

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud MAMMERI, Tizi-Ouzou



Faculté de Génie Electrique et d'Informatique
Département d'Automatique

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme

D'Ingénieur d'Etat en Automatique

Thème

***Tracking vidéo en temps réel par les
contours actifs***

Proposé et dirigé par :

M^{lle} Ouardia CHILALI.

Soutenu le : 07/07/2010

Présenté par :

M^r Samir HACHOUR

M^r Idir FEDDAG

M^r Youcef IABADENE

Promotion 2010



Dédicaces et remerciements



Remerciements

*Nous adressons nos sincères remerciements
premièrement à notre chère promotrice M^{lle} CHILALI
Ouardia pour avoir accepté de nous encadrer afin de
réaliser ce travail ainsi que pour l'aide précieuse qu'elle
nous a prodigué tout au long de son déroulement. Nous
tenons à lui dire que nous avons apprécié au mieux ses
considérations et son respect à notre égard et que nous
la respectons tout autant.*

*Nos remerciements à l'équipe constituant le jury
pour avoir accepté de juger notre travail.*

*Nos vifs remerciements à nos chères familles pour
leurs soutiens moraux et financiers.*

*Nous tenons à remercier tous nos amis qui nous ont
apporté leurs aides, de près ou de loin, en termes de
conseils et d'orientation pour mener à bien ce travail.*

Dédicaces

Je dédie ce modeste travail à mes chères parents qui, quotidiennement, m'ont soutenu vivement et m'ont orienté souvent pour trouver toujours le droit chemin.

Mes dédicaces à mon frère Salah, à mes sœurs Yasmine, Kahina, Wardia et particulièrement à ma petite sœur Liza.

Je dédie ce travail à toutes mes cousines et cousins et surtout à mon cousin Hamid ainsi qu'à tous mes amis.

F. Idir

Dédicaces

Je dédie ce modeste travail à :

- **Mes Chers parents.**
- **Mes frères et sœurs : Hassen, Amar, Samia, Nacera, Kahina et Dalila.**
- **Mes beaux frères : Amar, Brahim et Hassen.**
- **Ma belle sœur Salima.**
- **Mes neveux et nièces : Celia, Lisa, Lounes, Axil, Thamina et Anaïs.**
- **Toute ma famille.**
- **Mes amis (es).**

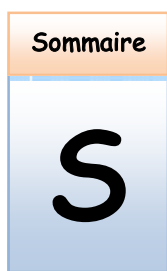
Youcef.

Dédicaces

Je dédie ce modeste travail à tous ceux qui me sont chers :

- Ma famille au grand complet.*
- Mes amis.*
- Et l'ASAM judo.*

H. Samir.



Sommaire



« Ce que le tableau représente ?
Cela dépend de celui qui le regarde »

James WHISTLER.

Sommaire

Introduction générale

Généralités sur le tracking vidéo

I.1 Introduction	3
I.2 Définition	3
I.3 Représentation d'un Objet	4
I.3.1 Représentation de la forme	4
I.3.2 Représentations de l'apparence	5
I.3.3 Paramètres de sélection d'un objet	6
I.4 Détection d'un objet en mouvement	9
I.5 Suivi d'objets	11
I.5.1 Suivi par point	11
I.5.2 Suivi par forme géométrique de base	12
I.5.3 Suivi de la silhouette	14
I.6 Contours actifs	15
I.7 Conclusion	16

II. Les contours actifs pour la segmentation

II.1 Introduction	18
II.2 Définition des contours actifs	18
II.3 Energies d'un contour actif	19
II.3.1 Energie interne	19
II.3.2 Energie externe	19
II.3.3 Energie de contexte	20
II.4 Modélisation d'un contour actif selon sa représentation	20
II.4.1 Approche explicite	21
II.4.2 Approche implicite	21
II.5 Approches de contours actifs selon l'énergie externe	22
II.5.1 Contours actifs basés contours	22
II.5.2 Les contours actifs basés région	24
II.6 Conclusion	28

III Approche adoptée

III.1 Introduction	30
III.2 Présentation du modèle de contour actif adopté	30

Sommaire

III.3 Implémentation	33
III.3.1 Application à une image statique	36
III.3.2 Application au suivi d'objet	38
III.4 Conclusion	41

IV. Tests et résultats

IV.1 Introduction	42
IV.2 Ressources Matérielles et Logicielle	42
IV.3 Interface dédiée	43
IV.4 Tests et résultats	44
IV.4.1 Image statique	44
IV.4.1.1 Segmentation locale	44
IV.4.1.2 Segmentation Globale	46
IV.4.2 Application au vidéo tracking	48
IV.4.2.1 Etude de l'image à segmenter	48
IV.4.2.2 Tracking Vidéo	51
IV.5 Conclusion	57

Conclusion générale.....	59
--------------------------	----

Annexes

Bibliographie

Résumé :

Le suivi d'objet en mouvement est une problématique d'actualité dans le domaine de la vision artificielle.

La méthode que nous avons présentée tire la propriété de segmentation locale des contours actifs géodésique qui sont basés contours et la propriété de segmentation globale du modèle de Chan et Vese qui est basé région et la possibilité de gérer automatiquement le changement de topologie par son implémentation implicite.

Le temps de calcul est réduit grâce à la nouvelle considération de la fonction Level-Set et aux simplifications des termes lourds pour le calcul.

Des résultats satisfaisants sont obtenus après application de notre algorithme pour un matériel pas assez performant.

Intro

I

Introduction générale



« L'invention scientifique réside dans la création d'une hypothèse heureuse et féconde ; elle est donnée par le génie même du savant qui l'a créée. »

Claude Bernard.

L'Homme, dans sa culture et vision humaine, parvient à prouver ses hautes capacités dans la compréhension et l'interprétation de son environnement. Possédant une véritable représentation globale du monde extérieur, il arrive à exprimer les attitudes qu'il a devant certaines situations. Voulant imiter ce comportement et le modéliser par des outils mathématiques afin de le simuler sur des machines ayant une capacité de traitement, relativement, de même que celle des systèmes biologiques, de nombreux chercheurs ont orienté leurs études vers l'amélioration des techniques de traitements de l'information. La vision par ordinateur est apparue comme fruit de ces recherches. Elle englobe, entre autres, la reconnaissance des formes et l'intelligence artificielle.

Dans le contexte de la vision artificielle, le traitement d'images se place après les étapes d'acquisition et de numérisation, assurant les transformations d'images et permettant d'aller vers une interprétation des images traitées. Le besoin d'automatiser le traitement d'images s'est fait sentir dans divers domaines tels que l'analyse de scènes dynamiques, qui nécessite un très grand nombre d'informations à traiter. En effet, le principal objectif d'une telle analyse est de détecter les informations qui ne peuvent être extraites à partir d'une image unique. Il revient à étudier les transformations subies entre les images successives de la séquence, afin de localiser les objets en mouvement.

Pour mesurer l'ampleur de la tâche, il faut se rendre compte de toutes les connaissances que l'être humain met en jeu pour séparer des objets d'un fond. La machine ne peut pas, comme lui, chercher dans sa mémoire l'objet qui, par sa forme, sa couleur, son intensité lumineuse, ses caractéristiques (définies par l'expérience et l'apprentissage), se rapproche le plus de l'image, éventuellement après des transformations multiples (rotation, étirement, etc.), pour ensuite séparer l'objet du fond.

Parmi les nombreuses méthodes efficaces déjà existantes, les approches par contours actifs apparaissent comme une technique prometteuse dans le domaine de suivi d'objets en mouvement. La question qui se pose, naturellement, est : « Quel type de contours actifs choisir et adopter pour une application au suivi d'objet en temps réel ? ». En réalité, la réponse la plus simple est de choisir les contours actifs qui se basent sur l'implémentation explicite, mais au détriment de beaucoup d'inconvénients tels le problème connu d'initialisation et

le non autorisation du changement de topologie. De ce fait, les contours actifs implicites (implémentation par ensembles de niveaux) sont plus avantageux. Cependant, si l'implémentation par ensembles de niveaux offrait des résultats probants, elle laisse entrevoir quelques faiblesses particulières pour autoriser un suivi en temps réel.

Notre travail consiste, donc, à développer un modèle de contour actif implicite pour une application au *suivi d'objet en temps réel*. Pour ce faire, nous avons organisé notre travail comme suit :

Dans le premier chapitre, nous avons décrit les outils de base en traitement d'images en général, et en suivi d'objets en particulier. Nous avons abordé les différentes modélisations d'objets en mouvement dans une vidéo. Par la suite, nous avons abordé quelques techniques de détection d'objets en mouvement qui constitue une étape, à la fois, initial et primordiale du *Tracking vidéo*. Nous avons consacré la dernière section pour discuter les principales techniques de suivi d'objet en mouvement, afin de justifier notre choix pour les contours actifs.

Le deuxième chapitre s'est voulu être un rappel sur les différentes approches des contours actifs, qui est une suite logique du chapitre introductif. Ce chapitre s'est conclu par le choix du modèle de suivi en temps réel.

Le troisième chapitre a été déployé pour détailler, en sa première partie, le modèle adopté qui est, théoriquement, assez bon pour une application en temps réel. Dans la deuxième partie de ce chapitre, nous avons décrit l'implémentation du modèle en passant par son application sur une image statique, pour en arriver à son application pour le suivi d'objet en temps réel.

Dans le quatrième chapitre, nous avons étalé les résultats obtenus pour différents tests tout en veillant à ce que chaque résultat soit scruté et conclu.

Nous terminerons ce mémoire en présentant une conclusion générale.

Chapitre

1

Généralité sur le tracking vidéo



*« Le commencement de toutes les sciences, c'est
l'étonnement de ce que les choses sont ce qu'elles sont »*

Aristote.

I.1 Introduction

Le suivi d'objet est une branche très importante dans le domaine de la vision artificielle. Les performances des ordinateurs récents, la disponibilité des caméras de haute qualité et le besoin crucial de traitement automatique de la vidéo, ont donné naissance à un grand challenge, et plusieurs recherches s'y sont intéressées.

Cependant, le suivi d'objets en mouvement est une tâche très difficile à mettre en œuvre. Les principaux problèmes rencontrés se traduisent par la spontanéité du mouvement des objets en question, le changement d'apparence des objets ou de l'arrière plan, une structure non rigide des objets, la mobilité de la caméra, etc.

Le présent chapitre fera l'objet d'une étude représentative et comparative des méthodes, les plus utilisées, de suivi d'objet et les principaux problèmes qui vont avec, comme la représentation des objets dans une image, et la détection de leurs mouvements.

I.2 Définition [11]

Littérairement, « *to track* » de l'anglais, veut dire suivre ou pister. Lorsqu'on parle de tracking vidéo, on sous-entend le suivi d'objet en mouvement.

Le suivi d'objet est le processus de localisation d'un (ou plusieurs) objet en mouvement dans le temps en utilisant, en général, une caméra. L'algorithme fait l'analyse de chaque image de la vidéo. Il nous donne, en sortie, les informations sur les objets ciblés et leurs mouvements.

Un algorithme de suivi d'objet doit en tenir compte des deux points suivants :

- La représentation et la localisation des objets.
- L'établissement de la correspondance entre les différentes images d'une vidéo.

Cela peut se faire en trois principales étapes, qui sont les suivantes :

- Détection des objets en mouvement.
- Suivi de chaque objet dans chaque trame (pour rappel une vidéo est une succession de trames ou d'images)

- Analyse du mouvement pour reconnaître le comportement de l'objet, voire le prédire.

Les domaines d'application du suivi d'objets sont très nombreux. On citera comme exemple :

- Domaine médical : suivi de l'évolution d'une tumeur.
- Images satellitaires : suivi de l'évolution des ouragans.
- Industrie : supervision de la chaîne de fabrication et la télésurveillance.
- Robotique : la vision artificielle serait une source d'informations quasi-parfaite pour un robot.
- Gestion des trafics aérien, maritime et routier.

I.3 Représentation d'un Objet [20]

Dans le vidéo tracking, un objet peut être de nature quelconque. Tous les scénarios sont possibles à imaginer : un bateau en pleine mer, un joueur dans un stade avec son ballon, un véhicule en autoroute, un avion dans le ciel, une pièce à usiner dans une chaîne de fabrication, des piétons au trottoir, etc. Selon la nature de l'objet et l'objectif de l'application à réaliser, de différentes représentations de la forme et d'apparence sont utilisées.

I.3.1 Représentation de la forme

Le but est de trouver une forme qui peut représenter l'objet à segmenter, en tenant compte de notre application et des informations qu'on veut avoir. Par exemple : un point suffirait pour représenter un avion dans l'air, par contre un contour est nécessaire pour la représentation d'un corps humain en mouvement.

- *Points* : L'objet est représenté par un point, qui est le centre de gravité de la surface de l'objet [1] (figure I.1. (a)), ou par un nuage de points (figure I.1. (b)). Cette représentation est utilisée, généralement, lorsqu'on a affaire avec un objet qui occupe une taille réduite dans l'image dont ses caractéristiques physiques sont peu déterministes.
- *Formes géométriques de base* : Un rectangle ou une ellipse (figures I.1.(c) et I.1.(d)), peuvent être affectés à la forme de l'objet. Cette représentation est souvent utilisée lorsqu'il s'agit d'un mouvement de translation et lorsqu'on a affaire à un objet rigide.

- **Modèle articulé** : Un objet comportant plusieurs membres interconnectés, est un objet articulé. On représente par une forme géométrique de base (ellipse ou cylindre) chacun de ces membres. Ainsi, on obtient un modèle de représentation articulé. Le meilleur exemple pour le modèle articulé est le corps humain (figure I.1. (e)).
- **Ossature** : Obtenu par représentation de l'axe médian de la silhouette de l'objet à suivre (figure I.1. (f)). Ce modèle peut être appliqué soit pour un objet rigide ou non rigide.
- **Silhouette et contour** : Mise en évidence des contours de l'objet (figures I.1. (g) et I.1.(h)) qui permettra de retrouver la silhouette de l'objet (figure I.1.(i)). C'est une représentation très appréciée dans le cas du tracking d'objets complexes et non rigides.

