

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université 08 Mai 1945-Guelma

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

Département d'Informatique



Mémoire de fin d'étude de Master

Filière : Informatique

Spécialité : Ingénierie des Medias

Thème :

**Proposition et réalisation d'un système d'élimination
d'ombre des objets appliqué aux images extraites à
partir d'une vidéo surveillance.**

Sous la direction de :

Mr. Farou Brahime

Présenté par :

Belhireche Azzedine,

Rouabhia Houssef

Juin 2012



Dédicaces

Nous dédions ce modeste travail
A nous chères parents, pour leur soutiens, leurs sacrifices et leur
amour durant toutes ces années.

A tout les membres de nous familles surtout :

*Allaa (lola), Chifa, Abd Elheye, Abd Elnore, Akrame, Abd
Elhamide, Ali & Noura, Lamyia & Hmed.*

Azzedine

Djoumana (Jojo), Ilyes, Amel.

Housseem

A *Farou Brahim* pour son aide.

A nous chères :

*Jiji, Meryem&Madjda, Mouna, Imene, Asma
Rachide (Ederga), Amine (ould Frihi), Amine (Lixtoon), Hamza,
Hmed, Chrife, Hichame&Hichame, Mouhssen, Saleh.*

A tout nous amis et les membres de la section *IM juan 2012*
sons exception.

Remerciement

Nous tenons à remercier avant tout, dieu le tout puissant qui nous à aidé pour réaliser ce travail et nous à donné la volonté et la santé pour suivre nos études.

Nous remercions notre encadreur *M : Farou Brahim* qui nous à bien soutenu et qui nous à aidé avec ces précieux conseils.

Nous remercions tous les professeurs du département d'informatique *E8* sons exception

Nous remercions tous ceux qui nous ont aidés de près et de loin dans la réalisation de ce travail.

Housseem & Azzedine

Résumé :

De nos jours, l'utilisation des caméras de vidéo surveillance est devenue de plus en plus importante. Certains logiciels seraient capables de réaliser des traitements sur les séquences vidéos et de gérer automatiquement différents évènements tels que la détection des mouvements, de bruit, compter le nombre de véhicules passant dans le champ de vision d'une caméra, lire les plaques d'immatriculation et même de pratiquer la reconnaissance faciale en temps réel, aussi de générer des actions comme le déclenchement d'une alarme, l'annonce d'un message, etc.

La détection automatique des objets en mouvement est une tâche très importante dans les systèmes de vidéo de surveillance. Les anciens systèmes basés sur les compétences des agents de sécurité ont montré leurs limites. Effectivement, beaucoup de séquences visuelles importantes peuvent leur échapper. Pour cela un système de vidéo surveillance s'impose permettant de seconder les agents dans leurs travaux. L'extraction de l'arrière-plan est l'une des méthodes les plus utilisées pour la détection et le suivi des objets en mouvement. Cette méthode permet de détecter n'importe quel objet en mouvement, mais présente l'inconvénient de ne pas faire la différence entre l'objet et son ombre, induisant ainsi le système en erreur. Dans ce travail, nous allons essayer de proposer une démarche qui permettra d'éliminer les ombres des objets pour un meilleur rendement des systèmes de vidéo de surveillance. Notre mémoire se compose de trois chapitres:

- Chapitre 1 : Réalisation d'une étude théorique sur les approches existantes dans le domaine, et les travaux connexes réalisés au cours des dernières années.
- Chapitre 2 : Proposition et implémentation d'un système qui élimine les ombres.
- Chapitre 3 : Réalisation d'une suite de tests sur une base de vidéos créée dans des conditions de nature différentes.

Sommaire

Liste des figures :	4
Liste des tableaux :	5
Introduction générale :	6

Chapitre 1 : état de l'art

Introduction.....	8
I. Quelques définitions :	9
1. La lumière :	9
1.1. Qu'est-ce que la lumière ?	9
1.2. Comment la lumière se propage-t-elle?	9
1.3. Les différentes sources de lumière :	9
1.4. Explications sur les bases de la qualité de la lumière :	10
2. Les espaces de couleur :	11
2.1. L'espace RGB :	11
2.2. L'espace HSV :	11
2.3. L'espace YUV :	12
3. La vidéo surveillance :	13
3.1. Approche technique :	13
3.2. Objectifs :	13
3.3. Typologie des systèmes existants :	14
3.4. La détection des changements :	16
3.4.1. La détection du fonds de scène et analyse de l'avant de scène :	16
3.4.2. L'estimation du mouvement :	17
4. Les textures :	18
4.1. Qu'est-ce qu'une texture ?	18
4.2. Distribution des textures :	19
4.3. Analyse et synthèse :	19
5. Les ombres :	20
II. Manipulation des ombres :	21
1. Caractéristiques utiles pour la détection d'ombre :	21
1.1. L'intensité :	21
1.2. La Chromaticité :	22
1.3. Les propriétés physiques :	22
1.4. La Géométrie :	22

1.5. Les Textures :	23
2. Les dernières méthodes de détection d'ombre :	23
3. Évaluation comparative :	26
3.1. Les performances de détection d'ombre :	26
3.1.1. Les résultats quantitatifs :	27
3.1.2. Les résultats qualitatifs :	28
Conclusion :	30

Chapitre 2 : Conception & Implémentation

Introduction :	31
I. Architecture globale du système :	32
II. Les Mixtures de Gaussien (GMM) :	33
III. Implémentation du système :	34
1. La méthode basée sur les textures larges :	34
2. La méthode basée sur les textures larges améliorée :	37
a. Etape 1 : L'approche basée sur la chromaticité :	37
b. Etape 2 : L'approche basée sur les textures larges :	39
3. Approche proposée :	40
3.1. Le filtre de "Canny" :	40
3.2. Création des régions :	41
IV. Extraction et élimination des ombres :	42
Conclusion :	43

Chapitre 3 : Tests & résultats

Introduction :	44
1. Outils utilisés :	44
1.1. Code-Blocks :	44
1.2. Qt creator :	45
1.3. OpenCV 2.0 :	46
1.3.1. Traitement des images et des vidéos avec OpenCV :	46
2. Présentation de l'interface de l'application :	49
3. Comparaison de la performance des méthodes :	50
3.1. Critère 1 : la performance de détection d'ombre :	56
3.2. Critère 2 : la discrimination entre les pixels d'ombre et d'objets :	56
3.3. Critère 3 : le temps d'exécution :	56
4. Conclusion :	57
Conclusion générale :	58

Référence :59

Liste des figures :

Figure 1.1 : les trios composants de l'espace RGB.

Figure 1.2 : système de caméra surveillance (caméras reliées à un simple écran).

Figure 1.3 : système de caméra surveillance (caméras analogiques).

Figure 1.4 : système de caméra surveillance (caméras numériques consultables).

Figure 1.5 : les zones d'ombre qu'elle se forme à l'arrière de l'objet qui obscurcit la lumière.

Figure 1.6 : Comparaison des résultats de la détection d'ombre par la séquence.

Figure 2.1 : schéma global du système.

Figure 2.2 : les étapes d'implémentation de notre méthode.

Figure 2.3: Étapes de création: (a) Image originale (b) Masque de premier plan obtenu avec GMM, (c) Masque d'arrière-plan obtenu avec GMM.

Figure 2.4: (a) l'image originale, (b) Masque des pixels candidates d'ombre obtenue après l'application de l'approche basée sur les textures larges, (c) l'image sans ombre.

Figure 2.5: (a) Image originale, (b) Masque des candidates pixels d'ombre (les pixels en blanc).

Figure 2.6: (a) l'image originale, (b) Masque des pixels candidates d'ombre obtenue après l'application de l'approche basée sur les textures larges, (c) l'image sans ombre.

Figure 2.7: (a) l'image originale, (b) Masque des contours obtenus avec le filtre de **Canny**.

Figure 2.8: (a) l'image originale, (b) Masque du premier plan obtenu avec GMM, (c) Masque des contours obtenu avec le filtre de **Canny**, (d) régions créées.

Figure 2.9: (a) régions détectées comme des régions d'ombre, (b) image sans ombre.

Figure 2.10: (a) image originale, (b) l'arrière-plan, (c) Masque des pixels d'ombre, (d) l'image sans l'ombre.

Figure 3.1: Environnement de développement Intégré (IDE) de Code-Blocks.

Figure 3.2: Environnement de développement Intégré de Qt Creator.

Figure 3.3: interface de l'application.

Figure 3.4 : résultats obtenus avec la méthode 1.

Figure 3.5 : résultats obtenus avec la méthode 2.

Figure 3.6 : résultats obtenus avec la méthode 3.

Dans les dernières années, de nombreux travaux ont été publiés sur le sujet de détection des ombres, cependant, il n'y a pas une méthode parfaite pour détecter des ombres dans les séquences d'images, de nombreuses méthodes ont besoin d'une étude spécialisée pour les rendre confinées à un environnement particulier.

Concernant notre objectif, nous allons proposer dans ce mémoire une méthode pour enlever les ombres à partir des séquences de vidéo surveillance en se basant sur les propriétés chromatiques dans les trois espaces de couleur RGB, HSV et YUV.

Introduction générale :

Beaucoup d'applications de vision par ordinateur traitant la vidéo nécessitent la détection et le suivi d'objets en mouvement. Lorsque les objets d'intérêt ont une forme bien définie, plusieurs classificateurs sophistiqués peuvent être utilisés pour segmenter les objets directement à partir de l'image. Ces techniques fonctionnent bien pour des objets bien définis tels que les véhicules, mais sont difficiles à mettre en œuvre pour les objets non rigides tels que les corps humains. Une approche plus commune pour la détection des objets en mouvement dans une séquence vidéo est de détecter les pixels de premier plan, par exemple via des modèles de mélange gaussien.

Cependant, les techniques actuelles ont généralement un inconvénient majeur, les ombres qui ont tendance à être classés comme faisant partie du premier plan, parce que les ombres partagent les mêmes habitudes de déplacement et ont une ampleur similaire du changement d'intensité que celle des objets au premier plan, sachant que l'ombre est supposé réduire l'intensité, mais ne change pas la teinte ou la saturation. En outre dans les scènes dynamiques, tous les points mobiles des objets ou des ombres sont détectés en même temps, les ombres et les objets partagent les mêmes caractéristiques visuelles de mouvement et de détectabilité.

Les ombres sont dues à l'occlusion de source de lumière par un objet dans la scène. Quand un objet projette une ombre sur une surface, il bloque partiellement ou complètement l'éclairage direct d'une source de lumière, produisant un changement dans son apparence. La mesure de ce changement, entre un pixel affecté par l'ombre et le même pixel en l'absence d'ombres portées, est l'une des principales propriétés utilisées pour classer les ombres dans les algorithmes de segmentation d'arrière-plan au premier plan.

Étant donné les caractéristiques d'ombre, la densité des ombres dépend de la quantité de lumière réfléchiée par la surface sur laquelle l'ombre de notre sujet est projetée. Pour prendre un simple exemple, si nous photographions un objet sur une table blanche, les ombres seront profondes (grande différence de luminosité entre l'ombre créée par l'objet et la surface blanche éclairée), si nous photographions un objet sur une table noire, les ombres seront relativement peu profondes (peu de différence entre le noir de l'ombre et le noir de la surface).

Chapitre 1

Etat de l'art



3.3. Typologie des systèmes existants :

On distingue plusieurs catégories de caméras : intérieur ou extérieur, fixes ou mobiles, couleur ou noir et blanc. Par l'utilisation de circuits amplificateurs de lumière ou par traitement infrarouge, certaines caméras sont capables aujourd'hui de filmer de jour comme de nuit, selon leur sensibilité. La caméra peut être attachée à un objet fixe (mur, lampadaire) ou bien mobile (automobile, drone).

