suivi dans une séquence d'images: application à la structuration et à l'indexation de vidéo", Thèse de doctorat, Université de Rennes I (France), Novembre 1998.
- [3]. V. Agnus, "Segmentation spatio-temporelle de séquences d'images par des opérateurs de morphologie mathématique", Thèse de Doctorat, LSIIT, Strasbourg, Octobre 2001.
- [4]. A. El maadi, " Vision infrarouge appliquée à la Vidéosurveillance extérieure automatisée reconnaissance, suivi, classification et détection d'événements", Thèse de doctorat, Université Laval, 2006.
- [5]. J. H. Duncan, and T. C. Chou, "Temporal edges: The detection of motion and the computation of optical flow", 1 in *Proc. IEEE 2<sup>nd</sup> Int. Conf. Computer Vision*, Florida, USA, Dec. 1988, pp.374-382.
- [6]. K. Yoshinari et M. Michihito, "A human motion estimation method us using 3-successive video frames", *Int. Conf. on Virtual Systems and Multimedia*, p. 135- 140, 1996.
- [7]. S-C.S. Cheung and C. Kamath. "Robust techniques for background subtraction in urban traffic video". In *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Nice, 2003.
- [8]. S. Cheung S et C. Kamath, "Robust techniques for background subtraction in urban traffic video", *Proceedings of the SPIE: Visual Communications and Image Processing*, Volume 5308, pp. 881-892, Janvier 2004.

- [9]. T. Malavika<sup>1</sup>, M. Poornima , "Moving object detection and velocity estimation using MATLAB", International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 1, January- 2013 ISSN: 2278-0181.
- [10]. A. LEMIEUX, "Système d'identification de personnes par vision numérique", thèse de doctorat, université LAVAL Québec, Décembre 2003.
- [11]. R. Rad, M. Jamzad, "Real time classification and tracking of multiple vehicles in highways", Pattern recognition letters, Vol. 26, Issue 10, pp. 1597-1607, 2005.
- [12]. <http://www.adcis.net/fr/Applications/ListeD'Exemples/SuiviDeMouvement.html#Example7>. Visité en mars 2016.
- [13]. Y.Dedeoğlu, "*Moving Object detection, Tracking and Classification for Smart Video Surveillance*". August, 2004.
- [14]. G.Welch et G. Bishop, "The Kalman filter" <http://www.cs.unc.edu/~welch/kalman/>.consulté en mars 2016.
- [15]. X.Wang, W.Wang and Y.Li, "A Multiple Object Tracking Method Using Kalman Filter", Proceedings of the 2010 IEEE International Conference on Information and Automation June 20 - 23, Harbin, China.
- [16]. Dora Luz Almanza-Ojeda, "Détection et suivi d'objets mobiles perçus depuis un capteur visuel embarqué", Thèse de doctorat, Université Paul Sabatier - Toulouse III, 2011.
- [17]. [www.ferdinandpiette.com](http://www.ferdinandpiette.com). Visité en Avril 2016.
- [18]. C.Tien Dung, "Vidéo Surveillance", Rapport final du Travail d'Intérêt Personnel Encadré (TIPE), Hanoi, Janvier 2007.

[19]. *Simulink library browser, MATLAB 2015a.*

## **LISTE DES ABBREVIATIONS**

<b>RGB</b>	camera ou espace couleur (Red, Green, Blue).
<b>BBox</b>	Boîte englobant (Bouding Box).
<b>DL</b>	Delay length.
<b>GUI</b>	Interface utilisateur graphique.
<b>KF</b>	filtre de kalman.
<b>Minarea</b>	Specify minimum blob area in pixels.
<b>Ns</b>	Neighborhood size.
<b>ThS</b>	Threshold scaling factor.



## Résumé

Les techniques de suivi et d'analyse d'images sont de mieux en mieux maîtrisées et leur utilisation est de plus en plus répandue. La vidéosurveillance est utilisée comme un des moyens les plus incontournables pour la sécurité et la surveillance de territoire, des biens ou des personnes. Malheureusement des développements techniques et scientifiques sont encore nécessaires afin de maîtriser le traitement afin d'en extraire les informations utiles. Pour cela, il faudra surmonter plusieurs contraintes.

L'objectif de notre travail est de détecter et suivre multiples objets en mouvement dans une scène filmée par caméra couleur fixe. Un algorithme de soustraction de fond est utilisé pour gérer une détection des véhicules et des personnes en mouvement. Ensuite, un algorithme de suivi de ce mouvement est assuré par le filtre de Kalman. Grâce à l'établissement d'un modèle de mouvement par filtre de Kalman avec les caractéristiques centre de masse et la zone d'objets en mouvement dans une seule scène, et c'est grâce à toutes ces informations qu'on peut avoir la dynamique des objets à surveiller ou bien des obstacles à éviter.

Les résultats montrent que l'algorithme proposé est efficace dans le suivi des objets mobiles multiples même dans des situations confuses.

**Mot clés :** détection, soustraction de fond, suivi, filtre de Kalman, objet dynamique, les obstacles, en temps réel.

## Abstract

Monitoring techniques and image analysis are increasingly well mastered and their use is becoming more widespread. Intelligent video surveillance is used as one of the most essential means for security and surveillance area, goods or people. Unfortunately technical and scientific developments are needed in order to control the processing, in order to extract useful information. To do so will require overcoming several constraints.

The aim of our work is to detect and track multiple moving objects in a scene filmed by fixed color camera. In order to solve the problem a feature algorithm is proposed using to manage detection of vehicles and people moving through the background subtraction algorithm and monitoring based on the Kalman filter. Through the establishment of a model of motion Kalman filter with the center of mass characteristics and the area of moving objects into a single scene, and for to all this information we can have dynamic objects to monitor or obstacles to avoid.

The results show that algorithm proposed achieves efficient tracking of multiple moving objects under the confusing situations.

**Key words:** detection, background subtraction, Tracking, Kalman filter, dynamic object, obstacles, real time.