

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique

Université Mohamed Sadik BENYAHIA - Jijel



Faculté des Sciences Exactes et Informatique
Département d'Informatique

Mémoire
de fin d'études pour l'obtention du diplôme
Master de Recherche en Informatique
Option : Intelligence Artificielle

Thème

*Etude comparative des méthodes de modélisation /
suppression de fond pour le suivi des objets*

Réalisé par :

Mr. Boumaza Antar



Encadré par :

Mr. Taffar Mokhtar

Promotion : 2016

Remerciements

02
02



*Avant tout, nous remercions Allah, le tout puissant,
Pour la force, la volonté, la santé et la patience qu'il nos
Donné pour accomplir ce travail.*

*Nous tenons à exprimer notre grand respect et notre gratitude
À notre encadreur Mr Taffar Mokhtar, pour nos avoir honoré
En acceptant de diriger notre mémoire, pour son encadrement de
Qualité, ses précieuses suggestions Scientifiques, sa présence
Encourageante, et sa patience tout Au long de ce travail.
Nos vifs remerciements aux membres de jury d'avoir bien voulu
Évaluer ce mémoire.*

*Nous exprimons notre profonde reconnaissance à tous les
Enseignants qui ont contribué à notre formation, depuis le
Primaire jusqu'à l'université, pour le savoir et les connaissances
Qu'ils nos ont inculqué.*

*Pour terminer, Merci pour tous ceux qui, par leurs remarques
Et leurs Conseils, ont contribué à la réalisation de ce travail.*





Dédicace

À ma très chère Famille,

À tous mes amis,

À mes collègues de Promotion 2016

À toutes celles et ceux qui ont beaucoup
compté pour moi.

..Je dédie ce travail.



Antar

Sommaire.....	
Introduction générale	1
Chapitre 1: Vidéosurveillance	
1.1. Introduction.....	3
1.2. Concepts généraux	3
1.2.1 Vidéo.....	3
1.2.2 Les frames.....	4
1.2.3 Objet	4
1.2.4 La détection de l'objet.....	4
1.3. La vidéosurveillance.....	6
1.4. Systèmes de vidéosurveillance.....	8
1.4.1. Les systèmes Analogiques.....	8
1.4.2. La vidéosurveillance sur IP.....	9
1.4.3. Les systèmes « Analogiques / IP ».....	10
1.5. Structure d'un système de vidéosurveillance.....	11
1.6. Domaines d'utilisations pour la vidéosurveillance.....	13
1.7. Problèmes lié à la vidéosurveillance.....	13
1.8. Les avantages et les inconvénients de la vidéosurveillance.....	14
1.8.1. Les avantages de la vidéosurveillance.....	14
1.8.1. Les inconvénients de la vidéosurveillance.....	14
1.9. Conclusion.....	14

Chapitre 2 : Les étapes des algorithmes de soustraction de fond

2.1. Introduction.....	16
2.2. Étapes principaux des algorithmes de soustraction de fond.....	16
2.2.1. Prétraitement.....	17
2.2.2. La modélisation d'arrière-plan.....	18
a) Non-récuratif.....	18
b) Récuratif.....	18
2.2.3. Détection de premier plan.....	18
2.2.4. Validation des données.....	19
2.3. Les algorithmes de soustraction de fond.....	19
2.3.1. Non-récuratif.....	19
2.3.1.1. Filtre temporel médian.....	19
2.3.2. Récuratif.....	19
2.3.2.1. Filtre médian approximation.....	19
2.3.2.2. Mixture de gaussiennes (MOG).....	20
2.4. Conclusion.....	20

Chapitre 3 : les algorithmes de soustraction de fond

3.1. Introduction.....	21
3.2. Algorithme de la différence de frame.....	21
3.3. Algorithme de Codebook.....	23
3.3.1. Fonctionnement de l'algorithme de codebook.....	24
3.3.1.1. Phase d'apprentissage.....	23
3.3.1.2. Phase de détection des objets mobiles	24

3.3.2. Construction de l'algorithme de codebook.....	24
3.3.3 Couleur et luminosité.....	26
3.3.3.1 Distorsion de couleur	26
3.3.3.2 Distorsion de luminosité	27
3.3.4 La détection de premier plan.....	27
3.3.5 Longueur d'exécution négative maximale	28
3.4 Algorithme de Codebook avec un seul mot de code.....	28
3.4.1. Fonctionnement de l'algorithme de codebook avec un seul mot de code.....	29
3.4.1.1 Phase modélisation de fond.....	29
3.4.1.2 Phase de détection des objets mobiles.....	29
3.4.2. Construction de l'algorithme de codebook avec un seul mot de code.....	29
3.4.3. Couleur et luminosité	30
3.4.3.1 La distorsion de couleur.....	30
3.4.3.2. Distorsion de luminosité.....	31
3.4.4. Détection de premier plan.....	31
3.5. Conclusion	32

Chapitre 4 : Expérimentation & Résultats

4.1. Introduction.....	33
4.2. Choix du logiciel	33
4.2.1 Avantage du python	33
4.2.2. Inconvénients du python	33
4.2.3. Open CV	34
4.3. Interface graphique d'utilisateur.....	34

4.3.1. Page d'accueil	34
4.3.2. Page principale « Application »	35
4.4. Résultats expérimentaux	37
4.4.1. Vérités terrains	37
4.4.2. Classification de pixel.....	37
4.4.3. Rappel et précision	37
4.4.3.1. Rappel.....	37
4.4.3.2. Précision.....	38
4.4.4. Résultat de rappel et précision sur une vidéo filmée.....	38
4.4.4.1. Résultats de rappel et précision pour la méthode codebook.....	39
4.4.4.2. Résultats de rappel et précision par la méthode de différence frame.	41
4.4.4.3. Résultat de rappel et précision par la méthode de codebook avec un seul mot de code.....	43
4.4.5. Les résultat de rappel et précision par une vidéo de base de données.....	45
4.4.5.1. Les résultat de rappel et précision par la méthode codebook	45
4.4.5.2. Résultats de rappel et précision pour la méthode de différence frame.....	47
4.4.5.3. Résultats de rappel et précision par la méthode de codebook avec un seul mot de code.....	49
4.4.6. Comparaison entre les méthodes.....	51
4.4.6.1. Rappel et précision	51
a) vidéo filmée	51
b) vidéo de base CVLAB.....	51
4.4.6.2. Comparaison sur le temps d'exécution	52

4.4.6.3. Qualité de suivi	53
4.5. Conclusion	53
Conclusion générale.....	54

Chapitre 1. Vidéosurveillance

Figure 1.1	Un flux de vidéo.....	3
Figure 1.2	Exemples d'une séquence d'image d'une vidéo.....	4
Figure 1.3	Un seul objet détecté dans la scène.....	5
Figure 1.4	Plusieurs objets détectés dans la scène.....	6
Figure 1.5	Exemple d'une caméra de surveillance.....	6
Figure 1.6	Exemple d'une caméra en haute définition.....	7
Figure 1.7	Les systèmes de vidéosurveillance intelligente.....	8
Figure 1.8	Les systèmes de vidéosurveillance analogiques.....	9
Figure 1.9	La vidéo sur IP (ou IP-Surveillance).....	10
Figure 1.10	Les Systèmes « Analogiques/IP ».....	11
Figure 1.11	Structure d'un système de vidéosurveillance.....	11
Figure 1.12	(a) image de la séquence vidéo, (b) Détection d'objet en mouvement.....	12
Figure 1.13	Exemple de suivre un objet.....	12
Figure 1.14	Exemple de reconnaissance d'objet.....	13

Chapitre 2. Etapes principales des algorithmes de soustraction de fond

Figure2.1	Les Étapes de l'algorithme de Soustraction fond	17
------------------	---	----

Chapitre 3. Méthodes de soustraction de fond

Figure 3.1	Méthode de la Méthode de différence de frame	22
-------------------	--	----

Chapitre 4. Expérimentation & Résultats

Figure 4.1	Page d'accueil de l'application.....	34
Figure 4.2	La page principale de l'application.....	35
Figure 4.3	La détection des objets pour la méthode de codebook. À (gauche) frame réelle . À (droite)frame traitée (frames d'une vidéo filmée)	39
Figure 4.4	Le rappel et précision de la méthode codebook (vidéo filmée)....	40

Figure 4.5	La détection des objets par la méthode de différence frame À (gauche) frame traitée . À (droite)frame réelle (frames d'une vidéo filmée).....	41
Figure 4.6	Le rappel et précision méthode la différence frame (vidéo filmée).	42
Figure 4.7	La détection des objets en mouvement par la méthode de différence frame avec un seuil =100.....	43
Figure 4.8	La détection des objets par la méthode de codebook avec un seul mot de code . À (gauche) frame traitée . À (droite)frame réelle(frames d'un vidéo filmée).	43
Figure 4.9	Le rappel et précision méthode de codebook avec un seul mot de code.(vidéo filmée).....	44
Figure 4.10	La détection des objets par la méthode de codebook. À (gauche) frame réelle . À (droite)frame traitée .(vidéo d'une base de donnée).....	45
Figure 4.11	Le rappel et précision méthode de codebook (vidéo de base de donnée).....	46
Figure 4.12	la détection des objets par la méthode de la différence frame. À (gauche) frame traitée . À (droite)frame réelle .(vidéo d'une base de donnée)..	47
Figure 4.13	Le rappel et précision méthode la différence frame (vidéo de base de donnée).....	48
Figure 4.14	La détection des objets par la méthode de codebook avec un seul mot de code . À (gauche) frame traitée .À(droite)frame réelle. (vidéo d'une base de donnée).....	49
Figure 4.15	Le rappel et précision méthode de codebook avec un seul mot de code.(vidéo de base de donnée).....	50

chapitre 4. Expérimentation & Résultats

Tableau 4.1	Tableau des résultats de l'algorithme de codebook avec un vidéo filmé.....	40
Tableau 4.2	Tableau des résultats de l'algorithme de différence frame avec une vidéo filmée	42
Tableau 4.3	Tableau des résultats de l'algorithme de codebook avec un seul mot de code un vidéo filmé	44
Tableau 4.4	Tableau des résultats de l'algorithme de codebook par une vidéo de base de donnée	46
Tableau 4.5	Tableau des résultats de l'algorithme de différence frame par une vidéo de base de donnée	48
Tableau 4.6	Tableau des résultats de l'algorithme de codebook avec un seul mot de code par une vidéo de base de donnée	50
Tableau 4.7	Tableau comparative entre les meilleurs rappel et précision des méthodes .(par une vidéo filmée).....	51
Tableau 4.8	Tableau comparative entre les meilleurs rappel et précision des méthodes .(par une vidéo de base de donnée).....	52
Tableau 4.9	Tableau comparative entre les temps d'exécutions.....	52

L'identification des objets en mouvement à partir d'une séquence vidéo est une tâche fondamentale et critique dans de nombreuses applications de vision par ordinateurs. L'approche des algorithmes de soustraction de fond identifie les objets mobiles de la partie d'une trame vidéo qui diffère d'un modèle d'arrière-plan. Il y a beaucoup de défis dans l'élaboration d'un bon algorithme de soustraction de fond; il doit être robuste contre les variations de l'éclairage, éviter de détecter des objets non-stationnaire dans le fond tels que des feuilles battantes, la pluie, la neige, et l'ombre projetée par des objets en mouvement, son modèle d'arrière-plan doit réagir rapidement aux changements d'arrière-plan, tels que le démarrage et l'arrêt des personnes. Nous étudions et comparons trois algorithmes de soustraction de fond pour détecter et suivre et les personnes en mouvement dans les séquences vidéo, ces méthodes sont la différence frame ,codebook et une méthode proposée codebook avec un seul mot de code. Nous effectuons plusieurs tests et nous présentons quelques résultats expérimentaux effectués sur des vidéos filmées et des vidéos de base de données. La méthode proposée codebook avec un seul mot de code et codebook présentent de bonnes performances par rapport à la méthode de différence frame.

Mot clés : Codebook, la détection et suivi des objet, la vidéosurveillance, vision par ordinateur, la différence frame.

Identifying moving objects from a video sequence is a fundamental and critical task in many computer-vision applications. Approach background subtraction, identifies moving objects from the portion of a video frame that differs significantly from a background model. There are many challenges in developing a good background subtraction algorithm. it must be robust against changes in illumination, it should avoid detecting non-stationary background objects such as swinging leaves, rain, snow, and shadow cast by moving objects, its internal background model should react quickly to changes in background such as starting and stopping of people. we study and compare three background subtraction algorithms to detect and track moving people and in the movies, these methods are the difference frame, codebook and a codebook proposed method with only one code word. We perform several tests and we present some experimental results performed on filmed videos and basic video data. The method proposed codebook with a single codeword codebook and have good performance against the frame difference method.

keywords : Codebook, detection tracking object, video surveillance, computer vision, the difference frame.

المخلص

تحديد الأجسام المتحركة من خلال سلسلة الفيديو هو مهمة أساسية وحاسمة في العديد من تطبيقات الرؤية بالكمبيوتر نعتد على طريقة طرح الخلفية لتحديد الأجسام المتحركة أثناء تشغيل فيديو. نهج طرح الخلفيات يحدد لنا الأجسام المتحركة هناك العديد من التحديات التي يجب أن نتخطاها أي خوارزمية طرح الخلفية. أولاً، يجب أن يكون هناك تصدي ضد التغيرات في الإضاءة. ثانياً، يجب تجنب الكشف عن الأشياء غير ثابتة في الخلفية مثل أوراق الشجر التي تتأرجح ، المطر، الثلج، والظل الساقط من الأجسام المتحركة. وأخيراً، يجب على نموذج الخلفية أن تستجيب بسرعة للتغيرات ، مثل تحرك ووقف الناس. سوف نقوم بدراسة ومقارنة ثلاثة خوارزميات خلفية الطرح لكشف متابعة ونقل الناس خلال مدة المراقبة ، وهذه الخوارزميات هي خوارزمية طرح الصور ، خوارزمية الكتاب، و خوارزمية مقترحة و هي الكتاب مع كلمة السر واحدة فقط في الخلفية. أجرينا عدة اختبارات باستعمال فيديو مصور و فيديو محمل من قاعدة البيانات، خوارزمية الكتاب و الكتاب مع كلمة السر واحدة فقط في الخلفية أعطونا نتائج جيدة مقارنة بخوارزمية طرح الصور.

الكلمات المفتاحية : خوارزمية الكتاب ،كشف وتتبع الأشياء،المراقبة بالفيديو، الرؤية بالكمبيوتر، خوارزمية طرح الخلفيات .

Introduction générale

Introduction générale

Le concept de soustraction d'arrière-plan (en anglais "Background Subtraction" : BGS) regroupe de multiples méthodes permettant une détection efficace des objets mobiles dans une séquence d'images. Nous les retrouvons dans des applications liées aux systèmes de transports intelligents comme par exemple la détection d'obstacle pour la conduite autonome, les applications surveiller les biens et les personnes dans les zones publiques ou isolées, les applications surveiller la chute des gens âgés dans les maisons de retraites etc ...

L'idée de base du concept de BGS consiste à déterminer un modèle de l'arrière-plan par apprentissage. Ce modèle correspond aux régions fixes de l'images (sans objet mobile). La différence entre ce modèle d'arrière-plan et l'image observée est ensuite effectuée afin de faire ressortir les éléments mobiles. Plusieurs méthodes de BGS ont été proposées depuis quelques années. Les méthodes les plus simples consistent en une distribution unimodale mais souffrent de mauvaises performances dans le cas d'arrière plans dynamiques (oscillations des feuillages d'arbres par exemple) et de forts changements d'illumination de la scène, causant ainsi beaucoup de fausses détections.

Même si un grand nombre d'algorithmes de soustraction d'arrière-plan ont été proposés dans la littérature, le problème de l'identification des objets en mouvement dans l'environnement est complexe . Il y a plusieurs problèmes qu'un bon algorithme de soustraction de fond doit résoudre correctement :

- Le mouvement d'objets tel que les branches d'arbre est assez courant. Le but est de détecter ces objets comme appartenant au premier plan de l'image ,ce mouvement est de façon générale assez répétitif. Les méthodes de suivi doivent donc être capable d'apprendre ces mouvement comme des objets de premier plan et pas comme des objets d'arrière plan .
- Les scènes sombres ont des niveaux de couleur assez faibles, les différences entre les couleurs sont moins marquées. Il peut donc être plus difficile de détecter les objets
- Le changement de luminosité est assez courant quand ceci peut être lié à un passage de nuage. Les couleurs sur l'image varient donc et l'algorithme peut penser qu'il s'agit d'un objet. L'idéal serait d'être assez peu sensible au changement de luminosité, ou d'être capable de s'adapter rapidement pour limiter les mauvaises détections.

➤ Les objets ayant une couleur proche du fond peuvent être plus difficilement détectés étant donné que les algorithmes se basent sur la couleur du pixel pour identifier les objets de premier plan.

➤ Les ombres sont des zones qui suivent les objets avec des changements de valeur de pixel par rapport au fond. Il est donc assez logique que les algorithmes puissent classer par erreur ces zones comme premier plan.

Ce mémoire est constitué de quatre chapitres :

Le premier chapitre parle de la vidéosurveillance où nous parlerons de la détection des objets puis des systèmes de la vidéosurveillance qui utilisent des algorithmes de soustraction de fond pour suivre les objets, puis nous parlerons des problèmes rencontrés dans ce domaine.

Le deuxième chapitre est dédié au principe général des algorithmes de soustraction de fond qui généralement chaque algorithme de soustraction de fond suit un processus, puis nous parlerons de quelques algorithmes de soustraction de fond.

Le troisième chapitre nous discuterons en détail de notre travail et en particulier des trois algorithmes de soustraction de fond : codebook, différence frame et une méthode proposée est codebook avec un seul mot de code qui a un fond statique.

Le quatrième chapitre conclut ce mémoire en présentant l'ensemble des résultats expérimentaux obtenus dans notre application.

Chapitre 1 : Vidéosurveillance.

1.1. Introduction

Dans les dernières années le monde a connu plusieurs actes criminels et événements terroristes, ces événements et ces actes ont causés par des objets suspects, donc la vision et la capacité d'analyser ces objets s'avère être d'une importance capitale. Cela a incité à faire la sécurité une priorité ce qui a permis le développement des systèmes de vidéosurveillance [1].

La vidéosurveillance consiste à placer des caméras de surveillance dans un lieu public ou privé et de recevoir le flux vidéo sur un PC localement ou à distance en vue d'augmenter le niveau de sécurité. Les causes de l'installation de systèmes de vidéosurveillance sont diverses, toutefois la sécurité publique ainsi que la protection des biens mobiliers ou immobiliers font office d'éléments phares dans la justification de la vidéosurveillance. L'industrie de la vidéosurveillance englobe aujourd'hui toute une variété de systèmes et d'équipements de surveillance et de protection des personnes et des biens. Une vidéosurveillance performante c'est une solution contre le vol à l'étalage, en cas de cambriolage et une certaine tranquillité les nuits de garde [13].

