

MA.004 - 336 - 1

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

Université Saad Dahlab Blida



Faculté des sciences

Département d'informatique

Mémoire Présenté par :

BELBACHIR Salima KOUSSA Narimene

En vue d'obtenir le diplôme de master

Domaine : Mathématique et informatique

Filière : Informatique
Spécialité : Informatique
Option : Génie de système d'informatique

Thème

Conception et réalisation d'un système de détection, suivi et comptage de personnes dans une vidéo numérique

Soutenu le : Juin 2016

Promoteur : M.KEMECH Hichem

Encadreur : DR.MEZIANE Abed-lakrim

le Président du Jury : *O E R R A R H a c e m e*

Promotion
2015 / 2016

MA-004-336-1

DEDICACES

Je dédie ce modeste travail

À ceux qui me sont les plus chers au monde, qui n'ont jamais cessé de m'encourager, de prier pour moi, et qui m'ont toujours entouré de toute leur affection et leur amour, Que Dieu vous protège :

Mon cher et tendre mari Imad, mes chers parents FAYZA et CHAKIB, mon adorable sœur AMINA et mon petit frère RYAD ;

À ma chère belle famille : Baya, Hocine, Kahina, Anissa, Nacera, Mohamed et à mes anges : Maya, Imène et Adem ;

À toute la famille BELBACHIR et MAHMOUDI.

À mon chère binôme Narimène, ainsi qu'à toute sa famille.

Et tant d'autres qui, je l'espère, ne m'en voudront pas de ne pas cités leurs noms.

DEDICACES

*Avec un énorme plaisir, un cœur ouvert et une immense joie, que
Je dédie mon travail a mes très chers, respectueuse et magnifiques
parents NADIA ET IBRAHIM qui m'ont soutenus tout au long de
ma vie ainsi a mes sœurs MARWA, CHAIMA et mes frères
RIADH, ISSAM, ABD-EL-AZIZ, mes chères amies RACHIDA,
NACHIDA, MOUNA*

*A mes tantes LINDA et BEATRICES pour leurs précieux
encouragements, leur amour, leur soutien*

*Et en particulier à mon binôme SALIMA pour sa douceur et sa
gentillesse, j'espère que la vie lui réserve le meilleur*

*A toutes personnes qui m'ont encouragé ou aidé au long
de mes études*

ملخص:

نقدم في هذه البحث طريقة لكشف ورصد وحساب عدد الناس داخل فيديو رقمي، مع الافتراض بأن الكاميرا المستعملة ثابتة وأن البيئة تعتبر داخلية. استخدمنا اولاً طريقة طرح الخلفية من أجل الكشف عن الناس المتحركة. و من ثم نتبعهم ونحدد اتجاههم، ومن هناك نستطيع أن نحسب عددهم مع تحديد عدد القادمين والمغادرين. أظهرت نتائج التجارب أن الطريقة المقترحة قوية وتوفر معدل دقة جيد.

المفاتيح: الكشف عن وجود اشخاص، تتبع الاشخاص، طرح الخلفية، المعالجة المورفولوجية الرياضية، OpenCv، CvTracks، CvBlobs ، MOG2 .

RESUME

Nous présentons dans ce mémoire une méthode de détection, suivi et comptage de personnes dans une vidéo numérique, entrants ou sortants d'une région d'intérêt, en posant l'hypothèse que notre caméra est fixe et que l'environnement est en intérieur. Nous utilisons tout d'abord la soustraction d'arrière-plan pour détecter les personnes en mouvement, ensuite nous les suivons et déterminons leur direction, et de là nous pouvons compter leurs nombre en spécifiant celui des entrants et des sortants. Les résultats tirés des expériences réalisées ont montré que la méthode proposée est robuste et fournit un bon taux de précision.

Mots-clés: Détection d'objets, suivi d'objets, soustraction d'arrière-plan, traitement morphologiques mathématiques, OpenCv, CvBlobs, CvTracks, MOG2.

ABSTRACT

In this paper, we present a human detection, tracking and counting system in a digital video, entering or exiting in a region of interest. By assuming that our camera is fixed and that the environment is indoors. We first use the background subtraction to detect people moving, then we track them and determine their direction, and from there we can count their number specifying the number of incoming and outgoing. Results from the experiments showed that the proposed method is robust and provides good accuracy rate.

Key-words: Objects detection, objects tracks, Background Subtraction, Mathematical Morphological Processes, OpenCv, CvBlobs, CvTracks, MOG2.

Sommaire

INTRODUCTION GENERALE	1
<u>CHAPITRE I : IMAGE ET VIDEO NUMERIQUE</u>	
I.1 INTRODUCTION	4
I.2 NOTIONS DE BASE SUR L'IMAGE NUMERIQUE	4
I.2.1 L'image numérique	4
I.2.2 Le pixel	5
I.2.3 La couleur et la profondeur de couleur	5
I.2.4 L'image matricielle et l'image vectorielle	7
I.2.5 La définition d'une image numérique	7
I.2.6 La résolution d'une image numérique	8
I.2.7 Formats d'image numérique	8
I.3 NOTIONS DE BASE SUR LA VIDEO NUMERIQUE	10
I.3.1 Définition	10
I.3.2 Vidéo analogique et vidéo numérique	10
I.3.3 Numérisation	11
I.3.4 Nombre d'images par seconde et résolution	12
I.3.5 Débit d'affichage d'une vidéo numérique	13
I.3.6 Compression	13
I.3.7 Les formats de vidéo numérique	16
I.4 CONCLUSION	16
<u>CHAPITRE II : GENERALITES SUR LE COMPTAGE DE PERSONNES</u>	
II.1 INTRODUCTION	17
II.2 LES AVANTAGE D'UN SYSTEME DE COMPTAGE DE PERSONNES	17
II.3 ANALYSE COMPARATIVE DES METHODES DE COMPTAGE	19
II.3.1 Capteurs directionnels passifs thermiques	19
II.3.2 Capteurs directionnels à infrarouge actifs	20
II.3.3 Rayons lumineux	20
II.3.4 Les tapis compteur « Tread mats »	21
II.3.5 Les tourniquets (compteurs mécaniques)	21
II.3.6 Technologies basées sur les caméras	22
II.3.7 Résumé comparatif	23
II.4 CONCLUSION.....	24

Sommaire

CHAPITRE III: PRINCIPAUX COMPOSANTS D'UN SYSTEME DE COMPTAGE DE PERSONNES

III.1 INTRODUCTION	25
III. 2 DETECTION DE PERSONNES	25
III.2.1 Introduction	25
III.2.2 Détection des éléments d'intérêt par soustraction de fond	26
III. 3 SUIVI DE PERSONNES	32
III.3.1 Introduction	32
III.3.2 Représentation de la forme d'une cible	33
III.3.3 Technique de suivi de cibles	34
III. 4 COMPTAGE DE PERSONNES	37
III.4.1 Comptage avec deux lignes virtuelles	37
III.4.2 Comptage avec une ligne virtuelle	37
III.5 CONCLUSION	39

CHAPITRE IV : MISE EN ŒUVRE DU SYSTEME DE COMPTAGE

IV.1 INTRODUCTION	40
IV.2 FONCTIONNEMENT DE NOTRE SYSTEME DE DETECTION, SUIVI ET COMPTAGE DE PERSONNES	40
IV.2.1 Processus de détection de personnes	41
IV. 2.1.1 Capture et traitement	42
IV. 2.1.2 Soustraction de fond BS.....	43
IV. 2.1.3 Génération de blobs	46
IV.2.2 Processus de suivi de personnes.....	52
IV.2.3 Processus de comptage de personnes.....	53
IV. 2.3.1 Comptage de personnes entrantes et sortantes de l'endroit.....	53
surveillé	
IV. 2.3.2 Comptage de personnes présentes dans la scène.....	55
IV. 3 CONCLUSION	55

Sommaire

CHAPITRE V : REALISATION ET EXPERIMENTATION

V.1 INTRODUCTION	56
V.2 ENVIRONNEMENT DE TRAVAIL	56
V.2.1 Matériels utilisés	56
V.2.2 Logiciels utilisés	57
V.3 DESCRIPTION DE L'APPLICATION	60
V.3.1 Système de détection de personnes	60
V.3.1.1 Capture et traitement	60
V.3.1.2 Soustraction de fond BS	61
V.3.1.3 Génération de blobs	64
V.3.2 Système de suivi de personnes	65
V.3.3 Système de comptage de personnes	68
V.4 ÉVALUATION DES PERFORMANCES	70
V.4.1 Notations et définitions standards	70
V.4.2 Test sur la résolution de la vidéo	71
V.4.3 Test sur la position des lignes de comptage	73
V.4.4 Test sur l'épaisseur des lignes de comptage	74
V.4.5 Comparaison entre des algorithmes de soustraction de fond	76
V.4.6 Comparaison entre CvTracks et CvBlobs	77
V.4.7 Test du programme sur un enregistrement en temps réel.....	78
V.5 Conclusion	79
CONCLUSION GENERALE	80
BIBLIOGRAPHIE	82
ANNEXE	

| *Table des figures*

Fig I.1 Pixels d'une image numérique	5
Fig I.2 Exemple d'image binaire 1 bit	6
Fig I.3 Exemple d'image en niveaux de gris 8bits	6
Fig I.4 Couleurs d'une image numérique	6
Fig I.5 Exemple d'image matricielle et vectorielle	7
Fig I.6 Exemples des résolutions d'une image.....	8
Fig I.7 Signaux vidéo	11
Fig I.8 Bruit dans les signaux vidéo	11
Fig I.9 Formats de vidéo	15
Fig II.1 Capteur directionnel DS-10.....	20
Fig II.2 Tapis de comptage de personnes	21
Fig II.3 Les tourniquets	22
Fig II.4 Caméra de surveillance	23
Fig III.1 Schéma de détection par soustraction de font	27
Fig III.2 Caractérisation de couleur d'un pixel (M) de l'image a l'aide de distributions gaussienne dans l'espace de représentation de couleur RBG	29
Fig III.3 Segmentation d'éléments de premier-plan	30
Fig III.4 Exemples de représentation de la forme d'une cible	34
Fig III.5 Etape de filtrage de Kalman	35
Fig III.6 Exemple de suivi de contours des objets	37
Fig III.7 Exemple de comptage avec deux lignes virtuelles	38
Fig III.8 Exemple de Comptage avec une seule ligne virtuelle	39
Fig IV.1 Approche de détection de personnes	41
Fig IV.2 Exemple d'application d'un filtre médian 3 x 3	43
Fig IV.3 Processus de soustraction de fon	45
Fig IV.4 Exemple d'application du MOG et MOG2	46
Fig IV. 5 La dilatation d'une image binaire	47
Fig IV.6 Elément structurant carrée 3x3	48

| *Table des figures*

Fig IV. 7 La dilatation d'une image binaire avec un élément structurant carré de 3x3 pixels	48
Fig IV.8 L'érosion d'une image binaire	49
Fig IV.9 L'érosion d'une image binaire avec un élément structurant carré de 3x3	49
Fig IV.10 Exemple d'étiquetage de blobs	50
Fig IV.11 Forme humaine perçue par une caméra installée en hauteur	51
Fig IV.12 Résultat de suivi d'une personne	52
Fig IV.13 Diagramme d'activité de l'algorithme de comptage	52
Fig V.1 Caméra utilisée	56
Fig V.2 Principaux modules d'OpenCV	58
Fig V.3 Génération du masque d'avant-plan avec MOG2	64
Fig V.4 Inconvénient du Label de cvBlobs	66
Fig V.5 Exemple d'utilisation de la fonction « cvRenderTracks »	67
Fig V.6 Résultat de suivi d'une personne	68
Fig V.7 Paramètres des lignes de comptage.....	68
Fig V.8 Fenêtre d'affichage de l'application de comptage	69
Fig V.9 Sauvegarde des résultats de comptage dans un fichier Excel	69
Fig V.10 Résultat et temps d'exécution par rapport à la variation de la résolution	71
Fig V.11 Résultat de précision par rapport à la variation de la résolution	72
Fig V.12 Résultat de précision par rapport à la variation de la position des lignes selon l'axe Y.....	73
Fig V.13 Résultat de précision par rapport à la variation de l'épaisseur	74
Fig V.14 Exemple de la nécessité de réglage de l'épaisseur.....	75
Fig V.15 Résultat de temps d'exécution par rapport aux différents algorithmes de détection.....	76
Fig V.16 Résultat de précision par rapport aux différents algorithmes de détection.....	76
Fig V.17 Résultat de temps d'exécution par rapport aux différents algorithmes de suivi....	77
Fig V.18 Résultat de précision par rapport aux différents algorithmes de suivi	77
Fig V.19 Résultat d'un comptage en temps réel	78

| *Liste des Tableaux*

Tab I.1 Les principaux formats d'image	10
Tab I.2 Exemples de codes vidéo avec domaines d'application	14
Tab I.3 Les principaux formats vidéo	16
Tab II.1 Comparaison de quelques méthodes de comptage de personnes	24
Tab V.1 Définitions standards : VP, VN, FN, FP	70

INTRODUCTION GENERALE

INTRODUCTION GENERALE

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

L'image est aujourd'hui un objet des plus importants dans notre société, c'est un média d'information (journaux papiers et télévisuelles, internet), publicitaire, artistique (cinéma, photographie,...) et social (Facebook, Picasa,...). Beaucoup de travaux visent donc à en améliorer et en faciliter l'accessibilité. Que ce soit au niveau de son acquisition (appareil photographique, camera numérique, webCam), de sa capacité de stockage et d'échange (formats, compression, ...) ou de son édition (par des logiciels tels que Photoshop ou Gimp).

En effet, la détection visuelle est la première étape vers des applications plus évoluées, à savoir, des applications en vidéosurveillance, biométrique, robotique, commande d'interface homme-machine, photographie, indexation d'images et de vidéos, recherche d'images par le contenu, etc. Elle permet également de faciliter l'automatisation complète d'autres processus tel que le comptage de personnes.

Le monde actuel, aussi bien pour des décisions économiques, commerciales, marketing, d'exploitation et de sécurité, exige des prises de décisions basées sur des données objectives et non discutables, tel que le nombre de personnes visitant un endroit donné, pendant une période bien précise.

La prise en compte des flux de personnes est un indicateur de valorisation très important, car il permet aux responsables d'exploitation et à leurs collaborateurs de mieux diriger les sites dont ils ont la responsabilité, tout en améliorant parallèlement la qualité du service rendu aux clients ou aux visiteurs. Parmi les domaines d'application, on peut citer : les aéroports, les centres commerciaux, les moyens de transport (métro, tramway...), les foires d'exposition...

Le comptage des personnes est une application importante dans divers domaines allant du marketing à la vidéosurveillance. Lorsque de nombreuses personnes traversent le hall d'une gare, d'un bâtiment ou d'un centre commercial, leur nombre représente une information pertinente pour les décideurs chargés d'assurer la sécurité publique et d'élaborer des stratégies commerciales. Ainsi, un certain nombre de personnes dans un contexte particulier est susceptible de refléter une situation anormale et potentiellement dangereuse. De plus, la possibilité de connaître le nombre d'individus dans une zone commerciale permet d'obtenir

INTRODUCTION GENERALE

de précieuses informations pour les gérants d'un magasin afin d'évaluer l'attractivité des offres promotionnelles qu'ils proposent.

Le centre de recherche sur l'information scientifique et technique CERIST a toujours eu pour mission, la recherche, la création, la mise en place et le développement des systèmes nationaux d'information scientifique et technique. Le travail qu'on nous a proposé d'accomplir est la conception et la réalisation d'un système de détection, suivi et comptage de personnes dans une vidéo numérique.

Beaucoup d'entreprises proposent un système de comptage sur la base de la caméra vidéo. Leur système est très précis et fiable, mais ces systèmes sont également très chers. L'objectif de l'ensemble de ce projet est de réaliser un prototype peu coûteux pour le comptage de personnes.

Eventuellement, avant de tenter de suivre et de compter les personnes dans une vidéo numérique, il est indispensable de les détecter en premier. La détection de personnes dans une vidéo revient à détecter des caractéristiques physiques spécifiques aux humains, tel que le visage, une partie du corps humain ou le corps en entier. Le choix d'une de ses caractéristiques dépend principalement de la **position** de la caméra utilisée.

En effet, les systèmes de comptage actuels n'utilisent pas le visage comme moyen de détection de personnes, cela revient au fait qu'il n'est pas souvent possible de filmer une personne sous l'angle le plus pertinent (celui qui permet la capture de la zone du visage portant le plus d'informations à savoir la zone autour des yeux) , vu que la majorité des caméras de surveillance sont installées soit dans le plafond, soit sur un mur à une hauteur allant de 1,50 à 4 mètres au dessus du sol. Cet emplacement est compréhensible, car ça permet une meilleure protection des caméras contre le vol ou la détérioration, et surtout, cet emplacement leurs permet de couvrir le maximum de zones que possible tout en réduisant significativement les cas d'occlusion. C'est pour cette raison que la plupart des méthodes actuelles de comptage utilisent la détection du corps ou une partie du corps humain comme moyen de détection de personnes.

C'est dans cette envergure que nous proposons une méthode de comptage de personne basée sur la détection de la forme humaine aperçu par une caméra installée en hauteur (ou sur le plafond).

Organisation du mémoire

Ce mémoire est organisé en cinq chapitres :

INTRODUCTION GENERALE

- Le premier chapitre «**IMAGE ET VIDEO NUMERIQUE**» : Ce chapitre présente les notions essentielles pour la compréhension de certains concepts basiques des images et des vidéos numériques, tel que : résolution, compression et formats.
- Le deuxième chapitre «**GENERALITES SUR LE COMPTAGE DE PERSONNES**» : Ce chapitre cite les multiples avantages qu'un système de comptage de personnes peut nous offrir, ensuite introduit quelques technologies et méthodes utilisées actuellement dans le domaine du comptage de personnes tels que les rayons infrarouges et les systèmes à base de caméra, et enfin explique les raisons de notre choix, qui est le comptage de personne en utilisant une caméra vidéo.
- Le troisième chapitre «**PRINCIPAUX COMPOSANTS D'UN SYSTEME DE COMPTAGE DE PERSONNES**» : Ce chapitre présente l'état de l'art des principaux composants que doit avoir un système de comptage de personnes dans une vidéo numérique, à savoir un sous-système de détection, un sous-système de suivi et un sous-système de comptage.
- Le quatrième chapitre «**MISE EN ŒUVRE DU SYSTEME DE COMPTAGE**» : Ce chapitre présente la méthodologie et la logique suivies tout au long de notre travail pour aboutir à notre objectif.
- Le cinquième chapitre «**REALISATION ET EXPERIMENTATION**» : Ce chapitre fournit une vision détaillée de l'implémentation de notre système de détection, suivi et comptage de personnes, puis analyse les résultats obtenus suite aux tests réalisés afin de mesurer son exactitude et conclure sur les situations dans lesquelles ce système est plus fiable.

CHAPITRE
I

**IMAGE ET VIDEO
NUMERIQUE**

CHAPITRE I : IMAGE ET VIDEO NUMERIQUE

I.1 INTRODUCTION

Avant d'aller plus loin dans notre travail, il est important de maîtriser certains concepts basiques sur les images et les vidéos numériques.

Concernant les images numériques, nous allons expliquer quelques notions de base essentielles, comme le pixel, la résolution d'image, ou encore les formats d'images les plus utilisés. Nous passerons aussi par les différents aspects concernant la technologie de la vidéo numérique, à savoir, la résolution, nombre d'images par seconde, la compression et les principaux formats de vidéo numérique.

Ces notions sont importantes et nécessaires pour une bonne manipulation des fichiers vidéo et une maîtrise des techniques qu'on va utiliser dans notre projet de comptage.

I.2 NOTIONS DE BASE SUR L'IMAGE NUMERIQUE

Nous expliquons dans cette partie les notions de base essentielles concernant les images numériques.

I.2.1 L'image numérique

On désigne sous le nom d'image numérique, toute image (dessin, icône, photographie...) pouvant être stockée sur un support informatique (ordinateur, clé USB, téléphone portable, iPod...) [W01].

Une fois sur support informatique, l'image numérique est alors ouverte à tout type de retouche (redimensionnement, suppression ou ajout d'éléments, travail de la couleur, ...), c'est ce qui fait toute la différence avec les images sur papier où on n'a pas cette possibilité de modification.

I.2.2 Le pixel

Une image numérique est une image composée entièrement de petits points élémentaires, que l'on appelle « pixels » (Une unité de surface). En agrandissant fortement une image, on distingue très bien les pixels, comme le montre la **Figure I.1**.

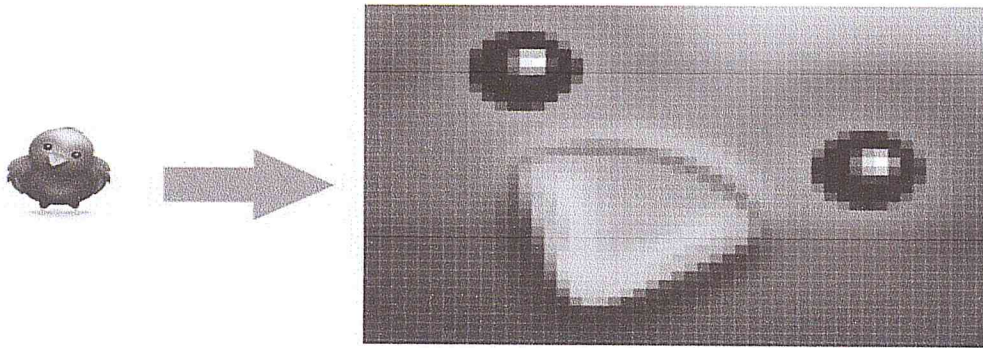


Fig I.1 Pixels d'une image numérique

I.2.3 La couleur et la profondeur de couleur

Il existe plusieurs modes de codage informatique des couleurs, le plus utilisé pour le maniement des images est l'espace colorimétrique Rouge, Vert, Bleu (RVB ou RGB). Cet espace est basé sur une synthèse additive des couleurs, c'est-à-dire que le mélange des trois composantes R, V, et B à leur valeur maximum donne du blanc, à l'instar de la lumière. Chaque pixel possède donc une couleur, qui est composée d'une part variable des trois couleurs de base : le rouge, le vert et le bleu.

Chaque couleur RVB est exprimée sur l'écran en un pourcentage de rouge, de vert et de bleu. Un pixel totalement rouge sur une image sera donc composé de 100% de rouge, 0% de vert et 0% de bleu [W02].

La profondeur de couleur est définie par le nombre de bits utilisés pour représenter chaque pixel. Plus la profondeur de bit est élevée, plus grand sera le nombre de teintes (niveaux de gris ou couleur) représenté :

24 bits $\Rightarrow 2^{24} = 16,7$ millions de couleurs

8 bits $\Rightarrow 2^8 = 256$ couleurs

1 bit $\Rightarrow 2$ couleurs (noir et blanc)

Les images numériques peuvent être produites en noir et blanc (deux couleurs), niveaux de gris ou couleur.

Une image binaire (deux couleurs) est représentée par des pixels de 1 bit chacun, pouvant représenter deux teintes (le noir et le blanc), en utilisant la valeur 0 pour le noir et 1 pour le blanc.

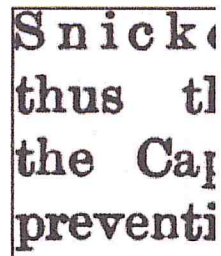


Fig I.2 Exemple d'image binaire 1 bit

Une image en niveaux de gris est composée de pixels à une profondeur allant de 2 à 8 bits. On utilise généralement une profondeur de 8 bits pour ce genre d'images, ça sera donc 256 niveaux de gris, de 0, le noir à 255, le blanc. La Figure I.3 montre un sous-tableau de 5×5 pixels extrait d'une image en niveaux de gris. On peut voir à la fois les valeurs qui composent le tableau et les niveaux de gris qui permettent d'afficher l'image à l'écran.

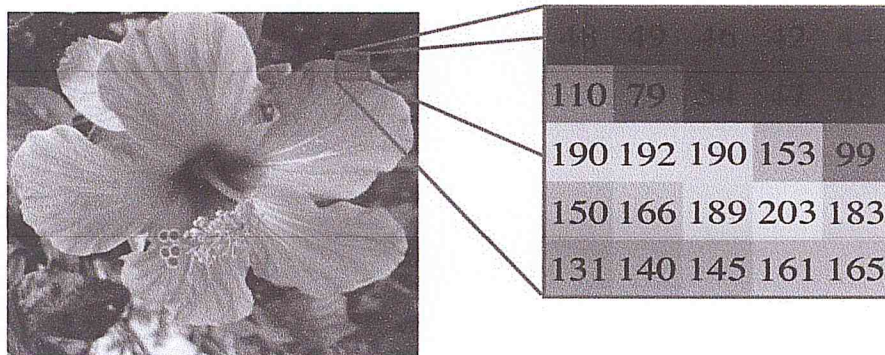


Fig I.3 Exemple d'image en niveaux de gris 8bits

Une image couleur est typiquement représentée par une profondeur de couleur variant de 8 à 24 bits ou plus. Dans une image 24 bits (le cas le plus général), les bits sont divisés en 3 groupes (3 octets) : 8 bits pour le rouge, 8 bits pour le vert et 8 bits pour le bleu. Chaque octet représentant la valeur d'une composante couleur par un entier de 0 à 255.

R	V	B	Couleur
0	0	0	noir
0	0	1	nuance de noir
255	0	0	rouge
0	255	0	vert
0	0	255	bleu
128	128	128	gris
255	255	255	blanc

Fig I.4 Couleurs d'une image numérique

I.2.4 L'image matricielle et l'image vectorielle

Dans le monde de l'imagerie numérique, il existe deux grands types d'images: les images **matricielles** (ou bitmap) et les images **vectorielles**. La différence entre le matriciel et le vectoriel réside dans la représentation des données: le matriciel utilise la matrice des points tandis que le vectoriel se servira de formes géométriques simples individuelles comme des lignes droites, des ellipses, des arcs de cercles, des triangles ou des courbes, chacune définie par des attributs de forme, de position ou de couleur [W01].

Cette différence est à l'origine de la qualité du redimensionnement, de telle sorte que lorsqu'on agrandit une image matricielle, elle perd inévitablement en qualité: les pixels seront visibles et les bords "en escaliers". De même, lorsqu'on la rétrécit, elle devient beaucoup moins nette.

Par contre, ce problème ne se pose pas avec les images vectorielles, qu'on l'agrandit ou qu'on la rétrécit, elle gardera exactement la même qualité et on ne verra pas apparaître des pixels.

