

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 : Systèmes Informatiques

Thème :

**Utilisation des techniques du Data Mining
pour la détection des objets en mouvements**

Encadré Par :

Dr. FAROU Brahim

Présenté par :

NEBILI Charefeddine

Juillet 2019



REMERCIEMENT

Je tiens à remercier dans un premier temps, Dieu(Allah) qui nous a donné la volonté, la patience et la force pour réaliser ce travail.

Nos vifs remerciements vont au Docteur FAROU BRAHIM pour les précieux conseils et l'aide durant toute la période du travail.

Nous tenons à présenter nos sincères remerciements :

Aux membres du jury en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

A tous les enseignants du département informatique, qui ont assisté à nos initiations en informatique, pour leurs précieux conseils.

Merci finalement à tous ceux qui de près ou de loin nous ont soutenu le long de nos parcours scolaires.

Nebili charefeddine



DEDICACE

Mon père

L'épaule solide, l'œil attentif, compréhensif et la personne la plus digne de mon estime et de mon respect.

Aucune dédicace ne saurait exprimer mes sentiments, que Allah te préserve et te procure santé et longue vie.

Ce travail est le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation.

Ma mère

Affable, honorable, aimable : tu représentes pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi.

Ta prière et ta bénédiction m'ont été d'un grand secours pour mener à bien mes études.

Aucune dédicace ne saurait être assez éloquente pour exprimer ce que tu mérites pour tous les sacrifices que tu n'as cessé de me donner depuis ma naissance, durant mon enfance et même à l'âge adulte.

Je te dédie ce travail en témoignage de mon profond amour. Puisse Allah, le tout puissant, te préserve et t'accorde santé, longue vie et bonheur.

Je remercie mes sœurs meriem, abir, wafa

Vous m'avez toujours soutenue durant toutes mes études, je vous souhaite beaucoup de réussite et de bonheur

Mes Chers amis

Monder, Didine, Zaki, Zinou, Sissouf, Hocine, Anis, Issam

En témoignage de l'amitié qui nous uni et des souvenirs de tous les moments que nous avons passé ensemble, je vous porte toujours dans mon cœur, je vous souhaite tout le bonheur, la joie et le succès dans votre vie.

Nebili charefdine

ملخص

تنوع البيئة الطبيعية مثل تغيرات الإضاءة، ديناميكية الخلفية، تأثير التمويه، والظل، .. إلخ. يطرح مشكلة كبيرة لطرق اكتشاف الأجسام المتحركة التي لا يمكنها تفسير تعدد النماذج في أنظمة المراقبة بالفيديو.

في هذه الأطروحة، سنقدم طريقة جديدة تسمح بالكشف الأفضل عن الأجسام المتحركة عن طريق الجمع بين قوة أنظمة المناعة الاصطناعية (AIRS) مع الاختلافات المحلية وقوة الخلائط الجوسية (GMM) لنمذجة التغييرات على مستوى البكسل.

دور AIRS هو توليد عدة نماذج GMM لكل بكسل. هذه الأخيرة تخضع لآلية تصفية من خلال عملية "المنافسة في الموارد وتطوير خلايا الذاكرة مرشحة". يتم دمج النماذج المتبقية في نموذج مرشح واحد ويتم تقديمها كنموذج جديد للخلفية بواسطة عملية "تقديم خلية الذاكرة".

لقد أثبت نتائج التقييم لقاعدة البيانات Wallflower فعالية أداء نظامنا مقارنة بالطرق الأخرى الموضحة في المناهج المقترحة.

الكلمات المفتاحية:

المراقبة بالفيديو، طرح الخلفية، GMM، AIRS، تجزئة المقدمة، تصنيف البكسل.

Résumé

La diversité de l'environnement naturel tel que les changements d'éclairage, la dynamique du fond, l'effet de camouflage, l'ombre, etc. pose un grand problème pour les méthodes de détection des objets en mouvement qui n'arrivent pas à bien gérer la multi-modalité des scènes dans les systèmes de vidéos de surveillance.

Dans ce mémoire nous allons présenter une nouvelle méthode qui permet une meilleure détection des objets en mouvement en combinant la robustesse des systèmes immunitaires artificiels (AIRS) par rapport aux variations locales et le pouvoir des mixtures de gaussiennes (GMM) à modéliser les changements au niveau pixel.

Le rôle de l'AIRS est de générer plusieurs modèles GMM pour chaque pixel. Ces derniers sont soumis à un mécanisme de filtrage par le processus de « *Compétition de ressources et développement de cellule mémoire candidate* ». Les modèles restants sont fusionnés dans un seul modèle candidat et introduits comme un nouveau modèle de l'arrière-plan par le processus « *Introduction de cellule mémoire* ».

Une évaluation sur la base de données Wollflower a prouvé la performance de notre système par rapport aux autres méthodes de l'état de l'art.

Mots-Clés : vidéosurveillance, soustraction de l'arrière-plan, GMM, AIRS, Segmentation de l'avant-plan, classification des pixels.

Abstract

The diversity of the natural environment such as changes in lighting, dynamics, etc. the background, the camouflage effect, shade, etc. is a major problem for the methods of detection of moving objects that do not manage to manage the multi-modality of scenes well in surveillance video systems.

In this paper we will present a new method that allows for better detection moving objects by combining the robustness of artificial immune systems (AIRS) in relation to local variations and the power of Gaussian mixtures (GMM) to model changes at the pixel level.

AIRS' role is to generate several GMM models for each pixel. The latter are subject to a filtering mechanism through the « *Competition for resources and candidate memory cell development* ». The remaining models are merged into a only candidate model and introduced as a new background model by the process « *Introduction of memory cell* ».

An evaluation on the Wollflower database proved the performance of our system compared to other state-of-the-art methods.

Keywords : video surveillance, background subtraction, GMM, AIRS, Segmentation in the foreground, pixel classification.

Table des matières

Liste des figures	iii
Liste des tableaux	iv
Liste des abréviations	v
Introduction Générale	1
1 État de l’art sur la soustraction de l’arrière-plan	3
1.1 Introduction	3
1.2 Sélection des caractéristiques	4
1.3 Les algorithmes utilisés	5
1.3.1 Modèle non paramétrique	5
1.3.2 Modèle paramétrique	14
1.3.3 Modèle sub-spatial	19
1.4 Conclusion	21
2 Système Immunitaire Artificiel de Reconnaissance	22
2.1 Introduction	22
2.2 Système Immunitaire Artificiel de Reconnaissance (AIRS)	23
2.2.1 Concepts	23
2.2.2 Description de l’AIRS	24
2.2.3 Classification	33
2.3 Conclusion	34
3 AIRS-GMM	35
3.1 Introduction	35

3.2	L'approche proposée	35
3.2.1	Prétraitement	38
3.2.2	Traitement	38
3.3	Conclusion	46
4	Implémentation	47
4.1	Introduction	47
4.2	Présentation des outils de développement	47
4.2.1	Python :	48
4.2.2	OpenCV (Open Source Computer Vision Library) :	48
4.2.3	PyQt :	48
4.3	Présentation de l'interface globale de l'application :	49
4.4	Tests et résultats	53
4.4.1	Choix des paramètres et définition des bases des données :	53
4.4.2	Évaluation des performances	54
4.4.3	Discutions des résultats	59
4.5	Conclusion	60
	Conclusion générale	61
	Bibliographie	62

Liste des figures

1.1	Taxonomie proposée	4
3.1	L'architecture globale du système proposé	37
3.2	Conversion de RGB à HSV	38
3.3	L'organigramme de la méthode proposée	45
4.1	L'interface Principal	49
4.2	Paramétrages de l'application	50
4.3	L'entrée de l'application sous forme d'images	50
4.4	L'entrée de l'application sous forme vidéo	51
4.5	L'entrée de l'application sous forme vidéo streaming	51
4.6	Les différents types de masque (médian, Erosion, Dilatation)	52
4.7	Documentation	52
4.8	Recall, Precision, F-measure de AIRS-GMM	58
4.9	Comparaison des résultats quantitatifs (F-measure) avec des méthodes connues de soustraction de fond sur la base des données Wallflower	59

Liste des tableaux

4.1	Description de la base de données Wallflower	53
4.2	Les résultats qualitatifs obtenus sur la base des données Wollflower	55
4.3	Les résultats qualitatifs sur une vidéo captée du monde réel	56
4.4	Comparaison des résultats quantitatifs (FP, FN, l'erreur totale) avec des méthodes connues de soustraction de fond sur la base des données Wallflower	57
4.5	Comparaison des résultats quantitatifs (Recall, Precision, F-measure) avec des méthodes connues de soustraction de fond sur la base des données Wallflower. .	58

Liste des abréviations

SVM (Support Vector Machine) : Machine à vecteur de support

SVR (Support Vector Regression) : Les séparateurs à vastes marges pour la régression

SVDD (Support Vector Data Description) : Les séparateurs à vastes marges pour la description de données

EM (Expectation Maximization) : Expectation Maximisation

KDE (Kernel Density Estimation) : Estimation de la densité du noyau

PCA (Principal Component Analysis) : Analyse en composantes principales

SL-PCA (Sequential Logistic Principal Component Analysis) : Analyse en composantes principales logistiques séquentielles

ICA (Independent Components Analysis) : analyse indépendante des composantes

SL-ICA (Sequential Logistic Independent Components Analysis) : analyse indépendante des composantes logistiques séquentielles

RGB (Red Green Blue) : Rouge Vert Bleu

HSV (hue, saturation, value) : Teinte Saturation Valeur

GMM (Gaussian Mixture Model) : Modèle de mélange gaussien

GGMM (Generalized Gaussian Mixture Model) : Modèle de mélange gaussien généralisée

SGG (Simple Generalized Gaussian) : Simple gaussienne généralisée

LMM (Minimum Length of Messages) : Longueur Minimale des Messages

NMF (Nonnegative Matrix Factorization) : Factorisation matricielle non négative

AIRS (Artificial Immune Recognition System) : Système immunitaire artificiel de reconnaissance

AIS (Artificial Immune System) : systèmes immunitaires artificiels

AT (Affinity Threshold) : seuil d'affinité

ATS (Affinity Threshold Scalar) : seuil d'affinité scalaire

Introduction Générale

Récemment, la sécurité des endroits publics et privés est devenue une tâche très importante. Cette tâche nécessite l'implantation de centaines de caméras de surveillance pour détecter les événements suspects et garder la sûreté au sein de ces endroits.

Devant ce nombre croissant de caméras, la gestion des stations de surveillance est devenue une tâche lourde qui nécessite une capacité d'attention énorme. En effet, il est très difficile, voire impossible, de placer un opérateur humain derrière chaque caméra. C'est dans ce cadre que les systèmes de vidéo de surveillance sont apparus offrant aux agents de surveillance des informations de haut niveau afin de faciliter la tâche de décision en un temps rapide en cas d'accident. Les différentes applications sur les vidéos de surveillance telle que : le suivi des objets en mouvement, la reconnaissance des formes, la détection des objets mobiles et l'interprétation des activités nécessitent d'entamer la phase de soustraction d'arrière-plan. La soustraction du fond est une opération de classification binaire qui donne à chaque pixel d'une séquence vidéo une étiquette, par exemple : les pixels des objets mobiles (premier plan) prennent la valeur 255 et les pixels des objets statiques sont étiquetés par 0.

La soustraction de l'arrière-plan a été étudiée depuis les années 1990 pour la détection des personnes, des véhicules, des animaux, etc. dans un flux vidéo. Cette opération est généralement constituée trois phases :

L'initialisation du modèle de l'arrière-plan : durant cette phase le modèle de l'arrière-plan est construit à partir d'un ensemble de frames dépourvus des objets en mouvement. La construction de ce modèle peut être effectuée de différentes manières (statistique, floue, neuro- inspirée, etc.).

La détection du premier plan : c'est la partie la plus importante, il s'agit d'une opération de soustraction du fond soit par l'application d'une méthode d'extraction ou par l'utilisation d'un vecteur de caractéristiques comme un détecteur d'objets mobiles. Cette partie commence dès qu'un objet mobile apparaît dans la scène.

Mise à jour du modèle de l'arrière-plan : cette phase n'est pas toujours nécessaire, puisque

les modèles qui reposent sur un apprentissage hors ligne ne maintiennent pas le modèle de l'arrière-plan au cours du processus de détection. En effet, les méthodes en ligne effectuent ce processus pour intégrer les objets qui ne bougent pas pendant une longue période comme faisant partie de l'arrière-plan. Cette mise à jour est effectuée au niveau des paramètres du modèle de fond appris à l'étape d'initialisation.

C'est dans ce cadre que nous allons proposer une nouvelle méthode menue d'un apprentissage en ligne pour la soustraction de l'arrière-plan. Cette méthode combine la robustesse des systèmes immunitaires artificiels (AIRS) par rapport aux variations locales et le pouvoir des mixtures de gaussiennes (GMM) à modéliser les changements au niveau pixel. L'AIRS génère pour chaque pixel plusieurs modèles GMM qui seront filtrés selon le processus « *Compétition de ressources et développement de cellule mémoire candidate* » pour garder uniquement les bons modèles. Les modèles restants après ce processus sont fusionnés dans un seul modèle candidat et introduits comme un nouveau modèle de l'arrière-plan par le processus « *Introduction de cellule mémoire* ».

Ce manuscrit est divisé en quatre chapitres et s'organise comme suit :

Chapitre 1 : est un état de l'art sur les méthodes de base dans la soustraction de l'arrière-plan. Nous avons également proposé une taxonomie afin d'extraire les avantages et les inconvénients de chaque catégorie de méthode.

Chapitre 2 : est consacré à la description du système immunitaire artificiel de reconnaissance (AIRS), ainsi que les différentes phases qui le compose.

Chapitre 3 : constitue le cœur de ce mémoire ou nous allons présenter l'architecture globale de notre système, détailler ces modules, expliquer le mode de fonctionnement par des pseudo algorithmes et définir les paramètres du système.

Chapitre 4 : a pour objectif de présenter les résultats qualitatifs et quantitatifs de notre méthode sur des bases de tests public et privée. Il contient également des comparaisons avec d'autres méthodes citées dans l'état de l'art.

Ce manuscrit se termine par une conclusion générale et quelques perspectives dans la détection et l'extraction des objets en mouvement.

Chapitre 1

État de l'art sur la soustraction de l'arrière-plan

1.1 Introduction

L'extraction de l'arrière-plan est une étape cruciale dans n'importe quel processus de traitement d'une séquence vidéo. L'objectif derrière ce processus consiste à classer les pixels en des pixels arrière-plan (représentant les objets statiques) et des pixels d'avant plan (représentant les objets en mouvement). À la fin de cette phase, les images entrantes sont transformées en des images binaires codées uniquement par le noir et le blanc, tel que les pixels des régions statiques prennent la valeur 0 et les pixels des régions dynamiques sont étiquetés par 255.

La robustesse d'un système de vidéosurveillance est extrêmement liée à l'exploitation d'une opération efficace qui doit assurer au mieux une bonne séparation entre l'arrière-plan et l'avant-plan.

Plusieurs travaux de recherche ont été proposés pour atteindre des résultats de bonne qualité en diminuant les mauvaises détections qui résultent principalement des variations de la luminosité et la dynamique des scènes. Toutes les contributions proposées et les travaux de recherche effectués peuvent être grossièrement divisés en deux groupes : des travaux qui se basent sur le choix des méthodes de classification et d'autres qui se focalisent sur la nature et le nombre de caractéristiques utilisées.

Dans ce chapitre nous allons présenter les méthodes les plus populaires dans la soustraction de l'arrière-plan avec leurs avantages et inconvénients.

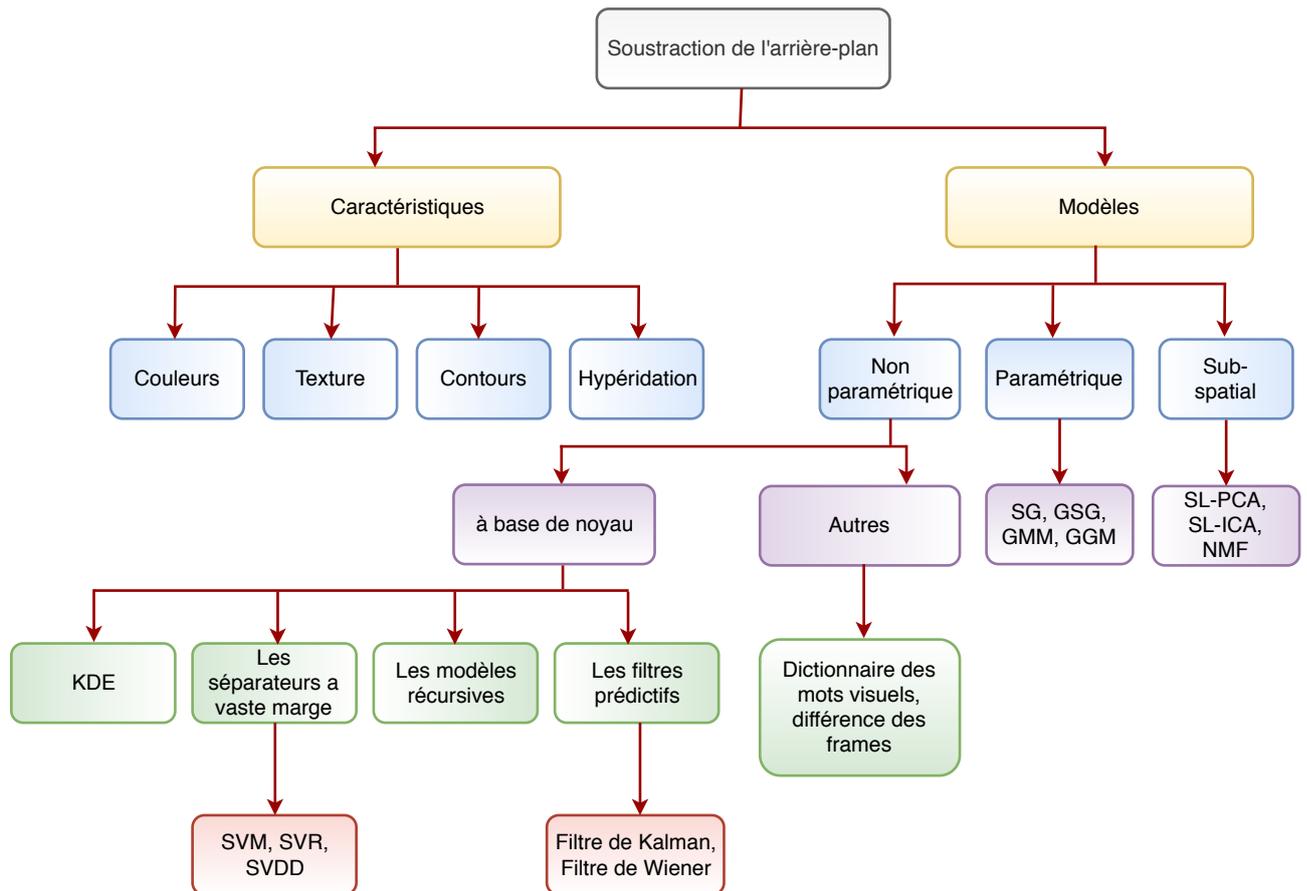


FIGURE 1.1 – Taxonomie proposée

1.2 Sélection des caractéristiques

Dans l'état de l'art, plusieurs approches se sont concentrées sur la combinaison de caractéristiques (couleurs, texture, contours, etc.) pour améliorer la qualité des résultats de la détection des objets en mouvement. Les auteurs dans [1] étudient les effets des différents espaces colorimétriques sur la qualité de segmentation. Dans le contexte des caractéristiques couleur, St-Charles et Bilodeau [2] exploite les caractéristiques binaires spatiotemporelles basées sur la couleur et la texture pour détecter les représentations locales au niveau du pixel. Dans [3], ils ont effectué une combinaison entre la texture, la couleur et l'intensité pour décrire le modèle de l'arrière-plan. Les auteurs dans [4] proposent une méthode robuste pour la détection des mouvements nommée M4CD, qui exploite les repères de couleur, texture, et d'autres caractéristiques hétérogènes. Les contours ont été largement utilisés pour la détection de mouvement. Dans [5], les auteurs ont proposé l'utilisation des descripteurs de contour à base de motif ternaire locale en plus des informations de couleur RGB pour identifier les pixels arrière-plan et les pixels avant plan. Un autre travail [1] propose une modélisation du fond par la fusion des

caractéristiques de texture, le mode de couleur RVB et le détecteur des contours Sobel.

