

Zedadia

17/004.4-11

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique



Mémoire de Fin d'études Master

Filière : Informatique

Option : Ingénierie des Medias

12/802

Thème :

Utilisation des mixtures de gaussiennes pour la détection et le suivi des objets en mouvement en temps réel dans une vidéo de surveillance

Encadré Par :

Mr.Farou Brahim

Présenté par :

Cheraitia Meryem

Mahboubi Majda

Juin 2012

Résumé

Notre travail de fin d'études entre dans le contexte de la réalisation d'un système informatique de tracking qui est un procédé de localisation d'un (ou plusieurs) objet en mouvement en temps réel en utilisant une camera fixe. Le rôle d'un algorithme de tracking est d'analyser les photogrammes dans l'ordre et d'estimer les paramètres de mouvement. Ces paramètres caractérisent la localisation de la cible.

Dans la littérature, la méthode la plus utilisée pour la modélisation de l'environnement est la mixture de gaussiennes qui est souvent jugée comme la plus efficace. Nous allons à travers de ce projet d'étudier cette méthode et de proposer des améliorations pour un meilleur rendement en temps réel.

En particulier, pour cette méthode d'estimation du fond, la modélisation du fond peut se faire en considérant l'histoire de chaque pixel sur n frames comme un mélange de courbes gaussiennes. Cette approche permet de modéliser des fonds complexes, mais ceux-ci se révèlent sensibles aux ombres importantes et face à de rapides variations du fond, l'approche est efficace, mais nécessite de traiter plusieurs Frames dans une étape d'apprentissage avant de pouvoir décider à quel groupe appartient le pixel au fond ou bien à la forme.

En effet, le système comprend trois algorithmes: le premier est celui de soustraction du fond utilisant la méthode de mixture gaussienne et l'algorithme de K-Means pour classifier les pixels comme appartenant à l'arrière plan ou bien aux objets, le deuxième de post-traitement pour mieux représenté les objets détectés et améliorer la qualité de détection, et le dernier c'est l'algorithme de suivi qui a pour but de relier les régions du même objet dans des frames différents pour avoir une trajectoire de mouvement de l'objet donné.

Mots Clés :

Vidéo surveillance, vision par machine, extraction de l'arrière plan, mixtures de gaussiennes, traitement d'images, détection de mouvement, systèmes temps réel, l'algorithme des k-means.

Remerciement

Nous tenons à remercies en premier lieu Allah qui nous a donné vie et santé pour le parachèvement de ce modeste travail.

Nous remercions après de tout cœur notre encadreur Mr Farou Brahim pour son soutien, sa sympathie, ses encouragements, la confiance qu'il nous témoignée en acceptant de diriger ce travail et pour avoir mis à notre disposition ses conseils pour une meilleure maitrise du sujet.

Nous remercions aussi les membres du jury pour l'honneur qu'ils me font en acceptant de juger ce travail.

Nous remercions également nos familles qui nous ont toujours donné la possibilité de faire ce que nous voulions durant nos études et qui ont toujours cru en nous.

En fin, nous remercions tous ceux qui ont contribué à ce travail par leurs remarques, leurs suggestions et leurs soutiens.

Cheraitia Meryem

Mahboubi Majda

0009

Dédicace

Je dédie ce modeste travail de mes années d'étude :

A mon exemple supérieur dans ma vie mon cher père qui m'a toujours servi de modèle et des baux principes pour une vie idéale par sa patience, ses conseils, sa compréhension et pour m'avoir donné la possibilité de faire ce que je voulais et pour leur soutien sans faille tout au long de mes années d'étude.

A ma très chère mère qui m'a assuré un soutien inconditionnel par ses encouragements sur tous les plans, sa disponibilité, leur affection et leur grand amour aussi.

A tous mes frères et sœurs que j'aime beaucoup et qui n'ont jamais cessé de renouveler la confiance qu'ils ont en moi, Khaled, Rayane et mon cher le benjamin Ammar, que Dieu vous protège et vous prête bonne santé et longue vie.

A mon agréable binôme Meryem, en témoignage de mon amour, de mon profond respect et de ma reconnaissance

A mon intime amie ou plutôt ma sœur Souad et tous mes amies : Hannane, Ibtissem, Oumaima, Sara, Jiji, Ahlem, Asma, Fatma, Chahrazed, Nesrine, Zeyneb, Soumia, Saida, Halima, Khadija, Meryem, Chourouk, Fayza, Halouma, Sandra, Imene Pour votre courage, Que Dieu vous Protège.

A mon groupe de révision : Meryem, Ibtissem, Jiji, Nardjisse, Housseem, Amine, Rachid, Azzedine, Walid, Adam et à toute la promotion 2012.

A mes cousines : Mira, Fadia, Halouma, Mina, Sara, Hanane, Izdihar, Amal et surtout l'adorable Assil

Un dédicace très spéciale à mon oncle maternel, MILI ALI, en témoignage de mon respect et ma gratitude et à toute ma famille et tous ceux qui me sont chers.

Merci pour votre soutien.

Mahboubi Majda

Dédicace

Je dédie ce modeste travail de mes années d'étude :

A mon exemple supérieur dans ma vie mon cher père qui m'a toujours servi de modèle et des bons principes pour une vie idéale par sa patience, ses conseils, sa compréhension et pour m'avoir donné la possibilité de faire ce que je voulais et pour leur soutien sans faille tout au long de mes années d'étude.

A ma très chère mère qui m'a assuré un soutien inconditionnel par ses encouragements sur tous les plans, sa disponibilité, leur affection et leur grand amour aussi.

A tous mes frères et sœurs que j'aime beaucoup et qui n'ont jamais cessé de renouveler qu'ils ont en moi, et surtout Salim, Housseem, Abd Allah, Mohamed, Salah, Walieb, Kamel mes sœur Saliha et la belle Lamia que je l'adore et son mari Abd Elhafid et bien sûr leur princesse Malak, je n'oublie pas le petit de la famille Adam.

A mon agréable binôme Majda, mon intime Sara, mes amies Hannane, Ibtissem, Oumaima, Sandra, Zeyneb, Imene, Ahlem, souad, Soumia, Saida, Amina, Halima, Khadija, Meryem, Fayza, et ma cousine Soltana.

A mon groupe de révision : Majda, Ibtissem, Jiji, Nardjisse, Housseem, Rachid, Cherif, Azzedine, Walid, Adam, Amine.

Et bien sûr je n'oublie pas mes amis de facebook : Sara, Toufik, Meriem, Hana, Yacine, Nadia, Abdou, Marwa.

Et à toute la promotion 2012.

Cheraitia Meryem

Sommaire

Sommaire

Introduction générale.....	06
CHAPITRE 1 : Introduction aux images et vidéo	
1. Introduction.....	07
2. Les images.....	07
2.1. Qu'est ce qu'une image numérique.....	07
2.2. caractéristiques d'une image	08
2.2.1. Dimension « définition ».....	08
2.2.2. Dynamique.....	08
2.2.3. Résolution.....	10
2.2.4. Luminance.....	10
2.2.5. Contraste.....	11
2.2.6. Taille du Fichier Image	11
2.2.7. Format	11
2.3. Domaines d'application des images numériques	11
2.4. Traitement des images numériques	12
3. Les vidéos numériques	13
3.1. Qu'est qu'une vidéo	13
3.2. caractéristiques d'une vidéo	14
3.2.1. Nombre d'armatures par seconde « Taux de vue ».....	14
3.2.2. Débit.....	14
3.2.3. Entrelacement	15
3.2.4. Résolution d'affichage	15
3.2.5. Allongement	16
3.2.6. Qualité vidéo.....	16
3.3. Domaines d'utilisation des vidéos.....	17
4. Les systèmes en temps réel	18
4.1. Définition de Temps réel	18
4.2. Définition d'un système temps réel (STR)	18
4.3. Validité d'un programme TR	18
4.4. Fonctionnement	19
4.5. caractéristiques des applications temps réel	19
4.5.1. Utilisation du temps concret	19
4.5.2. Découpage en tâches	19
4.5.3. Respect des contraintes temporelles.....	20
4.5.4. Ordonnement.....	20
4.6. Les Limites	20
4.7. Catégories des systèmes temps réel	20
4.8. Exemples d'applications temps réel.....	21
5. Conclusion.....	21

CHAPITRE 2 : *La détection et le suivi des objets dans une séquence vidéo*

1. Introduction.....	22
2. les méthodes de détection de mouvement	22
2.1. Détection sans modélisation de l'arrière-plan	23
2.1.1. Flot optique	23
2.1.2. Dérivée temporelle	23
2.2. Détection avec modélisation de l'arrière-plan	24
2.2.1. Modélisation globale de l'arrière-plan	24
2.2.2. Modélisation semi-locale de l'arrière-plan.....	24
2.2.2.1. Détection par région	24
2.2.2.2. Caractérisation par la texture	25
2.2.3. Modélisation locale de l'arrière-plan	25
2.2.3.1. Modélisation par une image	25
2.2.3.2. Modélisation statistique	26
3. Approche par mixture de gaussiennes	27
3.1. Exemple introductif	27
3.2. Fonction de densité de probabilité	28
3.3. Les algorithmes proposés pour la résolution du problème de mélange de gaussiennes ...	29
3.4. Modèle de mixture de gaussiennes généralisées (MGG).....	30
4. Les difficultés susceptibles de diminuer la qualité d'un suivi.....	32
4.1. Changement d'illumination.....	32
4.2. Changements d'échelle.....	32
4.3. Vidéo de basse qualité	32
4.4. Mouvement de caméra	32
4.5. Occultations.....	32
4.6. Déformation de l'objet.....	33
4.7. Objet de petite taille.....	33
4.8. Objet en mouvement rapide.....	33
4.9. Camouflage.....	33
4.10. Ombre	33
4.11. Contrainte de temps réel.....	33
5. Conclusion	34

CHAPITRE 3 : *Conception du système*

1. Introduction.....	35
2. Objectif du système	35
3. Architecture du système	36
4. Réalisation de l'architecture	38

4.1. Phase d'acquisition et de fragmentation.....	38
4.1.1 Acquisition	38
4.2.2. Fragmentation.....	39
4.2. Phase de traitement.....	39
4.2.1. Conversion d'image en niveau de gris	39
4.2.2. Apprentissage	40
4.2.3. Extraction d'arrière-plan.....	40
4.2.4. Modèle GMM.....	41
4.3. Phase Localisation d'objet en mouvement	42
4.3.1. Post traitement	43
4.3.1.1. Filtrage morphologique	43
4.3.1.2. Elimination des points isolés	44
4.3.2. Suivi d'objets	45
4.3.3. Affichage.....	45
5. Conclusion	45

CHAPITRE 4 : *Implémentation du système*

1. Introduction.....	46
2. Présentation des outils de développement.....	46
2.1. JDK «Java Development Kit ».....	46
2.2. Netbeans.....	46
2.3. JMF « Java Media Framework »	47
3. Description du système.....	47
3.1. résultats du test avec changement des paramètres.....	48
3.2. Les résultats de la mixture gaussienne	50
3.3. Les résultats de post traitement	52
3.4. Les résultats de suivi avec encadrement	55
4. Conclusion.....	58
Conclusion générale.....	59
Bibliographie.....	61

La liste des figures

La liste des figures

Liste des figures

Figure 2.1 : Taxinomie proposée pour la détection de mouvement.....	22
Figure 2.2 : diagramme en bâtons.....	28
Figure 3.1 : Architecture du système.....	37
Figure 4.1 : Détection et segmentation d'objet dans un environnement simple.....	50
Figure 4.2 : détection et segmentation d'un objet qui s'intègre à l'arrière-plan dans un environnement simple.....	50
Figure 4.3 : Détection et segmentation d'objet dans un environnement avec faible luminance.....	51
Figure 4.4 : Détection et segmentation d'objet dans un environnement avec lumière variante.....	51
Figure 4.5 : Problème d'ombre pour la détection et segmentation d'objet dans un environnement avec forte lumière.....	51
Figure 4.6 : détection et segmentation de plusieurs objets en mouvement.....	52
Figure 4.7 : Délation d'objet dans un environnement simple.....	52
Figure 4.8 : délétion d'objet qui s'intègre à l'arrière-plan dans un environnement simple.....	53
Figure 4.9 : Dilatation d'objet dans un environnement avec faible luminance.....	53
Figure 4.10 : Problème d'ombre pour la dilatation d'objet dans un environnement avec forte lumière.....	54
Figure 4.11 : Délétion d'objet dans un environnement avec lumière variante.....	54
Figure 4.12 : Délétion de plusieurs objets en mouvement.....	54
Figure 4.13 : Encadrement d'objet dans un environnement simple.....	55
Figure 4.14 : Encadrement d'objet qui s'intègre à l'arrière-plan dans un environnement simple....	55
Figure 4.15 : Encadrement d'objet dans un environnement avec faible luminance.....	56
Figure 4.16 : Problème d'ombre pour l'encadrement d'objet dans un environnement avec forte lumière.	56
Figure 4.17 : Encadrement d'objet dans un environnement avec lumière variante.....	57
Figure 4.18 : Encadrement de plusieurs objets en mouvement.....	57

La liste des tableaux

Liste des tableaux

Tableau 1.1 : Quelque standard de codage de la profondeur.....	10
Tableau 2.1 : Tableau des résultats.....	27
Tableau 4.1 : Résultats du test avec changement des paramètres.....	48

Introduction générale

Introduction générale

Introduction générale

Introduction générale

Ces dernières années, le domaine du traitement et de l'analyse numérique des images a connu un essor considérable engendrant un nombre conséquent de travaux de recherche.

En effet, la récente expansion des possibilités des ordinateurs a grandement facilité le traitement de masse d'informations numériques. De gigantesques quantités de calculs peuvent désormais être traitées dans des temps toujours plus courts. Des traitements plus complexes sont alors accessibles, ce qui ouvre considérablement les perspectives.

L'image est aujourd'hui un objet des plus importants dans notre société. Elle est un média d'information (journaux papiers et télévisuelles, internet), publicitaire, artistique (cinéma, photographie,...) et social (Facebook, Picasa,...) parmi les plus utilisés

Beaucoup de travaux visent donc à en améliorer et en faciliter l'accessibilité. Que ce soit au niveau de son acquisition (appareil photographique, caméra numérique, webcam), de sa capacité de stockage et d'échange (formats, compression, ...) ou de son édition par des logiciels tels que Photoshop ou Gimp. Mais l'image est aussi un moyen de récupérer de l'information. Une machine doit souvent être la plus autonome possible. Il faut minimiser le travail de l'utilisateur. Pour cela, la machine doit connaître l'environnement qu'elle manipule. Elle prend alors l'information nécessaire par l'intermédiaire d'un certain nombre de capteurs.

L'information visuelle donnée par une caméra en fait partie. Elle a notamment l'avantage de ne pas nécessiter de contact avec l'objet étudié. Il faut alors tirer l'information pertinente de l'image.

La reconnaissance et la segmentation des éléments d'une classe sont alors un domaine très important du traitement de l'image. Elles permettent de reconnaître, de localiser et d'isoler l'objet d'intérêt. Ces informations sont nécessaires aux fonctionnements de nombreuses applications.

Ainsi, depuis que le traitement des données vidéo en temps réel est devenu sérieusement envisageable et avec la généralisation de l'utilisation d'images numériques, la détection du mouvement dans les vidéos s'est révélée être un outil indispensable pour des applications aussi diverses que la compression et l'indexation vidéo, les applications propres à

la robotique (détection d'obstacles, guidage de missile, ...) ou encore celles liées à la télésurveillance (intrusion, trafic routier ou poursuites de cibles).