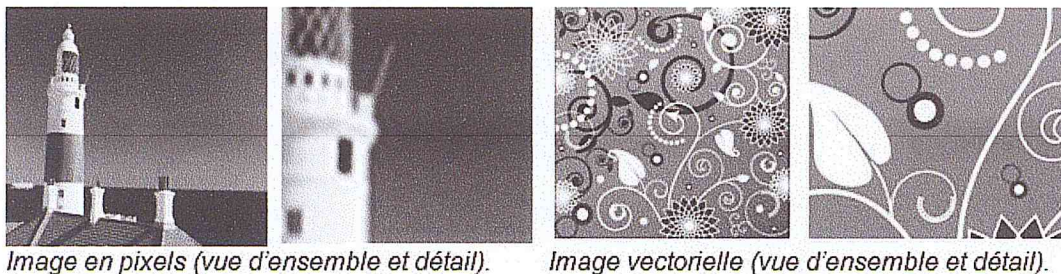


Fig I.5 Exemple d'image matricielle et vectorielle

Néanmoins, il est très difficile d'obtenir une image photo-réaliste avec du vectoriel car il faudrait superposer énormément de couches de formes géométriques et d'aplats de couleurs pour par exemple réaliser un dégradé.

En conséquence, le vectoriel est recommandé pour les polices, les logos ou les images destinées au web et susceptibles d'être agrandies. Par contre, on utilise le matricielle pour tout autre type d'image (photos, dessins complexes, tons continus).

I.2.5 La définition d'une image numérique

La définition d'une image numérique correspond simplement au nombre de pixels qui la composent. On peut l'exprimer en nombre de pixels en largeur et hauteur, ou en nombre de pixels global dans une image. Par exemple, pour l'image qui sort d'un appareil photo

numérique, sa définition peut être donnée sous la forme 3000x2000 pixels ou alors 6 mégapixels [W02].

I.2.6 La résolution d'une image numérique

La résolution d'une image est définie par un nombre de pixels par unité de longueur qui est en général le pouce « Inch » (1 pouce = 2,54 cm). Plus ce nombre est élevé, plus la qualité et la netteté de l'image est bonne [W03]. Toutefois, augmenter la résolution peut entraîner des temps de visualisation et d'impression plus longs, et conduire à une taille trop importante de l'image.

La résolution d'une image s'exprime en **DPI** (*Dote Per Inch* = point par pouce) pour une imprimante, ou en **PPP** (pixel par pouce) pour un fichier image.

Le nombre de pixels présents dans une surface donnée de l'image peut être par exemple en bonne résolution (300 PPP, pour les photos destinées à l'impression) ou basse résolution (72 PPP, pour les photos destinées à l'écran).

S'il faut agrandir une image avec une résolution trop basse, on risque d'obtenir une image pixellisée (voir la **Figure I.6** : les pixels deviennent apparents).



Fig I.6 Exemples des résolutions d'une image

Selon le support de diffusion de l'image, la résolution à choisir est différente :

- Pour l'Internet : 72 dpi, cette valeur correspond à la résolution d'un écran standard.
- Pour l'impression à usage personnel : 150 dpi, permet d'avoir un rendu correct de certains composants complexes d'une image.
- Pour l'impression professionnelle (journaux, magazines) et le travail de précision : 300 dpi et plus.

I.2.7 Formats d'image numérique

Un des facteurs des plus importants pour une image numérique est le poids qu'elle représente sur le disque dur. Le poids est exprimé en octets.

CHAPITRE I : IMAGE ET VIDEO NUMERIQUE

Nous avons vu précédemment que chaque composante de couleur RVB est codée sur 8 bits, soit un octet. Pour chaque pixel, la couleur sera donc codée sur 3x8 bits, soit 3 octets.

Dans un format de fichier sans compression, il est donc facile de calculer le poids d'une image. Si la photo fait 3000x2000 px, soit 6 millions de pixels, elle pèsera alors 6000000x3 octets, soit 18 millions d'octets. Un kiloOctet correspond à 1024 octets, et un mégaOctet à 1024 ko. L'image pèse donc 18000000/1024/1024 Mo, soit 17,2 Mo. Le poids de cette image est tout de même conséquent, malgré une définition qui est plutôt faible par rapport aux appareils actuels. C'est pourquoi on utilise généralement un format de fichier compressé, qui permet de réduire ce poids.

Un format d'image est une représentation informatique de l'image, associée à des informations sur la façon dont l'image est codée et fournissant éventuellement des indications sur la manière de la décoder et de la manipuler [W04].

Le format d'une image permet d'indiquer à l'ordinateur de quel type il s'agit, en plus de lui faire comprendre que c'est un fichier image et non un fichier musique ou texte. Il existe plusieurs formats d'image où les propriétés d'une même image, telles que son poids ou sa qualité, vont sensiblement différer d'un format à l'autre [W01].

Le tableau suivant récapitule les formats d'image les plus connus et les plus utilisés [W05]:

Nom / extention	Utilisation	Logiciels à utiliser
JPEG / .jpg (Joint Photographic Experts Group)	Ideal pour mettre et échanger des photos sur Internet. Pour la retouche, attention au niveau de compression.	Toutes Les visionneuses Windows (aperçu), diaporamas, tous les logiciels de retouche...
TIFF / .tiff (Tagged Image File Format)	Trop lourd pour Internet. A utiliser pour l'impression et les retouches.	Aperçu Windows, logiciels de retouche (Photoshop, PaintShop...).
PNG / .png (Portable Network Graphics)	Très bon sur Internet, gère les transparences. Bonne qualité mais un peu plus lourd que le JPEG.	Aperçu, diaporama. Pour la retouche, privilégier Fireworks dont c'est le format natif.
GIF / .gif (Graphics Interchange)	Très utile et léger sur Internet : schémas, cartes, icônes, petites animations. A oublier pour l'impression.	Se lit avec tous les logiciels, se retouche et se crée avec ImageReady, Fireworks, etc.

Format)		
PSD / .psd (PhotoShop Document)	Pas pour le Web. A utiliser pour les retouches sous Photoshop.	Format natif de Photoshop.
PSP / .psp (PaintShop Pro)	Idem que le PSD, mais pour PaintShop Pro.	Format natif de PaintShop Pro.
BMP / .bmp (Bitmap)	Format non compressé. C'est le format par défaut sous Paint.	Format Bitmap pour Windows, principalement utilisé avec Paint.

Tab I.1 Les principaux formats d'image

I.3 NOTIONS DE BASE SUR LA VIDEO NUMERIQUE

Nous expliquons dans cette partie les notions de base essentielles concernant les vidéos numériques.

I.3.1 Définition

La vidéo regroupe l'ensemble des techniques permettant l'enregistrement ainsi que la restitution d'images animées, accompagnées ou non de son (des données audio), sur un support électronique et non de type photochimique [W06].

I.3.2 Vidéo analogique et vidéo numérique

L'un des premiers concepts qu'on doit assimiler est la distinction entre vidéo analogique et vidéo numérique. La vidéo analogique représente l'information comme un flux continu de données analogiques, destiné à être affichées sur un écran de télévision (basé sur le principe du balayage). Il existe plusieurs normes pour la vidéo analogique. Les trois principales sont : PAL, NTSC, SECAM. La vidéo numérique, consiste par contre, à afficher une succession d'images numériques. Pour bien assimiler cette différence, il est nécessaire de comprendre le principe des signaux analogiques et numériques.

Le signal analogique, à un instant donné, peut prendre n'importe quelle valeur comprise entre le minimum et le maximum autorisés. Les signaux numériques, en revanche, sont exclusivement transmis sous forme de points sélectionnés par intervalles sur la courbe. Un ordinateur, utilise un signal numérique de type binaire, qui décrit ces points sous la forme

d'une suite de valeurs minimales ou maximales correspondant respectivement au zéro et au un [B01].

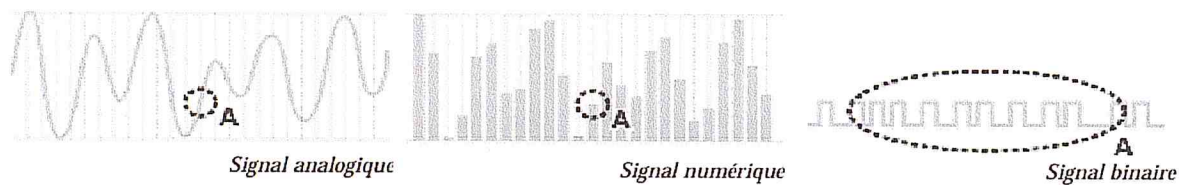


Fig I.7 Signaux vidéo

Les signaux numériques présentent plusieurs avantages, dont l'un des plus importants est la très haute fidélité de la transmission. Avec un signal analogique, l'extrémité réceptrice n'a aucun moyen de distinguer le signal original d'un bruit quelconque susceptible d'être introduit lors de la transmission. Cependant, avec un signal numérique, il est beaucoup plus facile de distinguer l'information émise originale des bruits éventuels. De ce fait, un signal numérique peut être transmis et dupliqué aussi souvent qu'il est nécessaire sans perte de fidélité.

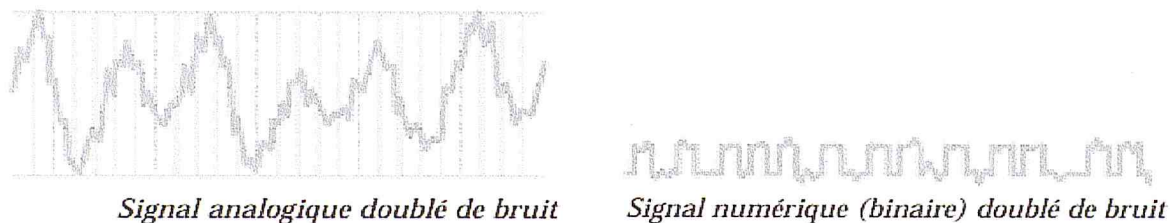


Fig I.8 Bruit dans les signaux vidéo

Un autre avantage du mode numérique est l'aisance de la manipulation des images animées. Le « document » animé peut être modifié rapidement, facilement et autant de fois qu'on le juge nécessaire, que ce soit pour y intégrer des sons, des titres ou des effets spéciaux.

De ce fait, le monde de la vidéo connaît actuellement une phase de mutation du mode analogique vers le mode numérique, et ce à tous les niveaux: production, postproduction, diffusion et conservation de données audiovisuelles.

I.3.3 Numérisation

Toute source audiovisuelle analogique doit être transformée en source numérique (binaire) afin qu'elle puisse être reconnue, traitée et diffusée via un ordinateur et un réseau informatique. Ce procédé s'appelle la numérisation.

Mode analogique : Lorsqu'on travaille avec une caméra vidéo analogique ou tout autre matériel analogique (comme un magnétoscope), on aura besoin d'un dispositif d'acquisition vidéo capable de « numériser » de la vidéo au format analogique. Il s'agit généralement d'une **carte d'acquisition vidéo** à installer sur notre ordinateur.

Mode numérique : De nos jours, les caméscopes vidéo numériques sont devenus largement disponibles sur le marché, à des prix abordables. Les caméscopes numériques « traduisent » les données qu'ils enregistrent au format numérique à l'intérieur de la caméra, en utilisant un format appelé **DV** (*Digital Video* ou vidéo numérique). Ainsi, l'ordinateur peut traiter ces données numérisées puisqu'elles sont alimentées directement à partir de la caméra via une interface de communication. La plus communément utilisée est l'interface **IEEE 1394**¹.

I.3.4 Nombre d'images par seconde et résolution

Afin que l'œil humain puisse percevoir une animation, les images doivent défiler à une certaine vitesse. La cadence de l'animation est désignée sous le terme de **nombre d'images par seconde**, exprimé en **FPS** (*Frames per second*, en français trames par seconde) [B02].

Pour qu'une légère animation, soit perceptible à l'œil, une cadence d'environ 10 images par seconde est nécessaire, sinon, l'effet produit est celui d'un rythme saccadé.

Au delà, la lecture des images est encore plus fluide. Au cinéma, les films sont tournés et projetés sur une base de 24 images par seconde. A la télévision, les films sont projetés sur une base de 30 images par seconde environ, variable selon les standards vidéo utilisés dans les pays de diffusion [B01].

Cependant, la qualité des films ne dépend pas seulement du nombre d'images par seconde. La quantité d'informations contenues dans chaque image est également déterminante. Elle est désignée sous le terme de **résolution** d'image. La résolution correspond en règle générale au nombre de pixels constituant l'image affichés à l'écran, par exemple, 640 x 480 ou 720 x 480.

Le nombre d'images par seconde et la résolution sont des paramètres très importants en matière de vidéo numérique, car ils déterminent le volume de données à transmettre et à enregistrer en vue de la diffusion.

¹ La norme **IEEE 1394** aussi connue sous le nom **FireWire** (technologie Apple) et **i.LINK** (Corporation Sony). Il s'agit de ports (sur l'ordinateur et la caméra DV) et un câble particulier permettant un transfert ultrarapide de données : autour de 400 méga-octets par seconde.

I.3.5 Débit d'affichage d'une vidéo numérique

Vu que la vidéo numérique s'agit d'un ensemble d'images numériques affichées à une certaine cadence, il est donc possible de connaître le débit nécessaire pour son affichage, c'est-à-dire le nombre d'octets affichés (ou transférés) par unité de temps. Ainsi le débit nécessaire pour afficher une vidéo (en octets par seconde) est égal à la taille d'une image que multiplie le nombre d'images par seconde. Par exemple, soit une image colorée (24 bits, soit 3 octets) ayant une définition de 640 x 480 pixels. Pour afficher correctement une vidéo possédant cette définition il est nécessaire d'afficher au moins 30 images par seconde, c'est-à-dire un débit égal à :

$$((640 * 480) * 3) * 30 = 27 \text{ Mo/s}$$

I.3.6 Compression

Quel que soit le dispositif qu'on utilise, carte d'acquisition ou caméscope numérique, dans la plupart des cas, la numérisation d'une vidéo entraîne également sa compression (*Bit Rate Reduction*). La compression est nécessaire compte tenu du volume important de données que représente une vidéo dans son état initial.

L'objectif de la compression est de réduire le volume de données tout en conservant une bonne qualité d'image. Il existe plusieurs méthodes de compression pour les fichiers vidéo :

a. Méthodes simples :

Ce type de méthodes consiste à réduire les caractéristiques suivantes :

- La taille de l'image, par exemple de 640*480 à 320*240 pixels ;
- Le nombre d'images par seconde, par exemple, une vidéo comprenant 15 images par seconde occupe moitié moins d'espace disque qu'une vidéo comprenant 30 images par seconde.

Cependant, ces méthodes de compression élémentaires ne fonctionnent pas si la vidéo doit être affichée sur un moniteur de télévision, avec une résolution et un nombre d'images par seconde optimisés.

b. Méthodes sophistiquées :

Il s'avère que l'œil humain est plus sensible aux changements survenus dans la luminance d'une image qu'à ceux survenus dans sa couleur. De ce fait, la plupart des schémas

CHAPITRE I : IMAGE ET VIDEO NUMERIQUE

de compression vidéo tirent parti de cette particularité de la perception humaine, en éliminant une bonne partie des informations relatives aux couleurs de l'image. Tant que ce type de compression n'est pas trop élevé, il est généralement indiscernable [B01]. Quant aux techniques, on distingue entre :

- La compression spatiale (*intratrame*) : Dans cette approche, chaque image est compressé séparément. On tient compte des similarités des informations dans des zones de l'image qui sont proches l'une de l'autre (dans une image, deux points voisins sont souvent similaires).
- La compression temporelle (*intertrame*) : Ce type de compression tire parti de la probable similitude entre une image donnée de la vidéo et celles qui l'entourent (précédente, suivante). De ce fait, au lieu de stocker la totalité de l'image, il suffit de stocker les différences entre celle-ci et l'image précédente.

La compression et la décompression de la vidéo sont gérées par un module appelé **codec**. Les codecs sont fournis soit par du matériel (caméscopes numériques ou cartes d'acquisition), soit par des logiciels (Première d'Adobe, Vidéo Studio d'Ulead,...). Certains codecs ont un taux de compression fixe, et donc un débit fixe, tandis que d'autres peuvent compresser chaque image différemment selon son contenu, avec par conséquent un débit susceptible de varier dans le temps. Certains codecs permettent de définir un attribut de qualité qui contrôle le débit autorisé. Le tableau ci-après présente différents types de codecs vidéo ainsi que leurs domaines d'application :

Format	Résolution	Type de compression	Débit	Application
MJPEG (Motion JPEG)	720 x 486 pixels/image	Intratrame	0,5 – 25 Mo/s	Générales
MPEG-1	352 x 240 pixels/image	Intratrame	0,01 – 0,06 Mo/s	CD-ROM, Internet
MPEG-2	720 x 486 pixels/image	Intratrame et intertrame	0,01 – 2 Mo/s	DVD, TV satellite
DV (Digital)	720 x 486 pixels/image	Intratrame	3,5 Mo/s	Grand public, professionnelles,

Vidéo)				télédiffusion
D1	720 x 486 pixels/image	aucun	25 Mo/s	Télédiffusion

Tab I.2 Exemples de codes vidéo avec domaines d'application

I.3.7 Les formats de vidéo numérique

Plusieurs fichiers vidéo différents peuvent avoir la même extension (exemple .avi ou .mts) mais utiliser des codecs différents. C'est que le format est un conteneur à l'intérieur duquel on trouve un flux vidéo et un flux audio, chacun pouvant être codé avec de nombreux programmes de compression qui n'ont pas les mêmes qualités [W07].

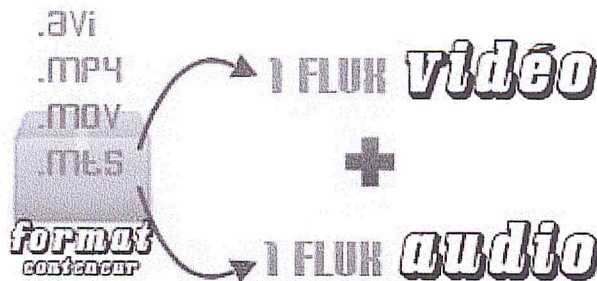


Fig I.9 Formats de vidéo

De la même manière qu'un fichier image, il existe plusieurs formats de fichiers vidéo, à l'intérieur desquels le choix de la définition (qualité) est inversement proportionnel à la taille du fichier généré. Le tableau suivant regroupe les principaux formats vidéo utilisés actuellement:

Format / extension	Utilisation	Logiciel / support
AVI / .avi (Audio Video Interleave)	Dans un fichier AVI, chaque composante audio ou vidéo peut être compressée par n'importe quel codec (DivX et mp3 par exemple). C'est un "conteneur" très pratique, le plus utilisé.	Normalement tous les logiciels de lecture vidéo.
WMV / wmv (Windows Media	Format conteneur d'Apple, pouvant contenir un très large choix de codec, y	Nécessite Quicktime

Video)	compris haute définition.	
DivX et Xvid	Les codecs vidéo les plus utilisés. Permet de diviser par 6 la taille d'un film en DVD sans trop de perte de qualité. Xvid est le concurrent libre de DivX.	Nécessite l'installation des codecs DivX / Xvid à jour, qui sont inclus dans de nombreux lecteurs vidéo.
MKV / .mkv .mka .mks (Matroska Video)	Format conteneur russe, pouvant contenir de très nombreux codecs.	Se lit avec VLC, Media Player Classic et quelques autres après l'installation des Codecs.
H264 / MPEG4 AVC (Advanced Video Coding)	C'est le Codec utilisé pour la haute définition.	Se retrouve dans les Blu-ray. Se lit avec la majorité des lecteurs vidéo à jour.
FLV / .flv (Flash Video)	Le format Flash Video, utilisé en quantité sur Internet (comme Linternaute Video).	C'est le format natif d'Adobe Flash Player, s'ouvre parfaitement avec VLC.
Real Video / .rmvb (RealMedia Variable Bitrate)	Codec vidéo propriétaire de Real Networks	S'ouvre avec RealPlayer ou RealAlternative

Tab I.3 Les principaux formats vidéo

I.4 CONCLUSION

Dans ce chapitre, nous avons abordé les différentes notions de base de l'imagerie et de la vidéo numérique, à savoir : la résolution, la compression et le format, ces trois paramètres sont intimement liés de telle sorte que la modification de l'un ou de l'autre a des conséquences immédiates sur la qualité du fichier image ou vidéo.

Après avoir balayé ces notions de base, nous pouvons maintenant rentrer dans le vif du sujet de notre projet, qui est la détection et le comptage de personnes dans une vidéo numérique.

CHAPITRE II : GENERALITES SUR LE COMPTAGE DE PERSONNES

II.1 INTRODUCTION

Un compteur de personnes est un dispositif utilisé pour mesurer le nombre et la direction des personnes qui traversent un certain passage ou à l'entrée des lieux publics, musées, centres commerciaux, aéroports, métro, etc. La résolution de la mesure est entièrement dépendante de la sophistication de la technologie employée. Le dispositif est souvent utilisé à l'entrée du lieu voulu, de sorte que le nombre total de visiteurs peut être enregistré

Différentes technologies sont utilisées dans les dispositifs de comptage de personnes, tels que les rayons infrarouges, la vision par ordinateur, l'imagerie thermique et des tapis sensibles à la pression.

Dans ce chapitre, nous allons citer les multiples avantages qu'un système de comptage de personnes peut nous offrir, afin de montrer son utilité et son impact sur différents domaines, à savoir : les décisions économiques, le marketing, la sécurité, ...

Nous allons ensuite voir un aperçu sur les différentes approches utilisées actuellement dans le domaine du comptage de personnes. Nous allons comparer les avantages et les inconvénients de chacune de ces méthodes ce qui va expliquer les raisons de notre choix, qui est le comptage de personne en utilisant une caméra.

II.2 LES AVANTAGES D'UN SYSTEME DE COMPTAGE DE PERSONNES

Les données tirées des systèmes de comptage de personnes sont cruciales. Elles permettent de nous procurer pas mal d'informations concernant les tendances de la clientèle fréquentant un endroit donné, cela permettra d'améliorer la rentabilité dans le cas de magasins ou centres commerciaux, et aussi de tenir des statistiques dans les lieux publics (aéroport, métro,...).

Dans le domaine du marketing, on sait que lorsque la circulation humaine fluctue, les affaires suivent. En conséquence, pour comprendre les facteurs qui affectent la circulation humaine, il ne suffit pas juste d'étudier les rapports de ventes ou d'une marche de contrôle

CHAPITRE II : GENERALITES SUR LE COMPTAGE DE PERSONNES

dans un centre commercial, casino ou musée. Un système de comptage de personnes est comme une armée d'observateurs dans un lieu donné, tout le temps, à tous les jours. Les systèmes de comptage permettent de voir plus loin que les chiffres de ventes ou d'achalandage.

Voici quelques constatations très utiles qu'on peut tirer d'un système de comptage de personnes :

- Identifier les effets d'un nouveau compétiteur ;
- Ajuster les taux de location selon la densité réelle de circulation humaine ;
- Comparer l'attraction de différents visuels, thèmes et boutiques dans un centre commercial ou musée ;
- Évaluer l'impact des changements et des rénovations ;
- Identifier où augmenter/diminuer le service ;
- Optimiser la gestion de l'énergie (climatisation et chauffage) ;
- Identifier les meilleures heures d'ouverture ;
- Confirmer le succès des promotions et d'autres activités de marketing tel que les campagnes publicitaires;
- Identifier les zones publiques populaires :
- Obtenir un instantané de l'activité pour chaque magasin ou chaque zone ;
- Adapter l'affectation des employés et optimiser leur planification ;
- Évaluer le succès d'une exposition, l'affluence dans une salle ;
- Gérer l'éclairage dans un lieu public ;
- Améliorer les mesures de sûreté et de sécurité en adaptant le niveau de sécurité selon le flux de personnes entrants et sortants ;
- Assurer la protection et l'assistance nécessaire, dans le cas d'une évacuation par exemple, il est essentiel de savoir combien de personnes sont à l'intérieur du bâtiment à un moment donné.

II.3 ANALYSE COMPARATIVE DES METHODES DE COMPTAGE

Plusieurs méthodes sont disponibles dans le monde du comptage automatique de personnes. Cependant, elles ne sont pas toutes adaptées aux environnements extérieurs et certaines sont simplement ni fiables ni efficaces [W08]. Nous analyserons chaque technologie individuellement dans ce qui suit.

II.3.1 Capteurs directionnels passifs thermiques

Le détecteur passif détecte un signal émis par un objet, dans notre cas un humain. Le signal infrarouge lumineux est émis par le corps à une température spécifique. L'intensité de la radiation thermique augmente selon la température corporelle. Elle est aussi affectée par la composition et la texture de l'objet.

Lorsqu'un humain passe dans le champ de vision d'un détecteur passif thermique, il y a tout d'abord une réduction de l'arrière-plan causée par l'opacité de la personne, puis une nouvelle source de radiation émise par la personne entrant dans le champ de vision. Comme la température corporelle est généralement plus élevée que l'environnement, il en résulte une augmentation de la lumière infrarouge captée par le détecteur. Le détecteur observe la différence de température entre l'humain et son arrière-plan [W08].

Cette technologie possède des faiblesses. Premièrement, si la température de l'environnement et la température corporelle sont similaires, le capteur devient aveugle. Deuxièmement, si une personne demeure dans le champ de vision du détecteur pendant une longue période, elle devient l'arrière-plan et le capteur devient aveugle. Finalement, les capteurs thermiques sont sensibles aux changements rapides de température ainsi qu'aux rayons solaire intenses.

Conséquemment, les capteurs directionnels passifs thermiques ne sont efficaces que dans les environnements où les conditions lumineuses et la chaleur sont contrôlées.

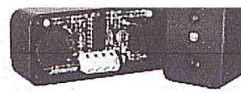
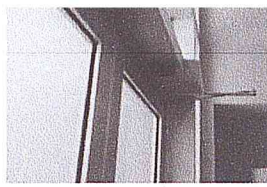


Fig II.1 Capteur directionnel DS-10

II.3.2 Capteurs directionnels à infrarouge actifs

Le capteur à infrarouge actif émet une lumière et détecte la luminosité réfléchiée par un humain. Comme ce capteur a sa propre source de lumière, il n'est pas influencé par la température externe, les conditions lumineuses extérieures ou la présence de personnes immobiles dans son champ de vision, ce qui arrive souvent lorsque les gens attendent en ligne pour payer leur passage dans un véhicule de transport en commun. La technologie infrarouge active est considérée comme la technologie la mieux adaptée aux environnements extérieurs. De plus, il est possible d'installer ce capteur en série pour couvrir des entrées larges [B03].