1.2. Concepts généraux

1.2.1. Vidéo

Nous considérons un flux de vidéo comme une suite d'images de deux dimensions. La résolution de la vidéo, exprimée en nombre de pixels, définit la dimension de ces images (Figure 1.1) [1].

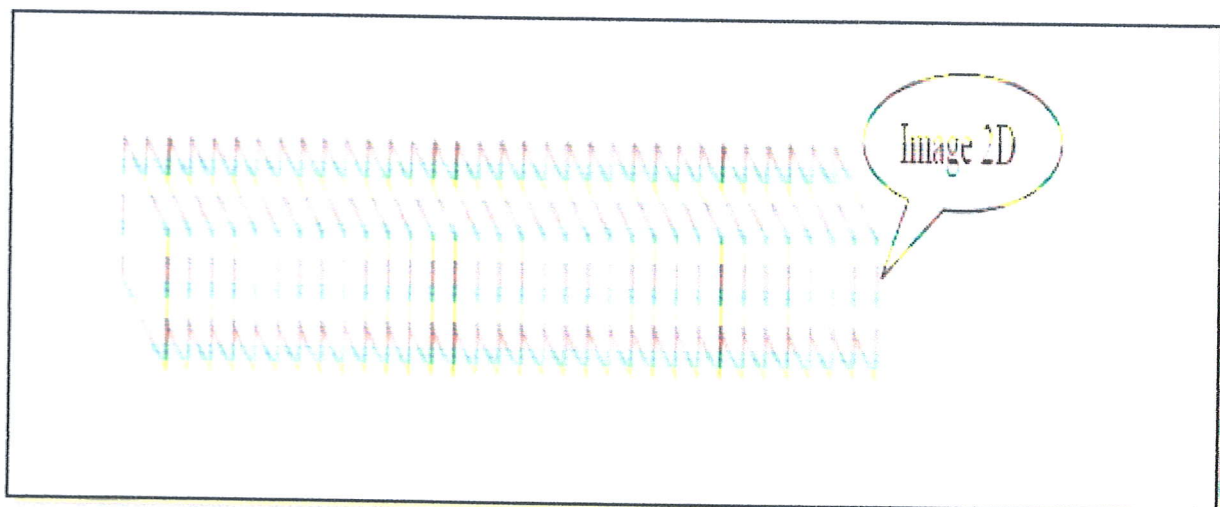


Figure 1.1 : Un flux de vidéo [1].

1.2.2. Frames

Les frames sont les images composantes la vidéo (séquences d'images (Figure 1.2) [1].



Figure 1.2 : Exemples d'une séquence d'image d'une vidéo [14].

1.2.3. Objet

Les objets physiques sont les objets du monde réel qui apparaissent dans les scènes observées par les caméras. Ils se décomposés en deux types, les objets de contexte et les objets mobiles [1].

- Les objets de contexte : sont des objets physiques qui sont habituellement statiques (comme les murs). Dans le cas dynamique (leurs mouvements), ils peuvent être prédits par les informations contextuelles (comme les chaises, les portes).
- Les objets mobiles : sont des objets physiques qui peuvent être perçus dans les scènes par leurs mouvements. Il est cependant difficile de prédire leurs mouvements (les personnes et les véhicules).

Les objets dans les vidéos sont des objets mobiles. Ils sont habituellement détectés et suivis dans plusieurs frames [1].

1.2.4. Détection de l'objet

La détection de l'objet comprend la détection d'objets et la reconnaissance de formes dans le cadre d'une séquence vidéo. Un mécanisme de détection d'objet est nécessaire dans

n'importe quel procédé de suivi soit dans chaque trame ou lorsque l'objet apparaît en premier dans la vidéo [1].

La détection d'objet est l'identification des objets dans le flux vidéo, et le regroupement des pixels de ces objets. Il existe de nombreuses méthodes et technique qui ont été proposées et développées [1].

➤ Un seul objet dans la scène : En prend comme exemple la détection d'un objet présent dans une scène du mouvement par ce programme (Figure 1.3) [1].

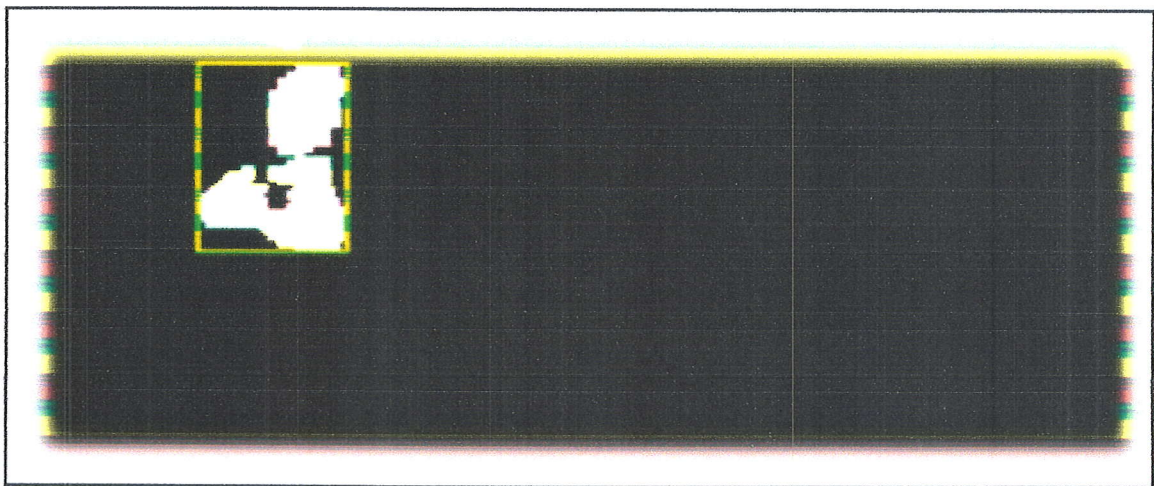


Figure 1.3 : Un seul objet détecté dans la scène [1].

➤ Plusieurs objets dans la scène : Ce programme peut détecter plusieurs objets retrouvés aux même temps dans une scène du mouvement (Figure 1.4) [1].

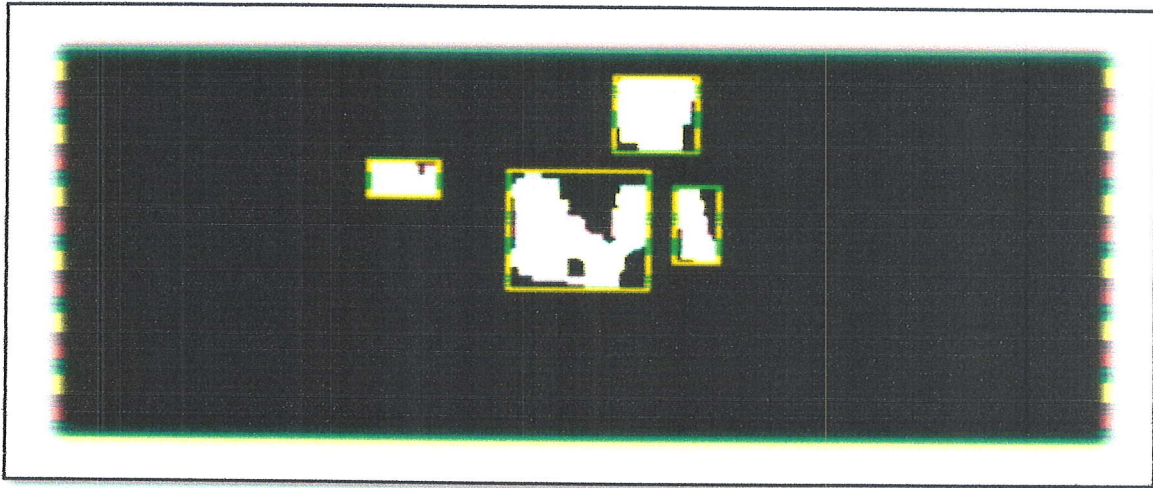


Figure 1.4 : Plusieurs objets détectés dans la scène [1].

1.3. Vidéosurveillance

La vidéosurveillance c'est un système de caméras permettant de surveiller à distance un espace privé ou public. Des images sont enregistrées avec ce système et sont par la suite visionnées et sauvegardées. Il se développe de plus en plus dans nos rues, dans les entreprises et chez certains particuliers [15](Figure 1.5).

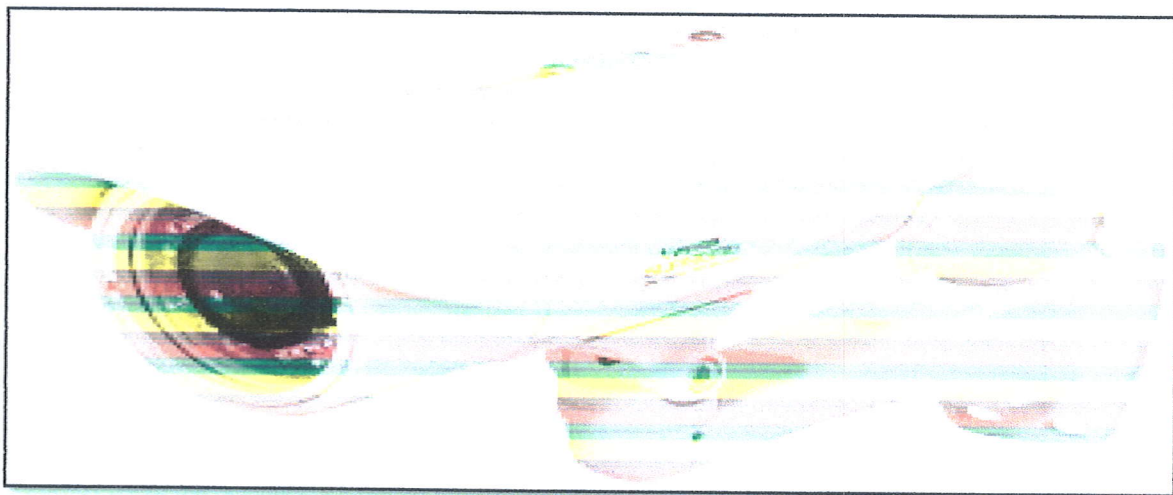


Figure 1.5 : Exemple d'une caméra de surveillance [16]

L'avenir de la vidéosurveillance passe par la haute définition, ce type offre des performances incomparables à celles des systèmes à la définition standard. La vidéosurveillance en haute définition de qualité six (06) fois supérieure aux solutions standards (PAL, NTSC) pour des images parfaites. **Le format haute définition** est idéal pour surveiller des zones spécifiques où le moindre détail compte en direct ou à posteriori (identification de visage, de plaque d'immatriculation, d'objet). La résolution supérieure apporte une qualité d'image incontestable et un niveau de détails élevé pour une meilleure analyse des événements (Figure 1.6) [17].



Figure 1.6 : Exemple d'une caméra en haute définition [17].

La vidéosurveillance intelligente (l'analytique vidéo), est une technologie qui permet au moyen de logiciels, d'identifier automatiquement, dans des séquences vidéo, des objets, des comportements ou des attitudes spécifiques. Elle transforme la vidéo en données qui seront transmises ou archivées pour permettre au système de vidéosurveillance d'agir en conséquence [1].

Les systèmes de vidéosurveillance intelligente utilisent des algorithmes mathématiques pour détecter des objets en mouvements dans l'image et filtrer les mouvements non pertinents (Figure 1.7).



Figure 1.7 : Les systèmes de vidéosurveillance intelligente [18].

1.4. systèmes de vidéosurveillance

Les systèmes de vidéosurveillance obéissent presque tous à un même schéma de base englobent tout un ensemble d'équipement orientés sécurité et peuvent être répartis en deux grands groupes [13]:

- Les systèmes de vidéosurveillance analogique.
- Les systèmes de vidéosurveillance IP.
- Les systèmes « hybrides ».

1.4.1. Systèmes Analogiques

Les systèmes de vidéosurveillance analogiques utilisent des caméras vidéo analogiques avec sortie coaxiale. Pour visualiser les images dans de tel systèmes, on connecte généralement un moniteur directement sur la camera via sa sortie coaxiale. Sur cette même sortie un magnétoscope peut être branchés pour enregistrer les images. Les caméras analogiques sont des caméras équipées d'une connexion coaxiale et qui diffusent tel quel les images qu'elles captent (Figure 1.8) [13].

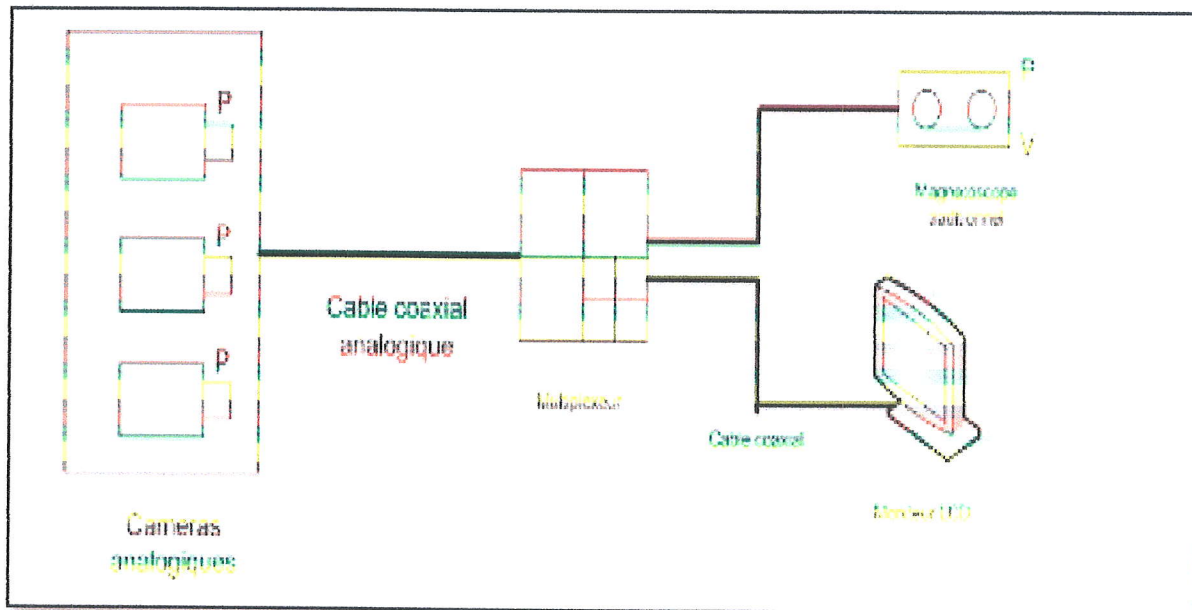


Figure 1.8 : Les systèmes de vidéosurveillance analogiques [13].

1.4.2. Vidéosurveillance sur IP

La vidéo sur IP (ou IP-Surveillance) est un système permettant à ses utilisateurs de visualiser et d'enregistrer des images vidéo via un réseau IP (LAN/WAN/Internet)

À la différence des systèmes analogiques, la vidéo sur IP utilise le réseau informatique plutôt qu'un système de câblage point-à-point pour transmettre les informations. Le terme vidéo sur IP englobe à la fois les sources vidéo et audio véhiculées par le système. Dans une application de vidéo sur IP, les flux d'images vidéo numériques peuvent être transférés n'importe où dans le monde via un réseau IP sécurisé, câblé ou sans fil, permettant une visualisation et un enregistrement vidéo en tout point du réseau.

La vidéo sur IP permet aux utilisateurs d'obtenir à tout instant et en tout lieu des informations sur une opération en cours, et de la suivre en temps réel. Cette caractéristique en fait une technologie idéale pour assurer le contrôle des installations, des personnes et des locaux, sur place ou à distance comme le contrôle de la circulation, le contrôle des lignes de production ou le contrôle des points de vente (Figure 1.9) [1].

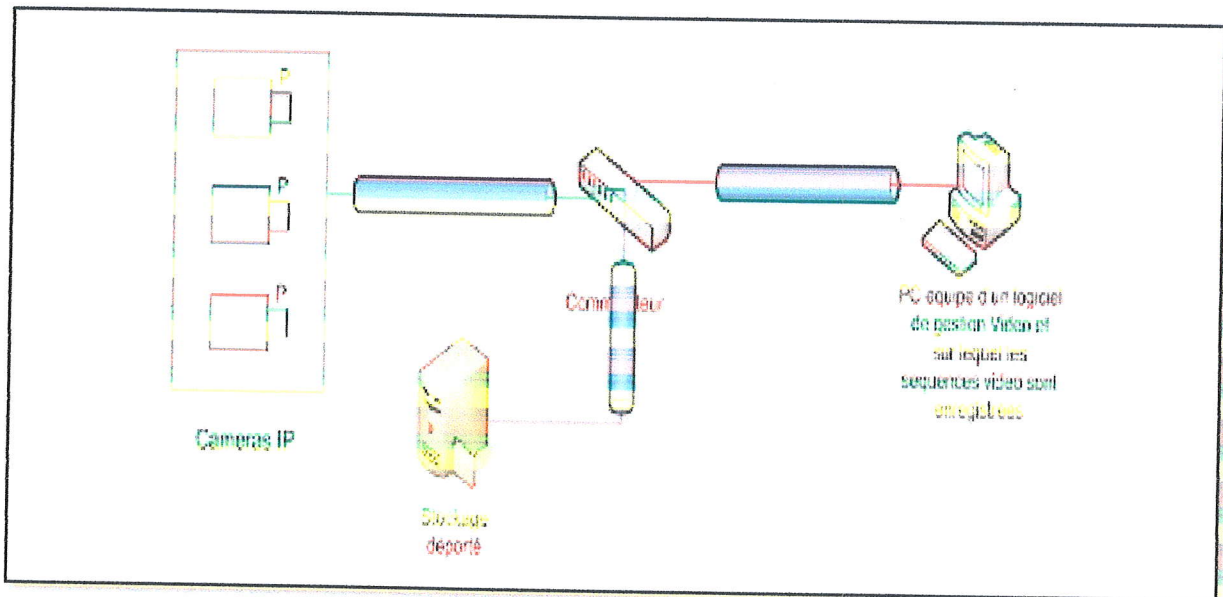


Figure 1.9 : La vidéo sur IP (ou IP-Surveillance) [13].