1.3 Les algorithmes utilisés

Cette section est consacrée à la description des différents algorithmes utilisés dans la détection des changements.

1.3.1 Modèle non paramétrique

1.3.1.1 Les modèles à base de noyau

1.3.1.1.1 Estimation de la densité du noyau (KDE)

KDE représente un estimateur de densité de probabilité à partir d'un échantillon sans aucune information à priori sur la forme de distribution.

Afin de basculer vers une modélisation non paramétrique et créer un modèle de fond flexible dans les scènes dynamiques, les auteurs dans [6] proposent d'estimer la fonction de densité de probabilité de chaque pixel par un estimateur de noyau k . L'idée principale consiste à construire un modèle d'arrière-plan à partir des N valeurs récentes de chaque pixel, chacun lissé avec un noyau.

Étant donnée un échantillon (P_1, P_2, \dots, P_N) des récentes valeurs d'intensité d'un pixel. La fonction de densité de probabilité de ce pixel P_t par l'estimateur de noyau K est donné par :

$$P(P_t) = \frac{1}{N} \sum_{i=1}^N K(P_t - P_i) \quad (1.1)$$

Où K représente une fonction de noyau supposé suivre une distribution gaussienne munie d'une loi normale $N(0, \Sigma)$ (voire (1.2)).

$$P(P_t) = \frac{1}{N} \sum_{i=1}^N \frac{1}{2\pi^d/2|\Sigma|^{1/2}} e^{-1/2(P_t - P_i)' \Sigma^{-1} (P_t - P_i)} \quad (1.2)$$

Chaque fonction de noyau K est caractérisée par une largeur de la bande notée Σ .

Si la fonction de noyau possède différentes largeurs de bande, et s'il y a une indépendance entre les canaux de couleur alors la largeur de la bande sera donnée par :

$$\Sigma = \begin{pmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{pmatrix}$$

La fonction de densité de probabilité est donnée par :

$$P(P_t) = \frac{1}{N} \sum_{i=1}^N \prod_{j=1}^d \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{(P_{t,j}-P_{i,j})^2}{\sigma_j^2}} \quad (1.3)$$

Après l'estimation de la fonction de densité de probabilités, les pixels avant plan sont identifiés par l'équation suivante :

$$P(P_t) < T \quad (1.4)$$

Où T est un seuil fixé empiriquement.

Dans le cas où l'équation (1.4) n'est pas satisfaite, le pixel est associé à l'ensemble des pixels d'arrière-plan.

Afin de garantir la robustesse de la méthode, une mise à jour des paramètres de modèle (les échantillons) est proposée selon deux façons [6]. L'échantillon est mis à jour en utilisant une adaptation très rapide suivant une maintenance sélective basée sur la classification des objets en mouvement pour les détections sensibles, ou par une adaptation à long terme selon une maintenance non sélective qui adapte le modèle d'une manière très lente pour la modélisation des représentants les plus stables.

Cependant, ce modèle présente quelques inconvénients tels que : la nécessité de plusieurs échantillons pour le processus de décision et de mise à jour, ce qui oblige le système à sauvegarder les échantillons dans une mémoire et par conséquent augmenté le cout de calcul. En plus, KDE est sensible dans les applications en temps réel.

Quelques travaux de recherche ont proposé des solutions à ces problèmes, plusieurs entre eux se sont focalisés sur le changement de la fonction noyau elle-même : changer la bonde passante de K [7], remplacer la fonction gaussienne par une fonction rectangulaire [8], proposer de changer la fonction gaussienne par une fonction polynomiale [9]. D'autres travaux jouent sur la réduction du nombre d'échantillons : la taille du tampon d'image [10], échantillonnage diversifié [11] et échantillonnage à base de la loi Monte-Carlo [12].

1.3.1.1.2 Séparateurs à vaste marge

Les séparateurs à vaste marge est une famille d'algorithmes inventés en 1962 par Vapnik et al [13]. Ces algorithmes sont capables de résoudre les problèmes linéairement séparables et ils utilisent les fonctions basées noyau pour traiter les problèmes chevauchés. Ils possèdent aussi

un mécanisme de contrôle au niveau de la marge ou dans les vecteurs de support.

a) Machine à vecteur de support (SVM)

Dans le domaine de classification on trouve le classifieur SVM (machine à vecteur de support). Ce dernier cherche à déterminer l'hyperplan optimal qui sépare le mieux les éléments d'apprentissage en deux classes. Le meilleur hyperplan est celui qui maximise la marge (distance) entre l'hyperplan séparateur et les données. Dans le cas où l'ensemble des données n'est pas linéairement séparable, le SVM utilise une fonction noyau (polynomial, sigmoïde...) qui permet d'augmenter la dimension de l'espace de recherche afin de mieux les visualiser et les séparer.

Les auteurs dans [14] proposent un modèle d'arrière-plan en se basant sur les SVMs. Ils utilisent une version probabiliste de ces derniers vu que la sortie du SVM est uniquement binaire. Pour produire une sortie probabiliste, une transformation par une fonction sigmoïdale est effectuée [15] suivant l'équation (1.5) :

$$P(y = 1|f) = \frac{1}{1+\exp(Af+B)} \quad (1.5)$$

Avec :

y : étiquette de la classe binaire.

f : score de la sortie de la fonction de décision des SVMs.

A et B : sont adaptés en utilisant une estimation du maximum de vraisemblance de l'ensemble (f, y), sont dérivées en minimisant la fonction log-vraisemblance négative décrite par l'Équation (1.6) :

$$\min - \sum_i t_i \log(p_i) + (1 - t_i) \log(1 - p_i) \quad (1.6)$$

Où :

$$t_i = \frac{y_i+1}{2} \quad \text{et} \quad p_i = \frac{1}{1+\exp(Af_i+B)} \quad (1.7)$$

L'optimisation de l'équation (1.6) indique une minimisation à deux paramètres. Une division de chaque ensemble d'apprentissages en 80% est effectuée [14] pour la formation du SVM et l'autre 20% pour la minimisation des deux paramètres.

Afin de détecter les divers changements dans les scènes, un ensemble d'images contenant l'arrière-plan comme une base d'apprentissage est choisi. Chaque image est divisée en plusieurs blocs de taille 4×4 .

Un bloc de l'image est classé comme arrière-plan si la probabilité de sortie est supérieure à un certain seuil Th .

b) Les séparateurs à vastes marges pour la régression (SVR)

Il y a un autre type des séparateurs à vaste marge étendue dans la régression (SVR) pour traiter les problèmes complexes.

Étant un ensemble de données d'apprentissage, la régression adopte une fonction de classification qui permet de prédire au mieux le comportement des nouvelles variables. Parmi les types de la régression on trouve la régression linéaire qui cherche un modèle qui prédit une variable y à partir d'une seule ou une série de variables statistiques (x_i) de tel sort que l'erreur entre y et $f(x_i)$ soit la minimum possible. Le SVR utilise l'hyperplan séparateur optimal de SVM pour prédire la valeur de y .

Les travaux présentés dans [16] présentent une modélisation d'arrière-plan par le SVR, l'idée consiste à créer un modèle SVR séparé pour chaque pixel d'arrière-plan durant la phase d'apprentissage.

Une fois le modèle du fond est établi (après l'apprentissage), chaque SVR reçoit une valeur d'intensité du pixel associé, la sortie du SVR est comparée à un seuil pour décider la nature du pixel (arrière-plan ou avant plan).

Soit l'ensemble de données D d'un pixel structuré de la façon suivante :

$$D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}, \text{ ou } x_i \in R^P \text{ et } y_i \in R \quad (1.8)$$

Où X est un vecteur d'entrée ayant comme sortie le vecteur Y .

Chaque x_i représente l'intensité du pixel dans le frame i , et y_i représente le degré de confiance du pixel p d'être un pixel du fond. Le degré de confiance $f(x_i)$, est calculé par la fonction de régression linéaire suivante :

$$f(x_i) = \sum_{i=1}^N (\alpha_i, \alpha'_j) K(x_i, x_j) + \varepsilon \quad (1.9)$$

Avec :

$K(x_i, x_j)$: une fonction de noyau.

$\alpha_i, \alpha_j, \varepsilon$: des multiplicateurs de Lagrange obtenus par la résolution d'un problème d'optimisation.

Donc le SVR prend comme entrée l'intensité d'un pixel et produit comme sortie un degré de confiance. Un pixel est classé comme un pixel d'arrière-plan si son degré de confiance est compris entre S_l et S_h (l'équation (1.10)).

$$M_{x_i}^t = \begin{cases} 0 & \text{si } S_l < f(x_i) < S_h \\ 1 & \text{sinon} \end{cases} \quad (1.10)$$

c) Les séparateurs à vastes marges pour la description de données (SVDD)

Les séparateurs à vaste marge, pour la description de données (SVDD) sont utilisés pour déterminer les frontières autour d'un ensemble de données. Ils sont souvent appliqués dans le domaine de la reconnaissance de l'écriture manuscrite, la détection et la reconnaissance de visage et la détection des anomalies.

La première utilisation des SVDD comme une méthode de soustraction d'arrière-plan a été proposée par [17] en 2006. Ce modèle est appliqué sur des scènes contenant des arrière-plans presque stationnaires, et il est décrit par des frontières d'un ensemble de données utilisées comme des détecteurs de nouvelles données.

Les frontières sont représentées par une sphère caractérisées par un centre α et un rayon $R > 0$. La réduction de la valeur R^2 permet de minimiser le volume de la sphère afin de garder toutes les données d'apprentissage x_i à l'intérieur de la sphère. La fonction de minimisation est donnée par :

$$F(R, \alpha) = R^2 \quad (1.11)$$

Ou :

$$\|x_i - \alpha\|^2 \leq R^2, \forall i \quad (1.12)$$

Durant la phase d'apprentissage et pour construire le modèle arrière-plan, la distance vers le centre de la sphère est calculée. Étant donné un objet z , z sera accepté si l'équation suivante est satisfaite :

$$\|z - \alpha\|^2 = (z \cdot z) - 2 \sum_i \alpha_i (z \cdot x_i) + \sum_{i,j} \alpha_i \alpha_j (x_i \cdot x_j) \leq R^2 \quad (1.13)$$

Ou α_i est un multiplicateur de Lagrange.

Cette méthode permet de construire une descriptive des frontières de chaque pixel pour définir le modèle d'arrière-plan.

La précision du modèle ne dépend pas de la précision des fonctions de densité de probabilité estimées, de plus le modèle utilise moins d'espace mémoire comparé aux autres méthodes non paramétriques, cependant la méthode nécessite un temps de calcul très coûteux pour optimiser les paramètres de Lagrange.

1.3.1.1.3 Les modèles récursives

Représente une méthode d'estimation non paramétrique du modèle d'arrière-plan qui suit une distribution multimodale. La méthode prend en considération les modèles générés précédemment (filtrage récursif [17]) pour construire les modèles (modèle arrière-plan, modèle avant plan) de pixels.

Pour assurer une convergence rapide du modèle. Les auteurs dans [17] proposent une technique d'apprentissage basée sur l'historique de ces valeurs et le taux d'apprentissage. Le taux d'apprentissage utilisé à chaque exécution est représenté par l'équation (1.14).

$$\alpha_t = \frac{1-\alpha_0}{h(t)} + \alpha_0 \quad (1.14)$$

Avec :

$$\alpha_0 = \frac{1}{256 \times \alpha_M} \quad (1.15)$$

et :

$$h(t) = t - t_0 + 1 \quad (1.16)$$

Le taux d'oubli planifier est utilisé pour la suppression des valeurs qui n'existent plus dans l'arrière-plan (apparue il y a longtemps).

Pendant la phase d'apprentissage, seuls les modèles d'arrière-plan sont mis à jour par l'équation (1.16), une fois l'apprentissage est terminé et la classification des pixels a commencé, les modèles de l'avant-plan sont mis à jour par l'équation (1.18).

$$M_t^B(.) = [1 - \beta_t^B] M_{t-1}^B(.) + \alpha_t^B H_\Delta [x_t^B, M_{t-1}^B(.)] \quad (1.17)$$

$$M_t^F(.) = [1 - \beta_t^F] M_{t-1}^F(.) + \alpha_t^F H_\Delta [x_t^F, M_{t-1}^F(.)] \quad (1.18)$$

Ou $M_t(.)$ est le modèle à l' instant t, et $H_\Delta(.)$ une fonction de noyau locale avec une bande passante de longueur Δ qui permet de mettre à jour le modèle.

α_t, β_t t sont respectivement le taux d'apprentissage et le taux d'oubli planifier.

La classification des pixels est donnée selon le critère MAP sur :

$$\ln \left(\frac{M_B(t)}{M_F(t)} \geq k \right) \quad (1.19)$$

Ou K représente un seuil fixé empiriquement. Si le critère est satisfait, le pixel est classé comme un pixel avant-plan.

1.3.1.1.4 Les filtres prédictifs

Le principe sous-jacent à cette méthode consiste à utiliser l'historique des anciennes valeurs prises par un pixel pour prédire leur état actuel.

a)- Filtre de Wiener

Le filtre de Wiener a été utilisé pour la première fois dans les prédictions probabilistes de l'arrière-plan en 1999 [18]. Le filtre de Wiener permet d'estimer la valeur du pixel observé à l'instant t . Etant donnée un pixel X_t , le pixel est considéré comme un pixel d'arrière-plan si la valeur de ce dernier est plus proche de la prédiction. Dans le cas où l'écart entre la valeur du pixel et sa prédiction est significatif, le pixel est associé aux ensembles des pixels d'avant-plan. La prédiction de la valeur de X_t est définie comme suit :

$$X_t = -\sum_{k=1}^P \alpha_k \cdot X_{t-k} \quad (1.20)$$

Tel que :

X_t : présente la prédiction de la valeur du pixel X à l'instant t .

X_{t-k} : la valeur du pixel X à l'instant $t-k$

α_k : le coefficient de prédiction

P : est le nombre d'échantillons les plus récents.

La méthode calcule également l'erreur carrée entre la valeur du pixel et sa prédiction par :

$$E [e_t^2] = E [X_t^2] + \sum_{k=1}^P \alpha_k E [X_t, X_{t-k}] \quad (1.21)$$

Dans l'équation (1.21) α_k peut être défini par la covariance des valeurs de X_t . L'erreur carrée est utilisée dans le processus de classification des pixels, le pixel est détecté comme avant plan si l'équation suivante est satisfaite :

$$\frac{|X - X_t|}{4} < \sqrt{E [e_t^2]} \quad (1.22)$$

Cette modélisation est très robuste dans les scènes dynamiques, et aux bruits intervenant dans des intervalles réguliers.

b)- Filtre de Kalman

Le Filtre de Kalman est une méthode prédictive qui offre une bonne solution aux problèmes sous forme linéaire perturbée par un bruit supposé gaussien.

Dans la modélisation de l'arrière-plan, plusieurs versions de filtre de Kalman ont été proposées [19], [20], [21], [22], [15] la différence entre elles réside dans le choix des caractéristiques utilisées.

L'algorithme est basé sur la notion d'estimation pour décrire les informations sur l'état de système. Chaque estimation inclut toutes les informations précédentes du système. Étant donné un vecteur X_t qui représente l'état de pixel I à l'instant t. X_t contient la valeur d'intensité du pixel ainsi que son dérivé temporel (voir la représentation qui se suit).

$$X_t = \begin{bmatrix} I_t \\ \dot{I}_t \end{bmatrix} \quad (1.23)$$

L'estimation de chaque pixel d'une nouvelle image est donnée par :

$$\begin{bmatrix} I_t \\ \dot{I}_t \end{bmatrix} = \begin{bmatrix} I_{t-1} \\ \dot{I}_{t-1} \end{bmatrix} + k \cdot \left(I_t - H.A \cdot \begin{bmatrix} I_{t-1} \\ \dot{I}_{t-1} \end{bmatrix} \right) \quad (1.24)$$

Tel que :

A : la matrice du système qui décrivant l'évolution de l'arrière-plan.

H : nommé la matrice de mesure qui décrit la relation entre la mesure et l'état.

K : la matrice du gain de Kalman qui définit le taux d'apprentissage, Ou $K = \begin{bmatrix} \alpha \\ \alpha \end{bmatrix}$

Le filtre de Kalman détecte le changement dans le système en se basant sur la prédiction, il compare la valeur estimée par la prédiction et la valeur réelle mesurée, puis il effectue une pondération de leur différence.

$$m(X_t) = \begin{cases} 1 & \text{si } d(\hat{X}_t, X_t) \geq T \\ 0 & \text{sinon} \end{cases} \quad (1.25)$$

Avec :

T : une valeur fixé

$d(\hat{X}_t, X_t)$: la différence entre la valeur estimée du pixel X_t et la valeur réelle mesurée.

1.3.1.2 Autres modèles

1.3.1.2.1 Dictionnaire de mots visuels

Dans l'approche basée sur le dictionnaire de mots visuels, chaque pixel dans le modèle d'arrière-plan est caractérisé par un ensemble de variables, appelées mots visuels qui décrivent leurs états actuels.

Cette représentation a été proposée pour la première fois par [23], qui a défini chaque pixel

par un jeu de trois mots visuel (valeurs), le minimum d'intensité $\min(x)$, le maximum d'intensité $\max(x)$ et le maximum de la différence d'intensité entre des images consécutives $\text{dif } f(x)$ observées dans un intervalle de temps. Le modèle est initialisé par N images d'arrière-plan consécutives. Les images sont stockées dans un tableau V , tel que $V^j(x)$ représente la valeur de l'intensité du pixel x dans la j^{ieme} image. Le modèle initial d'un pixel x est donné par :

$$\begin{bmatrix} \min(x) \\ \max(x) \\ \text{dif } f(x) \end{bmatrix} = \begin{bmatrix} \min_j \{V^j(x)\} \\ \max_j \{V^j(x)\} \\ \max \{|V^j(x) - V^{j-1}(x)|\} \end{bmatrix} \quad (1.26)$$

Avec :

$$|V^j(x) - u| < 2\sigma \quad (1.27)$$

Et :

u : la moyenne du pixel x dans les N échantillons

α : la variance du pixel x dans les N échantillons

Après l'initialisation, le modèle d'arrière-plan subit une mise à jour régulière au cours du temps par deux méthodes différentes, une méthode locale au niveau du pixel, et l'autre globale qui prend en considération les changements physiques dans le fond [23].