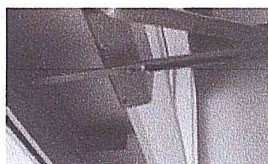
Il est constaté alors que l'utilisation de ces capteurs à infrarouge actif est utile dans les entrées qui sont soumises à d'importantes variations lumineuses ou qui sont exposées à des conditions environnementales difficiles.

II.3.3 Rayons lumineux

Ce type de détecteur agit comme un détecteur de présence qui peut être installé verticalement ou horizontalement. Un rayon lumineux est capturé par un récepteur. L'interruption du rayon lumineux par un objet indique qu'il y a quelqu'un ou quelque chose entre la source du rayon et son détecteur.

Le type "rayon lumineux" n'est pas aussi précis qu'il peut sembler. Les technologies de type rayon ont une zone de détection limitée et requièrent un espace libre entre le transmetteur et l'émetteur. Comme les modèles de circulation des personnes ou des passagers varient constamment, une personne peut passer dans une entrée sans traverser le champ de vision du capteur et ne pas être comptée.

CHAPITRE II : GENERALITES SUR LE COMPTAGE DE PERSONNES



De plus, pour la majorité de ces détecteurs, l'exposition directe à la lumière du soleil peut aveugler le détecteur et causer une perte de comptes. Ce facteur crée donc d'importantes erreurs dans le système.

II.3.4 Les tapis compteur « Tread mats »

Ces tapis comptent en détectant la pression des pieds des passagers ou des personnes alors qu'elles montent ou descendent les marches ou lorsqu'elles traversent l'entrée d'un lieu donné [W08].

Les tapis compteur ont généralement trois rôles: empêcher les portes de se refermer sur une personne qui se trouve dans l'entrée, ouvrir les portes automatiquement et compter les personnes ou les passagers.

Les tapis compteurs sont sujets à être endommagés par la circulation humaine, l'eau et l'exposition aux conditions environnementales. De plus, ces tapis peuvent s'avérer coûteux à installer, puisque l'ajustement de l'équipement est un facteur critique dans sa précision de comptage.

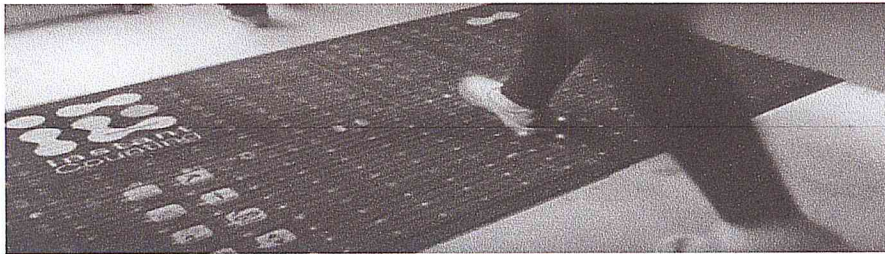


Fig II.2 Tapis de comptage de personnes

II.3.5 Les tourniquets (compteurs mécaniques)

Incontestablement, c'est la méthode la plus précise par rapport aux autres méthodes de comptage, sa précision est plus proche de la perfection. Cependant, cette méthode ne peut pas être utilisée dans les grandes surfaces tel que les supermarchés, car elle crée un obstacle à l'écoulement de la circulation et réduit considérablement le "sentiment de liberté" car elle obstrue les personnes lorsqu'elles entrent et sortent.

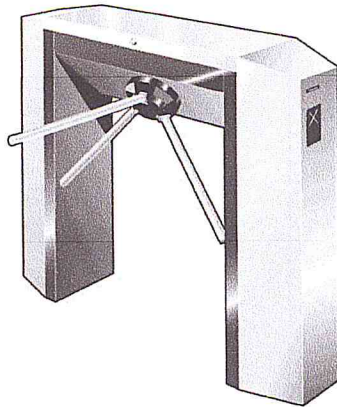


Fig II.3 Les tourniquets

II.3.6 Technologies basées sur les caméras

L'utilisation de caméras vidéo pour détecter et compter les personnes augmente considérablement au cours des dernières années, cela est dû à l'avancement des technologies de la vision par ordinateur et des algorithmes de traitement d'image. Sans oublier le fait que la caméra combine les données numériques et une image visuelle, la détection vidéo se distingue de tous les autres systèmes de détection. Les informations visuelles affichées immédiatement à l'écran par un tel système sont très précieuses pour l'opérateur. Avec une image, ce dernier peut évaluer la situation et prendre les décisions appropriées.

Les systèmes de comptage par caméras filment l'entrée ou une partie de l'entrée. La séquence d'images est ensuite analysée par logiciel pour identifier les gens, compter leur nombre et suivre leurs mouvements (entrant et sortant).

La qualité des composants optiques utilisés ainsi que la puissance de traitement nécessaire pour l'analyse des images contribuent largement à la précision des résultats donnés par ces systèmes. Sans oublier bien sûr la position (la hauteur) de la caméra qui joue un rôle important pour le bon fonctionnement et la précision du comptage.

Les caméras sont utilisées largement dans les environnements intérieurs, cependant elles le sont rarement dans les environnements extérieurs, car elles sont très sensibles à toute variation de lumière, comme le passage du jour à la nuit.

En résumé, le comptage de personnes réalisé par une caméra est la technologie la plus commune et populaire. Elle reste la plus acceptable puisqu'elle correspond à ce que les humains utilisent dans l'interaction visuelle. Par rapport aux autres méthodes, celle du comptage de personnes par caméra s'avère plus avantageuse, d'une part c'est une méthode non intrusive, c'est-à-dire elle n'exige pas la coopération du sujet (en observant les individus

à distance), et d'une autre part les capteurs utilisés sont peu coûteux (une simple caméra) et les coûts d'installation sont faibles (Les zones de détection peuvent être facilement déplacées ou adaptées).



Fig II.4 Caméra de surveillance

II.3.7 Résumé comparatif

Technologie	Les avantages	Les inconvénients
Capteur directionnels passifs thermiques	<ul style="list-style-type: none"> -Ne compte pas les objets ou quoi que ce soit qui n'est pas à la température du corps humain. -Non-intrusif. 	<ul style="list-style-type: none"> -Affecté par les changements rapides de température ou de lumière. -Aveugle dans les environnements chauds. -Affecté par les personnes ou passagers immobiles.
Capteurs directionnels à infrarouge actifs	<ul style="list-style-type: none"> -Insensible aux changements rapides de température ou de lumière. -Insensible aux personnes ou passagers immobiles. -Très précis. -L'installation en série permet de couvrir des entrées larges. -Non-intrusif 	<ul style="list-style-type: none"> - Dans le cas d'entrées larges, la nécessité d'installer plusieurs capteurs implique un coût élevé.
Rayons lumineux « <i>Light beams</i> »	<ul style="list-style-type: none"> -Coût bas. -Non-intrusif. 	<ul style="list-style-type: none"> -Zone de détection limitée (Des angles morts peuvent permettre à quelqu'un de passer sans être détecté). -Requiert une zone de détection sans obstruction.

CHAPITRE II : GENERALITES SUR LE COMPTAGE DE PERSONNES

Les tapis compteur « Tread mats »	-Non-intrusif.	-Rapidement endommagés par la circulation humaine. -Installation et maintenance couteuse. -Peu fiable.
Les tourniquets	- Sa précision est plus proche de la perfection.	- Ne peut pas être utilisée dans les grandes surfaces tel que les supermarchés, car elle crée un obstacle à l'écoulement de la circulation. - Elle obstrue les personnes lorsqu'elles entrent et sortent.
Les Caméras	-Peut fonctionner comme outil de surveillance. -Non-intrusif.	-Affecté par les vibrations et les changements de lumière, de hauteur et de température. -La qualité de l'image influence grandement la précision du traitement.

Tab II.1 Comparaison de quelques méthodes de comptage de personnes

II.4 CONCLUSION

Dans ce présent chapitre, nous avons essayé de balayer les technologies les plus utilisées dans le domaine du comptage de personnes, en comparant leurs avantages et inconvénients. Ce que nous pouvons constater de tout cela, c'est que la précision de chaque technologie dépend largement de l'environnement et des conditions externes.

Dans notre projet, nous avons opté pour l'utilisation des caméras comme technologie de comptage de personnes. Ce choix vient principalement de l'usage très répandu des caméras de surveillance dans les lieux publics. De ce fait, pour faire le comptage de personne, on n'a pas à installer d'éventuels équipements ou capteurs, ce qui réduit largement le coût de notre travail, vu que les caméras sont déjà installées.

Il ne nous reste donc qu'à choisir un bon logiciel de traitement d'image et de développer un bon algorithme de détection et de comptage de personnes. C'est ce que nous allons voir en détail dans les chapitres suivants.

CHAPITRE
III

**PRINCIPAUX
COMPOSANTS D'UN
SYSTEME DE COMPTAGE
DE PERSONNES**

CHAPITRE III: PRINCIPAUX COMPOSANTS D'UN SYSTEME DE COMPTAGE DE PERSONNES

III.1 INTRODUCTION

Les systèmes de comptage de personnes ont de nombreuses applications potentielles dans la vie pratique, à savoir, la vidéosurveillance et la gestion des ressources publiques.

La conception d'un système de comptage de personnes en utilisant des caméras de vidéosurveillance, doit passer impérativement par trois étapes essentielles :

- Détection de personnes en mouvement ;
- Suivi de ces personnes tout au long de la séquence vidéo ;
- Comptage du nombre de ces personnes entrants ou sortants de l'endroit surveillé.

Dans ce chapitre, nous allons présenter chacune de ces étapes ainsi que les techniques et les méthodes les plus utilisées de nos jours pour pouvoir réaliser un système de comptage performant.

III. 2 DETECTION DE PERSONNES

III.2.1 Introduction

La détection de personne dans une vidéo fait partie de l'analytique vidéo, cette dernière consiste à reproduire, à l'aide d'un ordinateur, l'analyse qu'effectuerait un humain en regardant les séquences vidéo provenant de caméras de surveillance.

La détection est une étape décisive et incontournable. En effet, avant de suivre et d'analyser les caractéristiques d'un individu cible (mouvement, vitesse, direction, taille, forme, etc.), il est nécessaire de le détecter et de le segmenter du reste de l'image.

Nous considérons le problème de détection comme un problème de segmentation consistant à séparer ou classer les pixels en 2 classes distinctes, l'**arrière-plan** (*background*) et **premier-plan** (*foreground*). Les zones de l'arrière-plan font référence à toute structure ou objet situé dans le champ de vision de la caméra et ne subissant pas (ou peu) de changements au cours du temps, tandis que les régions du premier-plan correspondent aux éléments de la scène en déplacement (ou susceptible de l'être).

Diverses méthodes ont été proposées dans la littérature afin d'effectuer cette tâche [B04]. Dans les applications où l'on utilise des caméras fixes avec fond statique invariant, l'approche de détection la plus commune et la plus utilisée est la **soustraction de fond** [B05], cela revient au fait que cette approche présente l'intérêt d'être relativement simple et efficace dans des environnements en intérieur pour lesquels le fond présente des caractéristiques suffisamment stables. Et comme les séquences que nous utilisons dans notre projet proviennent d'une caméra fixe, filmant une scène en intérieur avec un fond statique et dans des conditions supposées lentement variables (par rapport à la dynamique des entités d'intérêt), nous avons donc choisi d'utiliser une méthode de soustraction de fond afin d'effectuer la segmentation et la détection des individus.

Le résultat final du processus de soustraction de fond est une image dans laquelle apparaissent uniquement les blobs¹ correspondant aux éléments d'intérêt (éléments du premier-plan).

III.2.2 Détection des éléments d'intérêt par soustraction de fond

La soustraction de fond est une technique qui consiste à estimer une représentation appropriée de la scène appelée « modèle de fond », puis à chercher tout changement ou déviation par rapport au modèle dans chaque image d'entrée traitée. Les régions de l'image d'entrée où il y aura des changements significatifs par rapport au modèle de fond correspondront aux éléments d'intérêt (éléments en mouvement ou nouveaux éléments au premier-plan). Les pixels constituant ces régions sont marqués pour des traitements ultérieurs. Usuellement, des étapes de post-traitement telles qu'une procédure de regroupement en composantes connexes et des traitements de morphologie mathématique sont appliquées afin d'éliminer le bruit (dû à une soustraction de fond imparfaite) et de constituer les régions (blobs) correspondant aux éléments. La **Figure III.1** illustre le schéma du processus de détection par soustraction de fond.

¹ Un **blob** dans le traitement d'image est souvent définie comme un groupe de pixels connectés avec le même état logique. Une analyse de Blob est souvent utilisée pour détecter un objet en mouvement dans le traitement de l'image en temps réel.

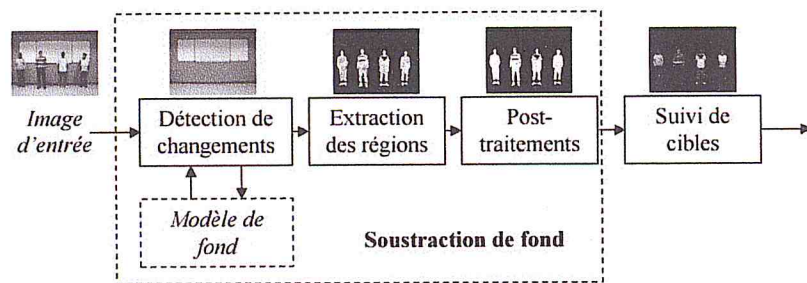


Figure III.1 Schéma de détection par soustraction de fond

Comme nous l'avons cité plus haut, les méthodes de soustractions de fond sont très efficaces lorsque le fond de la scène est statique et reste invariant ou légèrement changeant (pas ou peu de changement d'éclairage, pas d'objets déplacés ou mouvements très lent, etc.). Aussi, pour une meilleure tolérance aux variations légères du fond, des méthodes de soustraction de fond avec modèle adaptatif ont été proposées [B06] [B07], par exemple les méthodes MOG et MOG2. Dans ces méthodes, le modèle de fond est continuellement mis à jour à l'aide de nouvelles images de la séquence traitée. Elles permettent d'adapter le modèle de fond en incorporant de nouveaux éléments stationnaires apparus dans la scène et qui demeurent statiques durant une période suffisamment longue.

Ce genre de méthodes de détection utilise un modèle de fond adaptatif pour détecter des changements dans l'image et ainsi détecter les nouveaux éléments d'intérêt survenus. Le modèle de fond est construit au niveau du pixel (apprentissage et classification), c'est-à-dire que chaque pixel est présumé être le résultat d'un processus indépendant. La valeur de chaque pixel du fond est modélisée par une distribution gaussienne multidimensionnelle² dans l'espace RGB. Quand une nouvelle image est traitée, la valeur observée de chaque pixel est comparée à la distribution correspondante afin de décider si cette valeur correspond à l'arrière-plan ou à un nouvel élément au premier-plan. Les pixels marqués comme premier-plan sont ensuite regroupés en composantes connexes « blobs » qui subissent ensuite des traitements de morphologie mathématique (érosion, dilatations, ouverture, etc.).

Le résultat est alors une liste de zones « blobs » dans l'image qui représentent les régions des éléments d'intérêt (éléments dits de **premier-plan**), tout le reste de l'image étant alors supprimé (car correspondant à des éléments d'**arrière-plan**). A la fin du processus de

² Une loi normale multidimensionnelle ou loi de Gauss à plusieurs variables, est une loi de probabilité qui est la généralisation multidimensionnelle de la loi normale. Alors que la loi normale classique est paramétrée par sa moyenne μ et sa variance σ^2 , la loi multidimensionnelle est paramétrée par un vecteur $\mu \in \mathbb{R}^N$ représentant son centre et une matrice semi-définie positive $\Sigma \in M_N(\mathbb{R})$ qui est sa matrice de variance-covariance.

détection sur chaque image, la distribution de chaque pixel d'arrière-plan est mise à jour en utilisant la dernière observation (mesure), afin de prendre en compte des changements dans l'image de fond.

La première étape de détection d'éléments de premier-plan est donc de générer le modèle de fond. La section suivante décrit en détail le processus de génération de ce modèle.

a. Génération du modèle de fond

Avant que le système n'essaye de localiser une personne sur une scène, il doit connaître (étudier) cette scène afin de détecter les mouvements (variation importante de la scène), c'est ce qu'on appelle la génération du modèle de fond, ou aussi l'estimation de l'arrière-plan. Dans cette étape, on forme une image de référence qui représente la partie arrière-plan de la scène. Cette image est fondamentale pour détecter et poursuivre les objets en mouvement, et sera utilisée pour séparer l'arrière-plan et le premier-plan.

Afin de générer l'image arrière-plan (le modèle de fond), une séquence d'images de la scène de fond est tout d'abord enregistrée. Cette séquence contient un nombre précis d'images au format couleur RGB (par exemple 200 images correspondant à huit secondes d'acquisition avec une cadence de 25 images par seconde). Ces dites images représentent un échantillon représentatif pour la constitution d'un modèle gaussien. Cette approche probabiliste permet de prendre en compte les faibles variations d'intensité lumineuse considérées comme étant des bruits de mesures dans le modèle. Chaque pixel de chaque canal couleur (R, G et B) de l'image est traité séparément. Nous détaillons ici les opérations appliquées à un pixel, sachant que celles-ci seront appliquées à tous les pixels de l'image. Chaque pixel est modélisé par trois distributions gaussiennes (une distribution sur chaque canal), chacune caractérisée par sa moyenne μ et sa variance σ , de telle sorte que la moyenne caractérise la couleur dominante prise par le pixel (couleur d'arrière-plan), et la variance caractérise la variabilité autour de cette valeur. La **Figure III.2** illustre l'exemple de la distribution des couleurs d'un pixel de l'image. Il est à noter que même si les couleurs ne sont pas en réalité distribuées de façon gaussienne, une telle approximation peut être appliquée en pratique [B08].

Un pixel de fond de l'image à la position (i,j) est modélisé par trois distributions gaussiennes $[\mu_R(i,j), \sigma_R(i,j)]$, $[\mu_G(i,j), \sigma_G(i,j)]$ et $[\mu_B(i,j), \sigma_B(i,j)]$, qui correspondent à la modélisation de ses trois composantes couleurs, respectivement rouge (*Red*), verte (*Green*) et bleue (*Blue*). Initialement, la moyenne de chaque pixel sur chaque canal est initialisée

à la première trame de la séquence d'images par les valeurs respectives de chaque canal pour ce pixel. La valeur de la variance est toujours maintenue au-dessus d'une valeur minimale σ_{min} pour tolérer le bruit dans l'image. Après initialisation, deux tâches sont continuellement exécutées : Premièrement, une image de détection est générée en comparant l'image actuelle à l'image de référence (modèle du fond). Deuxièmement, la distribution du modèle du fond est mise à jour en utilisant des informations issues de l'image actuelle. Ces deux étapes seront détaillées dans les deux parties suivantes.

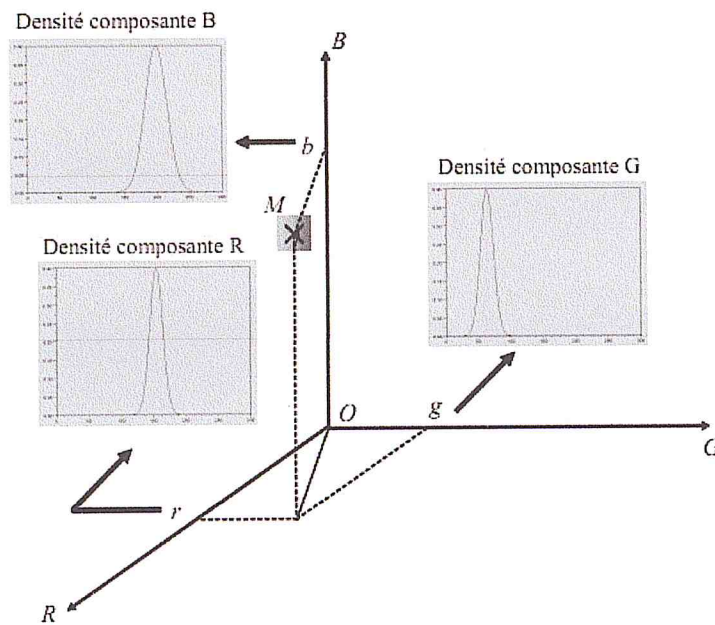


Figure III.2 Caractérisation de couleur d'un pixel (M) de l'image à l'aide de distributions gaussiennes dans l'espace de représentation de couleur RGB

b. Génération de l'image de détection

L'image de détection est l'image qui résulte de la comparaison de l'image courante avec le modèle de fond (arrière-plan). Elle contient donc les éléments de différence entre ces deux images. Aussi, chaque pixel de l'image doit être classé comme faisant partie de l'arrière-plan ou d'un élément de premier-plan. Les pixels classés dans la première catégorie auront la valeur 0 (noir) dans une image binaire de détection, et les pixels de la deuxième catégorie auront la valeur 1 (blanc) [B09].

Le système décide si un pixel occupant la position (i,j) dans l'image et ayant pour valeurs (pour ses trois composantes couleurs) $I(i,j) = [R(i,j) G(i,j) B(i,j)]^T$ appartient à l'arrière-plan ou à un élément de premier-plan en calculant la distance de type Manhattan, pour

chaque canal pris séparément, entre ses valeurs et les moyennes de distributions correspondantes pour cette même position, soit $\mu(i,j) = [\mu_R(i,j) \ \mu_G(i,j) \ \mu_B(i,j)]^T$ dans le modèle de fond. Nous devons donc calculer :

$$d_R(i,j) = |R(i,j) - \mu_R(i,j)|$$

$$d_G(i,j) = |G(i,j) - \mu_G(i,j)|$$

$$d_B(i,j) = |B(i,j) - \mu_B(i,j)|$$

Si pour ce pixel, la distance pour un des canaux est supérieure à un seuil $\lambda_{canal}(i,j)$ avec Canal $\in \{R,G,B\}$, alors ce pixel est marqué comme pixel de premier-plan (valeur 1), sinon comme pixel d'arrière-plan (valeur 0). Ce seuil $\lambda_{canal}(i,j)$ est déterminé en fonction de la variance de distribution gaussienne pré-calculée : $\lambda_{canal}(i,j) = 2 \sigma_{canal}(i,j)$ [B10].

Pour résumer, un pixel à la position (i,j) admettant pour composante couleur $I(i,j) = [R(i,j) \ G(i,j) \ B(i,j)]^T$ est classé comme pixel de premier-plan si :

$$|R(i,j) - \mu_R(i,j)| > 2 \sigma_R(i,j) \text{ ou}$$

$$|G(i,j) - \mu_G(i,j)| > 2 \sigma_G(i,j) \text{ ou}$$

$$|B(i,j) - \mu_B(i,j)| > 2 \sigma_B(i,j).$$

La **Figure III.3** illustre la segmentation d'éléments de premier-plan. On voit que les pixels sont classés sur chaque canal couleur séparément. Les résultats de la classification des trois canaux sont fusionnés pour obtenir une image binaire de détection générale. Cette image binaire subit ensuite des traitements morphologiques mathématiques pour réduire le bruit (érosion, dilatations, ouverture,...) et un regroupement en composantes connexes afin d'agréger les pixels de premier-plan en blob entiers.

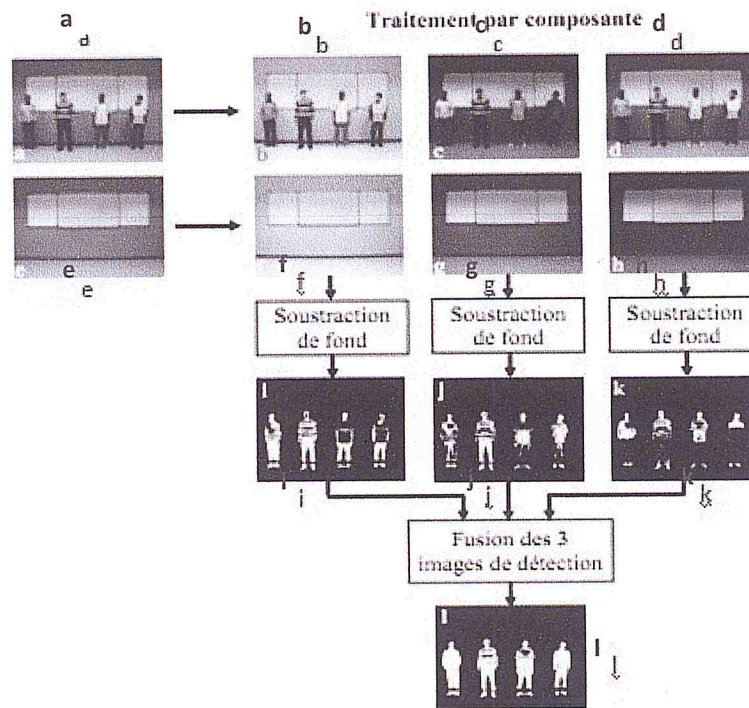


Fig III.3 Segmentation d'éléments de premier-plan

Dans la Figure III.3 ci-dessus, les images (a) et (e) représentent respectivement l'image d'entrée et le modèle de fond (RGB). Les images (b), (c) et (d) représentent respectivement les images d'entrée des canaux R, G et B. Les images (f), (g) et (h) représentent les images de fond (R, G et B). Les images (i), (j) et (k) représentent les images segmentées (R, G et B). Et enfin, l'image (l) représente le résultat de la segmentation et la détection des éléments du premier-plan.

Après cette opération, le modèle de fond doit être mis à jour et prendre en compte d'éventuelles variations de l'arrière-plan. En effet, même si le fond est censé être invariant, de faibles variations peuvent toujours survenir (variation d'éclairage, apparition d'ombres ou de reflets, présence de nouveaux objets stationnaires etc.). Ainsi, une mise à jour du modèle de fond est effectuée.

c. Mise à jour du modèle de fond

L'estimation de l'arrière-plan doit être dynamique, c'est-à-dire que l'image arrière-plan doit être mise à jour. Cette idée est très importante pour faire un bon algorithme de comptage de personnes. Par exemple, si le système de comptage de personnes est placé à l'entrée d'un bâtiment, quelques petites et lentes modifications se produiront durant toute

la journée et peuvent ainsi parasiter l'algorithme de comptage (plus particulièrement l'algorithme d'estimation de l'arrière-plan). En réalité, durant la journée, l'intensité de la lumière du soleil change, ou pour certaines raisons, quelques objets peuvent être déplacés ou ajoutés à la scène de travail. Alors, si l'estimation de l'arrière-plan n'est jamais recalculée, l'algorithme détectera toujours des variations.