De nos jours, les caméras vidéos sont mobiles (rotatives et contrôlables à distance). Certains logiciels seraient capables de compter le nombre de véhicules passant dans le champ de vision d'une caméra, de lire les plaques d'immatriculation et même de pratiquer la reconnaissance faciale en temps réel.

On distingue quatre types des systèmes de caméras de surveillance :

- Les caméras reliées à un simple écran de contrôle, sans enregistrement. On les trouve dans certains magasins. Elles permettent au vendeur de surveiller en direct certains rayons. C'est le système le plus économique.

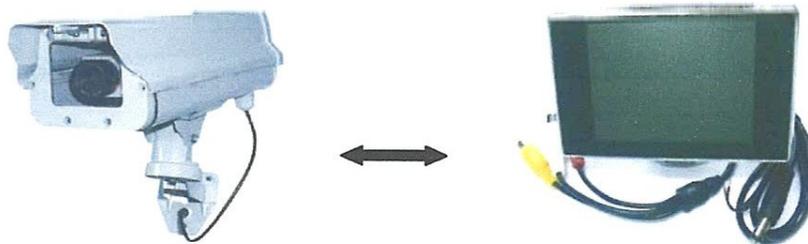


Figure 1.2 : système de caméra surveillance (caméras reliées à un simple écran).

- Le système de caméras analogiques, avec un enregistrement limité dans la durée. Il s'agit d'une des méthodes les plus anciennes donc également les plus répandues dans un grand nombre d'établissements.

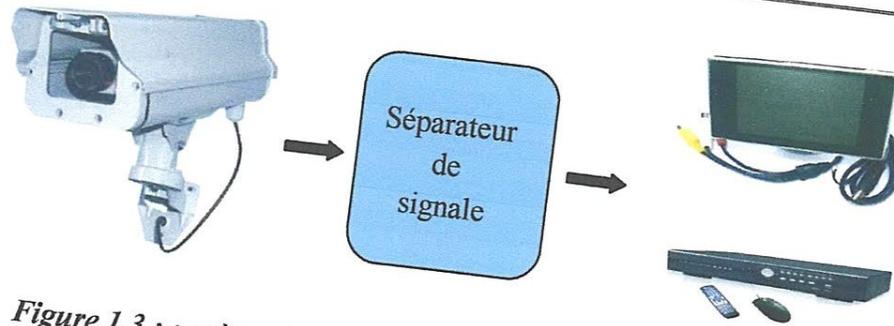


Figure 1.3 : système de caméra surveillance (caméras analogiques).

- Le système intégrant des caméras analogiques et d'autres numériques. Cette évolution du système analogique simple permet d'introduire de nouvelles fonctions comme la détection de disparition d'objet et le comptage d'objets ou des personnes.
- Les caméras numériques consultables via Internet. On parle alors de caméra IP. Les caméras IP compressent numériquement les images et limitant leur flux (image /s), elles sont capables de produire des vidéos en haute résolution tout en minimisant la bande passante utilisée. En contrepartie, la visualisation ne peut être réalisée que par l'intermédiaire d'un ordinateur ou un décodeur.

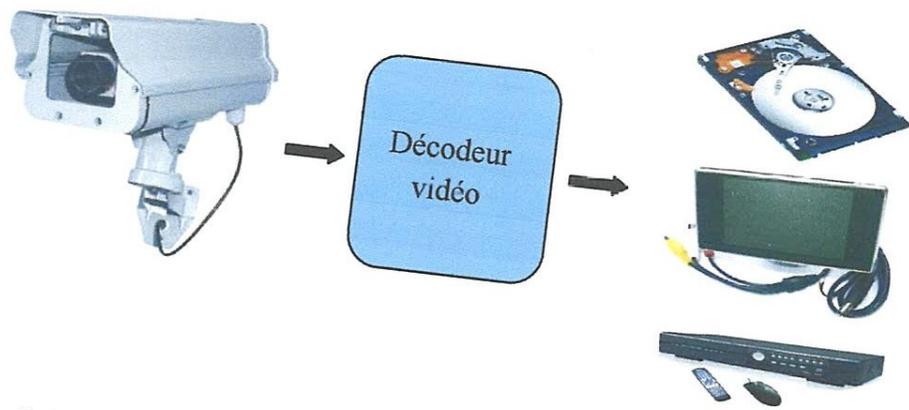


Figure 1.4 : système de caméra surveillance (caméras numériques consultables).

Les caméras IP, bien que plus coûteuses, sont l'avenir de la vidéo surveillance, car leur flexibilité et la standardisation des protocoles permettent le transport de l'image et du son à moindre coût par les réseaux intranet et internet existants. Dotées d'un processeur et d'un OS (operating system), elles sont capables de traiter en interne différents évènements tels que la détection des mouvements, de bruit, mais aussi de générer des actions comme le déclenchement d'une alarme, l'annonce d'un message... etc.

3.4. La détection des changements :

La détection des changements dans une séquence d'images constitue une source d'information très importante et voire même essentielle pour l'atteinte de plusieurs objectifs poursuivis par une vidéo surveillance automatique et intelligente. Dépendamment de l'application visée, la détection de changement peut varier et aller d'une forme simple vers des formes de plus en plus complexes. En effet, effectuer une simple différence temporelle entre les images successives d'une scène nous permet de nous donner une indication sur le taux de changement spatio-temporel de la séquence vidéo. Une différence temporelle nulle (bien que rare) représente un cas évident d'une scène statique où il ne se passe absolument rien. Par contre, une différence temporelle non-nulle peut être une indication des situations suivantes [6]:

- les changements de luminosité dans la scène (lumière on/off, lever du soleil...).
- les effets d'ajout de bruit lors de l'acquisition du signal.
- les mouvements de la caméra (translation, balayage, zoom,...).
- les mouvements dans la scène mais sans importance pour la vidéo surveillance (la végétation sous l'effet du vent, chute d'eau, fontaines, vagues d'eau,...).
- les mouvements d'objets pertinents pour la vidéo surveillance (les personnes qui se déplacent, les véhicules en circulation...).

Les approches de détection de changement dans les séquences d'images peuvent être classées en deux catégories [6]:

- approches basées sur la détection et l'analyse du fonds de scène.
- approches basées sur l'estimation du mouvement.

3.4.1. La détection du fonds de scène et analyse de l'avant de scène :

Une des approches classiques de détection des objets en mouvement dans une scène consiste dans une première étape, à identifier le fond de scène, puis, dans une seconde étape, de le soustraire de la scène entière afin d'isoler l'avant de scène.

Cette approche, communément appelée '*background subtraction*', est très efficace lorsqu'on est en présence d'objets en mouvement sur un fonds statique. Des extensions basées sur l'adaptation périodique du fond sont utilisées lorsque le fond de scène est dynamique (encas

de caméra en mouvement). Plusieurs techniques sont exploitées à cette fin allant de la modélisation gaussienne, aux méthodes de prédictions, aux différences temporelles et à l'estimation de mouvement. En général, les méthodes développées dans le cadre de cette approche sont capables de détecter les changements de lumière, les mouvements lents de la caméra, le bruit de la caméra, les objets à mouvements lents, les objets abandonnés ou disparus. Cependant, de grandes faiblesses sont constatées lorsque la scène contient des changements de lumière rapides ou brusques. Il s'agit d'une analyse de l'avant de scène effectuée une fois que le fond de scène est détecté. Le fond de scène est d'abord modélisé par une combinaison de gaussiennes. Des informations d'intensité et de texture sont intégrées afin d'éliminer les ombres et de s'adapter aux changements de lumières brusques ou rapides.

3.4.2. L'estimation du mouvement :

Dans les vidéos capturant des scènes complexes, le mouvement apparent des objets peut avoir des origines diverses. Car, en plus d'une origine due à un mouvement réel d'objets, les changements de l'illumination de la scène, le mouvement de la caméra, et les bruits introduits par les dispositifs électroniques d'acquisition du signal sont tous interprétés comme des changements apparents dans la séquence d'images. Par conséquent, un estimateur de mouvement classique aura tendance à détecter ces changements comme des mouvements apparents de la scène. Pour répondre à l'objectif de détection de changements dans une séquence de vidéo surveillance, il est important de distinguer ou bien d'identifier les objets en mouvement dans la scène (personnes, véhicules, etc.) et retenir uniquement les mouvements pertinents et ignorer les mouvements inintéressants ou inutiles tels que les effets du vent sur les branches d'arbres, les vagues d'eau, les fontaines, ...

L'approche de détection des changements par détection de mouvements pertinents s'avère plus efficace que l'approche de soustraction du fond de scène vu les faiblesses constatées pour cette dernière [6] :

- Nécessité d'une phase d'apprentissage avec de longues séquences vidéo.
- Difficulté de suivre les changements rapides.
- Non distinction entre les mouvements pertinents et les mouvements à caractère distrayant.

4. Les textures :

4.1. Qu'est-ce qu'une texture ?

Dans le domaine du traitement de l'image et de la vision, il n'existe pas de définition satisfaisante de la texture. Les définitions mathématiques construites à partir de propriétés statistiques sont soit trop générales et imprécises soit trop restrictives pour s'adapter à la diversité des cas rencontrés. Une texture est un champ de l'image qui apparaît comme un domaine cohérent et homogène, c'est-à-dire formant un tout pour un observateur. C'est cette propriété de cohérence de la texture placée dans son contexte d'être perçue comme un tout homogène par l'œil humain qui sera recherché le plus souvent par le traiteur des images, dans le but d'isoler les textures, soit pour segmenter l'image, soit pour reconnaître des régions.

Les capacités du système visuel humain à discerner les textures différentes sont remarquables et très mal expliquées. Nos capacités à mémoriser et discriminer des textures sont très variables. Elles deviennent excellentes dès lors que l'on a reconnu l'objet de la vie courante dont la texture est issue. Notre mémorisation est alors remarquable et invariante à de très nombreuses transformations (changement d'échelle, d'orientation, d'éclairage, de couleur...). Au contraire, des textures non reconnues (champs aléatoires créés mathématiquement par exemple, ou textures placées hors de leur contexte) sont mémorisées de façon très fugitive et sont peu robustes aux transformations [7].

Nos capacités à distinguer des textures différentes peuvent parfois s'expliquer par les capacités de filtrage du système visuel (bande passante des voies optiques, discrimination angulaire, sensibilité aux variations locales de luminosité).

L'une des remarquables capacités du système visuel est de reconnaître des textures déjà mémorisées même sous de très fortes distorsions géométriques (vues perspectives, recouvrement de surfaces gauches, etc.). Dans ces conditions, et même en l'absence d'un contexte favorable (par exemple présence de silhouettes d'objet ou d'alignements perspectifs), le système visuel interprète la déformation de la texture, non comme une perturbation des propriétés géométriques du champ texturé, mais comme une variation de la géométrie qui supporte la texture, celle-ci restant globalement invariante dans un repère imprécis. Ainsi un tronc d'arbre est vu comme une texture homogène enroulée sur un cylindre et non comme une texture plane dont les fréquences spatiales s'accroissent en approchant des bords [7].