Pour cela, nous avons organisé notre travail en quatre parties :

- La première partie constitue une introduction aux notions des images, vidéos leurs caractéristiques, traitements et domaines d'application, elle contient aussi un aperçu général sur les systèmes en temps réel.
- La deuxième partie est consacrée à un état de l'art sur les différentes méthodes de suivi des objets en mouvement, ainsi que les problèmes qui rencontrent ce suivi.
- La troisième partie est consacrée à la conception de notre système, nous commençons par les objectifs de notre système, ensuite nous présentons son architecture générale, en détaillant les processus de fonctionnement de chaque partie de l'architecture.
- La quatrième partie est l'implémentation du système, où nous allons présenter des résultats obtenus après un ensemble de tests dans différents environnements. Nous clôturons ce mémoire par une conclusion générale.

chapitre 01

Introduction aux images et vidéos

Introduction aux images et vidéos

1. Introduction :

Aujourd'hui, on distingue généralement plusieurs grandes familles «d'images animées» : le cinéma qui consiste à stocker sur une pellicule la succession d'images en négatif où la restitution du film se fait alors grâce à une source lumineuse projetant les images successives sur un écran, et la vidéo analogique qui représente l'information comme un flux continu de données analogiques et qui est destinée à être affichée sur un écran de télévision.

Il existe plusieurs normes pour la vidéo analogique, les trois principales sont : PAL, NTSC, SECAM. La vidéo numérique consiste à coder la vidéo en une succession d'images numériques.

Dans ce premier chapitre, nous allons présenter un petit résumé sur les vidéos et les images numériques : leurs définitions, caractéristiques, domaines d'utilisations, ainsi que quelques notions concernant le traitement d'image et l'analyse de la vidéo.

2. Les images :

Avec la parole, l'image constitue l'un des moyens les plus importants qu'utilise l'homme pour communiquer avec autrui. C'est un moyen de communication universel dont la richesse du contenu permet aux êtres humains de tout âge et de toute culture de se comprendre.

C'est aussi le moyen le plus efficace pour communiquer, chacun peut analyser l'image à sa manière, pour en dégager une impression et d'en extraire des informations précises.

De ce fait, le traitement d'images est l'ensemble des méthodes et techniques opérant sur celles-ci, dans le but de rendre cette opération possible, plus simple, plus efficace et plus agréable, d'améliorer l'aspect visuel de l'image et d'en extraire des informations jugées pertinentes.[S2]

2.1. Qu'est ce qu'une image numérique?

L'appellation « image numérique » désigne, dans son sens le plus général, toute image (dessin, icône, photographie...) créée, traitée et stockée sous forme binaire, acquise par des convertisseurs analogique-numérique situés dans des dispositifs comme les scanners, les appareils photos ou les caméscopes numériques, les cartes d'acquisition vidéo (qui numérisent

directement une source comme la télévision), créées directement par des programmes informatiques, grâce à une souris, des tablettes graphiques ou par de la modélisation 3D « images de synthèse », traitée grâce à des outils informatiques, de façon à la transformer, à en modifier la taille, les couleurs, d'y ajouter ou d'en supprimer des éléments, d'y appliquer des filtres variés, etc., stockés sur un support informatique (disquette, disque dur, CD-ROM...). [S2]

Les images numériques ne sont pas formées de zones dans lesquelles la luminosité et les couleurs varient de façon continue, mais de milliers ou de millions de petits carrés de teinte uniforme appelé pixels, contraction de l'expression anglaise « picture elements », autrement dit « éléments d'image ».

Le pixel représente ainsi le plus petit élément constitutif d'une image numérique. Les pixels sont représentés par des éléments informatiques appelés bits. Sur l'écran d'un ordinateur, ils sont visibles quand on agrandit fortement l'image. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image. [S1]

2.2. caractéristiques d'une image :

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants:

2.2.1. Dimension « définition »:

C'est la taille de l'image, elle est définie par le nombre de pixels qui composent l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes donne le nombre total de pixels dans une image. [MB, 05]

2.2.2. Dynamique :

La dynamique d'une image correspond à l'étendue de la gamme de couleurs ou de niveaux de gris que peuvent prendre les pixels. C'est une notion liée au nombre d'octets utilisés pour stocker l'information teinte de gris ou couleurs. Définie aussi comme le nombre de niveaux de gris d'une image, elle s'exprime en bit c.-à-d. en une suite de 0 et 1.

Une image est donc représentée par un tableau à deux dimensions dont chaque case est un pixel. La valeur stockée dans une case est codée sur un certain nombre de bits déterminant la couleur ou l'intensité du pixel, on l'appelle profondeur de codage (parfois profondeur de couleur). Il existe plusieurs standards de codage de la profondeur :

- **Bitmap noir et blanc** : en stockant un bit dans chaque case, il est possible de définir deux couleurs (noir ou blanc).
- **Bitmap 16 couleurs ou 16 niveaux de gris** : en stockant 4 bits dans chaque case, il est possible de définir 16 possibilités d'intensités pour chaque pixel, c'est-à-dire 16 dégradés de gris allant du noir au blanc ou bien 16 couleurs différentes. [S3]
- **Bitmap 256 couleurs ou 256 niveaux de gris** : en stockant un octet dans chaque case, il est possible de définir 256 intensités de pixels, c'est-à-dire 256 dégradés de gris allant du noir au blanc ou bien 256 couleurs différentes. [S3]
- **Palette de couleurs (colormap)** : grâce à cette méthode, il est possible de définir une palette, ou table des couleurs, contenant l'ensemble des couleurs pouvant être contenues dans l'image, à chacune est associé un indice. Le nombre de bits réservé au codage de chaque indice de la palette détermine le nombre de couleurs pouvant être utilisées. Ainsi en codant les indices sur 8 bits il est possible de définir 256 couleurs utilisables, c'est-à-dire que chaque case du tableau à deux dimensions représentant l'image va contenir un nombre indiquant l'indice de la couleur à utiliser. On appelle ainsi image en couleurs indexée une image dont les couleurs sont codées selon cette technique. [S3]
- **Truecolor**: permet de représenter une image en définissant chacune des composantes (RGB: rouge, vert, bleu). Chaque pixel est représenté par un entier comportant les trois composantes, chacune codée sur un octet, c'est-à-dire au total 24 bits (16 millions de couleurs) Il est possible de rajouter une quatrième composante permettant d'ajouter une information de transparence ou de texture, chaque pixel est alors codé sur 32 bits. [S3]

Mode	Nbre de valeurs	Dynamique en Bits
Noir et Blanc	2	1
Niveaux de gris	256 0 noir – 255 blanc	8
	65536	16
Couleur RVB	16 millions	24 8 bits x 3 couches
Couleur Indexe	256	8 bits

Tableau 1.1 : Quelque standard de codage de la profondeur

2.2.3. Résolution :

La résolution d'une image est définie par le nombre de pixels par unité de surface (classiquement en dpi (dots per inches) ou ppp (points par pouce)). Un pouce représentant 2.54 cm. La résolution permet ainsi d'établir le rapport entre le nombre de pixels d'une image et la taille réelle de sa représentation sur un support physique. Elle est définie lors de la numérisation et dépend principalement des caractéristiques du matériel utilisé lors de processus de numérisation. Plus le nombre de pixels est élevé par unité de longueur de la structure à numériser, plus la quantité d'information qui décrit cette structure est importante et plus la résolution est élevée. La résolution d'une image numérique définit le degré de détail qui va être représenté sur cette image. [MB, 05]

2.2.4. Luminance :

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance se caractérise par :

- Des images lumineuses (brillantes).
- Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir, ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- L'absence de parasites. [BA, 10]

2.2.5. Contraste :

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images.

Si L_1 et L_2 sont les degrés de luminosité respectivement de deux zones voisines A_1 et A_2 d'une image, le contraste C est défini par le rapport : $C = \frac{L_1 - L_2}{L_1 + L_2}$. [BA, 10]

2.2.6. Taille du Fichier Image :

C'est le nombre de pixels multiplié par le nombre d'octets nécessaires au codage d'un seul pixel. Les images brutes codifient en général chaque pixel sur un octet; certaines images dédiées à un type précis de carte graphique codifient le pixel sur un demi-octet ou même sur un seul bit (format d'image noir et blanc binaire). Par contre les images naturelles de très haute qualité sont codées sur 3, 4, voire 6 octets par pixel. Ces fichiers sont donc très volumineux et subissent une compression pour le stockage et la transmission. [S4]

Taille du fichier = Nbre Pixels Image x Dynamique.

2.2.7. Format :

C'est une spécification ou une norme qui définit la manière de coder les pixels d'une image dans un fichier on trouve :

- Format pour la diffusion (web, Impression): Jpeg, Jif, Png.
- Format pour l'archivage : Tiff, Bmp. [S5]

2.3. Domaines d'application des images numériques:

L'image numérique est omniprésente, elle trouve son application dans différents domaines classer en deux catégories : grand public et application industriels.

- **Grand public :**
 - Appareil photo, téléphone.
 - Télévision, Home Cinéma.
 - Cinéma (amateur et professionnel).

- Informatique Images réelle et de synthèse.
- **Applications industrielles (2D et 3D) :**
 - Imagerie satellitaire, guidage.
 - Radiologie, Biologie, génomique.
 - Contrôle industriel, Robotique (nucléaire).
 - Télésurveillance, biométrie, police scientifique (reconnaissance).
 - Post-traitement (audiovisuel).
 - Analyse du geste sportif.

2.4. Traitement des images numériques:

Le traitement d'images désigne une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information.

Il s'agit d'un sous-ensemble du traitement du signal dédié aux images et aux données dérivées comme la vidéo (par opposition aux parties du traitement du signal consacrées à d'autres types de données : son et autres signaux monodimensionnels notamment), tout en opérant dans le domaine numérique (par opposition aux techniques analogiques de traitement du signal, comme la photographie ou la télévision traditionnelle).

Dans le contexte de la vision artificielle, le traitement d'images se place après les étapes d'acquisition et de numérisation, assurant les transformations d'images et la partie de calcul permettant d'aller vers une interprétation des images traitées. Cette phase d'interprétation est d'ailleurs de plus en plus intégrée dans le traitement d'images, en faisant appel notamment à l'intelligence artificielle pour manipuler des connaissances, principalement sur les informations dont on dispose à propos de ce que représentent les images traitées (connaissance du « domaine »). [RR, 07]

La compréhension du traitement d'images commence par la compréhension de ce qu'est une image. Le mode et les conditions d'acquisition et de numérisation des images traitées conditionnent largement les opérations qu'il faudra réaliser pour extraire de l'information. En effet, de nombreux paramètres entrent en compte, les principaux étant :

- la résolution d'acquisition et le mode de codage utilisé lors de la numérisation, qui déterminent le degré de précision des éventuelles mesures de dimensions.
- les réglages optiques utilisés (dont la mise au point) qui déterminent par exemple la netteté de l'image.
- les conditions d'éclairage, qui déterminent une partie de la variabilité des images traitées.
- le bruit de la chaîne de transmission d'image. [RR, 07]

3. Les vidéos numériques :

L'histoire du numérique dans la vidéo commence véritablement de 1972 à 1982. À l'origine équipements de synchronisation, les appareils se sophistiquèrent avant d'entrer dans le milieu professionnel. Dès lors, les industriels prirent conscience de l'avènement de ce nouveau phénomène et présentèrent des normes en matière de numérisation. Une certaine anarchie numérique régna alors sur le marché ce qui força la main au CCIR (Comité consultatif international des radiocommunications) à normaliser un format vidéo en composantes numériques compatible dans le monde entier : cette norme c'est le 4:2:2, ou CCIR 601. Elle spécifie les paramètres de codage de signaux à numériser (échantillonnage, quantification...) Dès lors les innovations ne cessèrent de s'enchaîner pour permettre aujourd'hui à la vidéo numérique de se généraliser dans les centres de production, chaînes TV et régie de post production pour assister le montage vidéo. [S6]

3.1. Qu'est qu'une vidéo ?

Le mot « vidéo » vient du latin « *vidéo* » qui signifie : « je vois ». Il regroupe l'ensemble des techniques permettant l'enregistrement ainsi que la restitution d'images animées, accompagnées ou non de son, sur un support électronique et non de type photochimique.

Un flux vidéo est composé d'une succession d'images qui défilent à un rythme fixe (par exemple 25 par seconde dans la norme française ou 30 par seconde dans d'autres normes) pour donner l'illusion du mouvement, l'œil humain a comme caractéristique d'être capable de distinguer environ 20 images par seconde. Ainsi, en affichant plus de 20 images par seconde, il est possible de tromper l'œil et de lui faire croire à une image animée.

La fluidité d'une vidéo est caractérisée par le nombre d'images par secondes (en anglais *frame rate*), exprimé en FPS (Frames per second, en français trames par seconde). Chaque image est décomposée en lignes horizontales, chaque ligne étant une succession de points. La lecture et la restitution d'une image s'effectuent donc séquentiellement ligne par ligne comme un texte écrit : de gauche à droite puis de haut en bas. [JCF, 10]

3.2. caractéristiques d'une vidéo :

Les principales caractéristiques de la vidéo sont :

3.2.1. Nombre d'armatures par seconde « Taux de vue » :

Le nombre d'images immobiles par unité de période de vidéo s'étend de six ou huit frames par seconde (*frame/s*) pour de vieux appareils-photo mécaniques à 120 frames ou plus par seconde pour de nouveaux appareils-photo professionnels. Pal (L'Europe, l'Asie, l'Australie, etc.) et SECAM (La France, la Russie, régions de l'Afrique etc.) les normes indiquent 25 *frame/s*, alors que NTSC (les Etats-Unis, le Canada, le Japon, etc.) indique 29.97 *frame/s*. Le film est tiré au taux plus lent d'armature de 24*frame/s*, qui complique légèrement le processus de transférer un film cinématographique à la vidéo. Le taux minimum d'armature pour réaliser l'illusion d'une image mobile est environ quinze frames par seconde. [S7]

3.2.2. Débit :

Le Débit binaire est une mesure du taux de contenu de l'information dans un flux vidéo. Il est quantifié en utilisant l'unité **bit par seconde** (bit/s ou bps) unité ou Mega de bits par seconde (Mbit/s). Un débit binaire plus élevé permet une meilleure qualité vidéo. Par exemple VideoCD, avec un débit binaire d'environ 1 Mbit/s, est de qualité inférieure que DVD, avec un débit binaire d'environ 5 Mbit/s. HDTV a un plus de haute qualité encore, avec un débit binaire d'environ 20 Mbit/s.

Le Débit binaire variable (VBR) est une stratégie pour maximiser la qualité vidéo visuelle et pour réduire au minimum le débit binaire. Sur des scènes de mouvement rapide, un débit binaire variable utilise plus de bits qu'il fait sur des scènes de mouvement lent de la durée semblable pourtant réalise une qualité visuelle cohérente. Pour le temps réel et couler visuel non protégé quand la largeur de bande disponible est fixe, par exemple. Dans la

vidéoconférence fournie sur des canaux de largeur de bande fixe, à débit binaire constant (CBR) doit être utilisé. [S7]

3.2.3. Entrelacement :

La vidéo peut être Entrelacé ou progressif. L'entrelacement a été inventé comme manière de réaliser la bonne qualité visuelle dans les limitations d'une largeur de bande étroite. Les lignes de balayage horizontales de chaque armature entrelacée sont numérotés consécutivement et divisés dans deux champs: champ impair comprendre les lignes impaires et champ égal comprendre les lignes paires. NTSC, pal et SECAM sont des formats entrelacés. Les caractéristiques visuelles abrégées de résolution incluent souvent i pour indiquer l'entrelacement. Par exemple, le format visuel de pal est souvent indiqué As 576i50, où 576 indique la résolution horizontale, i indique l'entrelacement, et 50 indique 50 armatures (de simple-champ) par seconde. [S8]

Dans le balayage progressif les systèmes, chaque armature inclut toutes les lignes de balayage. Le résultat est une résolution perçue beaucoup plus élevée.