Pour prendre en compte ses éventuels changements, les paramètres de moyenne et de variance sont mis à jour régulièrement de façon récursive à l'aide d'un paramètre α appelé **taux d'apprentissage**, de telle sorte qu'après que chaque pixel de l'image actuelle k ait été classé comme premier-plan ou arrière-plan, les distributions couleur (moyenne et variance) de tous les pixels ayant été marqués comme pixels d'arrière-plan sont mises à jour par :

$$\begin{aligned}\mu_{canal}(i,j)_K &= (1-\alpha) \cdot \mu_{canal}(i,j)_{K-1} + \alpha \cdot I(i,j) \\ \sigma^2_{canal}(i,j)_K &= (1-\alpha) \cdot \sigma^2_{canal}(i,j)_{K-1} + \alpha \cdot (\mu_{canal}(i,j)_K - I(i,j))^2\end{aligned}$$

III. 3 SUIVI DE PERSONNES

III.3.1 Introduction

Une fois qu'un individu d'intérêt a été détecté dans la scène, la procédure de suivi est déclenchée. L'objectif du suivi d'une cible est de déterminer sa position de manière continue et fiable tout au long du flux vidéo. Dans les systèmes de comptages de personnes, le suivi représente une étape primordiale. En effet, le processus de suivi d'un individu cible permet de déterminer sa trajectoire et ainsi sa direction, c'est-à-dire permettre de préciser si cette personne est entrante « IN » ou sortante « OUT » de l'endroit surveillé. Ainsi, ceci permettra d'établir le comptage du nombre de personnes entrants et sortants de cet endroit. Aussi, le suivi d'une personne permet d'éviter de la compter plusieurs fois.

Le suivi de cible (« personnes » dans notre cas) consiste à associer les cibles détectées dans l'image courante avec celles détectées sur les images précédentes. Il s'agit de maintenir l'identité des cibles et l'évolution temporelle de leurs positions. Ce problème peut être vu comme un problème de localisation spatiale et temporelle des cibles présentes dans la scène. De nombreuses approches de suivi de cibles se basent sur l'apparence d'une cible. L'utilisation de ces méthodes nécessite une représentation pertinente de la cible possédant des primitives fiables pour décrire son contenu. Le choix d'un modèle d'apparence est une étape cruciale et joue un rôle central dans les techniques de suivi visuel d'objet [B04].

III.3.2 Représentation de la forme d'une cible

Les cibles peuvent être représentées de nombreuses façons et le choix de la représentation de la forme d'une cible dépend fortement du domaine d'application. Nous reprenons dans cette section la classification proposée dans [B11] :

- **Un ensemble de points** : Une cible peut être représentée par un point. Il peut s'agir par exemple de son centre de masse, du centre de sa boîte englobante³, ou tout autre point caractéristique de la forme. Il s'agit d'une représentation simple de la localisation 2D (ou 3D) de la cible. Cette représentation se généralise à un ensemble de points auxquels peuvent être associés des descripteurs locaux de couleur, de texture ou de mouvement.
- **Une forme géométrique** : La cible est représentée par une forme géométrique, par exemple un rectangle ou une ellipse, permettant une description de la dimension de la cible. Le mouvement des cibles associées est généralement modélisé à l'aide de transformations de translations, affines ou projectives. Cette représentation se généralise par des modèles articulés, composés d'un ensemble de formes géométriques 2D ou 3D particulièrement utilisé dans la modélisation du corps humain [B12] [B13].
- **Un contour** : La représentation d'une cible par son contour permet une description plus complète de la forme de la cible. Un contour peut être vu comme étant un ensemble de points ordonnés généralement estimé à l'aide d'une analyse du gradient d'intensité au voisinage d'un pixel. La région interne du contour est appelée silhouette de l'objet et peut être utilisée conjointement à l'information de contour pour le suivi d'objets [B14] [B15].
- **Un squelette**: Le squelette d'une cible peut être extrait pour caractériser la forme de la cible ou d'une forme géométrique. Ce modèle est utilisé en tant que descripteur de forme pour la reconnaissance d'objets [B16]. Cette représentation peut être aussi bien utilisée sur des objets déformables que des objets rigides.

La **Figure III.4** résume les différentes représentations de la forme d'une cible, où :

- (a)-(b) Représentation à l'aide d'un ensemble de points ;
- (c)-(d)-(e) Représentation à l'aide d'un ensemble de formes géométriques ;
- (f) Représentation à l'aide du squelette ;

³ Une **boîte englobante** est la plus petite forme rectangulaire qui entoure un blob.

- (g)-(h)-(i) Représentation à l'aide du contour (partiel ou non) et de la silhouette d'une cible.

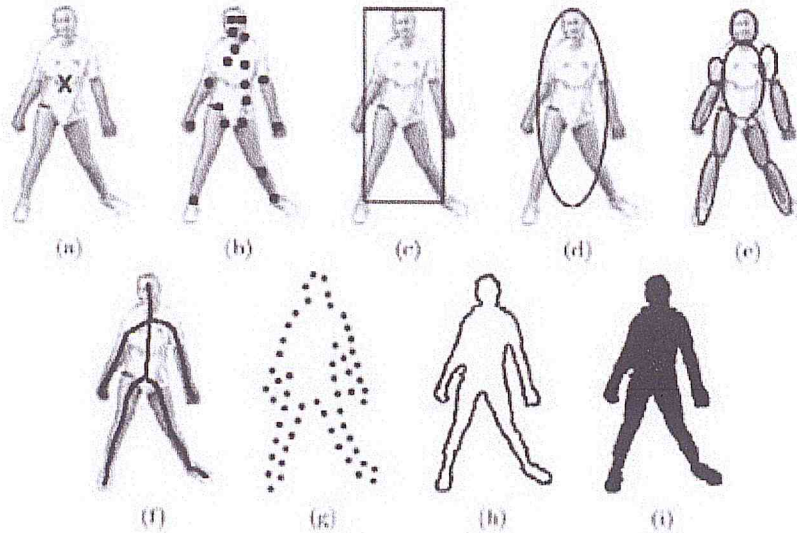


Fig III.4 Exemples de représentation de la forme d'une cible

III.3.3 Technique de suivi de cibles

Il existe de nombreux états de l'art dans la littérature traitant du sujet de suivi d'objets. Nous regroupons dans cette partie une brève présentation des algorithmes de suivi les plus populaires, à savoir : le filtre de Kalman, l'algorithme de condensation (les filtres particuliers), le Mean-Shift, le flot optique, le suivi de contour et l'algorithme du plus proche voisin.

a. Filtre de Kalman

Le filtrage de Kalman a été utilisé de façon intensive dans les algorithmes de suivi d'objets. Il est généralement utilisé afin de prédire la position de la cible suivie étant données ses positions précédentes. Cette étape de prédiction est une caractéristique importante pour les systèmes de suivi de cibles, puisqu'elle permet d'aider à la mise en correspondance des cibles et de maintenir une cohérence temporelle de la trajectoire grâce aux contraintes du modèle.

Le filtre de Kalman s'inscrit dans un cadre d'estimation récursive bayésienne dans laquelle l'estimation de la position est effectuée conditionnellement aux mesures et aux états précédents. L'algorithme de Kalman n'est équivalent à l'estimation bayésienne en terme d'optimisation que si les processus mis en jeu dans le modèle sont des processus markoviens

et gaussiens, ce qui n'est plus le cas lorsque les systèmes sont non linéaires [B04].

Le filtre de Kalman comporte 3 étapes, une étape de prédiction de l'état, une étape de mesure de l'observation et une étape de correction de la prédiction.

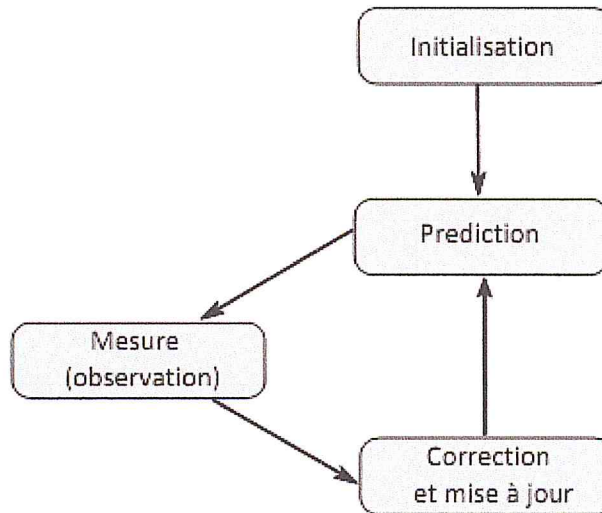


Fig III.5 Etape de filtrage de Kalman

b. Algorithme de Condensation - Filtres particulières

Une limitation du filtre de Kalman est l'hypothèse que les variables d'état possèdent une distribution gaussienne. Par conséquent, il donnera une estimation pauvre des variables qui ne suivent pas celle-ci [B17]. Cette limitation peut être contournée en utilisant l'algorithme de « Condensation ». L'algorithme de « Condensation », qui appartient au groupe des filtres particuliers, a été introduit par Isard et Blake [B18] pour le suivi de personnes. Cet algorithme est un estimateur de l'état d'un système non linéaire Markovien soumis à des excitations aléatoires possiblement non Gaussiennes.

c. Mean-Shift

L'algorithme Mean-Shift, au même titre que l'algorithme de condensation ou le filtre de Kalman, est très populaire et très utilisé par la communauté de vision par ordinateur. Initialement introduit par Comaniciu et Meer [B19], le Mean-Shift est une méthode qui maximise de façon itérative la similarité entre l'apparence d'un objet et celle d'un candidat autour d'une position estimée [B04]. Chaque objet d'intérêt est modélisé par la distribution statistique de ses caractéristiques telles que la couleur, la position, la texture, etc. Le principe est de rechercher, dans l'image courante, le maximum local de la densité de probabilité du vecteur d'observation permettant ainsi de faire évoluer les modèles. La méthode du Mean-Shift est surtout efficace lorsque les cibles suivies ont une apparence bien distincte du reste

de l'image (forme, couleur, texture, etc.).

d. Flot optique

Cette approche fait une approximation du mouvement des objets suivis par l'estimation des vecteurs origines des pixels dans des séquences d'images, de sorte à représenter le champ de vitesse qui déforme une image dans un autre espace de haute dimension. Ainsi, la technique du calcul du flot optique est une méthode différentielle où l'on cherche à estimer, en tout point, le vecteur déplacement comme étant fonction du changement d'intensité du pixel ainsi que de son voisinage [B20]. Les méthodes de flot optique sont sensibles au changement d'éclairage.

e. Contours Actifs

Les contours actifs, Appelés également *snake*, ont été introduits par Kass, Witkin et Terzopoulos [B21], ils permettent de prendre en compte la complexité des contours dans le suivi. Un contour actif est une structure dynamique d'un ensemble de points mobiles qui évoluent itérativement dans l'image afin d'épouser au mieux la forme d'un objet d'intérêt [B04].

La principale hypothèse de cette méthode est que le contour de l'objet suivi présente des caractéristiques particulières (contraste de texture ou discontinuité de mouvement), ce qui sert à recalculer la position du contour d'une image à l'image suivante en faisant converger celui-ci vers les points ayant les mêmes caractéristiques [B21] [B22]. L'objectif est alors, à partir d'une position d'initialisation située près de la cible, de déformer les contours actifs afin que ceux-ci s'ajustent aux contours de la cible tout au long de ses déplacements.

Ces méthodes peuvent suivre des cibles en mouvement avec des tailles et des formes différentes, et se veulent insensibles aux changements d'éclairage [B23]. Cependant, elles gèrent assez mal des déplacements rapides des cibles et sont en outre coûteuses en calcul.

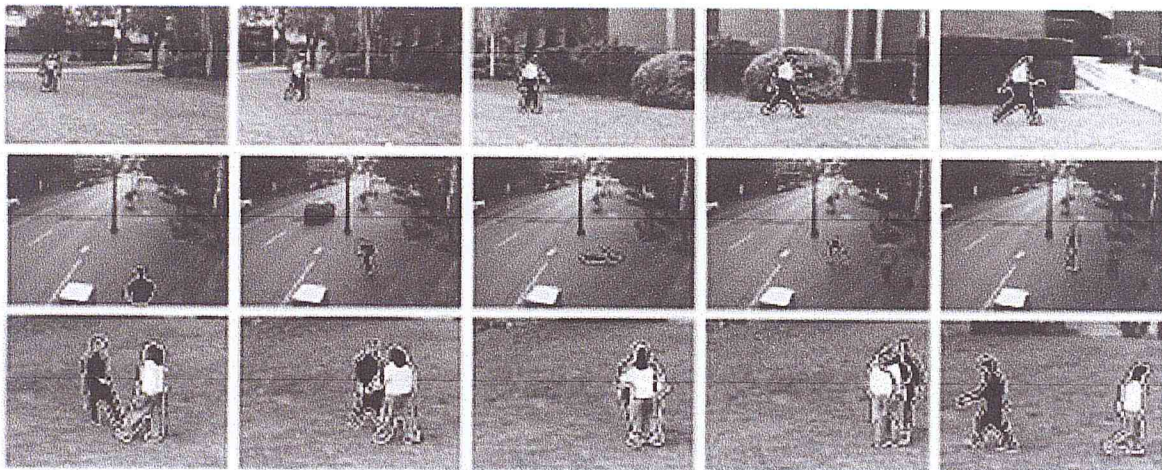


Fig III.6 Exemple de suivi de contours des objets

f. L'algorithme du plus proche voisin

L'algorithme du plus proche voisin « *Nearest Neighbor* » consiste à associer chaque objet au candidat le plus proche en terme de similarité. Cette méthode considère que chaque détection a été générée par la cible la plus proche. Les cibles sont caractérisées par une zone de confiance caractérisant la probabilité que la cible ait généré l'observation. Si on considère les densités de probabilité gaussiennes, la similarité s'exprime à l'aide de la distance de Mahalanobis [B04] donnée par l'équation (1), où m caractérise la cible, t caractérise l'état de suivi extrapolé et σ est la matrice de covariance [B24] :

$$D(m, t) = \frac{\sqrt{\frac{(m-t)^2}{\sigma(m)^2}} + \sqrt{\frac{(m-t)^2}{\sigma(t)^2}}}{2} \quad (1)$$

Equation 1 : Mesure de la distance de Mahalanobis

Une fois la similarité estimée, seules les mesures statistiquement proches d'une cible sont utilisées pour mettre à jour son vecteur d'état.

De nouveaux suivis (nouvelles détections) sont créés pour toutes les mesures qui ne pouvaient pas être appariés. Les cibles qui ne reçoivent pas de mesures qui leur sont assignées seront maintenues pendant un certain laps de temps après lequel ils sont retirés de l'ensemble actif de cibles.

III. 4 COMPTAGE DE PERSONNES

Une fois une personne identifiée et suivie, le processus de comptage est alors déclenché. Dans cette partie, nous allons expliquer le principe et les méthodes généralement

utilisées pour effectuer le processus de comptage de personnes dans une vidéo numérique.

L'approche suivie pour effectuer le comptage de personnes, consiste à déterminer la direction des blobs qui traversent une ligne virtuelle afin d'incrémenter le bon compteur (compteur entrée ou de sortie). Il existe deux façons différentes pour le faire : utilisation de deux lignes virtuelles ou bien juste une seule ligne virtuelle.

III.4.1 Comptage avec deux lignes virtuelles

La première méthode consiste à utiliser deux lignes virtuelles qui représentent l'entrée « *IN* » et la sortie « *OUT* ». Si un blob (représentant une personne) traverse complètement une ligne virtuelle, dans le bon sens, alors le compteur correspondant s'incrémente. Étant donné que le programme mémorise les caractéristiques de chacun des blobs pendant le processus du suivi, il pourra alors facilement en déduire le sens du mouvement (par exemple, en examinant la position du barycentre).

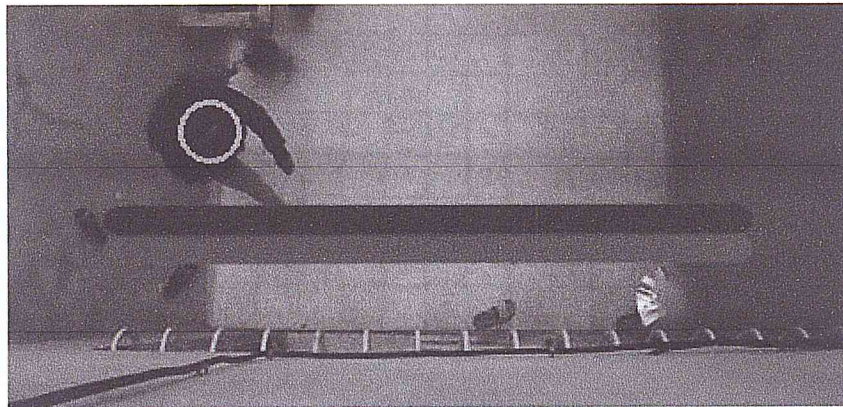


Fig III.7 Exemple de comptage avec deux lignes virtuelles

III.4.2 Comptage avec une ligne virtuelle

La deuxième méthode consiste à utiliser une seule ligne virtuelle qui délimite clairement la zone *IN* de la zone *OUT*. Pour compter les gens, le programme peut par exemple examiner la position du segment bas de chaque boîte englobante du blob suivi dans deux frames consécutifs. Si, dans le frame T, le segment est en dessous de la ligne virtuelle et, dans le frame suivant, le même segment est en dessus alors le programme incrémente sa valeur de comptage *IN*. L'autre façon de comptage en utilisant une seule ligne virtuelle est d'examiner la position du barycentre de chaque blob.



Fig III.8 : Exemple de Comptage avec une seule ligne virtuelle

III.5 CONCLUSION

Dans ce présent chapitre, nous avons présenté les concepts de base d'un système de détection et comptage de personnes dans une vidéo numérique, en fournissant une description générale de ses principaux composants, à savoir : un système de détection, un système de suivi et enfin un système de comptage de personnes.

Dans le chapitre suivant, nous allons décrire en détail la mise en œuvre d'un système de comptage pour le CERIST, en expliquant les approches que nous avons opté de suivre et les choix techniques qui nous ont conduit à la réalisation de notre système de comptage.

CHAPITRE
IV

**MISE EN ŒUVRE DU
SYSTEME DE COMPTAGE**

CHAPITRE IV : MISE EN ŒUVRE DU SYSTEME DE COMPTAGE

IV.1 INTRODUCTION

Au cours des chapitres précédents, nous avons essayé de présenter toute la base nécessaire à la compréhension du thème de manière générale, ainsi que les techniques nécessaires pour sa réalisation. Dans le présent chapitre nous allons expliquer en détail la méthodologie et la logique suivies tout au long de ce travail pour aboutir à notre objectif à savoir: La conception et la mise en œuvre d'un système de détection, suivi et comptage de personnes dans une vidéo numérique. Afin d'y parvenir, plusieurs problématiques se posent :

- Comment arriver à détecter une présence humaine dans une vidéo ?
- Après détection d'une personne, comment suivre son parcours tout au long de la séquence vidéo ?
- Comment compter le nombre de personnes visitant l'endroit surveillé (filmé) ?

IV.2 FONCTIONNEMENT DE NOTRE SYSTEME DE DETECTION, SUIVI ET COMPTAGE DE PERSONNES

Dans cette partie, nous allons expliquer en détail les étapes et les choix techniques qui nous ont conduits au développement et à la réalisation de notre système de détection, suivi et comptage de personnes dans une séquence vidéo.

Comme son nom l'indique, le développant de notre système doit passer impérativement par trois étapes ou processus essentiels :

- Processus de détection de personnes
- Processus de suivi de personnes
- Processus de comptage de personnes

Notre projet présente une méthode de comptage de personnes avec une caméra installée en hauteur (ou sur le plafond) utilisée à des buts de vidéosurveillance, pour des applications en intérieur, cela voudrait dire que l'endroit surveillé est supposé être un endroit fermé (l'entrée d'un bâtiment, d'un magasin, etc.) sous des conditions d'éclairage satisfaisantes, et que le fond de la scène est statique et reste invariant ou légèrement changeant

(pas ou peu de changement d'éclairage, pas d'objets déplacés, etc.).

Notre approche peut se résumer ainsi :

- Capture et lecture de la vidéo image par image (frame par frame) ;
- Traitement de chaque frame à travers une série de fonctions (redimensionnement, conversion et filtrage) ;
- Soustraction de fond (génération de l'arrière-plan et de l'avant-plan, puis mise-a-jour de l'arrière-plan) ;
- Génération des blobs candidats ;
- Suivi des blobs candidats (personnes) ;
- Comptage de personnes entrantes et sortantes de l'endroit surveillé.

IV.2.1 Processus de détection de personnes

Nous allons présenter dans cette partie les étapes de traitements que nous allons appliquer aux séquences d'images (frames) de notre vidéo pour détecter et isoler les individus présents dessus afin de tenter de les compter ultérieurement.

Le diagramme suivant résume notre approche de détection de personne :

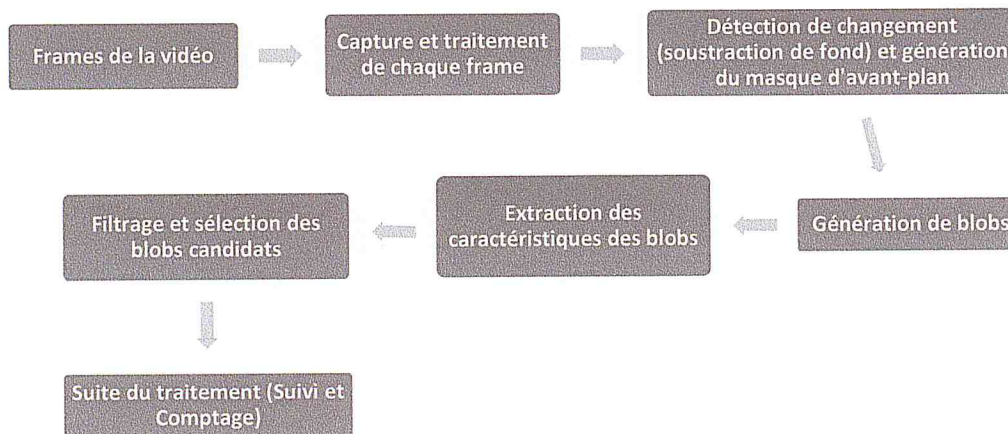


Fig IV.1 Approche de détection de personnes

IV. 2.1.1 Capture et traitement :

Chargement de l'image : Les frames de notre séquence vidéo seront acheminés un par un vers notre programme et reçus dans une structure spécifique aux images numériques.

Redimensionnement de l'image : Après chargement de l'image, on va redimensionner sa taille en une taille plus petite. Ceci est fait pour diminuer le volume des calculs et le temps de traitement de l'image, tout en veillant à ne pas trop perdre de la qualité de l'image.

Conversion et filtrage de l'image : Afin d'alléger les traitements sur chaque image, nous allons convertir chacune d'entre elle de l'espace couleur RGB vers l'espace niveaux de gris. Aussi, lors de l'acquisition, des effets indésirables peuvent s'introduire dans les données de l'image, sorte de parasites tels que le bruit (pixels isolés de valeur erronée). Ces bruits entraînent l'altération de l'image. Ainsi, nous allons atténuer ce bruit et estomper l'image en effectuant un lissage d'image avec un **filtre médian**.

L'idée principale du filtre médian est de remplacer chaque entrée par la valeur médiane de son voisinage. Par exemple, si on considère ces neuf pixels, dont une valeur aberrante (ici 111) :

5	6	7
6	111	8
7	8	9

Le filtre médian va considérer les valeurs du voisinage par valeurs croissantes :

5	6	6	7	7	8	8	9	111
---	---	---	---	---	---	---	---	-----

Et prendre la valeur médiane, ici la valeur 7. La sortie du filtre donnera donc :

5	6	7
6	7	8
7	8	9

Ce qui a permis de remplacer la valeur aberrante par une valeur "de consensus" entre les valeurs voisines.

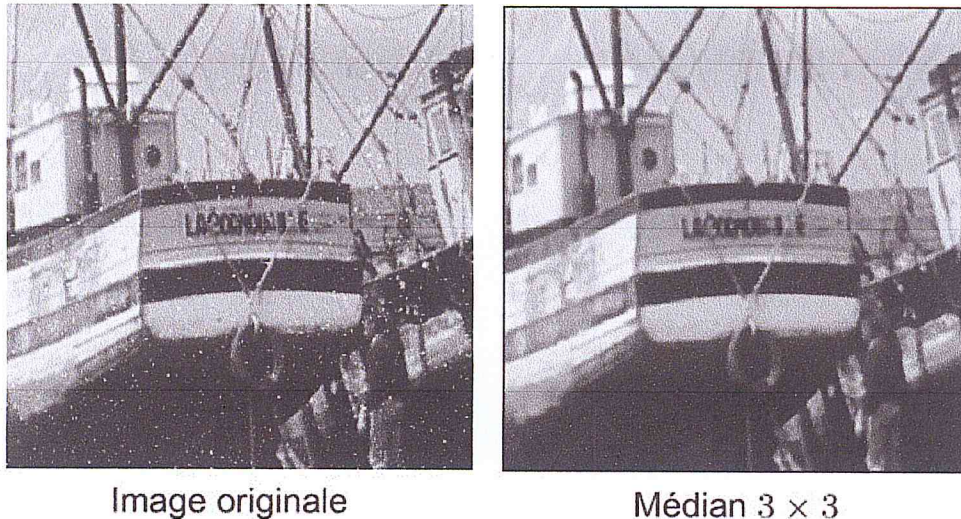


Fig IV.2 Exemple d'application d'un filtre médian 3×3

IV. 2.1.2 Soustraction de fond BS:

Comme nous l'avons cité dans le chapitre précédent, nous considérons le problème de détection comme un problème de segmentation consistant à séparer ou classer les pixels en deux classes distinctes, l'**arrière-plan** (*background*) et **premier-plan** (*foreground*). Les zones de l'arrière-plan font référence à toute structure ou objet situé dans le champ de vision de la caméra et ne subissant pas (ou peu) de changements au cours du temps, tandis que les régions du premier-plan correspondent aux éléments de la scène en déplacement. Techniquement, on devrait extraire le premier-plan en mouvement de l'arrière-plan statique. Ce processus s'appelle la soustraction de fond.