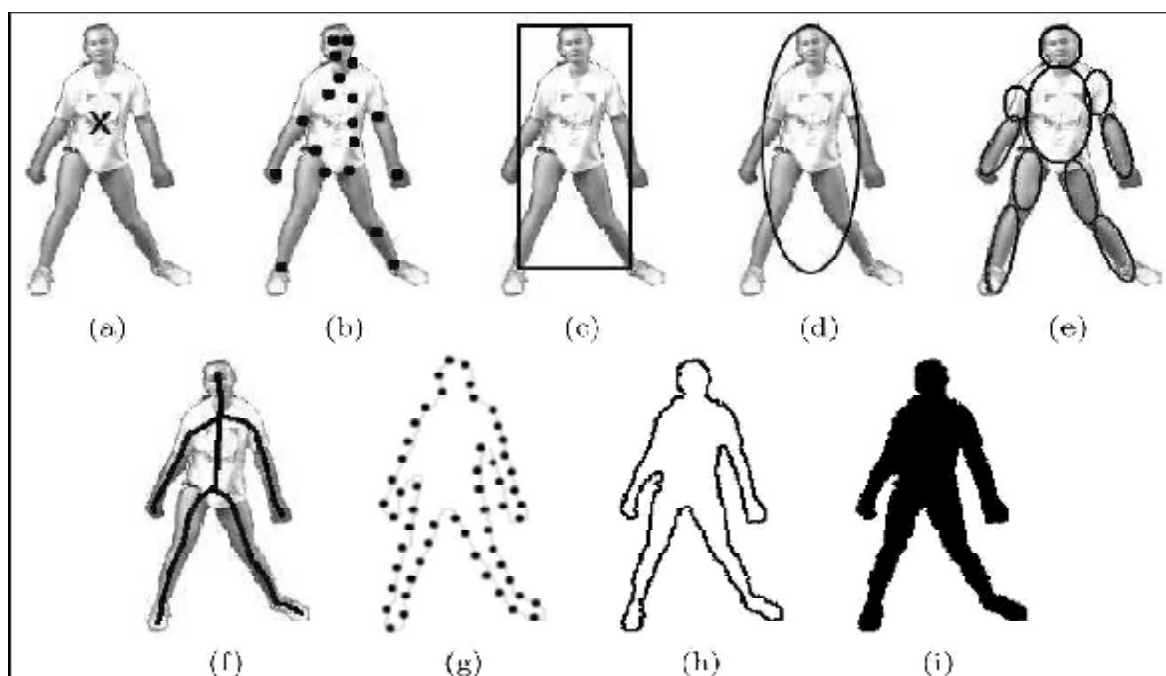


Figure.I.1 : Représentations d'objets : (a) centre de gravité (point), (b) ensemble de points, (c) rectangle, (d) ellipse, (e) modèle articulé, (f) ossature, (g) contour continu, (h) contour discret et (i) silhouette de l'objet.

I.3.2 Représentations de l'apparence

Souvent, l'apparence d'un objet en mouvement joue un rôle primordial, lorsqu'il s'agit de le distinguer : des autres objets qui sont comme lui en mouvement d'un côté et de son arrière plan de l'autre. Par exemple, la couleur de l'objet est un détail très exploitable lorsqu'on a à utiliser l'histogramme de l'image. Par contre, la lisière de l'objet est plus intéressante lorsqu'on se base sur

la méthode des contours. Il est à noter que l'apparence et la forme peuvent être combinées pour représenter l'objet, dans le contexte du suivi d'objets.

Il y a plusieurs manières possibles pour la représentation de l'apparence d'un objet.

- **La densité de probabilité d'un objet** : La distribution de la densité de probabilité d'un objet peut être soit paramétrique, où les éléments de l'image (pixels) sont représentés par leur coordonnées relativement à toute l'image, ou bien non paramétrique comme les représentations par histogrammes, qui sont souvent utilisés.
- **Modèle correspondant (Template)** : Il est formé en utilisant une forme géométrique de base. Le modèle correspondant a l'avantage de prendre en considération l'apparence et les informations spatiales de la région de l'image où il est appliqué. C'est une représentation recommandée pour les mouvements avec peu de déplacements.
- **Modèle d'apparence active (AAM)** : Il est généré en modélisant la forme et l'apparence de l'objet au même temps. La forme est représentée par un ensemble de points de repère, et pour chaque point, un vecteur de données statistiques (couleur, texture, etc.) est enregistré. Cette méthode est largement utilisée dans le suivi des visages et l'interprétation des images médicales.
- **Modèle d'apparence à plusieurs vues** : Ce modèle encode les différentes vues de l'objet, puis des méthodes de soustraction de fond sont utilisées pour définir l'objet en mouvement. Parmi ces méthodes on peut citer : méthodes d'analyse en composants principales (PCA) et la méthode d'analyse des composants indépendants (ICA). Elles sont toutes les deux utilisées pour la représentation de la forme et de l'apparence de l'objet.

I.3.3 Paramètres de sélection d'un objet

La sélection du bon paramètre est cruciale dans le suivi d'objets. En général, on désire avoir les propriétés visuelles d'un objet uniques, afin de le distinguer facilement de l'ensemble des autres objets. Les paramètres de sélection sont fortement rattachés aux représentations des objets. Par exemple : la couleur est un paramètre de sélection utilisé pour la représentation des apparences basées sur les histogrammes, les bordures de l'objet sont des

paramètres de sélection pour les représentations d'apparences basées sur les contours, mais aussi la combinaison de plus d'un paramètre de sélection. Quelques paramètres seront présentés dans ce qui suit.

- **Couleur** : L'apparence couleur d'un objet est influencée par deux paramètres physiques :
 - La distribution spectrale de la luminance.
 - Les propriétés de réflexion de la surface de l'objet.

Dans le traitement d'images, le RGB (Red, Green, Blue) est souvent utilisé comme espace de représentation de la couleur, mais il a le désavantage de ne pas être une distribution uniforme de la couleur. Les espaces $L^*u^*v^*$ et $L^*a^*b^*$ et HSV (Hue, Saturation, Value) sont des espaces de distribution uniforme de la couleur, mais ils ont l'inconvénient d'être sensibles aux bruits. Un exemple de représentation d'une image au niveau de gris et couleur est représenté dans la figure I.2.

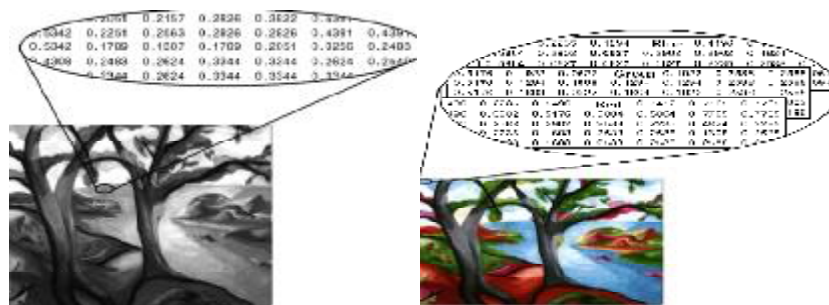
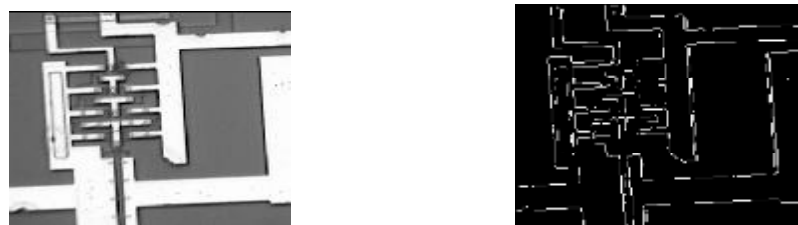


Image au niveau de gris.

Image dans l'espace RGB.

Figure I.2 : Représentation des couleurs.

- **Bordures** : Souvent, les grands changements d'intensités d'une image se trouvent sur les bordures (figure I.3) des objets, qui ont l'avantage d'être moins sensibles aux changements de luminance. Les algorithmes qui se basent sur la détection et le suivi des contours, utilisent souvent les bordures des objets comme paramètre de sélection.



(a) Image originale.

(b) Bordures de l'image.

Figure I.3 : Représentation des bords.

- **Flot optique** : C'est un champ de vecteurs qui représentent le déplacement de chaque pixel d'un objet dans une région (figure I.4). Il suppose l'intensité des pixels invariante. Le flot optique est, largement, utilisé dans la segmentation pour les applications du suivi d'objets. Pour représenter l'apparence, il donne des informations spatiales de l'objet d'intérêt.

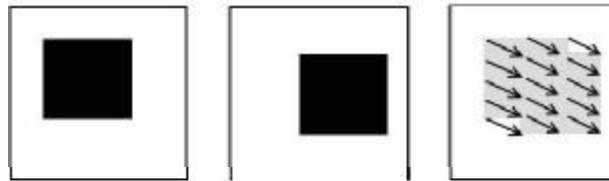
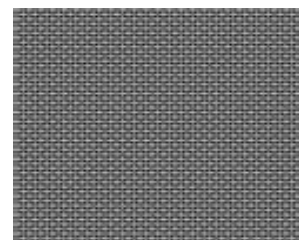


Figure I.4 : Exemple de flot optique.

- **Texture** : Une texture est utilisée pour la représentation d'une surface d'un objet caractérisé par une certaine homogénéité. La texture est une mesure qualitative qui peut décrire l'information caractérisant l'objet. Cette information peut avoir les aspects suivants : grossière, lisse, tachetée, granuleuse, fine, etc. La mise en œuvre de la texture d'une région nécessite une étape supplémentaire pour le calcul des descripteurs (texels), qui sont les plus petits éléments représentatifs d'une texture. Les images de la figure I.5 représentent deux textures différentes.



(a) Texture quelconque.



(b) Texture uniforme.

Figure I.5 : Exemples de textures.

I.4 Détection d'un objet en mouvement

Tous les algorithmes de suivi d'objets nécessitent un mécanisme de détection de mouvement, que ce soit sur toutes les images de la scène ou bien uniquement lorsque l'objet apparaît. Pour une caméra fixe, on utilise souvent les méthodes basées sur la soustraction de fond, ce qui n'est pas adapté à un mouvement vu par une caméra mobile. Dans ce cas, on utilise des méthodes basées sur la détection de points ou de contours par exemple.

En l'absence de mouvement de caméra, les objets en mouvement représentent la source principale de différence entre les images. Partant de ce constat, la segmentation courante d'une image, consiste à définir (figure I.6) :

- le fond de l'image comme la partie statique de l'image.
- le ou les objets comme la partie mobile de l'image.

Cette segmentation est donc une détection de mouvement.

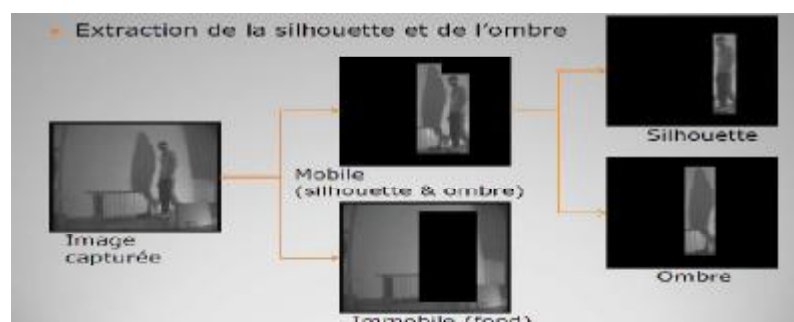


Figure I.6 : Décomposition d'une image en une partie mobile, et partie immobile (fond).

Le travail de *Wren et al.* (1996) [2], se porte sur la soustraction de fonds. Pour reconnaître les changements au cours de temps, ils ont proposé de modéliser la couleur de chaque pixel. Ensuite, des paramètres, tels que la moyenne et la covariance, feront l'objet d'informations sur la couleur de plusieurs images successives de la séquence. En se basant sur ces informations, le fond de la séquence est estimé et modélisé. Ainsi, pour détecter un mouvement, il suffirait de faire une soustraction directe entre les différentes images de la séquence et ce fond estimé.

D'autres approches utilisent, en plus des informations sur la couleur des pixels, des informations basées sur les régions de l'image. Par exemple, *Elgammal et al.* (2000) [3] ont utilisé une estimation non-paramétrique pour modéliser le

fond de la séquence. Durant le processus de soustraction, le pixel courant est comparé, non seulement avec le pixel qui lui correspond dans le fond, mais aussi avec les pixels qui lui sont voisins. Ce qui a donné à cette méthode la capacité de compenser les petits mouvements de la caméra.

En 2002, *Li et Lueng* [5] ont utilisé des blocks de pixels 5X5, ayant des informations sur la couleur et la texture, pour modéliser le fond de la séquence. L'utilisation des informations sur la texture a rendu cette méthode beaucoup moins sensible aux changements de luminance.

Rittscher et al. (2000) [4], ont utilisé le modèle caché de Markov(HMM), pour représenter la variation de l'intensité d'un pixel comme un état discret par rapport aux événements de son environnement.

Une autre méthode, basée sur la modélisation de la variation individuelle des pixels, est développée par *Oliver et al.* (2000) [6]. C'est une méthode globale, qui consiste en la décomposition de l'espace propre de l'image (figure I.7).

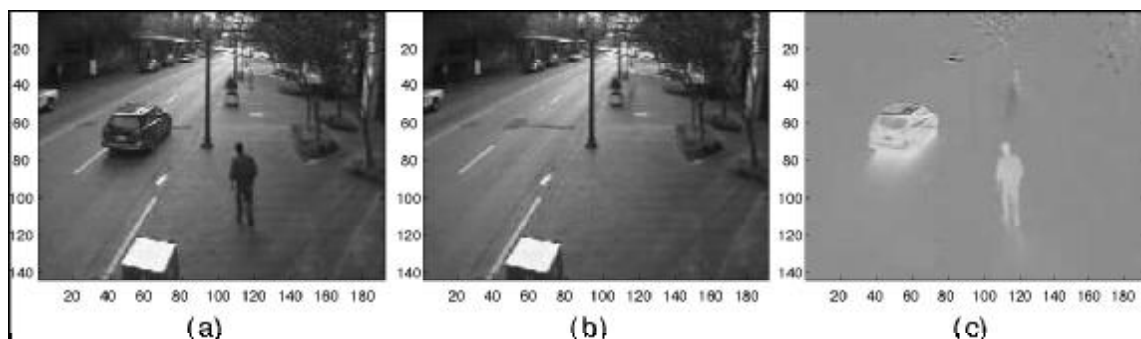


Figure I.7 : Résultats de soustraction entre le fond estimé et l'image contenant les objets.

Pour les séquences à caméra mobile, on ne peut plus définir l'objet comme l'ensemble des pixels en mouvement, car le mouvement de la caméra induit un mouvement de tous les pixels de l'image. Cependant, les pixels de l'objet ont un mouvement local qui s'ajoute au mouvement global de l'image. Donc, l'objet sera considéré comme l'ensemble des pixels en mouvement, mais qui ne suivent pas le mouvement global de l'image. La solution serait d'estimer le mouvement de la caméra ou bien d'estimer un fond qui est variant. Pour cela, des méthodes de construction de la mosaïque, par exemple, qui est le fond de la séquence pour une

caméra mobile, sont développées en se basant sur la reconstitution des fonds de toutes les images de la séquence.

I.5 Suivi d'objets

Le suivi d'objets en mouvement est une des techniques les plus avancées dans le domaine du traitement d'images. Le principe est de déterminer la trajectoire d'un objet en mouvement dans le temps, par la localisation de sa position dans chaque image de la séquence.

Notons que l'étape de la détection de mouvement, nécessaire pour les algorithmes de suivi d'objet, et l'établissement de la correspondance entre les positions de l'objet en mouvement à travers les images de la séquence (Tracking), peuvent se faire soit séparément, où séquentiellement. Dans les deux cas, l'objet à suivre est représenté par sa forme ou/et son apparence. Souvent, les algorithmes de suivi d'objets sont contraints par le type de la représentation choisie pour l'objet. Par exemple, si l'objet est représenté par un point, la translation et le seul mouvement à suivre par l'algorithme. Mais pour les objets non-rigides, qu'on représente par une silhouette ou un contour, un algorithme assez évolué est nécessaire pour extraire le maximum d'informations sur l'objet et son mouvement.

Dans cette section, on présentera les types d'algorithmes de suivi d'objet existants et on exposera, brièvement, quelques modèles de chaque type.

I.5.1 Suivi par point

Les objets détectés dans des images consécutives sont représentés par des points (figure I.8). La correspondance entre les points est faite par la connaissance de leurs états dans les images précédentes de la scène (que ce soit sa position ou son mouvement). Ces approches nécessitent un mécanisme de détection dans chaque image de la séquence.