Le mode entrelacé est noté i , ce qui donne les notations suivantes en mode vidéo analogique entrelacée :

- deux trames de 240 lignes à 60Hz = 480i@60Hz en NTSC
- deux trames de 288 lignes à 50Hz = 576i@50Hz en PAL [S8]

3.2.4. Résolution d'affichage :

La taille d'une image visuelle est mesurée en Pixel pour la vidéo numérique, ou les lignes de balayage verticales et horizontales de la résolution pour la vidéo analogue. Dans le domaine numérique (par exemple. Télévision de standard-définition de DVD) (SDTV) est indiqué comme 720/704/640×480i60 pour NTSC et 768/720×576i50 pour la résolution de PAL ou de SECAM. Cependant dans le domaine analogue, le nombre de lignes de balayage visibles reste constant (486 NTSC/576 pal), tandis que la mesure horizontale change avec la qualité du signal.

La résolution vidéo pour 3D-video est mesurée par voxels (élément d'image de volume, représentant une valeur dans l'espace tridimensionnel). [S8]

3.2.5. Allongement :

Allongement décrit les dimensions des écrans vidéo et d'éléments images vidéo. Tous les formats vidéo populaires sont rectilignes, et ainsi peuvent être décrits par un rapport à la largeur de la taille. L'allongement d'écran d'un écran traditionnel de télévision est 4:3, soit environ 1.33:1. Les télévisions à haute définition utilisent un allongement de 16:9, soit environ 1.78:1. L'allongement d'une pleine armature de film de 35 millimètres avec la bande sonore (également connue sous le nom de « norme d'académie ») est autour de 2.37:1. [S7]

3.2.6. Qualité vidéo :

Peut être mesurée avec la métrique formelle comme PSNR ou avec qualité visuelle subjective employer l'observation experte.

La qualité visuelle subjective d'un système de traitement visuel peut être évaluée comme suit :

- Choisissez les ordres de vidéo pour employer pour l'essai.
- Choisissez les arrangements du système pour évaluer.
- Choisissez un essai méthode pour la façon présenter les ordres visuels aux experts et rassembler leurs estimations.
- Invitez un nombre suffisant des experts, de préférence pas moins de 15.
- Effectuez l'essai.
- Calculez les marques moyennes pour chacun basé sur les estimations des experts.

[S8]

3.3. Domaines d'utilisation des vidéos:

On peut remarquer aujourd'hui, que la vidéo est omniprésente dans plusieurs domaines et pour plusieurs services où on trouve 2 grands domaines :

La vidéosurveillance qui consiste à placer des caméras de surveillance dans un lieu public ou privé pour visualiser en un endroit centralisé tous les flux de personnes au sein d'un lieu ouvert au public et prévenir vols, agressions et mouvements de foule, et **La vidéoconférence** « visioconférence » qui permet à des personnes de dialoguer en direct à distance, tout en se voyant, par exemple en peut citer : **Télé-enseignement** : e-Learning, **Téléconsultation** : consultation et suivi du patient à distance, **Télé-médico-social** : encadrement du patient maintenu à domicile, **Télé-Radiologie** : interprétation d'examens radiologiques à distance, **Télé-Chirurgie** : opération chirurgicale assistée à distance par ordinateur.

Il existe d'autres domaines tels que:

- **Vidéos sportives**: analyse de gestes et d'actions, statistiques de joueurs, annotation de vidéos, ... etc.
- **Postproduction (cinéma et TV)**: édition, colorisation, enlèvement d'objets, flous sélectifs, ralenti, restauration, stabilisation, ... etc.
- **Graphique** : capture de mouvement (jeux, films et avatars), ... etc.
- **Annotation, recherche, préparation, diffusion et protection de contenus vidéos (et multimédia)** : descripteurs compacts et recherche rapide, recherche d'instance ou de classe d'objets, de scène, d'action, recherche de copies vidéos, découpage en plan, résumé automatique, adaptation des contenus aux périphériques, création et insertion de métadonnées ou de marques, ... etc. [S9]

4. Les systèmes en temps réel :

En informatique temps réel, le comportement correct d'un système dépend, non seulement des résultats logiques des traitements, mais aussi du temps auquel les résultats sont produits.

4.1. Définition de Temps réel :

Le principe du temps réel est de contrôler quelle partie du programme intervient à chaque instant, sans laisser de place à l'aléatoire comme on peut le faire en programmation classique où tout un tas de codes en mémoire peuvent grignoter du temps. [S13]

Contrairement à une croyance commune, il ne s'agit pas d'être "immédiat" mais de maîtriser le temps de réponse, qui peut aller de quelques microsecondes à plusieurs dizaines de millisecondes. [S13]

Il s'agit également de la simulation du système par rapport au fonctionnement d'un autre système, avec la même rapidité que lui. Il peut être qualifié de mou lorsque le délai de réponse est proche de 0,5 seconde, ou dur en cas de temps de réponse plus court. [S10]

4.2. Définition d'un système temps réel (STR) :

Un système temps réel est un système (application ou ensemble d'applications) informatique dont le fonctionnement est assujéti à l'évolution dynamique d'un procédé extérieur qui lui est connecté et dont il doit contrôler le comportement. [SB]

Donc, un système est dit temps réel lorsque l'information après acquisition et traitement reste encore pertinente, ce qui signifie que dans le cas d'une information arrivant de façon régulière (sous forme d'une interruption périodique du système), les temps d'acquisition et de traitement doivent rester inférieurs à la période de rafraîchissement de cette information. [PF, PK, 03]

Il n'est pas un système « qui va vite / rapide » mais un système qui satisfait des contraintes temporelles (les contraintes de temps dépendent de l'application et de l'environnement alors que la rapidité dépend de la technologie utilisée, celle du processeur par exemple). [FS]

4.3. Validité d'un programme TR :

Outre la correction algorithmique, le **temps** intervient dans la **validité** du programme où un retard dans la production d'un résultat est considéré comme une erreur (pouvant entraîner de graves conséquences):

- le temps de réaction doit être adapté aux événements externes
- Le programme doit pouvoir fonctionner en continu en maintenant sa capacité à traiter le flux de données d'entrée.
- Les temps de calcul sont connus (estimés) et peuvent être utilisés dans une analyse de réactivité.

Valider un système temps réel c'est démontrer rigoureusement que le système a le comportement spécifié. [PD, 2002]

4.4. Fonctionnement :

- Les applications déclenchent des événements à occurrence périodique ou aléatoire. Elles imposent au système informatique qui leur est associé de réagir avant un délai fixé (ou à une date donnée).
- L'échelle de temps peut varier selon le contexte : la microseconde pour un radar, la seconde pour une IHM, la minute dans une chaîne de fabrication, l'heure pour une réaction chimique, ... [FC, JD, ZM]

4.5. caractéristiques des applications temps réel :

4.5.1. Utilisation du temps concret :

- Au sein d'une application ou d'un système temps réel, il faut pouvoir manipuler le temps concret (horloge)
- Le temps réel (ou temps concret) sera utilisé de plusieurs façons:
 - ❖ Soit en définissant la date à laquelle une action doit être commencée.
 - ❖ Soit en définissant la date à laquelle une action doit être finie.
- Il peut être nécessaire de pouvoir modifier ces paramètres en cours d'exécution et de pouvoir préciser les actions à prendre en cas de faute temporelle. [SB]

4.5.2. Découpage en tâches :

- Dans le monde réel, l'environnement du système temps réel peut consister en plusieurs actions qui évoluent simultanément (en parallèle ou en concurrence).
- Pour réduire la complexité de conception et calquer fidèlement la réalité, il faut utiliser la programmation concurrente :
 - ❖ utiliser un modèle de tâches ou processus concurrents.

- ❖ utiliser des moyens de communication et de synchronisation inter-tâches ou inter-procès (mémoire partagée, boîtes aux lettres, files de messages, moniteurs, etc.). [SB]

4.5.3. Respect des contraintes temporelles :

La limitation des ressources (en particulier du processeur) conduit à bloquer des processus (ils ne peuvent progresser du fait du manque de ressource)

- Afin de respecter en permanence les échéances, il faut gérer efficacement la pénurie et tenter de favoriser les processus dont l'avancement est le plus « urgent »
- Un ordonnancement consiste à définir un ordre sur l'utilisation des ressources du système afin de respecter les échéances. [SB]

4.5.4. Ordonnancement :

On appelle ordonnanceur (scheduler), le processus système qui gère l'ordonnancement des processus.

- Un algorithme d'ordonnancement est une méthode ou stratégie utilisée pour ordonnancer les processus.
- Un tel algorithme s'appuie sur la connaissance de certaines caractéristiques des processus ou du système :
 - ❖ processus périodiques ou aperiodiques.
 - ❖ Prémption possible ou non.
 - ❖ Échéance et pire temps d'exécution des processus.
 - ❖ système à priorité fixe ou à échéance. [SB]

4.6. Les Limites :

- Temps processeur est passé (perdu) dans de l'attente, de l'observation,...
- La solution c'est de récupérer les temps morts et de les affecter à d'autres activités (tâches)
- Suite la prochaine fois avec le multitâche. [PP]

4.7. Catégories des systèmes temps réel :

Il est évident que la structure de ce système dépendra de ces fameuses contraintes. On pourra diviser les systèmes en deux catégories :

1. Les systèmes dits à contraintes souples ou molles (soft real time). Ces systèmes acceptent des variations dans le traitement des données de l'ordre de la demi-seconde (ou 500 ms) ou la seconde. On peut citer l'exemple des systèmes multimédia : si quelques images ne sont pas affichées, cela ne met pas en péril le fonctionnement correct de l'ensemble du système. Ces systèmes se rapprochent fortement des systèmes d'exploitation classiques à temps partagé. Ils garantissent un temps moyen d'exécution pour chaque tâche. On a ici une répartition **égalitaire** du temps CPU entre processus. [HQ]
2. Les systèmes dits à contraintes dures (hard real time) pour lesquels une gestion stricte du temps est nécessaire pour conserver l'intégrité du service rendu. On citera comme exemples les contrôles de processus industriels sensibles comme la régulation des centrales nucléaires ou les systèmes embarqués utilisés dans l'aéronautique. Ces systèmes garantissent un temps maximum d'exécution pour chaque tâche. On a ici une répartition **totalitaire** du temps CPU entre tâches. [HQ]

4.8. Exemples d'applications temps réel :

- Commandes de procédés, Systèmes embarqués, Guidage de mobiles, Surveillance de centrales nucléaires, Conduite d'expériences scientifiques, Robotique.
- Fourniture d'images et son pour le multimédia, Suivi d'informations boursières.
- Suivi opératoire en milieu médical. [BS]

5. Conclusion :

Dans ce premier chapitre, nous avons présenté une introduction aux images et vidéos numériques qui sont aujourd'hui incontournables et omniprésents dans notre vie quotidienne.

Nous avons donné en premier temps une définition pour ces deux notions suivies par leurs différentes caractéristiques, et enfin, nous avons résumé leurs domaines de traitement et d'utilisation.

chapitre 02

La détection et le suivi des objets dans une séquence vidéo

La détection et le suivi des objets dans une séquence vidéo

1. Introduction :

La détection de mouvement (Motion detection) indique uniquement que l'on parle d'une méthode qui a comme objectif de trouver en quels points de l'image un mouvement a eu lieu. Un algorithme ayant cet objectif fournit en sortie une variable quantitative (quantité de mouvement) ou qualitative (booléenne) pour tout pixel de chaque image d'entrée. Toutes les méthodes présentées ci-après rentrent dans cette catégorie.

Dans ce chapitre nous présentons un état de l'art sur les différentes méthodes de détection de mouvement basées sur la modélisation de l'arrière-plan.

2. les méthodes de détection de mouvement :

La plupart des algorithmes de suivi d'objets, prenant en entrée les images fournies par une caméra fixe effectuent une première étape de détection de mouvement afin de déterminer parmi les pixels de l'image courante lesquels appartiennent à l'arrière-plan de la scène, et lesquels représentent des objets mobiles. Ce domaine de recherche est très actif depuis les débuts de l'analyse de séquences vidéo à la fin des années 1970.

Depuis cette date, le nombre d'articles publiés chaque année sur ce sujet ne cesse de croître, et en particulier depuis le milieu des années 1990, lorsque la puissance des ordinateurs grand public a permis d'envisager sérieusement un traitement en temps réel des données vidéo. Face à cette multitude de méthodes proposées dans la littérature, plusieurs états de l'art ont été publiés afin de tenter d'établir une taxinomie des algorithmes existants.

L'ensemble de ces états de l'art nous permet d'établir une classification hiérarchisée des différentes méthodes de détection de mouvement, ou nous distinguerons quatre grandes familles de méthodes en fonction de la modélisation de l'arrière-plan. [NV, 07]

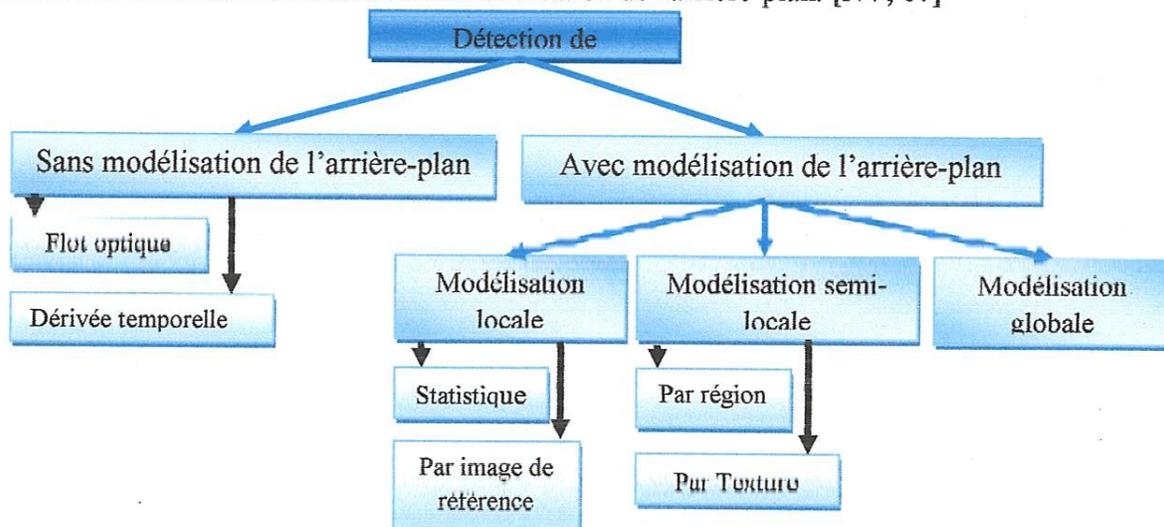


Figure 2.1 : Taxinomie proposée pour la détection de mouvement

2.1. Détection sans modélisation de l'arrière-plan :

Ces méthodes consistent à détecter le mouvement par le calcul en tout point de l'image d'une quantité mathématique qui est fonction de l'intensité ou de la couleur de l'ensemble des pixels et qui est censée refléter l'importance du mouvement visible dans la scène.