1.4.3. Systèmes « Analogiques/IP »

Les systèmes mentionnés ici sont des systèmes réunissant des systèmes analogiques et des réseaux IP. Ceci permet par exemple d'étendre un système analogique afin de le rendre plus efficace, de l'ouvrir sur l'extérieur. Ils sont généralement caractérisés par la présence d'un serveur vidéo. Un serveur vidéo permet de migrer vers un système de vidéo sur IP en conservant les installations analogiques existantes et en leur octroyant de nouvelles fonctionnalités. Il permet par ailleurs d'éliminer certains équipements spécifiques (câbles coaxiaux, moniteurs ou enregistreurs numériques). Un serveur vidéo possède en général de un à quatre ports analogiques pour la connexion de caméras analogiques, et un port Ethernet pour la connexion au réseau. Tout comme les caméras réseau, un serveur vidéo possède un serveur web intégré, et un système d'exploitation permettant la conversion des flux entrants en images vidéo numériques, ainsi que leur transmission et leur enregistrement sur le réseau informatique où elles pourront être visualisées et consultées plus facilement (Figure 1.10) [13].

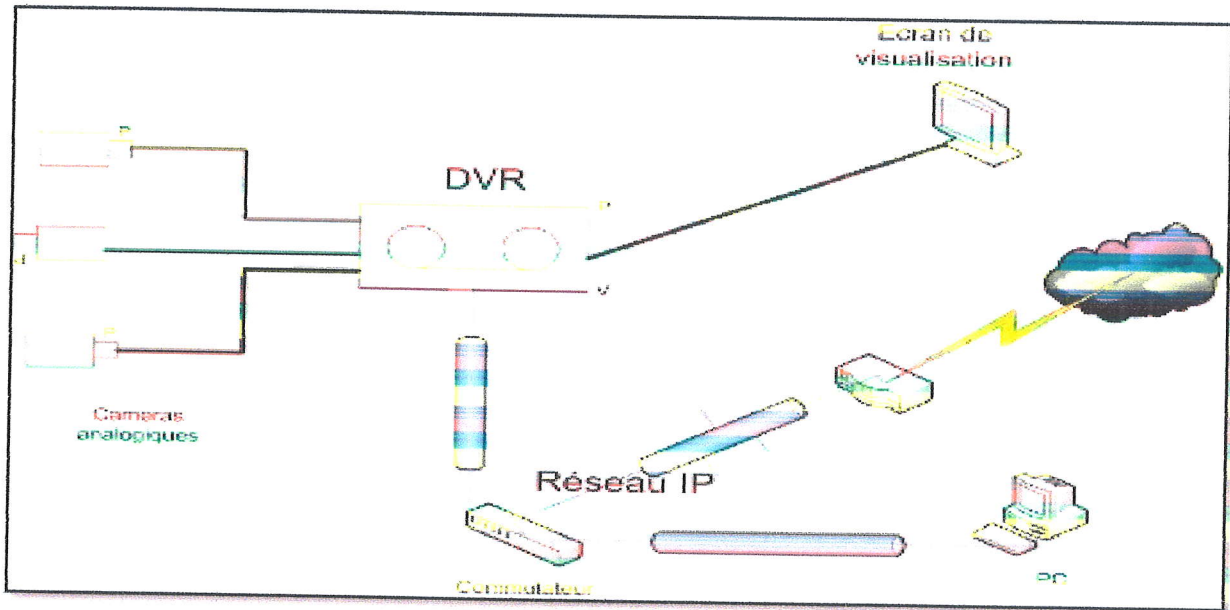


Figure 1.10 : Les Systèmes « Analogiques/IP » [13].

1.5. Structure d'un système de vidéosurveillance

Un système de vidéosurveillance est décomposé en trois modules généraux, détection, suivi et la reconnaissance (Figure 1.11) [1]

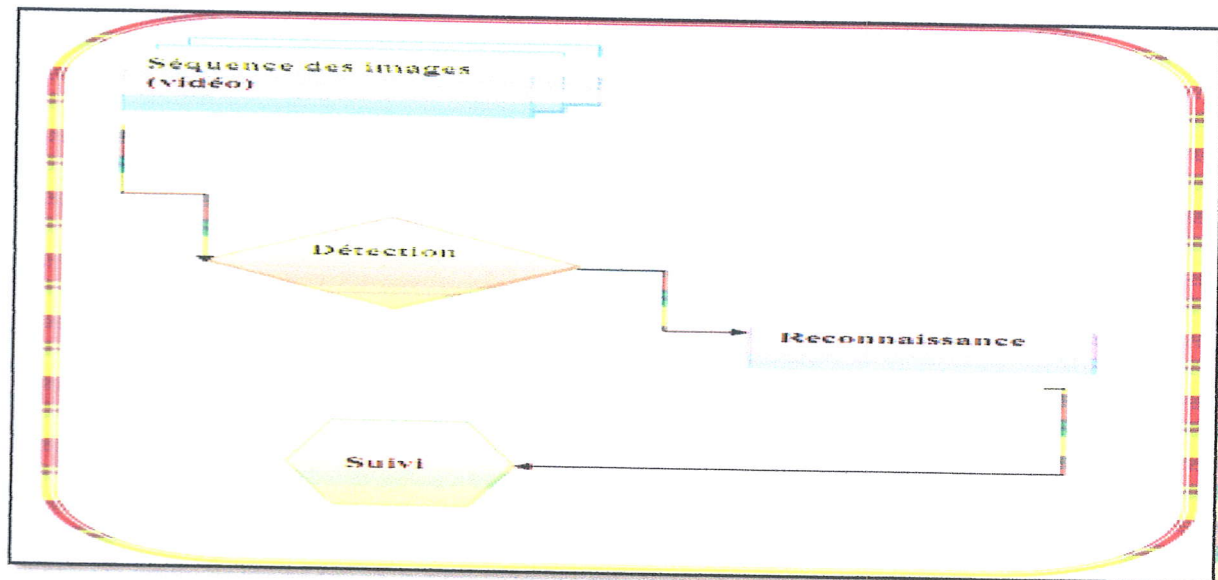


Figure 1.11 : Structure d'un système de vidéosurveillance [1].

➤ **Détection** : A chaque nouvelle image de la séquence vidéo, la personne en mouvement doit être détectée et segmentée en régions (Figure 1.12) [1].

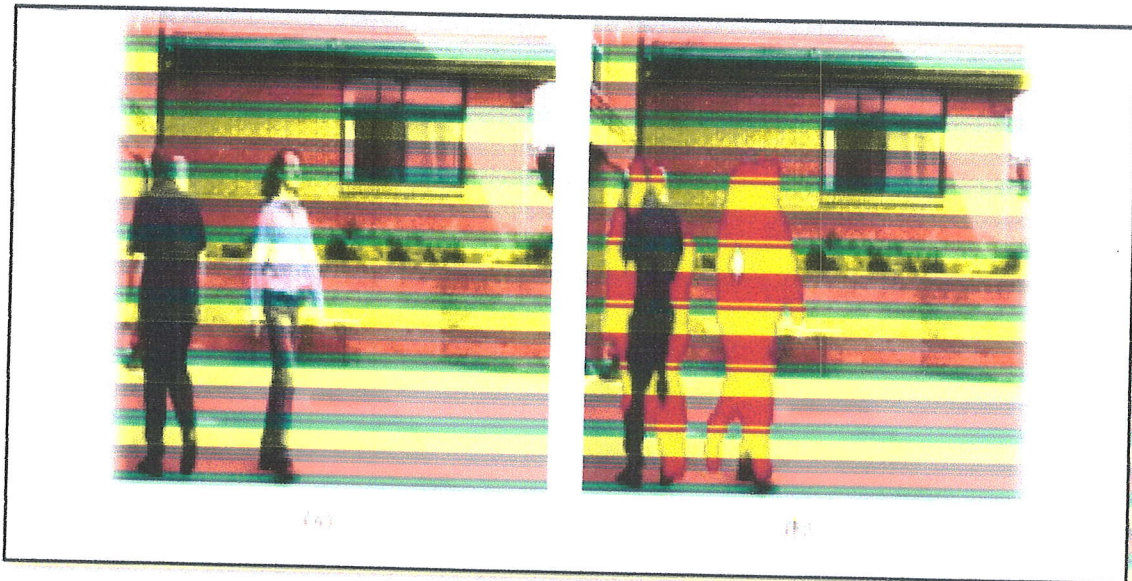


Figure 1.12 : (a) image de la séquence vidéo, (b) Détection d'objet en mouvement [1].

➤ **Suivi** : Il s'agit alors de suivre une personne ou un chemin tout au long de la séquence vidéo (Figure 1.13) [1].

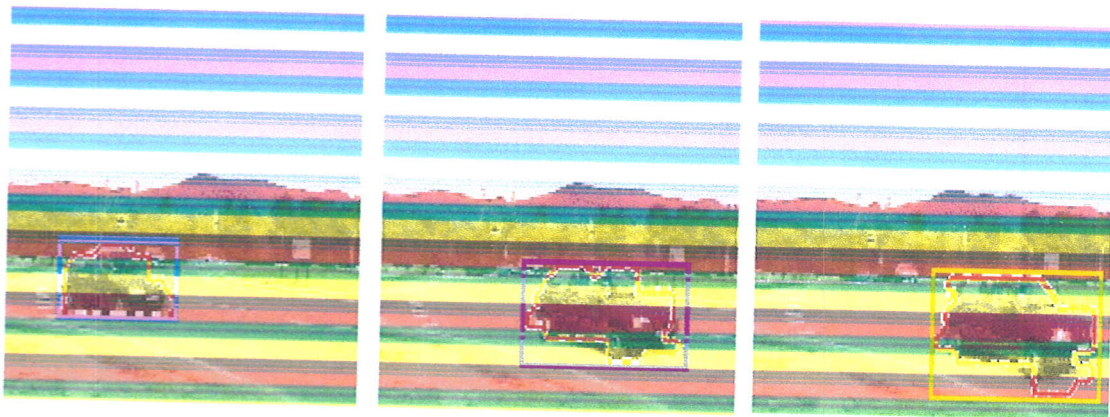


Figure 1.13 : Exemple de suivre un objet [19].

➤ **Reconnaissance** : La reconnaissance de comportement permet de détecter un objet suspect et fournit en sortie un résultat correspondant à une prise de décision à savoir si le comportement est normal ou anormal (Figure 1.14) [2].

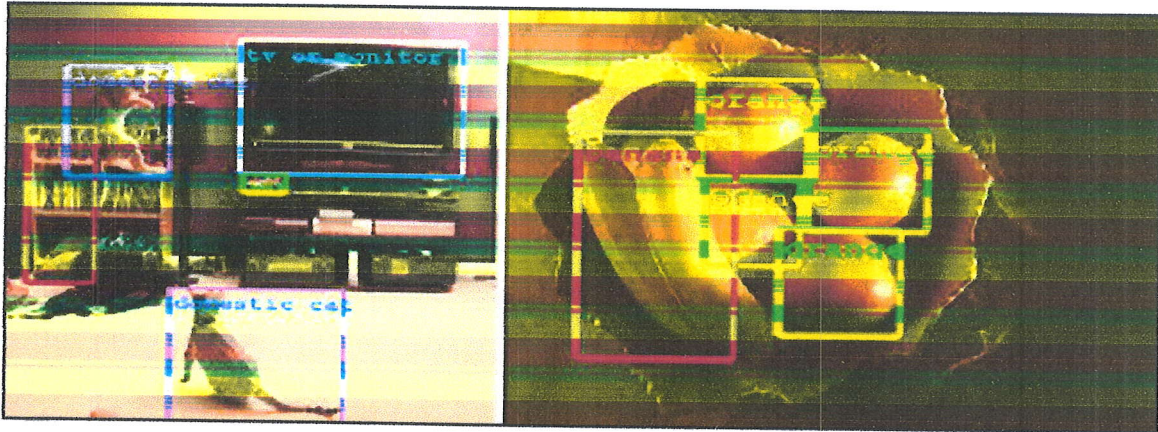


Figure 1.14 : Exemple de reconnaissance d'objet [20].

1.6. Domaines d'utilisation pour la vidéosurveillance

La vidéosurveillance, est un champ de plusieurs d'applications [1].

- L'industrie : Salle de contrôle, Zones de stockage.
- Le transport privé ou collectif.
- Le commerce et la distribution.
- Les administrations et les services publics : les banques.
- Les lieux publics : Cabinets médicaux et cliniques.
- L'enseignement : Les écoles et les établissements d'enseignement.
- Les loisirs : parc d'attraction, cinéma, les clubs.

1.7. Problèmes liés à la vidéosurveillance

Les systèmes de la vidéosurveillance ont plusieurs problèmes les importants sont :

➤ Les systèmes de vidéosurveillance intelligente utilisent des algorithmes mathématiques pour détecter des objets en mouvements . Il n'est pas facile de développer des algorithmes qui détectent en tout moment de la journée et ainsi qui s'adaptent aux changements du temps et aux variations de luminosité .par exemple presque tout les algorithmes ont des problèmes de détection en présence des zones d'ombre .

➤ Un autre problème important est lié à l'atteinte de la vie privée puisque nous sommes constamment surveillés. Les familles ne se sentent pas à l'aise en sachant qu'une caméra les surveille en permanence .Cela pose une restriction des libertés [21]

➤ La mise en place de caméra qui est très coûteuse puisque si l'on souhaite un système sophistiqué de vidéo surveillance cela représentera un investissement considérable et à cela s'ajoutera la maintenance du système qui devra être régulière. De plus, on ne peut pas surveiller toutes les rues d'une ville puisque pour le faire il faudrait un nombre important de caméras. Enfin, cela enlève de l'emploi des vigiles dans les rues .

1.8. Avantages et les inconvénients de la vidéosurveillance

Comme toute les applications et les différentes options utiliser dans notre vue, la vidéosurveillance a des avantages et dans d'autre côté elle a des inconvénients [1].

1.8.1 Avantages de la vidéosurveillance

- Surveillance de déplacements des objets et possibilité de prévenir des actions concertées.
- Éléments de preuves pour poursuivre en justice les auteurs de crimes ou de délits.
- Sentiment de sécurisation dans la population (c'est à la fois un avantage et un inconvénient).
- Effet dissuasif (incertain).

1.8.2 Inconvénients de la vidéosurveillance

- Création d'un faux sentiment de sécurité (Une caméra n'empêchera pratiquement jamais une agression).
- Déplacement de la criminalité vers des zones non protégées.
- Une fois l'installation effectuée, possibilité de détourner son utilisation à d'autres fins que la sécurité.
- Le coût.
- Toujours il y a une personne qui surveille le système.

1.9. Conclusion

Ce premier chapitre a introduit des concepts généraux de la vidéosurveillance comme la vidéo, frame, objet, ... etc. Ensuite, nous avons défini qu'est ce qu'une vidéosurveillance et présenté les différents systèmes et la structure de la vidéosurveillance, le schéma général qui associe à leur fonctions (détection, suivi, reconnaissance).

Nous avons présenté brièvement quelques avantages et inconvénients de l'utilisation de la vidéosurveillance pour détecter les objets.

Il existe plusieurs méthodes et algorithmes de détection de mouvement dans la vidéosurveillance comme par exemple la soustraction de fond, qui seront l'objet du prochain chapitre.

Chapitre 2 : les étapes principales des algorithmes soustraction de fond.

2.1. Introduction

Dans ce chapitre nous présenterons les étapes principaux qui sont généralement suivent par la plupart des algorithmes de soustraction de fond.

Les quatre principales étapes d'un algorithme de soustraction de fond sont :prétraitement, la modélisation d'arrière-plan, la détection de premier plan et la validation des données. Puis nous allons voir quelques algorithmes de soustraction de fond.

2.2. Étapes principales des algorithmes de Soustraction de fond

Même si il existe une multitude d'algorithmes de soustraction de fond dans la littérature, la plupart suivent un schéma a un principe simple, représenté sur la (figure 2.1). [3]

Les quatre principales étapes d'un algorithme de soustraction de fond sont prétraitement, la modélisation d'arrière-plan, la détection de premier plan et la validation des données.

Le prétraitement se compose d'une collecte des tâches simples de traitement d'image comme le changement de la vidéo d'entrée a de bruits, en un format qui peut être traitée par les étapes suivantes. La modélisation de fond c'est une étape importante où chaque méthode doit avoir un modèle de fond, à partir de ce dernier on compare les nouvelles images pour déterminer les objets en mouvement, cse modèle fournit une description statistique de l'ensemble de la scène d'arrière-plan. La détection de premier plan identifie les pixels de l'image vidéo courant qui sont différents par rapport les pixels de modèle d'arrière-plan, les sorties comme un masque binaire de premier plan.

Enfin la validation des données consiste à faire un masque qui élimine les pixels qui ne correspondent pas aux objets en mouvement.