Le pixel x de l'image I^t qui représente l'arrière-plan et l'avant-plan est défini selon équation (1.28) :

$$B(x) = \begin{cases} 0 & \text{si} \begin{cases} |I^t(x) - \min(x)| < S. \text{ dif } f_m \\ \text{ou } |I^t(x) - \max(x)| < S. \text{ dif } f_m \end{cases} \\ 1 & \text{sinon} \end{cases} \quad (1.28)$$

Avec :

$\text{dif } f_m$: Est le maximum de la différence absolue inter-images représentant l'arrière-plan.

S : un seuil fixe.

Il y a quelques extensions [24], [25], [26]. [24] qui proposent un dictionnaire de mots visuels avec une allure de cylindre, d'autres travaux des recherches se sont concentrés sur l'hybridation de dictionnaire de mots visuel avec d'autres techniques [27], [28], [29].

1.3.1.2.2 Différence des frames

La méthode la plus intuitive pour détecter les changements temporels d'intensité dans les séquences vidéos est de calculer la différence absolue (Δt) entre deux frames successifs [30]. La différence sera calculée pour chaque pixel $P_{x,y}$ dans le frame I_{t-1} et le pixel correspondant dans le frame I_t . Si l'image est codée en niveau de gris la différence est calculée par :

$$\Delta t_{x,y} = |I_t(x,y) - I_{t-1}(x,y)| \quad (1.29)$$

Dans le cas où l'image est codée en RGB la différence peut être calculée par plusieurs variantes telles que : la distance de Manhattan [31], distance euclidienne [32], ou la distance de Chebyshev [33].

Pour détecter les objets en mouvement, un masque binaire est appliqué sur le résultat de (Δt) selon un seuil prédéfini par l'utilisateur suivant l'équation suivantes :

$$I_{\text{avant-plan}}(x,y) = \begin{cases} 1 & \text{si } \Delta t_{x,y} > S \\ 0 & \text{sinon} \end{cases} \quad (1.30)$$

Le choix du seuil dépend de la vitesse de changement de la scène observée, une grande vitesse de mouvement des objets nécessite un grand seuil.

Il y a plusieurs améliorations de cette méthode, [34] propose une différence par trois frames consécutives, le résultat final du masque c'est la concaténation par le ET logique de deux masques résultants. Dans [35], les auteurs utilisent le même principe de [34] en prenant une image d'arrière-plan comme un troisième frame. Migliore et al [36] suggèrent une double différence afin de raffiner la qualité du résultat final. Il applique une différence entre deux frames consécutives puis il applique une autre différence entre le résultat précédent et une image de l'arrière-plan. On trouve aussi une méthode de soustraction de l'arrière-plan par l'image de référence qui consiste à calculer la différence entre une image I et une image de référence (image représente l'arrière-plan). Le choix de l'image de référence à une grande importance dans cette méthode.

1.3.2 Modèle paramétrique

1.3.2.1 La modélisation par une seule gaussienne

La modélisation de l'arrière-plan par une seule gaussienne [37] consiste à décrire les variations d'un pixel arrière-plan P dans un intervalle de temps Δt par une fonction de probabilité

gaussienne (1.31).

$$P(X) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(X-u)^2}{2\sigma^2}} \quad (1.31)$$

La gaussienne est caractérisée par la moyenne et la variance, ces deux paramètres permettent facilement de décrire le modèle d'arrière-plan, tel que la moyenne u représente la couleur dominante du pixel (P) courant et la variance σ représente la viabilité de P autour de la moyenne. Pour déterminer la nature d'un pixel P à l'instant t (pixel arrière-plan ou avant plan), on compare leur distance par rapport au modèle. Le pixel représente l'avant-plan si l'inégalité suivante est satisfaite :

$$\frac{|P_t - u_t|}{\sigma_t} > 2.5 \quad (1.32)$$

Dans le cas contraire, le pixel est classé comme un pixel d'arrière-plan. Afin d'adapter le modèle aux différents changements (luminosité, poussière, etc.) de la scène, une mise à jour récursive est appliquée sur la moyenne et la variance à chaque nouveau frame en fonction d'un biais α nommé le taux d'apprentissage qui détermine la vitesse d'adaptation du modèle.

$$u_{t+1} = (1 - \alpha)u_t + \alpha P_{t+1} \quad (1.33)$$

$$\sigma_{t+1}^2 = (1 - \alpha)\sigma_t^2 + \alpha (P_{t+1} - u_{t+1})(P_{t+1} - u_{t+1})^T \quad (1.34)$$

Il y a plusieurs améliorations qui ont été proposées dans cette modélisation [38], [39], [40], [41], [42], [43], [44] afin d'augmenter la qualité des résultats. Cependant, les contributions proposées restent limitées à cause des contraintes imposées par les auteurs sur l'environnement pour le bon fonctionnement de ces derniers. En plus, la modélisation par une seule gaussienne ne suffit pas pour sauvegarder tous les changements d'un pixel dans une séquence des frames.

1.3.2.2 Mixture de gaussiennes

Dans les scènes dynamiques, les variances des pixels sont tellement rapides, qu'une seule gaussienne n'arrive pas à mémoriser toutes les anciens états d'un pixel. Cela conduit à l'apparition d'une représentation multi-modèle de l'arrière-plan (multimodalité). La version initiale de mixture de gaussienne a été proposée par Friedman et Russel [45] en 1997 pour la surveillance du trafic routier. Le modèle représente chaque pixel par une mixture de trois gaussiennes et il utilise l'algorithme d'Expectation Maximisation (EM) comme un estimateur des paramètres. Cependant le modèle le plus populaire et le plus standard est celui de Stauffer and Grimson [46].

Dans ce modèle chaque pixel est représenté par une mixture de k distributions gaussiennes. La mise à jour des paramètres des gaussiennes s'effectue par l'algorithme de k -means au lieu l'algorithme d'Expectation Maximisation (EM).

La valeur de chaque pixel est comparée avec leur modèle GMM, qui représente les distributions de probabilité possible d'un arrière-plan. Une forte probabilité d'appartenance aux GMM indique un pixel arrière-plan.

La probabilité d'observer la valeur d'un pixel P dans le temps t est donnée par l'équation suivante :

$$P(P_t) = \sum_{i=1}^k w_{i,t} \eta(u_{i,t}, \Sigma_{i,t}, P_t) \quad (1.35)$$

Tel que :

P_t : La valeur du pixel dans le temps t .

k : Le nombre de gaussiennes associées dans le temps t .

$w_{i,t}$: Le poids calculé pour la i^{ieme} gaussienne dans le temps t .

$u_{i,t}$: La moyenne de la i ème gaussienne dans le temps t .

$\Sigma_{i,t}$: La matrice de covariance de la i ème gaussienne dans le temps t .

$$\eta(X_t, \mu_{i,t}, \Sigma_{i,t}) = \frac{1}{2\pi^d/2|\Sigma_{i,t}|^{1/2}} e^{-1/2(X_t - \mu_i)' \Sigma^{-1} (X_t - \mu_i)} \quad (1.36)$$

Après l'initialisation des paramètres des gaussiennes (w_k, u_k, σ_k), chaque pixel est comparé avec les K distribution existantes.

$$\frac{|P_t - u_i|}{\sigma_i} < 2.5 \quad (1.37)$$

Si l'équation (1.37) est vérifiée pour au moins une gaussienne, les paramètres de la gaussienne correspondante seront mis à jours par les équations ((1.38), (1.39), (1.40), (1.41)).

$$W_{i,t} = (1 - \alpha) W_{i,t-1} + \alpha \quad (1.38)$$

$$\mu_{i,t} = (1 - \varphi_i) \mu_{i,t-1} + \varphi_i P_t \quad (1.39)$$

$$\sigma_{i,t}^2 = (1 - \varphi_i) \sigma_{i,t-1}^2 + \varphi_i (P_t - \mu_{i,t})^T (P_t - \mu_{i,t}) \quad (1.40)$$

$$\varphi_i = \alpha \eta(P_t / \mu_i \sigma_i) \quad (1.41)$$

Pour les autres distributions, seuls les poids sont mis à jour selon l'équation (1.42).

$$W_{i,t} = (1 - \alpha) W_{i,t-1} \quad (1.42)$$

Après la mise à jour des paramètres, une étape de normalisation est effectuée pour garantir que la somme des poids est toujours égale à 1.

Si aucune distribution ne vérifie l'équation (1.37). La gaussienne la moins probable est remplacée par la moyenne, la variance et le poids du pixel courant suivant les équations (1.43), (1.44), (1.45)

$$\sigma_i^2 = \sigma_t^2 \quad (1.43)$$

$$W_t = \varphi_t \quad (1.44)$$

$$\mu_t = X_t \quad (1.45)$$

Pour décider la nature du pixel P_t s'il agit d'un pixel d'avant plan ou d'un pixel d'arrière-plan, les distributions sont ordonnées selon la valeur $\frac{W_{i,t}}{\sigma_{i,t}}$.

Les gaussiennes qui représentent l'état de P_t sont les β premières distributions qui vérifient l'équation (1.46).

$$\beta = \operatorname{argmin} \left(\sum_{i=1}^K W_{i,t} > B \right) \quad (1.46)$$

Où B détermine la partie minimale des données correspondant à l'arrière-plan, et $W_{i,t}$ est le poids de la distribution i .

Malgré le succès de la GMM surtout dans les environnements multimodaux et les environnements avec des changements d'éclairages graduels. Ce modèle présente certains inconvénients qui dépendent de l'initialisation des paramètres et du nombre de gaussiennes. Plusieurs travaux et contributions ont été proposés pour améliorer la qualité du GMM. Parmi ces méthodes on trouve ceux qui se sont concentrés sur l'amélioration de la vitesse d'adaptation du modèle [47] [48] [49]. D'autres se sont intéressés aux modèles hybrides tels que GMM et K-means [50], GMM et logique floue [51], GMM et arrière-plan adaptatif [52], Markov Random Fields [53], MOG avec PSO [54], GMM et arrière-plan Spotter [55] pour surmonter les inconvénients du GMM.

1.3.2.3 Simple gaussienne généralisée

En 2008, une modélisation du fond par un modèle gaussien généralisé (SGG) a été proposé [56]. L'idée consiste à substituer la gaussienne par la loi de Laplace. Dans la modélisation par la loi de Laplace on prend en considération que la variance des pixels.

Dans les scènes qui possèdent un arrière-plan presque statique, la modélisation de Laplace est plus proche qu'une modélisation gaussienne. Cependant la modélisation de Laplace est sensible dans les fonds qui possèdent divers changements.

La variation d'un pixel est définie selon l'équation (1.47) suivante :

$$P(X_t) = \frac{\rho\gamma}{2\Gamma\left(\frac{1}{\rho}\right)} e^{-(\gamma^\rho |X_t - \mu|^\rho)} \quad (1.47)$$

Ou :

$$\gamma = \frac{1}{\sigma} \left(\frac{\Gamma\left(\frac{3}{\rho}\right)}{\Gamma\left(\frac{1}{\rho}\right)} \right) \quad (1.48)$$

Avec :

$\Gamma(\cdot)$: représente une fonction gamma.

σ : la variance de la distribution.

Dans l'équation (1.47), si $\rho = 1$, la distribution est une distribution de Laplace, tandis que $\rho = 2$ définit une distribution gaussienne.

Afin de décider du modèle de chaque pixel, on calcule d'abord l'excès de kurtosis des m premiers frames. L'excès de kurtosis mesure la distribution des données par rapport à la distribution normale. L'excès de kurtosis pour les distributions de Laplace et de gaussiennes sont respectivement 3 et 0.

L'estimation des paramètres optimaux des modèles est effectuée par la maximisation de la vraisemblance de la valeur observée :

$$g2 = \frac{N \sum_{i=1}^N (x_i - \mu)^4}{\left(\sum_{i=1}^N (x_i - \mu)^2\right)^2} \quad (1.49)$$

Tel que N c'est le nombre des échantillons et μ la moyenne.

Les pixels sont classés en trois catégories selon deux seuils (k_1, k_2) suivant l'équation (1.50).

$$BD(p) = |L_1(p) - L_B(p)| \quad (1.50)$$

$$\left\{ \begin{array}{l} 0 \text{ si } BD(p) < k_1 b(p) \\ 1 \text{ si } k_2 b(p) \leq BD(p) \\ \text{suspicious si } k_1 b(p) \leq BD(p) \leq k_2 b(p) \end{array} \right.$$

Avec :

L_1 et L_B : sont respectivement les composantes de luminance du pixel p à l'instant actuel et dans

le modèle arrière-plan.

b : un paramètre d'échelle du modèle de fond.

La troisième région dans (1.50) sera raffinée par une composante de teinte pour traiter les zones d'ombres. Ensuite il est classé soit comme un pixel d'arrière-plan ou d'avant plan.

Cette méthode donne des résultats plus robustes. En plus, elle traite les problèmes de changement de luminosité dans le cas où les zones sont ombrées.

1.3.2.4 Mixture de gaussienne généralisée

Afin d'augmenter la précision dans la modélisation basée sur les mixtures de gaussiennes (GMM), les auteurs dans [57] proposent un modèle général de GMM (GGMM). Ce modèle est plus robuste dans les zones ombrée et plus flexible avec la forme des données. Dans le GGMM chaque pixel est représenté par son intensité dans l'espace couleur RGB. La probabilité d'observé la valeur du pixel courant en RGB est donnée par :

$$P(X_t) = \sum_{i=1}^k w_{i,t} \mathbf{n}(X_t, \mu_{i,t}, \Sigma_{i,t}, \lambda_i) \quad (1.51)$$

Avec :

$$\eta(X, \mu, \sigma, \lambda) = \prod_{j=1}^d A(\lambda_j) \exp\left(-B(\lambda_j) \left|\frac{X-\mu_j}{\sigma_j}\right|^{\lambda_j}\right) \quad (1.52)$$

Ou :

$$A(\lambda) = \frac{\left(\frac{\Gamma(\frac{3}{\lambda})}{\Gamma(\frac{1}{\lambda})}\right)^{\frac{1}{\lambda}}}{2\sigma\Gamma(\frac{1}{\lambda})} \quad et \quad B(\lambda) = \left(\frac{\Gamma(\frac{3}{\lambda})}{\Gamma(\frac{1}{\lambda})}\right)$$

K et d représente le nombre de distributions de chaque pixel, $w_{i,t}$ est le poids de la i^{eme} gaussiennes à l'instant t , $\mu_{i,t}$ et $\Sigma_{i,t}$ sont respectivement le vecteur moyenne et la matrice de variance de la i^{eme} distribution. Γ La fonction de densité de probabilité. Λ est un paramètre qui indique si le modèle utilise Laplace ou gauss, si $\lambda = 0$ la distribution est de type gaussienne, sinon si $\lambda = 3$ alors la distribution est de type Laplace. La mise à jour des paramètres reste la même que dans le GMM classique. Le nombre optimal de gaussiennes est calculé pour chaque frame par la minimisation du critère LMM (longueur minimale des messages).

1.3.3 Modèle sub-spatial

L'apprentissage sub-spatial offre une modélisation d'arrière-plan robuste aux changements de luminosité, puisqu'il prend en considération les informations spatiales.

Olivier et al en 2000 [58] ont développé le premier modèle du fond basé sur l'apprentissage par les composantes principales (SL-PCA), l'idée globale consiste à appliquer l'ACP sur les N images représentant l'arrière-plan pour extraire les valeurs propres. Les auteurs proposent d'abord de constituer un échantillon des images du fond prises dans des intervalles de temps non consécutifs et calculent leur moyenne I_u et leur matrice de covariance σ . Puis ils appliquent une diagonalisation sur la matrice σ afin de faciliter l'extraction des valeurs et vecteurs propres à partir de la matrice M .

$$M = v.\sigma.v^T \quad (1.53)$$

Pour diminuer la dimension de l'espace d'apprentissage, on ne garde que les K vecteurs propres qui ont des grandes valeurs propres ou $K < N$.

Après la phase d'apprentissage, chaque nouvelle image I_t est projetée sur l'espace généré dans l'étape précédente comme suit :

$$I'_t = (I_t - I_u)v^T \quad (1.54)$$

Le modèle d'arrière-plan est reconstitué aussi selon l'équation suivante :

$$B = v.I'_t + I_u \quad (1.55)$$

La différence absolue entre l'image courante I_t et le nouveau modèle d'arrière-plan B permet de détecter les objets en mouvement (l'équation (1.56)) [59]

$$|I_t - B| > T \quad (1.56)$$

Ce modèle souffre d'un ensemble d'inconvénients :

- La taille des objets en mouvement doit être petite.
- L'objet ne doit pas figurer dans le même emplacement durant une longue période.
- L'ACP consomme beaucoup de temps dans la mise à jour du modèle.
- Le modèle ne supporte pas les représentations multimodales.

Afin de corriger ces inconvénients plusieurs travaux de recherche ont été proposés, parmi ces travaux : [60], [61] qui suggèrent une compensation de l'erreur commise pour diminuer l'impact des objets en mouvement sur le modèle. [62] remplace la technique de projection et la reconstruction dans la phase de test de SL-ACP par un générateur du fond. Dans [63], les auteurs ont réglé le problème du temps d'exécution par l'analyse en composantes principales incrémentale, une version améliorée de ce travail est exposé dans [64] qui ajoute la notion de pondération à

chaque frame pour augmenter la robustesse du modèle. Dans le même contexte, une pondération au niveau de chaque pixel est utilisée [65]. Il y a des travaux qui se sont concentrés sur l'hybridation de l'ACP avec d'autres méthodes ACP avec modèle gaussien [66]. On trouve aussi d'autres versions de l'ACP : ACP incrémental 2D [67], ACP blocs spatio-temporels [68] et apprentissage multi spatiale [69].

Dans le même contexte (sub_spatial) les auteurs dans [70] proposent un schéma rapide (SL-ICA) pour la soustraction de l'arrière-plan avec l'analyse indépendante des composantes (ICA). Un autre travail [71] présente une décomposition du contenu de la vidéo par une factorisation matricielle non négative (NMF) incrémentale, etc.

1.4 Conclusion

Dans ce chapitre nous avons données une brève présentation sur quelques méthodes de modélisation du fond. Ces méthodes permettent de détecter les régions correspondant à des objets en mouvement ainsi les régions statiques dans les systèmes de vidéosurveillance. Cependant ces méthodes souffrent de plusieurs problèmes causés par la complexité des scènes, les variations dans l'éclairage, les arrière-plans dynamiques et les ombres d'objets statiques, etc. ce qui influence sur la qualité des résultats obtenus et augmente le nombre de mauvaises détections. Plusieurs changements et contributions ont été proposés pour surmonter ces problèmes, certains ont réussi à surmonter quelques problèmes dans des conditions bien précises, malheureusement, il n'existe aucune méthode robuste qui permet de pallier tous les problèmes rencontrés dans l'analyse de la scène.

Chapitre 2

Systeme Immunitaire Artificiel de Reconnaissance

2.1 Introduction

Dans les années récentes, il y a eu un intérêt considérable sur l'exploration et l'exploitation des techniques bio-inspiré pour résoudre des problèmes complexes soit dans le domaine d'apprentissage automatique ou dans le domaine d'optimisation. Ces techniques consistent à modéliser les systèmes naturels ou à décrire l'adaptation biologique des êtres vivants dans leur environnement.