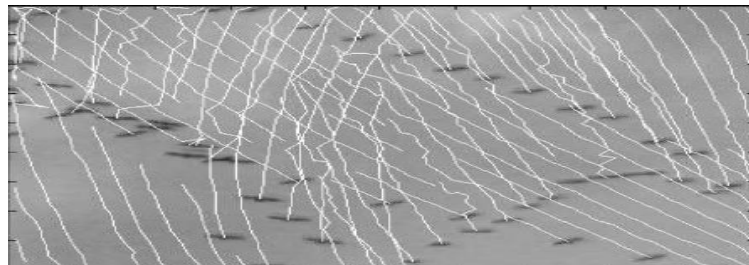


Figure I.8 : Suivi des oiseaux par des points.

Les méthodes basées sur le suivi de point peuvent être classées en deux catégories qui sont :

- **Méthodes statistiques:** Les méthodes statistiques de suivi de points sont, relativement, robustes par rapport aux bruits et aux perturbations, car dans le processus d'estimation de mouvement, elles tiennent en compte des erreurs de mesures et des incertitudes. Pour modéliser le mouvement, elles utilisent une information spatiale qui peut être : la position, la vitesse ou l'accélération de l'objet à suivre.
- **Méthodes déterministes :** A la place des fonctions de la densité de probabilité utilisées pour l'estimation du mouvement dans les méthodes statistiques, ces méthodes, dites déterministes, utilisent des mesures quantitatives sur des contraintes bien déterminées, comme la régularité et la rigidité. L'estimation du mouvement se fait par l'optimisation de ces mesures, où à chaque objet mesuré, dans une image à l'instant t , correspond un seul objet de la même mesure à l'instant $t+1$. L'avantage de ces méthodes est qu'on peut rajouter des contraintes additionnelles pour gérer le mouvement, ce qui pourrait aussi augmenter le temps de calculs.

Les domaines d'application des méthodes de suivi d'objets par un point, sont très limités vue le peu d'informations qu'on peut extraire de l'objet et son mouvement.

Les difficultés persistantes pour le suivi par point sont en principe :

- l'occlusion : le chevauchement des objets ayant les mêmes caractéristiques.
- la mauvaise détection : détection d'un bruit en tant que point d'intérêt, ou bien, considération des points d'intérêt en tant que bruit.
- changement de topologie : le changement de topologie n'est pas géré automatiquement.

1.5.2 Suivi par forme géométrique de base

L'idée est de suivre le mouvement d'un objet représenté par une forme géométrique de base (ellipse, rectangle, cercle, etc.) d'une image à une autre. La représentation de l'apparence utilisée, le nombre d'objets à suivre et la méthode utilisée pour estimer le mouvement font la différence entre les algorithmes de

cette approche. Selon la représentation de l'apparence utilisée, on peut distinguer deux catégories d'algorithmes :

a. Modèles se basant sur la densité de probabilité d'un critère de sélection à l'intérieur d'une forme de base donnée [11]

Ce sont des modèles largement utilisés grâce à leur relative simplicité. Selon le nombre d'objets à suivre, ces modèles peuvent se subdiviser en deux catégories :

- **Suivi d'un seul objet** : Le principe est de trouver dans l'image courante de la séquence, une région ayant des caractéristiques similaires à celles définies pour une forme géométrique de base dans l'image précédente. Le critère de sélection peut être soit l'intensité ou bien la couleur. *Birchfield* (1998) [8] a apporté une solution moins sensible aux changements de luminance, où il a utilisé le gradient de l'image comme critère de sélection. Le problème majeur de ces méthodes est le temps de calcul assez important. Chercher la région d'intérêt, au voisinage de son précédent emplacement, est la solution proposée par *Schweitzer et al.* (2002) [9], en vue de réduire le temps de calcul.
- **Suivi de plusieurs objets** : La modélisation du mouvement d'un seul objet ne prend pas en compte les interactions entre les objets de l'image, et les objets avec le fond de l'image. Dans les approches de suivi de plusieurs objets, le fond et les objets en mouvement sont indépendamment représentés et suivis. *Tao et al.* (2002)[10] ont proposé de représenter l'ensemble de l'image par une superposition d'un ensemble de couches, où le fond est représenté par une seule couche, et chaque objet en mouvement est représenté par une ellipse, dans la couche correspondante. Chaque ellipse est modélisée par les informations statistiques de l'objet (la région) qu'elle suit.

b. Suivi par des modèles à plusieurs vues d'apparence

Dans les méthodes précédentes, les algorithmes de suivi sont appliqués directement sur la séquence, donc les informations, prises connues à priori, sont supposées invariantes. Or, en réalité les objets changent souvent leurs apparences au cours de la séquence. Si ces changements sont importants, on peut perdre les informations sur l'objet à suivre. Pour y remédier à ce problème,

de différentes vues des objets sont enregistrées au préalable. Elles seront utilisées dans l'algorithme de suivi comme informations dynamiques des objets.

Cependant, malgré le gain en temps de calculs qu'apportent les méthodes de suivi d'objets par une forme géométrique de base, elles restent assez primitives quant à la qualité et la précision de l'information sur l'objet suivi [11]. Ces méthodes sont, généralement, utilisées pour le suivi des objets rigides. Pour le suivi des objets non-rigides, des méthodes plus évoluées se sont spécialisées, comme par exemple les méthodes de suivi de contours.

I.5.3 Suivi de la silhouette

Les objets sont souvent de forme complexe et variante. De ce fait, ils ne peuvent pas être bien décrits par une simple forme géométrique. Les algorithmes qui suivent la silhouette de l'objet sont bien adaptés à leur forme. Le principe est de suivre l'objet à travers toutes les images de la séquence, en se basant sur des connaissances à priori, comme par exemple l'histogramme de couleur, le contour de l'objet, etc.

Les méthodes basées sur le suivi de la silhouette peuvent être divisées en deux catégories différentes :

- **Suivi par correspondance de forme (Shape Matching) :** C'est des méthodes similaires à celles basées sur le suivi d'objets à l'aide d'une forme géométrique de base. Sauf qu'ici, la forme à suivre est celle de l'objet, où le but est de repérer dans chaque image de la séquence une région ayant la même forme et apparence que l'objet. Chercher une forme dans la nouvelle image revient à chercher les points qui la délimitent. Pour cela, des méthodes basées sur le calcul du gradient sont utilisées pour la recherche des bords de l'objet. On peut, aussi, utiliser les champs de vecteurs pour le calcul du mouvement de chaque pixel de la silhouette.
- **Suivi par contours actifs:** Dans ces méthodes, le suivi du mouvement se fait par l'évolution du contour de l'objet à travers les images de la séquence. Contrairement aux méthodes basées sur le suivi par correspondance de forme, celles-là acceptent les déformations s'opérant sur l'objet au cours de la séquence. Deux différentes catégories peuvent se distinguer : celles qui se basent sur la représentation spatiale de tous les points du contour, et celles qui font évoluer le contour par la

minimisation d'une variation d'énergie. La section suivante donnera un rappel rapide sur les contours actifs.

I.6 Contours actifs

Les algorithmes utilisées pour le suivi de contour sont similaires à ceux de la segmentation statique par contour actifs, sauf qu'ici le processus de segmentation est appliqué à chaque image de la séquence, tout en imposant que le contour initial de chaque image soit le contour final de l'image qui la précède.

Le principe du contour actif est de le faire évoluer vers les bords de l'objet à suivre. Cette évolution est gouvernée par la minimisation d'une fonctionnelle d'énergie, qui est formée de descripteurs (basés contour, ou basés région) caractérisant l'objet intéressant. D'une manière générale, cette énergie prend la forme suivante :

$$E(\Gamma) = \int_0^1 E_{int}(v) + E_{im}(v) + E_{contexte}(v) ds \quad (1.1)$$

Avec :

- v : la position d'un point.
- E_{int} : c'est l'énergie interne. Elle comprend souvent un terme qui gère la courbure du contour, et qui gère sa régularité.
- $E_{contexte}$: comprend des contraintes additionnelles.
- E_{im} : peut être vue comme l'énergie potentielle du contour, elle contient soit des termes descripteurs locaux comme par exemple le gradient aux alentours de l'objet d'où la nomination basé contour, ou soit des termes globaux comme les informations sur la couleur ou la texture dans les régions internes et externes au contour, d'où la nomination basé région.

La restriction majeure dans les méthodes basées-contour, est que le contour doit être initialisé pas très loin de l'objet, afin qu'il soit détecté. Dans les méthodes basées-région cette restriction peut être enlevée, et le contour peut être initialisé : soit à l'intérieur, à l'extérieur de l'objet ou à n'importe quel autre endroit. Récemment, l'énergie basée sur des termes basés-région et des termes basés-contour est très répandue dans les algorithmes de suivi d'objets par contour actifs. *Paragios et Deriche (2002) [12]*, ont utilisé le gradient de l'image et des termes basés région pour formuler l'énergie :

$$E_{image} = \lambda E_{contour} + (1 - \lambda) E_{région} \quad (I.2)$$

où λ est une constante positive.

En plus de l'initialisation et de la fonctionnelle d'énergie à minimiser. Les recherches se sont basées aussi sur la représentation optimale du contour actif. Ainsi, un contour peut être représenté explicitement (courbe plane), ou implicitement (Level set). Dans une représentation explicite les relations entre les différents points du contour et de l'image sont représentées par des équations. Par contre, dans une représentation implicite, l'image est vue comme une grille, où les distances de tous les points de la grille par rapport au contour sont encodées positivement ou négativement, selon leur position par rapport au contour. L'évolution du contour se fait par le changement des valeurs de la grille évaluée en chaque point, et sera gouvernée par une fonctionnelle d'énergie à minimiser. L'apport le plus important de la représentation implicite est qu'elle est flexible aux changements de topologie (division et fusionnement du contour).

I.7. Conclusion

La conclusion importante qu'on peut tirer de cette étude est l'inexistence d'un modèle normalisé, que ce soit pour la représentation, la détection, ou le suivi d'objets.

Selon l'application et le type d'objet qui nous intéresse, et en tenant compte des contraintes qui peuvent surgir, on peut choisir un modèle de représentation bien spécifié. Par exemple, si on est appelé à suivre un objet de taille minuscule, il suffirait de représenter sa forme par un point. Par contre, si l'objet était de taille plus importante, rigide ou pas, sa représentation de forme pourrait être une forme géométrique de base, ou une représentation de silhouette, combinée avec une représentation d'apparence qui peut être aussi l'intensité, ou la couleur.

Aussi, l'algorithme utilisé pour le suivi d'objet est fortement lié à la représentation choisie. En effet, si le mouvement était une translation, la représentation par forme géométrique de base est bien adaptée. D'autre part, si on a affaire à un mouvement complexe, la représentation et le suivi par contours actifs est recommandé.

Etant donné que notre travail se porte sur l'adaptation d'une approche de suivi d'objet en mouvement par contours actifs, et pour appuyer notre choix, un aperçu sur les méthodes de contours actifs sera développé dans le chapitre suivant.



Les contours actifs



« Imagination is more importante than knowledge »

A Einstein

11.1 Introduction

Les contours actifs ont été introduits, pour la première fois, par l'équipe : Kass, Witkin et Terzopoulos (1988), qui ont mis en œuvre les snakes. Ces derniers tiennent leur nom de leur aptitude à se déformer comme des serpents. Depuis la publication de cette équipe, les contours actifs sont devenus un thème de recherche d'une importance majeure. Plusieurs équipes s'y sont intéressées. Leurs domaines d'utilisation sont nombreux tant en 2D qu'en 3D tels : la reconnaissance de formes, la simulation, le suivi, la segmentation d'images, etc.

Dans ce chapitre, nous allons s'introduire au modèle du contour actif, en exposant son principe et ses quelques modèles.

11.2 Définition des contours actifs

Les contours actifs sont définis comme une courbe continue fermée ou non, à extrémités fixes ou non. Ils se déforment à partir d'une position initiale située près de l'objet d'intérêt pour quelques modèles, mais pas nécessairement pour d'autres [13]. Le modèle est soumis à des forces qui le déforment.

Soit Γ une courbe déformable spatialement et temporellement. Son domaine de définition est normalisé à $[0, 1]$. Soient $v(s, t)$ la position d'un point de la courbe à un instant t et x, y ses coordonnées cartésiennes dans l'image (Figure 11.1) :

$$v(s, t) = (x(s, t), y(s, t)) \quad \forall (s, t) \in [0, 1] \times [0, +\infty] \quad (11.1)$$

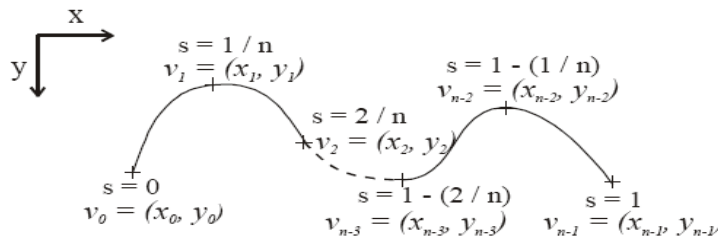


Figure 11.1 Exemple de coordonnées cartésiennes et abscisses curvilignes.

Avec s le paramètre de position spatiale le long de la courbe.

Cette représentation nous renseigne que les contours actifs peuvent être un outil de résolution d'un problème de minimisation d'une fonctionnelle d'énergie.

11.3 Energies d'un contour actif

La fonctionnelle d'énergie, caractérisant le contour actif, peut être écrite sous la forme générale représentée dans l'équation (11.2).

$$E_{\text{contour}} = E_{\text{interne}} + E_{\text{externe}} + E_{\text{contexte}} \quad (11.2)$$

11.3.1 Energie interne

L'énergie interne gère la cohérence de la courbe. Elle maintient la cohésion des points et la raideur de la courbe. C'est un terme de régularisation [15] donné par l'équation (11.3).

$$E_{\text{int}} = \int_0^1 (a \|v'\|^2 + b \|v''\|^2) ds \quad (11.3)$$

Où : les termes v' et v'' sont les dérivées premières et secondes de v par rapport à s . a et b sont des constantes.

Le terme de 1^{er} ordre correspond à la tension. Il prend une valeur importante quand la courbe se distend. Lorsque $a = 0$ la courbe peut présenter des discontinuités, il s'agit donc d'une énergie de continuité.

Le terme de 2^{ème} ordre correspond à la courbure. Il prend une valeur importante lorsque la courbe est très convexe. Si $b = 0$, la courbe prend une grande convexité, mais lorsque b est très grand, la courbe tend vers un cercle.

11.3.2 Energie externe

L'énergie interne fait la régularisation du contour, tandis que l'énergie externe contient des termes d'attache aux données de l'image, qui peuvent être des critères descriptifs de l'apparence ou la forme de l'objet à suivre. Ils peuvent être basés contour ou bien basés région. Dans ce qui suit on citera deux exemples de termes d'énergie externe.

- Le gradient : C'est un terme d'énergie externe qui se base sur le calcul du gradient. Il peut s'écrire sous la forme suivante :

$$E_{\text{externe}(v)} = - \int_0^1 \|\nabla I(v(s))\|^2 ds \quad (11.4)$$

Où $\nabla I(v(s))$ représente le gradient de l'image I .