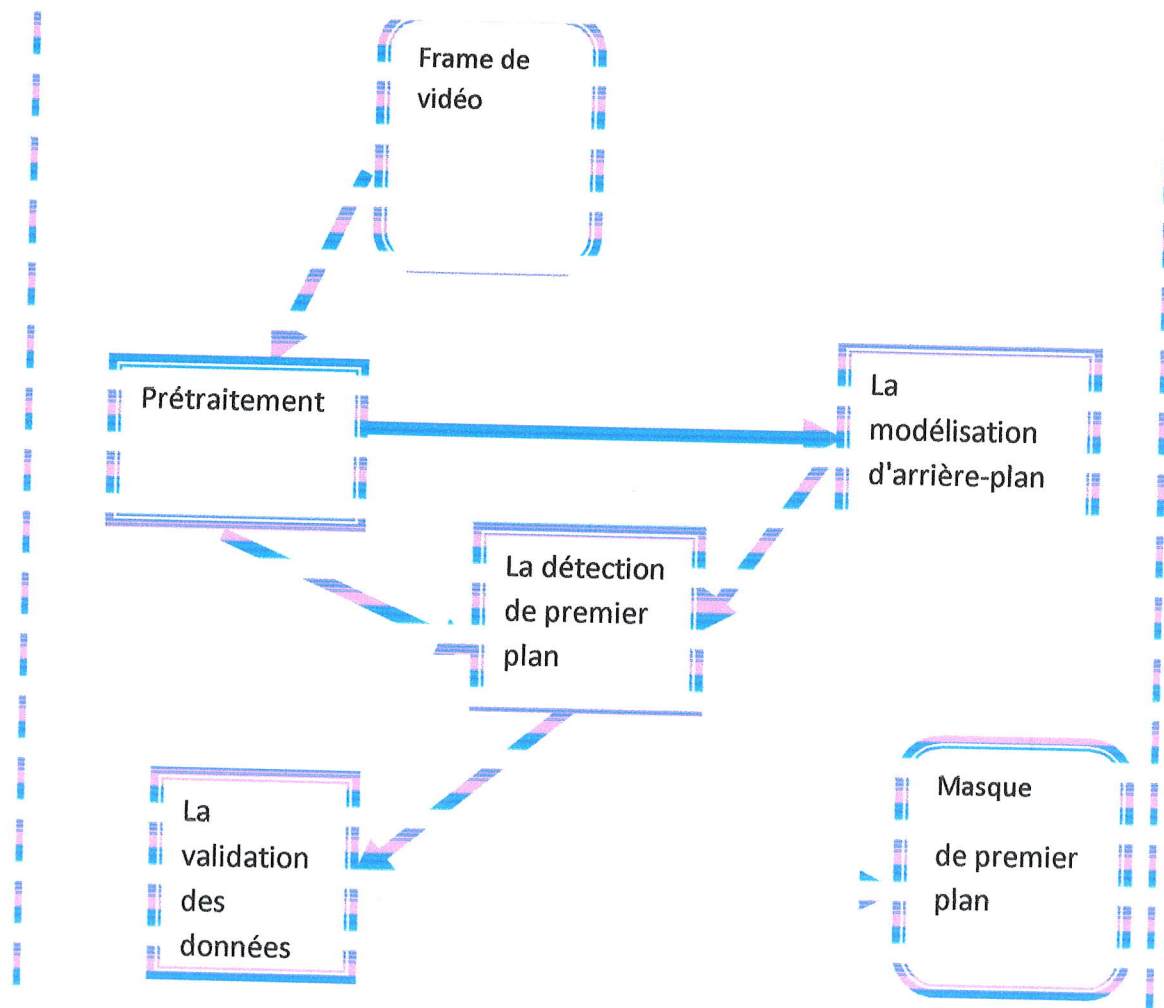


Figure 2.1 : Les Étapes de l'algorithme de Soustraction fond

2.2.1. Prétraitement

Dans l'étape du prétraitement on se trouve des opérations qui simplifient le travail de l'algorithme de soustraction de fond, pour obtenir des bons résultats généralement on a :

- la plupart des systèmes de vision par ordinateur utilisent Les filtres de lissage pour réduire le bruit de l'appareil photo et pour éliminer le bruit d'environnement comme la pluie et le neige extérieure capturé par la caméra.
- la réduction de la taille d'image dans la vidéo pour réduire le taux de traitement des données.
- les algorithmes de soustraction de fond utilisent souvent les espaces de couleur RVB ou HSV où on peut mieux identifier des objets dans les zones de faible contraste. [3]

2.2.2. La modélisation d'arrière-plan

Dans la littérature, il existe de nombreux algorithmes de modélisation de fond proposés. La modélisation de fond est nécessaire avec tous les algorithmes de soustraction de fond, beaucoup de recherches ont été consacrées à l'élaboration d'un modèle de fond qui doit être:

- robuste au changement de la luminosité et aux mouvements lents.
- éviter la détection des objets mobiles dans le fond et les ombres projetées par les objets mobiles. Les techniques de modélisation d'arrière-plan sont classées en deux catégories: Non-récurrent et récurrent. [5]

a) Non-récurrent

Une technique non-récurrente utilise un tampon qui contient " L " trames d'un vidéo ces trames représentent l'arrière-plan. Les techniques non récurrentes sont très adaptatifs, car ils ne dépendent pas de l'histoire des trames stockées dans la mémoire tampon. D'autre part, l'exigence de stockage peut être importante si un grand tampon est nécessaire pour faire face au mouvement lent, on donne un tampon de taille fixe, ce problème peut être partiellement atténué en stockant les images vidéo on utilise un faible taux de trame "L".[5]

b) récurrent

Les techniques récurrentes ne nécessitent pas un tampon pour l'estimation de fond. Au lieu de cela, ils sont mis à jour de manière récurrente un modèle de fond unique basé sur chaque trame d'entrée. Par rapport aux techniques non récurrentes, les techniques récurrentes nécessitent moins de mémoire.[3]

2.2.3. Détection de premier plan

La détection de premier plan se réalise généralement par une comparaison entre l'arrière-plan, et la trame traitée de la vidéo. Cette comparaison consiste un seuil de détection comme la méthode de la différence trame :

$$| I_t(x, y) - B_t(x, y) | > T . \quad (2.1)$$

où I_t : est le frame courant et B_t : est l'arrière-plan [5]. Où d'autre critère comme la méthode de codebook qui utilise la distorsion de couleur et la luminosité pour faire la comparaison et détecter les objets en mouvement.

2.2.4. Validation des données

La validation des données consiste à éliminer les pixels qui ne correspondent pas aux objets en mouvement, et délivre en sortie le masque de premier plan final pour examiner ce masque de la frame traitée. Cela nous permet de faire une séparation entre les objets de premier plan et l'arrière-plan, voir les changements qui ont été faits sur le fond et comparer les résultats à la vérité terrain .[5]

2.3. Algorithmes de soustraction de fond

Les algorithmes de soustraction de fond sont classés en deux catégories non récursif et récursif, la plupart de ces algorithmes sont récursif.

2.3.1. Non-récursif

2.3.1.1. Filtre temporel médian

Dans la méthode de filtre médian temporel, tout d'abord, le modèle d'arrière-plan est construit en calculant la valeur médiane d'un tampon de n dernières trames. Ensuite, un nouveau pixel est par rapport au modèle, si la différence entre le pixel entrant et le pixel de fond est inférieur d'un certain seuil donné, il est comme considéré fond, sinon il est considéré comme premier plan. La mise à jour de l'arrière-plan se fait en ajoutant la valeur actuelle du pixel dans la mémoire tampon tant que la taille de la mémoire tampon le permet.[4]

2.3.2. Récursif

2.3.2.1. Filtre médian Approximation

La méthode de la médiane approximative utilise une technique récursive pour estimer un modèle d'arrière-plan.

Cette méthode considère d'abord la première frame d'un séquence vidéo comme arrière-plan, puis pour chaque nouvelle observation, le pixel est comparé à l'arrière-plan, pour



identifier le premier plan on fait une différence entre les pixels de l'arrière-plan et pixels de la frame courante, si la différence est inférieur d'un seuil donné le pixel est considéré comme arrière-plan, sinon comme premier plan.

La mise à jour du fond est faite de la manière suivante; si le pixel de frame courante est supérieur au pixel de l'arrière-plan, l'arrière-plan est augmenté par 1, sinon l'arrière-plan est décrémenté par 1.[6]

2.3.2.2. Mixture de gaussiennes (MOG)

Dans cette méthodes chaque pixel de fond est modélisé par un mélange de gaussienne où un élément de la mélange de gaussienne est représenté par une moyenne μ_n , un écart type σ_n , et un poids α_n .

À chaque nouvelle image le pixel est correspondant à un gaussienne si satisfait des conditions, donc ce pixel est considéré comme arrière plan. L'adaptabilité du modèle de fond est assurée en mettant à jour les paramètres de cette gaussienne qui satisfait les conditions, les autres gaussiennes on normalise juste les valeurs de poids, mais la moyenne et le écart type ne changent pas.

Si aucun gaussienne n'est satisfait pas les conditions le pixel est considéré comme premier plan, la mise à jour de fond est faite par l'utilisation de gaussienne qui a le poids α_n faible par rapport les autres gaussienne, par l'utilisation les valeurs de pixel entrant.[7]

2.4. Conclusion

Ce premier chapitre a introduit les étapes principales des algorithmes de soustraction de fond où nous avons défini dans chaque étape les opérations qui vont passer, l'étape la plus importante est de modéliser l'arrière-plan car à partir le fond on peut déterminer les objets en mouvement. Nous avons vu aussi les deux grandes catégories des algorithmes de soustraction de fond avec un exemple de chaque catégorie .

Dans le chapitre suivant nous allons voir les différentes méthodes de soustraction de fond étudiés .

*Chapitre 3 : les méthodes de soustraction de
fond.*

3.1. Introduction

Dans ce chapitre, nous présentons en détail trois méthodes de soustraction de fond : la différence frame, codebook et une méthode proposée codebook avec un seul mot de code. ces méthodes nous permettent à la fin de traiter une vidéo de détecter et suivre les objets en mouvement .

Nous commencerons par la méthode la plus simple dans la littérature qui est la différence frame. Elle fonctionne avec le principe de la soustraction entre les pixels de deux frames.

Nous présentons par la suite une méthode plus complexe que la méthode de la différence frame c'est la méthode de codebook. Cette méthode consiste d'abord à modéliser un fond à partir d'une séquence frame pour obtenir une référence qui nous permet de détecter les objets en mouvement. Le codebook se base à construire pour chaque pixel un dictionnaire (codebook) qui contient un ou plusieurs mots de code, ce dernier contient des valeurs de couleur et des valeurs d'intensité. Lors la phase de la détection, elle utilise deux critères principaux, la distorsion de couleur et la distorsion de luminosité qui décident si le pixel fait partis du fond ou de l'arrière plan.

Finalement nous présentons une méthode que nous proposons, c'est la méthode de codebook avec des modifications. Elle a un fond statique qui contient un seul mot de code pour chaque pixel. Elle traite les vidéos qui contiennent des fonds statiques et pas des fonds adaptatifs comme la méthode de codebook. Cette méthode est plus rapide que la méthode de codebook dans le traitement des données car ne contient pas beaucoup de calculs.

3.2. Algorithme de la différence de frame

La différence de frame est la méthode la plus simple dans les méthodes de soustraction de fonds. Il est facile à mettre en œuvre et comprendre, le principe de cette méthode est de considérer la frame dans le temps (t) comme un fond, et la frame dans le temps(t+1) comme frame de traitement. Avant de détecter les objets de mouvement on convertit d'abord les pixels en niveau de gris, après chaque valeur de pixel d'image de fond (image t) est soustraite à partir de la valeur du pixel correspondant de l'image (image t+1). Si la valeur obtenu est inférieur à un seuil donné alors ce pixel est considéré comme pixel d'arrière plan, sinon comme premier plan. Dans cette méthode la détection d'objets est bien si

on a choisis un bon seuil [8]. Le diagramme de la méthode de différence de trame est indiqué ci-dessous:

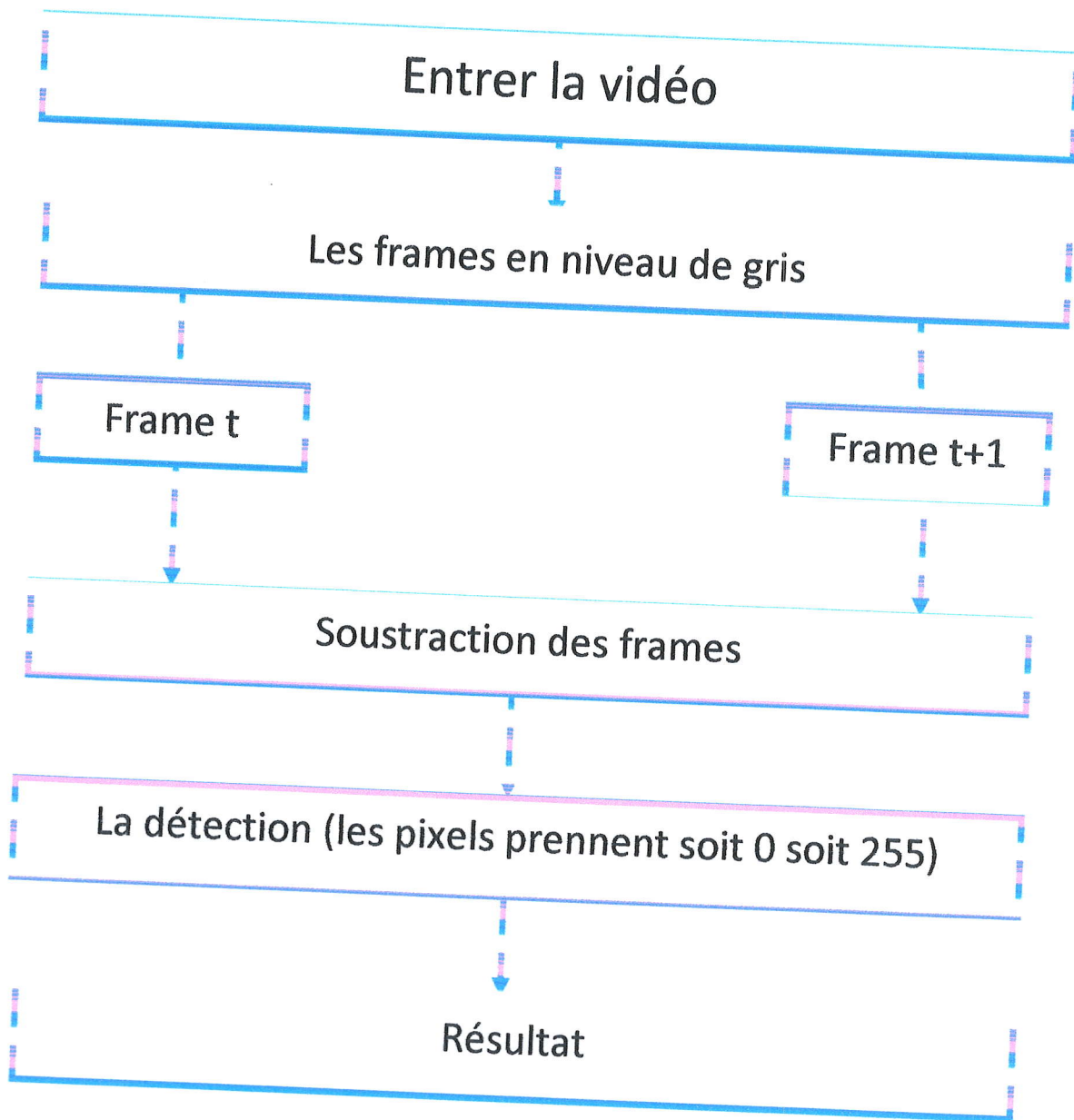


Figure 3.1 : La méthode de différence de frame.

► L'algorithme de différence de frame

1. Entrer une vidéo .
2. Considérer la frame (i) comme un fond et la frame (i+1) comme frame de traitement .
3. Convertir les valeurs de pixels de la frame de fond et la frame traitée de rgb au niveau gris.
4. Pour tout les pixels de la frame(i)et la frame (i+1), effectuez une soustraction :

Résultat de la différence ← pixel (frame i) - pixel (frame i+1)

Si (résultat de la différence > seuil)

Frame (i+1) = 255

sinon

Frame (i+1) = 0

6. Afficher l'image de sortie (frame i+1).

3.3. Algorithme de Codebook

L'algorithme de codebook a été proposée par Kim et al.(Kim et al, 2005) [9]. Il est devenu une référence de plus en plus utilisé dans les domaines de la détection et du suivi d'objets mobiles. Cette méthode est robuste et efficace dans un grand nombre de cas d'utilisation. Elle est largement utilisée pour la détection des objets en cas d'arrière-plan fixe et dynamique.

3.3.1. Fonctionnement de l'algorithme de codebook

L'algorithme de codebook a deux phase, la phase d'apprentissage et la phase de détection des objets mobiles.

3.3.1.1. Phase d'apprentissage

La méthode de Codebook consiste en un partitionnement de l'image afin de construire un modèle d'arrière-plan pendant une période d'apprentissage. Le fond est une matrice à chaque case on a un codebook qui a un ou plusieurs mot de code. Dans cette phase on prend par exemple les 20 premiers frames pour construire un modèle de fond, initialement tout les codebook sont vide à l'entrant de la première frame, chaque codebook aura un mot de code

puis à l'entrant des frames suivantes dans chaque codebook on peut trouver des nombres différents de mot de code, tout dépend des conditions de la distorsion de couleur et luminosité. S'il existe un mot de code du modèle correspondant au pixel observé alors on fait la mise à jour de ce mot de code sinon on va créer un autre mot de code.

3.3.1.2. Phase de détection des objets mobiles

Après la modélisation de fond, on a maintenant un modèle de référence pour détecter les objets de premier plan. À chaque frame entrante pour tous les pixels de cette frame, s'il existe un mot de code du modèle correspondant au pixel observé, alors on fait la mise à jour de ce mot de code et le pixel considéré comme arrière plan, sinon le pixel considéré comme premier plan.

3.3.2. Construction de l'algorithme de codebook

L'algorithme est décrit pour l'imagerie couleur, mais il peut également être utilisé pour l'imagerie de niveau gris avec des modifications mineures. Dans cette méthode chaque pixel est représenté par une table de codage $mc = \{C_1, C_2, \dots, C_L\}$ où mc représente un codebook pour un pixel où $C_i, i=1 \dots L$ représente un mot de code. Chaque pixel a une taille de codebook différente en fonction de sa variation de l'échantillon. Chaque mot de code $C_i = i \dots L$ se compose d'un vecteur RGB $V_i = (R_i, G_i, B_i)$ et 6-tuple auxi $= \{I_{\min}, I_{\max}, f, p, \lambda, q\}$, le tuple auxi contient les valeurs de l'intensité (luminosité) et les variables temporelles décrites ci-dessous :

- I_{\max} : limite maximale d'intensité du mot de code
- I_{\min} : limite minimale d'intensité du mot de code
- f : fréquence du mot de code (nombre d'occurrences)
- λ : nombre maximal d'images où le mot de code ne correspond à aucun pixel
- p : première occurrence du mot de code
- q : dernière occurrence du mot de code

Dans la période d'apprentissage, chaque valeur x_t d'un pixel à l'instant (t) est comparée à la table de codage pour déterminer le mot de code qui est correspondant, si il existe, on fait la mise à jour de ce mot de code, sinon on va créer un nouveau mot de code.

Nous utilisons deux conditions pour déterminer le mot de code qui va mettre à jour la distorsion de couleur et la distorsion de luminosité [9]. Le détail de l'algorithme est donné ci-dessous :

ALGORITHME DE CODEBOOK

Input: séquence vidéo S

Output: objet mobile

$mc \leftarrow 0, L \leftarrow 0$ (ensemble vide)/* initialement on n'a pas des mot de code liste est vide */.