Des nombreuses approches bio-inspiré peuvent servir de métaphores pour les systèmes informatiques [72]. Parmi les approches qui ont réussi, les systèmes immunitaires artificiels (AIS). Ces systèmes regroupent un ensemble de techniques exploitant les divers aspects du système immunitaire naturel, exemple le concept de mémorisation par les cellules mémoire, la reconnaissance des cellules de soi et non-soi, la maturation des cellules, le clonage, etc. Tous ces aspects ont été exploité par les chercheurs pour créent des applications impliquant la sécurité informatique, la détection des intrusions sur le réseau et la reconnaissance des formes [73] [74] [75]. Dans la littérature, on distinct plusieurs algorithmes de systèmes immunitaires artificiels, Réseaux immunitaire [76], la sélection négative [73], la sélection clonale [77] et le système immunitaire artificiel de reconnaissance (AIRS) [72]. Dans ce chapitre nous allons présenter l'algorithme AIRS et détailler les étapes principales de son fonctionnement.

2.2 Système Immunitaire Artificiel de Reconnaissance (AIRS)

Dans le système immunitaire naturel et lors de l'invasion d'un élément étranger, le corps utilise un ensemble de mécanismes de défense afin d'empêcher ce dernier à quitter le corps. Ce système biologique a une capacité intelligente de reconnaître les cellules qu'il appartient (soi) et les cellules étrangères (non-soi). Inspiré de ce mécanisme biologique, l'informatique a développé des systèmes qui peuvent servir la métaphore de quelques aspects du système immunitaire naturel. L'AIRS représente un type de système immunitaire artificiel défini comme un algorithme d'apprentissage supervisé. L'algorithme décrit la réponse immunitaire provoquée lors de la détection d'un antigène, et il repose sur la représentation antigène- anticorps. Cette représentation permet de mesurer un degré de correspondance entre les données à reconnaître et les classes d'appartenance. Après avoir rencontré un antigène, le système engendre un ensemble de cellules mémoires qui constitue plus tard le modèle d'apprentissage.

Avant de décrire les phases importantes du processus d'apprentissage d'AIRS nous allons d'abord présenter quelques concepts et notions de base.

2.2.1 Concepts

Dans cette sous-section on va donner quelques définitions des mots clés utilisés dans l'AIRS [78].

- **nc** : Le nombre de classes des antigènes.
- **MC** : Ensemble de cellules mémoires et mc un membre ou un individu de cet ensemble.
- **mc.c** : La classe d'une cellule mémoire mc , où $mc.c \in C = \{1, 2, \dots, nc\}$.
- **mc.f** : Le vecteur de caractéristiques d'un mc où $mc.f_i$ représente la valeur de l^i ^{ème} caractéristique de mc .
- **AG** : Représente l'ensemble des antigènes et ag un individu de cet ensemble.
- **ag.c** : La classe d'un antigène ag , où $ag.c \in C = \{1, 2, \dots, nc\}$
- **ag.f** : Le vecteur de caractéristique d'un antigène et $ag.f_i$ représente la valeur de l^i ^{ème} caractéristique d' ag .
- **ARB** : Balle de reconnaissance artificielle (ARB) appelée également cellule B. Il se compose d'un anticorps, d'un compteur du nombre de ressources détenues.
- **AB** : Un ensemble des ARBs, ou la population des cellules existante. ab représente un seul ARB ($ab \in AB$).

- **ab.c** : La classe d'un ARB donnée, où $ab.c \in C = \{1, 2, \dots, nc\}$.
- **ab.stim** : Le degré de stimulation d'un ARB avec un antigène ag.
- **ab.resources** : Le nombre des ressources détenues par l'ARB (ab).
- **TotalNumResources** : Le nombre total de ressources du système autorisées.
- **Affinité** : Une mesure de similarité entre deux anticorps ou antigène-anticorps, cette valeur est toujours entre 0 et 1, et elle est calculée en utilisant une distance euclidienne entre deux vecteurs de caractéristiques.
- **Stimulation** : Fonction utilisée pour mesurer la réponse d'un ARB à un antigène ou à un autre ARB. La stimulation est l'inverse de l'affinité.
- **Seuil d'affinité (AT)** : C'est l'affinité moyenne de tous les antigènes de l'ensemble d'apprentissages.
- **Seuil d'affinité scalaire (ATS)** : Valeur comprise entre 0 et 1 qui intervient dans la décision du remplacement des cellules mémoires.
- **Taux de mutation** : Une valeur comprise entre 0 et 1 indiquant la probabilité qu'une caractéristique donnée d'un ARB soit mutée.
- **Taux de clonage** : Une valeur entière fixée par l'utilisateur pour déterminer le nombre de clones mutés qu'un ARB donné peut produire.
- **Taux hyper mutation** : Une valeur entière choisie par l'utilisateur utilisé avec le taux de clonage pour déterminer le nombre de clones mutés qu'un ARB donné peut produire.
- **Seuil de stimulation** : Une valeur comprise entre 0 et 1 qui permet de déterminer si le critère d'arrêt pour l'entraînement d'un antigène est atteint ou non.

2.2.2 Description de l'AIRS

Au court de cette sous-section, nous allons détailler la version originale de l'AIRS. En particulier nous essayons de donner un aperçu sur le principe de fonctionnement des différentes étapes, les équations, ainsi que les pseudo algorithmes utilisés dans le processus d'apprentissage du classifieur AIRS.

L'algorithme possède quatre grandes phases, l'initialisation, l'identification de la cellule mémoire et génération des ARBs, compétition pour les ressources et nomination d'une nouvelle cellule mémoire et enfin l'introduction de la cellule mémoire.

a) Initialisation

On peut la considérée comme une phase de prétraitement, puisque durant cette phase tous

les vecteurs de caractéristiques (les antigènes ag) sont normalisés de telle sorte que la distance entre deux vecteurs reste dans l'intervalle $[0,1]$. Pendant cette phase, on calcule le seuil d'affinité AT qui intervient plus tard dans le processus de remplacement des cellules mémoires mc . L'affinité est représentée par une distance euclidienne. La partie initialisation se termine par une initialisation de l'ensemble des cellules mémoires MC et l'ensemble ARB . Le choix de ces deux derniers est effectué d'une manière aléatoire. Cette étape reste une étape facultative on peut initialiser les deux ensembles MC et ARB avec \emptyset . Les éléments de l'ensemble MC et les ARB s sont regroupés en des classes ou des groupes homogènes contenant respectivement les mc et les ARB s de la même classe ($MC_{ag.c}, AB_{ag.c}$).

Algorithme 1 Initialisation

Entrée : AG (les vecteurs de l'ensemble d'apprentissages)

Sortie : MC initial, ARB initial

Variables

Début

nbrMC, nbrARB : Le nombre initial des mc et ARB s choisis par l'utilisateur

min f_i : L' i^{eme} minimale dans les vecteurs de caractéristiques

max f_i : L' i^{eme} valeur maximale dans les vecteurs de caractéristiques

Fin

Pour chaque $ag \in AG$ **faire**

 Normalisé ag :

Pour chaque $ag.f_i \in ag.f$ **faire**

$$ag.f_i \leftarrow \frac{(ag.f_i - \min f_i)}{(\max f_i - \min f_i)} |ag.f|$$

Fin pour

Fin pour

$$AT \leftarrow \frac{\sum_{i=1}^{|AG|-1} \sum_{j=i+1}^{|AG|} Affinity(ag_i, ag_j)}{\frac{|AG|(|AG|-1)}{2}}$$

$nbrMC \leftarrow$ valeur

Tant que $|MC| \leq nbrMC$ **faire**

 Choisir aléatoirement un $ag \in AG$

$$MC_{ag.c} \leftarrow MC_{ag.c} \cup ag$$

Fin tant que

```

nbrARB ← valeur
Tant que |AB| ≤ nbrARB faire
    Choisir aléatoirement un ag ∈ AG
    ABag,c ← ABag,c ∪ ag
Fin tant que

```

Les phases suivantes sont appliquées pour chaque antigène $ag \in AG$.

b) Identification de la cellule mémoire et génération d'ARB

Cette phase est composée de deux sous phases essentielles : identification de la cellule mémoire mc_{match} et génération d'ARB. La cellule mémoire mc_{match} est identifiée à partir de l'ensemble MC en choisissant le mc qui possède la même classe que l'antigène ag et qui a la plus grande stimulation avec ce dernier. Si $MC_{ag,c}$ est vide dans ce cas $mc_{match} \leftarrow ag$ et $MC_{ag,c} \leftarrow MC_{ag,c} \cup ag$. La stimulation est identifié par :

$$Stimulation(ag,mc) = 1 - affinité(ag, mc)$$

Algorithme 2 Identification de cellule mémoire mc_{match}

```

Entrée : ag ∈ AG.
Sortie : une cellule mémoire  $mc_{match}$ 
Si ( $MC_{ag,c} = \emptyset$ ) alors
     $mc_{match} \leftarrow ag$ 
     $MC_{ag,c} \leftarrow MC_{ag,c} \cup ag$ 
Sinon
    maxStim ← Valeur initiale
    Pour j = 1 to | $MC_{ag,c}$ | faire
        valStim ← Stimulation ( $ag, mc_j$ )
        Si (valStim > maxStim) alors
             $mc_{match} \leftarrow mc_j$ 
            maxStim ← valStim
    Fin si
Fin pour
Fin si

```

La deuxième sous phase (génération de l'ensemble ARB) consiste à effectuer une opération de

mutation sur le mc_{match} pour engendrer de nouveaux mc nommé les clones. Le nombre de clones engendrés est calculé par :

$$NumClones = Taux\ hyper\ mutation \times Taux\ de\ clonage \times Stimulation(ag, mc_{match})$$

Le taux hyper mutation et le taux de clonage sont deux valeurs entières choisies par l'utilisateur, le taux de mutation est une valeur entre 0 et 1 fixée aussi par l'utilisateur.

Algorithme 3 Génération d'ARB

Entrée : mc_{match}

Sortie : ARB (AB)

Variables

Début

MU : l'ensemble des ARBs mutés (clones)

mut : une valeur logique indiquant si l'opération de mutation produit un nouvel élément ou non

Fin

$MU \leftarrow \emptyset$

$MU \leftarrow MU \cup \text{prendARB}(mc_{match})$

$stim \leftarrow \text{stimulation}(ag, mc_{match})$

$NumClones \leftarrow Taux\ hyper\ mutation \times Taux\ clonage \times stim$

Tant que $|MU| < NumClones$ **faire**

$mut \leftarrow \text{faux}$

$mc_{clone} \leftarrow mc_{match}$

$mc_{clone} \leftarrow \text{mutation}(mc_{clone}, mut)$ **Si** ($mut = \text{vraie}$) **alors**

$MU \leftarrow MU \cup \text{prendARB}(mc_{clone})$

Fin si

Fin tant que

$AB \leftarrow AB \cup MU$

La fonction prendARB (x) renvoie un ARB avec x comme anticorps de cette ARB. La fonction de mutation utilisée est la suivante :

Algorithme 4 Fonction de mutation**Entrée :** x : vecteur de caractéristique, b : valeur logique**Sortie :** x , b **Pour chaque** $x.f_i \in x.f$ **faire** Change \leftarrow drandom() Changeto \leftarrow drandom() **Si** (Change < taux de mutation) **alors** $x.f_i \leftarrow$ Changeto \times valeur de normalisation $b \leftarrow$ Vraie **Fin si****Fin pour****Si** ($b =$ Vraie) **alors** Change \leftarrow drandom() Changeto \leftarrow Irandom() **Si** (Change < taux de mutation) **alors** $x.c \leftarrow$ Changeto **Fin si****Fin si**

drandom() : Une fonction qui renvoie un nombre aléatoire entre 0 et 1.

Irandom() : Une fonction qui renvoie un nombre entier entre 0 et n_c .**c) Compétition de ressources et développement d'une cellule mémoire candidate**

L'objectif de cette phase est de filtrer l'ensemble AB afin de ne garder que les ab le plus représentatives et le plus similaires à l'antigène ag. Pour cela [72] propose une technique qui organise la survie des individus au sein de la population AB. Cette technique comporte un processus de partage des ressources cumulées selon la nature de la classe de l'antigène ag. Les ab qui possèdent la même classe d'ag ont la moitié des ressources cumulées, l'autre moitié sera divisée sur le reste des ab qui n'appartiennent pas à la même classe de l'antigène ag. Les membres de chaque classe d'AB ont une allocation de ressource maximale, si la somme des ressources allouées dans une classe dépasse leur allocation maximale, les ressources supplémentaires seront d'abord supprimées des ARBs les moins stimulés de cette classe. Tout ab possédant un nombre de ressources inférieur ou égal à zéro par ce processus sera supprimée du pool des ARBs (voir

l'algorithme qui se suit).

Algorithme 5 Stimulation, allocation des ressources et suppression des ARBs

Entrée : AB

Sortie : AB

Variables

Début

MAX, MIN : deux valeurs entières

resAllowed : le nombre de ressources alloué pour une classe

TotalNumRessources : le nombre total des ressources du système

Fin

minStim \leftarrow MAX

maxStim \leftarrow MIN

Pour chaque ab \in AB **faire**

stim \leftarrow stimulation (ag, ab)

Si (stim < minStim) **alors**

minStim \leftarrow stim

Si (stim > maxStim) **alors**

maxStim \leftarrow stim

ab.stim \leftarrow stim

Fin pour

Pour chaque ab \in AB **faire**

Si (ab.c = ag.c) **alors**

ab.stim $\leftarrow \frac{ab.stim - minStim}{maxStim - minStim}$

Sinon

ab.stim $\leftarrow 1 - \frac{ab.stim - minStim}{maxStim - minStim}$

Fin si

ab.ressources \leftarrow ab.stim \times Taux de clonage

Fin pour

i \leftarrow 1

Tant que $i < nc$ **faire**

$resAllowed \leftarrow \sum_{j=1}^{|AB_i|} ab_j.ressources, ab_j \in AB_i$

Si ($i = ag.c$) **alors**

$numRes Allowed \leftarrow \frac{TotalNumRessources}{2}$

Sinon

$numRes Allowed \leftarrow \frac{TotalNumRessources}{2 \times (nc-1)}$

Fin si

Tant que $resAllowed > numResAllowed$ **faire**

$numResRemove \leftarrow resAllowed - numResAllowed$

$ab_{remove} \leftarrow \operatorname{argmin}_{ab \in AB_i}(ab.stim)$

Si ($ab_{remove}.ressources \leq numResRemove$) **alors**

$AB_i \leftarrow AB_i - ab_{remove}$

$resAllowed \leftarrow resAllowed - ab_{remove}.ressources$

Sinon

$Ab_{remove}.ressources \leftarrow Ab_{remove}.ressources - numResRemove$

$resAllowed \leftarrow resAllowed - numResRemove$

Fin si

Fin tant que

$i \leftarrow i+1$

Fin tant que

L'ensemble AB produit par le mécanisme précédent contient les ARBs le plus stimulé avec l'antigène ag et qui ont un succès dans l'acquisition des ressources. Après cette étape, un critère d'arrêt sera calculé pour vérifier si la stimulation des ARBs avec ag est suffisante ou non.

Étant donné un vecteur \vec{S} de taille nc contenant la stimulation moyenne de chaque classe d'AB. Le critère d'arrêt est atteint si et seulement si $S_i \geq$ seuil de stimulation pour tous les éléments de \vec{S} .

$$s_i = \frac{\sum_{j=1}^{|AB_i|} ab_j.Stim}{|AB_i|}, ab_j \in AB_i$$

Pendant ce temps, chaque $ab \in AB$ a une chance de produire une progéniture mutée afin d'augmenter la diversification des membres au sein de l'ensemble AB. Le mécanisme de mutation est presque similaire à la phase de génération d'ARB (voir la fonction qui se suit).

Algorithme 6 Mutation de l'ARB survivant**Entrée :** AB**Sortie :** AB**Variables****Début****MU :** L'ensemble de clones d'ARB mutés**mut :** Une valeur logique indiquant si l'opération de mutation a produit un nouvel élément ou non**Fin**MU $\leftarrow \emptyset$ **Pour chaque** ab \in AB **faire**rd \leftarrow drandom()**Si** (ab.stim > rd) **alors**NumClones \leftarrow ab.stim \times clonal_ratei \leftarrow 1**Tant que** (i \leq NumClones) **faire**mut \leftarrow fauxab_{clone} \leftarrow abab_{clone} \leftarrow mutation (ab_{clone}, mut)**Si** (mut = vraie) **alors**MU \leftarrow MU \cup ab_{clone}**Fin si**i \leftarrow i + 1**Fin tant que****Fin si****Fin pour**AB \leftarrow AB \cup MU

Après que la mutation des ARBs est terminée, le critère d'arrêt est testé. Si le critère d'arrêt est atteint, on passe à la prochaine étape, sinon les processus de cette phase (**Compétition de ressources et développement d'une cellule mémoire candidate**) seront répétés jusqu'à ce que le

critère d'arrêt soit vérifié. La seule exception durant cette répétition, est que pendant le passage entre les processus de cette phase (sauf le premier passage), si le critère d'arrêt est atteint après l'algorithme de **Stimulation, allocation des ressources et suppression des ARBs**, la mutation des ARBs n'est pas effectuée.

Une fois le critère d'arrêt est vérifié, la cellule mémoire candidate est choisi à partir de l'ensemble des ARBs restants. $mc_{candidate}$ représente l'ARB de la même classe d'ag la plus stimulée.

d) Introduction de la cellule mémoire

Cette phase permet d'introduire à l'ensemble des cellules mémoires MC existant une nouvelle cellule mémoire $mc_{candidate}$. Le $mc_{candidate}$ qui a une stimulation avec ag supérieur à celui de mc_{match} est ajoutée à l'ensemble MC. Dans le cas où l'affinité de $mc_{candidate}$ est inférieur à $(AT \times ATS)$, le $mc_{candidate}$ remplace le mc_{match} identifié précédemment.

Une fois que l'évaluation de la cellule $mc_{candidate}$ est terminée, on passe à l'apprentissage de l'antigène suivant. L'apprentissage se termine quand tous les antigènes sont présentés au système.

Algorithme 7 Introduction de la cellule mémoire

Entrée : $mc_{candidate}$ (généralisé dans l'étape précédente)

Sortie : MC

Variables

Début

AT : seuil d'affinité

ATS : seuil d'affinité scalaire

Fin

$CandStim \leftarrow stimulation(ag, mc_{candidate})$

$MatchStim \leftarrow stimulation(ag, mc_{match})$

$CellAff \leftarrow affinité(mc_{candidate}, mc_{match})$

Si $(CandStim > MatchStim)$ **alors**

Si $(CellAff < AT \times ATS)$ **alors**

$MC \leftarrow MC - mc_{match}$

Fin si

$MC \leftarrow MC \cup mc_{candidate}$

Fin si

2.2.3 Classification

Après l'entraînement, l'ensemble MC est préparé pour la classification. La classification est effectuée par le classifieur KNN (k plus proche voisin) [79] selon un vote de majoritaire des k cellules mémoires les plus proches à l'élément en entrée. La recherche sur les k plus proches voisins dans KNN, nécessite un parcours itératif de tous les éléments du modèle d'apprentissage (MC). Une fois les k voisins sont identifiés le nouvel élément est étiqueté par la classe de la majorité de voisins.