2.1.1. Flot optique :

Le flot optique est un champ de vecteurs à deux dimensions correspondant au mouvement perçu au niveau des pixels, permet la mise en œuvre d'une analyse globale du mouvement à l'aide d'une équation reliant la variation d'intensité lumineuse en un point avec la vitesse de déplacement de ce point.

Il n'est pas seulement l'étude de la détermination du champ d'écoulement optique lui-même, mais aussi de son utilisation dans l'estimation de la nature en trois dimensions et la structure de la scène, ainsi que le mouvement des objets 3D et de l'observateur par rapport à la scène.

Son estimation est un intermédiaire essentiel dans de très nombreuses applications n'est pas seulement la détection des objets mobiles : le codage et la compression vidéo, l'aide au guidage de véhicules et jusqu'à certaines applications en mécanique des fluides (Particle Image Velocimetry) ou la mesure des déformations en mécanique des solides.

De nombreuses méthodes ont été proposées, mais quelle que soit la méthode choisie, le calcul du flot optique reste une opération très coûteuse en temps de calcul. [RR]

2.1.2. Dérivée temporelle :

La différence temporelle détecte la région de mouvement dans un champ de vision grâce à la différence de pixel par pixel de deux trames consécutives dans un flux vidéo en mesurant le changement d'apparence de ces pixels en tout point de l'image courante et la précédente. [CT, 07]

Alors, elle consiste à soustraire une image acquise au temps t d'une autre au temps $t+k$, où k est habituellement égal à 1. Ainsi, l'image résultante sera vide si aucun mouvement ne s'est produit pendant l'intervalle de temps observé, car l'intensité et la couleur des pixels seront presque identiques. Par contre, si du mouvement a lieu dans le champ de vue, les pixels

frontière des objets en déplacement devraient changer complètement de valeurs, démontrant alors la présence d'activité dans la scène. [MH, 07]

Cette méthode adapte le changement de la scène. Mais elle est moins d'efficacité face à des phénomènes tels que les mouvements lents ou non réguliers, les arrêts brefs d'un objet en mouvement, ou encore la présence de trames redondantes dans certaines séquences vidéo, par exemple dans une durée du temps Δt , peut être, on détecte seulement une partie d'objet, Et le vide (la région où l'objet s'est déplacé l'autre lieu) est aussi détecté. Dans ce cas-là, c'est très difficile à suivre et extraire les propriétés de mouvement (la taille, la position, la vitesse, etc.). [NV, 07]

En effet, certains traitements supplémentaires sont nécessaires afin de déterminer la zone en mouvement, car l'information disponible ne concerne que les contours des régions en déplacement (ce qui inclut également les zones intérieures d'un objet). [MH, 07]

2.2. Détection avec modélisation de l'arrière-plan :

2.2.1. Modélisation globale de l'arrière-plan :

Ces méthodes utilisent à chaque instant l'ensemble des observations pour construire un modèle de l'ensemble de l'arrière-plan.

La détection de mouvement est effectuée avec chacun des modèles de l'arrière-plan gardés en mémoire et celui qui détecte le moins de pixels en mouvement est retenu pour la décision finale.

L'ensemble de ces modèles est obtenu après une phase d'apprentissage. Les modèles peuvent être mis à jour si une large majorité des pixels de l'image sont détectés comme étant en mouvement. Ces traitements peuvent être réalisés par l'algorithme k-means. [NV, 07]

2.2.2. Modélisation semi-locale de l'arrière-plan :

Pour ces méthodes on s'intéresse aux observations qui ont eu lieu dans un certain voisinage de ce point, ou dans la région de l'image à laquelle il appartient.

2.2.2.1. Détection par région :

À cause des imprécisions dues aux caméras vidéo utilisées ainsi que la nature progressive des mouvements, il est préconisé de prendre en compte l'ensemble des pixels d'un

voisinage au lieu de chercher à détecter les mouvements en un point donné sans prendre en compte les pixels alentour. [NV, 07]

2.2.2.2. Caractérisation par la texture :

Consiste à calculer en tout point, un vecteur caractérisant la texture à cet endroit, et l'ensemble des vecteurs calculés est utilisé comme espace de représentation des données. Le modèle d'arrière-plan est conçu en un lieu donné comme un ensemble de modes dotés d'un poids. [NV, 07]

2.2.3. Modélisation locale de l'arrière-plan :

Méthode très populaire qui forme la plus grande famille de méthodes de détection du mouvement, reposent sur la modélisation du fond qui présente l'avantage de permettre une détection plus précise des objets en mouvement ainsi leur simplicité théorique et leur faible complexité algorithmique. Elle consiste à associer à tout point de l'image une valeur ou une fonction permettant de modéliser l'apparence de l'arrière-plan en ce point. Le modèle d'apparence de l'arrière-plan en un point ne dépend que des observations qui ont eu lieu en ce point.

Les autres pixels de l'image n'interviennent pas. Plusieurs conditions sont nécessaires à l'utilisation de ces techniques. Tout d'abord, la caméra doit être maintenue fixe, il est préférable aussi d'avoir, au début de la séquence, un certain nombre d'images sans objets en mouvement afin de pouvoir apprendre correctement le fond.

Le principe fondamental repose sur une estimation statistique de la scène observée. [AB]

2.2.3.1. Modélisation par une image :

Utilise une image de référence (le fond), représentant les éléments stationnaires de la scène. Ici la mesure de vraisemblance du mouvement est la différence entre la trame courante et le fond. Ces méthodes sont moins dépendantes de la vitesse et de la taille des objets.

Cependant, l'adaptation aux environnements dynamiques est une tâche bien plus ardue, pénalisant la détection des mouvements de faible amplitude (des objets petits, très lents ou très peu contrastés).

Par exemple, en environnement extérieur, les variations d'intensité lumineuse rendent rapidement obsolète un tel modèle, et il est nécessaire de mettre à jour cette image de l'arrière-plan.

Par ailleurs, il n'est pas toujours possible d'obtenir d'une image de la scène totalement vide. Dans ces conditions, il est nécessaire de mettre à jour l'image de l'arrière-plan.

Plusieurs méthodes étaient proposées pour créer l'image de fond :

- Calculer de moyenne de N premières images.
- Choisir l'image qui a peu de changement de scène ou qu'il n'y a pas d'objet mobile pour l'image de fond.
- Choisir la première image dans la séquence de vidéo. [CP, 08]

2.2.3.2. Modélisation statistique :

Il s'agit, pour chaque pixel, d'estimer la probabilité d'y observer telle ou telle intensité en se basant sur un modèle appris, censé représenter l'arrière-plan de la scène. Ce modèle est une description mathématique approximative du mécanisme qui a généré les observations, Il s'exprime généralement à l'aide d'une famille de distributions (« F » : fonctions de densité de probabilité) et d'hypothèses sur les variables aléatoires X_1, \dots, X_n .

Chaque membre de la famille est une approximation possible de F : l'inférence consiste donc à déterminer parmi tous les pixels de l'image lesquels appartiennent à l'arrière plan où pour toute probabilité élevée le pixel est considéré comme arrière-plan sinon il est considéré comme avant plan.

On constate que les algorithmes utilisant une modélisation statistique de l'arrière-plan font apparaître les contours des objets mobiles de manière plus précise. En revanche, ces méthodes sont très sensibles au bruit, et en l'absence de post traitement, les résultats obtenus ne constituent pas une bonne segmentation des objets mobiles. [NVI, 07]

Il a été proposé de modéliser l'intensité des points de l'arrière-plan par une distribution gaussienne. Cette approche sera détaillée dans ce qui suit.

3. Approche par mixture de gaussiennes :

Dans la littérature scientifique, le mouvement est un élément informatif essentiel dans l'image. Techniquement, il s'agit de reproduire, sous forme d'une discrimination binaire noir et blanc, cette information en mettant en blanc les éléments en mouvement à l'avant-plan (foreground), et en laissant en noir les éléments fixes considérés comme constituant l'arrière-plan (background). À partir de là, la problématique centrale est de savoir comment la caméra va organiser dans le temps la hiérarchisation des images précédemment enregistrées pour identifier un élément nouveau. »

C'est donc la question du passé et de son organisation qui constitue la pierre d'achoppement de la recherche. « Dans les systèmes qui existent déjà, les images du passé sont modélisées sous une forme gaussienne. C'est-à-dire que le système considère un nombre d'images antérieures, par exemple 100, et les organise sous la forme d'une courbe de Gauss. Celle-ci construit sa moyenne sur un échantillon afin de déterminer, pour chaque nouvel élément, s'il se retrouve soit dans la moyenne et n'est donc pas informatif (arrière-plan), soit hors de la moyenne et passe alors, de facto, à l'avant-plan. ».

Dans les parties qui se suivent on va présenter par une simple description l'approche gaussienne simple, les différents algorithmes de résolution de problème de mixture gaussienne et en termine par l'approche mixture gaussienne généralisée laquelle on a utilisée pour réaliser notre système. [NV, 07]

3.1. Exemple introductif :

Un centre d'examen du baccalauréat a relevé les 600 notes obtenues, arrondies à l'entier supérieur, par des élèves de Première en enseignement scientifique.

Les résultats de ce relevé sont consignés dans le tableau suivant :

Notes	[1 ; 2]	[3 ; 4]	[5 ; 6]	[7 ; 8]	[9 ; 10]	[11 ; 12]	[13 ; 14]	[15 ; 16]	[17 ; 18]	[19 ; 20]
Nombre d'élèves	17	38	52	60	92	116	98	65	42	20

Tableau 2.1 : Tableau des résultats

Représentons de cette série statistique sous forme de « diagramme en bâtons » :

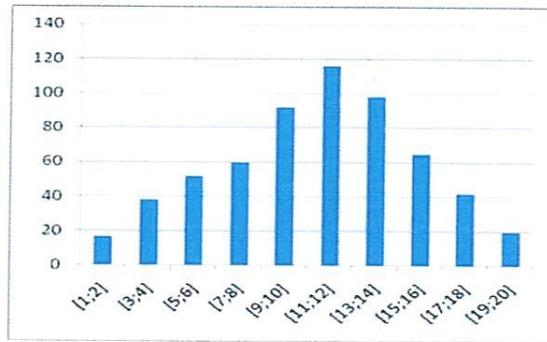


Figure 2.2 : diagramme en bâtons

Cette répartition des notes a une forme particulière appelée « courbe en cloche » et elle est caractéristique d'une « distribution gaussienne ». [S11]

3.2. Fonction de densité de probabilité :

D'une façon plus générale, une variable aléatoire est une certaine fonction, qui dépend du résultat d'une expérience aléatoire.

Une variable aléatoire à valeurs dans \mathbb{R} est dite de type gaussien (ou normale) si elle suit la fonction de densité de probabilité décrite par l'équation $f_x(X)$. On représente généralement une variable aléatoire X , une variable aléatoire de type gaussien de la façon suivante : $X \sim N(\mu, \sigma^2)$, où μ représentent la moyenne et σ^2 la variance.

$\Pr \{x\} = f_x(X)$: définis la fonction de densité $f_x(X)$.

L'intégrale sous la courbe de la fonction de densité de probabilité est toujours égale à 1. Le tracé de la fonction de densité de probabilité gaussienne a la forme d'une cloche symétrique autour de la moyenne. [S11]

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (F1)$$

Aussi appelée "Loi normale" ou "Distribution gaussienne", ou "Loi gaussienne"

L'espérance du gain correspond alors à la moyenne des gains au cours de nombreuses parties.

$$E(X) = \sum_{i=1}^n p_i x_i \quad (F2)$$

La variance s'écrit alors, par analogie avec la formule de la variance avec fréquences:

$$V(X) = \sum_{i=1}^n p_i (x_i - E(X))^2 = \left(\sum_{i=1}^n p_i x_i^2 \right) - (E(X))^2 = E(X^2) - (E(X))^2 \quad (\text{F3})$$

L'écart type reste toujours la racine carrée de la variance : $\sigma_{x=\sqrt{V(X)}}$

3.3. Les algorithmes proposés pour la résolution du problème de mélange de gaussiennes :

Modèle de mélange est une combinaison pondérée des distributions de probabilité, c'est un outil puissant et bien compris pour divers problèmes en intelligence artificielle, vision par ordinateur, et les statistiques. Dans ce post, nous allons examiner les différents algorithmes pour résoudre le problème de mélange de gaussienne.

Les algorithmes K-Means, Soft K-Means et Expectation-Maximization (EM) d'estimation de paramètres de gaussiennes sont parmi les algorithmes de la méthode du seuil optimal qui est la version non supervisée de la méthode du seuil, puisque l'utilisateur n'a pas à choisir un seuil de façon empirique.

L'idée de base derrière cette méthode est que nous pouvons approximer l'histogramme de chacune des classes par une gaussienne. En ayant plusieurs classes, l'histogramme de l'image représente une mixture de gaussiennes, chacune des gaussienne possédant une moyenne, un écart-type et une proportion qui lui est propre. Ayant une mixture de gaussienne, la segmentation peut être réalisée en vérifiant, pour une intensité donnée, quelle classe est la plus probable, ce que nous venons l'implémenter dans ce projet de fin d'études est l'algorithme KM, et pour présenter leurs avantages et une idée générale dans ce qui concerne leur fonctionnement on donne une simple étude comparative entre ces algorithmes :

La majeure différence entre ces algorithmes est la manière dont ils s'y prennent pour calculer les paramètres des gaussiennes.

Les algorithmes K-Means et Soft K-Means calculent la moyenne de chacune des classes de façon itérative jusqu'à convergence et estiment par la suite leur écart-type ainsi que leur proportion. Contrairement à K-Means qui suppose qu'un pixel ne peut être associé qu'à une seule classe, Soft K-Means propose qu'un même pixel puisse appartenir à chaque classe, mais avec une proportion différente. Pour sa part, l'algorithme EM calcule tous les paramètres de chacune des classes de façon itérative jusqu'à convergence.

Ces algorithmes étant de type itératif, leur temps d'exécution dépend principalement de leur vitesse de convergence. Puisque l'algorithme *K-Means* ne calcule que la moyenne de chacune des classes, il est l'algorithme convergeant le plus rapidement et, par le fait même, celui offrant les moins bons résultats. Dans le même ordre d'idée, puisque l'algorithme *EM* doit calculer tous les paramètres de chacune des gaussiennes, il est l'algorithme le plus lent à converger, mais présente des résultats supérieurs. [FB, GC, 07]

3.4. Modèle de mixture de gaussiennes généralisées (MGG) :

Mélanges gaussiens (MGM) sont parmi les méthodes les plus matures sur le plan statistique pour le clustering.

Dans cette approche, chaque pixel est modélisé par une mixture de \mathbf{K} gaussiennes, $2 \leq \mathbf{K} \leq 7$.