$t \leftarrow 1$

pour $t=1$ à N **faire** /*N est le nombre de frames utilisées pour l'apprentissage */

pour tout pixel X_t de trame N_t **faire**

$$X_t = (R, G, B), I = \sqrt{R^2 + G^2 + B^2}$$

/* X_t les valeurs de pixel entrant , I luminosité de pixel entrant */

pour $i = 1$ à L **faire**

si (distorsion-couleur (X_t, v_i)) \leq seuil1

et (distorsion -luminosité (I, {I min, I max})) = True **alors:**

Sélectionnez un ci de mot de code appariés

sortir(break)

si aucun mot code sélectionné ou $L = 0$ **alors :**

$$mc \leftarrow mc + I, l \leftarrow l + 1$$

créer un nouveau mot de code avec les valeurs suivantes

$$v_l \leftarrow (R, G, B) \text{ et } AUXL \leftarrow \{I, I, 1, t-1, t\}$$

```

sinon
   $v_i \leftarrow ( (f_i * R_i + R) / (f_i + 1) , (f_i * G_i + G) / (f_i + 1) , (f_i * B_i + B) / (f_i + 1) )$ 
   $aux_i \leftarrow \{ \min(I, I_{min_i}), \max(I, I_{max_i}), f_i + 1, \max(\lambda_i, t - q_i), p_i, t \}$ 
  pour chaque mot de code ci faire :
     $\lambda_i \leftarrow \max\{\lambda_i, (N - q_i + p_i - 1)\}$ 

```

3.3.3. Couleur et luminosité

Pour traiter les changements d'illumination tels que des points ombres et des points lumineux, les algorithmes utilisent généralement des rapports de couleur. Cette technique fonctionne typiquement mal dans des zones sombres de l'image. Lors la phase de détection sur un fond qui contient des ombres ce n'est pas facile de distinguer entre un point ombre d'un fond ou un déplacement des objets de premier plan, c'est pour ça l'algorithme de codebook utilise un nouveau modèle de couleur, où elle prend la distorsion de couleur et la distorsion de la luminosité comme deux principaux critères pour faire la détection. Ce modèle sépare la distorsion des couleurs et de la distorsion de luminosité.[9]

3.3.3.1. Distorsion de couleur

L'évaluation de la distorsion de couleur consiste essentiellement à déterminer la distance entre la couleur d'un pixel d'entrée $X_t = (R, G, B)$ et $v_i = (R_i, G_i, B_i)$ de mot de code C_i . [9][10]

$$\|X_t\|^2 = R^2 + G^2 + B^2 \quad (3.1)$$

$$\|v_i\|^2 = R_i^2 + G_i^2 + B_i^2 \quad (3.2)$$

$$(X_t, v_i)^2 = (R_i R + G_i G + B_i B)^2 \quad (3.3)$$

$$p^2 = \frac{(x_t, v_i)^2}{\|v_i\|^2} \quad (3.4)$$

La distorsion des couleurs δ peut être calculée par :

$$\text{couleur distorsion}(x_t, v_i) = \delta = \sqrt{\|x_t\|^2 - p^2} .$$

3.3.3.2. Distorsion de luminosité

Dans chaque composant de mot de code on a les valeurs min et max de la luminosité de pixel liée à ce mot de code. [9][10]

Un pixel x_t avec une intensité $I_t = \sqrt{R^2 + G^2 + B^2}$ correspond à un mot de code c_i avec une valeur de pixel v_i et I_{\min} , I_{\max} si I_t est dans l'intervalle $[I_{\text{bas}}, I_{\text{haut}}]$ où :

$$I_{\text{bas}} = \alpha I_{\max} \quad (3.5)$$

$$I_{\text{haut}} = \min\{\beta I_{\max}, I_{\min}/\alpha\} \quad (3.6)$$

En générale, α est compris entre 0,4 à 0,8, et β est compris entre 1,1 à 1,5.

$$\text{distorsion luminosité}(I_t, (I_{\min}, I_{\max})) = \begin{cases} \text{true si } I_{\text{bas}} \leq \|x_t\| \leq I_{\text{haut}} \\ \text{false sinon} \end{cases}$$

3.3.4. Détection de premier plan

Pendant la phase de soustraction la détection de premier plan avec la méthode de codebook consiste une comparaison les pixels des frames entrantes avec les informations des mots de codes de fond, où s'il existe un mot de code du modèle correspondant au pixel observé, alors il est étiqueté comme appartenant au fond et le mot de code correspondant est mis à jour, Sinon il est étiqueté comme appartenant à un objet mobile, ici on ne va pas créer un nouveau mot de code comme la phase d'apprentissage [9][10].

L'opération se passe comme suit :

Détection de premier plan

$$I) X_t = (R, G, B) \quad , \quad I_t = \sqrt{R^2 + G^2 + B^2}$$

II) Pour tous les mots de code trouver le mot de code c_i correspondant au X_t sur la base de deux conditions :

$couleur\ dis(xt, vi) \leq seuil2$ et luminosité $(I_i, (I\ min, I\ max))=True$

Mettre à jour le mot de code adapté comme suit :

$$1) v_i \leftarrow ((f_i * R_i + R) / (f_i + 1), (f_i * G_i + G) / (f_i + 1), (f_i * B_i + B) / (f_i + 1))$$

$$2) aux_i \leftarrow \{ \min(I, I\ min_i), \max(I, I\ max_i), f_i + 1, \max(\lambda_i, t - q_i), p_i, t \}$$

III)

BGS(x) = $\left\{ \begin{array}{l} \text{premier plan} \text{ s'il n'y a pas de mot de code correspondant} \\ \text{Fond} \text{ sinon} \end{array} \right.$

3.3.5. Longueur d'exécution négative maximale

La longueur d'exécution négative maximale en anglais Maximum Negative Run-Length (MNRL) est le plus long intervalle dans la période d'apprentissage pendant lequel le mot de code n'a pas été observé. Ainsi, si la valeur MNRL d'un mot de code est faible, cela signifie que ce mot de code est fréquemment observé, inversement si cette valeur est importante, cela signifie que ce mot de code est moins fréquemment observé et donc qu'il n'appartient pas à l'arrière-plan.

Le mot de code qui a un MNRL élevé sera éliminer pour nettoyer l'arrière plan à des objets mobiles. L'élimination s'applique si le MNRL est plus grand que à la moitié du nombre de trames N de formation ($MNRL > N/2$). [11]

3.4. Algorithme de Codebook avec un seul mot de code

Nous proposons une modification de l'algorithme de codebook, et pour cela en utilisant un seul mot de code pour chaque pixel. L'algorithme de codebook avec un seul mot de code fonctionne comme l'algorithme codebook seulement au lieu de modéliser le fond par plusieurs mot de code pour chaque pixel, on modélise le fond par un mot de code dans chaque codebook.

3.4.1. Fonctionnement de l'algorithme de codebook avec un seul mot de code

3.4.1.1. Phase de modélisation de fond

Pour modéliser le fond on prend la première frame à partir de tous les pixels de cette frame on va construire le mot de code de chaque codebook, qui contient les valeur de ce pixel et la luminosité min et max, donc notre fond est un matrice à chaque case on trouve un codebook contient un seul mot de code .ce fond devient une référence pour déterminer les objets en mouvement.

3.4.1.2. Phase de détection des objets mobiles

La phase de détection des objets mobiles est une phase principale dans laquelle on peut détecter les objets en mouvement, la méthode de codebook avec un seul mot de code a le même principe de codebook avec un différent, dans la méthode de codebook s'il existe un mot de code correspondant au pixel observé on fait la mise à jour de ce mot de code, ici dans cette méthode on ne fait pas la mise à jour tout simplement, si le mot de code du modèle correspondant au pixel observé ce dernier considéré comme arrière plan, sinon le pixel considéré comme premier plan.

3.4.2. Construction de l'algorithme de codebook avec un seul mot de code

Dans cette méthode chaque pixel est représenté par une table de codage $mc = \{C_i\}$ où mc représente un codebook et $C_i, i=1$ représente le seul mot de code. Dans ce codebook, chaque mot de code représente par : C_i se compose d'un vecteur RGB $V=(R_i, G_i, B_i)$ et 2-tuple aux= $\{ I_{min_i}, I_{max_i} \}$ Le aux tuple contient les valeurs de l'intensité (luminosité) décrites ci-dessous:

- I_{max} : limite maximale d'intensité du mot de code
- I_{min} : limite minimale d'intensité du mot de code
- **Algorithme de codebook avec un seul mot de code**

ALGORITHME DE CODEBOOK AVEC UN SEUL MOT DE CODE

Input: séquence vidéo S

Output: objet mobile

prendre la première frame de la séquence vidéo

pour tout les pixels de la première

$$X_i = (R, G, B), I = \sqrt{R^2 + G^2 + B^2}$$

créer le mot de code comme suit

$$v \leftarrow (R, G, B) \text{ et } \text{AUXL} \leftarrow \{I, I\}$$

3.4.3 Couleur et luminosité

La méthode de codebook avec un seul mot de code fonctionne comme la méthode de codebook où pour traiter les changements d'illumination nous utilisons deux critères, la distorsion luminosité et la distorsion de couleur.

3.4.3.1. Distorsion de couleur

L'évaluation de la distorsion des couleurs consiste essentiellement à déterminer la distance entre la couleur d'un pixel d'entrée $X_i=(R,G,B)$ et $V_i=(R_i,G_i,B_i)$ de mot de code C_i

$$\|X_i\|^2 = R^2 + G^2 + B^2 \quad (3.7)$$

$$\|V_i\|^2 = R_i^2 + G_i^2 + B_i^2 \quad (3.8)$$

$$(X_i, V_i)^2 = (R_i R + G_i G + B_i B)^2 \quad (3.9)$$

$$p^2 = \frac{(x_i, v_i)^2}{\|v_i\|^2} \quad (3.10)$$

La distorsion des couleurs δ peut être calculée par :

$$\text{couleur distorsion}(x_i, v_i) = \delta = \sqrt{\|x_i\|^2 - p^2}$$

3.4.3.2. Distorsion de luminosité

Dans chaque composant de mot de code on a les valeurs min et max de la luminosité de pixel liée a ce mot de code .

Un pixel X_t avec une intensité $I_t = \sqrt{R^2 + G^2 + B^2}$ correspond au mot de code C_i avec une valeur de pixel V_i et I_{\min} , I_{\max} . si I_t est dans l'intervalle $[I_{\text{bas}}, I_{\text{haut}}]$ où :

$$I_{\text{bas}} = \alpha I_{\max} \quad (3.11)$$

$$I_{\text{haut}} = \min\{\beta I_{\max}, I_{\min}/\alpha\} \quad (3.12)$$

En générale , α est compris entre 0,4 à 0,8, et β est compris entre 1,1 à 1,5.

luminosité ($I_t, (I_{\min}, I_{\max})$) = $\begin{cases} \text{true si } I_{\text{bas}} \leq I_t \leq I_{\text{haut}} \\ \text{false sinon .} \end{cases}$

3.4.4. Détection de premier plan

Après la modélisation de fond nous avons une matrice de code où à chaque case on trouve un codebook qui contient un mot de code .

Pour tout les frames entrantes , si le mot de code du modèle correspondant au pixel observé, alors il est étiqueté comme appartenant au fond Sinon, il est étiqueté comme appartenant à un objet mobile .l'opération se passe comme suit :

Détection de premier plan

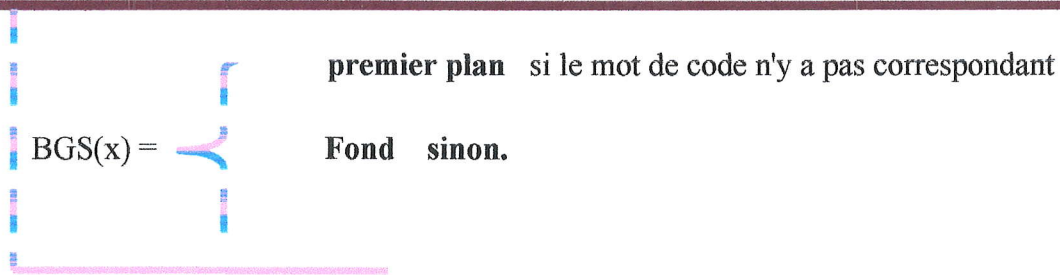
I) $X_t = (R, G, B)$, $I_t = \sqrt{R^2 + G^2 + B^2}$

II) vérifier le mot de code est ce que correspondant au X_t sur la base de deux conditions :

$$\text{couleur } dis(xt, vi) \leq \text{seuil}$$

$$\text{luminosité } (I_t, (I_{\min}, I_{\max})) = \text{True}$$

III)



3.5. Conclusion

Ce chapitre a présenté trois méthodes de soustraction de fond en détail la différence frame, codebook et une méthode proposé le codebook avec un seul mot de code. Ces méthodes ont un objectif commun est de détecter les objets en mouvement dans une séquence vidéo. Tout ces méthodes ont besoin d'un fond pour faire la comparaison à chaque image entrante et détecter les objets en mouvement.

On a vu que la méthode de la différence frame sans doute est la plus simple, elle fait une soustraction des pixels entre les pixels de fond et les pixels de la frame actuelle pour déterminer le changement et donc déterminer les objets en mouvement. Le codebook et codebook avec un seul mot de code sont des méthodes plus complexe que la méthode de la différence frame, elles utilisent la couleur et la luminosité pour détecter les objets en mouvement.

Dans le chapitre suivant, nous allons implémenté et expérimenté nos algorithmes. Puis interpréter ces résultats.

Chapitre 4 : Expérimentation et résultats.

4.1. Introduction

Dans ce chapitre nous allons évaluer les résultats de ces trois algorithmes codebook et différence frame et codebook avec un seul mot de code. on va changer à chaque fois les paramètres liés pour chaque algorithme pour déterminer avec quels paramètres ils donnent des meilleurs résultats .

Pour évaluer la détection de chaque algorithme on va calculer le rappel et la précision pour décider est que ce un algorithme a une bonne détection ou pas . Aussi on va comparer le temps d'exécution de chaque algorithme pour savoir laquelle est la plus rapide .

4.2. Choix du logiciel

Le choix d'un bon environnement de programmation est très important pour le développement des projets. cela se fait suivant plusieurs facteurs: la puissance de compilation , la facilité d'utilisation ,la disponibilité des fonctionnalités essentielles et la communication avec d'autres environnements etc. À fin de réaliser notre système nous avons eu recours à un langage intéressant : python .

Python est un langage portable ,dynamique extensible gratuit qui permet (sans l'imposer)une approche modulaire et orientée objet de la programmation .python est développé depuis 1989 par guido van Rossumet de nombreux contributeurs bénévoles .

4.2.1 Avantage du python

- interprété
- python est orienté objet il supporte l'héritage et la surcharge des opérateurs
- haut niveau
- A syntaxe positionnelle
- python est portable
- lisible
- python est extensible

4.2.2. Inconvénients du python

- lent
- absence de pointeurs

➤ typage différent

4.2.3. Open CV

Open CV (Open Source Computer vision) est une bibliothèque graphique libre, initialement développée par Intel et maintenant soutenue par la société de robotique Willow Garage, spécialisée dans le traitement d'images. Elle est livrée avec une interface de programmation en C, C++, Python et Android. La bibliothèque Open CV met à disposition de nombreuses fonctionnalités très diversifiées, elle propose la plupart des opérations classiques en traitement d'images et traitement de vidéo où elle nous permet de réaliser notre application et d'appliquer les trois algorithmes de soustractions de fond.

4.3. Interface graphique d'utilisateur

L'interface utilisateur comporte une page principale qui contient des boutons permettant de faciliter l'exploitation des fonctionnalités de notre application de la suivi de personne.

4.3.1. Page d'accueil

Cette interface permet d'accéder à l'application :

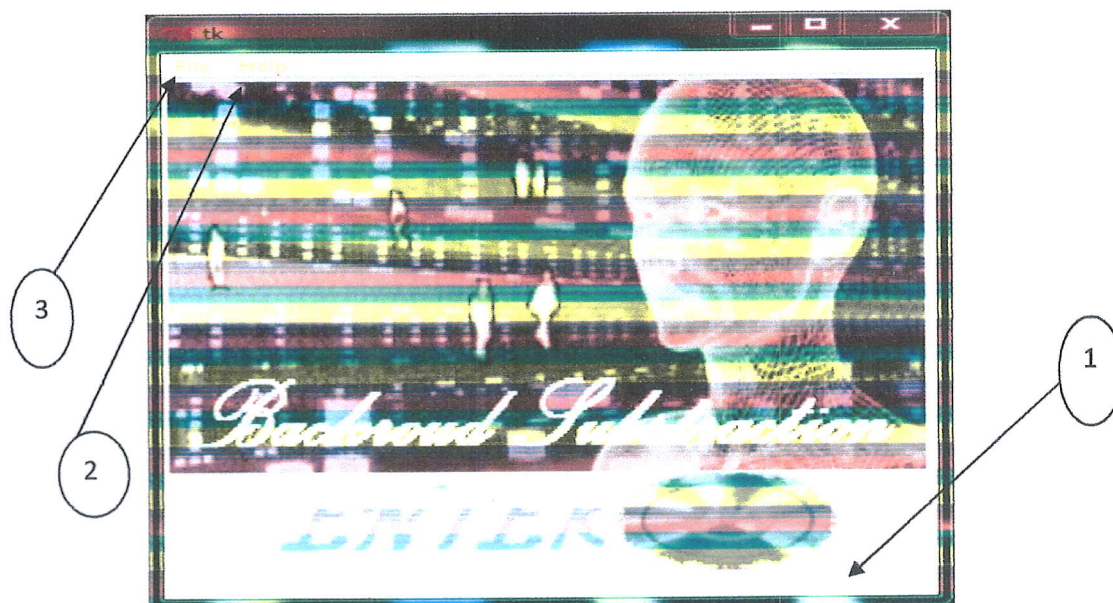


Figure 4.1 : Page d'accueil de l'application.

1 : Pour accéder à l'application on doit cliquer sur le bouton << entrer >>.

2 : Pour quitter l'application on doit cliquer sur le bouton << file >> ,on trouve un autre bouton << exit >> ce bouton pour quitter l'application .

3 : Pour plus d'information on doit cliquer sur le bouton << Help >> ,on trouve deux d'autre bouton << About >> pour des informations générale sur étudiant ,encadreur , et << A propre >> pour le guide d'utilisation de l'application.