La soustraction de fond, appelée en anglais « *Background subtraction* », est une technique courante et largement utilisée pour générer un masque de premier-plan (une image binaire contenant les pixels appartenant à des objets en mouvement dans la scène) en utilisant des caméras statiques. Comme son nom l'indique, cette méthode calcule le masque de premier-plan en effectuant une soustraction entre l'image courante et le modèle de fond qui contient la partie statique de la scène ou, plus en général, tout ce qui peut être considéré comme arrière-plan étant donné les caractéristiques de la scène observée [W09]. Ce processus passe par deux étapes principales :

- Initialisation de l'arrière-plan, où un modèle initial de l'arrière-plan est calculé ;
- Mise-à-jour de l'arrière-plan, où ce modèle est mis à jour afin de l'adapter aux changements possibles dans la scène.

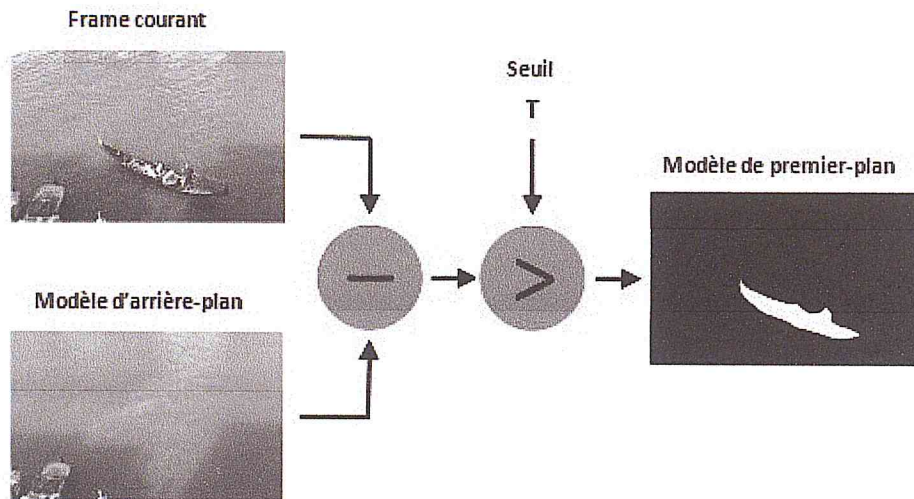


Fig IV.3 Processus de soustraction de fond

MOG et MOG2 font partie des algorithmes les plus connus et les plus utilisés pour effectuer une soustraction de fond et générer un masque de premier-plan. Ces deux algorithmes implémentent un objet qui va mémoriser n frames afin de réaliser un modèle de l'évolution des pixels au fil du temps. Ce modèle va être analysé afin d'en extraire les populations de pixels cohérentes entre-elles, ce qui va permettre de dégager le fond (background) de l'avant-plan (foreground). Nous expliquons tout de suite le principe de fonctionnement de chacun des deux algorithmes ainsi que leur différence.

❖ **Mixture Of Gaussian MOG :**

Cet algorithme a été introduit dans le document [B25]. Il se base sur la modélisation de chaque pixel d'arrière-plan par un mélange de distributions gaussiennes K ($K = 3$ à 5). Les poids du mélange représentent les proportions du temps que ces pixels restent dans la scène. Les probables pixels d'arrière-plan sont ceux qui restent le plus longtemps statique.

Pour utiliser cette méthode dans notre approche, nous devons passer par 2 étapes :

1- **Initialisation de l'arrière-plan:**

L'initialisation de l'arrière-plan se fait à travers les paramètres suivants :

- **history** : nombre de frame à utiliser pour le modèle de fond. Plus on utilise d'image, et plus la soustraction sera lente.
- **nmixture** : Le nombre K de "Mixture Of Gaussian" à utiliser, c'est à dire le nombre population de pixels à utiliser.

- *backgroundRatio* : un seuil définissant si un composant est assez significatif pour faire partie de l'arrière-plan.
- *noiseSigma* : paramètre fixant la gestion du bruit.

2- Génération du masque avant-plan et mise-à-jour de l'arrière-plan:

La mise-à-jour d'arrière-plan se fait à travers le paramètre *learningRate*. La mise à zéro de ce paramètre désactive la mise-à-jour de l'arrière-plan et le rend statique (ne change pas) tout au long du traitement des frames de la vidéo. Pour l'activer, on doit mettre ce paramètre à « -1 » et le mode devient dynamique, c'est-à-dire que chaque frame de la vidéo sera utilisé à la fois pour le calcul du masque de l'avant-plan et aussi pour la mise-à-jour de l'arrière-plan.

Ainsi, pour générer le masque d'avant-plan, nous aurons besoin des paramètres suivants :

- *InputArray* : Le prochain frame de la vidéo.
- *OutputArray* : C'est le masque de l'avant-plan renvoyé par la fonction, c'est une image binaire où les pixels considérés comme appartenant à l'avant-plan seront en blanc, le reste en noir.
- *LearningRate* : fixe le mode de fonctionnement de l'algorithme. On utilise 0 pour le mode statique ou - 1 pour une actualisation automatique de l'arrière-plan.

❖ Mixture Of Gaussian MOG2 :

Cet algorithme est une amélioration du précédent MOG, il a été présenté par Z.Zivkovic dans deux documents [B26] et [B27]. La principale caractéristique de cet algorithme c'est qu'il sélectionne le nombre approprié de distribution gaussienne K pour chaque pixel (dans MOG, K est fixe par contre ici il change pour chaque pixel), ce qui offre une meilleure adaptabilité aux variations scènes dues aux des changements d'éclairage, etc.

Pareil que précédemment, pour utiliser cette méthode dans notre approche, nous devons passer par 2 étapes :

1- Initialisation de l'arrière-plan:

En plus des paramètres que nous avons expliqués dans le MOG, MOG2 ajoute d'autres paramètres pour effectuer l'initialisation de l'arrière-plan:

- ***varThreshold*** : C'est un seuil utilisé pour le calcul de la distance de Mahalanobis (correspond à T_g) qui aide à décider quand un échantillon est proche des composants existants? S'il n'est près d'aucun composant alors un nouveau composant est généré. Une valeur T_g plus faible génère plusieurs composants. Une valeur T_g élevée peut entraîner un petit nombre de composants, mais ils peuvent devenir trop volumineux.
- ***bShadowDetection*** : Paramètre qui active (ou désactive) la détection de l'ombre.

2- Génération du masque avant-plan et mise à jour de l'arrière-plan:

Pareil que dans MOG, la mise-à-jour d'arrière-plan se fait à travers le paramètre *learningRate*. La valeur « -1 » implique une mise-à-jour du modèle d'arrière-plan pour chaque frame de la vidéo et la valeur « 0 » implique un modèle d'arrière-plan statique.

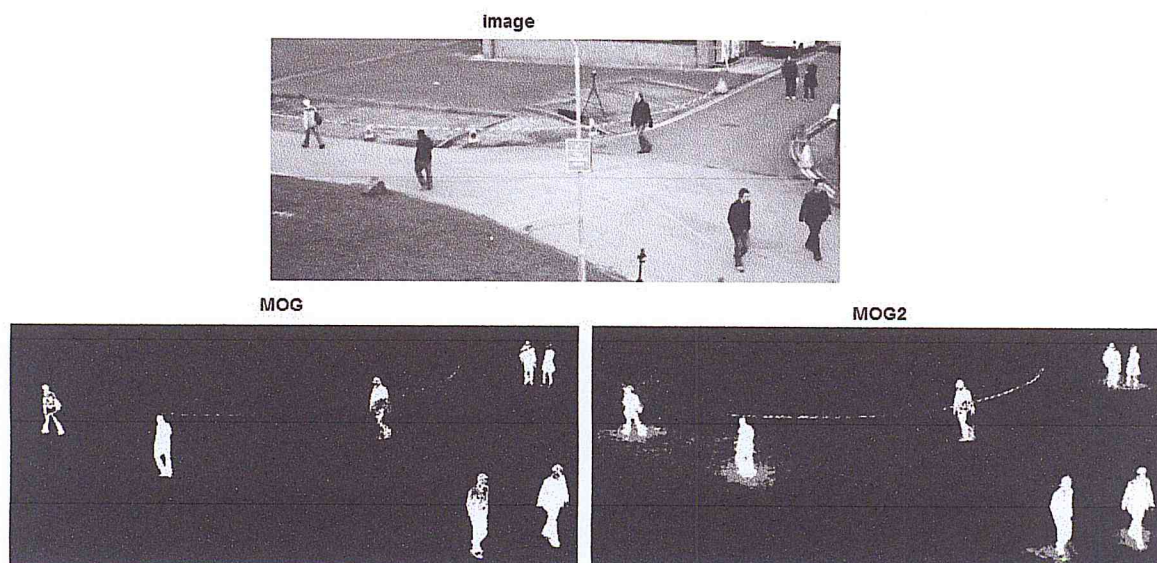


FIG IV.4 Exemple d'application du MOG et MOG2

IV. 2.1.3 Génération de blobs :

Le résultat final du processus de soustraction de fond est une image binaire appelé « masque du premier-plan », dans laquelle apparaissent uniquement les blobs correspondant aux éléments en mouvement. A ce stade, la prochaine étape est de vérifier que les blobs candidats présentent des caractéristiques spécifiques aux humains.

Nous décrivons dans ce qui suit les caractéristiques choisies et les procédures relatives à leur extraction.

CHAPITRE IV : MISE EN ŒUVRE DU SYSTÈME DE COMPTAGE

Correction de blobs :

Une fois le masque premier-plan généré, on rencontre souvent des cas de bruit et de déformation de bords. Pour cette raison, on a souvent recours aux opérations morphologiques mathématiques afin d'obtenir une forme lisse et générique des blobs. Ainsi, et pour un meilleur filtrage du bruit, nous allons appliquer deux opérations consécutives : l'**Ouverture** « open » en suite la **Fermeture** « close » : $Fermeture(Ouverture(image))$.

Ces deux opérations combinent entre deux opérations morphologiques basiques, à savoir l'**érosion** et la **dilatation**, cela en utilisant le même élément structurant¹. En effet, la fermeture c'est l'érosion de la dilatation (c-à-d une dilatation suivie d'une érosion), et l'ouverture c'est la dilatation de l'érosion (c-à-d une érosion suivie d'une dilatation).

Ce traitement est appliqué afin d'agréger les pixels de premier-plan en blob entiers, tout en comblant les trous isolés et les canaux étroits, en lissant les contours et en fusionnant les zones proches. Nous expliquons tout de suite le principe des opérations de l'érosion et la dilatation qu'on va appliquer sur notre masque premier-plan (une image binaire) :

a. La dilatation :

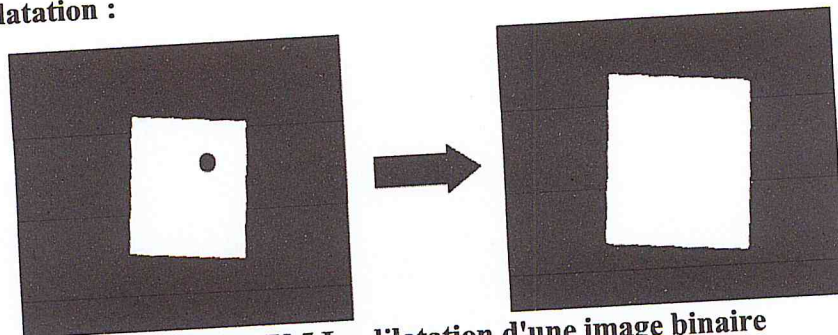


Fig IV.5 La dilatation d'une image binaire

La première opération morphologique utilisée est la dilatation. C'est une opération de base dont la principale caractéristique est de dilater les bords des différentes régions du premier-plan. Ainsi, ces régions du premier-plan deviendront plus grandes et les trous dans ces régions seront plus petits (certains disparaîtront complètement).

Dans les grandes lignes, tous les pixels de l'arrière-plan qui peuvent toucher les régions du premier-plan, en y mettant l'élément structurant, seront contenus dans l'objet dilaté.

¹ Un **élément structurant** est un masque binaire muni d'un point d'ancrage (généralement son centre). Il permet de prendre en compte le voisinage du pixel. Il est promené partout sur l'image binaire à traiter, et à chaque position on étudie sa relation avec cette image. Les éléments structurants les plus classiquement utilisés sont la croix et le carré.

Pour calculer une dilatation binaire, tous les pixels de l'arrière-plan doivent être traités. Par exemple, prenons un élément structurant carré de taille 3x3 pixels comme suit:

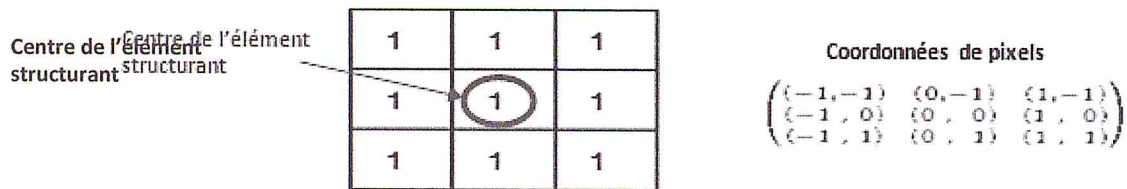


Fig IV.6 Élément structurant carrée 3x3

Pour chaque pixel de l'arrière-plan, on lui superpose l'élément structurant (le centre de l'élément structurant sur le pixel) et on teste si l'élément structurant est en contact avec au moins un pixel de premier-plan. Si c'est le cas, le pixel sera considéré comme premier-plan, dans le cas contraire, le pixel restera comme pixel d'arrière-plan. En d'autre terme, si l'un des pixels du voisinage (définis par l'élément structurant) est à 1, la valeur de sortie du pixel est alors 1.

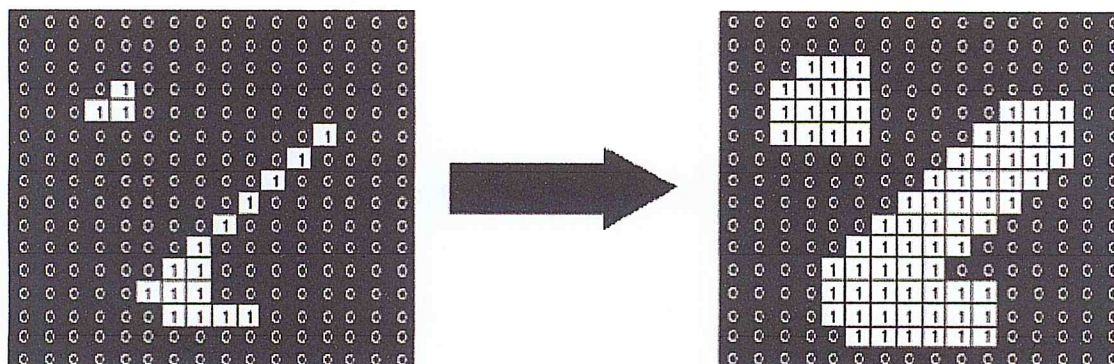


Fig IV.7 La dilatation d'une image binaire avec un élément structurant carré de 3x3 pixels

b. L'érosion :

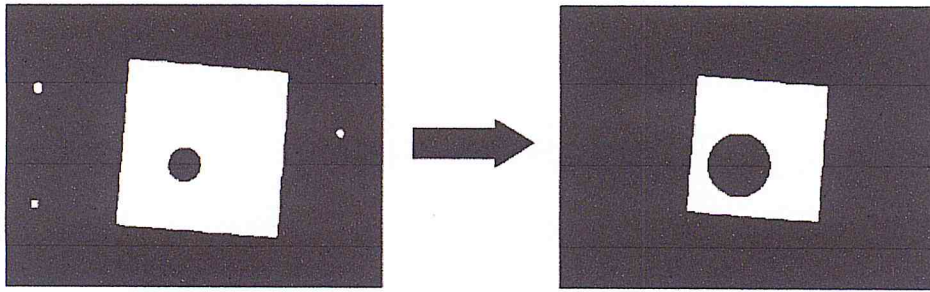


Fig IV.8 L'érosion d'une image binaire

Comme la dilatation, l'érosion est la deuxième opération de base appliquée dont la caractéristique principale est d'éroder les contours des différentes régions du premier-plan. Ainsi ces régions de premier-plan vont devenir plus petites (certains pixels isolés vont totalement disparaître) et les trous dans ces régions seront plus grands.

Dans les grandes lignes, tous les pixels des régions de premier-plan qui peuvent contenir totalement l'élément structurant seront contenus dans l'objet érodé.

Pour calculer une érosion binaire, tous les pixels de premier-plan doivent être traités. Par exemple, prenons un élément structurant carré de taille 3x3 pixels (voir Figure IV.6). Pour chaque pixel de premier-plan, on lui superpose l'élément structurant (le centre de l'élément structurant sur le pixel) et on teste s'il est complètement contenu dans le premier-plan. Si c'est le cas, le pixel traité sera contenu dans le premier-plan érodé, sinon, il sera considéré comme arrière-plan. En d'autre terme, si l'un des pixels du voisinage (définis par l'élément structurant) est à 0, la valeur de sortie du pixel est alors automatiquement 0.

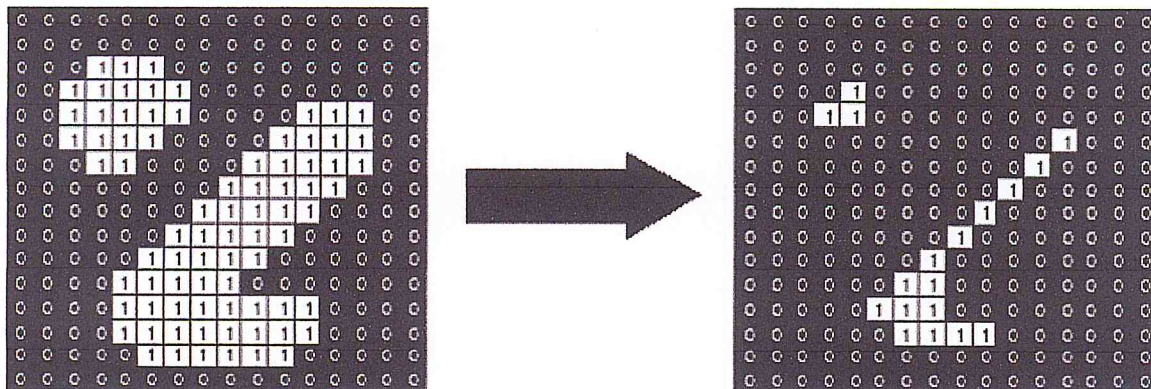


Fig IV.9 L'érosion d'une image binaire avec un élément structurant carré de 3x3

Étiquetage de blobs :

Une fois les blobs corrigés, on passe maintenant à l'opération suivante qui est l'**étiquetage** appelée aussi **analyse en composantes connexes**. Cette opération est basée sur le principe suivant : Partant d'une image binaire, on doit trouver les groupes de pixels connectés (les composantes connexes ou blobs) afin d'obtenir une image dans laquelle chaque blob est identifié de manière unique par une étiquette. L'image résultat est appelée **carte des régions**.

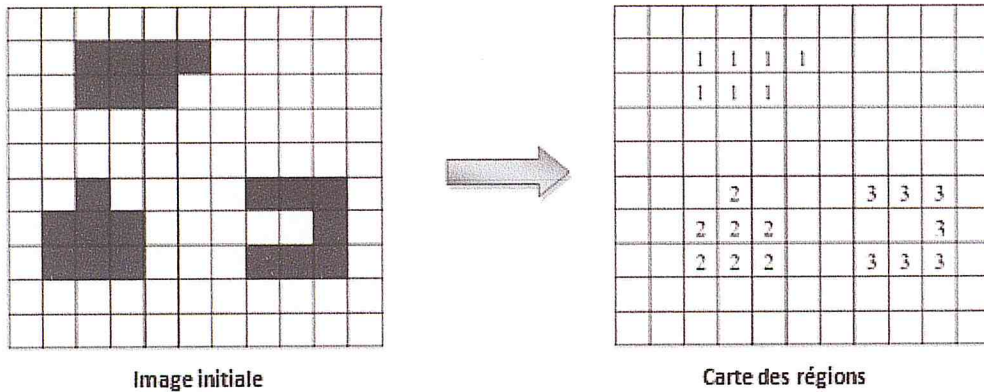


Fig IV.10 Exemple d'étiquetage de blobs

Le principe de ce processus est le suivant : Dans un balayage gauche droite et de haut en bas de l'image :

- On attribut une étiquette au pixel en fonction de ses voisins déjà étiquetés (à gauche et au dessus) ;
- Lorsque deux branches se connectent, on dit que leurs étiquettes sont équivalentes.

Voici l'algorithme qui permet d'effectuer cette tâche [B28]:

```
numéro <- 0;
pour tous les pixels P de gauche à droite et de haut en bas :
  si P est différent du fond
    alors si seulement un des deux voisins sup et gauche a une étiquette e
      alors marquer P avec e ;
    sinon si ces deux voisins ont la même étiquette e
      alors marquer P avec e ;
    sinon si ces deux voisins ont deux étiquettes
      différentes e1 et e2
      alors marquer P avec min(e1, e2) ;
      mettre (e1, e2) dans une table d'équivalence ;
    sinon numéro <- numéro+1;
    marquer P avec numéro;
  fin si
fin pour
prendre le plus petit élément comme représentant de chaque classe;
```

Sélection des blobs candidats :

La prise en compte des conditions opérationnelles permet d'éliminer d'emblée certains blobs ne respectant pas un ensemble de critères simples, tels que la taille que peut avoir un blob correspondant à une personne filmée sous certaines conditions. Nous appellerons cette procédure d'élimination la « sélection des blobs ». Pour cela, nous utilisons des informations sur les modalités de prise de vue de la séquence vidéo sur laquelle sont appliqués nos différents algorithmes. Nous pouvons ainsi poser des conditions sur la taille que doit avoir un blob correspondant à une personne.

La forme humaine détectée par une caméra fixée à une hauteur de presque 2 mètres sera vu comme la Figure IV.7, Cette forme a certains aspects bien spécifiques à elle tel que la taille, cela en raison de la distance fixe entre la caméra et le sol.

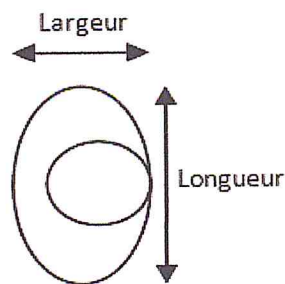


Fig IV.11 Forme humaine perçue par une caméra installée en hauteur

CHAPITRE IV : MISE EN ŒUVRE DU SYSTEME DE COMPTAGE

La taille d'une personne détectée par notre caméra va être un paramètre d'entrée à notre algorithme, elle sera calculée après surveillance de la surface couverte par les personnes passant dans le champ filmé. Ainsi, le Min et le Max de la taille d'une personne doivent être calculés initialement.

Subséquentement, nous devons filtrer notre ensemble de blobs en ne gardant que les blobs dont la superficie correspond à une taille humaine.

IV.2.2 Processus de suivi de personnes

Après avoir procédé comme décrit précédemment, nous sommes censés avoir un ou plusieurs blobs. Pour pouvoir suivre chacun de ses blobs dans la séquence vidéo, on propose que chaque blob doive être sauvegardé dans une structure qui comprend les paramètres requis suivants :

- L'identifiant de chaque blob, celui-ci doit rester stable et interchangeable tout au long du passage de la personne dans la vidéo.
- Les coordonnées x et y du centre de chaque blob appelé « centroid ».
- La superficie de chaque blob.

Pour vérifier le résultat de notre suivi et le rendre plus lisible, nous allons entourer chaque personne par un cercle correspondant à sa boîte englobante. Ce cercle devrait suivre cette personne tout au long de son passage dans la séquence vidéo.

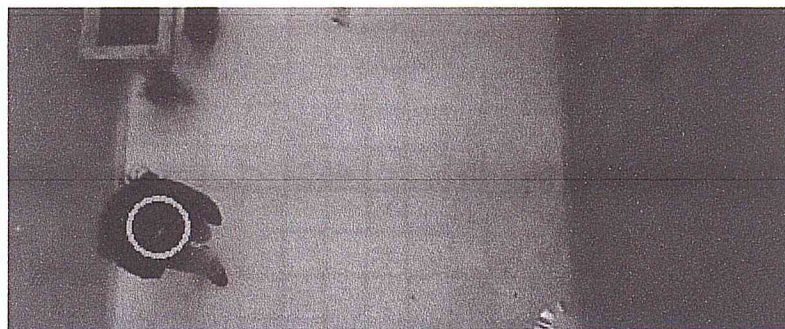


Fig IV.12 Résultat de suivi d'une personne

IV.2.3 Processus de comptage de personnes

Nous allons effectuer deux manières de comptage, la première pour compter le nombre de personnes entrantes et sortantes de l'endroit surveillé, et la deuxième pour donner le nombre de personnes présente dans la scène filmée.

IV. 2.3.1 Comptage de personnes entrantes et sortantes de l'endroit surveillé

Le comptage de personnes est un phénomène dynamique dans lequel les gens entrent d'une direction et quittent l'endroit de l'autre côté. Il existe différentes possibilités ou deux personnes ou plus peuvent entrer à la fois. Ils peuvent aussi se déplacer dans le sens identique ou opposé. Pour ces différents cas, l'idée de base de notre algorithme de comptage est l'utilisation de deux lignes de comptage, une pour indiquer les personnes entrantes et l'autre pour les sortantes, de telle sorte que le comptage d'une direction (IN ou OUT) ne s'incrémente que si la personne passe par les deux lignes pour confirmer qu'elle vient d'une direction et la quitte pour passer à l'autre direction.

Pour cela, nous avons tracé deux lignes, l'une de couleur bleu et l'autre de couleur rouge, indiquant respectivement les lignes d'entrée/sortie. Il est à noter que s'il ya besoin, nous pouvons échanger le statut de la ligne d'entrée et celle de sortie.

Ces deux lignes vont servir à définir le sens de passage des personnes et ainsi d'incrémenter leur nombre de la façon suivante:

- 1- Extraction de la couleur du centroid de chaque blob suivi (personne) ;
- 2- Si la personne passe de la ligne bleu à la ligne rouge, alors son centroid va prendre successivement la couleur bleu ensuite la couleur rouge, le programme incrémente alors la variable IN.
- 3- Si la personne passe de la ligne rouge à la ligne bleu, alors son centroid va prendre successivement la couleur rouge ensuite la couleur bleu, le programme incrémente alors la variable OUT.
- 4- Dans le cas ou une personne passe juste par la ligne bleu (idem pour la ligne rouge) sans pour autant passer par l'autre ligne, il n'y aura pas d'incrémentation.