Pour une nouvelle image traitée, la mixture de gaussiennes (pour tous les pixels) est mise à jour pour expliquer correctement les couleurs affichées par chaque pixel. Pour faire ceci, à un instant t , on considère que le modèle \mathbf{M}_t généré pour chaque pixel à partir des mesures $\{P_0, P_1, \dots, P_{t-1}\}$ est correct. La vraisemblance pour qu'un pixel appartienne au fond est :

$$P(X_t | \mathbf{M}_t) = \sum_{i=1}^{i=K} W_{i,t} \eta(X_t, \mu_{i,t}, \sigma_{i,t}^2) \quad (\text{F4})$$

$$\eta(X_t / \mu_{i,t}, \sigma_{i,t}) = \frac{1}{\sigma_{i,t} \sqrt{2\pi}} e^{-\frac{(X_t - \mu_{i,t})^2}{2\sigma_{i,t}^2}} \quad (\text{F5})$$

Avec :

P_t : La valeur du pixel dans le temps t .

\mathbf{K} : Le nombre de gaussiennes associées dans le temps t .

$W_{i,t}$: Le poids calculé pour la $i^{\text{ème}}$ gaussienne dans le temps t .

$\mu_{i,t}$: La moyenne de la $i^{\text{ème}}$ gaussienne dans le temps t .

$\sigma_{i,t}$: L'écart type de la $i^{\text{ème}}$ gaussienne dans le temps t .

Pour la mise à jour du modèle, on utilise l'algorithme d'approximation *K-Means*. Chaque valeur d'un nouveau pixel est vérifiée à travers les \mathbf{K} distributions de gaussiennes pour déterminer si la distance entre le pixel et chacune des gaussiennes est inférieure à 2.5 selon la formule suivante :

$$\frac{|X_t - \mu_i|}{\sigma_i} < 2.5 \quad (\text{F6})$$

L'opérateur < est vrai si toutes les composantes du vecteur à gauche sont inférieures pour toutes les distributions. Cette mesure représente le fond si la $i^{\text{ème}}$ gaussienne explique le fond de l'image, alors les poids des K distributions dans le temps t sont mis à jour selon la formule suivante :

$$W_{i,t} = (1 - \alpha)W_{i,t-1} + \alpha M_{i,t} \quad (\text{F7})$$

Avec :

α : Le pas d'apprentissage qui permet de déterminer la vitesse d'adaptation du modèle.

$M_{k,t}$: Egal 1 pour la gaussienne qui correspond à l'arrière-plan et 0 pour les autres.

Après la mise à jour des poids, une étape de normalisation est effectuée pour garantir que la somme des poids est toujours égale à 1. Les autres paramètres sont mis à jour selon les formules suivantes :

$$\mu_{i,t} = (1 - \varphi_i)\mu_{i,t-1} + \varphi_i X_t \quad (\text{F8})$$

$$\sigma_{i,t}^2 = (1 - \varphi_i)\sigma_{i,t-1}^2 + \varphi_i (X_t - \mu_{i,t})^T (X_t - \mu_{i,t}) \quad (\text{F10})$$

$$\varphi_i = \alpha \eta(X_t / \mu_i, \sigma_t) \quad (\text{F11})$$

Les paramètres des autres distributions restent les mêmes. Si aucune distribution ne vérifie la formule précédente, le pixel est associé au 1^{er} plan. La gaussienne ayant le plus petit poids est remplacée par la moyenne, la variance et le poids du pixel courant suivant les formules suivantes :

$$\sigma_i^2 = \sigma_t^2$$

$$W_t = \varphi_t$$

$$\mu_t = X_t$$

Pour décider si X_t est inclus dans l'arrière-plan, les distributions sont ordonnées selon la valeur $\frac{W_{i,t}}{\sigma_{i,t}}$ est les premières β distributions qui vérifions la formule sont choisies.

$$B = \text{argmin} (\sum_{i=1}^K W_{i,t} > T) \quad (\text{F12})$$

Avec T détermine la proportion de données minimum correspondant à l'arrière-plan.

Ce modèle permet d'obtenir de bons résultats pour des scènes d'intérieur où l'arrière-plan est parfaitement statique, mais en environnement extérieur, des phénomènes périodiques tels que l'ondulation d'une surface d'eau ou le balancement d'une branche d'arbre peuvent le

4.6. Déformation de l'objet :

Cet événement modifie la quantité d'information disponible et le patron de l'objet, rendant fréquemment ce dernier caduc. [TR, 08]

4.7. Objet de petite taille :

Plus l'objet est petit, plus la quantité d'information disponible est réduite, plus son suivi pourra être influencé par les perturbations extérieures. De plus, les petits objets sont fréquemment rapides. [TR, 08]

4.8. Objet en mouvement rapide :

Il résulte d'un mouvement rapide le phénomène de « flou de bougé » gommant les contours et diluant les couleurs et textures. Les applications se basant sur ces traits seront donc pénalisées. De plus, lors d'un mouvement rapide, un suivi imprécis perdra plus facilement l'objet. [TR, 08]

4.9. Camouflage :

Lorsque les caractéristiques de l'objet sont proches de celles du fond, celui-ci est difficilement détectable. [SC, 08]

4.10. Ombre :

L'objet projette une ombre dans la scène, même si la luminosité est contrôlée. Les ombres sont parfois segmentées avec l'objet, suivant la méthode utilisée, ce qui rend le résultat de segmentation difficilement exploitable. [SC, 08]

4.11. Contrainte de temps réel :

Cette difficulté n'est pas la moindre. En effet, la plupart des applications nécessiteront un compromis entre leur complexité et leur capacité à gérer les difficultés précédemment décrites. Cette contrainte sera donc un facteur majeur dans le choix de la technique à utiliser. [TR, 08]

5. Conclusion :

Dans ce chapitre nous avons présenté un état de l'art sur les différentes méthodes de détection de mouvement basées sur l'extraction de l'arrière-plan, par la suite nous avons parlé de la méthode de mixture gaussienne où nous avons commencé par un exemple introductif et ensuite nous avons donné les différents algorithmes de résolution de ce problème de mixture gaussienne et enfin nous avons bien détaillé cette approche utilisant l'algorithme de K-means.

4.6. Déformation de l'objet :

Cet événement modifie la quantité d'information disponible et le patron de l'objet, rendant fréquemment ce dernier caduc. [TR, 08]

4.7. Objet de petite taille :

Plus l'objet est petit, plus la quantité d'information disponible est réduite, plus son suivi pourra être influencé par les perturbations extérieures. De plus, les petits objets sont fréquemment rapides. [TR, 08]

4.8. Objet en mouvement rapide :

Il résulte d'un mouvement rapide le phénomène de « flou de bougé » gommant les contours et diluant les couleurs et textures. Les applications se basant sur ces traits seront donc pénalisées. De plus, lors d'un mouvement rapide, un suivi imprécis perdra plus facilement l'objet. [TR, 08]

4.9. Camouflage :

Lorsque les caractéristiques de l'objet sont proches de celles du fond, celui-ci est difficilement détectable. [SC, 08]

4.10. Ombre :

L'objet projette une ombre dans la scène, même si la luminosité est contrôlée. Les ombres sont parfois segmentées avec l'objet, suivant la méthode utilisée, ce qui rend le résultat de segmentation difficilement exploitable. [SC, 08]

4.11. Contrainte de temps réel :

Cette difficulté n'est pas la moindre. En effet, la plupart des applications nécessiteront un compromis entre leur complexité et leur capacité à gérer les difficultés précédemment décrites. Cette contrainte sera donc un facteur majeur dans le choix de la technique à utiliser. [TR, 08]

chapitre 03

Conception du système

Conception du système

1. Introduction :

De nos jours, la vidéosurveillance est omniprésente et on la retrouve dans de nombreux secteurs d'activité (banque, transports, industrie, grande distribution, etc.) ou lieux de vie (villes, immeubles de bureau, équipements collectifs, etc.). Elle nous aide à observer beaucoup de positions différentes en même temps et à reconnaître vite les événements anormaux qui se passent dans la scène.

Cependant, comment un agent de sécurité peut-il analyser en temps réel des dizaines d'écrans de contrôle avec un risque d'erreur minimal? D'autre part, l'observation des écrans dans un temps long est un travail ennuyeux. La réponse de ces deux problèmes tient en deux mots « vidéo intelligente » : un nouveau domaine qui étudie ce qui se passe dans une vidéo. Ce terme s'exprime un axe de recherche assez large, qui est appliqué dans des domaines différents : robotique, médecine, par exemple. En particulier, beaucoup de recherches, de travaux dans ce domaine sont déjà implantés dans la vidéosurveillance : la reconnaissance des objets, des activités dans la vidéo... etc., ce sont des sujets qui sont actuellement intéressés par plusieurs chercheurs.

L'objectif de ce chapitre est de présenter la structure de notre système à réaliser concernant la détection et le suivi des objets dans une séquence vidéo en utilisant une approche basée sur l'extraction de l'arrière-plan qui est la mixture gaussienne.

2. Objectif du système :

Le contexte de notre travail est marqué par la volonté de mettre en place un système informatique qui permettra d'achever la détection, la segmentation et le suivi des objets dans une vidéo de surveillance extraite à partir d'une caméra fixe en temps réel.

Le système est conçu pour atteindre les objectifs suivants :

- Acquisition d'un flux vidéo provenant d'une webcam.
- Traitement des fragments reçus dans l'étape d'acquisition afin de détecter les différents objets en mouvement.
- Affichage de résultat obtenu pour chaque fragment.
- Optimisation du système pour : accélérer le traitement du flux en temps réel, être robuste aux bruits et aux changements globaux et rapides d'intensité lumineuse dans la scène.

3. Architecture du système :

Pour atteindre les objectifs du système cités précédemment, nous avons devisé notre système en trois phases : la première est la phase d'acquisition des fragments du flux vidéo, la deuxième phase de traitement et d'extraction d'arrière-plan, et la dernière pour le suivi des objets en mouvement et affichage des résultats.

L'architecture proposée est illustrée dans la figure suivante, elle rassemble les différents modules qui composent le système :

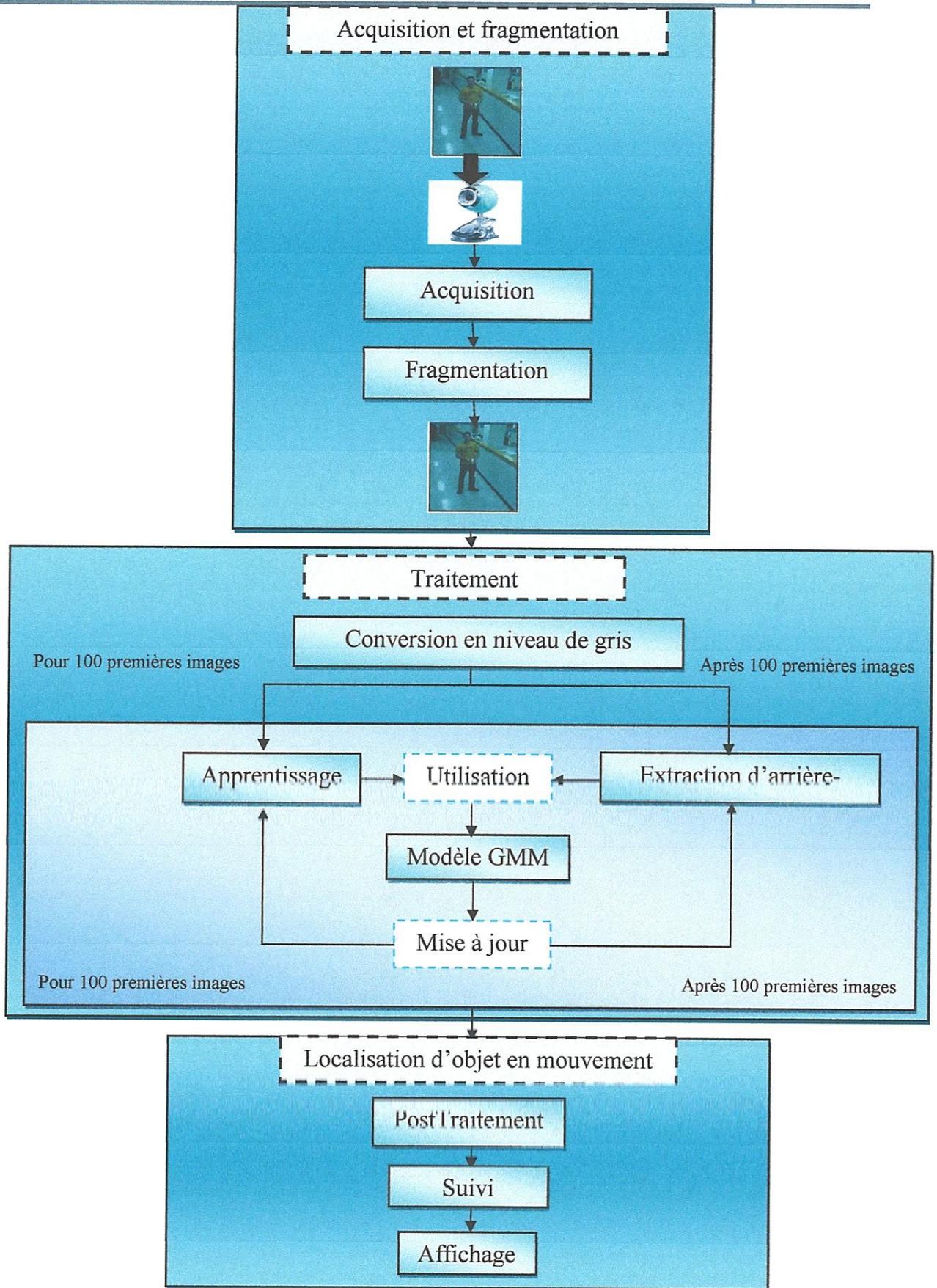


Figure 3.1 : Architecture du système

4. Réalisation de l'architecture :

Les phases de l'architecture de notre système vue précédemment seront détaillées dans ce qui suit :

4.1. Phase d'acquisition et de fragmentation :

Cette première étape repose dans un premier temps sur la visualisation en temps réel du flux vidéo qui arrive sur notre ordinateur, la deuxième sur la capture des séquences d'images de la source pour pouvoir les traiter par la suite.

Ce module d'acquisition d'images représente alors l'entrée de notre système, cette tâche doit être effectuée via un dispositif de capture vidéo, ils existent pour cela plusieurs solutions pour le choix matériel dans ce module, mais dans notre application nous avons exploité une simple webcam qui est caractérisée par une simple installation sur les différents systèmes d'exploitation et qui se connecte à un PC via un port USB.

Nous considérons un flux de vidéo : comme une suite d'images 2D. La résolution de la vidéo, exprimée en nombre de pixels, définit la dimension de ces images. La durée du temps entre deux images (Δt) est très petite parce que nous savons que la vitesse de film, en général, est de 24 à 60 images par seconde. Ça dépend de taux de rafraîchissement. Cette cadence influence sur le suivi, la reconnaissance et la détection du mouvement.

Cette phase est composée de 2 parties qui sont :

4.1.1. Acquisition :

C'est la lecture du flux à partir d'un périphérique de capture vidéo pour pouvoir le récupérer et l'afficher dans une Frame suivant ces étapes :

- Sélection du périphérique vidéo à utiliser pour la capture.
- Création d'un Media Locator qui est l'adresse vers la source de données.
- Création d'un Player (prends le flux vidéo en entrée pour le rendre via l'écran. C'est un objet à états, ce qui signifie qu'il doit se préparer avant de pouvoir être utilisé) à partir du Manager et du Media Locator créés précédemment.
- Appel de la méthode realize () du Player (afin d'opérer les initialisations).
- Écoute sur le Player afin de savoir lorsque les initialisations sont terminées pour lancer la lecture (via la méthode start () du Player).