4.2. Distribution des textures :

Une première constatation que l'on fait en examinant des textures naturelles est le rôle particulier que joue l'aléatoire dans la texture. On distingue assez naturellement deux modèles extrêmes de textures, entre les quels se positionnent un peu toutes les textures [7]:

- i. les textures régulières, dans lesquelles la périodicité du motif est évidente : grilles, murs, tissus, etc.
- ii. Les textures aléatoires pour les quelles la distribution des intensités n'est l'objet d'aucune régularité apparente : sable, nuages, herbe, foule.

La première famille sera bien décrite par des approches fréquentielles ou des approches structurelles dans les quelles on associera un motif et des règles de placement sur un pavage régulier. La seconde approche sera plutôt décrite par des lois statistiques, des moments, une description spectrale en termes de densité de puissance, des propriétés de corrélation.

4.3. Analyse et synthèse :

Un effort très important a été fait dans le domaine de la synthèse des images pour créer artificiellement ou copier des textures naturelles : forêts ou cultures pour des simulateurs de vols, bois, tissus, métaux pour des représentations d'intérieur, murs, toitures, eau, pour des jeux vidéo, etc. Ces travaux se distinguent notablement de ceux conduits en traitement d'image qui visent plutôt à extraire des paramètres discriminants et robustes permettant de séparer des textures différentes. Mais ces travaux se rejoignent en ce qu'ils contribuent simultanément à une meilleure connaissance et compréhension des textures à travers la boucle analyse / synthèse qui a fait ses preuves en reconnaissance des formes.

1.2. La Chromaticité :

La plupart des méthodes de détection d'ombre en fonction des caractéristiques spectrales utilisent les informations de couleur. Ils utilisent la supposition que les régions sous ombre deviennent plus foncées, mais conservent leur chromaticité. La chromaticité est une mesure de couleur qui est indépendante de la densité. Par exemple, après un pixel vert est couvert par l'ombre, il devient vert foncé, qui est plus foncé que le vert, mais à la même chromaticité [8]. Les méthodes qui utilisent ce modèle pour la détection des ombres choisissent souvent un espace de couleur avec une meilleure séparation entre la chromaticité et l'intensité (par exemple les espaces de couleurs RVB, HSV, YUV, normalisé RGB, $c_1c_2c_3...$), ou une combinaison entre eux.

La plupart de ces méthodes sont simples à mettre en œuvre et de calcul peu coûteux. Cependant, parce qu'ils font des comparaisons au niveau du pixel, ils sont sensibles au bruit et aux changements forts d'illumination (échouer avec les ombres foncées).

1.3. Les propriétés physiques :

Des travaux ont été réalisés pour créer des modèles plus généraux d'atténuation non linéaire, tenant compte des conditions d'éclairage différentes dans des scénarios à la fois intérieurs et extérieurs, parce que les méthodes qui tentent de modéliser ou d'apprendre l'apparition spécifique des pixels dans les ombres sont généralement dénommées approches physiques. Toutefois, étant donné qu'ils sont encore limités aux propriétés spectrales, leur principale inconvénient implique de traiter avec des objets ayant une chromaticité similaire à celle de l'arrière-plan.

1.4. La Géométrie :

En théorie, l'orientation, la taille et même la forme d'ombre peuvent être prédites avec une bonne connaissance de la source d'éclairage, la forme de l'objet et le plan de masse. Certaines méthodes utilisent cette information pour couper les ombres des objets. Le principal avantage des caractéristiques géométriques, c'est qu'ils travaillent directement dans la trame d'entrée et, par conséquent, ils ne reposent pas sur une estimation précise de la référence d'arrière-plan [9]. Cependant, les méthodes qui utilisent des fonctionnalités géométrie imposent des limitations de scène tels que: les objets de types spécifiques, généralement des piétons (personnes debout) ou des véhicules.

1.5. Les Textures :

Certaines méthodes exploitent le fait que les régions sous l'ombre conservent la plupart de leur texture. Les méthodes de détection d'ombres à base de texture suivent généralement deux étapes : la sélection des pixels ou des régions candidats à être ombrés, et la classification des pixels ou des régions candidates que ce soit au premier plan ou ombre, selon la corrélation de la texture. La sélection des candidats se fait avec un détecteur d'ombre faible, généralement basé sur les caractéristiques spectrales. Puis, chaque candidat est classé soit comme objet ou ombre en corrélant la texture dans l'image avec la texture dans l'arrière-plan de référence [8].

Si la texture d'un candidat est similaire à la fois à l'image et l'arrière-plan, elle est classée comme ombre. La corrélation de texture est une méthode potentiellement puissante pour détecter des ombres en tant que textures fortement distinctives, ne dépend pas des couleurs, et robustes aux changements d'illumination. Cependant, les méthodes de détection d'ombres à base de texture ont tendance à être lentes (calculer une ou plusieurs comparaisons de voisinage pour chaque pixel).

2. Les dernières méthodes de détection d'ombre :

Les méthodes de détection d'ombre publiées au cours des dernières années sont classées en fonction de leur choix de caractéristique.

Comme il est mentionné précédemment, les caractéristiques d'intensité sont utilisées principalement comme une première étape pour la détection des ombres, et les caractéristiques temporelles sont principalement utilisées pour filtrer les résultats de détection. Par conséquent, toutes les méthodes examinées ont été classées en quatre catégories: les méthodes basées sur la chromaticité, les méthodes basées sur les propriétés physiques, les méthodes basées sur la géométrie et les méthodes basées sur la texture. La taxonomie est détaillée dans les tableaux suivants [9]:

Les méthodes basées sur la Chromaticité				
Article	L'espace de couleur	Niveau	Vérification spatiale	vérification Temporal
Cucchiara et al. 2003	HSV	Pixel	—	—
Salvador et al. 2004	c1c2c3	Window	Pas d'ombre interne	—
Cavallaro et al. 2005	Normalised RGB	Pixel	Pas d'ombre interne	Suivi la durée de vie
Chen et al. 2010	YUV	Pixel	Morphologie	—
Sun and Li 2010	HSI et c1c2c3	Pixel	Morphologie	—

Tableau 1.1 : les méthodes basées sur la Chromaticité publiées au cours des dernières années.

Les méthodes basées sur les propriétés Physique			
Article	Modèle	Apprentissage	indices spatiale ou temporelle
Nadimi and Bhanu 2004	Dichromatique	—	Spatio-temporelle test
Porikli and Thornton 2005	—	flux de l'ombre	—
Liu et al. 2007	—	Gaussian mixture model	Markov champs aléatoires et le suivi
Martel-Brisson and Zaccarin 2007	—	Gaussian mixture model	—
Martel-Brisson and Zaccarin 2008	Général	Kernel based	Gradients (direction)
Joshi and Papanikolopoulos 2008	—	Semi supervised (SVM)	Edges (Les bords)
Huang and Chen 2009	Général	Gaussian mixture model	Gradients (atténuation)

Tableau 1.2 : les méthodes basées sur les propriétés physiques publiées récemment.

Les méthodes basées sur la Géométrie				
Article	Objets	Segmentation	Indice principal	Autres indices
Hsieh et al. 2003	Personnes	Par détection des têtes	Orientation	L'intensité et la localisation
Yoneyama et al. 2003	Véhicules	—	2D modèles	Point de fuite

Nicolas and Pinel 2006	N'importe	—	Source de lumière	filtre temporel
Fang et al. 2008	Véhicules	—	Transformation d'onde	Spectral
Chen and Aggarwal 2010	Personnes	—	Coordonnées Log-polaires	Couleurs et gradients orientés

Tableau 1.3 : les méthodes basées sur la Géométrie publiées récemment.

Les méthodes basées sur les propriétés des Textures			
Article	Détecteur faible	Corrélation de texture	Niveau de corrélation
Javed and Shah 2002	Segmentation de couleur	Direction de gradient	Region moyen
Xu et al. 2005	—	Corrélation de statique de bord	Pixel
Tian et al. 2005	Gamme d'intensités	Normalisé du corrélation croisée	Pixel
Wang et al. 2006	—	Intensité et bordure DCRF filter	Petite region
Zhang et al. 2006	—	Transformations orthogonales	Petite region
Leone and Distanto 2007	Le gain photométrique	Filter de Gabor	Petite region
Zhang et al. 2007	Contrainte d'intensité	Test par rapport aux bordures	Petite region
Nghiem et al. 2008	Basé sur Chromaticité	Rapport de réduction d'intensité	Petite region
Shoalb et al. 2009	—	Dégradé de soustraction de fond	Pixel
Pei and Wang 2009	—	Basé sur PCA	Petite region
Sanin et al. 2010	Basé sur Chromaticité	Direction de gradient	Large region
Nakagami and Nishitani 2010	—	Walsh, transformation de domaine	Petite region
Panicker and Wilscy 2010	—	Détection de contours premier plan	Pixel
Qin et al. 2010	Modèle de couleur d'ombre	Local motif ternaire MRI	Petite region

Tableau 1.4 : les méthodes basées sur les propriétés des Textures publiées récemment.

Les méthodes sont classées par année dans la première colonne et regroupées par catégorie. Les colonnes supplémentaires montrent les classifications secondaires au sein de chaque catégorie.

3.1.1. Les résultats quantitatifs :

Pour tester les performances de détection d'ombre des méthodes, ils calculent les deux métriques proposées par *Prati et al* [10], le taux de détection des ombres à savoir (η) et le taux de discrimination ombre (ξ), utilisent les formules (7) et (8):

$$\eta = \frac{TP_S}{TP_S + FN_S} \quad (7)$$

$$\xi = \frac{TP_F}{TP_F + FN_F} \quad (8)$$

où TP et FN se présenter aux vrais positifs et des faux négatifs pixels par rapport à soit ombres (s) ou objets au premier plan (F).

Le taux de détection des ombres est préoccupé par l'étiquetage du maximum nombre de pixels d'ombre portée qu'ils sont des ombres. Le taux de discrimination est l'ombre préoccupé par le maintien des pixels qui appartiennent à l'objet en mouvement comme au premier plan. Ici, ils utilisent souvent la moyenne des deux taux comme une mesure de performance unique.

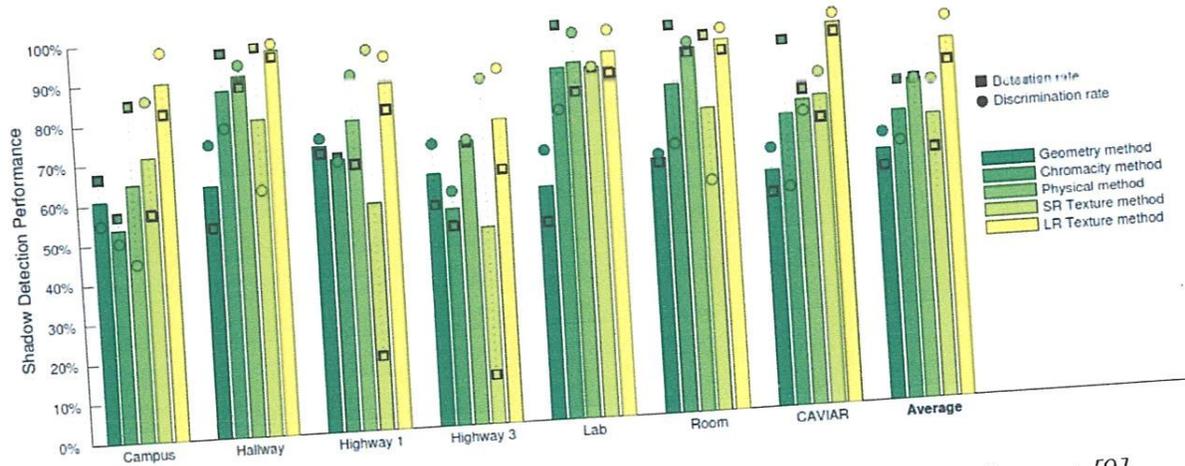


Figure 1.6 : Comparaison des résultats de la détection d'ombre par la séquence [9].