Très souvent, c'est le gradient gaussien qui est utilisé, comme dans l'équation (11.5):

$$E_{\text{externe}}(v) = - \int_0^1 \|\nabla(g\sigma * I)(v(s))\|^2 ds \quad (11.5)$$

Où : g_σ est la Gaussienne centrée d'écart type σ .

- L'intensité : l'intensité peut être le critère de sélection qui nous permet de reconnaître l'objet à suivre. En fonction de ce critère, il y a des termes d'énergies basées régions qui appliquent une force sur le contour, afin de le guider à la satisfaction de ce critère. Par exemple, la méthode de Chan et Vese utilise la moyenne des intensités pour distinguer les objets d'intérêt et le fond selon l'expression :

$$E_{\text{externe}}(v) = \int_{\Omega} |I(v) - c1|^2 dv \quad (11.6)$$

Où : Ω est le domaine de l'image, $I(v)$ la moyenne des intensités de l'image, $c1$ la moyenne des intensités à l'intérieur du contour.

11.3.3 Energie de contexte

L'énergie de contexte, appelée aussi énergie de contrainte, permet l'introduction des connaissances à priori sur l'objet recherché. Comme exemple, on peut citer, l'énergie ballon introduite par *Laurent D. Cohen* en 1991.

11.4 Modélisation d'un contour actif selon sa représentation

Si la définition de l'énergie constitue une étape fondamentale des méthodes de segmentation par contours actifs, la conception du modèle du contour représente, elle aussi, un enjeu majeur quant à la précision et la stabilité de la méthode.

Principalement, deux approches s'opposent pour la modélisation des contours actifs. La première, qualifiée de formulation Lagrangienne [13] représente les contours actifs de manière explicite (ces contours sont souvent paramétriques). La seconde, qualifiée de formulation Eulérienne [13], représente les contours actifs de manière implicite.

11.4.2 Approche explicite

Parmi les modèles représentés explicitement, on peut considérer que les snakes sont les premiers exemples de contours actifs ayant des propriétés intrinsèques. Ce modèle de courbe élastique, introduit par Kass et *al.* a pour propriété de se positionner autour de l'objet qu'on désire segmenter, en minimisant la fonctionnelle (11.2). Cependant, pour ce genre de modèles qui sont dits basés contour, l'initialisation du contour actif demeure un problème, puisque si le contour est initialisé dans une zone où il n'y a plus d'énergie potentielle (force externe), le contour n'évoluera plus, donc, la convergence n'est jamais atteinte. Ce qui explique l'aspect local des contours actifs basés contour.

Par ailleurs, il existe d'autres méthodes qui sont représentées explicitement et qui peuvent s'affranchir de la contrainte d'initialisation, c'est celles basées région [15].

11.4.1 Approche implicite

Plusieurs méthodes de segmentation par contours actifs se basent sur la modélisation implicite. Parmi ces approches, on trouve celle utilisant les ensembles de niveaux, introduites par Osher et *al.* Son principal avantage est de permettre la gestion automatique des changements de topologie. En effet, le principe de la méthode consiste à considérer le contour actif comme le niveau zéro d'une fonction $\phi(s, t)$ de dimension supérieure (Figure 11.2), avec :

$$\phi(s, t + dt) = \phi(s, t) + \Delta t V(s, t) \quad (11.7)$$

Où : V est la vitesse de propagation. Cette formule, qui est représentée d'une manière générale, gère l'évolution du contour. La fonction ϕ étant une fonction signée de distances par rapport au contour.

Les principaux avantages de cette approche sont :

- * La courbe (ou le contour) peut se diviser ou se fusionner au cours de son évolution : d'où la gestion automatique des changements de topologie.
- * Les résultats obtenus sont très précis, vu que chaque pixel du contour est représenté et repéré par rapport à chaque pixel appartenant à l'image.

- * Par la connaissance des positions de chaque point du contour, on peut reconnaître sa forme.
- * Le contour, considéré comme la courbe de niveau 0, est défini par son voisinage, donc il ne possède pas de structure intrinsèque (paramétrique) (Figure 11.2).

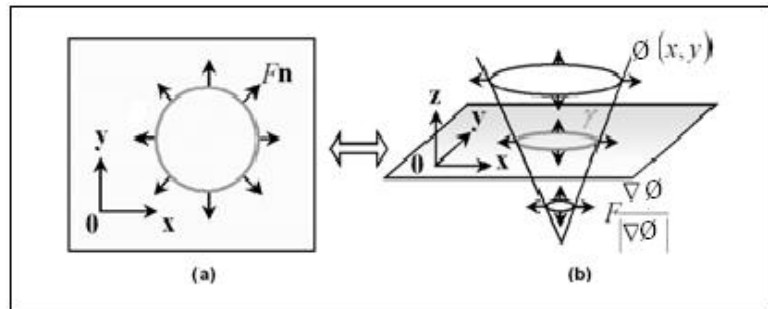


Figure 11.2 Représentation implicite d'un contour actif.

11.5 Approches de contours actifs selon l'énergie externe

De puis la découverte des contours actifs, plusieurs algorithmes se sont développés pour essayer de leurs apporter des améliorations, dans le but de les optimiser tout en s'affranchissant du maximums de contraintes et de limites. Dans la section suivante, on exposera quelques modèles de contours actifs dits basés contours et d'autres qui sont dits basés région.

11.5.1 Contours actifs basés contours

Ils ne tiennent en compte que de l'information présente sur les bords des objets à segmenter. Les contours des objets sont, en général, définis par une forte valeur du gradient. Ce terme est représenté selon l'expression suivante :

$$J_c(\Gamma) = \int_{\Gamma} k_c(s) da \quad (11.8)$$

Où k_c est le descripteur du contour Γ .

da : Élément de longueur.

L'évolution du contour peut se faire sans passer par la minimisation de la fonctionnelle d'énergie, mais par la minimisation d'une équation d'évolution définie au préalable [14]. Un exemple est donné par l'équation proposée par Caselles et al. :

$$\frac{\partial \Gamma}{\partial t} = g(|\nabla I(\Gamma)|)mN \quad (11.9)$$

Avec : Γ le contour, m une constante, N la composante normale de la vitesse d'évolution et g une fonction d'arrêt.

L'évolution du contour est faite, soit d'une manière variationnelle (minimisation d'un critère d'énergie), ou bien par l'approche dite par EDPs où on introduit directement l'équation d'évolution [15].

Parmi, les approches de cette catégorie, on citera les contours actifs géodésiques. Pour minimiser les aspects négatifs du premier modèle de contours actifs (snakes) proposé par Kass et *al.*, Caselles et *al.* ont proposé les contours actifs géodésiques (CAG). Géodésique parce qu'au lieu d'utiliser directement le gradient, comme terme d'attache aux données, ils ont plutôt utilisé une fonction du gradient $g(|\nabla I|)$, ce qui est une caractéristique des courbes géodésique dans l'espace Riemannien (une courbe géodésique est un chemin de distance minimale entre des points donnés) :

$$J(\Gamma) = \alpha \int_a^b |\Gamma'(s)|^2 ds + \beta \int_a^b |\Gamma''(s)|^2 ds - \lambda \int_a^b |\nabla I(\Gamma(s))|^2 ds \quad (\text{II.10})$$

avec : α, β, λ , sont des constantes positives.

En plus à cette modification, l'équipe a jugé inutile d'utiliser le terme de rigidité, qui est à l'origine de plusieurs inconvénients des *snakes*. Ils ont, alors, pris $\beta = 0$, puisqu'ils estiment que les autres termes de la fonctionnelle ont aussi un rôle dans la régularisation du contour. Les auteurs ont abouti à la fonctionnelle suivante :

$$J(\Gamma) = \alpha \int_a^b |\Gamma'(s)|^2 ds + \lambda \int_a^b g(|\nabla I(\Gamma(s))|)^2 ds \quad (\text{II.11})$$

La fonction g est une fonction de détection, $g : [0, \infty[\rightarrow \mathbb{R}^+$, elle est décroissante de telle sorte que : $g(r) \rightarrow 0$, lorsque $r \rightarrow \infty$. Elle est, généralement, donnée sous la forme $g(r) = \frac{1}{1+r^m}$, avec $m = 1$ ou 2 et r un terme contenant le gradient.

Cependant, la méthode n'est toujours pas intrinsèque puisqu'elle dépend de la paramétrisation de la courbe. Les auteurs ont montré que la minimisation de l'équation (II.11) est équivalente à la minimisation du terme de l'équation (II.12), qui combine le terme d'attache et celui de lissage :

$$J(\Gamma) = \int_a^b g(|\nabla I(\Gamma(s))|) |\Gamma'(s)| ds \quad (\text{II.12})$$

Cette dernière fonctionnelle, présente l'avantage d'être intrinsèque. La démonstration du passage de l'équation (II.11) à l'équation (II.12) se trouve dans [15]. Après minimisation, ils ont obtenu l'équation (II.13) suivante :

$$\frac{\partial \Gamma}{\partial t} = g(I)k\vec{N} - (\nabla g \cdot \vec{N})\vec{N} \quad (\text{II.13})$$

Avec k la courbure du contour Γ .

Néanmoins, malgré les avantages de ce modèle (CAG), il reste condamné par quelques lacunes comme : la sensibilité aux minima locaux et la sensibilité par rapport à l'initialisation du contour.

Le modèle que nous avons choisi de présenter dans le chapitre suivant et qui n'est autre que l'approche adoptée, tire du modèle géodésique l'avantage de la segmentation locale, à condition d'initialiser le contour près de l'objet à segmenter.

II.5.2 Les contours actifs basés région

Le principal objectif des contours actifs basés région est d'introduire des informations globales dans l'équation d'évolution du contour, en plus aux informations locales contenues dans les termes basés contour. Grâce aux termes basés région, le contour peut s'accroître ou se rétrécir, selon la force générée par la différence entre les deux régions séparées par le contour. Ils ont aussi une influence sur la vitesse d'évolution du contour [15].

Ces termes sont représentés par des intégrales de région, selon la formulation suivante :

$$J_r(\Omega) = \int_{\Omega} (k, \Omega) ds \quad (\text{II.14})$$

Où k est le descripteur de la région Ω .

Historiquement, les termes basés région étaient utilisés, pour la première fois, dans les travaux de *Cohen et al.* [16], pour présenter une méthode de reconstruction de surface avec contours actifs.

Un autre travail a été fait par *Ronfard* [17], pour la segmentation d'image en deux régions. L'auteur a eu l'idée d'utiliser la force induite par la différence entre les caractéristiques statistiques de la région à segmenter et celle du fond de l'image. C'est aussi le principe utilisé pour la segmentation dans la méthode que nous avons adopté, où l'information statistique est la moyenne des intensités à l'intérieur et à l'extérieur de l'objet.

Pour bien illustrer le principe de segmentation basée région, on prend l'exemple de segmenter une image en deux régions distinctes. Soit une image composée de deux régions : Ω_{int} et Ω_{ext} , où Ω_{int} est la région de l'objet à segmenter. L'interface entre ces deux régions est le contour, qui est noté par G . Chaque région forme une partition de l'image. On a donc $\Omega_{int} \cup \Omega_{ext} \cup \Gamma = \Omega_I$, où Ω_I est le domaine de l'image et $\Omega_{int} \cap \Omega_{ext} = \emptyset$. Nous prendrons $G = \partial \Omega_{int}$. On peut, alors, introduire une fonctionnelle générale à minimiser (II.15) :

$$J(\Omega_{int}, \Omega_{ext}, \Gamma) = \int_{\Omega_{int}} k_{int}(s, \Omega_{int}) ds + \int_{\Omega_{ext}} k_{ext}(s, \Omega_{ext}) ds + \int_{\Gamma} k_c(s) da \quad (II.15)$$

Avec : k_{int} : désigne le descripteur de l'objet à segmenter,

k_{ext} : désigne le descripteur de la région du fond,

k_c : le descripteur du contour.

L'objectif, alors, est de faire évoluer un contour initial vers la solution optimale de cette fonctionnelle.

Plusieurs modèles de contours actifs basés régions existent dans la littérature, parmi eux le modèle de *Chan* et *Vese*. Introduit par *F. Chan* et *A. Vese* en 2001 [18], ce modèle est dit basée région et il implémenté implicitement par ensembles de niveaux. Contrairement aux méthodes basées contour, qui utilise la descente du gradient pour détecter le contour, l'approche de *Chan* et *Vese*, utilise des informations sur la région à détecter et son fond. Le déséquilibre d'intensité entre la région interne et externe du contour et l'intensité de l'image, génère la force qui est susceptible de mener le contour vers les bords de l'objet et puis le stabiliser à l'égalité des intensités.

Soit : $\Gamma = \{(x) \in \Omega_I : \phi(x) = 0\}$, $C1 = \{(x) \in \Omega_I : \phi(x) > 0\}$ et $C2 = \{(x) \in \Omega_I : \phi(x) < 0\}$

Où : f la fonction level-set, $C1$: l'intensité à l'intérieur du contour Γ et $C2$: l'intensité à l'extérieur du contour Γ . L'énergie basée région à minimiser est une combinaison des deux termes $F1$ et $F2$:

$$F1 = \int_{\text{intérieur}(\Gamma)} |I(s) - c1|^2 ds \quad (II.16)$$

$$F2 = \int_{\text{extérieure}(\Gamma)} |I(s) - c2|^2 ds \quad (II.17)$$

On voit bien que l'énergie $E_{\text{région}}$ ne s'annule que si les deux termes $F1$ et $F2$ sont nuls. Si $F1 > F2$ le contour va dans le sens de la région externe, donc, il va s'accroître. Si $F1 < F2$, il va dans le sens de compenser la différence, donc, il va se rétrécir.

En plus à cette énergie basée région utilisée, d'autres termes sont rajoutés pour régulariser le contour :

- * $\mu \int d(\mathcal{E}) |\nabla I| ds$: c'est le terme de courbure, il a pour rôle la régularisation du contour.
- * $v \int H(\mathcal{E}) ds$: ce terme est rajouté pour accroître la vitesse d'évolution du contour.

Ainsi, l'écriture formelle de la fonctionnelle d'énergie à minimiser sera (II.18) :

$$E_{(c1,c2,\Gamma)} = \mu \int d(\mathcal{E}) |\nabla I| ds + v \int H(\mathcal{E}) ds + \lambda_1 \int |I(s) - c1|^2 H(\mathcal{E}) ds + \lambda_2 \int |I(s) - c2|^2 (1 - H(\mathcal{E})) ds \quad (II.18)$$

Avec : $H(\mathcal{E})$ la fonction Heaviside, $d(\mathcal{E})$ la fonction de Dirac et $\mu, v, \lambda_1, \lambda_2$ Sont des constantes positives.