CHAPITRE IV : MISE EN ŒUVRE DU SYSTEME DE COMPTAGE

Pour réaliser ceci, nous allons utiliser une structure nommé « **personne** » déclaré comme suit :

```
typedef struct {  
    int idPrs; // l'identifiant de chaque personne  
    string direction; // peut prendre comme valeur: "", "R", "B", "RB", "BR"  
    bool ligneRouge; // true or false  
    bool ligneBleu; // true or false  
}personne;
```

Le diagramme d'activité suivant résume le principe de fonctionnement de notre algorithme de comptage:

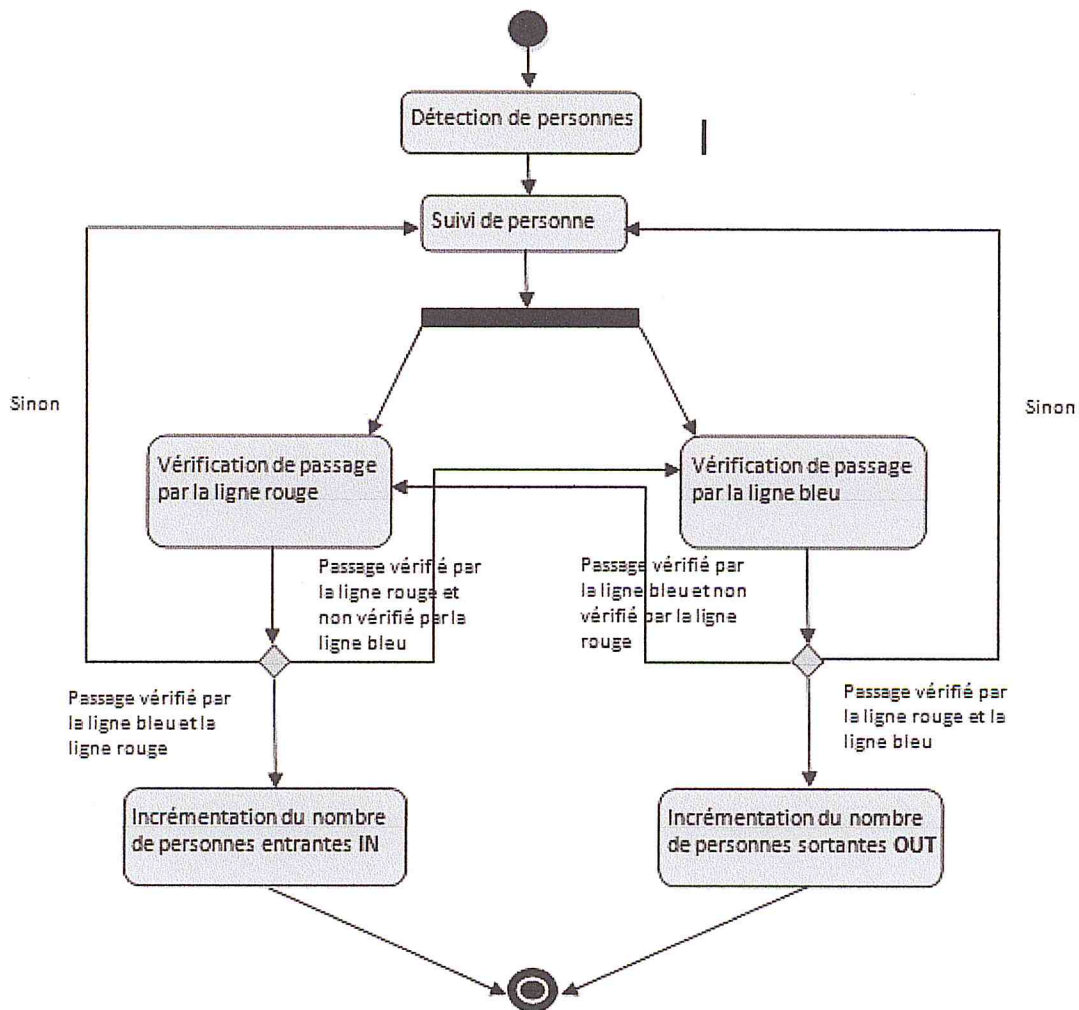


Fig IV.13 Diagramme d'activité de l'algorithme de comptage

Il existe des situations où on peut observer un blob géant représentant deux personnes ou plus (c'est dans le cas où deux personnes ou plus rentrent dans la même direction en étant très proches les unes aux autres), dans ce cas de figure, notre algorithme de comptage va donner des résultats erronés. Pour remédier à ce problème, nous allons ajouter une condition sur la surface de chaque blob avant d'effectuer l'incrémentation, cette surface va être comparé à la taille de blob correspondant à une personne (nous avons stipulé dans la section IV.2.1.3 que la taille d'une personne détectée par notre caméra est un paramètre d'entrée à notre algorithme) et ainsi ajouter au compteur le nombre adéquat.

IV. 2.3.2 Comptage de personnes présentes dans la scène

En plus de l'information donné par notre algorithme de comptage expliqué précédemment, ça serai utile de connaitre aussi le nombre de personnes présentes dans la scène à tout moment, sans pour autant que ces personnes rentrent ou pas dans l'endroit que nous surveillons (films). Cette information peut parfois être un élément déclencheur de l'attention de l'opérateur dans le cas d'incidents majeurs tel que vol, incendie, prise d'otage, etc.

Comme les blobs détectés vont être sauvegardé dans une structure bien spécifique, nous pouvons alors déterminer le nombre de blobs détecté à chaque frame, ce nombre sera exploité pour déterminer le nombre de personnes présentes dans la scène. Et de la même manière que précédemment, nous allons ajouter une condition sur la surface de chaque blob avant d'effectuer le comptage.

IV. 3 CONCLUSION

Dans ce chapitre, nous avons fourni une vision détaillée de la conception de notre système de détection et comptage de personnes dans une vidéo numérique, cela en décrivant les sous-systèmes qui le composent, à savoir, le sous-système de détection de personnes, le sous-système de suivi de personnes et le sous-système de comptage de personnes.

Dans le prochain chapitre, nous allons présenter les résultats obtenus suite à l'implémentation de notre programme, et aussi analyser la fiabilité de ses résultats en effectuant des tests réels.

CHAPITRE
V

**REALISATION ET
EXPERIMENTATION**

CHAPITRE V : REALISATION ET EXPERIMENTATION

V.1 INTRODUCTION

Dans ce présent chapitre nous allons d'abord présenter notre environnement de travail en passant par les outils matériels et logiciels utilisés, ensuite nous allons fournir une vision détaillée de l'implémentation de notre système de détection, suivi et comptage de personnes, et enfin nous allons analyser les résultats obtenus suite aux tests réalisés sur une vidéo numérique.

V.2 ENVIRONNEMENT DE TRAVAIL

Avant de fournir une description détaillée de notre système de comptage, nous présentons d'abord les outils matériels et logiciels retenus pour sa mise en œuvre.

V.2.1 Matériels utilisés

Notre système se compose principalement d'une caméra vidéo numérique utilisée pour filmer des séquences vidéo, et d'un serveur pour effectuer les traitements nécessaires sur ces séquences.

a. Caméra

Pour enregistrer la vidéo sur laquelle nous avons testé notre algorithme, nous avons utilisé une Webcam caméra, avec une résolution de 1600 x 1200 pixels. Notre caméra est dotée d'une connexion USB.



Fig V.1 Caméra utilisée

b. Serveur

Nous avons utilisé un ordinateur portable *DELL* pour jouer le rôle du serveur sur lequel les séquences vidéos filmées par caméra seront transmises et seront sauvegardées et traitées.

Notre serveur a les caractéristiques suivantes :

- Environnement (Système d'exploitation) : Windows 7, 32 bits.
- Processeur : Intel(R) Core(TM) 2 Duo CPU T6670 @ 2.20GHz 2.20GHz
- RAM : 4,00 GO

V.2.2 Logiciels utilisés

a. Langage de programmation :

Pour implémenter notre programme nous avons utilisé C++ comme langage de programmation.

b. Bibliothèque OpenCV :

Comme notre projet requiert des fonctions de traitement d'images et de détection et suivi d'objet, nous n'avons pas trouvé de meilleur choix que d'utiliser la bibliothèque **OpenCV** (Open Source Computer Vision) qui est une bibliothèque graphique libre « open source », initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. Cette bibliothèque est écrite en C et C++ et peut être utilisée sous Linux, Windows et Mac OS X. Des interfaces ont été développées pour Python, Ruby, Matlab et autre langage.

Un des buts d'OpenCV est d'aider les gens à construire rapidement des applications sophistiquées de vision à l'aide d'infrastructure simple de vision par ordinateur. La bibliothèque d'OpenCV contient près de 500 fonctions permettant:

- La manipulation d'images (chargement, sauvegarde, copie, conversion...);
- Le traitement d'images (filtrage, redimensionnement, extraction, morphologie mathématique...);
- L'analyse d'images (composantes connexes, ajustement de primitives...);
- La manipulation et acquisition de vidéos (Utilisation directe des images prises par caméra ou précisément une image extraite d'une vidéo);
- La vision (calibration de caméra, stéréovision, recherche d'association...);
- Les manipulations de matrices et algèbre linéaire.

OpenCV est constitué des principaux modules suivants :

- ❖ **CV et CVAUX** : Assure les fonctions de traitement d'images, à savoir :
 - Gradient, contours et contours actifs ;
 - Opérations Morphologiques (érosion, dilatation, fermeture, ouverture...)
 - Filtrages divers (lissage, rehaussement de contraste, suppression de fond...)
 - Conversion d'espace couleur (RGB, HSV, YCbCr...)
 - Étiquetage, manipulation de contours ;
 - Transformations diverses (Fourier, Hough, ...)
 - Histogrammes ;
 - Analyse de mouvement et suivi ;
 - Reconstruction (Calibration, mise en correspondance...)
 - Détection et reconnaissance de formes.
- ❖ **CXCORE** : Assure la définition des structures élémentaires de données (matrices, tableaux, listes, files, graphes, arbres...) ainsi que tous les opérateurs standards (arithmétiques et logiques) indispensables au traitement de ces structures. Ce module assure aussi de nombreuses fonctions de dessin :
 - Lignes, cercles, ellipses, arcs, etc.
 - Polygone plein ou contours.
 - Textes (avec différentes fonts).
 - Trousse à outils pour gérer les couleurs, les tailles, etc.
- ❖ **HIGHGUI** : Assure les fonctions responsables de la manipulation des images et des vidéo (lecture, écriture,...) et gère aussi les interfaces utilisateurs comme les fenêtres GUI et les entrées/sorties (clavier, souris,...).

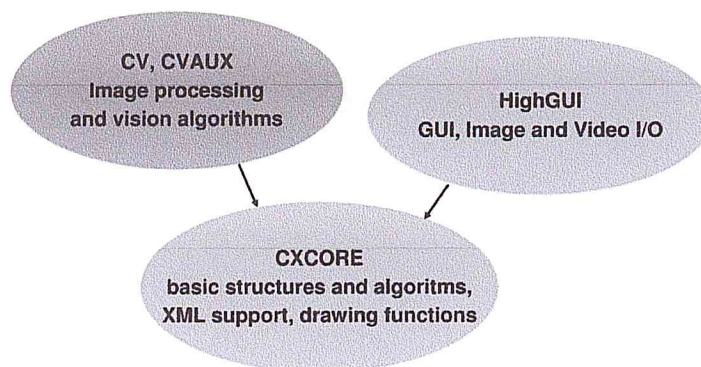


Fig V.2 Principaux modules d'OpenCV

c. Outils installés

Pour pouvoir exploiter correctement les fonctions d'OpenCV, nous avons utilisé les outils suivants (Pour voir les étapes d'installation détaillées, se référer à l'annexe : Installation et configuration d'OpenCV) :

- **CodeBlocks-16.01** : c'est un EDI (Environnement de Développement Intégré) C, C++ et Fortran gratuit, conçu pour être très extensible et entièrement configurable. La dernière version peut être téléchargée à partir du lien suivant :
« <http://www.codeblocks.org/downloads/26> »
- **Opencv-2.4.12** : La dernière version stable est disponible sur le lien officiel:
<http://opencv.org/downloads.html>
- **Cmake-3.5.0** : Afin d'utiliser les différentes bibliothèques open sources dans un projet, il est nécessaire de les configurer par rapport à son EDI. CMake est justement un système de compilation et de construction de projets multiplateforme et Open Source et disponible sur son site officiel : « <https://cmake.org/download/> »
- **Mingw-w64 version 5.3.0** : MinGW "*Minimalist Gui for Windows*" offre des compilateurs gcc, C, C++, ADA et Fortran, et aussi des utilitaires GNU pour assembler, lier des bibliothèques (DLLs), gérer les archives (assembler, linker, archive manager).
La dernière version peut être téléchargée à partir du lien suivant :
« <https://sourceforge.net/projects/mingw-w64> »
- **CvBlobsLib** : C'est une bibliothèque basée sur l'API¹ d'OpenCV, également *open source* et multiplateformes, développée en C++, qu'on doit l'intégrer au projet. Elle offre une API de haut niveau pour la détection des objets « blobs ». La dernière version stable est disponible sur le lien officiel:
« <https://code.google.com/archive/p/cvblob/downloads> »

¹ **API**: une interface de programmation applicative (en anglais « *Application Programming Interface* ») est un ensemble normalisé de classes, de méthodes ou de fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels.

V.3 DESCRIPTION DE L'APPLICATION

Pour tester notre programme, nous avons enregistré une séquence vidéo au format «.avi ». Cette séquence a une durée de 4 min. La caméra utilisée a été fixée à une hauteur de 2 mètres au-dessus d'une porte d'un immeuble permettant de prendre tout le champ visuel de son entrée. Les conditions d'éclairage étaient relativement stables et satisfaisantes.

V.3.1 Système de détection de personnes

Nous allons présenter dans cette partie les étapes de traitements que nous avons appliqué aux séquences d'images (frames) de notre vidéo afin de détecter et d'isoler les individus présents dessus

V.3.1.1 Capture et traitement :

Chargement de l'image : Les frames de notre séquence vidéo ont été acheminés un par un vers notre programme et reçus dans une structure de type **Mat**, en utilisant les commandes suivantes :

```
VideoCapture cap("NotreVideo.avi"); // ouvrir notre vidéo

Mat img; // declaration d'une variable de type Mat
for(;;)
{
    cap >> img; // chargement de l'image vers la variable img
    .....// suite du programme
}
```

Sous OpenCv, Une image peut être mémorisée à l'intérieur d'une structure du type **Cvmat** ou **Iplimage**. Ces structures sont issues des versions 1.x d'Opencv. La structure **Iplimage** est un vieux format d'image compatible intel IPP. Depuis la version 2.1 d'OpenCV l'accent a été mis sur les matrices et les opérations sur celles-ci. En effet, la structure de base est la matrice. Une image peut être considérée comme une matrice de pixel. Ainsi, toutes les opérations de bases des matrices sont disponibles. La classe **Mat** est un nouveau format qui est né depuis la version 2.x d'OpenCV, c'est la version C++ de **Cvmat** [B29].

Redimensionnement de l'image : Après chargement de l'image, on va redimensionner sa taille en 300 x 200 pixels. Ceci est fait pour diminuer le volume des calculs et le temps de traitement de l'image, tout en veillant à ne pas trop perdre de la qualité de l'image.

```
Size taille(300,200) ;  
resize(imgSource,imgDest, taille); // redimensionnement de l'image
```

Conversion et filtrage de l'image : Afin d'alléger les traitements sur chaque image, nous allons convertir chacune d'entre elle de l'espace couleur RGB vers l'espace niveaux de gris.

```
// Conversion en niveaux de gris  
cvCvtColor(IplImg* imgSource, IplImg* imgDest, CV_BGR2GRAY);
```

Afin d'atténuer le bruit et estomper l'image nous avons effectué un filtre médian. Pour réaliser cela, nous avons utilisé la fonction *medianBlur* d'OpenCv.

```
// filtre médian  
void medianBlur( InputArray src, OutputArray dst, int ksize );
```

Ksize : La taille de l'élément structurant utilisé (Ksize x Ksize), elle doit être impair et supérieur à 1 (3,5,7,...)

V.3.1.2 Soustraction de fond BS:

Afin d'effectuer une soustraction d'arrière-plan et de générer un masque d'avant-plan sous OpenCv, en utilisant un des deux algorithmes MOG ou MOG2, nous allons utiliser les fonctions suivantes : `BackgroundSubtractorMOG()` et `BackgroundSubtractorMOG2()` ensuite nous allons choisir la meilleure d'entre elles. Ces fonctions implémentent un objet natif avancé de la librairie OpenCV qui permet de réaliser la soustraction du fond au sein d'une image issue d'un flux vidéo en se basant sur les "*Mixture Of Gaussian*" (MOG).

Toute la puissance de ces fonctions réside dans le fait de n'utiliser qu'une ou 2 lignes dans un programme pour implémenter un algorithme complexe.

Pour exécuter ces fonctions, nous devons d'abord déclarer l'objet qui va contenir le masque d'avant-plan :

```
Mat fgMaskMOG; // Le masque d'avant-plan généré par MOG  
Mat fgMaskMOG2; // Le masque d'avant-plan généré par MOG2
```

❖ **Mixture Of Gaussian MOG** :

Pour utiliser cette méthode dans notre programme, nous devons passer par 2 étapes :

1- Initialisation de l'arrière-plan:

Premièrement, nous créons l'objet Background qu'on appelle *bgs* en utilisant la fonction :

```
CV_ :BackgroundSubtractorMOG() bgs
```

Cet objet a quelques paramètres optionnels (nous les avons expliqués dans le chapitre précédent section IV.2.1.2) qui sont déclarés dans la classe « CV_EXPORTS_W BackgroundSubtractorMOG » comme éléments **protégés** (*protected*), ce qui veut dire qu'on ne peut pas les modifier et que les valeurs restent par défaut dans ce cas d'utilisation :

- *history* : La valeur par défaut est 200 frames, à utiliser pour la génération du modèle d'arrière-plan.
- *nmixture* : La valeur par défaut du nombre K de "Mixture Of Gaussian" est 5.
- *backgroundRatio* : La valeur par défaut est 0.7 .
- *noiseSigma* : La valeur par défaut est 15.

2- Génération du masque avant-plan et mise à jour de l'arrière-plan:

Nous exécutons maintenant l'algorithme MOG en utilisant la commande suivante :

```
bgs : : operator ( ) ( InputArray image, OutputArray fgMaskMOG,
double learningRate =0)
```

Les paramètres :

- *InputArray* : Le prochain frame de la vidéo.
- *OutputArray* : Le masque de l'avant-plan renvoyé par la fonction,
- *LearningRate* : fixe le mode de fonctionnement de l'algorithme. On utilise 0 pour le mode statique ou - 1 pour une actualisation automatique de l'arrière-plan.

Comme nous l'avons expliqué dans le chapitre précédent, la mise-à-jour d'arrière-plan se fait à travers le paramètre **learningRate**. Dans l'algorithme MOG, ce paramètre est égale à zéro « 0 » par défaut, ce qui veut dire que le modèle d'arrière-plan généré au début du traitement restera statique (ne change pas) tout au long du traitement des frames de la vidéo. Toutefois, pour activer la mise-à-jour de l'arrière-plan, on n'a qu'à mettre ce paramètre à « -1 » et le mode devient dynamique, c'est-à-dire que chaque frame de la vidéo sera utilisé à la fois pour le calcul du masque de l'avant-plan et aussi pour la mise-à-jour de l'arrière-plan.

Remarque : lorsqu'on met ce paramètre à « -1 », les objets de l'avant-plan qui restent immobiles vont peu à peu être inclus dans l'arrière-plan.

❖ Mixture Of Gaussian MOG2 :

Pareil que précédemment, pour utiliser cette méthode dans notre programme, nous devons passer par 2 étapes :

1- Initialisation de l'arrière-plan:

Premièrement, nous créons l'objet Background qu'on appelle *bgs2* en utilisant la fonction :

```
cv : : BackgroundSubtractorMOG2() bgs2
```

Cet objet a quelques paramètres optionnels (nous les avons expliqués dans le chapitre précédent section IV.2.1.2) qui sont déclarés dans la classe « CV_EXPORTS_W BackgroundSubtractorMOG2 » comme éléments **public** (*public*), on peut donc modifier leurs valeurs selon nos besoins.

- **history** : nous l'avons mis à 300 frames (200 par défaut).
- **varThreshold** : Par défaut la valeur est 3 sigma => $T_g = 3 * 3 = 9$. Nous l'avons mis à 10.
- **bShadowDetection** : La valeur par défaut est *true*.

2- Génération du masque avant-plan et mise à jour de l'arrière-plan:

Nous exécutons maintenant l'algorithme MOG en utilisant la commande suivante :

```
Bgs2 : : operator () (InputArray image, OutputArray fgMaskMOG2,  
double learningRate = -1)
```

Pareil que dans MOG, la mise-à-jour d'arrière-plan se fait à travers le paramètre **learningRate**, la différence c'est que dans MOG2 ce paramètre est par défaut égale à « -1 » ce qui implique une mise-à-jour du modèle d'arrière-plan pour chaque frame de la vidéo.

Dans notre projet, nous avons décidé d'implémenter la méthode **MOG2** car il nous donne la possibilité de modifier les paramètres selon nos besoins. Aussi, son rendement était meilleur que celui du **MOG**.



Fig V.3 Génération du masque d'avant-plan avec MOG2

V.3.1.3 Génération de blobs :

Après génération du masque de premier-plan, nous devons maintenant vérifier que les blobs résultants correspondent bien aux personnes en mouvement, pour cela nous devons d'abord corriger ces blobs en éliminant le bruit, puis les étiqueter et enfin les filtrer en ne gardant que les blobs dont la taille correspond à une personne filmée sous nos conditions.

Correction de blobs :

Dans le but d'éliminer les cas de bruit et agréger les pixels de premier-plan en blob entiers ayant une forme lisse et générique, nous allons appliquer à notre masque du premier-plan les opérations morphologiques mathématiques suivantes : une **ouverture** « *OPEN* » suivie d'une **fermeture** « *CLOSE* ». Afin d'y parvenir, nous avons eu recours à la fonction d'OpenCv « **morphologyEx** ».

Pour pouvoir utiliser cette fonction, nous devons d'abord définir les paramètres spécifiques à l'élément structurant utilisé (appelé *Kernel*), à savoir :

- Sa forme morphologique : nous avons choisis la forme rectangulaire ;
- Sa taille : pour définir le voisinage utilisé. Dans notre cas nous avons utilisé une taille de 5 x 5 pixels ;
- Son point d'ancrage : pour définir la position de son point d'ancrage. Les valeurs négatives veulent dire que le point d'ancrage correspond au centre du *Kernel*.

```
Mat Kernel = getStructuringElement(MORPH_RECT, Size(5, 5), Point(-1,-1) );  
morphologyEx(imgSource, imgDest, CV_MOP_OPEN, Kernel)  
morphologyEx(imgSource, imgDest, CV_MOP_CLOSE, Kernel);
```


Etiquetage de blobs et extraction de leurs caractéristiques :

Comme nous l'avons expliqué dans le chapitre précédent, l'opération de l'étiquetage à pour but d'identifier chaque blob détecté par une étiquette unique.

Sous OpenCv, la librairie *CvBlobs.h* nous offre la possibilité d'exploiter cette fonctionnalité en identifiant chaque blob par un *label* et lui attribuant une structure de type *CvBlobs* qui comprend les caractéristiques suivantes :

- L'identifiant de chaque blob appelé *label* ;
- Les coordonnées x et y du centre de chaque blob appelé « *centroid* » ;
- La superficie de chaque blobs appelé « *area* ».

Ce travail est réalisé grâce à la fonction *CvLabel* qu'on va appliquer sur le masque d'avant-plan comme suit:

```
CvBlobs blobs; // declaration de la structure Cvblobs
unsigned int cvLabel (IplImg* imgSource, IplImg* imgDest, CvBlobs &blobs);
```

Sélection des blobs candidats :

Pour filtrer nos blobs et ne garder que ceux correspondant à une taille humaine telle que perçue par notre caméra (selon nos modalités de prise de vue), nous utilisons la fonction *cvFilterByArea* pour supprimer tous les blobs qui ne répondent pas à l'intervalle de superficie précisé dans ses paramètres :

```
cvFilterByArea (CvBlobs &blobs, Int MinArea, Int MaxArea);
```

V.3.2 Système de suivi de personnes

Après avoir procédé comme décrit précédemment, nous somme censé avoir un ou plusieurs blobs. Le paramètre requis pour pouvoir suivre chacun de ces blobs est son identifiant, celui-ci doit rester stable et interchangeable tout au long du passage de la personne dans la vidéo.