Algorithme 8 KNN (k plus proche voisin)

Entrée : x données a classifié, MC : l'ensemble d'apprentissages, k : le nombre des voisins

Sortie : La classe de x

Variables

Début

i : compteur

k-nn : l'ensemble des k voisins les plus proches

Fin

i ← 1

Tant qu'il y a un élément de MC n'est pas traité faire

 Calculer la distance (x, mc)

Si $i \leq k$ **alors**

 Ajouter mc a l'ensemble de k-nn

 i ← i+1

Sinon Si mc est plus proche de x qu'au moins un de k-nn **alors**

 Supprimé l'élément le plus éloigné de l'ensemble k-nn

 Ajouter mc a l'ensemble de k-nn

Fin si

Fin tant que

x.classe ← $\text{argmax}_{c \in k\text{-nn.classe}} (\text{Occurrence}(k\text{-nn}))$

2.3 Conclusion

Dans ce chapitre nous avons détaillé les différentes étapes du système immunitaire artificiel de reconnaissance (AIRS) qui est introduit dans [72]. Nous avons également présenté quelques pseudo code sur le fonctionnement de l'AIRS pour une meilleure visibilité.

L'AIRS a connu un grand succès dans la résolution des problèmes d'optimisation et de classification. Dans le chapitre qui se suit, nous allons essayer de modéliser la soustraction de l'arrière-plan par l'AIRS.

Chapitre 3

AIRS-GMM

3.1 Introduction

Récemment, la méthode GMM a obtenu un succès considérable dans la détection des objets en mouvement dans les vidéos. Cependant cette méthode reste sensible aux scènes ayant une grande variation de luminosité et des zones camouflées. Ces problèmes sont dus à la nature du modèle GMM qui représente les variations de pixel par un nombre fixe de gaussiennes durant toute la durée de vie du système.

Afin de pallier les problèmes décrits précédemment, nous allons présenter dans ce chapitre une nouvelle méthode efficace et robuste pour la soustraction de l'arrière-plan basé sur la méthode GMM et l'algorithme AIRS.

3.2 L'approche proposée

Nous avons proposé une nouvelle méthode pour la soustraction de l'arrière-plan à partir des scènes vidéos capturées par une caméra fixe. Le système proposé est une combinaison entre la méthode GMM et l'algorithme bio-inspiré AIRS. Le GMM est utilisé comme un outil de classification des pixels tandis que l'AIRS est utilisé pour générer de nouveaux modèles GMM qui représentent l'arrière-plan. Chaque pixel est modélisé par un vecteur de caractéristique contenant plusieurs gaussiennes ou chacune d'elle est représentée par la moyenne, la variance et le poids. Notre système commence d'abord par modéliser chaque pixel avec une seule gaussienne. Ce nombre va subir des changements au fur et à mesure que le système fonctionne en prenant en considération la dynamique de l'environnement. L'apprentissage des modèles est effectué sui-

vant le processus traditionnel d'une GMM. Après la création du modèle d'arrière-plan, notre système commence la phase de classification des pixels, il choisit parmi les modèles GMM existant, celui qui représente le mieux l'état du pixel en cours de traitement, puis il effectue une classification des pixels selon la méthode GMM de base. Afin de prendre en considération tous les changements des pixels d'arrière-plan, nous avons décrit les variations de pixel par plusieurs modèles GMM. Ces modèles sont créés par le processus de génération des ARBs de l'algorithme AIRS. Après cette étape, l'ensemble des nouveaux clones sera filtré par le processus de compétition de ressources de l'AIRS pour garder uniquement les meilleurs GMM qui représentent l'arrière-plan. Les modèles GMM restants sont combinés dans un seul modèle GMM candidat, qui peut être introduit comme un nouveau modèle d'arrière-plan selon le processus d'introduction de cellule mémoire.

Pour atteindre notre objectif, notre système passe par trois principales étapes : Prétraitement, extraction de l'arrière-plan et poste traitement. L'architecture globale de notre système est illustrée dans la figure 1.

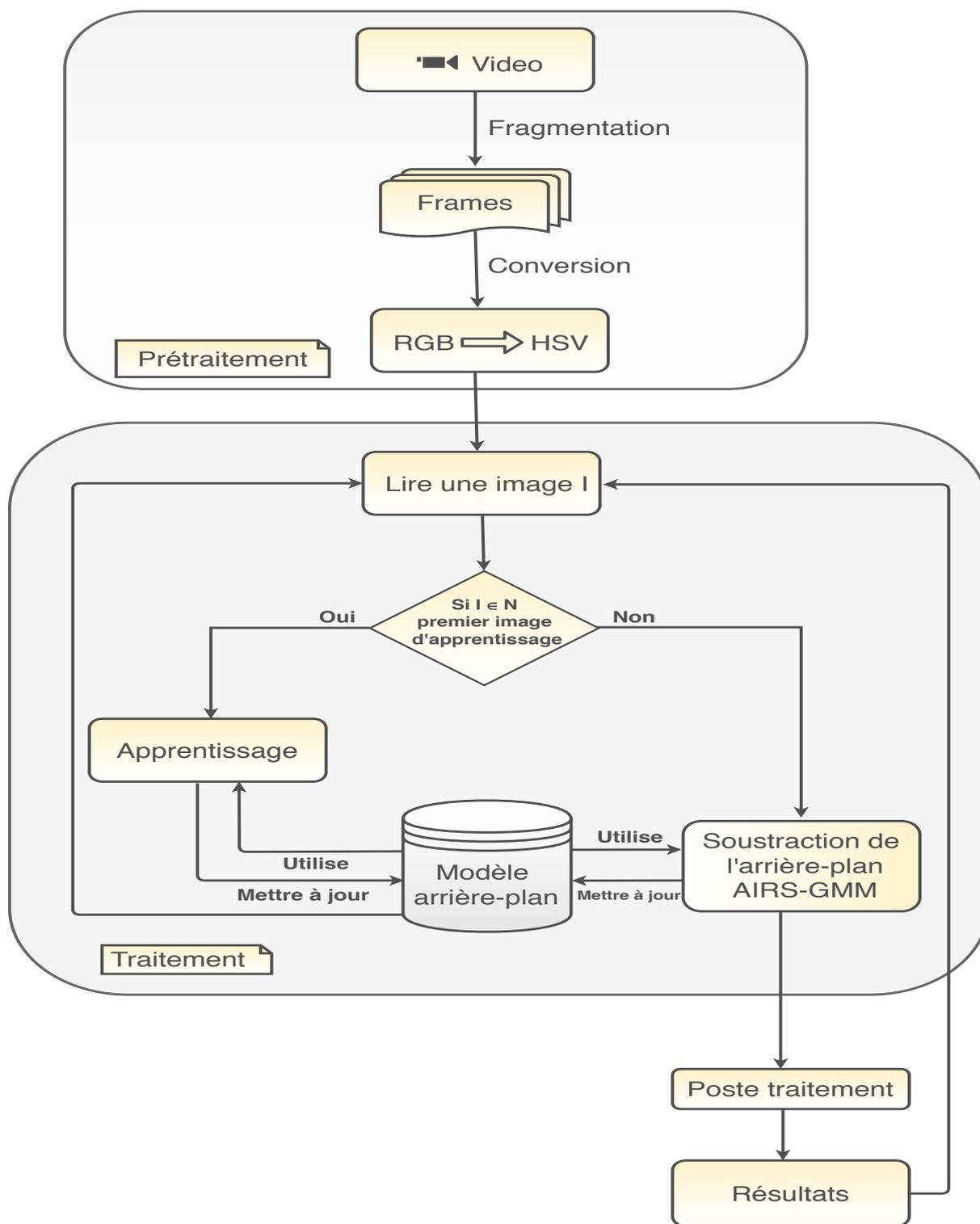


FIGURE 3.1 – L'architecture globale du système proposé

3.2.1 Prétraitement

La phase de prétraitement représente une étape préliminaire dans n'importe quel processus de traitement de vidéo. Pendant cette phase le flux vidéo est découpé en un ensemble d'images 2D nommé les frames, ses images sont codées en mode RGB. Cependant, ce mode et d'après les travaux effectués dans [80] est très sensible aux changements de luminosité et il ne garantit pas la création d'un modèle puissant. Pour cette raison nous avons choisi de transférer les images en mode HSV. Cet espace nous a permis de spécifier les couleurs d'une façon plus proche à la perception humaine [80], puisqu'il permet une séparation entre l'intensité de luminosité représentée par la composante V et les propriétés chromatiques représentées par les composantes H et S, ce qui permet de détecter les changements dans les hautes variations d'intensité d'éclairage.

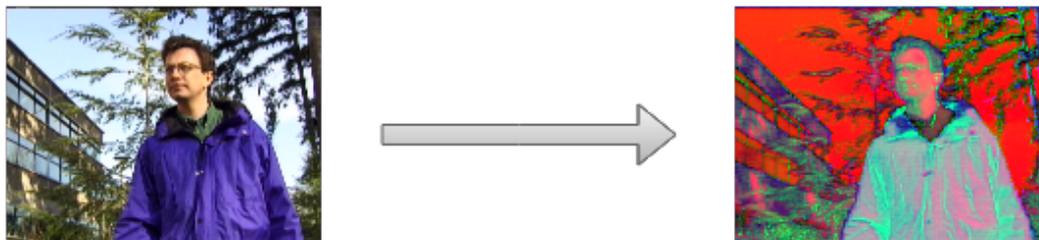


FIGURE 3.2 – Conversion de RGB à HSV

3.2.2 Traitement

Cette phase représente le corps de notre travail, puisqu'il comporte les modules essentiels de processus de soustraction de l'arrière-plan basé sur l'approche GMM et l'algorithme AIRS. Les modules de cette phase ainsi que leur fonctionnement sont détaillés dans les sous-sections qui suivent.

3.2.2.1 Apprentissage

Tous processus de soustraction d'arrière-plan nécessitent une phase d'apprentissage. Durant cette phase le modèle d'arrière-plan est créé et mis à jour par un ensemble d'images dépourvues des objets en mouvement (représente uniquement l'arrière-plan).

Tout d'abord notre système est initialisé par un seul modèle de mélange de n Gaussiennes pour chaque pixel. Chaque gaussienne g_i dans le modèle GMM est représenté par : valeur de pixel P_i , la moyenne u_i , la variance σ_i et un poids w_i .

$$g_i = \{P_t, \mu_i, \sigma_i, w_i\} \quad (3.1)$$

Après l'initialisation du modèle, les paramètres des gaussiennes sont mis à jour par l'approche GMM en effectuant, pour chaque pixel, une recherche sur les gaussiennes qui le correspondent le mieux à dans les K distributions existantes.

$$\frac{|P_t - \mu_i|}{\sigma_i} < 2.5 \quad (3.2)$$

Si l'Équation (3.2) est satisfaite pour au moins une gaussienne, les paramètres de la gaussienne correspondante sont mis à jour par l'Équations ((3.3), (3.4), (3.5), (3.6)).

$$W_{i,t} = (1 - \alpha)W_{i,t-1} + \alpha \quad (3.3)$$

$$\mu_{i,t} = (1 - \varphi_i)\mu_{i,t-1} + \varphi_i P_t \quad (3.4)$$

$$\sigma_{i,t}^2 = (1 - \varphi_i)\sigma_{i,t-1}^2 + \varphi_i (P_t - \mu_{i,t})^T (P_t - \mu_{i,t}) \quad (3.5)$$

$$\varphi_i = \alpha \eta (P_t / \mu_i \sigma_t) \quad (3.6)$$

Pour les distributions qui ne satisfont pas l'Équation (3.2), seuls les poids sont mis à jour selon l'équation (3.7).

$$W_{i,t} = (1 - \alpha)W_{i,t-1} \quad (3.7)$$

Après la mise à jour des paramètres, une étape de normalisation est effectuée pour garantir que la somme des poids est toujours égale à 1.

Si aucune distribution ne vérifie l'équation (3.2). La gaussienne la moins probable est remplacée par la moyenne, la variance et le poids du pixel courant suivant les équations (3.8), (3.9), (3.10)

$$\sigma_i^2 = \sigma_t^2 \quad (3.8)$$

$$W_t = \varphi_t \quad (3.9)$$

$$\mu_t = X_t \quad (3.10)$$

3.2.2.2 Soustraction de l'arrière-plan

Après la création du modèle de l'arrière-plan dans la phase précédente, on commence la phase de classification des pixels en pixels d'arrière-plan et pixels d'avant plan. Pour surmonter les problèmes de la méthode GMM de base, nous avons proposé une nouvelle méthode de soustraction de l'arrière-plan en se basant sur l'approche GMM et l'algorithme AIRS. Notre méthode exploite la robustesse de l'algorithme AIRS pour générer de nouveaux modèles d'arrière-plan. Le système proposé considère la valeur H de chaque pixel comme un anticorps et l'ensemble des modèles GMM qui le correspondent comme des cellules mémoires qui ont reconnu cet antigène. L'ensemble des cellules mémoires constituent les modèles d'arrière-plan. Nous allons dans ce qui suit détailler le principe de fonctionnement de la méthode AIRS-GMM.

La méthode AIRS-GMM est constituée de cinq grandes phases, quatre phases de l'algorithme AIRS plus une phase de classification des pixels par l'approche GMM.

a) La phase d'initialisation du modèle AIRS-GMM :

Cette phase permet d'initialiser les paramètres de l'algorithme AIRS décrit dans les chapitres précédents. Dans notre cas, l'ensemble des cellules mémoire MC est initialisé par le modèle d'arrière-plan (GMM) obtenu dans la phase d'apprentissage. Cette étape est importante pour assurer une convergence rapide du système de reconnaissance de l'AIRS. L'ensemble des ARBs est considéré vide dans cette phase sachant que sa construction sera effectuée au fur et à mesure que le système fonctionne.

La mesure d'affinité décrite dans la version de l'AIRS de base, n'est pas adéquate dans notre contexte à cause de l'utilisation des gaussiennes. Pour cela, nous avons proposé de changer l'affinité par la moyenne de la distance de chaque valeur du pixel avec les gaussiens d'arrière-plans qui lui correspondent dans le modèle GMM (voir l'équation (3.11)).

$$\text{Affinité}(ag, mc) = \text{Moyenne} \left(\frac{|P_i - u_i|}{\sigma_i} < 2.5 \right) \quad (3.11)$$

Puisque nous avons changé la mesure d'affinité, le seuil d'affinité sera calculé par :

$$AT = \frac{\sum_{i=1}^n \sum_{j=1}^m \left(\frac{P_{i,j} - u_{i,j}}{\sigma_{i,j}} < 2.5 \right)}{n \times m} \quad (3.12)$$

Tel que :

n, m : le nombre de lignes et de colonnes d'une image.

$P_{i,j}, u_{i,j}, \sigma_{i,j}$: sont respectivement la valeur du pixel, la moyenne et la variance du modèle fourni par GMM durant l'apprentissage.

Cette étape est une étape exceptionnelle effectuée une seule fois après que l'apprentissage est terminé. Les autres étapes sont répétées à chaque itération durant la phase de classification.

b) Identification de cellule mémoire et classification de pixel

Avant de classer les pixels, nous avons identifié parmi l'ensemble des cellules mémoire MC qui existent, une seule cellule mémoire qui représente le mieux l'état du pixel courant P_t . La cellule mémoire choisie, nommée mc_{match} reflète le modèle GMM qui possède un grand nombre des gaussiens d'arrière-plans qui vérifient l'équation (3.2).

$$mc_{match} = \operatorname{argmax}_{GMM_j} \left\{ \frac{P_t - u_i}{\sigma_i} < 2.5, g_i \in GMM_j \right\} \quad (3.13)$$

Avec : i varie de 1 à $|GMM_j|$ et j varie de 1 au nombre des modèles GMM de chaque pixel.

Une fois le mc_{match} identifié, les gaussiennes sont ordonnées selon la valeur de $w_{k,t}/\sigma_{k,t}$ pour décider si P_t représente l'arrière-plan ou non. Les B premiers gaussiens qui vérifient l'Équation (3.14) sont sélectionnés pour décrire l'arrière-plan.

$$\beta = \operatorname{argmin} \left(\sum_{k=1}^b w_{k,t} > B \right) \quad (3.14)$$

Où :

B : détermine la partie minimale des données correspondant à l'arrière-plan.

b : le nombre de gaussiennes dans le modèle GMM.

S'il existe au moins une gaussienne parmi les B gaussiennes sélectionnées qui vérifie l'équation (3.2), le pixel est classé comme un pixel du fond, et les paramètres du modèle mc_{match} seront mis à jour selon les Équations (3.3), (3.4), (3.5), (3.6).

c) Génération des ARBs :

Les modèles mc_{match} des pixels classés comme arrière-plan sont mutés avec un taux de mutation entre 0 et 1 pour générer de nouveaux clones. Ces clones sont placés avec le mc_{match} dans l'ensemble AB. La mutation est effectuée au niveau des Gaussiennes g_i qui vérifient l'Équation (3.2). La fonction de mutation a été adoptée selon notre vecteur de caractéristique.

Fonction de mutation

Entrée : $g_t(P_t, u_t, \sigma_t, w_t)$ une gaussienne, b : valeur logique

Sortie : g_t , b

Pour chaque $g_{ti} \in g_t, i \in [2, 4]$ **faire**

```

change ← valeur aléatoire entre 0 et 1
changeto ← une valeur entre 0 et 254
Si change < taux de mutation alors
    Si i== 4 alors
         $g_{ti} \leftarrow$  valeur aléatoire entre 0 et 1
    Sinon
         $g_{ti} \leftarrow$  changeto  $\times$  valeur de normalisation+1
        b ←Vraie
    Fin Si
Fin Si
Fin Pour

```

Le nombre de clones est calculé par :

$$\text{NumClones} = \text{Taux hyper mutation} \times \text{Taux de clonage} \times \text{Affinité}(P_t, mc_{\text{match}}) \quad (3.15)$$

Le taux de clonage et le taux d'hyper mutation sont deux valeurs entières choisies par l'utilisateur.

d) Compétition de ressources et développement d'une cellule mémoire candidate

Durant cette phase, l'ensemble AB généré précédemment est filtré de sorte qu'on garde que les modèles GMM dont au moins une de leur gaussiennes muté g_i vérifie toujours l'équation (3.2).

L'ensemble GMM restant après le processus de Compétition de ressources est combiné dans un seul modèle GMM nommé $mc_{\text{candidate}}$, ce modèle est généré en calculant la moyenne de toutes les ab restantes (l'équation (3.16)).

$$mc_{\text{candidate}} = \text{Moyenne}(AB) \quad (3.16)$$

Notons que le but de cette phase est de garder les modèles GMM qui représentent le mieux l'arrière-plan pour être introduits comme de nouvelles cellules mémoires dans la phase qui suit.

e) Introduction de cellule mémoire

Cette phase représente la dernière phase dans notre méthode. Le $mc_{\text{candidate}}$ ayant une affinité inférieure à l'affinité de mc_{match} sera introduire dans le modèle MC comme une nouvelle cellule mémoire. Dans le cas où l'affinité entre mc_{match} et $mc_{\text{candidate}}$ est inférieur a la valeur de $AT \times ATS$ (équation (3.17)), le mc match est supprimé de l'ensemble MC. En effet,

notre méthode permet d'introduire à chaque itération un nouveau modèle pour les pixels classés comme arrière-plan.

$$\text{Affinité}(mc_{match}, GMM_{candidate}) < AT \times ATS \quad (3.17)$$

Avec :

ATS : est une valeur entre 0 et 1 choisit par l'utilisateur.

Affinité $(mc_{match}, GMM_{candidate})$: est la moyenne entre l'affinité (P_t, mc_{match}) et $(P_t, mc_{candidate})$

Ce processus est répété pour chaque itération de la phase **Identification de cellule mémoire et classification de pixel** jusqu'à **Introduction de la cellule mémoire**.