4.1.2. Fragmentation :

La fragmentation est un processus qui se répète après la fin de traitement de l'image courante, et pour cela on a utilisé un thread.

Pour fragmenter le flux vidéo et afin d'obtenir une image on a utilisé ces instructions :

- **Récupérer un contrôle sur le flux :** `FrameGrabbingControlControle_De_Capture = (FrameGrabbingControl) Afficheur.getControl ("javax.media.control.FrameGrabbingControl");`
- **Mettre le résultat dans un buffer :** `Tampon = Controle_De_Capture.grabFrame ();`
- **Convertir le buffer à un buffer d'image :** `Tampon_A_Image = new BufferToImage ((VideoFormat) Tampon.getFormat ());`
- **Créer l'image :** `Picture2 = Tampon_A_Image.createImage (Tampon);`
L'image résultat de cette étape c'est sur laquelle qu'on va appliquer le traitement.

4.2. Phase de traitement :

Cette phase englobe le corps de notre système qui est l'extraction avant-plan/arrière-plan en utilisant l'approche de mixture de gaussienne où nous avons implémenté notre propre algorithme optimisé pour qu'il donne un meilleur résultat concernant le traitement en temps réel.

Avant d'appliquer cet algorithme nous avons converti l'image fragmentée en niveau de gris parce qu'il a donné un résultat plus efficace que le mode RVB(en terme de temps de calcul).

4.2.1. Conversion d'image en niveau de gris :

Pour les images couleur, un pixel dispose généralement des trois composantes RGB (Rouge, Vert, Bleu). Un pixel gris a ses trois valeurs RGB identiques. Une méthode simple pour convertir une image couleur en niveau de gris est de calculer la moyenne des trois composantes RGB et d'utiliser cette valeur moyenne pour chacune des composantes.

Nous avons appliqué ce mode de conversion afin de réduire la taille des fichiers image rvb, gagner du temps d'exécution, et restituer mieux les nuances colorimétriques.

Alors, l'algorithme utilisé est le suivant :

```
Pour chaque pixel :  
  
    {  
  
        Rouge= récupérer la valeur de la couleur rouge du pixel ;  
        Vert= récupérer la valeur de la couleur verte du pixel;  
        Bleu= récupérer la valeur de la couleur bleu du pixel;  
        Niveau-gris= ((rouge+vert+bleu)/3);  
        Nouvelle-Valeur-Pixel = (0xff000000 |Rouge<< 16 |Vert<< 8 | Bleu);  
  
    }
```

4.2.2. Apprentissage :

Après la mise en niveau de gris des images en provenance de la caméra, elles seront analysées suivant la méthode des mixtures de gaussiennes. Ce traitement consiste à détecter la présence d'objets ou de mouvements dans le défilement des images.

Les valeurs prises par les pixels du fond sont modélisées par plusieurs gaussiennes. Cela permet de prendre en compte des valeurs différentes pour chacun d'entre eux. Cette modélisation ne peut être présentée d'une manière correcte à l'aide d'un modèle utilisant une seule gaussienne. Il faut donc choisir une représentation plus complète. Dans ce cas, une représentation sous la forme de mixture de gaussiennes est préférée, on peut prendre entre 2 et 7. L'avantage de cette étape est de garder l'arrière-plan ou sauver les caractéristiques du fond. Alors on ne laisse passer pour quelques secondes que des images ne comportent que le fond (environ de 100 images).

4.2.3. Extraction d'arrière-plan :

Après la mémorisation des caractéristiques de l'arrière-plan dans la phase d'apprentissage, on laisse passer un objet en mouvement afin de le séparer de l'arrière-plan.

Pour cela, on calcul la distance entre chaque pixel de l'image et chacune des gaussiennes qu'ils l'associent, ensuite on passe à la mise à jour des paramètres selon les résultats trouvés précédemment.

Ce traitement est fait selon le modèle de GMM détaillé par la suite.

4.2.4. Modèle GMM :

C'est le modèle qu'on a choisi pour réaliser notre application, ses détails sont décrits dans l'algorithme suivant :

Pour chaque pixel :

```

{
    Pour i allant de 1 à K faire
        {
            Si (  $\frac{|X_t - \mu_t|}{\sigma_t} < 2.5$  ) alors
                {
                    Existe=1 ;
                    Tab[i]=1 ;
                }
            Sinon tab[i]=0 ;
        }
    Si (Existe == 1)
        {
            Pour i allant de 1 à k
                {
                    Si (tab[i] == 1)
                        {
                             $W_{i,t} = (1 - \alpha)W_{i,t-1} + \alpha$  ;
                             $\varphi_i = \alpha \eta(X_t / \mu_i \sigma_t)$  ;
                             $\mu_{i,t} = (1 - \varphi_i)\mu_{i,t-1} + \varphi_i X_t$  ;
                             $\sigma_{i,t}^2 = (1 - \varphi_i) \sigma_{i,t-1}^2 + \varphi_i (X_t - \mu_{i,t})^T (X_t - \mu_{i,t})$  ;
                        }
                    Sinon  $W_{i,t} = (1 - \alpha)W_{i,t-1}$  ;
                }
            }
        }
    Sinon  $\{\sigma_t^2 = \sigma_t^2$  ;
     $W_t = \varphi_t$  ;
     $\mu_t = X_t$  ;
    }
    Normalisation ;

```

```

Test=vrai ;
nb : nombre de voisinage du pixel portant la couleur blanche ;
Tant que (test) faire
    {
        Test=faux ;
        Pour chaque pixel
            {
                Si ((masque1)or(masque2)or(masque3)or(masque4)or(masque5) or(masque6))
or((nb=3)and ((masque7)or(masque8)))
or((nb=4)and((masque7)or(masque8)or(masque9)))
or(nb>=5))
                    {
                        Pixel= couleur blanche ;
                        Test=vrai ;
                    }
            }
    }
}

```

4.3.1.2. Elimination des points isolés :

L'apparition des points isolés qui sont des points rebelles diminue la qualité de la détection donc il est nécessaire de les supprimer pour cela l'algorithme qui se suit manipule cette partie :

```

Test=vrai ;
nb : nombre de voisinage du pixel portant la couleur noire ;
Tant que (test) faire
    {
        Test=faux ;
        Pour chaque pixel
            {
                Si (nb=8)
                    {
                        Pixel= couleur blanche ;
                        Test=vrai ;
                    }
            }
    }
}

```

1. Introduction :

En informatique, l'implémentation désigne la création d'un produit fini à partir d'un document de conception ou de spécification. Elle doit répondre à des contraintes (les performances, la fiabilité, etc.) qui lui sont propres, et ne sont généralement pas toutes explicites dans les documents précités.

Dans ce chapitre, nous présentons les principes de l'implémentation de notre système, en commençant par la définition des différents outils de programmation utilisés pour atteindre les objectifs du système, puis nous montrons les résultats obtenus suite aux tests réalisés sur des vidéos dans des différents environnements et enfin en termine par un tableau comparatif entre ces résultats.

2. Présentation des outils de développement :

Pour réaliser ce système, on a utilisé les logiciels suivants :

2.1. JDK «Java Development Kit »:

C'est un logiciel édité par Sun pour le développement d'application en Java, où le code Java est compilé pour être transformé en bytecode afin que la machine virtuelle Java (JVM) puisse l'interpréter. [S12]

2.2. Netbeans :

C'est un environnement de développement intégré (EDI) qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plate forme. En plus de Java, il permet également de supporter différents autres langages comme Python, C, C++, JavaScript, XML, Ruby, PHP etHTML.

Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 etSPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java. [WM, MSA, 11]

2.3. JMF « Java Media Framework » :

JMF est une API (application programming interface) Java permettant de manipuler aisément toutes sortes de contenus multimédias avec Java tels que du son ou de la vidéo. Elle offre les outils nécessaires pour faire de l'acquisition, du traitement et du transport de médias basés sur le temps. L'avantage est de pouvoir concevoir des applications utilisant des éléments multimédias (Webcam, micro, vidéos...) et pouvant s'exécuter sur différentes plates-formes logicielles (principalement Windows et Linux). La version actuelle de JMF est la 2.1. Cette API est une initiative de SUN qui souhaite apporter une solution « time-based media processing » (traitement de média basé sur un timeline) à Java. Les médias basés sur le temps sont des données qui changent par rapport au temps. Nous les retrouvons bien entendu dans les vidéos, l'audio, les séquences MIDI et autres animations. Cependant, JMF ne fait pas de distinction entre les différentes webcams installées sur la machine, Il les prend toutes pour : « vfw:Microsoft WDM Image Capture (Win32):0 ». Du coup, on ne peut en utiliser qu'une seule à la fois, sans même pouvoir la spécifier.

Grâce au jmf on peut retourner l'image instantanée du flux sur la webcam, d'une autre façon la convertir à un instant donné en un objet Image. [FM]

3. Description du système :

Notre système est composé de trois parties : la première donne le résultat de l'algorithme de mixture gaussienne, la deuxième comporte un post traitement pour mieux représenté l'objet en mouvement, et la dernière partie concerne l'affichage de la vidéo lui-même avec un encadrement des objets en mouvement.

Dans cette partie nous allons présenter les résultats obtenus pour les trois parties dans les environnements choisis et qui sont :

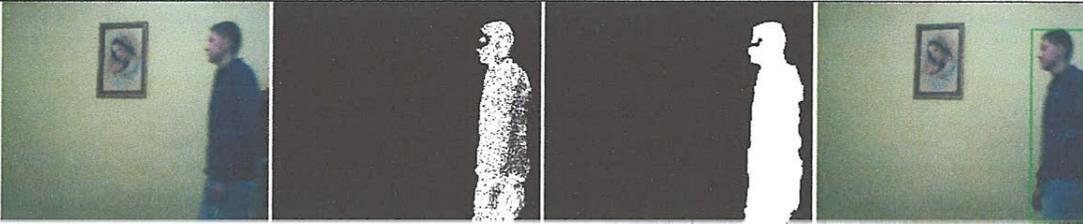
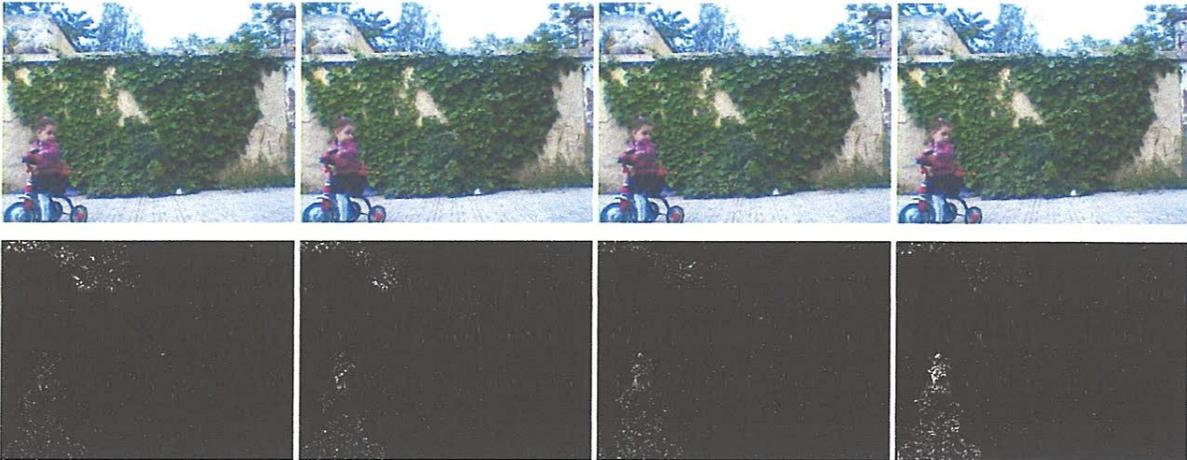
- **Un environnement simple** : caractérisé par un arrière-plan simple, un seul objet en mouvements avec un éclairage stable.
- **Un environnement complexe** : contiens beaucoup de détail concernant l'arrière-plan et plusieurs objets qui bougent.
- **Un environnement avec éclairage variant** : un milieu ouvert ou la lumière changeante(jardin, route,...).

- Un environnement avec faible éclairage : un espace fermé (chambre, bureau, boutique,...)

En premier temps nous avons testé notre application en changeant les paramètres (T : la proportion de données minimum correspondant à l'arrière-plan, α : taux d'apprentissage et le nombre de gaussiennes) pour arriver à une initialisation optimale en suite cette dernière est utilisée pour effectuer une suite de tests.

Pour la première étape les exemples de détection sont présentés dans le tableau qui se suit, pour la deuxième étape les résultats sont organisés par le type de traitement.

3.1. résultats du test avec changement des paramètres:

Paramètres	Résultat
gaussiennes=7, T=0.30, $\alpha = 0.01$	C'est la meilleure estimation des paramètres trouvés après une suite de tests réalisée, où les objets qui bougent sont bien détectés et représentés.
	
gaussiennes=7, T=0.30, $\alpha = 0.1$	la détection d'objet n'est pas assez bien ainsi que l'intégration à l'arrière-plan est très lente.
	

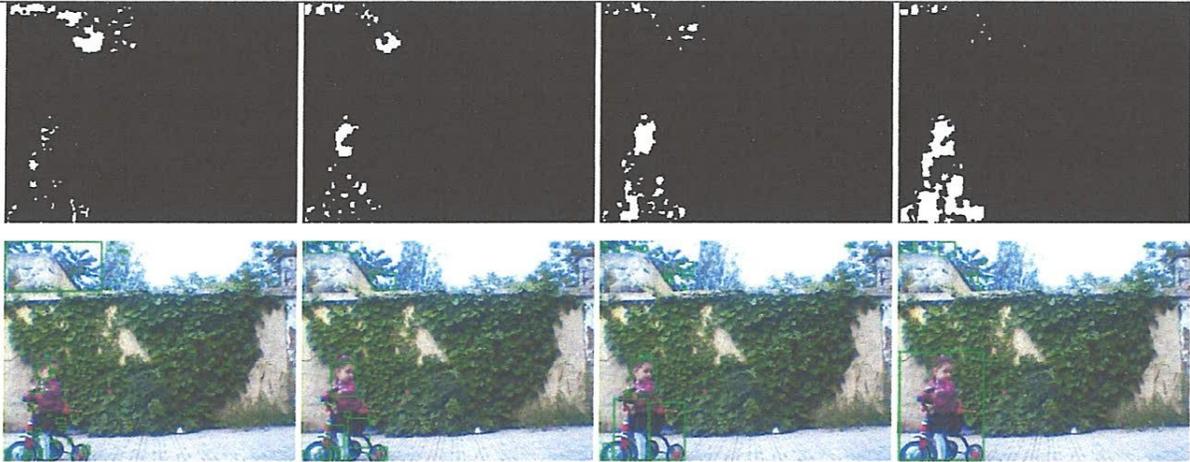
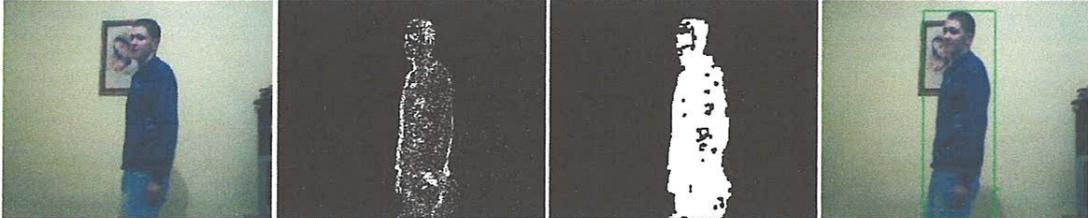
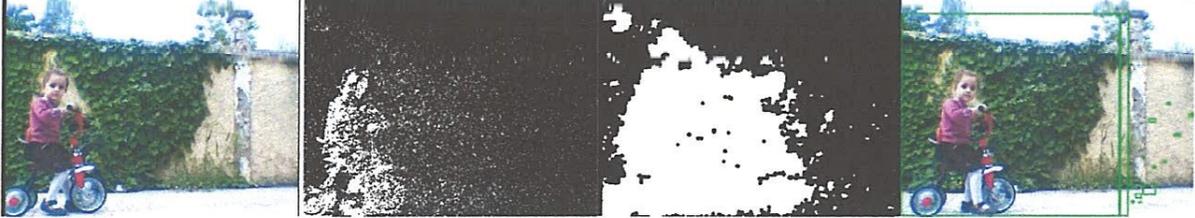
			