1- Détermination de l'identifiant de chaque blob :

Lorsque nous avons utilisé l'identifiant *Label* de *cvBlobs* décrit précédemment, nous avons remarqué que sa valeur n'est pas toujours stable pour la même personne, ce qui peut induire nos résultat de comptage en erreur. C'est pour cette raison que nous somme passés vers une autre structure de suivi, à savoir *cvTracks* (appartient toujours à la bibliothèque *cvblob.h*) dont l'identifiant appelé *ID* est plus stable que L'identifiant *Label* de *cvBlobs* comme le montre la figure suivante :

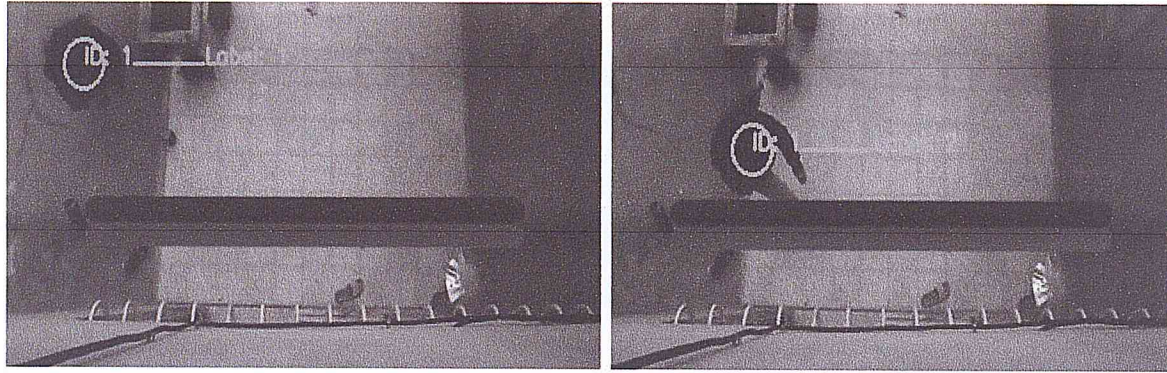


Fig V.4 Inconvénient du Label de cvBlobs

Afin d'y parvenir, nous déclarons la structure *tracks* comme suit :

```
CvTracks tracks; // declaration de la structure Cvtracks
```

Cette structure représente une liste d'éléments qu'on appelle *tracks* où chaque *track* correspond à un blob, comme le montre la structure de *CvTrack* suivante:

```
struct CvTrack
{
    CvID id; /// l'identifiant de chaque track.
    CvLabel label; /// Le label assigné au blob relié à ce track.
    unsigned int minx; /// coordonnée X min du track.
    unsigned int maxx; /// coordonnée X max du track.
    unsigned int miny; /// coordonnée Y min du track.
    unsigned int maxy; /// coordonnée y max du track..
    CvPoint2D64f centroid; ///Le point centre de chaque track.
    unsigned int lifetime; /// indique dans combien de frame l'objet reste en scène.
    unsigned int active; ///indique le nombre de frame où l'objet est actif à partir de la
        ///dernière période ou il était inactif.
    unsigned int inactive; /// indique nombre de frame où l'objet à été inactf.
};
```

Nous mettons à jour cette liste de tracks à travers la fonction suivante :

```
void cvUpdateTracks(CvBlobs const &b, CvTracks &t, const double thDistance,
const unsigned int thInactive, const unsigned int thActive=0);
```

Les paramètres :

- *b* : liste de blobs ;
- *t* : liste de tracks ;
- *thDistance* : la distance Max qui détermine quand un track va correspondre au blob ;
- *thInactive* : Le nombre Max de frame ou le track peut être inactive ;
- *thActive* : si un track devient inactive alors qu'il a été active pendant une période < *thActive* frames, alors le track va être éliminé.

2- Affichage du rendement du suivi:

Pour vérifier le résultat de notre suivi et le rendre plus lisible, nous avons affiché l'*ID* et la boîte englobante (*Bounding Box*) de chaque track. Pour cela, il existe une fonction prédéfinis « *cvRenderTracks* »:

```
void cvRenderTracks(CvTracks const tracks, IplImage *imgSource, IplImage *imgDest, unsigned short mode=0x000f);
```

Les paramètres :

- *tracks* : liste des tracks ;
- *imgSource*: L'image source;
- *imgDest*: L'image destination;
- *mode* : Le mode d'affichage. Par défaut il est égale à « CV_TRACK_RENDER_ID|CV_TRACK_RENDER_BOUNDING_BOX » c.-à-d. afficher l'*ID* et la boîte englobante ;

Toutefois, le rendement de cette fonction n'a pas été satisfaisant, car les boîtes dessinées laissent des traces figées un bon moment sur place, comme le montre la figure suivante :

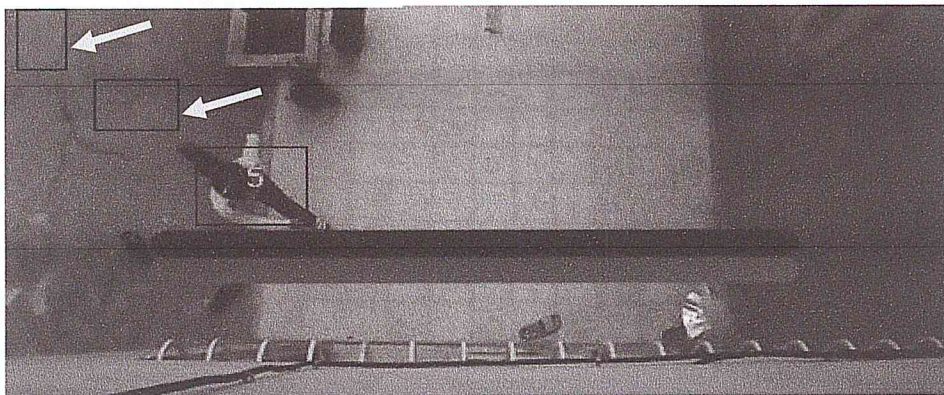


Fig V.5 Exemple d'utilisation de la fonction « *cvRenderTracks* »

Ce problème d'affichage nous a poussé à utiliser une autre façon pour afficher le suivi de chaque personne. Nous avons donc préféré dessiner un cercle autour du centroid de chaque blob pour encercler chaque personne. Le résultat d'affichage a été nettement meilleur que précédemment, comme le montre la figure suivante :

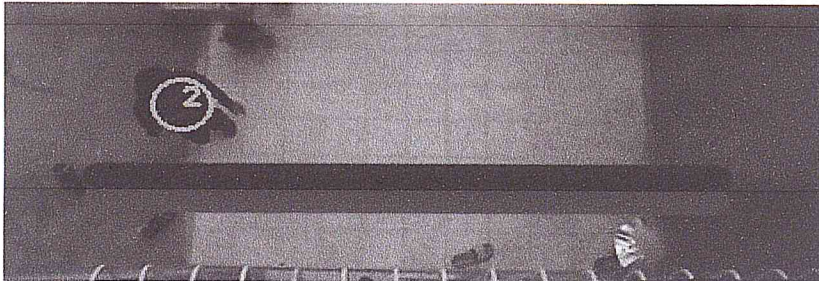


Fig V.6 Résultat de suivi d'une personne

V.3.3 Système de comptage de personnes

Comme nous l'avons mentionné dans le chapitre précédent (section IV.2.3), nous avons utilisé deux lignes de comptage, l'une de couleur bleu et l'autre de couleur rouge, indiquant respectivement les lignes d'entrée/sortie.

Dans le but d'améliorer l'efficacité et la précision de notre algorithme et rendre l'application plus filable et facile à manipuler, nous avons créé une interface qui permettra le réglage des paramètres des lignes de comptage de personnes comme suit :

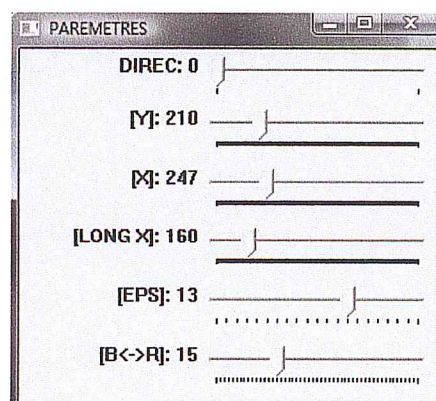


Figure V.7 Paramètres des lignes de comptage

Les paramètres sont :

- **DIREC** : Permet de spécifier la direction de l'entrée *IN* et de la sortie *OUT* des personnes, de telle sorte que si on inverse les couleurs des deux lignes de comptage, la direction s'inverse aussi.
- **[Y]** : Permet de modifier la position des lignes de comptage selon l'axe Y.

CHAPITRE V : REALISATION ET EXPERIMENTATION

- [X] : Permet de modifier la position des lignes de comptage selon l'axe X.
- [LONG X] : Permet de modifier la longueur des lignes de comptage selon la longueur de l'entrée surveillée.
- EPS : Permet de modifier l'épaisseur des deux lignes de comptage jusqu'à arriver à une épaisseur adéquate ou même si une personne va circuler a une grande vitesse pourra toucher successivement les deux lignes.
- [R<->B] : Permet de modifier la distance entre les deux lignes de comptage.

Afin de pouvoir analyser et vérifier la précision de notre application, nous avons décidé de garder un historique de nos résultats de comptage sur un fichier Excel dont lequel notre application va sauvegarder automatiquement les informations suivantes : Le nombre *IN*, Le nombre *OUT*, Le nombre de personnes présentes dans la scène et le temps du traitement.

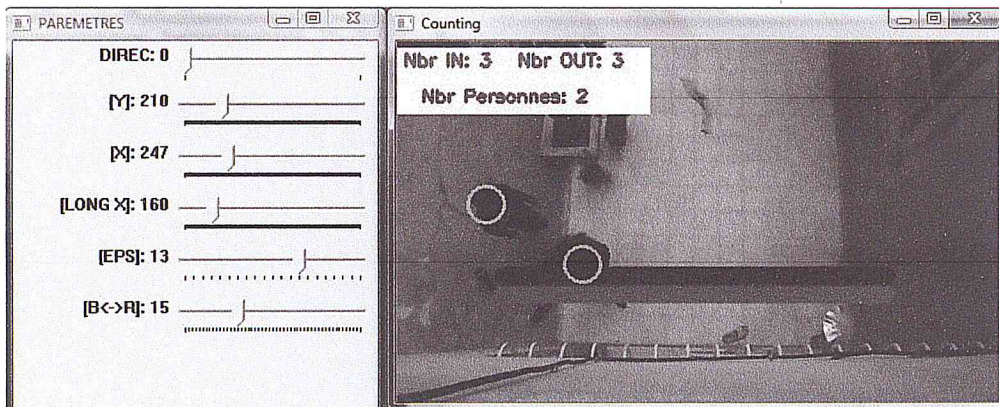


Fig V.8 Fenêtre d'affichage de l'application de comptage

The image shows a screenshot of a Microsoft Excel spreadsheet titled 'historique.xls'. The spreadsheet contains a table with the following data:

OUT	IN	NOMBRE DE	MILLISEC	DATE
0	1	1	1464213057	Wed May 25 23:50:57 2016
0	2	1	1464213059	Wed May 25 23:50:59 2016
1	2	1	1464213062	Wed May 25 23:51:02 2016
1	3	1	1464213063	Wed May 25 23:51:03 2016
1	4	2	1464213068	Wed May 25 23:51:08 2016
2	4	2	1464213073	Wed May 25 23:51:13 2016
3	4	1	1464213073	Wed May 25 23:51:13 2016
4	4	1	1464213078	Wed May 25 23:51:18 2016
5	4	1	1464213083	Wed May 25 23:51:23 2016
6	4	1	1464213085	Wed May 25 23:51:25 2016

Fig V.9 Sauvegarde des résultats de comptage dans un fichier Excel

V.4 ÉVALUATION DES PERFORMANCES

Ces dix dernières années, de nombreux systèmes de vidéo-surveillance ont été proposés dans la littérature. Chaque nouveau système proposé vise à augmenter la robustesse et la performance des résultats face aux nombreuses difficultés rencontrées en vision par ordinateur. L'évaluation des résultats devient impératif, particulièrement lorsque les algorithmes font face à des problèmes encore non résolus. D'un point de vue expérimental, la résolution d'un problème nécessite la validation par un protocole d'évaluation bien défini, permettant ainsi la comparaison avec d'autres algorithmes afin d'en identifier les faiblesses et les points forts.

Appliquer l'algorithme sur des séquences différentes donnera des performances différentes, il est donc nécessaire de bien choisir le jeu de vidéos de test en fonction de la tâche à analyser.

Ainsi, nous avons mené quatre expériences sur une vidéo enregistrée, afin de s'assurer de la cohérence de notre application et d'améliorer ses performances :

- Un test sur la résolution, en la faisant varier entre [50x50] px et [500x500] px, dans le but de chercher la valeur qui nous donne une meilleure précision ;
- Un test sur la position des deux lignes de comptage selon l'axe des « Y » ;
- Un test sur l'épaisseur des deux lignes de comptage ;
- Une comparaison entre quelques algorithmes de détection et aussi de suivi ;

En dernier, nous avons vérifié le bon fonctionnement de notre programme sur une vidéo filmée en temps réel.

V.4.1 Notations et définitions standards

Les définitions usuelles suivantes sont utilisées dans le calcul de nombreuses métriques

Vraie positif VP	Le système a détecté une situation réelle. La situation existe aussi bien dans le résultat de l'algorithme que dans la vérité terrain.
Vraie négatif VN	La situation n'existe ni dans l'algorithme, ni dans la vérité-terrain.
Faux négatif FN	Une situation réelle a été ratée par l'algorithme. La situation n'existe pas dans le résultat de l'algorithme tandis qu'elle existe dans la vérité-terrain.
Faux positif FP	Le système a détecté une situation qui n'est pas réelle. Cette situation existe dans le résultat de l'algorithme mais n'existe pas dans la vérité-terrain.

Tab V.1 Définitions standards : VP, VN, FN, FP.

CHAPITRE V : REALISATION ET EXPERIMENTATION

Ces mesures permettent de caractériser un algorithme vis à vis du nombre de réussites et d'échecs (VP, VN, FP, FN) face à un problème donné. Les définitions précédentes permettent d'estimer la précision comme suit:

$$Précision = \frac{VP}{VP + FP}$$

V.4 .2 Test sur la résolution de la vidéo :

Dans ce tout premier scénario de test, notre objectif est d'améliorer les performances de notre système en termes de consommation d'énergie. Nous avons enregistré une vidéo de test sur un échantillon de soixante personnes. Les circonstances de cette expérience sont les suivantes :

- Emplacement de la camera à deux mètres au-dessus de la porte d'entrée de l'immeuble ;
- Les conditions d'éclairage sont relativement stables et satisfaisantes (la vidéo a été enregistré à 16h sous la lumière du jour) ;
- La « résolution va être variée entre [50x50] px et [500x500] px.

Résultats :

Les résultats obtenus de cette première expérience sont présentés sous forme d'un graphe comme suit :

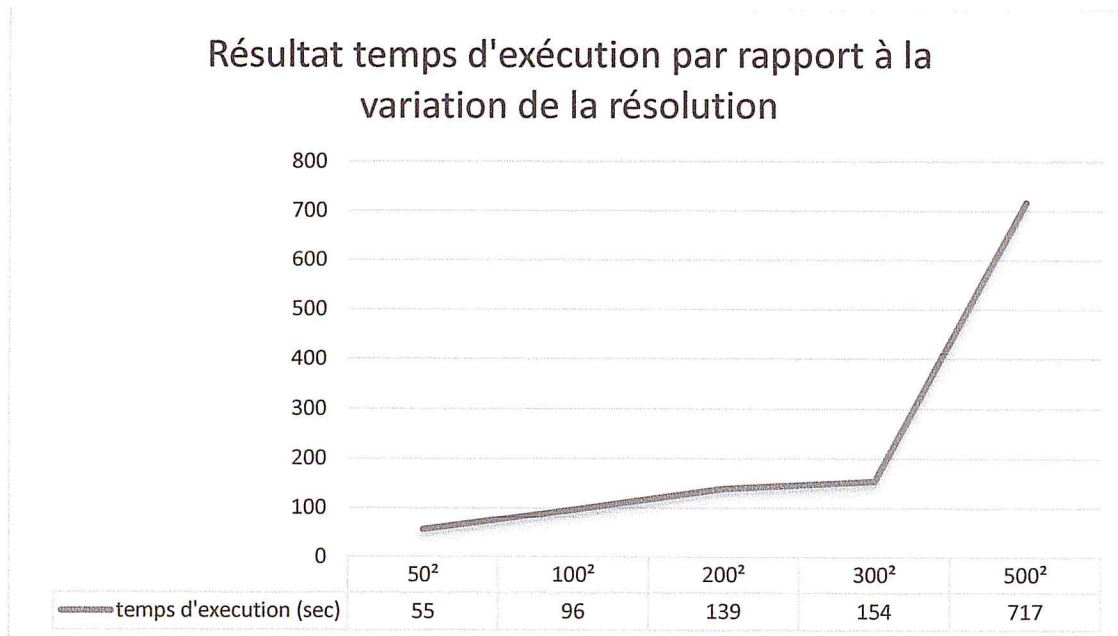


Fig V.10 Résultat de temps d'exécution par rapport à la variation de la résolution

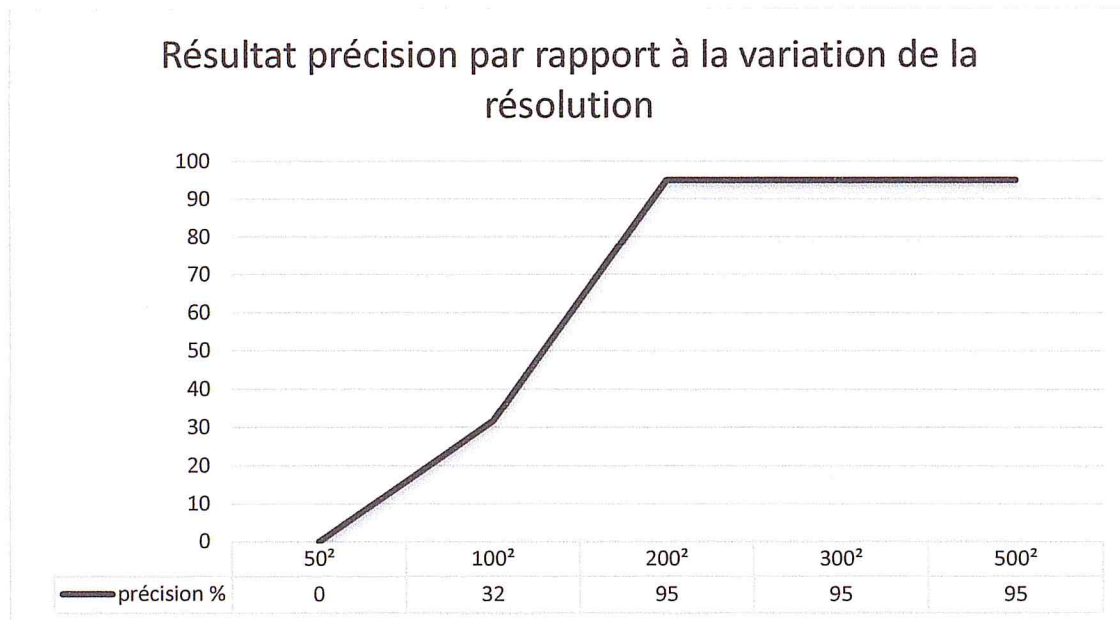


Fig V.11 Résultat de précision par rapport à la variation de la résolution

Analyse :

Sur soixante personnes de différent âges et taille, et à de différentes résolution d'image, nous remarquons que les résultats changent comme suit :

Si la résolution est mauvaise (50x50) px à (100x100) px :

- Le traitement sera rapide, par conséquent le temps d'exécution sera plus petit.
- La précision de détection sera moins élevée à cause de la faible précision de l'image.

Si la résolution est meilleure de (200x200) px à (500x500) px :

- Le traitement sera un peu lent, le temps d'exécution sera donc plus grand.
- La précision de détection sera plus élevée à cause de la forte précision de l'image.

Déduction

La résolution doit être augmentée progressivement jusqu'à l'obtention d'une précision optimale. Dans le cas présent nous avons constaté que la résolution (200x200 px) à permit d'obtenir une précision de détection et de comptage plus élevée, avec un temps d'exécution plus convenable d'où une petite consommation de mémoire.

V.4 .3 Test sur la position des lignes de comptage

Les résultats obtenus de la première expérience (résolution optimale 200x200 px) vont nous servir pour cette deuxième expérience, où l'objectif est d'améliorer les performances de notre système en termes de précision de comptage.

CHAPITRE V : REALISATION ET EXPERIMENTATION

Les circonstances de cette expérience sont les mêmes que celles de la première, en ajoutant le paramètre résolution fixé à 200x200 px.

- Le paramètre à faire varier dans cette expérience est la position selon l'axe des «Y» que l'on fixera entre 100px et 175px.

Résultats

Les résultats obtenus de cette deuxième expérience sont présentés sous forme d'un graphe comme suit :

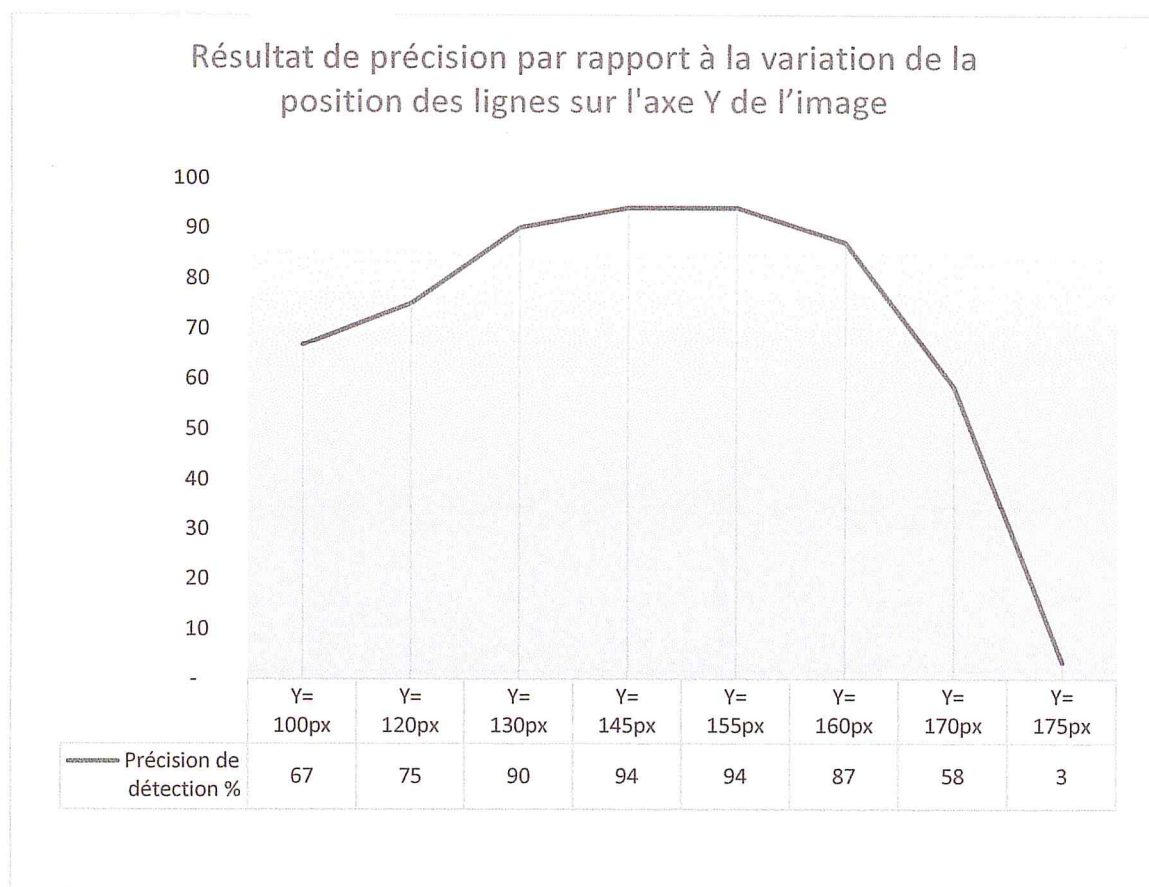


Fig V.12 Résultat de précision par rapport à la variation de la position des lignes selon l'axe Y

Analyse :

Nous pouvons voir qu'après modification de la position des lignes selon l'axe Y, la précision de comptage a augmenté jusqu'à atteindre 94 % aux positions Y=145 px et Y=155px.

Déduction

Nous déduisons de ces deux expériences que la résolution de 200 x 200 px et la position de Y=155px est une configuration convenable mais qui peut être améliorée en affinant le paramètre « épaisseur ».

V.4 .4 Test sur l'épaisseur des lignes de comptage

Cette expérience vient en complément des deux précédentes, pour améliorer la fiabilité du programme et augmenter la précision du comptage. Nous allons procéder dans cette expérience à la variation de l'épaisseur, Et ce, sous les conditions suivantes :

- Résolution de l'image à 200px
- Position des lignes à Y=155px
- Epaisseur variant de 1px à 19px

Résultat

Les résultats obtenus de cette deuxième expérience sont présentés sous forme d'un graphe comme suit :

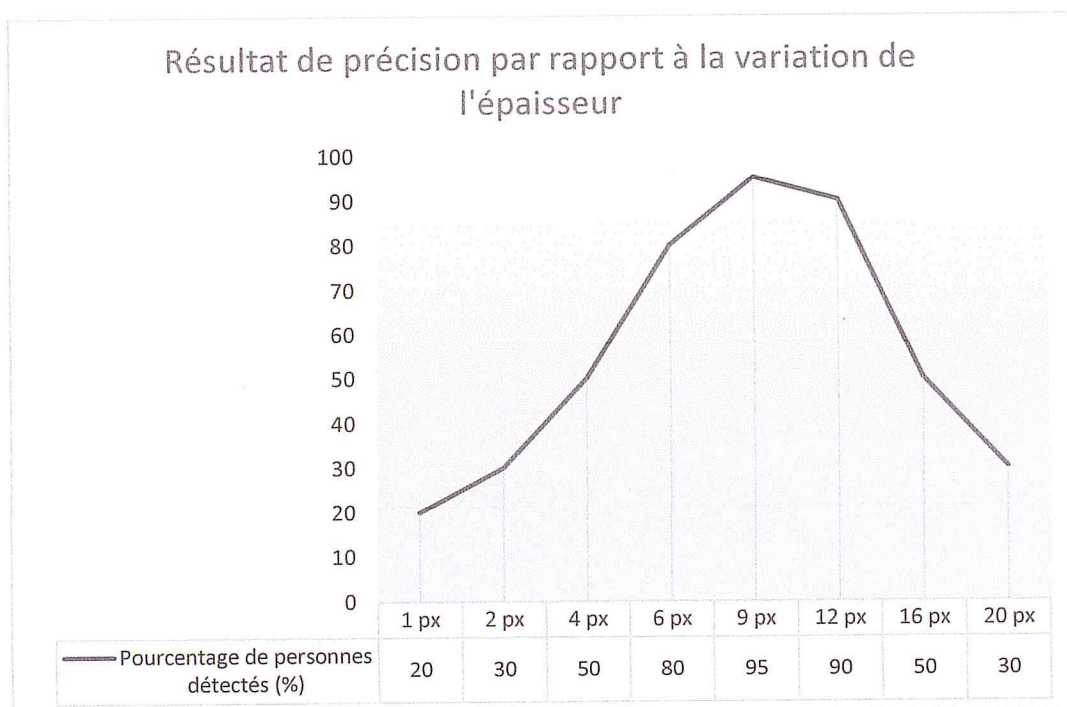


Fig V.13 Résultat de précision par rapport à la variation de l'épaisseur

Analyse

Nous avons remarqué que lorsque l'épaisseur est à 9px le résultat obtenu est à 95% mais si elle est inférieure ou supérieure à 9 px nos constatons un manque de précision du comptage, cela peut être expliqué comme suit :

- Cas d'une petite épaisseur (dans notre cas $> 9\text{px}$) : Dans le cas d'une personne qui marche rapidement, elle peut traverser les deux lignes de comptage sans toucher l'une ou l'autre ou les deux à la fois, entre deux frame successifs.