La mesure du rendement est la moyenne entre la détection des ombres et des taux de discrimination. Chaque barre représente la performance moyenne d'un segment vertical dont la longueur indique la différence moyenne entre le taux de détection (marqué d'un carré) et le taux de discrimination (marqué d'un cercle).

Cette figure montre la détection des ombres moyenne et le taux de discrimination sur chaque séquence de test. Chaque barre représente la moyenne des taux de détection et de discrimination sur chaque séquence, tandis que la détection et le taux de discrimination sont indiqués par un carré et un cercle, respectivement.

Des tests sont effectués (dans des lieux déferents : campus, couloir, route, chambre, laboratoire...) pour estimer le délai moyen de traitement par image de chaque méthode de détection d'ombre, les résultats sont montrés dans le tableau 1.5.

Les temps sont montrés en millisecondes, obtenus sur un processeur 32 bits Intel cadencé à 2,6 GHz. La méthode basée sur chromaticité était le plus rapide à mettre en œuvre et exécuter. Les méthodes basées sur la géométrie et les propriétés physiques sont besoin de plus de opérations pour le calcul des moments centraux et la mise à jour des modèles d'ombre, respectivement. La méthode basée sur grande région de texture nécessite des mesures supplémentaires afin de générer des zones d'ombre candidats et calculer des gradients pour chaque pixel, c'est la plus lent méthode parmi ces quatre.

	Chromacity	Geometry	Physical	SR Textures	LR Textures
Campus	8.72	9.44	10.00	156.46 (48.81)	20.76
Hallway	11.28	8.91	12.81	223.64 (77.37)	21.59
Highway1	10.73	24.75	16.93	341.32 (116.28)	34.71
Highway3	6.82	6.49	7.15	120.36 (37.07)	11.75
Lab	8.95	17.68	15.34	253.82 (82.48)	22.73
Room	7.14	8.41	8.51	144.87 (47.07)	16.25
Caviar	10.82	13.94	14.08	243.07 (82.98)	24.70
Average	9.21	12.80	12.12	211.93 (70.30)	21.78

Tableau 1.5: Délai moyen de traitement de trame (en millisecondes) par séquence pour les différentes méthodes de détection d'ombre [9].

3.1.2. Les résultats qualitatifs :

Observations spécifiques peuvent également être réalisées pour chaque méthode. La méthode basée sur la géométrie ne fonctionne que lorsque chaque ombre a une orientation unique qui diffère de l'orientation de l'objet, comme c'est le cas dans l'exemple du campus,

mais échoue lorsque les ombres ont les mêmes orientations que les objets ou quand les ombres ont des directions multiples comme dans le laboratoire par exemple. La méthode basée sur chromaticité est affectée dans le cas de bruit ou niveau-pixel, et échoue lorsque les objets sont plus foncés et ont des couleurs similaires à l'arrière-plan comme dans l'exemple du campus. Bien que la méthode physique utilise également des caractéristiques de chromaticité, il a la capacité de s'adapter à chaque scène et il est moins susceptible d'échouer lorsque la méthode fondée sur chromaticité échoue.

La méthode basée sur la texture (petites régions) fonctionne bien dans les scènes avec des fonds texturés, mais échoue pour des pixels situés dans des quartiers non texturés, par contre dans le cas de grande région de texture fonctionnent bien dans la plupart des cas, même si elle peut fausser les contours de l'objet.

Les résultats de deux expériences quantitatives et qualitatives des observations sont résumés dans le tableau 1.6. Ils attribuent des notes à chaque méthode en fonction de plusieurs critères. La méthode basée sur la géométrie a des hypothèses fortes concernant l'objet et le forme de l'ombre, mais quand ces hypothèses sont remplies la méthode fonctionne bien de façon indépendante sur la qualité des caractéristiques spectrales et la texture. La méthode basée sur chromaticité est simple et rapide, ses hypothèses peu l'amener à fonctionner raisonnablement dans la plupart des scènes. Toutefois, elle a un fort compromis entre la détection d'ombre et de la discrimination. La méthode physique réduit les limites de la méthode basée sur chromaticité, à condition qu'il existe suffisamment d'exemples pour apprendre et s'adapter à l'apparition de déplacement d'ombre portée dans l'arrière-plan. Cependant, comme la méthode basée sur chromaticité, elle est sensible au bruit dans le niveau-pixel et les scènes avec de faibles couleurs saturées. La méthode basée sur la texture de petite région est robuste aux conditions d'éclairage différentes et est facile à mettre en œuvre, mais elle nécessite un arrière-plan pour être texturée et a besoin d'une coûteuse opération de texture de corrélation par pixel.

La méthode basée sur grande région de texture n'est pas sensible au bruit dans le niveau pixel, et elle est indépendante du type des ombres, des objets et des scènes. Elle présente les meilleurs résultats au coût de la charge de calcul supplémentaire.

	Geometry	Chromacity	Physical	SR Textures	LR Textures
1. Scene independence	medium	medium	medium	medium	high
2. Object independence	low	high	high	high	high
3. Shadow independence	low	high	high	high	high
4. Penumbra detection	medium	low	medium	high	high
5. Robustness to noise	medium	low	medium	high	high
6. Detection/discrimination trade-off	low	high	medium	high	low
7. Computational load	low	low	low	high	medium

Tableau 1.6: Résumé de l'évaluation comparative [9].

Les méthodes de détection d'ombre sont classées comme faibles, moyennes ou élevées selon sept critères. Pour les cinq premiers critères, élevé indique que la méthode donne des bons résultats dans tous les cas. Une méthode à faible scène, objet ou ombre indépendance, ne fonctionne bien que pour un ensemble restreint des scènes, des objets et des types d'ombres. Faible détection de la pénombre à signifie que la méthode n'est pas appropriée pour la détection des frontières d'ombre. Faible robustesse au bruit signifie que les performances de détection d'ombre sont significativement affectées par le bruit de la scène. Pour les deux derniers critères (c.-à-d.6 et 7) inférieure signifie mieux.

Conclusion :

Dans ce chapitre nous avons présenté les méthodes de détection d'ombre et les travaux publiés au cours des dernières années. On a trouvé que ces méthodes sont classées en fonction de leur choix de caractéristiques. Comme il est mentionné précédemment, les caractéristiques d'intensité sont utilisées principalement comme une première étape pour la détection des ombres, et les caractéristiques temporelles sont principalement utilisées pour filtrer les résultats de détection. Par conséquent, toutes les méthodes examinées ont été classées en quatre catégories: les méthodes basées sur la chromaticité, les méthodes basées sur les propriétés physiques, les méthodes basées sur la géométrie et les méthodes basées sur la texture. Nous avons aussi présenté une comparaison entre des méthodes de chaque catégories, d'où la conclusion que les méthodes basées sur la texture sont les plus performantes.

Chapitre 2

Conception & Implémentation



Chapitre 2 : Conception & Implémentation

Introduction :

La capacité d'extraire des objets en mouvement à partir d'une séquence vidéo est un problème crucial dans des nombreux systèmes de vision qui incluent le suivi d'objets dans des contextes de vidéo surveillance. La tâche de traitement d'image n'est pas la détection précise des détails de l'objet, mais la détection robuste des formes d'objets en mouvement. Concernant notre travaille, la segmentation de la scène permet la suppression des ombres qui sont générées à la suite des sources d'éclairage qui produisent des formes de distorsion si elles tombent sur les objets rencontrés. En outre dans les scènes dynamiques, tous les points mobiles des objets ou des ombres sont détectés en même temps, les ombres et les objets partagent les mêmes caractéristiques visuelles de mouvement et de détectabilité. Pour ces raisons, une méthode simple, fiable et précise est nécessaire pour identifier les ombres et les différenciés par rapport aux objets.

Comme il est mentionné dans le premier chapitre, les méthodes les plus puissantes sont les méthodes basées sur la chromaticité (détection précise d'ombre) et les méthodes basées sur les textures larges (bonne distinction entre les objets et les ombres). Tout d'abord dans ce chapitre, nous présentons une méthode basée sur les textures larges, proposée par *Andres Sanin et al* [9], puis nous améliorons cette méthode on se base sur les trois espaces de couleur RGB, HSV et YUV dans l'étape de détection des pixels candidates d'ombre. Mais comme il est mentionné précédemment, les méthodes basées sur les textures larges sont lourdes. Pour cela, nous avons proposé et implémenté notre propre approche basée sur la chromaticité. Dans chaque frame, nous utilisons le filtre de *Canny* pour détecter les contours existants, puis à partir ces contours nous décomposent les objets, détectés en mouvement, en des régions. Par la suite, nous classons chaque région comme ombre ou pas.

I. Architecture globale du système :

De nombreuses applications de vision par ordinateur, relatives avec les vidéo-surveillance, nécessitent la détection et le suivi des objets mobiles. Cependant, les techniques actuelles ont généralement un problème majeur, les ombres qui partagent les mêmes habitudes des déplacements et ont une ampleur similaire du changement d'intensité que celle des objets en premier plan.

L'objectif de notre système est de détecter et d'extraire (éliminer) les ombres dans des séquences de vidéo-surveillance. La figure suivante montre l'architecture globale du système :

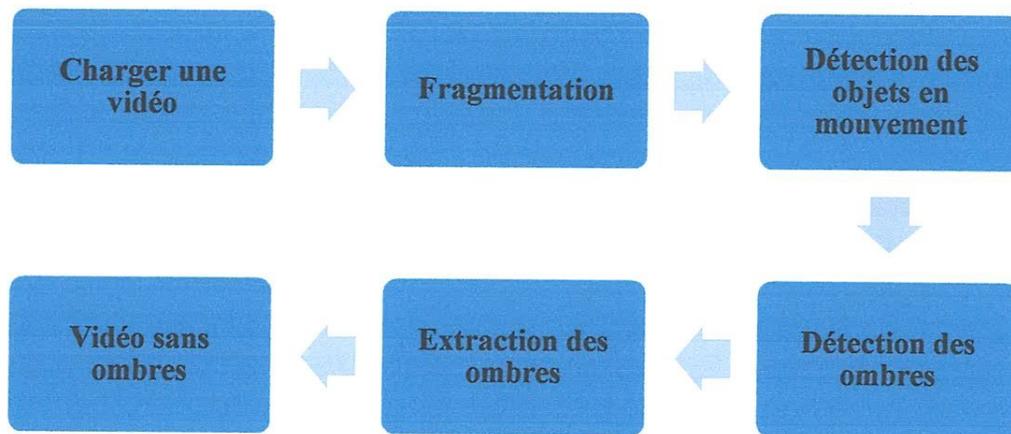


Figure 2.1 : schéma global du système.