Pour le calcul de la moyenne des intensités à l'intérieur et à l'extérieur du contour ($C1$ et $C2$), la méthode utilise les formules suivantes:

$$C1(\mathcal{E}) = \text{moyenne}(I)_{\text{sur}(\mathcal{E} \geq 0)} = \frac{\int_{\Omega_I} I \cdot H(\mathcal{E}) ds}{\int_{\Omega_I} H(\mathcal{E}) ds} \quad (II.19)$$

$$C2(\mathcal{E}) = \text{moyenne}(I)_{\text{sur}(\mathcal{E} < 0)} = \frac{\int_{\Omega_I} I \cdot (1 - H(\mathcal{E})) ds}{\int_{\Omega_I} (1 - H(\mathcal{E})) ds} \quad (II.20)$$

L'estimation des fonctions Heaviside et Dirac sont comme suit [18] :

$$H_\epsilon(z) = \frac{1}{2} \left(1 + \frac{2}{\pi} \arctan \left(\frac{z}{\epsilon} \right) \right) \quad (II.21)$$

$$d(z) = H'_\epsilon(z) = \frac{1}{\pi \epsilon} \frac{1}{\left(1 + \left(\frac{z}{\epsilon} \right)^2 \right)} \quad (II.22)$$

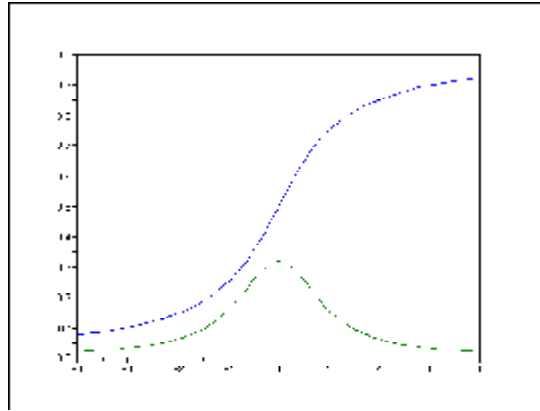


Figure 11.3 Représentation de H (bleu), et d (vert).

De la fonctionnelle d'énergie représentée dans l'équation (11.18), et à l'aide du théorème d'Euler-Lagrange, les auteurs ont déduit l'équation aux dérivées partielles à minimiser, qui est la suivante :

$$\frac{\partial \Gamma}{\partial t} = \delta_{\epsilon}(\phi) \left[\mu \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - v - \lambda_1 (I - C_1)^2 + \lambda_2 (I - C_2)^2 \right] \quad (11.23)$$

Avec :

- * μ constante positive, coefficient du terme de courbure, qui a pour rôle la régularisation du contour.
- * v constante positive, rajoutée pour augmenter la vitesse de propagation du contour.

Ce modèle présente beaucoup de qualités, notamment, la capacité de gérer automatiquement les changements de topologie et l'insensibilité par rapport à l'initialisation du contour. Cependant, malgré toutes ces qualités, cette approche reste très coûteuse en temps de calculs. Cela est dû au calcul de la carte des distances pour le contour (figure 11.4), à chaque itération de son évolution.

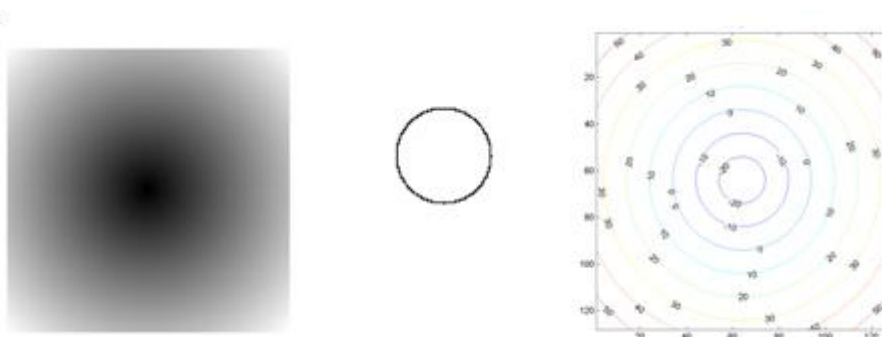


Figure 11.4 exemple de la carte des distances.

Pour essayer d'adapter les méthodes d'ensembles de niveaux aux applications en temps réel, plusieurs algorithmes ont été proposés pour accélérer le temps de convergence du modèle. On citera par exemple, l'approche dite de bande étroite (narrow band, en anglais) (Figure 11.5). Cette approche procède par la définition d'un voisinage interne et externe au contour. Chaque fois qu'on sera appelé à localiser le contour, on le fera uniquement par rapport aux points de la bande qui engendrent le contour. Le calcul des différents descripteurs de la fonctionnelle se fait uniquement à l'intérieur de la bande. Cette méthode a présenté de très bons résultats quant à la réduction du temps de calculs [19] (Tab. 11.1).

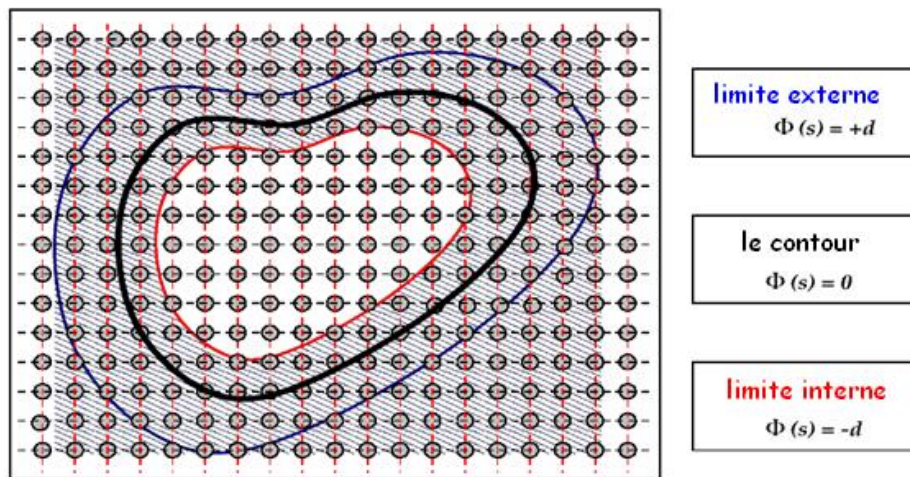


Figure 11.5 Méthode de bande étroite.

Dimension	level-set classique	level-set (narrow band)
64X64	0,0021s	0,000135s
128X128	0,0156s	0,000569s
256X256	0,144s	0,00234s
512X512	1,527s	0,0101s

Tab 11.1 Tests sur différentes images d'après [19].

11.6 Conclusion

Dans ce chapitre, nous avons présenté différents modèles de contours actifs et nous les avons classé selon la fonctionnelle d'énergie à minimiser (basée contour et basée région) et selon le type de représentation, à savoir, la représentation explicite (paramétrique), et la représentation implicite

(ensembles de niveau). Nous avons essayé de donner quelques-uns des avantages ainsi que les inconvénients que présente chacun de ces modèles.

Les avantages et les inconvénients sont relatifs aux types d'applications à réaliser. Par exemple, pour réaliser une application de suivi d'objets en temps réel, il est recommandé d'utiliser une représentation explicite du contour, ou bien, une méthode implicite accélérée.

La méthode que nous avons adoptée, dans le chapitre suivant, donne de bons résultats sur les applications en temps réel. Elle ne se base pas sur la solution de bande étroite, comme l'approche décrite ci-haut, pour être rapide, elle ne calcule pas la carte des distances.

Chapitre

3

Approche adoptée



« L'approche est toujours plus belle que l'arrivée »

Henri Alban Fournier.

III.1 Introduction

Notre objectif était de trouver un modèle de contour actif pourra être appliqué au suivi d'objet en mouvement en temps réel. En se basant sur les informations données dans les deux chapitres précédents, nous avons pu construire un algorithme de suivi d'objet en mouvement, et cela en combinant les points suivants : la représentation de l'objet à suivre, la détection de mouvement et l'algorithme de segmentation statique ou dynamique.

Ainsi, dans ce chapitre, nous allons procéder par la présentation du modèle mathématique de l'approche adoptée, son implémentation et les étapes de son application à une segmentation statique et dynamique.

III.2 Présentation du modèle de contour actif adopté

On rappelle qu'il y a deux catégories de contours actifs, ceux basés contour qui ont la propriété de segmentation locale, et ceux basés région qui ont la propriété de segmentation globale.

Notre but est de trouver une méthode assez rapide, ayant à la fois, la propriété de segmentation locale et globale. Pour ce faire, le modèle adopté [20] est basé sur le modèle de contours actifs géodésique (CAG) (section II.5.1) qui est basé contour, et le modèle de Chan-Vese (C-V) (section II.5.2) qui est basé région. Le modèle est implémenté implicitement par ensemble de niveaux (*Level-set*).

On rappelle, que la fonctionnelle à minimiser dans les CAG est donnée sous la forme suivante :

$$\frac{\partial \Gamma}{\partial t} = g|\nabla I| \left(\operatorname{div} \left(\frac{\nabla I}{|\nabla I|} \right) + \alpha \right) + \nabla g \cdot \nabla I \quad (\text{III. 1})$$

Où : g est la fonction d'arrêt.

Par contre, la fonctionnelle du modèle de Chan-Vese (II.18) est donnée comme suit :

$$\frac{\partial \Gamma}{\partial t} = \delta g \left[\mu \nabla \left(\frac{\nabla I}{|\nabla I|} \right) - \nu - \lambda_1 (I - C_1)^2 + \lambda_2 (I - C_2)^2 \right] \quad (\text{III. 2})$$

Comme nous l'avons déjà évoqué, ce modèle adopté combine les avantages des deux modèles CAG et C-V pour sa construction. En effet, il utilise l'information statistique qu'il calcule à base des C_1 et C_2 de l'équation (III.2) et exploite la formulation de l'équation (III.1) pour aboutir à un modèle pour implémentation en remplaçant la fonction g de (III.1) par une autre fonction dite SPF.

- La force signée de pression (SPF) : C'est une force générée par la différence de la moyenne des intensités à l'intérieur et à l'extérieur de l'objet (C_1 et C_2 respectivement) et leur moyennes. Cette force applique une pression incitant le contour à se rétrécir ou s'accroître selon son signe. La formule de la SPF est donnée par l'équation suivante :

$$SPF(I(x)) = \frac{I(x) - \frac{C_1 + C_2}{2}}{\max \left(\left| I(x) - \frac{C_1 + C_2}{2} \right| \right)} , x \in \Omega_I \quad \text{III.3}$$

La représentation de l'apparence est faite par la moyenne des intensités des différentes régions de l'image. Dans la formule (III.3), le numérateur donne le signe de la SPF, ainsi gérer la sens de l'évolution du contour. Le dénominateur lui, est introduit pour donner une valeur à la SPF allant de $[-1, 1]$, i.e. la normalisation. Une fois que la SPF de l'équation (III.3) est calculée, elle sera remplacée dans l'équation (III.1) au lieu de la fonction g . Ce qui nous permet d'aboutir à la formulation suivante :

$$\frac{\partial \Gamma}{\partial t} = \mathbf{f} \mathbf{sp}(I) \cdot \left(\left(\text{div} \left(\frac{\nabla I}{|\nabla I|} \right) + \alpha \right) |\nabla I| + \nabla \mathbf{f} \mathbf{sp}(I) \cdot \nabla I \right) \quad \text{III.4}$$

Avec : $\text{div} \left(\left(\frac{\nabla I}{|\nabla I|} \right) \right)$ le terme de courbure, il a pour rôle la régularisation du contour.

En plus à l'introduction de l'information région dans la formule (III.2), le modèle adopté s'en passe des termes suivants :

- $\nabla \mathbf{f} \mathbf{sp}(I(x)) \cdot \nabla I$: C'est le terme basé contour utilisé pour attirer le contour vers les bords des objets à détecter. Ce qui est assuré, d'une manière plus robuste [20], avec la nouvelle force basée région SPF.
- $\text{div} \left(\frac{\nabla I}{|\nabla I|} \right)$: C'est un terme de courbure utilisé pour régulariser le contour. Dans le modèle qu'on a adopté, la fonction Level-set est initialisée en tant que constante ayant un signe opposé à l'intérieur et à l'extérieur du contour, ce qui fait que $|\nabla \mathcal{E}| = 1$ [20]. Le terme de

régularisation peut être remplacé par $\Delta \mathcal{E}$, qui est le Laplacien de $\mathcal{E}$. En se basant sur la théorie de (scale-space) [20], on peut dire que l'évolution de la fonction level-set ($\mathcal{E}$) avec le Laplacien, est équivalent au filtrage avec le noyau Gaussien (Annexe B). En s'inspirant de cette idée, le modèle utilise un filtre à noyau Gaussien, indépendamment, du critère d'évolution afin de régulariser le contour, et cela, on effectuant un produit de convolution après chaque itération du processus de convergence.

Nous aboutissons au modèle adopté suivant, qui sera implémenté par les ensembles de niveaux :

$$\frac{\partial \phi}{\partial t} = SPF(I(v)) \cdot \alpha |\nabla I|, \quad v \in \Omega. \quad \text{III.5}$$

Où : α une constante qui est rajoutée pour augmenter la vitesse de l'évolution.

Si nous voulons segmenter un seul objet désiré (segmentation locale), les conditions suivantes doivent être satisfaites :

→ La fonction ϕ ne doit prendre que deux valeurs possibles :

• $\phi = +1$ pour toute les valeurs positive de ϕ .

• $\phi = -1$ ailleurs.

→ Le contour doit être initialisé tout près de l'objet à détecter.

→ Le signe de la SPF doit être choisit d'une manière à ce que l'évolution du contour se fera du milieu moins intense au plus intense, dans le cas où l'objet est d'intensité moindre par rapport à celle de l'image, ou bien, du milieu le plus intense au moins intense et cela si l'objet à segmenter est d'intensité plus importante que celle de l'image.

Le modèle adopté présente plusieurs avantages par rapport à ceux de CAG et C-V, ce qui nous a motivés à l'adopter pour notre application. Par ailleurs, une petite comparaison entre les trois modèles justifie davantage notre choix. Ainsi, nous pouvons résumer ces comparaisons en ces quelques points :

- * Contrairement au modèle CAG, le modèle adopté utilise l'information statistique pour faire évoluer et arrêter la courbe sur les bords des objets. Il est, donc, plus robuste et résistant aux bruits.

- * Ce modèle peut s'appliquer avec succès sur des images avec de faibles bords (Figure III.1).

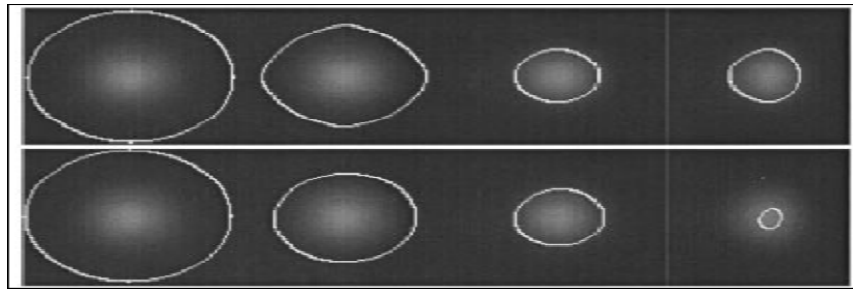


Figure III.1 Segmentation par l'approche adoptée en haut et segmentation par CAG en bas.

- * Le modèle adopté peut détecter les bords internes des objets en initialisant le contour n'importe où dans l'image.
- * Comparé au modèle de C-V, ce modèle permet l'extraction d'objets ayant une intensité interne inhomogène.
- * Le modèle adopté peut, sélectivement, extraire l'objet désiré en démarrant le contour initial en entourant ou en interceptant les bords de l'objet à détecter (segmentation locale), alors que le modèle C-V extrait tous les objets.
- * le signe de la SPF peut contrôler la direction d'évolution de notre contour.
- * Le modèle présenté à moins de complexité de calcul par rapport aux modèles C-V et GAC.

III-3 Implémentation

Le travail d'implémentation que nous avons effectué se résume par l'organigramme de la figure (III.2).