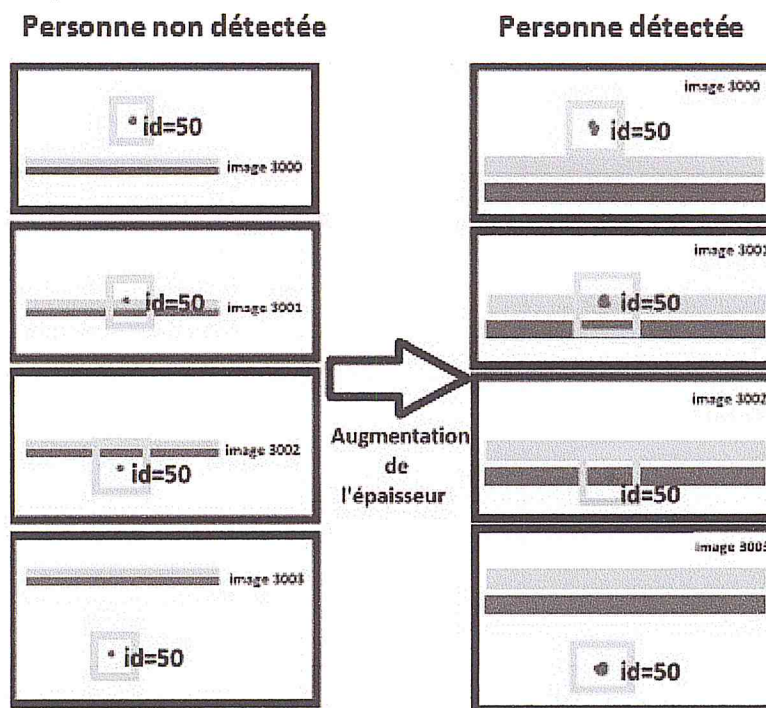


Fig V.14 Exemple de la nécessité de réglage de l'épaisseur

- Cas d'une grande épaisseur (dans notre cas $< 9\text{px}$) : Dans le cas de vibration et de manque de stabilité de la caméra, l'identifiant de chaque personne peut facilement changer d'un frame à l'autre, ce qui implique une possibilité de plusieurs détections différentes sous une seule ligne de comptage trop épaisse.

Déduction

Après avoir analysé les résultats obtenus lors de ces expériences, nous constatons que la résolution de 200 x 200 px et la position des deux lignes à $Y=155\text{px}$ et l'épaisseur de 9px est la configuration la plus efficace selon nos modalités de prise de vue pour une détection optimale.

V.4 .5 Comparaison entre des algorithmes de soustraction de fond :

Nous avons effectué une comparaison entre quatre algorithmes effectuant la soustraction de fond, à savoir : les deux algorithmes d'OpenCV « MOG » et « MOG2 » et deux autres algorithmes du BGS Library « Adaptive Background Learning » et « Adaptive Selective Background Learning »

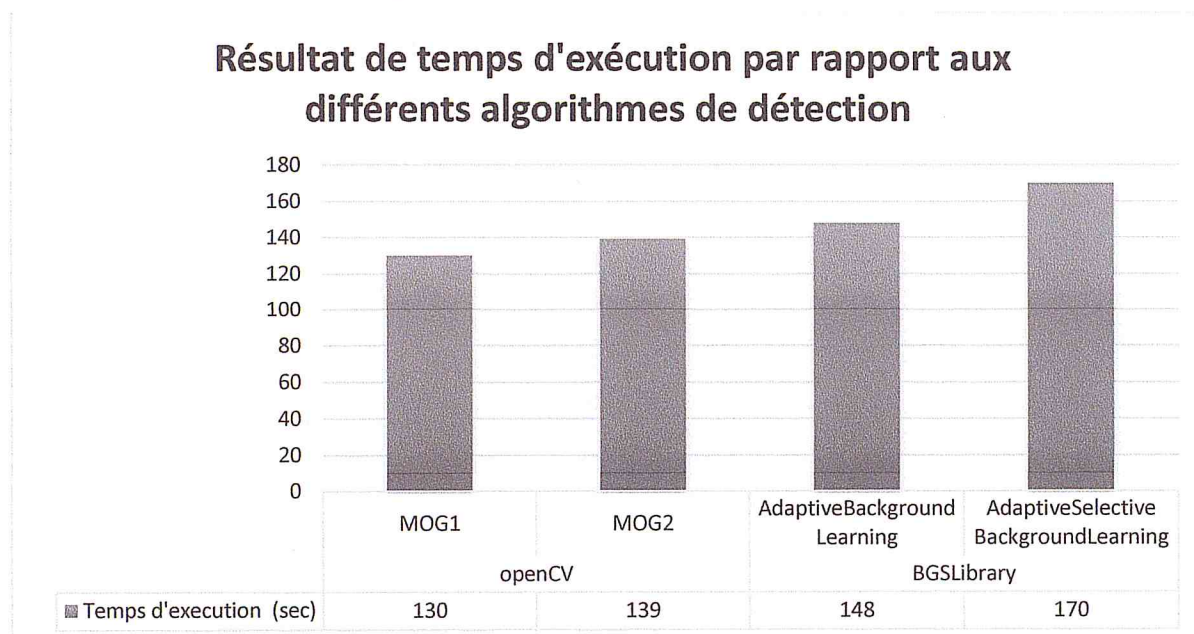


Fig V.15 Résultat de temps d'exécution par rapport aux différents algorithmes de détection

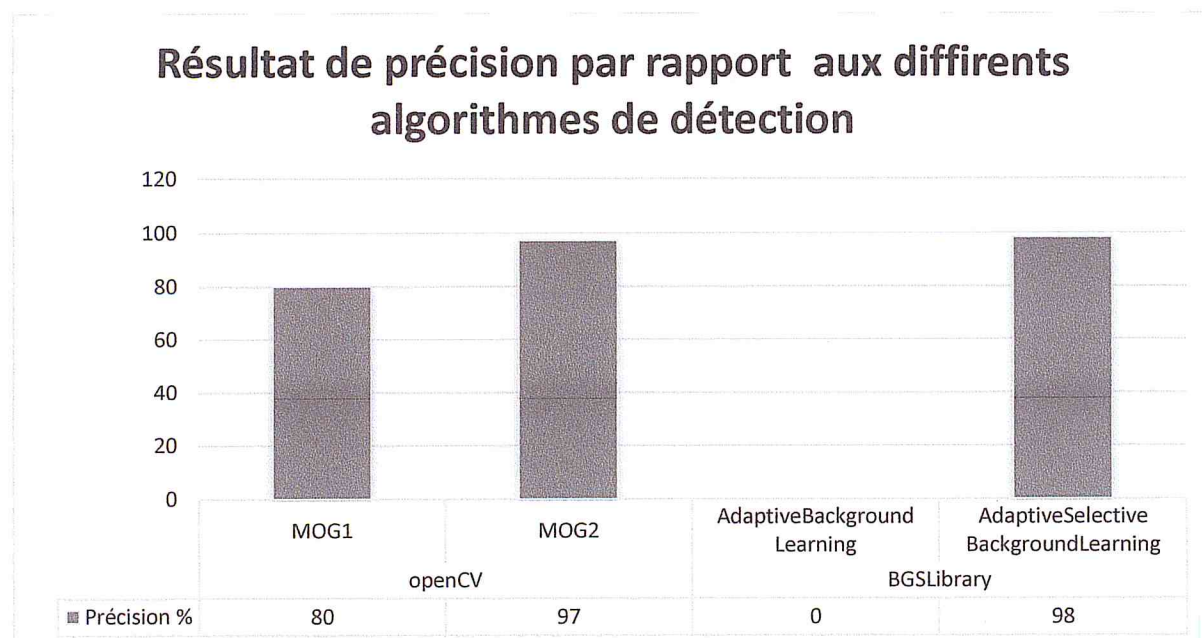


Fig V.16 Résultat de précision par rapport aux différents algorithmes de détection

Analyse

L' *AdaptiveBackgroundLearning* nous a donnée une précision de 0% et un temps d'exécution de 148 s, par contre le *MOG1* nous a donnée une précision de 80% et un temps d'exécution de 130 s, aussi l' *AdaptiveSelectBackgroundLearning* nous a donné une précision de 98 % et un temps d'exécution de 170 s, enfin le *MOG2* a atteint une précision de 97.%.

Déduction

Nous constatons que le *MOG2* est la meilleure méthode de détection car elle donne un meilleur rendement.

V.4 .6 Comparaison entre CvTracks et CvBlobs

Afin de bien choisir la meilleure méthode de suivi nous, avons comparé entre les deux méthodes de suivi de la librairie *CvBlob.h*, et le résultat est dans le graphe sous cité ;

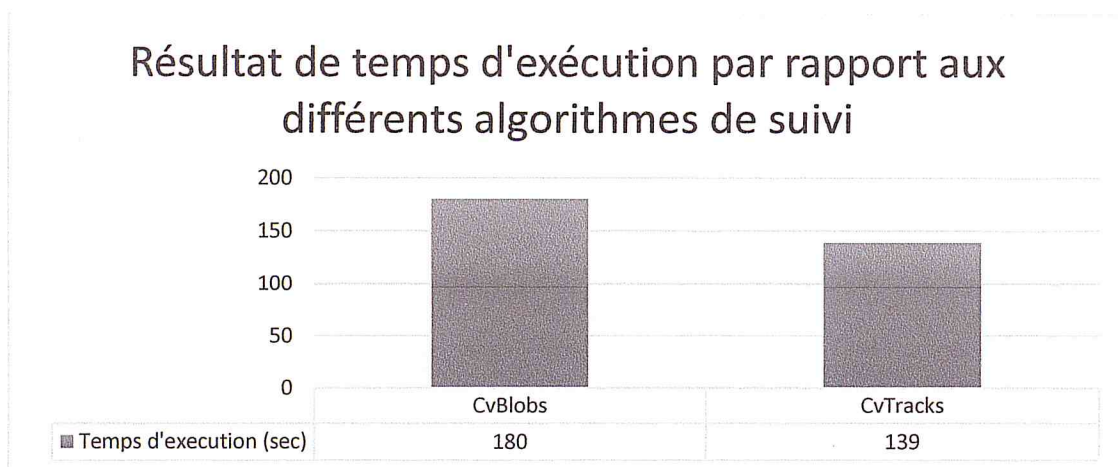


Fig V.17 Résultat de temps d'exécution par rapport aux différents algorithmes de suivi

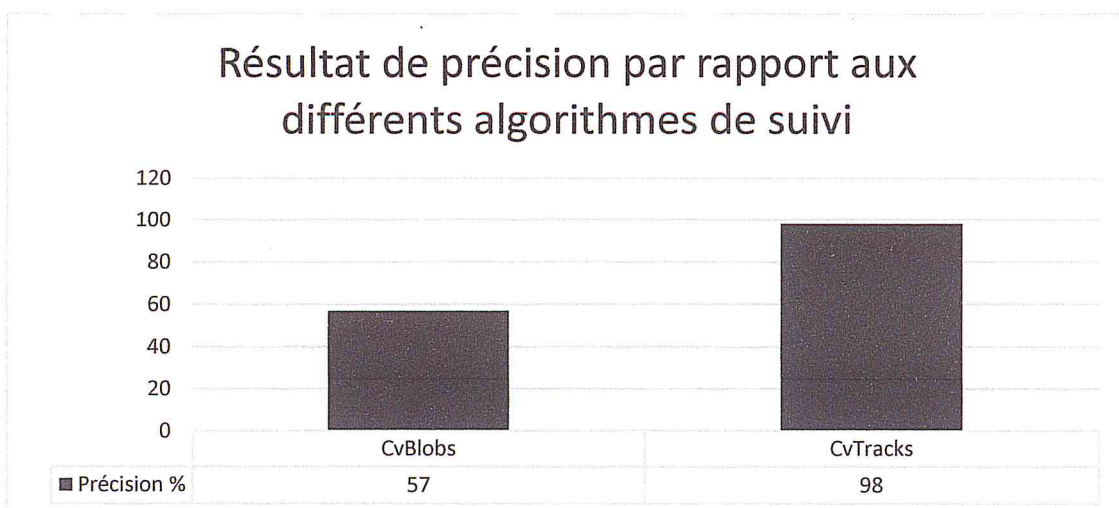


Fig V.18 Résultat de précision par rapport aux différents algorithmes de suivi

Analyse

Nous avons remarqué que *CvBlobs* nous a donné une précision de 57% et un temps d'exécution de 180 s car le CvBlobs fonctionne avec un Label instable (voir la section V.3.2), par contre nous avons atteint une précision de 98% au CvTracks avec un temps d'exécution de 139 s.

Déduction

Le CvTracks est la meilleure méthode de suivi qui permet une bonne fonctionnalité de l'application à 98%

V.4 .7 Test du programme sur un enregistrement en temps réel

Nous avons placé notre caméra a l'entrée d'une pharmacie pour s'assurer du bon fonctionnement de notre application en temps réel. Et pour vérifier la précision de notre comptage, nous avons effectué un comptage manuel de personnes entrantes et sortantes. Le résultat obtenu est détaillé dans le graphe suivant.

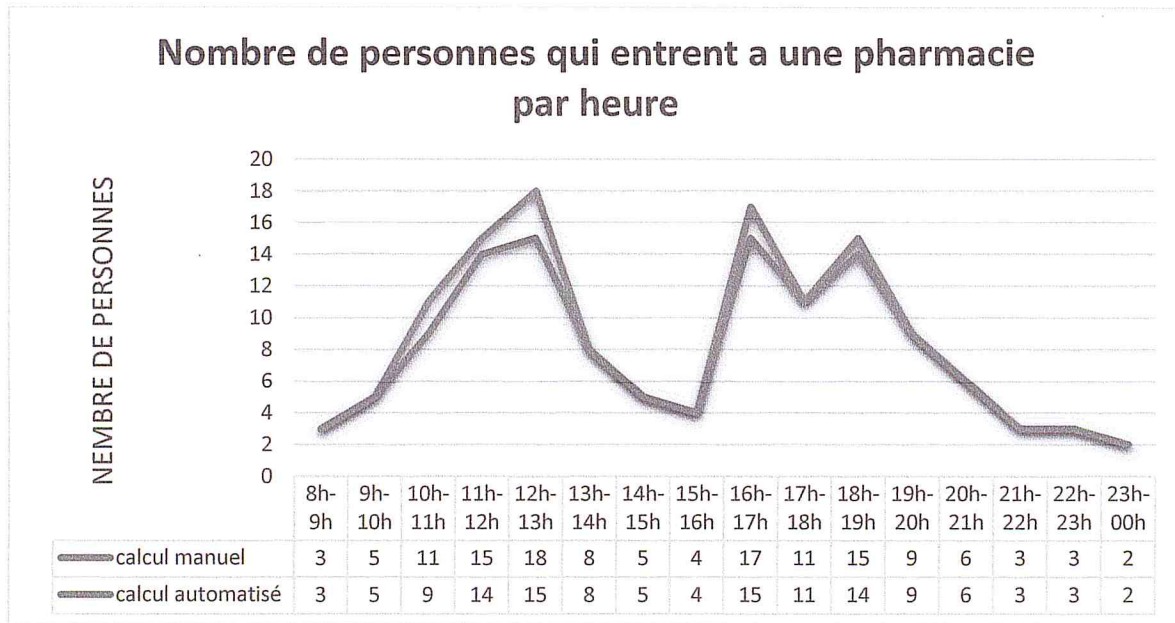


Fig V.19 Résultat d'un comptage en temps réel

Analyse

Nous remarquons que dans ce graphe le comptage manuel se rapproche beaucoup de celui de la vidéo enregistrée.

Déduction

A partir de l'analyse du graphe, nous remarquons que nos résultats sont très satisfaisants et cela a prouvé la réussite de notre programme.

V.5 Conclusion

Dans ce présent chapitre nous avons fourni une description détaillée de l'implémentation de notre système de détection, suivi et comptage de personnes, ensuite nous avons analysé les résultats obtenus suite aux tests réalisés sur différents paramètres, à savoir, la résolution de la vidéo utilisée, la position et l'épaisseur des lignes de comptage, quelques méthodes de soustraction de fond et d'autres de suivi. Cela est fait dans le but de mesurer l'exactitude et la cohérence de notre programme, améliorer ses performances et conclure sur les situations dans lesquelles ce système est plus fiable. Les résultats tirés des expériences réalisées ont montré que la méthode proposée est robuste et fournit un bon taux de précision.

CONCLUSION GENERALE

CONCLUSION GENERALE

Nous avons présenté dans ce mémoire le fruit de notre travail qui relève du domaine de traitement d'image. Ce travail s'est soldé par la mise en œuvre d'un système de détection, suivi et comptage de personnes dans une vidéo numérique, au profit de CERIST.

Dans ce mémoire, nous avons tout d'abord essayé d'exposer les différents concepts basiques qui entourent les images et des vidéos numériques. Nous avons ensuite mis l'accent sur les multiples avantages qu'un système de comptage de personnes peut nous offrir, et expliqué les raisons de notre choix d'utiliser une caméra vidéo numérique comme moyen de comptage. Puis, nous avons décrit les différents aspects théoriques qui entourent les principaux composants que doit avoir un système de comptage de personnes dans une vidéo numérique. Enfin, nous avons fourni une description détaillée de notre système de comptage réalisé, notamment, l'approche de sa conception, sa mise-en-œuvre, son fonctionnement et sa précision sous différentes situations de test.

Le travail que nous avons réalisé, profite des avantages offerts par le langage de programmation C++, en particulier, ceux relatifs à l'exploitation des fonctionnalités de la bibliothèque graphique libre *OpenCv* qui est spécialisée dans le traitement d'images en temps réel.

Par ailleurs, ce projet nous a permis de s'initier aux mécanismes passionnants du traitement d'image, depuis la capture de l'image jusqu'à l'utilisation pratique des algorithmes relativement complexe dans des cas concrets, notamment des algorithmes de traitements morphologiques, de détection d'objets et ceux de suivi.

Au-delà du travail réalisé, nous avons appris durant ce projet que la précision d'un système de comptage basé sur l'utilisation de la vision par ordinateur est un terme relatif, elle n'est jamais absolue, car elle est souvent sensible aux changements rapides d'éclairage, de la position de la caméra et des problèmes d'occlusion, ce qui induit à de fausses détections et de faux résultats de comptage de personnes. Néanmoins, il est possible d'améliorer les conditions d'observation en optimisant le lieu d'implantation du système. Ainsi, il est préférable d'installer la caméra dans des environnements en intérieur où les conditions d'éclairage sont relativement stable, plus précisément, dans des endroits où la foule est réduite (ex : couloir, accès, porte, ...), cela va permettre d'augmenter considérablement le taux de fiabilité d'un tel système.

Difficultés rencontrés:

Tout au long de notre travail, nous étions amenées à surmonter certaines difficultés qui ne nous ont pas facilité le progrès dans la réalisation de notre système de détection, suivi et comptage de personnes :

- Le manque de documentations sur la procédure d'installation et de configuration de la bibliothèque **OpenCv** ;
- Le manque de documentations concernant les systèmes actuels de détection, suivi et comptage ;
- Difficultés de détection de personnes dus aux changements de luminosité, conditions climatiques, arrière-plan en mouvement, etc ;
- Difficulté de confirmer qu'une détection correspond sûrement à une personne, surtout dans le cas où un simple objet a une taille presque similaire à celle d'une personne, par exemple, un poussette ou un panier pourrait être compté comme une personne ;
- Lorsque les personnes sont proches (cas de grande foule où plusieurs personnes qui s'entrecroisent), il peut y avoir des occultations rendant difficile la tâche d'extraction et de suivi de personnes ;
- Difficultés de détection de personne lorsqu'elle porte une couleur proche de la couleur d'arrière-plan, car l'algorithme de soustraction de fond va la considérer comme élément d'arrière-plan.

Amélioration et perspective :

Ce projet n'a pas la prétention d'être exhaustif sur la question du comptage, de nombreux aspects restent encore à explorer. Voici quelques-uns :

- Amélioration de notre approche en ajoutant un système de reconnaissance de forme qui permettra de détecter les têtes de personnes, ce qui va éliminer considérablement les fausses détections ;
- Ajout d'un système pour estimer la profondeur de la scène selon la position de la caméra ;
- Implémenter une interface pour faciliter la manipulation de notre programme.

BIBLIOGRAPHIE

BIBLIOGRAPHIE

- [B01]: GROUPE ADOBE DYNAMIC MEDIA : Initiation à la vidéo numérique, Juin 2000.
- [B02]: SLIMANE LARABI : Systèmes multimedia, Cours Master RSD, USTHB, 2014/2015.
- [B03] : J.M.BONNEFOND : Les capteurs de mesures du rayonnement naturel, 2008.
- [B04] : M.BRULIN : Analyse sémantique d'un trafic routier dans un contexte de vidéo-surveillance, Université Bordeaux 1, Octobre 2012.
- [B05] : A. BAYONA, J.C. SANMIGUEL, J.M. MARTINEZ : Comparative evaluation of stationary foreground object detection algorithms based on background subtraction techniques, Proceedings of the 6th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 25-30, genova, Italy, 2009.
- [B06] : R. MATHEW, Z. YU, J. ZHANG: Detecting new stable objects in surveillance video, Proceedings of the IEEE 7th Workshop on Multimedia Signal Processing, pages 1-4, Shanghai, 2005.
- [B07] : F. PORIKLI: Detection of temporarily static regions by processing video at different frame rates, Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 236-241, London, UK, 2007.
- [B08] : A. FRANÇOIS, and G. MEDIONI: Adaptive color background modeling for real-time segmentation of video streams, Proceedings of the International Conference on Imaging Science Systems and Technology, page 227-232, 1999.
- [B09] : LYES HAMOUDI : Application de techniques d'apprentissage pour la détection et la reconnaissance d'individus, Ecole des Mines de Douai et l'Université Lille 1 Sciences et Technologies, Juin 2011.
- [B10] : N. SEBE, M. S. LEW: A maximum likelihood, investigation into color indexing, Proceedings Visual Interface, pages 101-106, Montreal, Canada, 2000.
- [B11] : A.YILMAZ, O. JAVED et M.SHAH: Object tracking : A survey. Acm Computing Surveys (CSUR), vol. 38, no.4, page 13, 2006.
- [B12] : T. BROX, B. ROSENHAHN, J. GALL et D. CREMERS: Combined region and motion-based 3d tracking of rigid and articulated objects. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 32, no. 3, pages 402-415, 2010.
- [B13] : M.H. YANG et J. HO: Toward Robust Online Visual Tracking. Distributed Video Sensor Networks, pages 119-136, 2011.

BIBLIOGRAPHIE

- [B14] : B. ROSENHAHN , U. KERSTING, S. ANDREW, T. BROX, R. KLETTE et H.P.SEIDEL: A silhouette based human motion tracking system. Rapport technique, CITR, The University of Auckland, New Zealand, 2005.
- [B15] : A. YILMAZ, X. LI et M. SHAH: Contour-based object tracking with occlusion handling in video acquired using mobile cameras. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 26, no. 11, pages 1531-1536, 2004.
- [B16] : L. HERDA, P. FUA, R. PLANKERS, R. BOULIC et D. THALMANN: Using skeleton based tracking to increase the reliability of optical motion capture. Human movement science, vol. 20, no. 3, pages 313-341, 2001.
- [B17] : G. WELCH, G. BISHOP: An Introduction to the Kalman Filter, the Association for Computing Machinery's Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH) Course, 2001.
- [B18] : M. ISARD, A. BLAKE: Contour tracking by stochastic propagation of conditional density, Proceedings of the 4th European Conference on Computer Vision (ECCV), pages 343-356, Cambridge, UK, 1996.
- [B19] : D. COMANICIU and P. MEER: Mean shift analysis and application, Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 1197-1203, Kerkyra, Corfu, Greece, 1999.
- [B20] : B.K.P. HORN and B.G. SCHUNK: Detenninig optical flow, Artificial Intelligence, vol. 17, pages 185-203, 1981.
- [B21] : M. KASS, A. WITKIN, D. TERZOPOULOS, SNAKES: active contour models, International Journal of Computer Vision, vol. 1 (4), pages 321-331, 1988.
- [B22] : A. XIN and X. LI, M. SHAH: Object contour tracking using level sets, Asian Conference on Computer Vision (ACCV), Jeju, Korea, 2004.
- [B23] : K. TOYAMA, J. KRUMM, B. BRUMITT, B. MEYERS, WALLFLOWER: principles and practice of background maintenance, Proceedings International Conference Computer Vision, Kerkyra, Greece, 1999.
- [B24] : TIM VAN OOSTERHOUT, SANDER BAKKES, BEN KRÖSE : Head Detection In Stereo Data For People Counting And Segmentation, 6th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Application, 2011.
- [B25] : P. KADEWTRAKUPONG AND R. BOWDEN : An improved adaptive background mixture model for real-time tracking with shadow detection, Proc. 2nd European Workshop on Advanced Video-Based Surveillance Systems, Sept 2001.

WEBOGRAPHIE

[W01]: <https://openclassrooms.com/courses/debuter-dans-l-infographie-avec-gimp/les-images-numeriques>

[W02]: <http://www.tomshardware.fr/articles/bases-photo-numerique,2-349.html>

[W03]: <http://www.mycube.fr/quest-ce-quune-image-numerique>

[W04]: https://fr.wikipedia.org/wiki/Image_numérique

[W05]: <http://www.linternaute.com/hightech/micro/dossier/tout-sur-les-formats-multimedia/tableau-recapitulatif-des-principaux-formats-d-image.shtml>

[W06]: <https://fr.wikipedia.org/wiki/Vidéo>

[W07]: <https://linuxaudiovideo.wordpress.com/2014/03/18/notions-de-base-pour-la-video-definition-frequence-dimage-entrelacement-codecs-et-codage-des-pixels/>

[W08]: <http://www.infodev.ca/fr/about/lisez-nos-articles/analyse-comparative-des-technologies-de-comptage.html>

[W10]: http://docs.opencv.org/3.1.0/d1/dc5/tutorial_background_subtraction.html

[W11]: <http://images.math.cnrs.fr/Le-traitement-numerique-des-images.html>

[W12]: http://opencv.itseez.com/2.4/modules/video/doc/motion_analysis_and_object_tracking.html

[W13]: https://fr.wikipedia.org/wiki/Modèle_de_mélanges_gaussiens