AIRS –GMM :

Entrée : un ensemble de modèles GMM_m, P_t : la valeur de pixel

Sortie : la valeur de pixel (0 ou 255)

$$mc_{match} = \operatorname{argmax}_{GMM_j \in GMM_m} \left(\frac{P_t - u_{gieGMM_j}}{\sigma_{gieGMM_j}} < 2.5 \right)$$

Si $mc_{match} == null$ **alors**

$$P_t = 255$$

Sinon

$P_t \leftarrow$ Appliquer GMM de base sur mc_{match} pour décider

Si $P_t == 0$ **alors**

$$MU \leftarrow \emptyset$$

$$MU \leftarrow MU \cup mc_{match}$$

$$\text{affinité}_{mc_{match}} \leftarrow \text{Moyenne} \left(\frac{P_t - u_{mc_{match}i}}{\sigma_{mc_{match}i}} \right)$$

$$\text{NumClone} \leftarrow \text{Taux hyper mutation} \times \text{Taux de clonage} \times \text{affinité}_{mc_{match}}$$

Tant que $(|MU| < \text{NumClones})$ **faire**

$$mut \leftarrow \text{faux}$$

$$mc_{clone} \leftarrow mc_{match}$$

$$mc_{clone}, mut \leftarrow \text{mutation}(mc_{clone}, mut)$$

Si $(mut == \text{vraie})$ **alors**

$$MU \leftarrow MU \cup mc_{clone}$$

Fin Tant que

$$AB \leftarrow AB \cup MU$$

Pour chaque $ab \in AB$ **faire**

S'il n'existe pas aux moins un $ab_i \in ab$ qui vérifie $\frac{P_i - u_{ab_i}}{\sigma_{ab_i}} < 2.5$ au niveau des gaussiennes mutés **alors**

$AB \leftarrow AB - ab$

Fin Pour

$mc_{candidate} \leftarrow \text{moyenne}(AB)$

$\text{affinitéCandidate} \leftarrow \text{affinité}(P_t, mc_{candidate})$

Si ($\text{affinitéCandidate} < \text{affinité}_{mc_{match}}$) **alors**

$\text{affinitéCellule} \leftarrow (\text{affinitéCandidate} + \text{affinité}_{mc_{match}})/2$

Si ($\text{affinitéCellule} < AT \times ATS$) **alors**

$MC \leftarrow MC - mc_{match}$

$MC \leftarrow MC \cup mc_{candidate}$

Fin Si

Fin Si

Fin Si

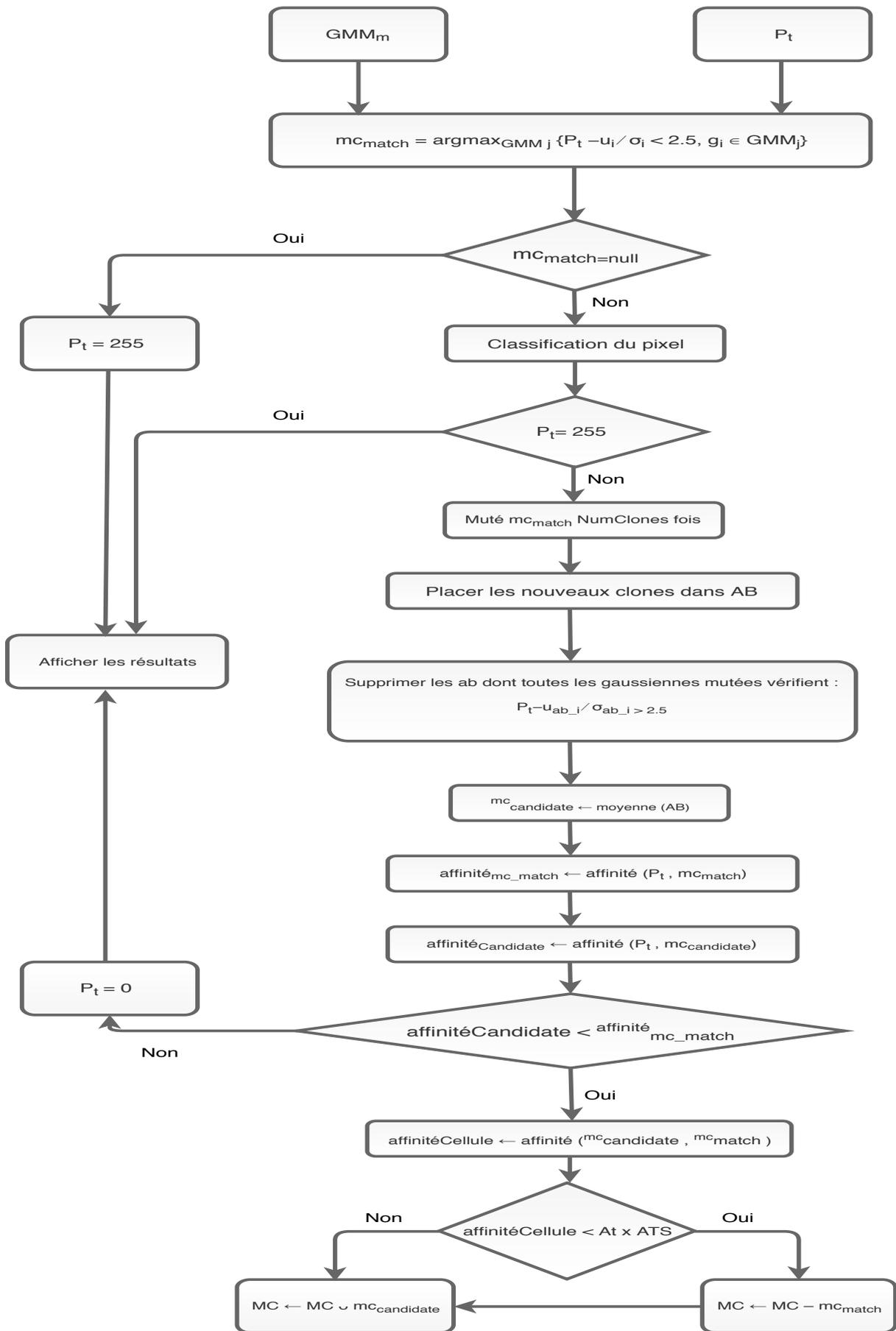


FIGURE 3.3 – L’organigramme de la méthode proposée

3.2.2.3 Poste traitement

Après l'opération de soustraction de l'arrière-plan, les images produites sont toutes codées en mode noir et blanc, telle que les pixels qui représentent l'arrière-plan prennent la valeur 0 et les autres pixels sont étiquetés par la valeur 255. Cependant, les méthodes utilisées dans cette opération ne garantissent pas que les pixels qui ont formé l'avant-plan (objet en mouvement) sont tous connectés. De plus il y a de fausses détections qui sont apparues sous la forme de pixels blanc isolés au niveau de l'arrière-plan ou sous forme de trous (pixels noirs) au niveau des objets en mouvement.

Afin de surmonter ces problèmes, nous avons appliqué un filtre morphologique (dilatation, érosion) et un filtre médian sur les images bruitées.

L'opération de dilatation permet de boucher les trous à l'intérieur des objets détectés, par contre l'opération d'érosion et le filtre médian suppriment les pixels isolés au niveau de l'arrière-plan générés par l'effet de la poussière et le changement de l'éclairage. De plus le filtre médian permet de corriger les contours des objets.

3.3 Conclusion

Les GMMs ont réalisé un succès considérable dans le domaine de la soustraction d'arrière-plan. Cependant cette approche souffre de certains inconvénients qui peuvent influencer sur la qualité des résultats.

Dans ce chapitre nous avons donné une brève explication de notre méthode utilisée pour la soustraction de l'arrière-plan. Le système proposé est une combinaison entre l'approche GMM et l'algorithme bio-inspiré AIRS. En effet, l'algorithme AIRS nous a permis de créer de nouveau modèle GMM qui représentent l'arrière-plan au lieu d'utiliser un seul modèle GMM durant toute l'exécution, de plus on a bénéficié de l'approche GMM dans la classification des pixels en des pixels arrière-plan et pixels avant plan.

Chapitre 4

Implémentation

4.1 Introduction

En informatique, le concept de l'implémentation désigne l'élaboration d'un produit final à partir d'un document de conception qui répond aux exigences requises dans la phase d'analyse de besoin tout en respectant les contraintes de performance et fiabilité du système.

Durant ce chapitre, nous allons présenter les résultats de l'expérimentation de notre méthode proposée (AIRS-GMM) sur les vidéos de la base de données publique Wollflower et sur une vidéo captée au niveau du département d'informatique de l'université 8 mai 1945 Guelma.

Pour mieux visualiser la performance de notre système par rapport aux autres méthodes de l'état de l'art, nous avons calculé quelques critères de tests (FP, FN, Recall, Precision, F-measure) qui permettront de simplifier la tâche de comparaison.

4.2 Présentation des outils de développement

Le système que nous avons présenté dans le chapitre précédent est implémenté en Python sur un ordinateur muni d'un processeur Intel Core i7 de septième génération et d'une capacité de mémoire de 8 Go. Notons aussi que notre application fonctionne sous le système d'exploitation Linux (distribution Mint 19), et Windows.

Nous avons utilisé la bibliothèque "OpenCV", "PyQT5" respectivement dans le traitement des images et l'implémentation de l'interface graphique. Les sous-sections suivantes permettent de donner plus de détail sur ces outils.

4.2.1 Python :

C'est un langage de programmation orienté objet de haut niveau qui possède une sémantique dynamique intégrée. Ce langage est dû principalement au développement des applications rapide et des sites Web, puisqu'il offre aux utilisateurs une option de typage et de liaison dynamique [82].

Pour installer python sous Linux, il suffit de taper la commande « **sudo apt-get install python3.6** » dans l'invite de commande. La commande « **sudo apt install python3-pip** » permet d'installer le package manager de Python. Si vous voulez installer Python sous Windows, il faut le télécharger gratuitement à partir de ce lien « <https://www.python.org/downloads/release/python-360/> », puis l'installer. L'installation de Python package manager sous Windows est détaillé dans [83].

4.2.2 OpenCV (Open Source Computer Vision Library) :

L'OpenCV représente une bibliothèque des logiciels libres qui offre une infrastructure des applications de vision par ordinateur et d'apprentissage automatique. OpenCV permet d'accélérer l'utilisation de la perception machine dans les produits commerciaux, et il compte plus de 2500 algorithmes optimisés pour : la détection et la reconnaissance des visages, l'identification des objets, la classification des actions humaines dans des vidéos, l'extraction des modèles 3D d'objets, l'assemblage des images pour produire une image haute résolution d'une scène entière, etc. [84].

L'installation de cette bibliothèque dans Linux est effectuée par la commande « **pip3 install opencv-python** ». Dans Windows l'installation de OpenCV est effectuée sous anaconda (voir [85]) à travers la commande « **conda install opencv** » ou bien par python package manager selon la commande « **pip install opencv-python** ».

4.2.3 PyQt :

C'est une interface Python pour Qt, la plus puissantes, et la plus populaire dans les bibliothèques graphiques multiplateformes. Il représente un mélange du langage de programmation Python et de la bibliothèque Qt [86]. Pour installer PyQt il suffit de taper respectivement sous Linux et Windows les commandes : « **sudo apt-get install python3-pyqt5** », « **pip3 install pyqt5** ».

4.3 Présentation de l'interface globale de l'application :

Lors de l'ouverture de l'application, on trouve l'interface de notre système AIRS-GMM qui offre aux utilisateurs la possibilité de saisir les paramètres du système de façon manuelle ou automatiquement. Cette interface nous permet de fixer le nombre d'images de traitement, de tests et le nombre d'images utilisées pour l'initialisation du système. On peut également choisir le répertoire de sauvegarde des résultats en utilisant le bouton Folder. L'interface principale contient également un menu qui offre quatre options :

- **Treatment** : pour convertir les images en HSV, ou enregistrer la vidéo directement en HSV pour le traitement.
- **View** : Affiche les résultats obtenus directement dans le répertoire ou il se trouve.
- **Mask** : Permet aux utilisateurs d'appliquer des opérations de post traitement sur les résultats obtenus (masque médian, Erosion, Dilatation).
- **Help** : Contient la documentation sur notre application, des tests unitaires et un petit tutoriel sur l'utilisation de l'application.

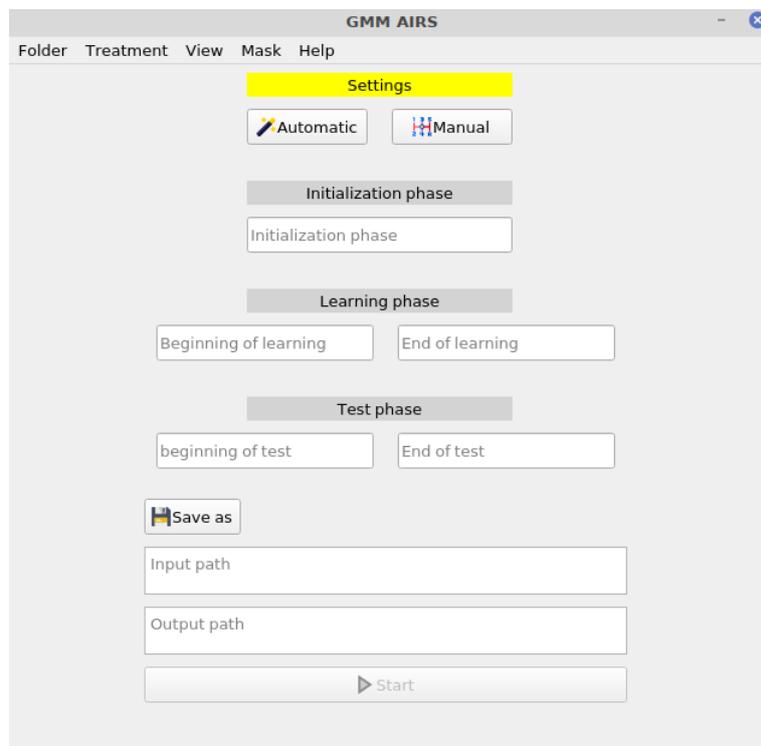


FIGURE 4.1 – L'interface Principale

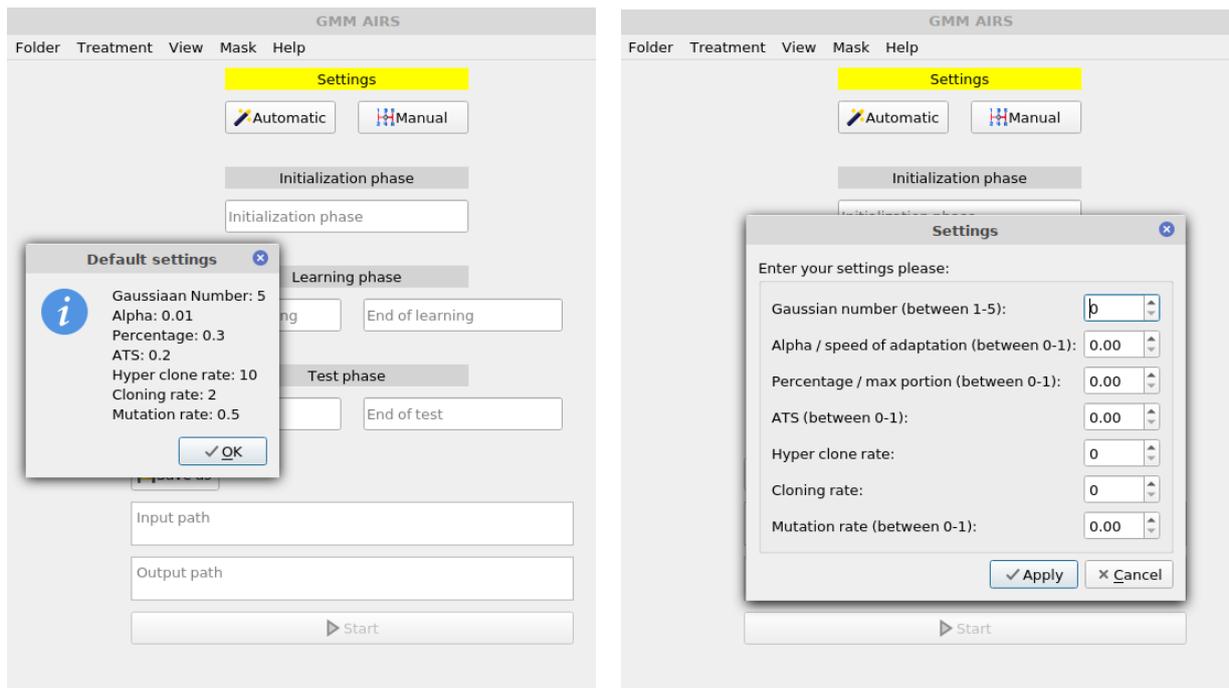


FIGURE 4.2 – Paramétrages de l'application

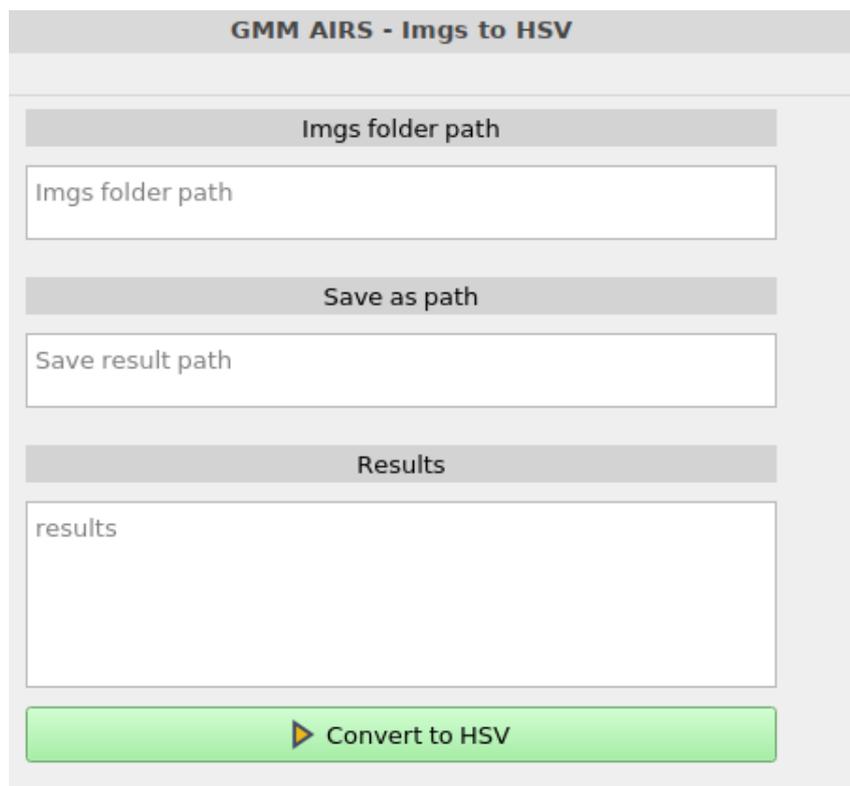


FIGURE 4.3 – L'entrée de l'application sous forme d'images

GMM AIRS - Video to HSV

Video path

Video path

Save the frames as

Save result path

Frame options

Width PX Height PX Extension: .bmp

Results

results

▶ Convert to HSV

FIGURE 4.4 – L'entrée de l'application sous forme vidéo

GMM AIRS - Live recording

IP cam address

xxx.xxx.x.xx:xxxx

Save the frames as

Save result path

Frame options

Width PX Height PX Extension: .bmp

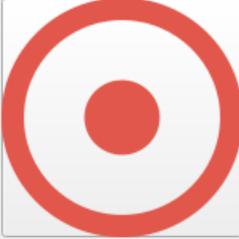


FIGURE 4.5 – L'entrée de l'application sous forme vidéo streaming

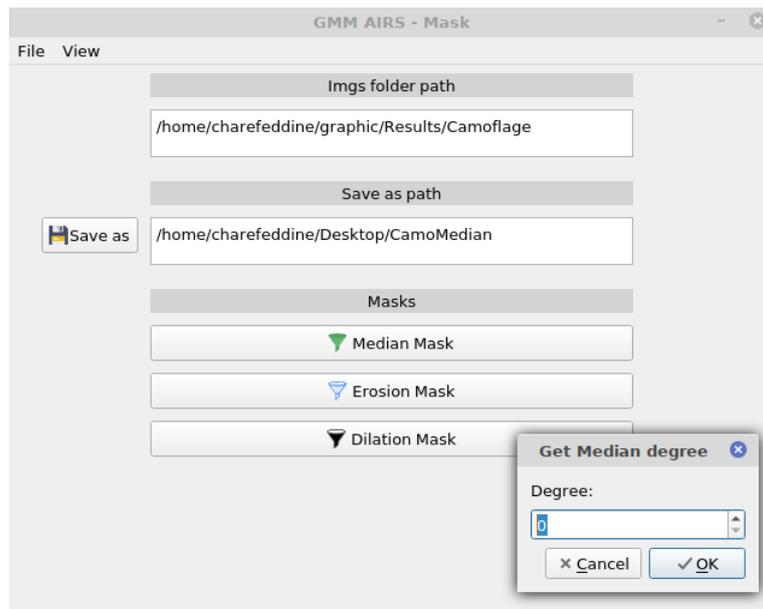


FIGURE 4.6 – Les différents types de masque (médian, Erosion, Dilatation)