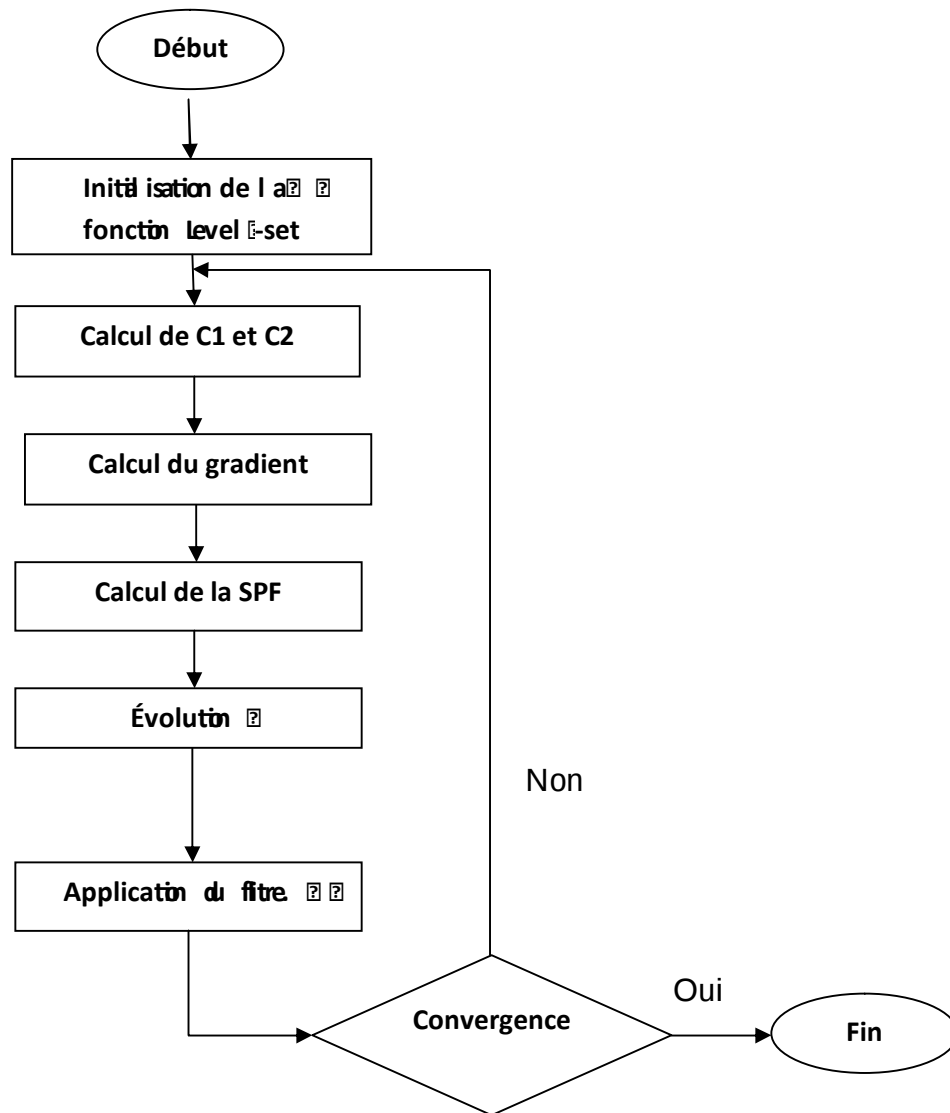


Figure III.2 Organigramme de segmentation.

Les différentes étapes de l'algorithme sont décrites comme suit :

- a. Initialisation de la fonction Level-set ($\emptyset$) : Toutes les formes géométriques de base (rectangle, cercle, ellipse, etc.) peuvent être prises comme forme initiale du contour. L'initialisation de $\emptyset$ se fait selon l'algorithme suivant :

```

Début
Pour chaque (x, y) de  $\emptyset$ 
  Si (x, y) est à l'intérieur
     $\emptyset(x, y) = -1$ ;
  Sinon
    Si (x, y) est à l'extérieur
       $\emptyset(x, y) = 1$ ;
    Sinon
       $\emptyset(x, y) = 0$ ;
    Fin si;
  Fin si;
Fin pour;
Fin.

```

b. Calcul des intensités moyennes à l'intérieur et à l'extérieur du contour :

```

Début
  Intérieur = trouver ( $\emptyset < 0$ );
  Extérieur = trouver ( $\emptyset > 0$ );
  c1 = sum (Img (intérieur)) / size (intérieur);
  c2 = sum (Img (extérieur)) / size (extérieur);
Fin.

```

Avec Img (intérieur) et Img (extérieur) les intensités des pixels respectivement, à l'intérieur et extérieur du contour.

c. L'évolution de la fonction level-set se fait par l'algorithme :

```

Début
   $\emptyset_x(x, y) = (\emptyset(x+1, y) - \emptyset(x-1, y)) / 2$ ;
   $\emptyset_y(x, y) = (\emptyset(x, y+1) - \emptyset(x, y-1)) / 2$ ;
  Grad =  $\sqrt{\emptyset_x(x, y)^2 + \emptyset_y(x, y)^2}$ 
  spf =  $1 - (c1 + c2) / 2$ ;
  spf = spf / max|spf|;
   $\emptyset = \emptyset + \alpha * \text{spf} * \text{Grad}$  ;
Fin.

```

d. Pour la régularisation de la fonction Level-set $\emptyset$, nous l'avons convolué avec un filtre de noyau Gaussien centré, de l'écart-type (σ) (annexe B).

e. Critère de convergence :

```

Début
  X = alpha * spf * Grad ;
  Si X = 0 sortir
  Sinon retourner au calcul de C1 et C2
  Fin si ;
Fin.

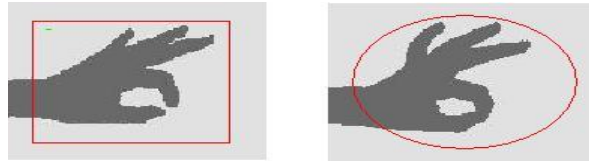
```

Pour mettre en évidence les qualités et les limites de l'approche que nous avons adoptée, nous avons choisi de partager notre travail en deux parties. La première consiste en son application à des images statiques. La deuxième concerne son adoption au suivi d'objets dans une séquence d'images, en temps réel.

III.3.1 Application à une image statique

Les étapes que nous avons suivies pour appliquer l'algorithme de segmentation à une image statique sont les suivantes :

- Ø Acquisition de l'image : l'image à segmenter est prise à l'aide d'une caméra, ou bien chargée à partir d'une bibliothèque d'images, et cela à l'aide de l'interface spéciale pour la segmentation statique (Annexe A).
- Ø Conversion au niveau de gris : si l'image acquise est représentée dans l'espace couleur RGB, elle sera convertie au niveau de gris. Pour ce faire, nous avons utilisé une commande Matlab dédiée pour cette transformation de l'image, où chaque pixel de l'image est représenté avec une valeur entre 0 et 255. En effet, il y a 256 niveaux de gris du fait que chaque valeur d'un pixel est codée sur un octet (8 bits).
- Ø Initialisation : Selon le type de la segmentation, nous avons mis au point plusieurs méthodes d'initialisation du contour (Figure III.3), à savoir : l'initialisation par rectangle automatique, rectangle manuel, cercle automatique, etc. Après avoir défini la forme géométrique de notre initialisation, nous avons effectué l'étape d'initialisation de la fonction Level-set (Etape (a) de l'algorithme).



a) Initialisation par rectangle. b) Initialisation par ellipse.

Figure III.3 Exemples d'initialisations.

- Ø Evolution : L'évolution se résume par une boucle représentant le processus itératif. Ce processus applique les étapes (a,b,c,d,e) pour produire une itération et boucle l'étape (e) vers l'étape (b) pour reproduire la même opération, et cela jusqu'à convergence du contour.
- Ø Critère d'arrêt : Peut être choisi comme le nombre d'itérations fixées au préalable, ou bien, d'une manière plus précise et dans le souci d'un gain de temps (Rappelant que notre objectif est le suivi en temps réel), l'annulation de SPF.

L'organigramme de figure III.4, illustre les étapes suivies pour la segmentation d'une image statique.

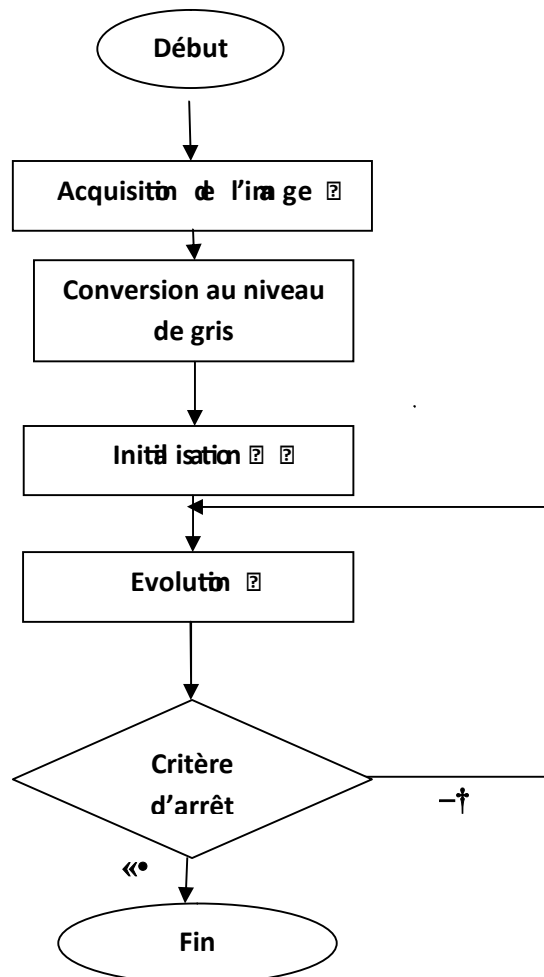


Figure III.4 Segmentation d'image statique.

▼ Exemples de segmentations d'images statiques :

Ici, nous représentons deux résultats qui mettent en évidence la propriété de segmentation locale et globale de l'approche que nous avons adoptée.



(a) Segmentation locale.

(b) Segmentation globale.

Figure 111.5 Segmentation locale et globale.

D'autres tests sur des images statiques ont été faits, nous les avons exposés dans le chapitre IV.

111.3.2 Application au suivi d'objet

Afin de réaliser notre application sur le suivi d'objet en mouvement, dans une séquence d'image, nous avons procédé par le choix d'une méthode de détection de l'objet en mouvement, puis l'application de l'algorithme de segmentation pour le suivi. Le processus de suivi est représenté par l'organigramme de la figure 111.6

Les différentes étapes du processus sont décrites comme suit :

- Ø Estimation du fond : Dans notre cas, le fond est considéré comme l'image en absence d'un objet quelconque. C'est une image prise à l'aide d'une caméra fixe (Annexe A).
- Ø Acquisition de la première image de la séquence : Cette image est prise à l'apparition de l'objet à suivre.
- Ø Détection de l'objet en mouvement : Etant donné que la caméra de l'expérience est fixe (fond fixe) et que le suivi se veut d'être une application en temps réel, nous avons choisi une méthode de détection de mouvement basée sur la soustraction des images successives de la séquence avec le fond (figure 111.7).

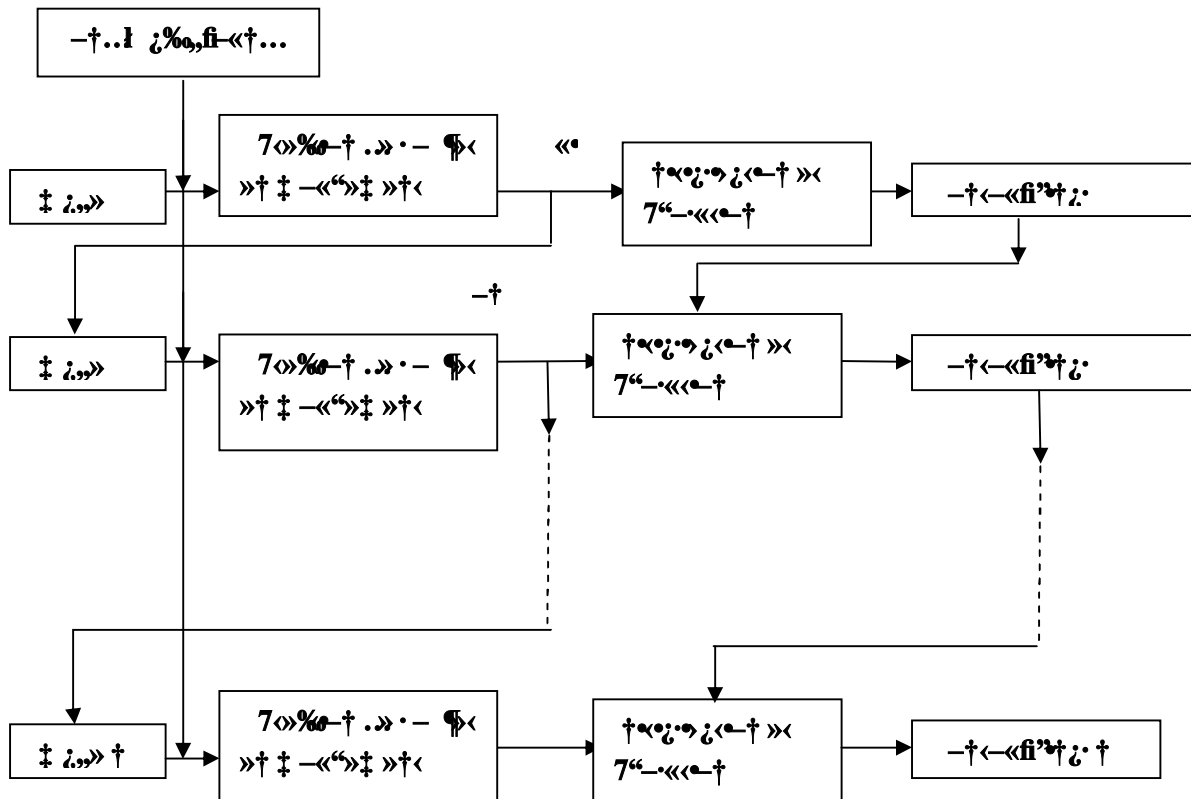


Figure III.6 Suivi d'objet

La méthode de détection de mouvement que nous avons adopté se base sur les étapes suivantes :

- * Soustraction : l'algorithme effectue une soustraction directe, pixel à pixel entre l'image de la séquence et le fond.
- * Traitement de l'image résultante de la soustraction : Afin de bien réussir le suivi, un traitement est effectué pour distinguer les différentes régions de l'image résultante. Pour cela, nous avons effectué à l'aide de quelques commandes Matlab :
 - Un ajustement : Augmentation du contraste de l'image, par le rehaussement des intensités des pixels.