<p>gaussiennes=7, $\alpha = 0.01$</p>	<p>T=0.90,</p>	<p>Les objets en mouvements ne sont pas bien détectés et les pixels de l'arrière-plan sont les plus répondus ainsi il ya beaucoup de point isolé.</p>	
			
<p>gaussiennes=3, T=0.30, $\alpha =0.01$</p>	<p>Une mauvaise extraction d'arrière-plans/avant-plan par rapport aux résultats obtenus précédemment.</p>		
			

Tableau 4.1 : Résultats du test avec changement des paramètres

Après ces tests nous avons arriver à une estimation des paramètres satisfaisante concernant les résultats obtenus, laquelle nous allons utiliser dans les tests suivants :

3.2. Les résultats de la mixture gaussienne :

L'algorithme que nous avons appliqué nous a permis d'extraire et segmenté l'objet en mouvement par rapport à l'arrière-plan :

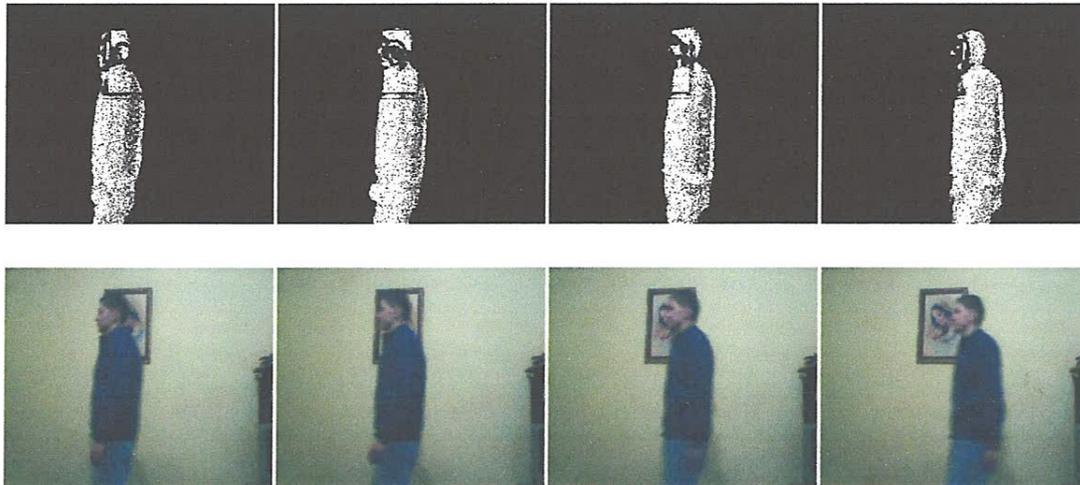


Figure 4.1 : Détection et segmentation d'objet dans un environnement simple

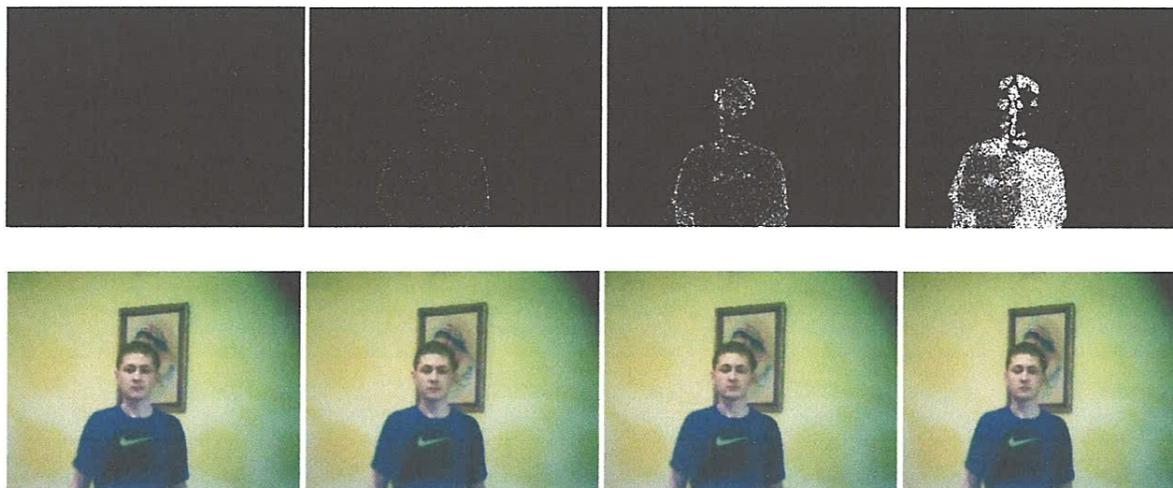


Figure 4.2 : détection et segmentation d'un objet qui s'intègre à l'arrière-plan dans un environnement simple

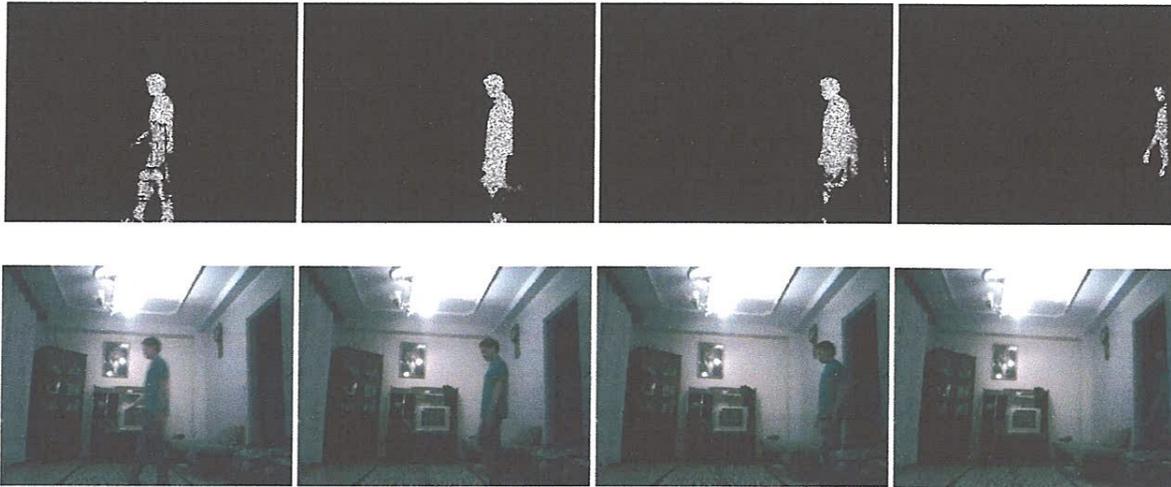


Figure 4.3 : Détection et segmentation d'objet dans un environnement avec faible luminance

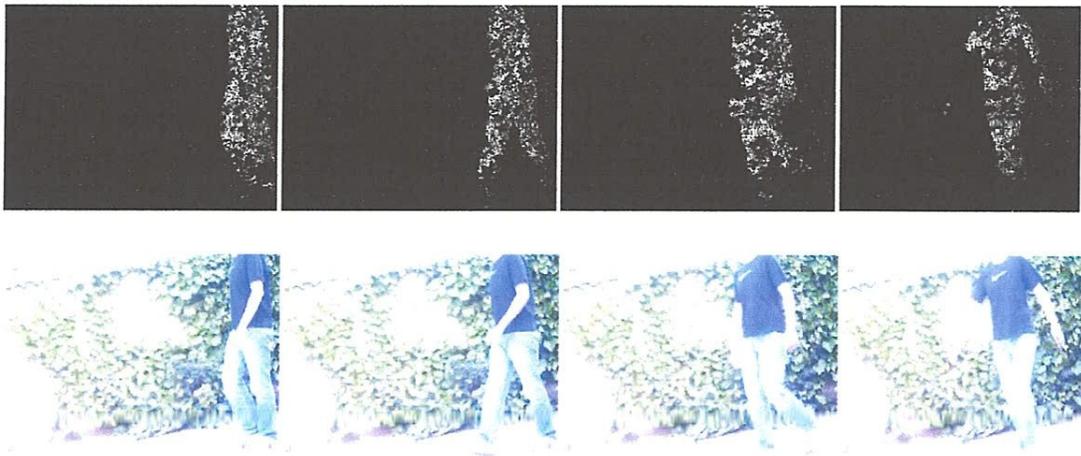


Figure 4.4 : Détection et segmentation d'objet dans un environnement avec lumière variante

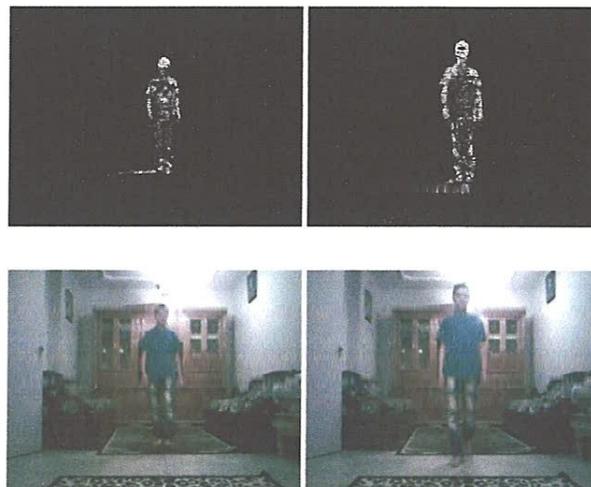


Figure 4.5 : Problème d'ombre pour la détection et segmentation d'objet dans un environnement avec forte lumière



Figure 4.6 : détection et segmentation de plusieurs objets en mouvement

3.3. Les résultats de post traitement :

Cette phase consiste à l'élimination des points isolés et remplissage des petits vides pour améliorer la qualité de l'image résultante où l'objet détecté sera plus clair que celui de la phase précédente.

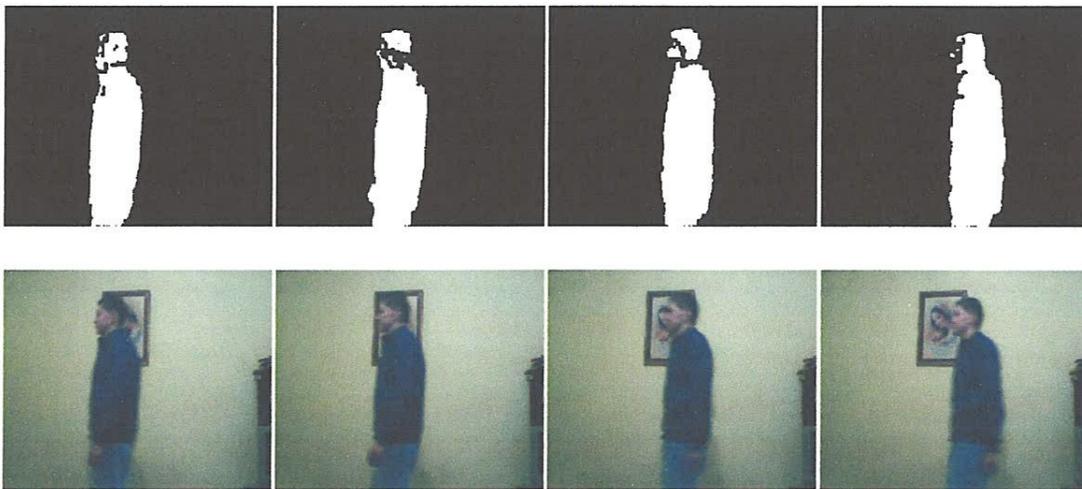


Figure 4.7 : Délation d'objet dans un environnement simple

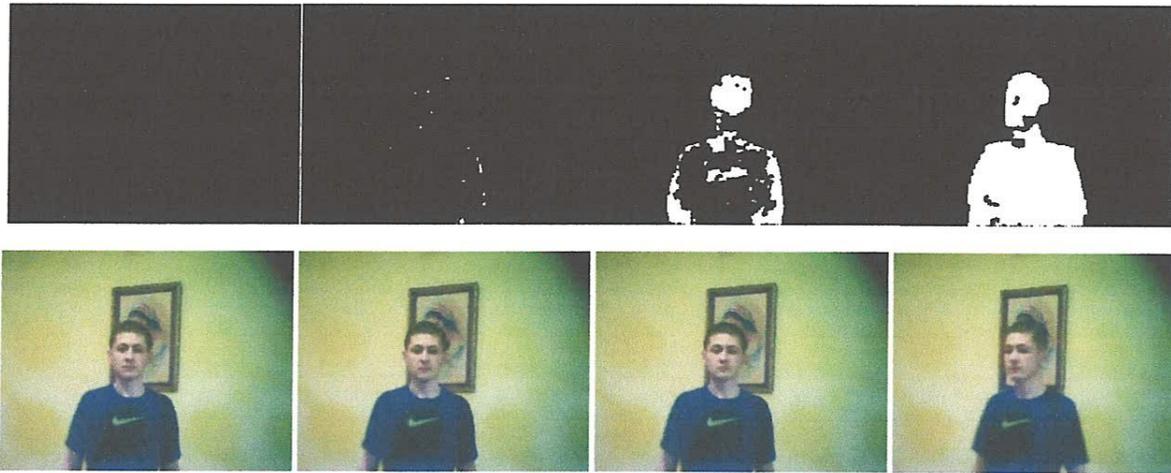


Figure 4.8 : délation d'objet qui s'intègre à l'arrière-plan dans un environnement simple