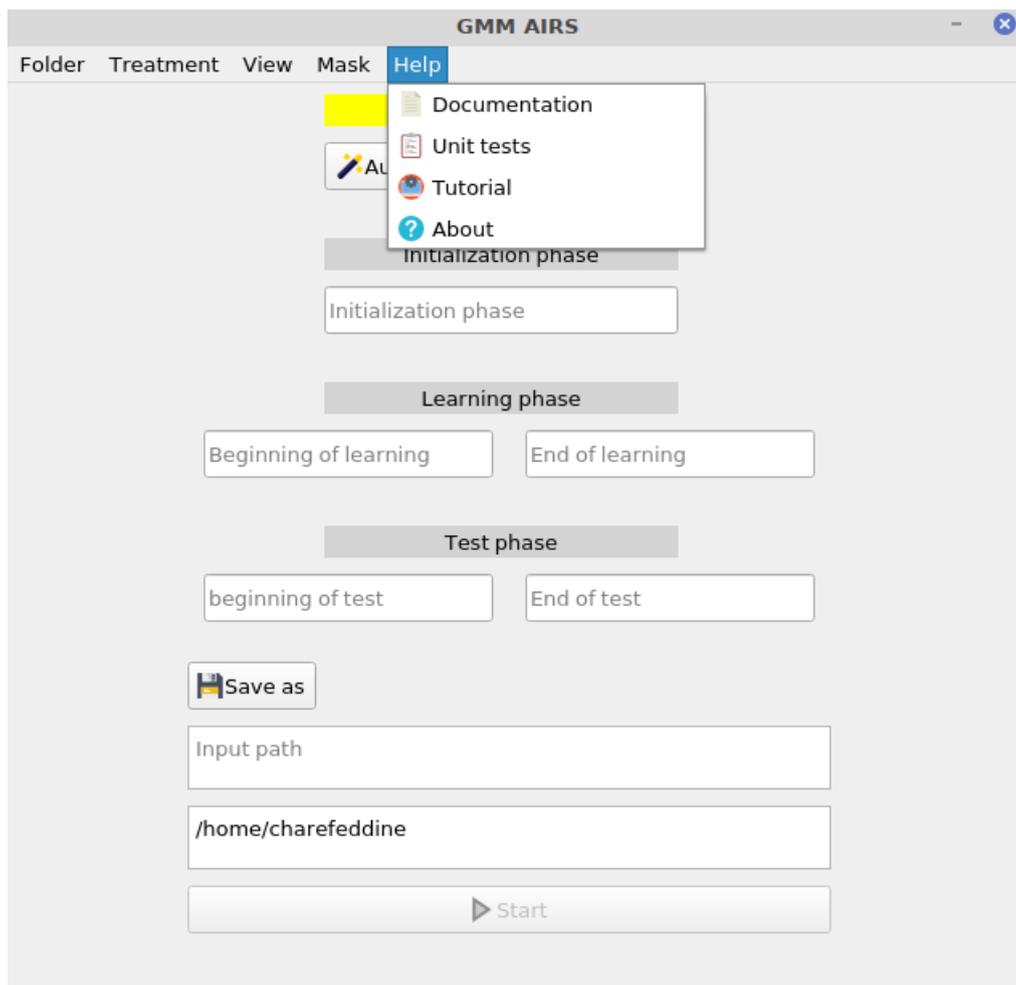


FIGURE 4.7 – Documentation

4.4 Tests et résultats

4.4.1 Choix des paramètres et définition des bases des données :

Cette section présente les résultats obtenus par notre méthode sur la base de données Wollflower [18] et sur une vidéo captée au niveau du département d'informatique.

Wollflower est une base de données publique qui contient sept vidéos (Moved Object (MO), Time of Day (TD), Light Switch (LS), Waving Trees (WT), Camouflage (Ca), Bootstrap (Bo), Foreground Aperture (FA)) avec une résolution de 160×120 . Les auteurs qui ont proposé cette base ont défini les séquences d'images utilisées pour l'apprentissage et les séquences d'images utilisées pour le test ainsi que l'image d'évaluation. Pour plus de détail voir le tableau 1.

Nom de vidéo	Nombre des frames	Résolution	Numéro de frame d'évaluation	Description de vidéo
MO	1745	160×120	985	L'arrière-plan contient un objet qui bouge
TD	5890	160×120	1850	Les changements d'éclairage progressifs
LS	2715	160×120	1865	Des changements soudains d'éclairage
WT	243	160×120	247	Arrière-plan dynamique
Ca	281	160×120	251	Les pixels 'avant plan englobent les pixels d'arrière-plan.
Bo	3055	160×120	299	Une absence de la phase d'entraînement des objets en mouvement dans certain environnement
Fa	2113	160×120	489	Les couleurs des objets en mouvement et de l'arrière-plan sont similaires

TABLE 4.1 – Description de la base de données Wallflower

Pour assurer la stabilité de notre système durant la phase de test, les valeurs du taux d'apprentissage α , la partie minimale des données correspondant à l'arrière-plan B, le nombre des

gaussiennes dans un modèle GMM b , taux d'hyper mutation, taux de clonage, ATS, taux de mutation sont fixés respectivement et après plusieurs tests empiriques à 0.001, 0.3, 5, 10, 2, 0.2, 0.1. Notons aussi que nous avons utilisé les mêmes paramètres pour toutes les vidéos.

4.4.2 Évaluation des performances

4.4.2.1 Les résultats qualitatifs :

Le tableau 2 présente quelques résultats obtenus par notre système sur les vidéos de la base de données Wollflower. Il comporte aussi une comparaison avec les vérités de terrain et avec d'autres méthodes citées dans l'état de l'art. Cependant les résultats qualitatifs ne permettent pas de tirer des conclusions profondes sur la performance du système. Pour cela, on a passé à des tests quantitatifs pour décrire de façon objective la robustesse de notre système selon un ensemble de critères.

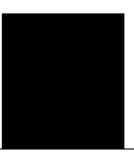
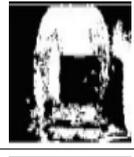
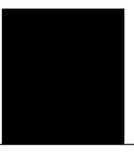
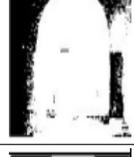
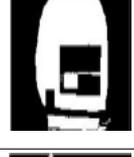
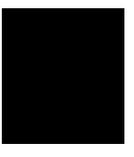
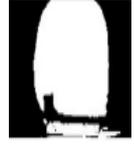
	MO	TD	LS	WT	Ca	Bo	FA
Image de test							
Vérité de terrain							
SG [37]							
MOG [46]							
KDE [6]							
SL-ICA [70]							
SL-INMF[71]							
SL-IRT [81]							
AIRS-GMM							

TABLE 4.2 – Les résultats qualitatifs obtenus sur la base des données Wollflower

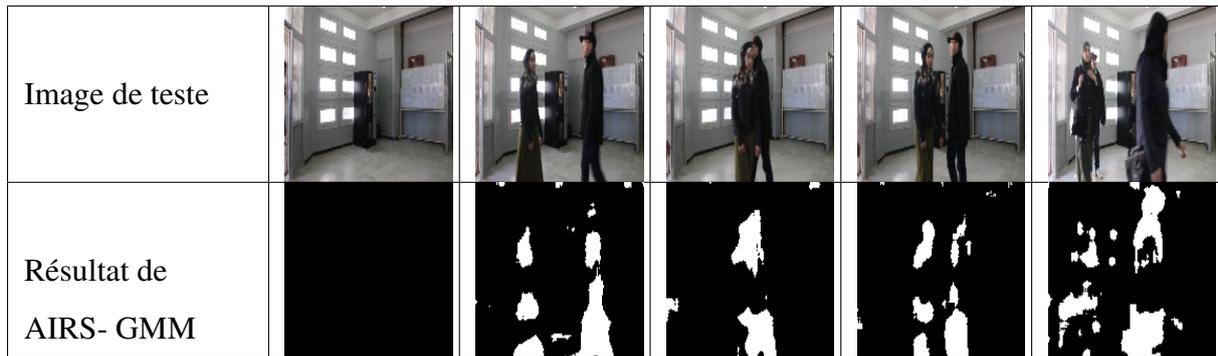


TABLE 4.3 – Les résultats qualitatifs sur une vidéo captée du monde réel

— **Les résultats quantitatifs :**

Pour évaluer la performance de notre système dans la détection des changements, nous avons calculé le nombre de faux positifs (FP), faux négatifs (FN), et l'erreur totale. De plus, nous avons choisi d'utiliser parmi les métriques de performance :

— Recall : qui représente le taux de détection des objets en mouvement.

$$\text{Recall (Re)} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

— Precision : qui montre le taux de discrimination des objets avant plan dans le système.

$$\text{Precision (Pr)} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

— F-measure : qui décrit la moyenne pondérée entre Recall et la precision.

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

— False negative (FN) : le résultat est négative (0), mais la vérité de terrain est positive (255)

— False positive (FP) : le résultat est positif (255), mais la vérité de terrain est négative (0)

— True positive (TP) : le résultat est positif (255) et la vérité de terrain est positive (255)

Tous ses critères sont calculés par rapport à la segmentation à la main des images. Nous avons comparé les résultats avec six méthodes de l'état de l'art (SG, MOG, KDE, SL-ICA, SL-INMF, SL-IRT).

	L'erreur	MO	TD	LS	WT	Ca	Bo	FA	L'erreur total
SG	FN	0	949	1857	3110	4101	2215	3464	21522
	FP	0	535	1512	357	2040	92	1290	
MOG	FN	0	1008	1633	1323	398	1874	2442	27053
	FP	0	20	14169	341	3098	217	530	
KDE	FN	0	1298	760	170	238	1755	2413	26450
	FP	0	125	14153	589	3392	933	624	
SL-ICA	FN	0	1199	1557	3372	3054	2560	2721	15308
	FP	0	0	210	148	43	16	428	
SL-INMF	FN	0	724	1593	3317	6626	1401	3412	19098
	FP	0	481	303	652	234	190	165	
SL-IRT	FN	0	1282	2822	4525	1491	1734	2438	18176
	FP	0	159	1512	7	114	2080	12	
AIRS - GMM	FN	0	1024	950	12	1	471	35	5669
	FP	0	1204	370	341	198	397	666	

TABLE 4.4 – Comparaison des résultats quantitatifs (FP, FN, l'erreur totale) avec des méthodes connues de soustraction de fond sur la base des données Wallflower

Algorithme	Critère de performance	MO	TD	LS	WT	Ca	Bo	FA
SG	Recall	1.000	0.949	0.545	0.835	0.761	0.884	0.807
	Precision	1.000	0.971	0.128	0.978	0.865	0.995	0.918
	F-measure	1.000	0.960	0.207	0.901	0.810	0.936	0.859
MOG	Recall	1,000	0,947	0,675	0,930	0,975	0,901	0,869
	Precision	1,000	0,999	0,193	0,981	0,835	0,987	0,968
	F-measure	1,000	0,972	0,301	0,955	0,900	0,942	0,916
KDE	Recall	1.000	0.932	0.849	0.991	0.985	0.904	0.870
	Precision	1.000	0.993	0.232	0.969	0.821	0.947	0.963
	F-measure	1.000	0.962	0.365	0.980	0.896	0.925	0.914

SL-ICA	Recall	1.000	0.938	0.918	0.823	0.841	0.867	0.855
	Precision	1.000	1.000	0.988	0.991	0.997	0.999	0.974
	F-measure	1.000	0.968	0.952	0.899	0.912	0.928	0.911
SL-INMF	Recall	1.000	0.961	0.916	0.821	0.651	0.926	0.821
	Precision	1.000	0.974	0.983	0.959	0.981	0.989	0.990
	F-measure	1.000	0.968	0.948	0.885	0.782	0.957	0.897
SL-IRT	Recall	1.000	0.933	0.850	0.764	0.922	0.899	0.993
	Precision	1.000	0.991	0.976	1.000	0.994	0.881	0.880
	F-measure	1.000	0.961	0.909	0.866	0.956	0.890	0.933
AIRS-GMM	Recall	1.000	0.211	0.681	0.998	0.999	0.812	0.932
	Precision	1.000	0.185	0.846	0.945	0.981	0.837	0.918
	F-measure	1.000	0.197	0.754	0.971	0.990	0.824	0.925

TABLE 4.5 – Comparaison des résultats quantitatifs (Recall, Precision, F-measure) avec des méthodes connues de soustraction de fond sur la base des données Wallflower.

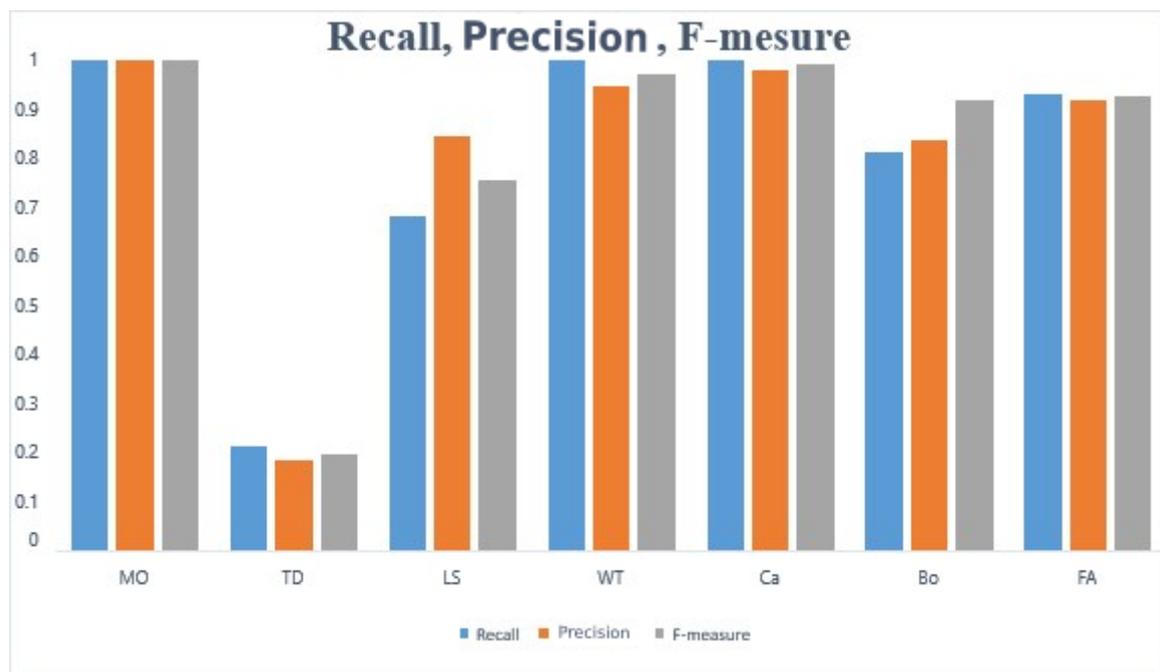


FIGURE 4.8 – Recall, Precision, F-measure de AIRS-GMM

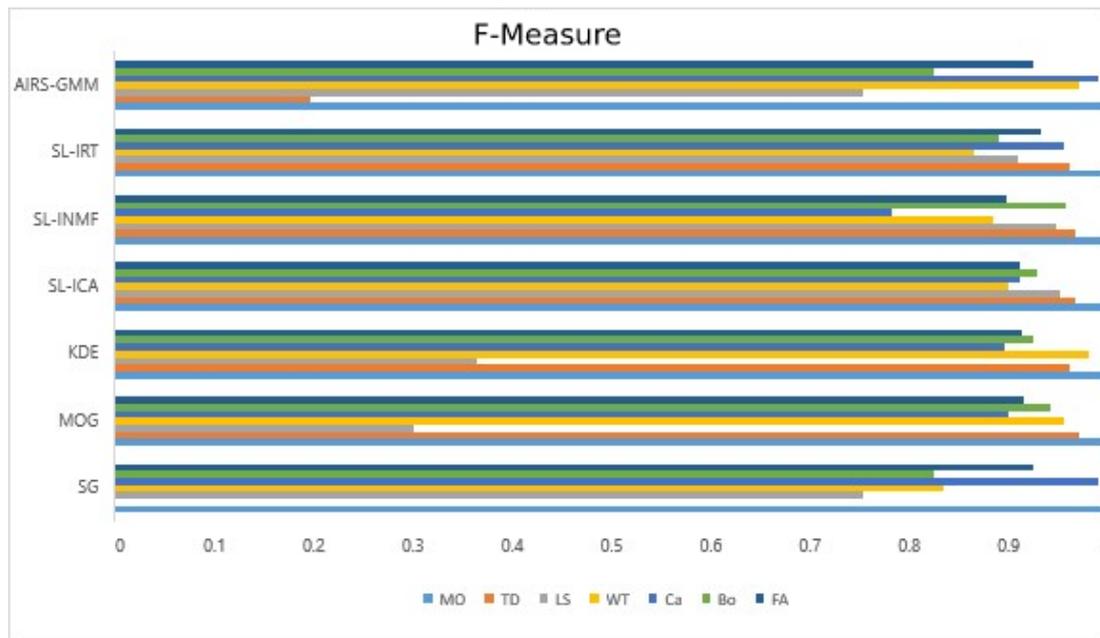


FIGURE 4.9 – Comparaison des résultats quantitatifs (*F-measure*) avec des méthodes connues de soustraction de fond sur la base des données Wallflower

4.4.3 Discussions des résultats

Notre méthode a obtenu de bons résultats dans les vidéos : Moved Object, Camouflage, Waving Trees, Foreground Aperture, Bootstrap et des résultats acceptables pour la vidéo : Light Switch.

Cependant, le système proposé a échoué dans la résolution des problèmes liés aux scènes qui possèdent un changement progressif d'éclairage (vidéo Time of Day), cela est dû à la nature du vecteur de caractéristique utilisé.