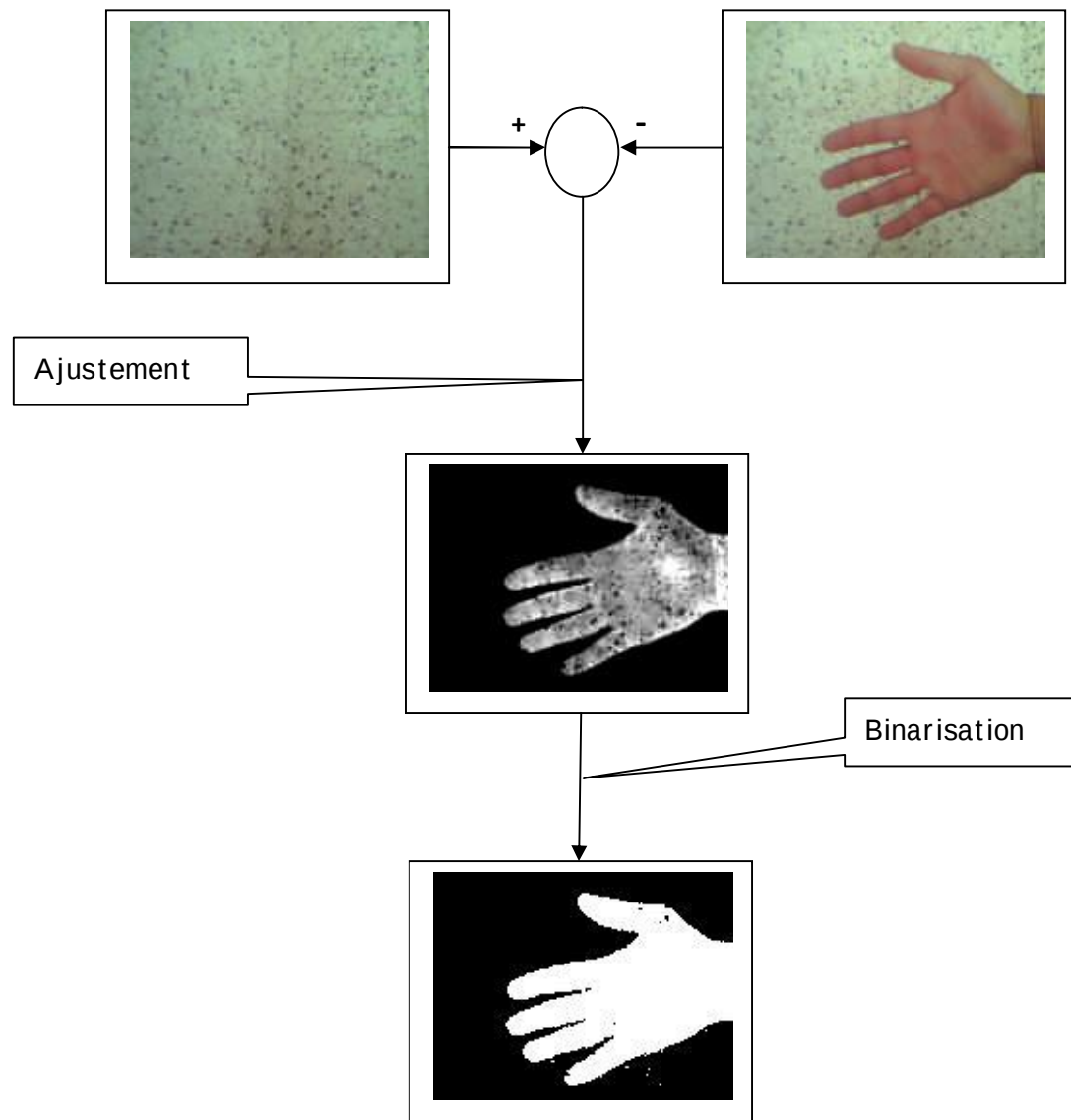


Figure III.7 *Soustraction d'images.*

→ Binarisation de l'image : Pour bien séparer les différentes régions de l'image, nous avons effectué une binarisation (couleur noire ou blanche) selon l'algorithme suivant :

```

Début
  Pour i = 1 jusqu'à n
    Pour j = 1 jusqu'à m
      Faire
        Si  $I(i, j) > \text{seuil}$ 
           $I(i, j) = 255$ ;
        Sinon
           $I(i, j) = 0$ ;
        Fin si ;
      Fin pour ;
    Fin pour ;
  Fin.
Avec, n : le nombre de colonnes de l'image.
      m : Le nombre de lignes de l'image.
      seuil : C'est une valeur à définir manuellement.
  
```

- Ø Initialisation du contour sur l'image résultante. L'initialisation est faite automatiquement par une forme géométrique donnée.
- Ø Evolution et convergence du contour. L'évolution est régie par l'application des étapes résumées par l'organigramme de la Figure III.2.
- Ø Définition du contour final comme contour initial du résultat de la soustraction de l'image suivante avec le fond (background).
- Ø Répéter les deux dernières instructions jusqu'à la fin de la séquence.

III.4 CONCLUSION

Dans ce chapitre, nous avons adopté un nouveau modèle de contour actif basé région pour la segmentation d'images, qui est implémenté avec une nouvelle initialisation de la fonction Level-set. Cette dernière, réduit, considérablement, le temps de calcul par rapport aux modèles Level-set classiques pour la rendre plus rapide, chose qui nous conviendra pour notre objectif de suivi en temps réel.

Ce modèle possède, aussi, la propriété de segmentation locale ou globale. Cette propriété, nous permettra de suivre un objet bien précis dans une séquence à plusieurs objets en mouvement.

Notons que l'application de ce modèle sur des images statiques donne de bons résultats. Reste à savoir, maintenant, ses résultats pour une application en suivi d'objets. C'est ce que nous allons aborder dans le chapitre qui suit.



Tests et Résultats



« If my theory of relativity is proven successful, Germany will claim me as a German, and France will declare that I am a citizen of the world. Should my theory prove untrue, France will say that I am a German, and Germany will declare that I am a Jew ».

A. Einstein

IV.1 Introduction

Ce chapitre est consacré aux tests de l'approche adoptée, qui est assez détaillée dans le chapitre III. Les tests sont, premièrement, effectués sur des images statiques et ceci en prenant en compte le type de segmentation locale ou globale, afin de mieux apprécier les performances de l'approche adoptée.

Des applications aux suivis d'objets, dans des séquences enregistrées, sont faites, suivis des applications en temps réel.

Nous avons acquis, comparé et jugé les différents résultats obtenus avec les différentes applications.

IV.2 Ressources Matérielles et Logicielle

Notre matériel (figure IV.1) se résume à ce qui suit :

- a. PC Portable : *Acer 5100 processeur AMD Mono-Core (un seul noyau) 1,7GHz. Une RAM de 1Go. Disque dur de 100G. Carte graphique ATI Radeon 128Mo. Webcam (Acer Orbicam) intégrée.*
- b. Caméra : *D-Tech USB (digital technologie). Modèle FM806. Résolution 1,3 méga pixel, max 1280x960. Formats vidéo RGB24 et I420.*

Notre application est développée sous Matlab : Version 7.8.0.347 (R2009a) ; 32bit Sous Windows.



Figure IV.1 Station de travail.

IV.3 Interface dédiée

Nous avons assemblé notre travail dans une interface principale (figure. IV.2), nous permettant, ainsi, un accès simple et rapide à la phase de suivi d'objet en mouvement, en passant par toute une configuration préliminaire du matériel et quelques paramètres que nous citerons dans la suite de ce chapitre.

L'accès aux différentes interfaces de configuration et de paramétrage se fait par une simple clique sur les boutons contenus dans le menu principal. Nous reviendrons sur cette interface dans le détail en annexe A.

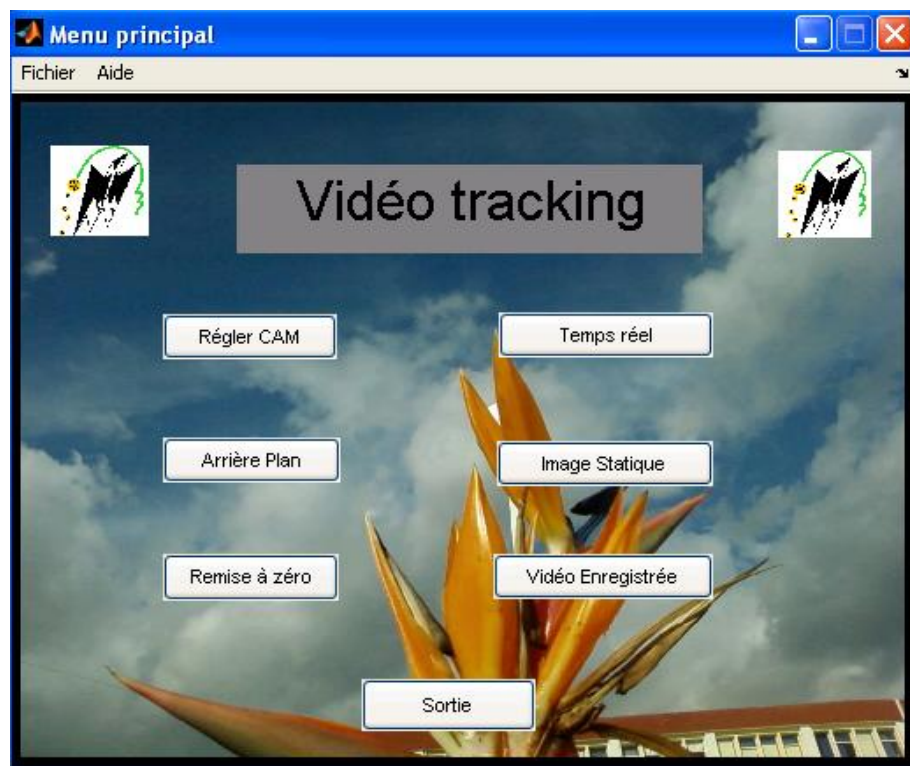


Figure IV.2 Menu principal de notre interface.

Avant de procéder, par exemple, au tracking en temps réel, il est nécessaire de faire une configuration initiale comme suit :

- On doit choisir le format d'acquisition en cliquant sur le bouton « Régler CAM », qui nous conduira vers l'interface de réglage de caméra (Annexe A). Dans notre cas, nous avons le choix entre I420 et RGB24 qui sont les formats supportés par notre matériel. RGB24 nous permet d'avoir une meilleure représentation des couleurs.

- Une fois que nous avons choisi le format, il est nécessaire, selon les performances de notre matériel, de choisir la résolution de l'image. Pour nos expériences, nous avons travaillé avec un format RGB24 d'une résolution de 160x120.
- Si la configuration matérielle a réussi, nous pouvons enregistrer les modifications pour qu'elles soient exploitées dans les autres interfaces.
- Nous procédons, par la suite, à la capture de l'image d'arrière plan, qui nous servira de référence pour le tracking, en cliquant sur le bouton « Arrière plan » qui nous mènera vers l'interface dédiée à cette tâche (Annexe A).
- Dés maintenant, nous pouvons procéder au tracking en temps réel via le bouton « Temps réel », qui nous conduira vers l'interface qui lui a été dédiée (Annexe A).

IV.4 Tests et résultats

Pour étudier et trouver des applications à notre modèle, nous sommes astreints de supposer différentes situations, de la plus simple à la plus complexe. Durant ces tests, nous tenant à mettre en évidence chaque avantage et surtout les limites de l'approche adoptée. Nous exposerons les résultats obtenus d'ici la fin de ce chapitre.

Vu que notre travail consiste en le suivi d'objets en temps réel, nous nous arrêterons dans chaque test pour aborder le temps de calcul de chaque résultat.

IV.4.1 Image statique

Nous avons choisi de procéder, en premier lieu, par des images statiques enregistrées sur disque. Pour mettre en évidence les avantages et les limites de l'approche adoptée, nous avons effectué des tests sur des images contenant des objets à segmenter plus ou moins complexes.

IV.4.1.1 Segmentation locale

La segmentation locale, dans une image, consiste à isoler un objet désiré dans l'image. Pour le modèle adopté, il faut que le contour soit initialisé tout près de l'objet que nous voulons segmenter. En voici quelques résultats obtenus :

a. Forme complexe :

L'objet segmenté dans la figure IV.3 est dit complexe, avec des bords non réguliers. Nous voyons bien, qu'avec une segmentation locale les autres objets sont ignorés et l'objet d'intérêt est très bien détecté, après 18 itérations et un temps d'exécution de 1,152secondes.

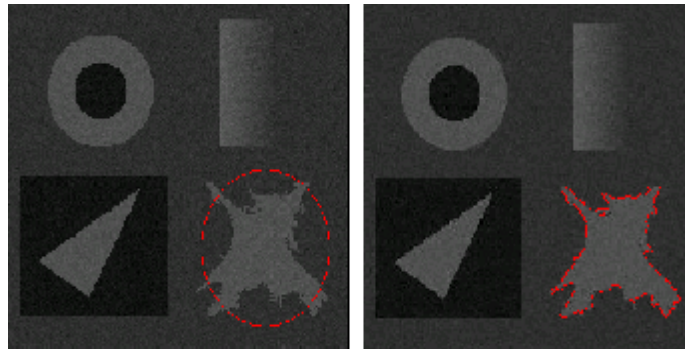


Figure IV.3 Initialisation cercle manuel, Segmentation locale, $\alpha = 25$.

b. Initialisation au bord de l'objet :

Le résultat de la figure IV.4, met en évidence la propriété de segmentation locale de la méthode, avec une initialisation proche de l'objet à détecter, que nous avons fait manuellement par un rectangle. Le contour a pu se converger en un temps total de 1,18 secondes en 23 itérations.

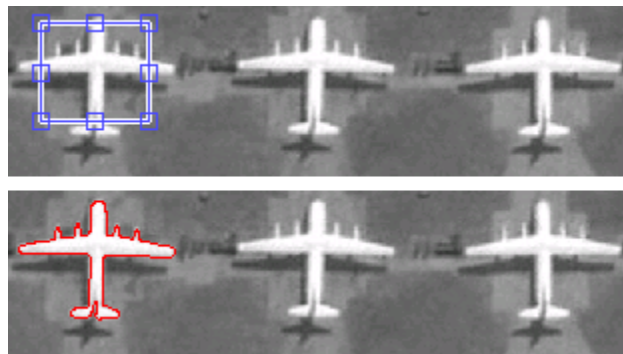


Figure IV.4 Initialisation rectangle manuel, Segmentation locale, $\alpha = 25$.

c. Objet sans bords :

L'expérience de la figure IV.5, met en évidence la faculté de la méthode pour la détection des objets ayant des bords non apparents. La convergence est atteinte au bout de 27 itérations en 1,635 secondes.