4.3.2. Page principale <<application >>

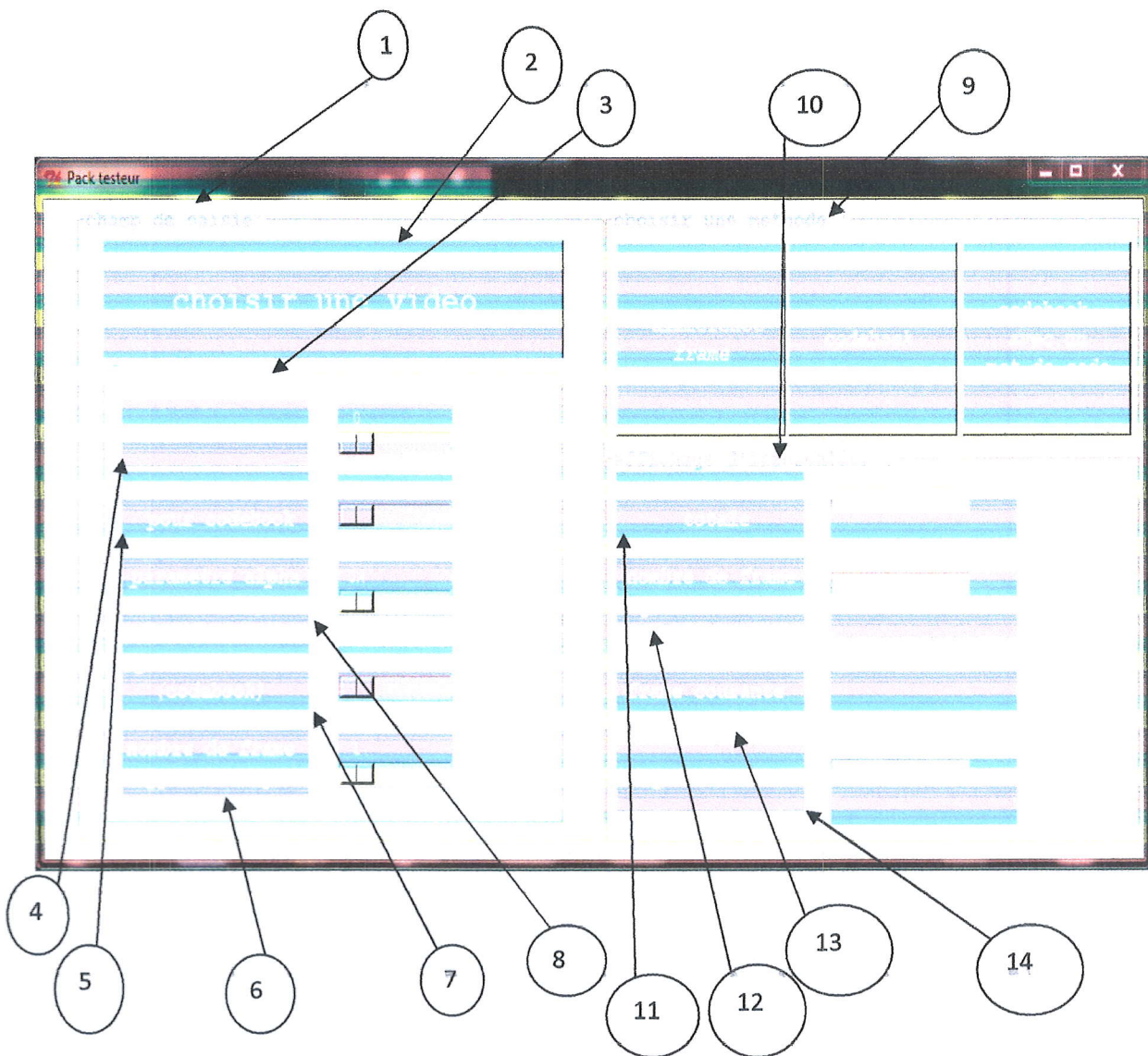


Figure 4.2 : La page principale <<application>>

1: champ de saisie pour saisir les informations pour députer votre exécution .

2 : pour choisir une vidéo vous devez cliquer sur le bouton <choisir une vidéo>> ,vous trouverez des vidéos choisissez une .

3 : Champ de paramètres vous trouverez dans ce champs, les paramètres liées pour chaque méthodes, où vous devez les saisir avant chaque exécution.

4 : Seuil 1 pour tout les méthodes .

5: Seuil 2 pour seulement la méthodes de codebook ,ce seuil est pour faire l'apprentissage .

6 : Nombre de frame d'apprentissage pour seulement la méthodes de codebook vous devez entrer un nombre d'apprentissage .

7 : Paramètre beta pour la méthodes de codebook et codebook avec un seul mot de code .

8 : Paramètre alpha pour la méthodes de codebook et codebook avec un seul mot de code .

9 : Choisir une méthode ,vous trouverez dans ce champ trois bouton concernant à nos méthodes (codebook, différence frame, codebook avec un seul mot de code) clique sur un pour démarrer le traitement de la vidéo. À moment où vous cliquez sur le bouton la couleur va changer pour distinguer la méthode qui est marche actuellement par rapport à les autres méthodes .

10 : Champ d'affichage est pour afficher les 'information liées à la vidéo et à la méthode à exécuter .

11 : Nombre de frame totale de la vidéo.

12 : Nombre de frame par seconde dans la vidéo .

13: Frame courante dans ce label va afficher le numéro de frame à exécuter au cour de l'exécution total de la méthode.

14 : Temps d'exécution est pour afficher le temps totale d'exécution d'une méthode, après la détermination d'exécution .

4.4. Résultats expérimentaux

4.4.1. Vérités terrains

Le principal problème pour mesurer la qualité de la détection, provient de la difficulté à obtenir une vérité terrain que l'on pourra ensuite comparer à l'image extraite. La solution est d'identifier à l'aide d'un logiciel de dessin, de façon manuelle, les zones de formes et les zones de fond, et cela pour chacune des images de la séquence à analyser.

Ce travail est assez fastidieux, et ne permet pas de tester les algorithmes avec beaucoup de séquences étant donné le temps nécessaire à la réalisation des vérités terrains.

4.4.2. Classification de pixel

Avant d'analyser la qualité de l'extraction, l'image i obtenue après traitement par l'un des algorithmes est comparée avec l'image de vérité terrain correspondante, nous devons classer les pixels selon les catégories suivantes :[12]

- VP (vrais positifs) : premier plan détecté comme premier plan.
- FP (faux positifs) : fond détecté comme premier plan.
- VN (vrais négatifs) : fond détecté comme fond.
- FN (faux négatifs) : premier plan détecté comme fond.

4.4.3. Rappel et précision

Pour comparer nos résultats obtenus avec la vérité terrain on calcule le rappel et la précision sont deux critères pour mesurer la qualité de l'extraction .

4.4.3.1. Rappel

Le rappel est définie comme le nombre de vrais positifs divisé par le nombre total d'éléments qui appartiennent au premier plan.

$$\text{Rappel} = \frac{TP}{TP+FN} \quad (4.1)$$

En d'autres termes, il peut être réécrite comme suit :

$$\text{Rappel} = \frac{\text{nombre de pixels de premier plan correctement identifiés}}{\text{nombre de pixels de premier plan dans la réalité de terrain}} \quad (4.2)$$

4.4.3.2. Précision

La précision peut être considérée comme une mesure de précision, elle évalue en divisant le nombre d'objets de premier plan correctement détectée par le nombre total de pixels classés en premier plan par l'algorithme.

$$\text{précision} = \frac{TP}{TP+FP} \quad (4.3)$$

En d'autres termes, il peut être réécrite comme:

$$\text{précision} = \frac{\text{nombre de pixels de premier plan correctement identifiés}}{\text{nombre de pixels de premier plan détectés par algorithme}} \quad (4.4)$$

➤ Les valeurs du rappel et de la précision sont entre 0 et 1. Une bonne algorithme qui détecte bien les objets en mouvement a des valeurs de rappel et de précision approchées de 1 .

4.4.4. Résultat de rappel et précision sur une vidéo filmée

Dans cette partie nous allons calculer le rappel et la précision avec les trois algorithmes codebook, codebook avec un seul mot de code et la différence frame. Pour calculer le rappel et la précision on utilise logiciel Paint pour comparer manuellement avec la vérité terrain.

À chaque fois on va changer les paramètres de chaque algorithme pour connaître pour quels valeurs elle donne de bons résultats .

On va tester l'algorithme de la différence frame avec plusieurs seuil pour obtenir un seuil idéal qui nous permet d'obtenir une bonne détection . Aussi on va tester l'algorithme de codebook et codebook avec un seul mot de code avec des différents paramètres de distorsion de luminosité où on va changer α et β et le seuil de distorsion de couleur .

La taille de frame de vidéo utilisée est 200×250 et le nombre de frames par seconde est 25.

4.4.4.1. Résultats de rappel et précision pour la méthode codebook

➤ Voici une image traitée obtenu à près l'exécution de l'algorithme de codebook par une vidéo filmée (figure 4.3)

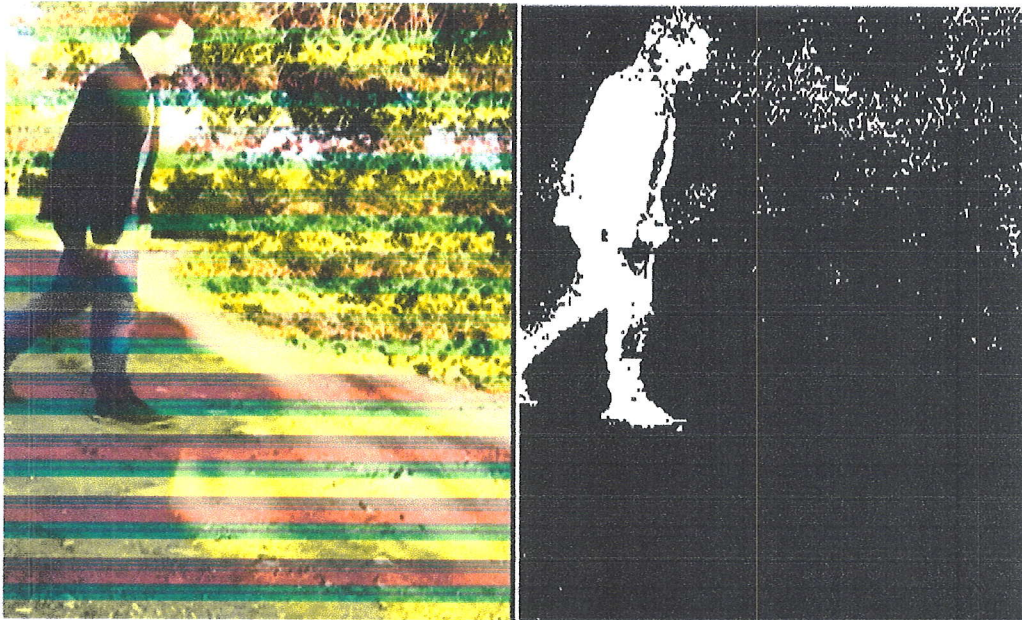


Figure 4.3 : La détection des objets pour la méthode de codebook. À (gauche) frame réelle. À (droite) frame traitée (frames d'une vidéo filmée)

➤ Pour calculer le rappel et la précision on a choisi un ensemble de frames aléatoirement où à chaque fois on va changer α , β et le seuil2 de distorsion de couleur, le seuil d'apprentissage est fixé avec tout les essais (seuil1=0.2). Les résultats obtenus sont présentés dans le tableau suivant :

Taille Vidéo	Paramètres	TP	FP	FN	frames	Rappel	Précision
00:00:05	$\alpha=0.4 \beta=1.1$ <i>seuil2=80</i>	700	220	250	Frame 1	0.73	0.76
		670	290	250	Frame 2	0.72	0.69
	$\alpha=0.4 \beta=1.4$ <i>seuil2=25</i>	755	300	245	Frame 1	0.75	0.71
		600	295	220	Frame 2	0.73	0.66
	$\alpha=0.6 \beta=1.3$ <i>seuil2=50</i>	820	170	260	Frame 1	0.75	0.82
		640	180	250	Frame 2	0.71	0.78
	$\alpha=0.7 \beta=1.4$ <i>seuil2=35</i>	810	120	160	Frame 1	0.75	0.87
		830	80	200	Frame 2	0.80	0.91
	$\alpha=0.8 \beta=1.8$ <i>seuil2=30</i>	1000	65	100	Frame 1	0.93	0.90
		1040	85	90	Frame 2	0.92	0.92

Tableau 4.1 : Tableau des résultats de l'algorithme de codebook avec une vidéo filmée.

Le diagramme de rappel et précision (Recall Precision Curve -RPC) sera dans la (figure 4.4)

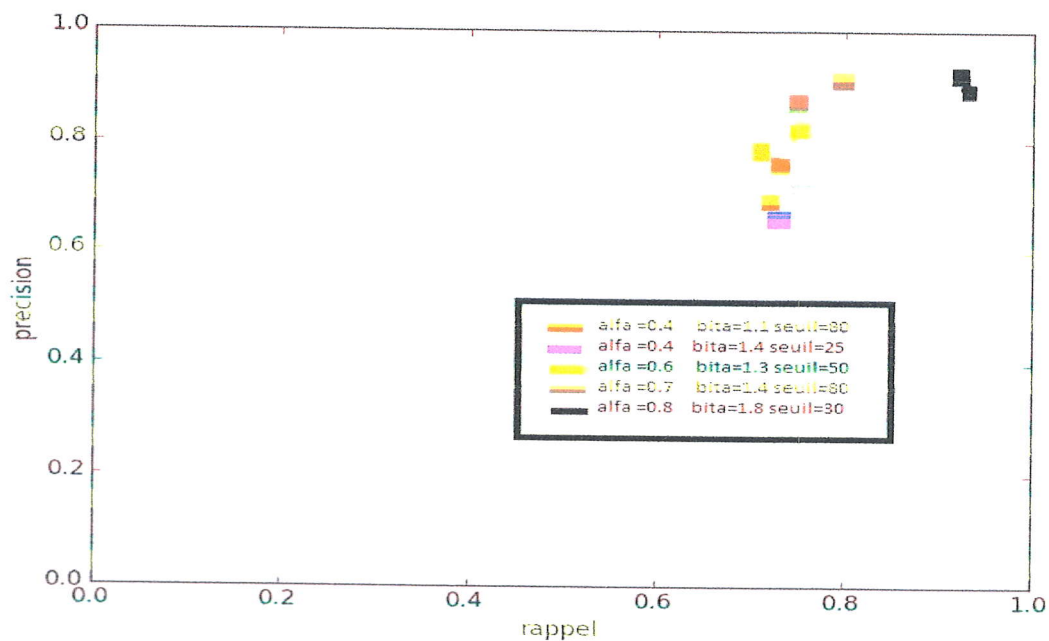


Figure 4.4 : Le rappel et précision de la méthode codebook (vidéo filmée).

➤ D'après le tableau et le diagramme nous observons que l'algorithme de codebook a donné une bonne détection avec les paramètres suivantes $\alpha=0.8$ $\beta=1.8$ $\text{seuil}=30$. Ces paramètres choisis dépend de plusieurs critères : la luminosité, l'ombre, et la différence de couleur de premier plan et les couleurs de l'arrière plan.

4.4.4.2. Résultats de rappel et précision par la méthode de différence frame

➤ Voici une image traitée obtenue à près l'exécution de l'algorithme de différence frame par une vidéo filmée (figure 4.5).

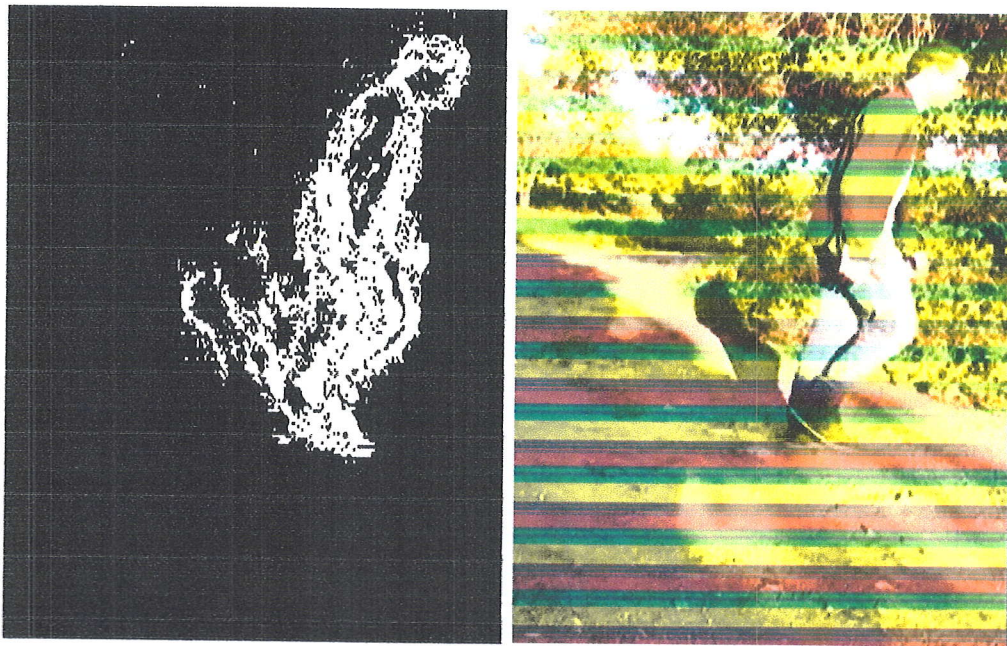


Figure 4.5 : la détection des objets par la méthode de différence frame. À (gauche) frame traitée. À (droite) frame réelle (frames d'une vidéo filmée).

➤ Pour calculer le rappel et la précision on a choisi un ensemble de frames aléatoirement où à chaque fois on va changer le seuil de la détection. Les résultats obtenus sont présentés dans le tableau suivant.

Taille Vidéo	paramètres	TP	FP	FN	frames	Rappel	Précision
00:00:05	Seuil=100	120	0	680	Frame 1	0.15	1
		145	0	700	Frame 2	0.17	1
	Seuil=80	170	0	640	Frame 1	0.20	1
		220	0	700	Frame 2	0.26	1
	Seuil=50	270	0	600	Frame 1	0.31	1
		300	0	600	Frame 2	0.33	1
	Seuil=30	350	0	420	Frame 1	0.45	1
		400	0	450	Frame 2	0.47	1
	Seuil=10	660	30	210	Frame 1	0.75	0.95
		645	40	230	Frame 2	0.73	0.94

Tableau 4.2 : Tableau des résultats de l'algorithme de différence frame avec une vidéo filmée.

Le diagramme de rappel et précision (Recall Precision Curve -RPC) sera dans la (figure 4.6)

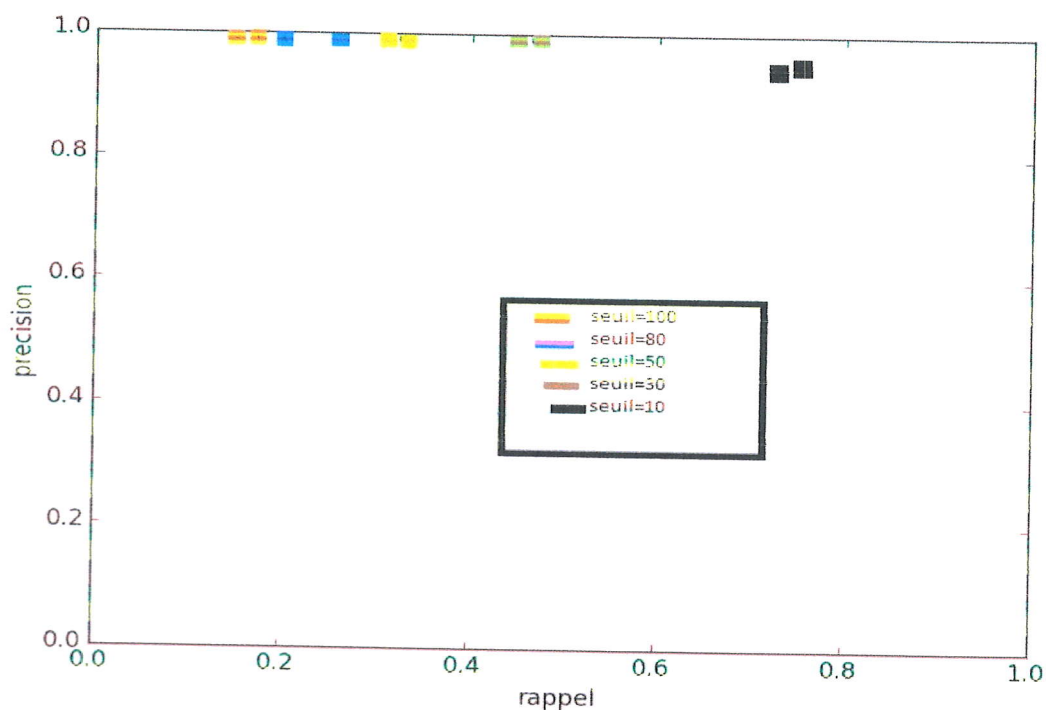


Figure 4.6 : Le rappel et précision méthode la différence frame (vidéo filmée).