Le système peut obtenir des résultats encore meilleurs en ajoutant d'autres caractéristiques. En effet, nous avons utilisé la composante H de l'espace de couleur HSV comme caractéristique. Ce choix a été basé sur la puissance de cet espace par rapport à l'espace RGB vu qu'il permet de canaliser la lumière dans un seul composant (V). Malgré que les variations liées à la lumière sont diminuées, une seule caractéristique reste insuffisante pour la modélisation de l'arrière-plan et nous avons utilisé uniquement la composante H, puisque notre intérêt dans ce travail est de proposer une nouvelle méthode de soustraction de l'arrière-plan et non pas le choix du vecteur de caractéristique.

Les résultats observables montrent clairement que notre système a obtenu un bon taux de détection, puisque nous avons détecté tous les objets en mouvements avec uniquement quelques fausses détections dans la vidéo Time of Day.

4.5 Conclusion

Dans ce chapitre nous avons présenté les résultats de notre méthode sur la base de données publique Wollflower. Les résultats obtenus sur cette base de données montrent l'efficacité de notre système dans les scènes qui possèdent une petite variation dans l'arrière-plan. Notons aussi que cette méthode nous a permis de traiter les problèmes de la GMM dans les scènes où les variations des pixels arrière-plan sont très rapides comme dans les vidéos WT et LS.

Conclusion générale

Dans ce mémoire nous avons présenté une nouvelle approche pour la soustraction de l'arrière-plan basée sur un mécanisme dynamique pour la gestion du nombre de modèles dans le système. L'idée consiste à effectuer une combinaison entre l'approche GMM et l'algorithme bio-inspiré AIRS.

Initialement, le système démarre avec un seul modèle GMM dans la phase d'apprentissage. Ensuite, pour chaque pixel d'arrière-plan plusieurs modèles GMM sont créés par le processus de « **Identification de cellule mémoire et génération d'ARB** » de l'algorithme AIRS. Ces modèles sont filtrés en fonction de la concurrence des ressources et du processus de développement des cellules mémoires de l'algorithme AIRS. Ce mécanisme permet de ne sélectionner que les meilleurs modèles ou ils seront combinés pour créer un seul modèle candidat. Ce modèle peut être introduit comme un nouveau représentant de l'arrière-plan par le processus « **Introduction de cellule mémoire** ».

Les résultats obtenus sur la base des données publique Wollflower ont montré l'efficacité de notre approche dans les vidéos avec de petites variations de l'arrière-plan. Notons aussi que cette méthode nous a permis de traiter les problèmes de la GMM dans les scènes où le changement de luminosité est très rapide.

Comme futurs travaux et pour surmonter les inconvénients de ce système, nous allons consacrer notre étude sur le choix du vecteur de caractéristiques, et nous allons appliquer cette méthode sur d'autres bases des données.

Bibliographie

- [1] M. M. Azab, H. A. Shedeed, and A. S. Hussein. A new technique for background modeling and subtraction for motion detection in real-time videos. In *2010 IEEE International Conference on Image Processing*, pages 3453–3456, Sep. 2010.
- [2] P. St-Charles, G. Bilodeau, and R. Bergevin. Subsense : A universal change detection method with local adaptive sensitivity. *IEEE Transactions on Image Processing*, 24(1) :359–373, Jan 2015.
- [3] Xu Jian, Ding Xiao-qing, Wang Sheng-jin, and Wu You-shou. Background subtraction based on a combination of texture, color and intensity. In *2008 9th International Conference on Signal Processing*, pages 1400–1405. IEEE, 2008.
- [4] K. Wang, C. Gou, and F. Wang. m^4cd : A robust change detection method for intelligent visual surveillance. *IEEE Access*, 6 :15505–15520, 2018.
- [5] Gianni Allebosch, David Van Hamme, Francis Deboeverie, Peter Veelaert, and Wilfried Philips. C-EFIC : Color and edge based foreground background segmentation with interior classification. In *Communications in Computer and Information Science*, pages 433–454. Springer International Publishing, 2016.
- [6] Ahmed Elgammal, David Harwood, and Larry Davis. Non-parametric model for background subtraction. In *European conference on computer vision*, pages 751–767. Springer, 2000.
- [7] A Tavakkoli. Automatic video object plane extraction using non-parametric kernel density estimation. *Mathematical Methods in Computer Vision, University of Nevada, Reno, NV*, 2005.
- [8] Tatsuya Tanaka, Atsushi Shimada, Daisaku Arita, and Rin-ichiro Taniguchi. A fast algorithm for adaptive background model construction using parzen density estimation. In

- 2007 *IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 528–533. IEEE, 2007.
- [9] Sascha Cvetkovic, Peter Bakker, Johan Schirris, and Peter Hn De With. Background estimation and adaptation model with light-change removal for heavily down-sampled video surveillance signals. In *2006 International Conference on Image Processing*, pages 1829–1832. IEEE, 2006.
- [10] Codrut Ianasi, Vasile Gui, Corneliu I Toma, and Dan Pescaru. A fast algorithm for background tracking in video surveillance, using nonparametric kernel density estimation. *Facta universitatis-series : Electronics and Energetics*, 18(1) :127–144, 2005.
- [11] Yan-Fen Mao and Peng-Fei Shi. Multimodal background model with noise and shadow suppression for moving object detection. *Journal of Southeast University (English Edition)*, 20(4) :423–426, 2004.
- [12] Peng Tang, Lin Gao, and Zhifang Liu. Salient moving object detection using stochastic approach filtering. In *Fourth International Conference on Image and Graphics (ICIG 2007)*, pages 530–535. IEEE, 2007.
- [13] Vladimir Naumovich Vapnik. Vlamimirvapnik. “statistical learning theory”, vol. 1, 1998.
- [14] Horng-Horng Lin, Tyng-Luh Liu, and Jen-Hui Chuang. A probabilistic svm approach for background scene initialization. In *Proceedings. International Conference on Image Processing*, volume 3, pages 893–896. IEEE, 2002.
- [15] KA Ahmad, Z Saad, Noramalina Abdullah, Z Hussain, and MH Mohd Noor. Moving vehicle segmentation in a dynamic background using self-adaptive kalman background method. In *2011 IEEE 7th International Colloquium on Signal Processing and its Applications*, pages 439–442. IEEE, 2011.
- [16] Junxian Wang, George Bebis, Mircea Nicolescu, Monica Nicolescu, and Ronald Miller. Improving target detection by coupling it with tracking. *Machine Vision and Applications*, 20(4) :205–223, 2009.
- [17] Alireza Tavakkoli, Mircea Nicolescu, and George Bebis. Robust recursive learning for foreground region detection in videos with quasi-stationary backgrounds. In *18th International Conference on Pattern Recognition (ICPR’06)*, volume 1, pages 315–318. IEEE, 2006.

- [18] Kentaro Toyama, John Krumm, Barry Brumitt, and Brian Meyers. Wallflower : Principles and practice of background maintenance. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 255–261. IEEE, 1999.
- [19] Christof Ridder, Olaf Munkelt, and Harald Kirchner. Adaptive background estimation and foreground detection using kalman-filtering. In *Proceedings of International Conference on recent Advances in Mechatronics*, pages 193–199. Citeseer, 1995.
- [20] Dashan Gao and Jie Zhou. Adaptive background estimation for real-time traffic monitoring. In *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. O1TH8585)*, pages 330–333. IEEE, 2001.
- [21] Jing Zhong et al. Segmenting foreground objects from a dynamic textured background via a robust kalman filter. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 44–50. IEEE, 2003.
- [22] Fei Lei and Xiaoxia Zhao. Adaptive background estimation of underwater using kalman-filtering. In *2010 3rd International Congress on Image and Signal Processing*, volume 1, pages 64–67. IEEE, 2010.
- [23] Ismail Haritaoglu, David Harwood, and Larry S. Davis. W/sup 4 : real-time surveillance of people and their activities. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8) :809–830, 2000.
- [24] Kyungnam Kim, Thanarat H Chalidabhongse, David Harwood, and Larry Davis. Background modeling and subtraction by codebook construction. In *2004 International Conference on Image Processing, 2004. ICIP'04.*, volume 5, pages 3061–3064. IEEE, 2004.
- [25] Ruian Liu, Xin Zhou, Nailin Wang, and Mimi Zhang. Adaptive regulation of ccd camera in eye gaze tracking system. In *2009 2nd International Congress on Image and Signal Processing*, pages 1–4. IEEE, 2009.
- [26] Munir Shah, Jeremiah Deng, and Brendon Woodford. Enhanced codebook model for real-time background subtraction. In *International Conference on Neural Information Processing*, pages 449–458. Springer, 2011.
- [27] Yongbin Li, Feng Chen, Wenli Xu, and Youtian Du. Gaussian-based codebook model for video background subtraction. In *International Conference on Natural Computation*, pages 762–765. Springer, 2006.

- [28] S ITing, Shih-Chung Hsu, and CL Huang. Hybrid codebook model for foreground object segmentation and shadow/highlight removal. *Journal of Information Science and Engineering*, 30 :1965–1984, 2014.
- [29] Mikaël A Mousse, Eugène C Ezin, and Cina Motamed. Foreground-background segmentation based on codebook and edge detector. In *2014 Tenth International Conference on Signal-Image Technology and Internet-Based Systems*, pages 119–124. IEEE, 2014.
- [30] Robert T Collins, Alan J Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yan-ghai Tsin, David Tolliver, Nobuyoshi Enomoto, Osamu Hasegawa, Peter Burt, et al. A system for video surveillance and monitoring. *VSAM final report*, pages 1–68, 2000.
- [31] Manoranjan Paul, Shah ME Haque, and Subrata Chakraborty. Human detection in surveillance videos and its applications-a review. *EURASIP Journal on Advances in Signal Processing*, 2013(1) :176, 2013.
- [32] Weiming Hu, Tieniu Tan, Liang Wang, and Steve Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 34(3) :334–352, 2004.
- [33] Massimo Piccardi. Background subtraction techniques : a review. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, volume 4, pages 3099–3104. IEEE, 2004.
- [34] Yoshinari Kameda and Michihiko Minoh. A human motion estimation method using 3-successive video frames. In *International conference on virtual systems and multimedia*, pages 135–140, 1996.
- [35] Paolo Spagnolo, M Leo, Arcangelo Distanto, et al. Moving object segmentation by background subtraction and temporal analysis. *Image and Vision Computing*, 24(5) :411–423, 2006.
- [36] Davide A Migliore, Matteo Matteucci, and Matteo Naccari. A reevaluation of frame difference in fast and robust motion detection. In *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, pages 215–218. ACM, 2006.
- [37] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Pentland. Pfunder : Real-time tracking of the human body. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19 :780–785, 1997.

- [38] Dongpyo Hong and Woontack Woo. A background subtraction for a vision-based user interface. In *Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint*, volume 1, pages 263–267. IEEE, 2003.
- [39] A Homayoun Kamkar-Parsi, Robert Laganière, and Martin Bouchard. A multi-criteria model for robust foreground extraction. In *Proceedings of the third ACM international workshop on Video surveillance & sensor networks*, pages 67–70. ACM, 2005.
- [40] Mohamed F Abdelkader, Rama Chellappa, Qinfen Zheng, and Alex L Chan. Integrated motion detection and tracking for visual surveillance. In *Fourth IEEE International Conference on Computer Vision Systems (ICVS'06)*, pages 28–28. IEEE, 2006.
- [41] Tao Chen, Julian Morris, and Elaine Martin. Gaussian process regression for multivariate spectroscopic calibration. *Chemometrics and Intelligent Laboratory Systems*, 87(1) :59–71, 2007.
- [42] Sinno Jialin Pan, James T Kwok, and Qiang Yang. Transfer learning via dimensionality reduction. In *AAAI*, volume 8, pages 677–682, 2008.
- [43] Chinta Rambabu, Kiyoun Kim, and Woontack Woo. Fast and accurate extraction of moving object silhouette for personalized virtual reality studio@ home. *Journal of Real-Time Image Processing*, 4(4) :317, 2009.
- [44] Miguel López-Benítez and Fernando Casadevall. Versatile, accurate, and analytically tractable approximation for the gaussian q-function. *IEEE Transactions on Communications*, 59(4) :917–922, 2011.
- [45] Nir Friedman and Stuart Russell. Image segmentation in video sequences : A probabilistic approach. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pages 175–181. Morgan Kaufmann Publishers Inc., 1997.
- [46] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 2, pages 246–252. IEEE, 1999.
- [47] P Wayne Power and Johann A Schoonees. Understanding background mixture models for foreground segmentation. In *Proceedings image and vision computing New Zealand*, volume 2002, 2002.

- [48] Eric Hayman and Jan-Olof Eklundh. Statistical background subtraction for a mobile observer. In *null*, page 67. IEEE, 2003.
- [49] Pakorn KaewTraKulPong and Richard Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Video-based surveillance systems*, pages 135–144. Springer, 2002.
- [50] Theekapun Charoenpong, Ajaree Supasuteekul, and Chaiwat Nuthong. Adaptive background modeling from an image sequence by using k-means clustering. In *ECTI-CON2010 : The 2010 ECTI International Confernce on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pages 880–883. IEEE, 2010.
- [51] Fida El Baf, Thierry Bouwmans, and Bertrand Vachon. Fuzzy statistical modeling of dynamic backgrounds for moving object detection in infrared videos. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 60–65. IEEE, 2009.
- [52] Anastasis Doulamis, Ilias Kalisperakis, Christos Stentoumis, and Nicolaos Matsatsinis. Self adaptive background modeling for identifying persons’ falls. In *2010 Fifth International Workshop Semantic Media Adaptation and Personalization*, pages 57–63. IEEE, 2010.
- [53] Konrad Schindler and Hanzi Wang. Smooth foreground-background segmentation for video processing. In *Asian Conference on Computer Vision*, pages 581–590. Springer, 2006.
- [54] Brandyn White and Mubarak Shah. Automatically tuning background subtraction parameters using particle swarm optimization. In *2007 IEEE International Conference on Multimedia and Expo*, pages 1826–1829. IEEE, 2007.
- [55] Brahim Farou, Hamid Seridi, and Herman Akdag. Improved gaussian mixture model with background spotter for the extraction of moving objects. *Int. Arab J. Inf. Technol.*, 13(6A) :807–816, 2016.
- [56] Hansung Kim, Ryuuki Sakamoto, Itaru Kitahara, Tomoji Toriyama, and Kiyoshi Kogure. Background subtraction using generalised gaussian family model. *Electronics letters*, 44(3) :189–190, 2008.

- [57] M. S. Allili, N. Bouguila, and D. Ziou. Finite generalized gaussian mixture modeling and applications to image and video foreground segmentation. In *Fourth Canadian Conference on Computer and Robot Vision (CRV '07)*, pages 183–190, May 2007.
- [58] Nuria M Oliver, Barbara Rosario, and Alex P Pentland. A bayesian computer vision system for modeling human interactions. *IEEE transactions on pattern analysis and machine intelligence*, 22(8) :831–843, 2000.
- [59] Kamal Sehairi, Fatima Chouireb, and Jean Meunier. Comparative study of motion detection methods for video surveillance systems. *Journal of Electronic Imaging*, 26(2) :023025, 2017.
- [60] Zhifei Xu, Pengfei Shi, and Irene Yu-Hua Gu. An eigenbackground subtraction method using recursive error compensation. In *Pacific-Rim Conference on Multimedia*, pages 779–787. Springer, 2006.
- [61] Zhifei Xu, Irene Yu-Hua Gu, and Pengfei Shi. Recursive error-compensated dynamic eigenbackground learning and adaptive background subtraction in video. *Optical Engineering*, 47(5) :057001, 2008.
- [62] Charles-Henri Quivy and Itsuo Kumazawa. Background images generation based on the nelder-mead simplex algorithm using the eigenbackground model. In *International Conference Image Analysis and Recognition*, pages 21–29. Springer, 2011.
- [63] Yongmin Li. On incremental and robust subspace learning. *Pattern recognition*, 37(7) :1509–1518, 2004.
- [64] Danijel Skočaj and Aleš Leonardis. Incremental and robust learning of subspace representations. *Image and Vision Computing*, 26(1) :27–38, 2008.
- [65] Jian Zhang and Yueting Zhuang. Adaptive weight selection for incremental eigenbackground modeling. In *2007 IEEE International Conference on Multimedia and Expo*, pages 851–854. IEEE, 2007.
- [66] Xiaoyu Wu, Lei Yang, and Cheng Yang. Real-time foreground segmentation based on a fused background model. In *2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, volume 2, pages 585–588. IEEE, 2010.
- [67] Bohyung Han and Ramesh Jain. Real-time subspace-based background modeling using multi-channel data. In *International Symposium on Visual Computing*, pages 162–172. Springer, 2007.

- [68] Youdong Zhao, Haifeng Gong, Liang Lin, and Yunde Jia. Spatio-temporal patches for night background modeling by subspace learning. In *2008 19th International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008.
- [69] Y Dong and Guilherme N DeSouza. Adaptive learning of multi-subspace for foreground detection under illumination changes. *Computer Vision and Image Understanding*, 115(1) :31–49, 2011.
- [70] D. Tsai and S. Lai. Independent component analysis-based background subtraction for indoor surveillance. *IEEE Transactions on Image Processing*, 18(1) :158–167, Jan 2009.
- [71] S. S. Bucak and B. Günsel. Video content representation by incremental non-negative matrix factorization. In *2007 IEEE International Conference on Image Processing*, volume 2, pages II – 113–II – 116, Sep. 2007.
- [72] A. Watkins and L. Boggess. A new classifier based on resource limited artificial immune systems. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, volume 2, pages 1546–1551 vol.2, May 2002.
- [73] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri. Self-nonsel self discrimination in a computer. In *Proceedings of 1994 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 202–212, May 1994.
- [74] Steven A. Hofmeyr and Stephanie Forrest. Architecture for an artificial immune system. *Evolutionary Computation*, 8(4) :443–473, 2000.
- [75] Jungwon Kim and Peter J Bentley. Towards an artificial immune system for network intrusion detection : an investigation of dynamic clonal selection. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, volume 2, pages 1015–1020 vol.2, May 2002.
- [76] N. K. JERNE. Towards a network theory of the immune system. *Ann.Immunol.*, 125 :373–389, 1974.
- [77] L. N. De Castro and F. J. Von Zuben. Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 6(3) :239–251, June 2002.
- [78] Andrew Watkins, Jon Timmis, and Lois Boggess. Artificial immune recognition system (airs) : An immune-inspired supervised learning algorithm. *Genetic Programming and Evolvable Machines*, 5(3) :291–317, September 2004.

- [79] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1) :21–27, January 1967.
- [80] Anwaar-ul Haq, Iqbal Gondal, and M Murshed. Automated multi-sensor color video fusion for nighttime video surveillance. In *The IEEE symposium on Computers and Communications*, pages 529–534. IEEE, 2010.
- [81] Xi Li, Weiming Hu, Zhongfei Zhang, and Xiaoqin Zhang. Robust foreground segmentation based on two effective background models. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, pages 223–228. ACM, 2008.
- [82] <https://www.pythonforbeginners.com/learn-python/what-is-python/>, 21/04/2019
- [83] <https://www.makeuseof.com/tag/install-pip-for-python/>, 21/04/2019
- [84] <https://opencv.org/about/>, 21/04/2019
- [85] <https://www.learnopencv.com/install-opencv-3-and-dlib-on-windows-python-only/>, 21/04/2019
- [86] <https://www.tutorialspoint.com/pyqt/>, 21/04/2019