La partie principale du système est la quatrième, qui consiste à détecter et à éliminer les ombres. Dans cette partie, nous avons montré les limites d'une méthode, basée sur les textures larges, puis nous avons amélioré cette méthode. Par la suite nous présentons notre propre approche, basée sur la chromaticité, le schéma suivant montre les étapes d'implémentation de cette approche :

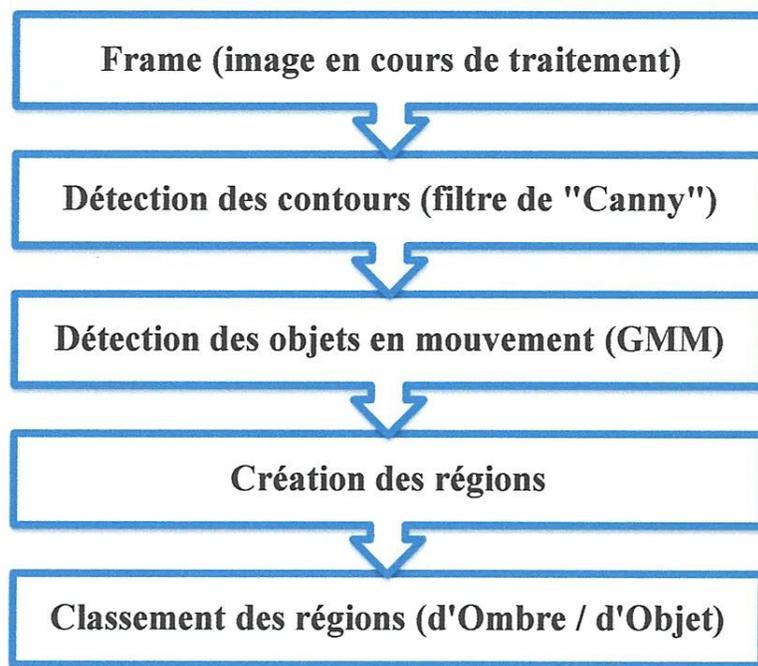


Figure 2.2 : les étapes d'implémentation de notre approche.

Dans la première étape, nous utilisons le filtre de Canny pour détecter les contours existant dans l'image. Par la suite on fait une soustraction entre le masque du premier plan (obtenu avec GMM) et le masque des contours (obtenu avec le filtre de Canny) pour créer des régions, puis nous testons chaque région si elle est une région d'ombre ou pas.

II. Les Mixtures de Gaussien (GMM) :

Nous avons utilisé les mixtures de Gaussien pour segmenter les objets directement à partir de l'image. Cependant, les ombres ont tendance à être classées dans le cadre du premier plan parce qu'ils partagent les mêmes caractéristiques et les mêmes ordres de grandeur de changement d'intensité que celle des objets du premier plan.

Manuellement, l'étiquetage des pixels d'ombre dans une image est beaucoup plus difficile que l'étiquetage des pixels d'objet. Par exemple, il peut être facile de déterminer quels pixels appartiennent à des personnes dans le cadre d'une image, mais il est difficile de dire quels sont les pixels correspondants à leurs ombres. Pour cette raison, nous avons d'abord utilisé le GMM pour extraire un masque avant-plan sur chaque image.

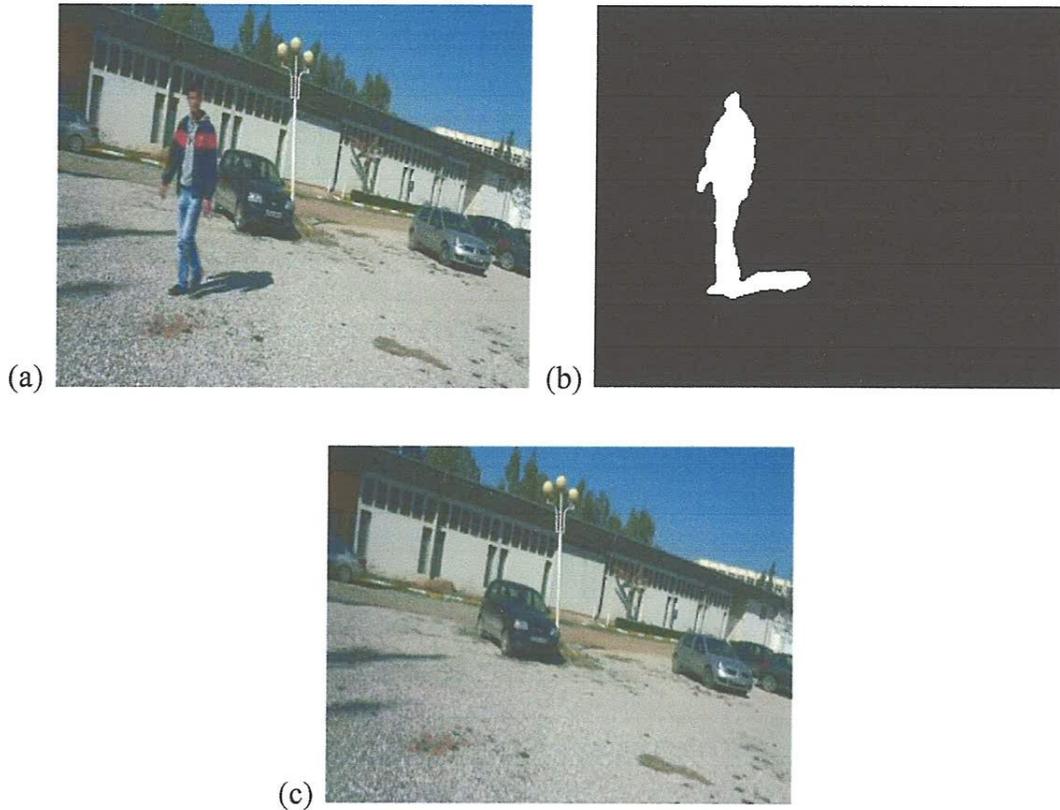


Figure 2.3: Étapes de création: (a) Image originale (b) Masque de premier plan obtenu avec GMM, (c) Masque d'arrière-plan obtenu avec GMM.

III. Implémentation du système :

1. La méthode basée sur les textures larges :

Andres Sanin et al [9], ont proposé une méthode utilisant les caractéristiques couleurs pour créer des grandes zones candidats d'ombre (idéalement contenant l'ensemble des zones d'ombre), puis les diviser en régions (chacune contient des pixels d'ombre ou des pixels d'objets), en suite ils utilisent la corrélation basée sur la texture et la direction de gradient pour discriminer ces régions (que soit région d'ombre ou d'objet).

Le but de la première étape est de créer des régions les plus grandes possible (une région est un ensemble des pixels, chaque région doit représenter une partie d'un objet ou bien d'une ombre) qui contiennent des pixels d'ombre ou des pixels d'objets, mais pas les deux. Les pixels candidats d'ombre sont trouvés par l'utilisation des caractéristiques d'intensité et de chromacité dans l'espace de couleur HSV, mais ils adaptent des seuils pour assurer la précision de la détection.

Le bon classement des pixels d'ombre est important dans cette étape, puisqu'il va imposer la limite supérieure de la précision de la détection finale. Mauvaise classification des pixels d'objet comme des ombres n'est pas un problème, car ils sont plus tard discriminés par l'utilisation des fonctions de texture, la texture pour chaque région est corrélée entre l'image et l'arrière-plan de référence. De plus, les ombres ont tendance à préserver les textures sous-jacentes (les zones d'ombre devraient avoir une haute corrélation de texture).

A ce stade, ils vont distinguer entre les régions d'ombre et les régions d'objets, ils se basent sur le principe de gradient. Dans cette étape, la texture des régions candidates est corrélée entre l'image et l'arrière-plan de référence. De plus les zones d'ombre devraient avoir une haute corrélation de texture.

Premièrement, ils calculent l'amplitude de gradient (\mathbf{P}) et la direction (\mathbf{D}) de chaque région candidate, dont chaque pixel $p(x, y)$ (ils utilisent les formules (9) et (10):

$$|\mathbf{P}| = \sqrt{P_x^2 + P_y^2} \dots (9)$$

$$\mathbf{D} = \arctan(P_x / P_y) \dots (10)$$

Où P_y est le gradient vertical (différence d'intensité entre le pixel $P(x,y)$ et le pixel dans la rangée suivante), P_x est le gradient horizontal. La fonction $\arctan()$ retourne un angle dans la gamme angulaire $[-\pi ; \pi]$, permettant à la direction du gradient d'être traité comme une véritable variable circulaire. Seulement les pixels avec \mathbf{P} supérieure à un certain seuil sont pris en compte afin d'éviter les effets du bruit, qui sont plus forts dans les régions lisses de l'image.

En outre, la direction du gradient est une variable circulaire, la différence doit être calculée comme une distance angulaire. Pour chaque pixel $P(x,y)$ qui a été sélectionnée en raison d'une grande ampleur, la différence de direction de gradient entre le frame (l'image) F et le fond (l'arrière-plan) de référence B est calculée en utilisant la formule (11):

$$GDif = \frac{P_x^F P_x^B + P_y^F P_y^B}{\sqrt{(P_x^F^2 + P_y^F^2)(P_x^B^2 + P_y^B^2)}} \dots (11)$$

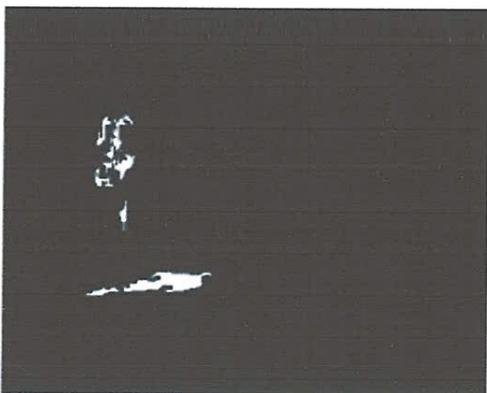
La corrélation de direction du gradient entre l'image et l'arrière-plan est estimée en utilisant la formule (12) :

$$C = \frac{\sum_{p=1}^n H(T_a - GDif)}{n} \dots (12)$$

Où n est le nombre des pixels sélectionnés dans la région candidate d'ombre, $H ()$ est la fonction qui renvoie 1 si la différence angulaire est inférieure ou égale au seuil T_a , et 0 sinon. En substance, C est la fraction des pixels dans la région dont la direction de gradient est similaire dans l'image et l'arrière-plan (le fond). Si C est supérieur au seuil T_c , la région candidate est considérée comme une région d'ombre et elle est retirée du masque d'avant-plan.



(a)



(b)



(c)

Figure 2.4: (a) l'image originale (b) Masque des pixels candidates d'ombre obtenue après l'application de L'approche basée sur les textures larges (c) l'image sans ombre.

2. La méthode basée sur les textures larges améliorée :

Nous avons améliorée la méthode basée sur les textures larges, présentée précédemment, nous utilisons les trois espaces de couleur (RGB, HSV et YUV), pour extraire les pixels pouvant être des pixels d'ombre (première étape de la méthode). Généralement les approches basées sur la chromaticité posent des problèmes lorsqu'il y a des zones d'objets qui ont une couleur similaire ou bien proche à la couleur d'ombre, par la suite elles vont être considérées comme des zones d'ombre et forcément seront éliminées. Pour éviter ce problème, nous avons utilisé la même méthode basée sur les textures larges pour discriminer les zones d'ombre.

a. Etape 1 : L'approche basée sur la chromaticité :

Le facteur le plus important pour les méthodes basées sur la chromaticité, est de choisir un espace de couleur avec une séparation de l'intensité et de la chromaticité. Plusieurs espaces colorimétriques tels que HSV, YUV... sont révélés être robuste pour la détection d'ombre. Nous avons proposé une approche, dans laquelle nous utilisons les trois espaces de couleurs RGB, HSV et YUV parce que chacun a un avantage, étant donné que l'espace de couleur YUV offre une séparation naturelle entre la chromaticité et la luminosité.