➤ D'après le tableau et le diagramme nous observons que l'algorithme de la différence frame a donnée une bonne détection avec le seuil=10. Ce dernier obtenu dépend de plusieurs critère l'ombre , la différence de couleur entre les objets mobile et les couleur de fond .

Le seuil est très important pour la détection des objets dans la méthode de différence frame donc il faut le bien choisir . Ainsi, si on a mal choisi (trop grand ou trop faible), la détection sera de mauvaise qualité comme dans (la figure 4.7)où on a choisir le seuil égal à 100.

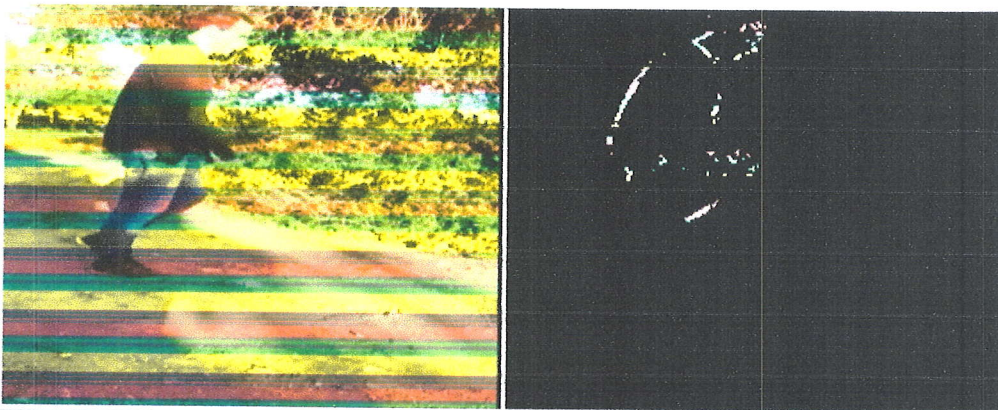


Figure 4.7 : La détection des objets en mouvement par la méthode de différence frame avec un seuil =100.

4.4.4.3. Résultat de rappel et précision par la méthode de codebook avec un seul mot de code

➤ Voici une image traitée obtenu à près l'exécution de l'algorithme de codebook avec un seul mot de code par une vidéo filmée (figure 4.8).

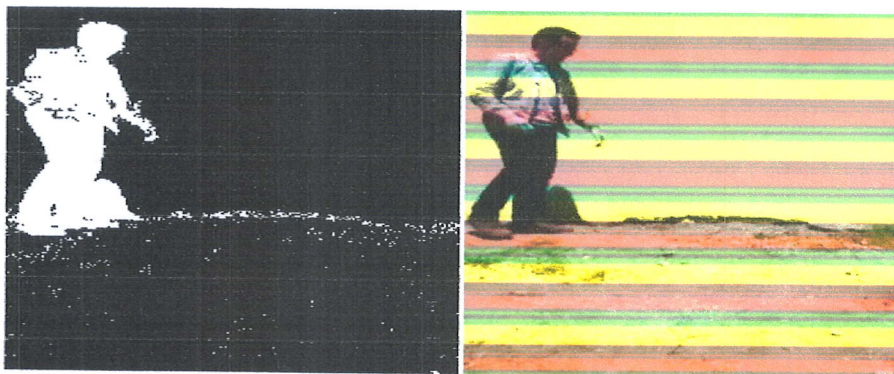


Figure 4.8 : La détection des objets par la méthode de codebook avec un seul mot de code. À (gauche) frame traitée. À (droite) frame réelle (frames d'un vidéo filmé).

➤ Pour calculer le rappel et la précision on a choisi un ensemble de frames aléatoirement où à chaque fois on va changer α , β et le seuil de distorsion de couleur. Les résultats obtenus sont présentés dans le tableau suivant :

Taille Vidéo	paramètres	TP	FP	FN	frames	rappel	Précision
00:00:05	$\alpha=0.5 \beta=1.2$ <i>seuil=70</i>	600	50	240	Frame 1	0.71	0.92
		680	60	250	Frame 2	0.81	0.91
	$\alpha=0.4 \beta=1.1$ <i>seuil=80</i>	770	90	275	Frame 1	0.73	0.88
		650	80	285	Frame 2	0.69	0.89
	$\alpha=0.6 \beta=1.4$ <i>seuil=30</i>	679	25	117	Frame 1	0.85	0.96
		620	23	190	Frame 2	0.76	0.96
	$\alpha=0.7 \beta=1.5$ <i>seuil=20</i>	623	27	45	Frame 1	0.93	0.95
		750	20	40	Frame 2	0.94	0.97
	$\alpha=0.7 \beta=1.3$ <i>seuil=40</i>	790	17	110	Frame 1	0.87	0.97
		687	22	125	Frame 2	0.84	0.96

Tableau 4.3 : Tableau des résultats de l'algorithme de codebook avec un seul mot de code une vidéo filmée .

Le diagramme de rappel et précision (Recall Precision Curve -RPC) sera dans la (figure 4.9)

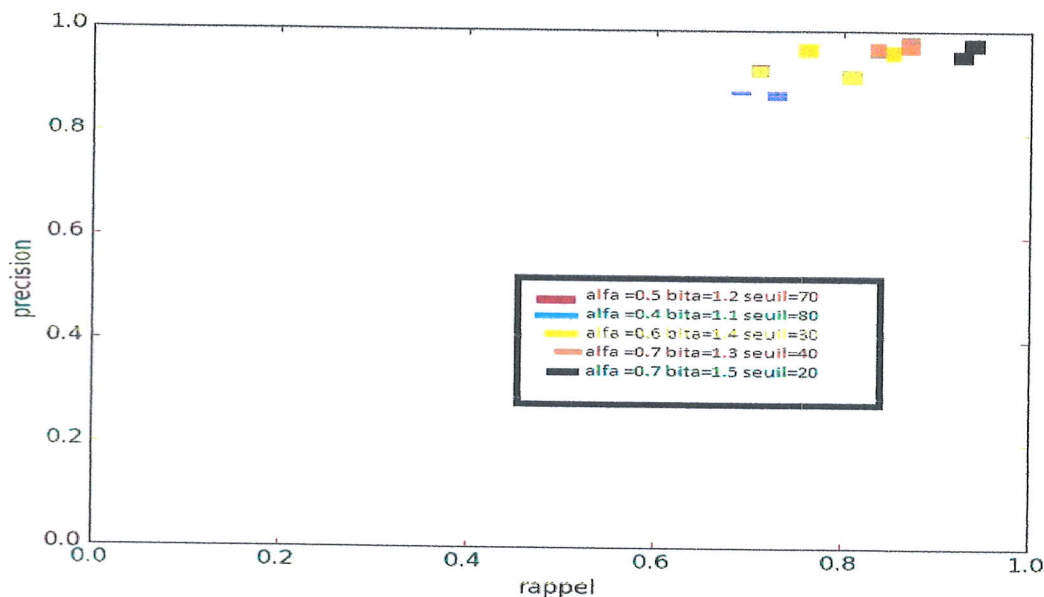


Figure 4.9 : Le rappel et précision méthode de codebook avec un seul mot de code(vidéo filmée).

➤ D'après le tableau et le diagramme nous observons que l'algorithme de codebook a donné une bonne détection avec les paramètres suivantes $\alpha=0.7$, $\beta=1.5$ et seuil=20. Comme la méthode de codebook le choix de ces paramètres dépend de plusieurs critères : la luminosité, l'ombre, et la différence de couleur de premier plan et des couleurs de l'arrière plan.

4.4.5. Résultat de rappel et précision par une vidéo de base de données

Pour tester nos algorithmes, nous avons téléchargé une vidéo de base de données "computer vision laboratory cvlab" [18]. Comme nous avons fait avec la vidéo filmée, on va calculer le rappel et la précision avec les trois algorithmes où à chaque fois on va changer les paramètres pour connaître lesquelles nous donnent les meilleurs résultats. La taille de frame de vidéo utilisée est 200×250 et le nombre de frame par seconde est 25.

4.4.5.1. Résultat de rappel et précision par la méthode codebook

➤ Voici une image traitée obtenue à près l'exécution de l'algorithme de codebook par une vidéo de base de données (figure 4.10)+

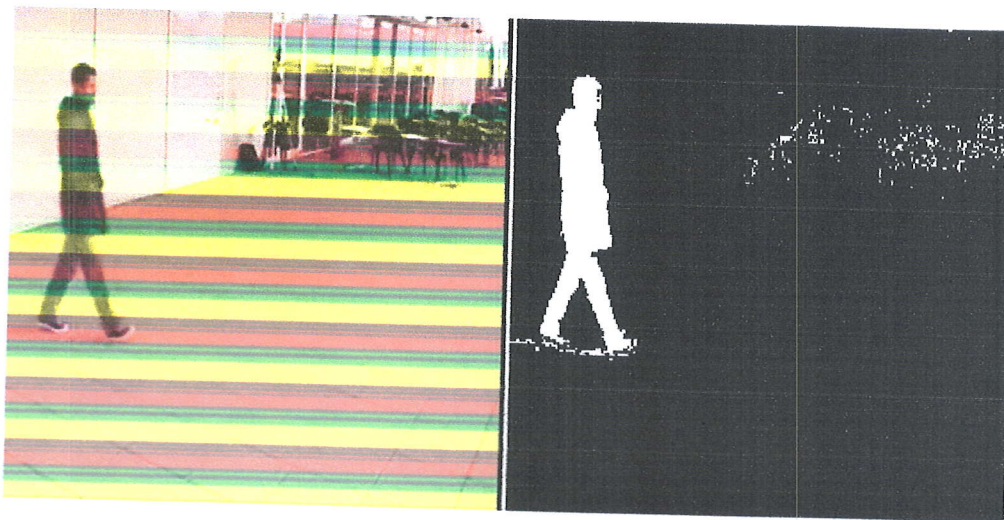


Figure 4.10 : la détection des objets par la méthode de codebook. À (gauche) frame réelle. À (droite) frame traitée. (vidéo d'une base de données).

➤ Pour calculer le rappel et la précision on a choisi un ensemble de frames aléatoirement où à chaque fois on va changer α , β et le seuil de distorsion de couleur, le seuil d'apprentissage est fixé avec tous les essais seuil=0.2. Les résultats obtenus sont présentés dans le tableau suivant :

Taille	paramètres	TP	FP	FN	frames	Rappel	Précision
00:00:04	$\alpha=0.5 \beta=1.5$ <i>seuil2=60</i>	420	150	250	Frame 1	0.62	0.73
		380	200	280	Frame 2	0.57	0.65
	$\alpha=0.6 \beta=1.4$ <i>seuil2=40</i>	450	220	180	Frame 1	0.71	0.67
		390	260	160	Frame 2	0.70	0.60
	$\alpha=0.5 \beta=1.3$ <i>seuil2=50</i>	500	240	190	Frame 1	0.72	0.67
		300	300	270	Frame 2	0.52	0.50
	$\alpha=0.7 \beta=1.7$ <i>seuil2=30</i>	500	30	80	Frame 1	0.86	0.94
		600	35	85	Frame 2	0.87	0.93
	$\alpha=0.9 \beta=1.8$ <i>seuil2=35</i>	675	35	23	Frame 1	0.96	0.95
		690	34	18	Frame 2	0.97	0.95

Tableau 4.4 : Tableau des résultats de l'algorithme de codebook par une vidéo de base de donnée .

Le diagramme de rappel et précision (Recall Precision Curve -RPC) sera dans la (figure4.11)

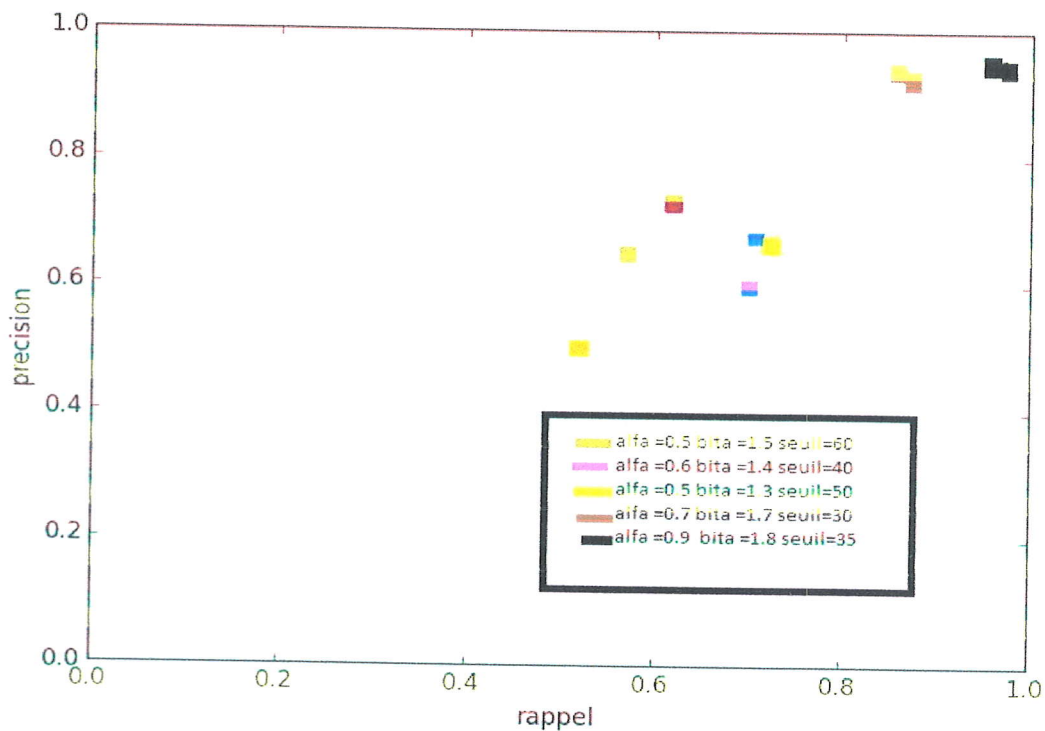


Figure 4.11 : Le rappel et précision méthode de codebook .(vidéo de base de donnée).

➤ D'après le tableau et le diagramme nous observons que l'algorithme de codebook a donné une bonne détection avec les paramètres suivantes $\alpha=0.8$ $\beta=1.8$ $\text{seuil}=35$.

4.4.5.2. Résultats de rappel et précision pour la méthode de différence frame

➤ Voici une image traitée obtenue à près l'exécution de l'algorithme de différence frame par une vidéo de base de donnée (figure 4.12).

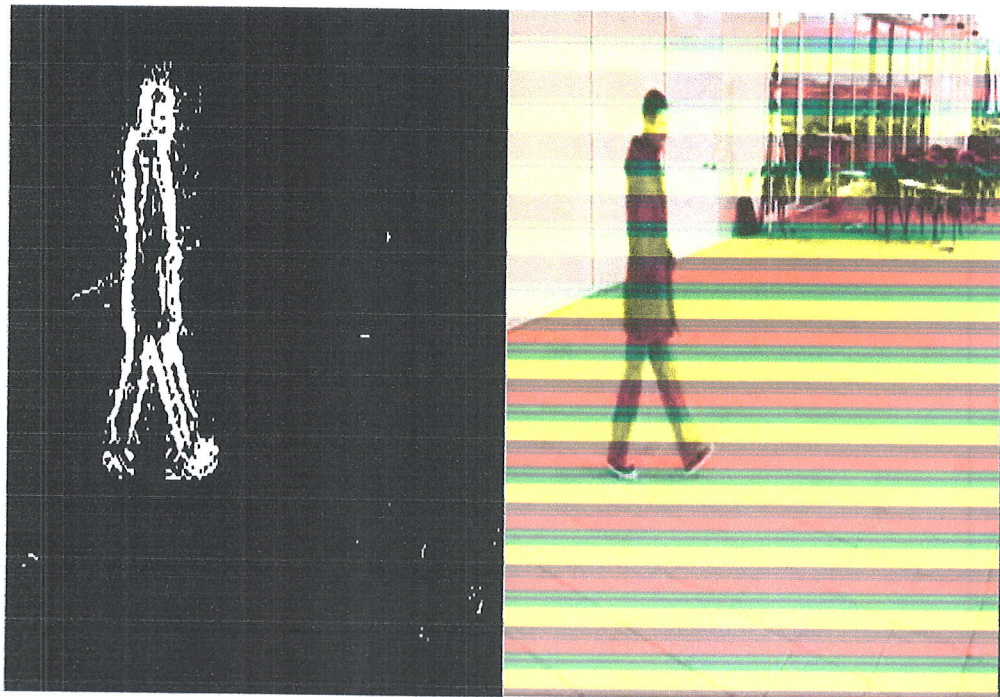


Figure 4.12 : la détection des objets par la méthode de la différence frame. À (gauche) frame traitée . À (droite) frame réelle .(vidéo d'une base de donnée).