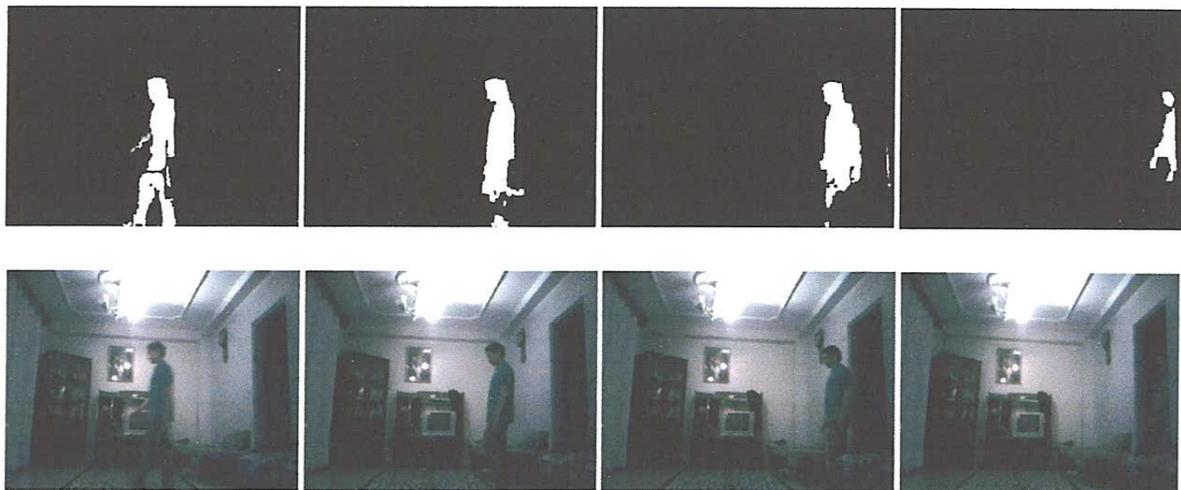


Figure 4.9 : Dilatation d'objet dans un environnement avec faible luminance

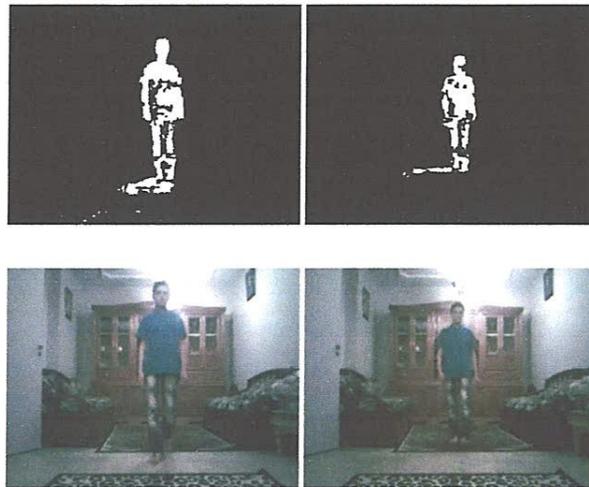


Figure 4.10 : Problème d'ombre pour la dilatation d'objet dans un environnement avec forte lumière

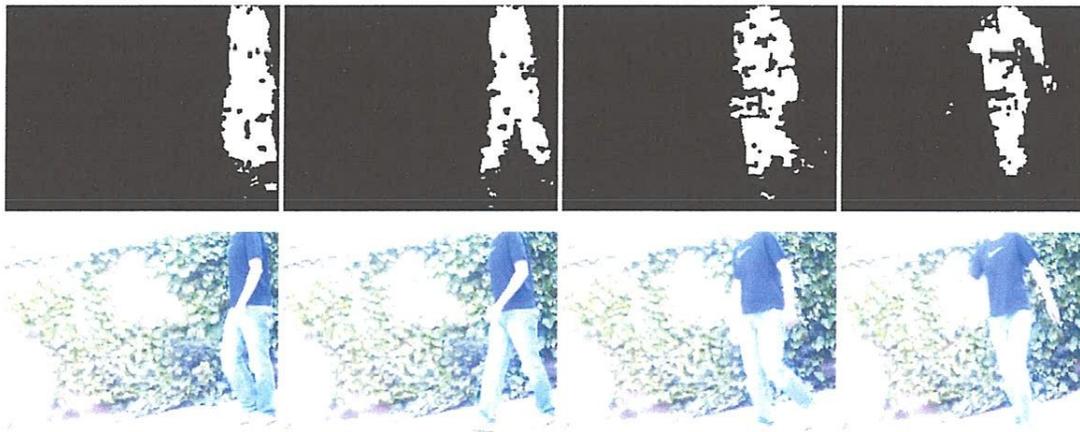


Figure 4.11 : Délétion d'objet dans un environnement avec lumière variante



Figure 4.12 : Délétion de plusieurs objets en mouvement

3.4. Les résultats de suivi avec encadrement :

Le résultat de cette phase est la vidéo elle-même en temps réel avec des boîtes qui englobe les objets en mouvement.



Figure 4.13 : Encadrement d'objet dans un environnement simple

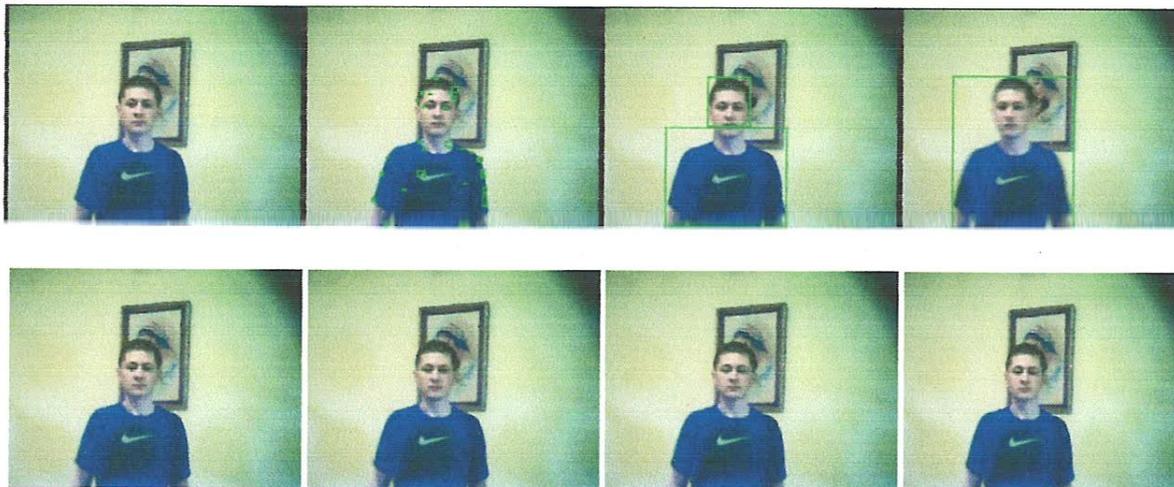


Figure 4.14 : Encadrement d'objet qui s'intègre à l'arrière plan dans un environnement simple



Figure 4.15 : Encadrement d'objet dans un environnement avec faible luminance



Figure 4.16 : Problème d'ombre pour l'encadrement d'objet dans un environnement avec forte lumière



Figure 4.17 : Encadrement d'objet dans un environnement avec lumière variable



Figure 4.18 : Encadrement de plusieurs objets en mouvement

Les résultats obtenus sont différents d'un milieu à un autre, ils sont satisfaisants pour des milieux simples et d'un éclairage stable (Figure 4.1), quelque soit pour un seul objet ou plusieurs (Figure 4.6).

Contrairement pour des milieux ouverts avec lumière variable, cette dernière présente un grand problème concernant notre méthode parce que les conditions d'éclairage variables déclenchent de fausses détections (Figure 4.17), les résultats sont très mauvais par rapport aux autres environnements.

Les tests qu'on a réalisés montrent les points forts et les points faibles de notre application où ces derniers seront notre futur travail pour améliorer la méthode et obtenir des bons résultats.

4. Conclusion :

Dans ce dernier chapitre nous avons en premier temps décrit les différents outils utilisés pour l'implémentation de notre système, ensuite nous avons présenté les résultats de testes réalisés dans plusieurs et divers environnements suivi d'un tableau de comparaison des résultats obtenus.

Conclusion générale

Conclusion générale

Conclusion générale

Conclusion générale

Les idées soutenues dans ce mémoire ont une finalité pratique et un objectif précis, qui sont la conception et l'implémentation d'un système de localisation d'un (ou plusieurs) objet en mouvement en temps réel en utilisant une caméra fixe, l'approche utilisée pour atteindre les objectifs de notre système est basée sur la modélisation de l'arrière-plan qui s'appelle la mixture de gaussienne.

Le principe de base de notre système repose sur la visualisation en temps réel du flux vidéo qui arrive sur notre ordinateur et la capture des séquences d'images de la source pour pouvoir les traiter par la suite afin d'extraire et d'encadrer les objets qui bougent pour les présenter à un administrateur ou le responsable de la surveillance.

En effet, le travail que nous avons effectué nous a permis de réaliser par des algorithmes les objectifs de notre recherche. Les résultats obtenus sont acceptables et encourageants, mais nous suggérons de continuer la recherche dans ce sens pour améliorer encore plus les résultats.

Comme perspective, le système pourrait être amélioré, tout d'abord améliorer sa vitesse d'exécution en le simplifiant, améliorer son résultat en améliorant les techniques, ou encore en facilitant son utilisation, car un des problèmes majeurs est le nombre important de variables qui peut jouer un rôle dans la reconnaissance de l'objet.

Pour élargir les types de vidéo, travailler avec une caméra qui n'est pas fixe pourrait être intéressant. Cet objectif ne faisait pas partie des objectifs principaux, mais il permettrait de travailler avec d'autres vidéos. Le problème est que certains algorithmes étudiés seraient inutilisables dans ce cas, en particulier celles qui utilisent un background

Pour cela, il serait envisageable d'étudier de nouveaux algorithmes qui pourraient améliorer les résultats d'un type de vidéo ou d'objet.

Alors, le suivi d'objets visuel reste un sujet de recherche ouvert de par le manque de système de suivi holistique satisfaisant, prenant en compte les distorsions intrinsèques et

extrinsèques, variations d'éclairage, occlusions, bruits et erreurs dans la correspondance et la classification d'objets ne sont qu'une partie des problèmes actuellement rencontrés en suivi d'objets.

Bibliographie

Bibliographie

Bibliographie

Bibliographie

- ❖ [AB, 07] : Aurélie Bugeau « Traitement du signal et télécommunications », Thèse de l'université de Rennes 1, 2007.
- ❖ [BA, 10] : BOUCETTA ALDJIA « Etude de l'effet des Transformées de Décorrélation en Compression des Images Couleurs RGB », Mémoire de Magister : Ingénierie des systèmes informatique, Université de Batna, 2010.
- ❖ [CT, 07] : Cao Tien Dung « Vidéo Surveillance », Rapport de TIPE « Travail d'Intérêt Personnel Encadré », Institut de la Francophonie pour l'Informatique, 2007.
- ❖ [CP, 08] : CHAU Duc Phu « REPARATION DES TRAJECTOIRES DE PERSONNES SUIVIES BASEE SUR LE CLUSTERING DE POINTS », Mémoire de Master, INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE , 2008.
- ❖ [FB, GC, 07] : Francis Guillaume, Brissette Comeau « La méthode du seuil optimal »
- ❖ [FC, JD, ZM] : Francis COTTET, Joëlle DELACROIX, Claude KAISER, Zoubir MAMMERI « Ordonnancement temps réel, Ordonnancement centralisé »
- ❖ [FM] : FERJANI Mohammed « Vidéo Surveillance à Distance »
- ❖ [FS] : Frank Singhoff « Introduction aux systèmes temps réel »
- ❖ [JCF, 10] : Jean-Charles Fouché « Comprendre la vidéo numérique », 2010.
- ❖ [HQ] : Hongyang QU « Introduction aux systèmes temps réel »
- ❖ [MB, 05] : Dr M. Benmohammed « Implémentation d'un Environnement Parallèle pour la Compression d'Images à l'aide des Fractales », Thèse doctorat, Université Mentouri Constantine, 2005.
- ❖ [MI, 07] : Mohamed ILAMMAMI « Modèle de peau et application à la classification d'images et au filtrage des sites Web », Thèse de doctorat, Ecole Centrale de Lyon, 2007
- ❖ [MM, 09] : Mme Maratray « Méthodes de suivi d'un objet en mouvement sur une vidéo » Stage d'application : L'université Jaume I12071 Castellón de la plaña Espagne, 2009.

- ❖ **[NV, 07]** : NICOLAS VERBEKE « Suivi d'objets en mouvement dans une séquence vidéo», Thèse : Université PARIS DESCARTES, Centre universitaire des Saints-Peres,UFR DE MATHÉMATIQUES ET INFORMATIQUE ,2007.
- ❖ **[NVI, 07]** : Nicole Vincent « Détection de Mouvements Cohérents dans une Séquence Vidéo», Article : Université René Descartes France, 2007.
- ❖ **[PD, 2002]** : Pierre-Yves Duval « Introduction à l'informatique temps réel», Ecole d'informatique temps réel - La Londe les Maures, Octobre 2002.
- ❖ **[PF, PK, 03]** : Pierre Ficheux, Patrice Kadionik « Temps réel sous LINUX», 2003.
- ❖ **[PP]** : Pierre.Paradinas « Systèmes Embarqués (SEE) : Cours A7 : Temps Réel »
- ❖ **[RR, 07]** : Rafael C. Gonzalez, Richard E. Woods « Digital Image Processing», Livre, 2007.
- ❖ **[RR]** : Romain RAFFIN « Traitement d'images »
- ❖ **[SB]** : Samia Bouzafrane « Introduction aux systèmes temps réel »
- ❖ **[SC, 08]** : Simon Conseil « SUIVI TRIDIMENSIONNEL DE LA MAIN ET RECONNAISSANCE DE GESTES POUR LES INTERFACES HOMME -MACHINE», Thèse de Doctorat, Université Paul Cézanne, 2008.
- ❖ **[SD, 10]** : Séverine Dubuisson « Prise en compte des variations d'illumination dans le filtrage particulière pour le suivi dans les séquences vidéo », 2010.
- ❖ **[TGG]** : N. Tronson, Y. Goyat, D. Gruyer « Comparaison de méthodes d'extraction fond/forme pour des scènes de circulation routière »
- ❖ **[TR, 08]** : Trichet Rémi « SUIVI D'OBJET POUR LA TELEVISION INTERACTIVE », Séminaire : Ecole d'ingénierie et centre de recherche en télécommunications, 2008.
- ❖ **[WM, MSA, 11]** : Wahid Mejri & Mohamed Slim Arafa « Conception et réalisation d'une application de gestion des comptes mail et internet», Mémoire de Licence Appliquées en Sciences et Techniques de l'Information et de Communications, Université VIRTUELLE DE TUNIS, 2011.

Webographie

Webographie

Webographie

Webographie

- ❖ [S1] : <http://www.docstoc.com/docs/10528561/Quest--ce-quune-image-num%C3%A9rique> « 2011»
- ❖ [S2]:http://www.memoireonline.com/12/09/3040/m_La-liaison-automatique-des-plusieurs-images-percues-sur-un-scanner3.html«2008»
- ❖ [S3] : tp://urfist.enc.sorbonne.fr/anciensite/image_numerique/image.htmh « Février 2005»
- ❖ [S4] : <http://www.library.cornell.edu/preservation/tutorial-french/intro/intro-06.html> « 2003»
- ❖ [S5] : <http://www.portices.fr/formation/Res/ImageCreer/Format.html> « 5 janvier 2006»
- ❖ [S6]:http://documentation.flypix.info/VIDEO%20NUMERIQUE/FLYPIX_Document_VIDEO_NUMERIQUE.html «22 Novembre 2006»
- ❖ [S7]: http://www.multilingualarchive.com/ma/enwiki/fr/Video#Number_of_frames_per_second « 2011 »
- ❖ [S8] : <http://wikipedia.qwika.com/en2fr/Video>
- ❖ [S9] : http://www.idvideo.com/solutions/visioconference/domaines_d_application.html «2008»
- ❖ [S10] : <http://dictionnaire.phpmyvisites.net/definition-Temps-reel--13226.htm> « mars 2012 »
- ❖ [S11] : <http://www.maxicours.com/soutien-scolaire/mathematiques/1re-stg/144895.html> «2012»
- ❖ [S12] : <http://www.inseadima.com/t369-j2ee-environnement-ubuntu> «2006»
- ❖ [S13] : <http://www.pobot.org/+-temps-reel-+.html>«2012»