Après l'extraction des objets en mouvement (avec le GMM), nous calculons le pourcentage de changement (dans chaque pixel appartenant au masque de premier plan) entre sa valeur dans l'arrière-plan et sa valeur dans l'image en cours de traitement, dans les trois espaces :

- L'espace RGB : nous calculons le pourcentage de changement dans les valeurs R, G et B (nous utilisons les formules 13, 14 et 15 respectivement):

$$\% \text{Changement}_{de R} = (R_{bg} - R_{frame}) * 100 / R_{bg} \dots (13)$$

$$\% \text{Changement}_{de V} = (G_{bg} - G_{frame}) * 100 / V_{bg} \dots (14)$$

$$\% \text{Changement}_{de B} = (B_{bg} - B_{frame}) * 100 / B_{bg} \dots (15)$$

Où (R_{bg} , G_{bg} , B_{bg}) et (R_{frame} , G_{frame} , B_{frame}) représentent les valeurs des composants des pixels dans l'arrière-plan (bg) et l'image en cours de traitement (frame) respectivement.

- L'espace HSV : après la conversion de l'arrière-plan et le Frame en espace HSV, nous calculons les changements dans les valeurs H, S et V entre l'arrière-plan et le Frame (dans tous les pixels détectés), nous utilisons les formules 16, 17 et 18 :

$$Hdif = Hframe - Hbg \dots (16)$$

$$Sdif = Sframe - Sbg \dots (17)$$

$$Vdif = Vframe - Vbg \dots (18)$$

Où (Hbg , Sbg , Vbg) et ($Hframe$, $Sframe$, $Vframe$) représentent les valeurs des composants des pixels dans l'espace HSV, dans l'arrière-plan (bg) et l'image en cours de traitement (frame) respectivement.

- L'espace YUV : nous utilisons les formules (4, 5 et 6) mentionnées dans le chapitre 1 pour convertir l'arrière-plan et le Frame dans l'espace YUV, en suite, nous calculons le pourcentage de changement dans la valeur Y (représente l'intensité) entre l'arrière-plan et le Frame dans tous les pixels (nous utilisons la formule 19):

$$Ydif = (Ybg - Yframe) * 100 / Ybg \dots (19)$$

Où Ybg et $Yframe$ représentent la valeur de Y du pixel dans l'espace YUV, dans l'arrière-plan et l'image en cours de traitement (frame) respectivement.

Puisque la valeur V de l'espace HSV et la valeur Y de l'espace YUV sont les mesures directes de l'intensité et de la lumière respectivement, les pixels dans l'ombre devraient avoir des valeurs inférieures par rapport aux pixels de l'arrière-plan. D'après les propriétés de chromaticité, une ombre sur le fond ne change pas sa teinte (H) et les ombres souvent baissent la saturation (S) des points. Pour les pixels foncés, qui peuvent être générés par l'ombre, les valeurs chromatiques U et V sont normalisées par la valeur Y pour créer une mesure d'illumination invariant. Toutefois, cette mesure n'est pas totalement l'éclairage invariant, car les valeurs RVB peut être convertie en YUV. Par conséquent, un pixel est considéré comme faisant partie de l'ombre si les conditions suivantes sont satisfaites :

$$Ydif > P_1 \text{ et } Ydif < P_2 \dots (I)$$

$$Hdif < P_3 \text{ et } Sdif < P_4 \dots (II)$$

$$Y-r < 3 \text{ et } Y-v < 3 \text{ et } Y-b < 3 \dots (III)$$

Où P_1 , P_2 , P_3 et P_4 , représentent des seuils qui sont optimisés de façon empirique.

La figure 2.5, montre les candidates pixels d'ombre (les pixels en blanc) qu'ils satisfont les conditions (I), (II) et (III) :

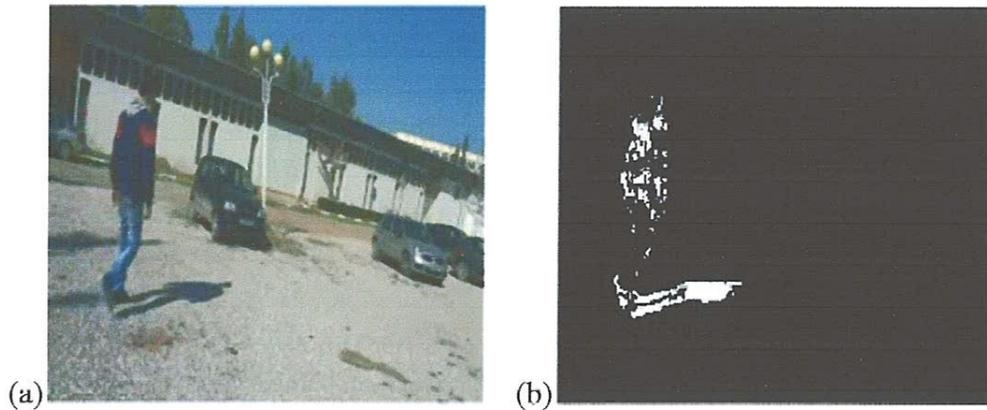


Figure 2.5: (a) Image originale; (b) Masque des candidates pixels d'ombre (les pixels en blanc).

Remarque : le masque des pixels candidats d'ombre obtenue par l'utilisation des propriétés de la chromaticité contient des pixels d'objets ayant des couleurs proches ou bien similaires aux couleurs d'ombre.

b. Etape 2 : L'approche basée sur les textures larges :

Dans cette étape, on va passer comme paramètre le masque qui contient les pixels candidats d'ombre (obtenu avec l'approche basée sur la chromaticité dans la première étape), puis on suit les mêmes étapes de la méthode basée sur les textures larges, lesquelles nous avons parlé précédemment, pour discriminer entre les zones d'ombre et les zones d'objet. Les régions qui ont été trouvées appartiennent aux objets sont colorées en vert (figure 2.6).



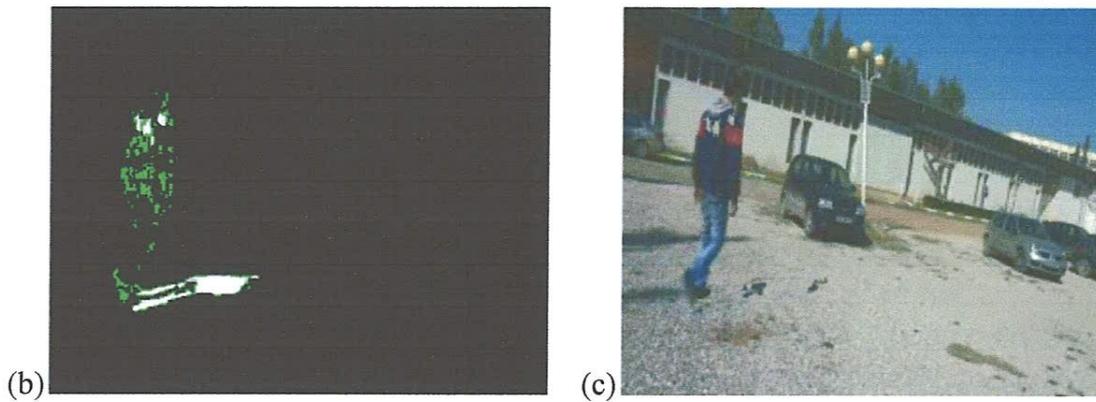


Figure 2.6 : (a) l'image originale, (b) Masque des pixels candidates d'ombre obtenue après l'application de L'approche basée sur les textures larges, (c) l'image sans ombre.

3. Approche proposée :

Dans cette approche qu'on a proposée, on se base seulement sur la chromaticité (nous utilisons toujours les trois espaces de couleur RGB, HSV et YUV). Dans la première étape, nous utilisons le filtre de *Canny* pour détecter les contours existant dans l'image. Par la suite on fait une soustraction entre le masque du premier plan (obtenu avec GMM) et le masque des contours (obtenu avec le filtre de Canny) pour créer des régions, puis nous testons chaque région si elle est une région d'ombre ou pas.

3.1. Le filtre de "Canny" :

Dans cette étape, nous utilisons le filtre de Canny (d'*Open-CV 2.0*). Le détecteur de *Canny* est utilisé en traitement d'images pour la détection des contours. L'auteur l'a conçu pour être optimal suivant trois critères clairement explicités [11] :

- bonne détection : faible taux d'erreur dans la signalisation des contours,
- bonne localisation : minimisation des distances entre les contours détectés et les contours réels,
- clarté de la réponse : une seule réponse par contour et pas de faux positifs.

La figure ci-dessous montre le masque des contours obtenu avec le filtre de Canny :

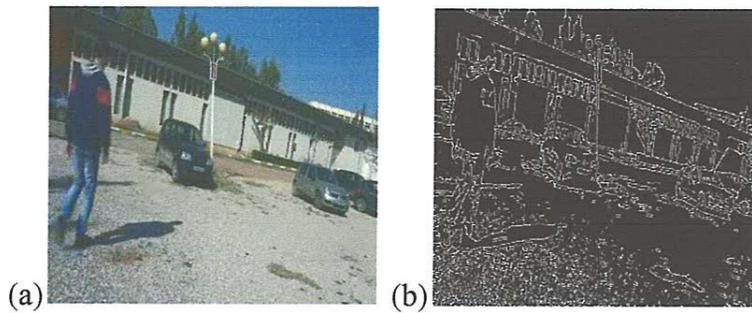


Figure 2.7: (a) l'image originale (b) Masque des contours obtenus avec le filtre de Canny.

3.2. Création des régions :

Après avoir obtenu le masque des contours avec le filtre de Canny, on fait une soustraction entre ce masque et le masque du premier plan pour décomposer les objets en mouvement en des régions :

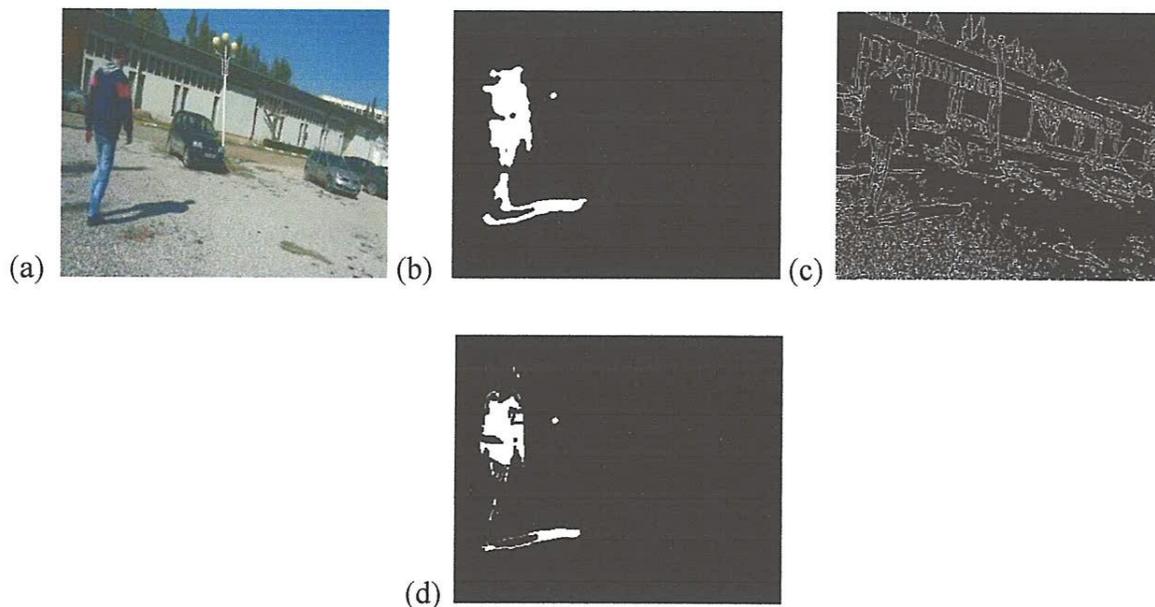


Figure 2.8: (a) l'image originale, (b) Masque du premier plan obtenu avec GMM, (c) Masque des contours obtenu avec le filtre de Canny, (d) régions créer.