➤ pour calculer le rappel et la précision on a choisi un ensemble de frames aléatoirement où à chaque fois on va changer le seuil de la détection. Les résultats obtenus sont présentés dans le tableau suivant :

s

Taille	paramètres	TP	FP	FN	frames	Rappel	Précision
00:00:04	Seuil=100	100	0	380	Frame 1	0.20	1
		140	0	420	Frame 2	0.25	1
	Seuil=80	110	0	370	Frame 1	0.22	1
		160	0	400	Frame 2	0.28	1
	Seuil=50	165	0	370	Frame 1	0.30	1
		200	0	380	Frame 2	0.34	1
	Seuil=30	255	0	310	Frame 1	0.45	1
		310	0	280	Frame 2	0.52	1
	Seuil=15	540	15	110	Frame 1	0.83	0.97
		520	48	95	Frame 2	0.84	0.91

Tableau 4.5 : Tableau des résultats de l'algorithme de différence frame par une vidéo de base de donnée ,

Le diagramme de rappel et précision (Recall Precision Curve -RPC) sera dans la (figure 4.13)

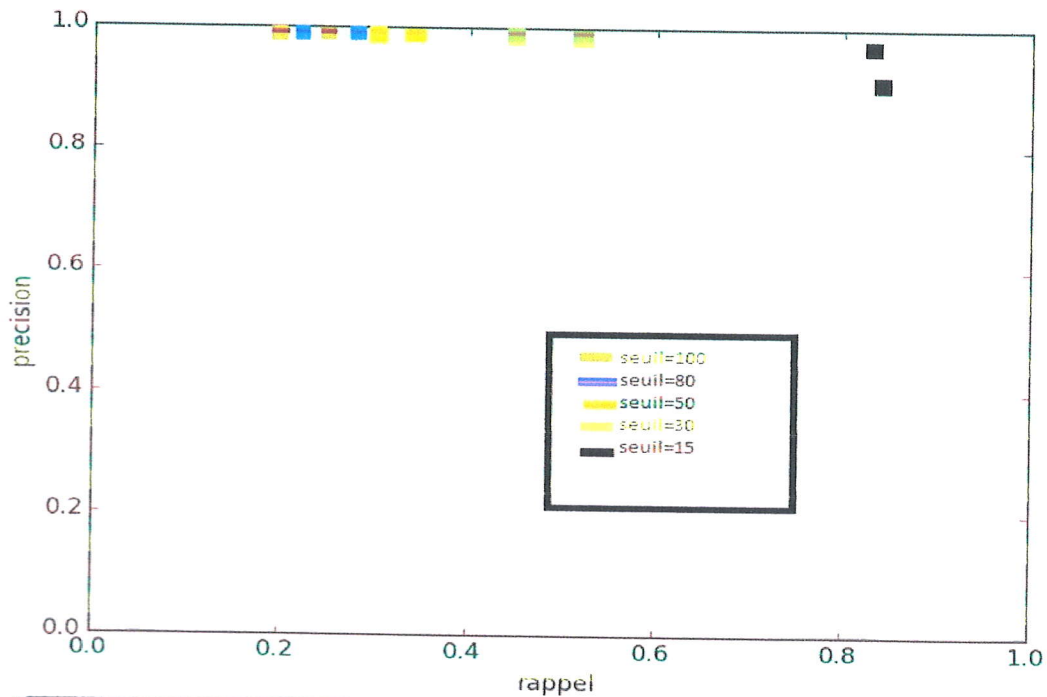


Figure 4.13 : le rappel et précision méthode la différence frame.(vidéo de base de donnée).

➤ D'après le tableau et le diagramme nous observons que l'algorithme de la différence frame a donnée une bonne détection avec le seuil $\tau=10$.

4.4.5.3. Résultats de rappel et précision par la méthode de codebook avec un seul mot de code

➤ voici une image traitée obtenu à près l'exécution de l'algorithme de codebook avec un seul mot de code par une vidéo de base de donnée (figure 4.14).

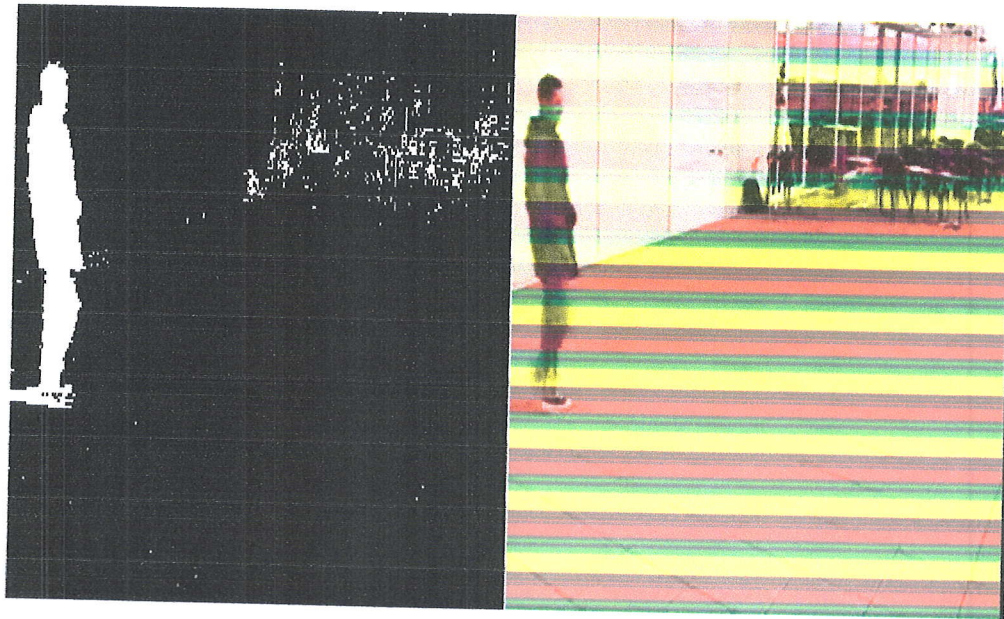


Figure 4.14 : la détection des objets par la méthode de codebook avec un seul mot de code . À (gauche) frame traitée. À (droite) frame réelle. (vidéo d'une base de donnée).

➤ Pour calculer le rappel et la précision on a choisi un ensemble de frames aléatoirement où à chaque fois on va changer α , β et le seuil de distorsion de couleur. Les résultats obtenus sont présentés dans le tableau suivant :

Taille	paramètres	TP	FP	FN	frames	Rappel	Précision
00:00:04	$\alpha=0.4 \beta=1.1$ seuil=80	350	95	180	Frame 1	0.71	0.93
		340	160	185	Frame 2	0.68	0.93
	$\alpha=0.5 \beta=1.2$ seuil=50	460	20	120	Frame 1	0.79	0.95
		480	22	80	Frame 2	0.85	0.95
	$\alpha=0.6 \beta=1.3$ seuil=30	500	18	90	Frame 1	0.84	0.96
		520	21	85	Frame 2	0.85	0.96
	$\alpha=0.7 \beta=1.5$ seuil=30	623	27	45	Frame 1	0.93	0.96
		655	31	60	Frame 2	0.91	0.95
	$\alpha=0.8 \beta=1.8$ seuil=35	673	18	20	Frame 1	0.97	0.97
		685	15	30	Frame 2	0.95	0.97

Tableau 4.6 : Tableau des résultats de l'algorithme de codebook avec un seul mot de code par une vidéo de base de donnée .

Le diagramme de rappel et précision (Recall Precision Curve -RPC) sera dans la (figure 4.15)

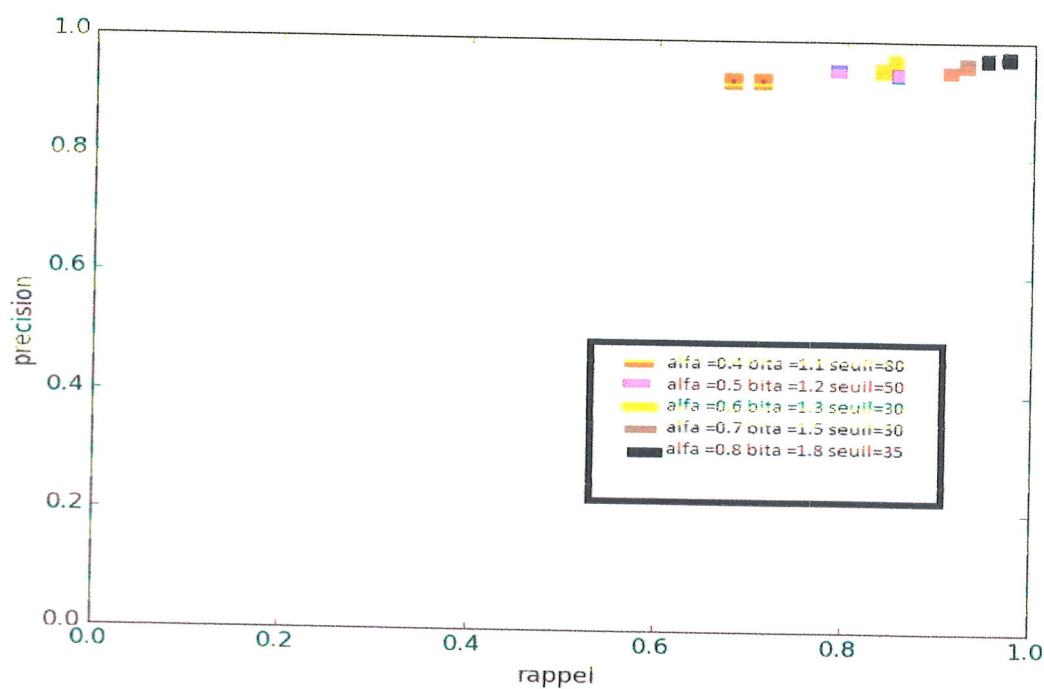


Figure 4.15 : le rappel et précision méthode de codebook avec un seul mot de code.(vidéo de base de donnée)

➤ D'après le tableau et le diagramme nous observons que l'algorithme de codebook avec un seul mot de code a donné une bonne détection avec les paramètres suivantes $\alpha=0.8$ $\beta=1.8$ $\text{seuil}=35$.

4.4.6. Comparaison entre les méthodes

4.4.6.1. Rappel et précision

a) Vidéo filmée

Le tableau suivant(tableau 4.7) représente une comparaison entre les trois méthodes on particulier sur les valeurs de rappel et précision. Chaque méthode est présenté avec ces meilleurs valeurs de paramètre pour lesquels elle a donnée la meilleurs détection .

Méthode \ Paramètre	Codebook	Différence frame	Codebook avec un seul mot de code
α	0.8	/	0.7
β	1.8	/	1.5
Seuil	30	20	20
Nombre de frame d'apprentissage	20	/	/
Rappel	0.925	0.74	0.935
Précision	0.91	0.945	0.95

Tableau 4.7 : Tableau comparative entre les meilleurs rappel et précision des méthodes .(par une vidéo filmée).

➤ Les valeurs de rappel et de précision dans ce tableau sont obtenu par la moyenne de deux valeurs de rappel et de précision .

b) Vidéo de base CVLAB

Le tableau suivant(tableau 4.8), représente une comparaison entre les trois méthodes chaque méthode avec les paramètre pour lesquels la méthode a donne une bonne détection .

Méthode Paramètre	Codebook	Différence frame	Codebook avec un seul mot de code
α	0.9	/	0.8
β	1.8	/	1.8
Seuil	35	15	35
Nombre de frame d'apprentissage	20	/	/
Rappel	0.965	0.835	0.96
Précision	0.95	0.94	0.97

Tableau 4.8 : Tableau comparative entre les meilleurs rappel et précision des méthodes. (par une vidéo de base de donnée).

➤ Les valeurs de rappel et de précision dans ce tableau sont obtenu par la moyenne de deux valeurs de rappel et de précision .

➤ D'après les(tableau4.7 et tableau 4.8) nous observons que les valeurs de rappel et de précision sont presque identique entre la méthode de codebook et codebook avec un seul mot de code. Contrairement à la méthode de la différence frame où on voit que son rappel est faible par rapport les deux d'autre méthodes, elle a des zones mal détectées .

4.4.6.2 Comparaison sur le temps d'exécution :

Les tests ont été réalisé sur un processeur intel core i3-380M, la vitesse d'exécution dépend donc de ces caractéristiques. Pour que ces vitesses soient comparables tous les algorithmes testées ont été codé en python, on trouve dans le tableau (tableau 4.10) le temps d'exécution de trois méthodes par une vidéo filmée et vidéo de base de donnée .

Méthode Vidéo (s)	Différence frame	Codebook avec un seul mot de code	Codebook
Vidéo filmée (00:00:05)	61,49(s)	1165,36(s)	1210,36(s)
Vidéo base de donnée (00:00:04)	40,61 (s)	776,75(s)	808,95(s)
Nombre de frame d'apprentissage	/	/	20

Tableau 4.9 : Tableau comparative entre les temps d'exécutions.

➤ D'après le tableau nous observons que la vitesse d'exécution est plus rapide avec différence frame , puis codebook avec un seul mot de code , puis codebook.

4.4.6.3. Qualité de suivi

Après le classement des pixels et les calculs de rappel et de précision on peut donc mesurer la qualité de détection de chaque méthode, l'idéal est d'avoir un rappel et un précision proche de 1. On remarque que la détection de la méthode de différence frame est moyenne généralement elle a des zones dans les objets en mouvement sont mal détecter.

La qualité de détection est presque identique entre codebook et codebook avec un seul mot de code.

4.4. Conclusion

Dans ce chapitre nous avons présenté les résultat de ces trois algorithmes, où nous avons vu que pour obtenir de bons résultats il faut faire plusieurs testes avec différents paramètre .

D'après les résultats nous concluons que la qualité de suivi est mieux avec codebook et codebook avec un seul mot de code et moyenne par différence frame. mais le temps d'exécution la différence frame est meilleur .

Conclusion générale

Conclusion générale

Dans ce mémoire nous avons proposés trois algorithmes de suivi et détection des objets en mouvement. Le but principal de ce mémoire est d'implémenter une modélisation des méthodes de soustraction de fond capable de suivi des objets en mouvement d'une séquence vidéo .

Comme nous avons vu dans le deuxième chapitre chaque algorithme de soustraction de fond consiste à suivre quatre étapes pour atteindre à la fin la détection des objets. L'étape la plus importante est de modéliser un fond pour être une référence où chaque nouvelle image va être comparé à ce fond pour décider les pixels qui appartiennent au fond ou qui appartiennent au premier plan.

Nous avons choisi l'environnement python pour implémenter nos méthodes ce langage a une bibliothèque qui traite les vidéos est << opencv>>. Nous avons implémenté trois méthodes sur une vidéo de base de donnée cvlab et une vidéo filmée .

Les résultats des tests qui ont été effectués sur des vidéos montrent que les algorithmes fonctionnent bien dans les zones qui n'ont pas des ombres, ont une luminosité forte et les couleurs de fond sont éloignés à les couleurs des objets. Les méthodes que nous avons implémenté sont acceptables mais le temps d'exécutions est grand surtout avec l'algorithme de codebook et codebook avec un seul mot de code .

Enfin ,malgré la diversité des algorithmes de soustraction de fond pour le suivi ,il y a plusieurs défis où chaque algorithme doit être capable de les résoudre. comme le changement de luminosité et l'environnement dans lequel est dédié.

Bibliographie

- [1] Nesrine, Brougui Manel et Boumaraf. Détection et suivi d'un objet suspect dans le contexte de vidéosurveillance. *Faculté des Nouvelles Technologies de l'Information et de la Communication Département d'Informatique et Technologie de l'information*. s.l. : Université Kasdi Merbah -Ouargla-, 2014 – 2015 .
- [2] Mokhtari, Djamilia. "Détection des chutes par calcul homographique ". Université de Montréal Département d'Informatique et de Recherche Opérationnelle Faculté des Arts et des Sciences : s.n., Août 2012.
- [3]Harsha VarwaniHeenaChoithwani Tina GyanchandaniDashrathMane," Understanding various Techniques for Background Subtraction and Implementation of Shadow Detection "in *Harsha Varwani et al, Int.J.ComputerTechnology&Applications, Vol 4 (5),822-827, Sept-Oct 2013*
- [4] B.P.L. Lo and S. Velastin, "Automatic congestion detection system for underground platforms," in *Intelligent Multimedia, Video and Speech Processing, 2001. Proceedings of 2001 International Symposium on*, 2001, pp. 158–161
- [5]Sen-Ching, S. Cheung and C. Kamath, 0000. Robust techniques for background subtraction in urban traffic video Center for Applied Scientific Computing Lawrence Livermore National Laboratory 7000 East Avenue, Livermore, CA 94550.
- [6]SudheerReddyBandi ,A.Varadharajan , M.Masthan," Performance Evaluation of Various Foreground Extraction Algorithms for Object detection in Visual Surveillance". *International Journal Of Computational Engineering Research (ijceronline.com) Vol. 2 Issue. 5.*
- [7] N. Tronson1 Y. Goyat1 D. Gruyer ," Comparaison de methodes d'extraction fond/forme pour des scènes de circulation routière",LCPC (Laboratoire Central des Ponts et Chaussées) ' Route de Bouaye, BP 4129, 44341 Bouguenais – FRANCE
- [8] Er. Monica Goyal, " MOVING OBJECT DETECTION USING BACKGROUND SUBTRACTION TECHNIQUES" INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY January, 2016

- [9] Kyungnam Kim, Thanarat H. Chalidabhongse, David Harwooda, Larry Davis "Real-time foreground-background segmentation using codebook model".
- [10] Mikael A. Mousse , Eugene C. Ezin and Cina Motamed "Foreground-Background Segmentation Based on Codebook and Edge Detector" .Unité de Recherche en Informatique et Sciences Appliquées 'Institut de Mathématiques et de Sciences Physiques 'Université d'Abomey-Calavi, Benin BP 613 Porto-Novo.
- [11] Rafael Rodriguez-Gomez · Enrique J. Fernandez-Sanchez · Javier Diaz · Eduardo Ros" Codebook hardware implementation on FPGA for background subtraction " JOURNAL OF REAL-TIME IMAGE PROCESSING · MARCH 2012.
- [12]N. Tronson¹ Y. Goyat¹ D. Gruyer² "Comparaison de méthodes d'extraction fond/forme pour des scènes de circulation routière",LCPC (Laboratoire Central des Ponts et Chaussées) Route de Bouaye, BP 4129, 44341 Bouguenais – FRANCE.

Webographie

- [13] (http://www.memoireonline.com/01/13/6765/m_Etude-et-mise-en-place-d-un-systeme-de-videosurveillance-Cas-de-l-immeuble-Folepe--Bali12.html).
- [14]http://e-cours-arts-plastiques.com/wp-content/uploads/2014/01/chance-meetingduanemichals_thumb.jpg.
- [15] (<http://tpemultimedia.e-monsite.com/pages/qu-est-ce-que-la-video-surveillance.html>).
- [16]https://encryptedtbn3.gstatic.com/images?q=tbn:ANd9GcTSqHfR572uovFCOC-3uRiRa0oP1SX4ImQsRvIZ_-kQu_UJMv5Xuw.
- [17] <http://www.teb-online.com/fr/technologies/videosurveillance-haute-definition-hd.php>.
- [18]https://www.google.dz/search?q=Les+syst%C3%A8mes+de+vid%C3%A9osurveillance+intelligente&source=lnms&tbn=isch&sa=X&ved=0ahUKEwjZ5dHlwMvMAhUCtxQKHURU0CikQ_AUIBygB&biw=1366&bih=667
- [19]https://www.google.dz/search?q=suivi+un+objet&biw=1366&bih=667&source=lnms&tbn=isch&sa=X&ved=0ahUKEwiNt8jh7aLMAhXJFJoKHW6MC1cQ_AUIBygC#imgcr=_.
- [20] <http://img.phonandroid.com/2014/09/google-detection-objet.jpg>.
- [21] <http://tpemultimedia.e-monsite.com/pages/qu-est-ce-que-la-video-surveillance.html>.