Après, nous testons chaque région si elle est une région d'ombre ou pas. Une région est considérée comme une région d'ombre si le pourcentage des pixels (de cette région) qu'ils satisfont les conditions (I), (II) et (III) (mentionner précédemment) est supérieur à un seuil T . (entre 80% et 85%).

La figure suivante montre les régions détectées comme des régions d'ombre :

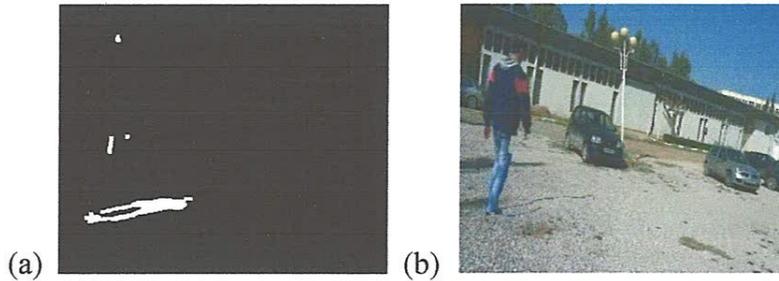


Figure 2.9: (a) régions détectées comme des régions d'ombre, (b) image sans ombre.

IV. Extraction et élimination des ombres :

Après avoir détecté les pixels d'ombre dans l'étape de détection (les pixels en blanc du masque d'ombre), on doit remplacer les valeurs des pixels détectés comme ombre par les valeurs correspondantes dans l'arrière-plan.

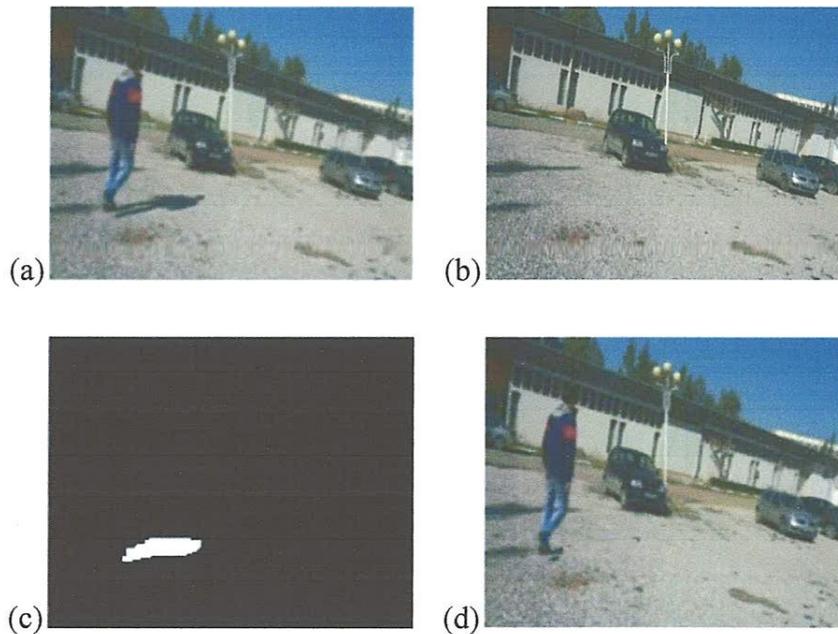


Figure 2.10: (a) image original, (b) l'arrière plan, (c) Masque des pixels d'ombre, (d) l'image sans l'ombre.

Conclusion :

Dans ce chapitre, nous avons présenté une méthode basée sur les textures larges, proposée par Andres Sanin et al, puis nous avons amélioré cette méthode en se basant sur les trois espaces de couleur RGB, HSV et YUV dans l'étape de détection des pixels candidats d'ombre. En suite, nous avons proposé et implémenté notre propre méthode basée sur la chromaticité dans laquelle le filtre de *Canny* a été utilisé pour détecter les contours existants dans le frame. À partir de ces contours nous avons décomposé les objets, détectés en mouvement, en plusieurs régions. Par la suite, un classement est effectué pour décider si la région est une ombre ou pas.

Dans le chapitre suivant, nous allons réaliser une étude comparative entre les trois méthodes pour tester les performances de chacune d'entre elles.

Chapitre 3

Tests & résultats



Chapitre 3 : Tests & résultats

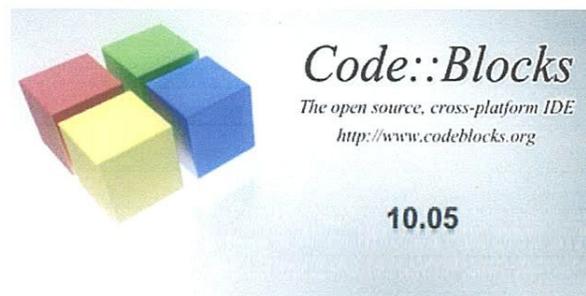
Introduction :

Dans ce chapitre, on va présenter les outils utilisés pour la réalisation du système (Code-Blocks, Qt Creator, Open-CV 2.0), et l'interface du système.

En outre, nous réalisons une séquence des tests pour comparer la performance des trois méthodes implémentées dans le deuxième chapitre. Dans ces tests nous tenons compte de trois critères (la performance de détection d'ombre, la discrimination entre les pixels d'ombre et les pixels d'objets, et le temps d'exécution) pour faire la comparaison et pour montrer les avantages de chaque méthode.

1. Outils utilisés :

1.1. Code-Blocks :



Code-Blocks est un environnement de développement intégré (un IDE est un logiciel permettant au développeur d'écrire, compiler et déboguer son programme), libre, gratuit et multi plate-forme. Il est écrit en C++ et il est développé pour Linux, Windows et Mac, code-blocks est orienté C et C++, mais il supporte d'autres langages.

La première version stable a vu le jour le 28 février 2008, portant le numéro de version **8.02**, où le premier et le second nombre représentent respectivement l'année et le mois de la sortie. A ce jour, la version **10.05** est la plus récente et la plus stable. Plusieurs compilateurs supportés : GCC, MinGW (la version Win32 de GCC), Borland C++ 5.5 ..., en outre code-blocks nous a permis d'importer des projets Dev-C++. Pour l'installation il suffit d'installer le paquet code-blocks [12].

L'illustration ci-dessous montre l'apparence de la fenêtre de l'interface utilisateur de Code-Blocks.

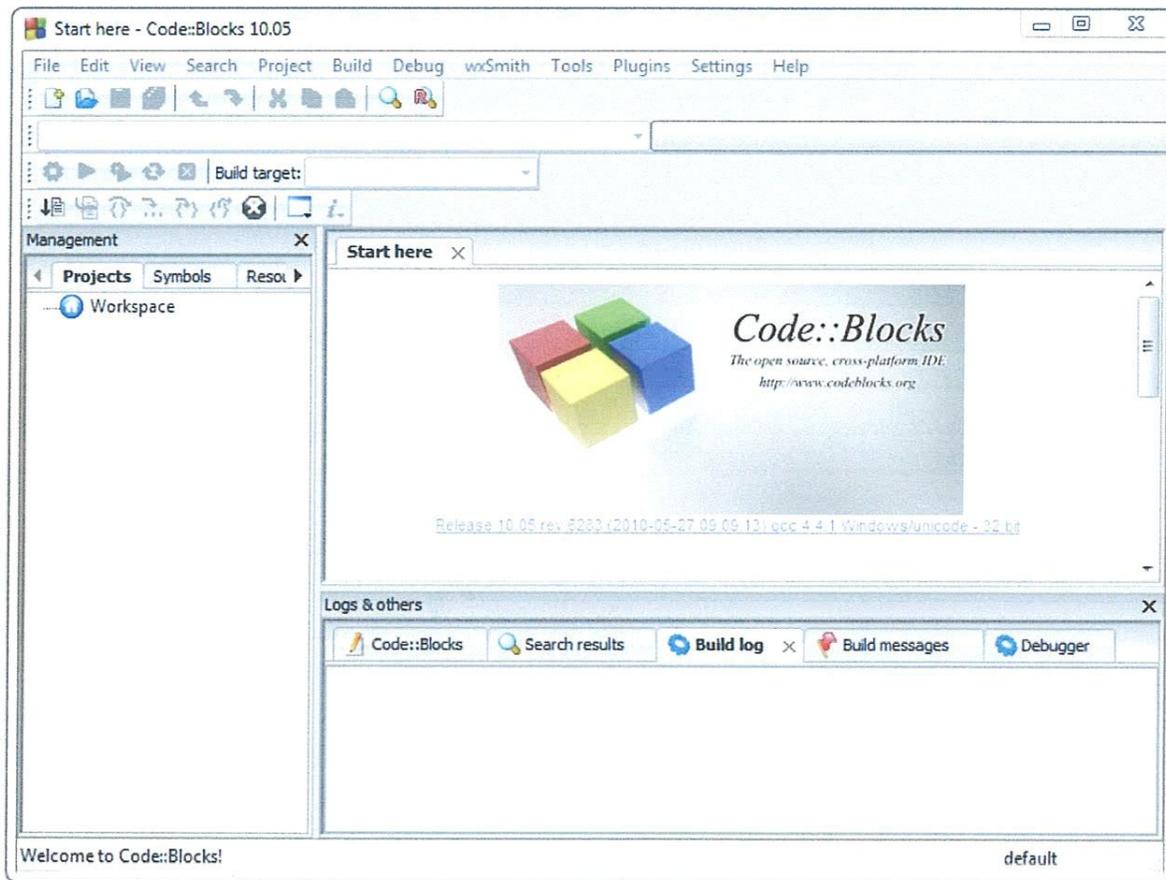


Figure 3.1: Environnement de développement intégré (IDE) de Code-Blocks.

1.2. Qt creator :

Qt Software (anciennement Trolltech), une branche de Nokia, a commencé récemment le développement d'un EDI consacré exclusivement au développement avec Qt. Ce projet, nommé Qt Creator. Alors Qt Creator est un environnement de développement intégré multi plate-forme, il est orienté pour la programmation en C++.Qt Creator utilise sous Linux le compilateur GCC et MinGW par défaut sous Windows.

Qt Designer, un outil de création d'interfaces graphiques, a été totalement intégré à Qt Creator, il est tout à fait envisageable de passer de Qt Creator à une gestion manuelle et vice-versa durant le développement d'un projet [13].

L'illustration ci-dessous montre l'apparence de la fenêtre de l'interface utilisateur de Code-Blocks.

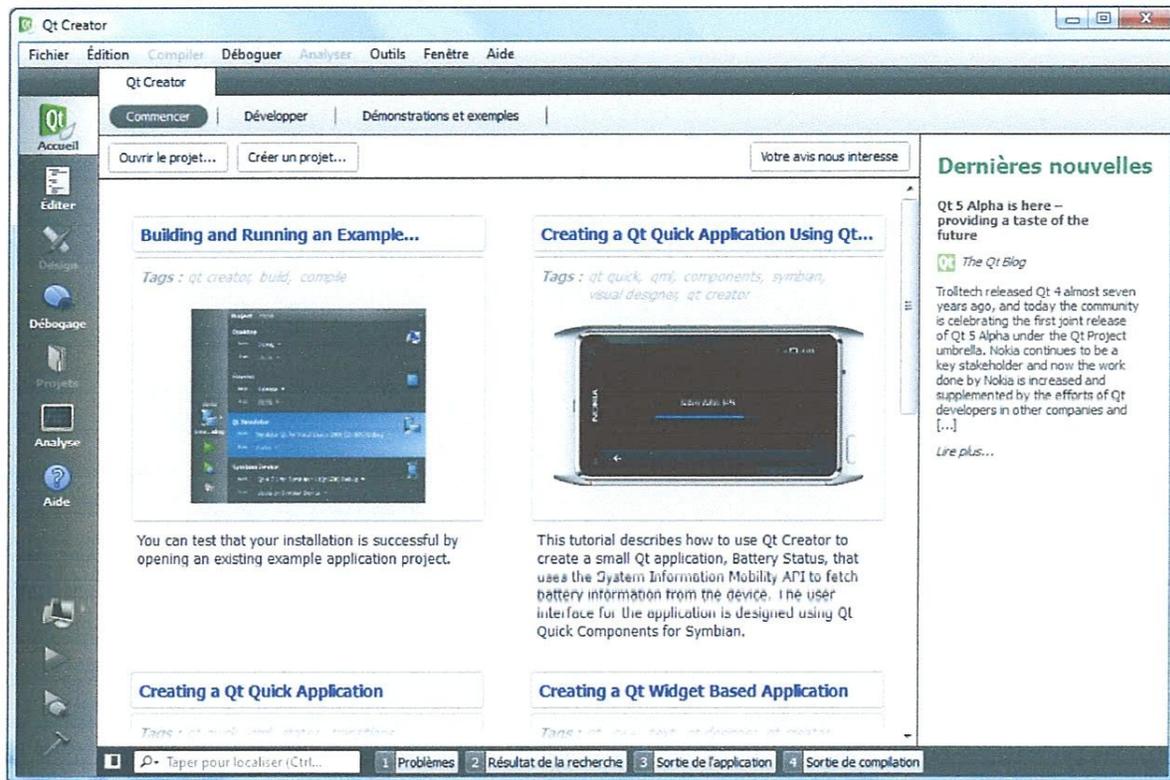


Figure 3.2: Environnement de développement intégré de Qt Creator.

1.3. OpenCV 2.0 :

OpenCV (pour Open Computer Vision) est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. La société de robotique Willow Garage assure le support de cette bibliothèque en 2010. La bibliothèque OpenCV met à disposition de nombreuses fonctionnalités très diversifiées permettant de créer des programmes partant des données brutes pour aller jusqu'à la création d'interfaces graphiques [14].

1.3.1. Traitement des images et des vidéos avec OpenCV :

OpenCV est une bibliothèque de traitement d'images et de vision par ordinateur en langage C/C++, optimisée et proposée par Intel pour Windows et Linux [15].

Elle comprend un très grand nombre d'opérateurs, Parmi lesquels :

- Création / libération d'images, macros d'accès rapide aux pixels.
- Opérateurs standards : morphologie, filtres dérivatifs, filtres de contours, suppression de fond, recherche de coins...
- Recherche, manipulation, traitement de contours.
- Dessins de primitives géométriques (lignes, rectangles, ellipses, polygones . . . et même du texte).
- Tableaux, matrices et matrices creuses +opérateurs associés (opérateurs classiques de matrices : inversion, déterminant, transposée, valeurs/vecteurs propres, ...)
- Changements d'espaces de couleurs (RGB, HSV, YCrCb...).
- Lecture d'images à la volée directement depuis une vidéo AVI ou une caméra.
- Détection de mouvement, historique du mouvement.

Voici quelques exemples sur les traitements avec openCV [15]:

➤ Déclaration et création d'une image :

```
#include<cv.h>
IplImage *im;
im = cvCreateImage(cvSize(512,256), IPL_DEPTH_8U, 3);
/* Image 512 x 256 de unsigned char avec 3 canaux */
```

➤ Remplissage : (codage par défaut = BGR) :

```
pteur = (unsigned char*)im->imageData;
fin = im->width*im->height;
for(i=0; i<fin; i++)
{
*(pteur++) = 255; /* Bleu */
*(pteur++) = 0; /* Vert */
*(pteur++) = 0; /* Rouge */
}
```

➤ Libération de l'espace mémoire

```
cvReleaseImage(&im);
```

➤ Lire et afficher une image (highgui.h) :

- Lire une image :

```
#include<highgui.h>
IplImage *im;
im = cvLoadImage("mon_image.jpg", 1);
```

- Afficher une image

```
cvNamedWindow("Non dufenetre", CV_WINDOW_AUTOSIZE);
cvShowImage("Non dufenetre", im);
```

➤ Détection de contours :

On trouve les filtres de Canny, Sobel, Laplacien, détection de coins, ...

- Exemple du Canny :

```
im = cvLoadImage(argv[1], 0);
im2 = cvCreateImage(cvGetSize(im), IPL_DEPTH_16S, 1);
cvCanny(im, im2);
```

➤ Lecture d'une vidéo AVI :

Il est possible de lire une séquence d'images à partir d'un fichier vidéo. Pour chaque étape, une image de type **IplImage** est automatiquement allouée (puis libérée) en mémoire.

```
/* Variables */
IplImage *im;
CvCapture *avi;
/* Ouverture de la video */
avi = cvCaptureFromAVI("Non_video.AVI");
while(cvGrabFrame(avi))
{
im = cvRetrieveFrame(avi);
/* Traitement de l'image */
}
```

- Les Bottons (**MLT** “ méthode basée sur les larges textures ”, **MLR_A** “méthode basée sur les larges textures améliorée”, **N_Approche** “Notre propre approche”) : pour lancer chaque méthodes a part.

3. Comparaison de la performance des méthodes :

Dans cette section, nous présentons une série des tests pour réaliser une comparaison entre les trois méthodes (la méthode basée sur les textures larges, la méthode basée sur les textures larges améliorée et notre propre approche qu'on a proposé), nous tenons compte de trois critères pour faire cette comparaison. Le premier critère est la performance de détection d'ombre, nous expérimentons la méthode par rapport à la détection de l'ombre, totalement ou partiellement. Le deuxième critère est la discrimination entre les pixels d'ombre et les pixels d'objets, nous testons la méthode si elle fait la distinction entre les zones des ombres et celles des objets. Le troisième critère est le temps d'exécution, nous testons si la méthode est lourde ou pas.

Pour réaliser ces tests, nous avons pris cinq vidéos dans des environnements différents (ouverte, fermée...). Par la suite nous testons chaque une des trois méthodes sur les cinq vidéos.

Soit :

- Méthode 1 : la méthode basée sur les textures larges proposée par *Andres Sanin et al.*
- Méthode 2 : la méthode basée sur les textures larges et la chromaticité qu'on a améliorée.
- Méthode 3 : notre propre approche.

Les figures suivantes montrent des résultats obtenus après l'application de chaque méthode sur les différentes vidéos (nous prenons 2 frames à partir chaque vidéo) :

Conclusion générale :

Depuis la fin des années 90, la numérisation des contenus et la progression de la puissance des ordinateurs ont rendu possible le traitement des vidéo surveillances en temps réel pour en extraire des interprétations (que voit-on à l'image, que se passe-t-il, qui va où, etc). Ces traitements ont commencé à sortir des laboratoires de recherche dans cette période, et à constituer des solutions exploitables, pour la surveillance des personnes, d'objets et des lieux, etc.

L'apparition des ombres est l'un des problèmes majeurs rencontrés lors de ces traitements appliqués sur les vidéos surveillances. De ce fait, nous avons présenté dans ce travail des solutions pour détecter et éliminer les ombres. D'abord, nous avons amélioré une méthode basée sur les textures larges proposées par Andres Sanin et al, en se basant sur les trois espaces de couleur RGB, HSV et YUV dans l'étape de détection des pixels candidats d'ombre. Nous avons aussi proposé une méthode propre à nous, dans la qu'elle nous utilisons le filtre de *Canny* pour détecter les contours existants dans chaque frame (image), puis nous créons des régions à partir des objets du premier plan (détecter en mouvement) et les contours détectés avec le filtre de *Canny*, puis nous classons chaque région comme ombre ou pas.

En suite nous avons réalisé une étude comparative entre les méthodes implémentées en se basant sur trois critères (la détection d'ombre, la discrimination entre les pixels d'ombre et les pixels d'objets, et le temps d'exécution). Dans cette étude, nous avons montré que notre approche est la plus forte et la plus performante en fonction des trois critères : une détection entière d'ombre, une bonne distinction entre les zones d'ombre et les zones d'objets, et un temps d'exécution plus court par rapport à les autres méthodes.

Référence :

- [1] : <http://www.lamap.fr/> (la vitesse de la lumière), 11-04-2012.
- [2] : Antoine MANZANERA, Enseignant-Chercheur, « Cours Traitement et reconnaissance d'images – Master 2 », Université Pierre et Marie CURIE, Paris, 2001.
- [3] : Jean Serra, « Espaces couleur adaptés au traitement d'image », centre de morphologie mathématique Ecole des Mines de Paris, mars 2003.
- [4] : Dornberger, Walter: (V-2) Ballantine Books, ASIN: B000P6L1ES, page 14.
- [5] : Journal officiel de la république française, « Arrêté du 3 août 2007 portant définition des normes techniques des systèmes de vidéosurveillance ». PDF : [http : //www.legifrance.gouv.fr/jopdf/jopdf/2007/0821/joe_20070821_0192_0004.pdf](http://www.legifrance.gouv.fr/jopdf/jopdf/2007/0821/joe_20070821_0192_0004.pdf), 21/05/2012).
- [6] : Nadia Baaziz, « La vidéo surveillance automatique: sécurisation du contenu et traitements coopératifs », Université du Québec en Outaouais, Rapport de recherche 07/06-1, Juin 2007).
- [7] : Henri MAITRE, ^{Titre} École Nationale Supérieure des Télécommunications, Laboratoire Image, Département Images, Sons et Vidéo, 46, rue Barrault, 75634 PARI.
- [8] : Ariel Amato, Mikhail G. Mozerov, Andrew D. Bagdanov, and Jordi González, « Accurate Moving Cast Shadow Suppression Based on Local Color Constancy Detection », IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 20, NO. 10, OCTOBER 2011.
- [9] : Andres Sanin, Conrad Sanderson, Brian C. Lovell, « Shadow Detection: A Survey and Comparative Evaluation of Recent Methods », The University of Queensland, School of ITEE, QLD 4072, Australia, 10-2011.
- [10] : A. Prati, I. Mikic, M. Trivedi, and R. Cucchiara. « Detecting moving shadows : algorithms and evaluation », IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(7):918–923, 2003.
- [11] : Canny, J. A, « Computational Approach To Edge Detection », IEEE Transactions on Pattern Analysis and Machine Intelligence, 8:679-714 (1986).
- [12] : Site officiel de codeBlocks : <http://www.codeblocks.org>, 03/05/2012.
- [13] : Site officiel de Qt Creator : <http://qt.nokia.com/products/developer-tools>, 03/05/2012.

[14] : http://www.amazon.com/Learning-OpenCV-Computer-VisionLibrary/dp/0596516134#reader_0596516134, 22/05/2012.

[15] : http://www.irit.fr/~Alain.Crouzil/jaffre/LOGICIELS/OPENCV/pres_opencv_irit.pdf.