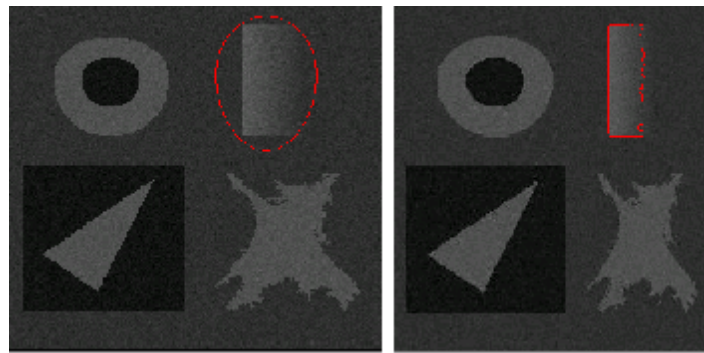


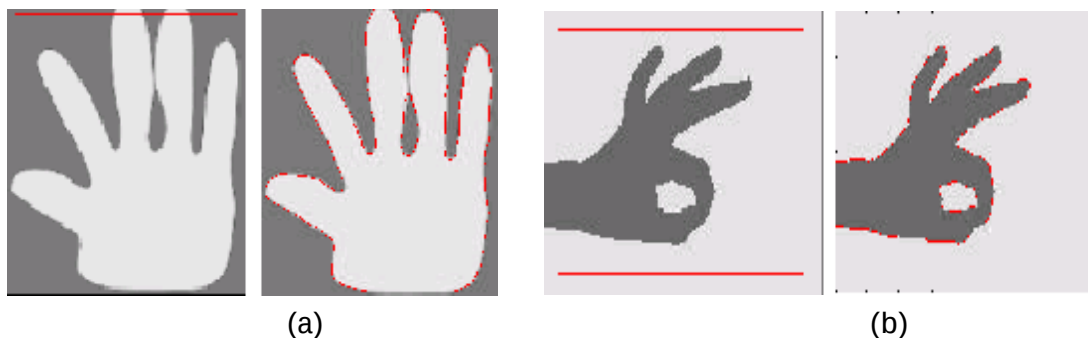
Figure IV.5 Initialisation ellipse manuelle, Segmentation locale, $\alpha = 25$

Remarque : il est à noter que dans les trois dernières expériences, lorsque nous avons utilisé la segmentation globale, la méthode s'est convergé aux bords ayant une forte différence du gradient, les autres bords de différence moins forte, sont considérés comme des minimums locaux.

IV.4.1.2 Segmentation globale

a. Objet concave :

Le modèle que nous avons adoptée a pu détecter la totalité des bords des deux mains de la figure IV.6. Pour la figure IV.6. (a), le contour a convergé après 49 itérations d'une durée totale de 0.73476 secondes, à raison de 0.014 secondes par itération. Une initialisation avec un cercle nous a permet l'obtention de performances meilleures : 21 itérations en 0.29933 secondes. Nous avons, aussi, constaté que l'augmentation du nombre de cercles diminue, à la fois, le temps de calcul et le nombre d'itérations.



Figures IV.6 Initialisation rectangle automatique, Segmentation globale, $\alpha = 25$.

b. Segmentation avec initialisation quelconque :

Le résultat de la figure IV.7, montre que pour une segmentation globale, l'emplacement du contour initial n'est pas assez important. Cette dernière propriété nous rappelle le modèle C-V. Il faut noter que le changement de topologie est géré automatiquement. Nous constatons, aussi, que la convergence avec une telle initialisation prend plus de temps : 3.4318 secondes pour 86 itérations.

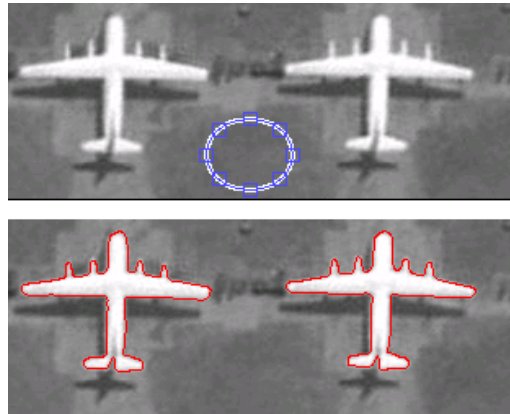


Figure IV.7 *Initialisation ellipse manuelle, Segmentation globale, $\alpha = 25$.*

c. Détection des bords internes et externes :

Nous constatons dans la figure IV.8, la capacité du modèle, que nous avons adopté, à détecter les bords internes et externes d'un objet donné. L'utilisation d'une seule fonction Level-set a fait que notre algorithme ne puisse détecter qu'une seule différence d'intensité, soit deux régions dans l'image.

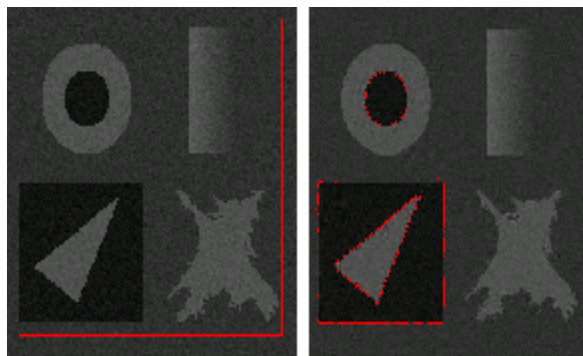


Figure IV.8 *Initialisation rectangle automatique, Segmentation globale, $\alpha = 25$.*

d. Image bruitée :

La méthode adoptée présente une certaine robustesse par rapport aux bruits. La détection de la zone étroite de l'objet met en évidence la précision du contour, qui peut être modifiée par le paramètre *Sigma* du filtre Gaussien, comme nous pouvons le voir dans la figure IV.9. L'objet est détecté après 0,26 secondes, au bout de 29 itérations.

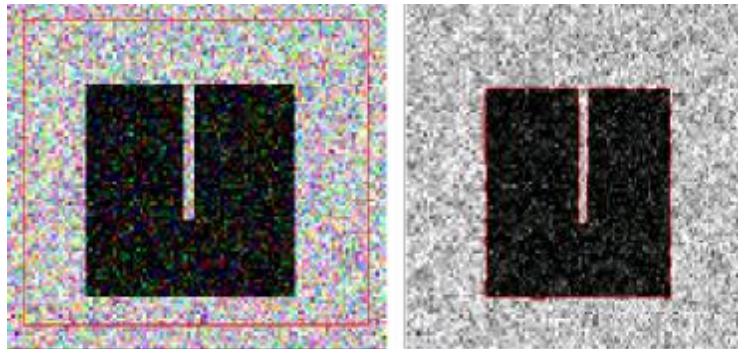


Figure IV.9 Initialisation rectangle automatique, Segmentation globale, $\alpha = 25$. $\sigma = 5$.

IV.4.2 Application au vidéo tracking

Notre objectif principal était de réaliser un tracking en temps réel. Pour cela, nous détaillons les différentes étapes suivantes :

IV.4.2.1 Etude de l'image à segmenter

Comme première partie, nous avons choisi d'étudier l'image obtenue par soustraction, pour trouver la valeur de « α » optimale correspondante. Pour cela, nous avons procédé comme suit:

- Ø Choix expérimental de « α » : L'image résultante a été expérimentée dans l'interface dédiée au traitement des images statiques (figure IV.10). En faisant varier « α », nous avons relevé à chaque fois les performances conséquentes. Dans notre interface, ces tests sont exportés, automatiquement, dans un fichier Excel. Ce fichier nous a servi, par la suite, pour étudier la variation des performances en fonction de « α ».

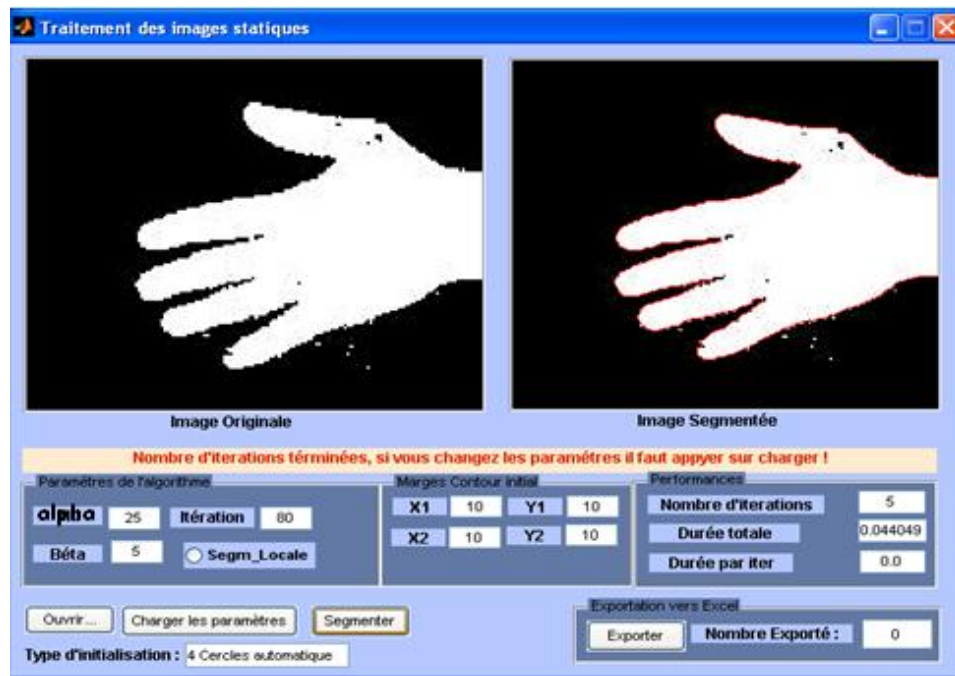


Figure IV.10 Interface d'expérimentation.

Cette étape nous a permis l'obtention des deux courbes des figures IV.11 (courbe 1) et IV.12 (courbe 2).

NB : Dans les figures IV.11, IV.12 et IV.14, l'axe horizontal représente le numéro de l'expérience.

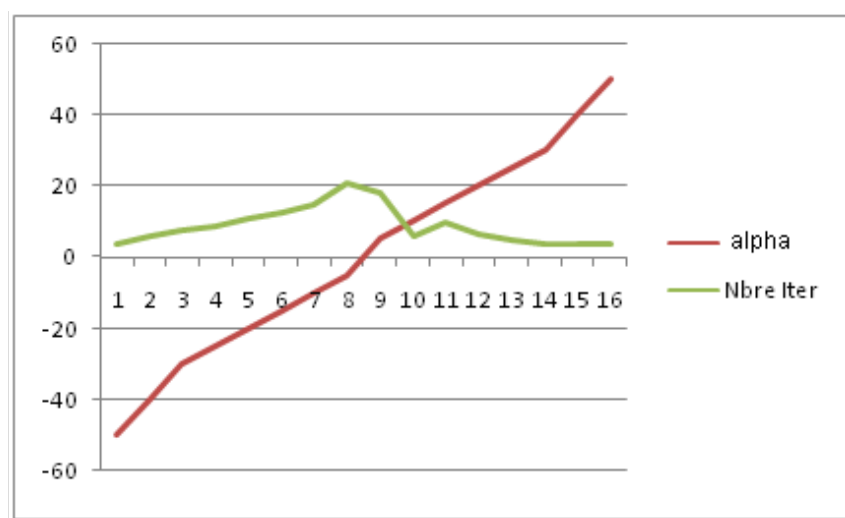


Figure IV.11 Alpha et nombre d'itérations correspondant.

Discussion

En valeur absolue, lorsque « α » est supérieur à 50, le nombre d'itérations nécessaires pour la convergence est égal à 4. Par contre, lorsqu'il est inférieur, le nombre d'itérations augmente avec la

décroissance de « α » jusqu'à atteindre 21 itérations pour un « α » égal à 5.

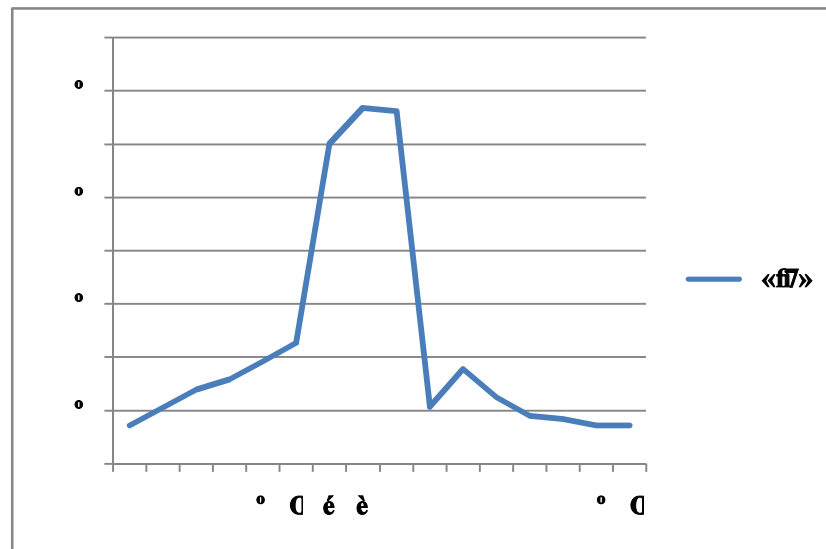


Figure IV.12 Temps de convergence.

Discussion

Comme la durée de la convergence est directement liée au nombre d'itérations, nous voyons bien que la courbe de la durée (figure IV.12) et celle du nombre d'itérations, évoluent de la même manière. Pour $|\alpha| > 50$, on a enregistré les performances maximale avec un temps de convergence de 0.035secondes. Par contre, pour $\alpha = 8$, notre algorithme a mis 10 fois plus de temps pour converger ce qui est équivalent à 0.334 secondes.

- Ø Test sur une autre image : Pour confirmer l'influence de la valeur du paramètre « α » sur l'évolution du contour, nous avons effectué une autre expérience sur l'image donnée par la Figure IV.13.



Figure IV.13 Image de test.

Pour ce test, nous avons relevé l'évolution de la durée de convergence (Figure IV.14), en faisant varier α de -50 à +50.

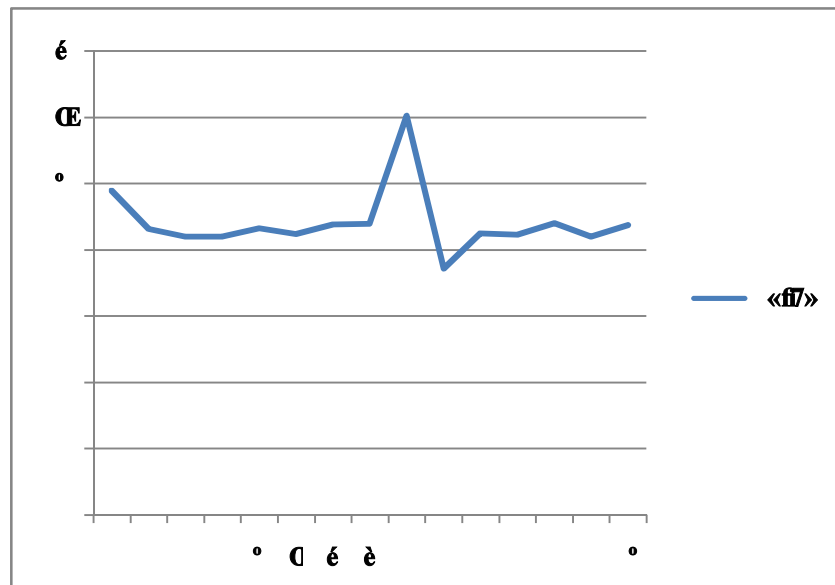


Figure IV.14 Evolution de la durée.

Discussion

De la courbe ci-dessus, nous déduisons, que le meilleur « α », qui assure la propagation la plus rapide du contour, c'est-à-dire le meilleur temps de convergence, est obtenu à l'abscisse 11 qui correspond à la valeur 15 de « α ».

Conclusion

Après avoir effectué les deux tests précédents sur deux images différentes, nous constatons que la valeur de « α » influe sur la vitesse de propagation du contour, à des seuils différents selon le milieu de propagation propre à chaque image.

IV.4.2.2 Tracking Vidéo

a. Application sur une vidéo enregistrée

La vidéo utilisée, pour ce test, est nommée « vipmen.avi » de Matlab.

- Application directe : Nous avons directement appliqué notre algorithme tel que nous l'avons annoncé dans la chapitre précédent. Le résultat est illustré dans la Figure IV.15.

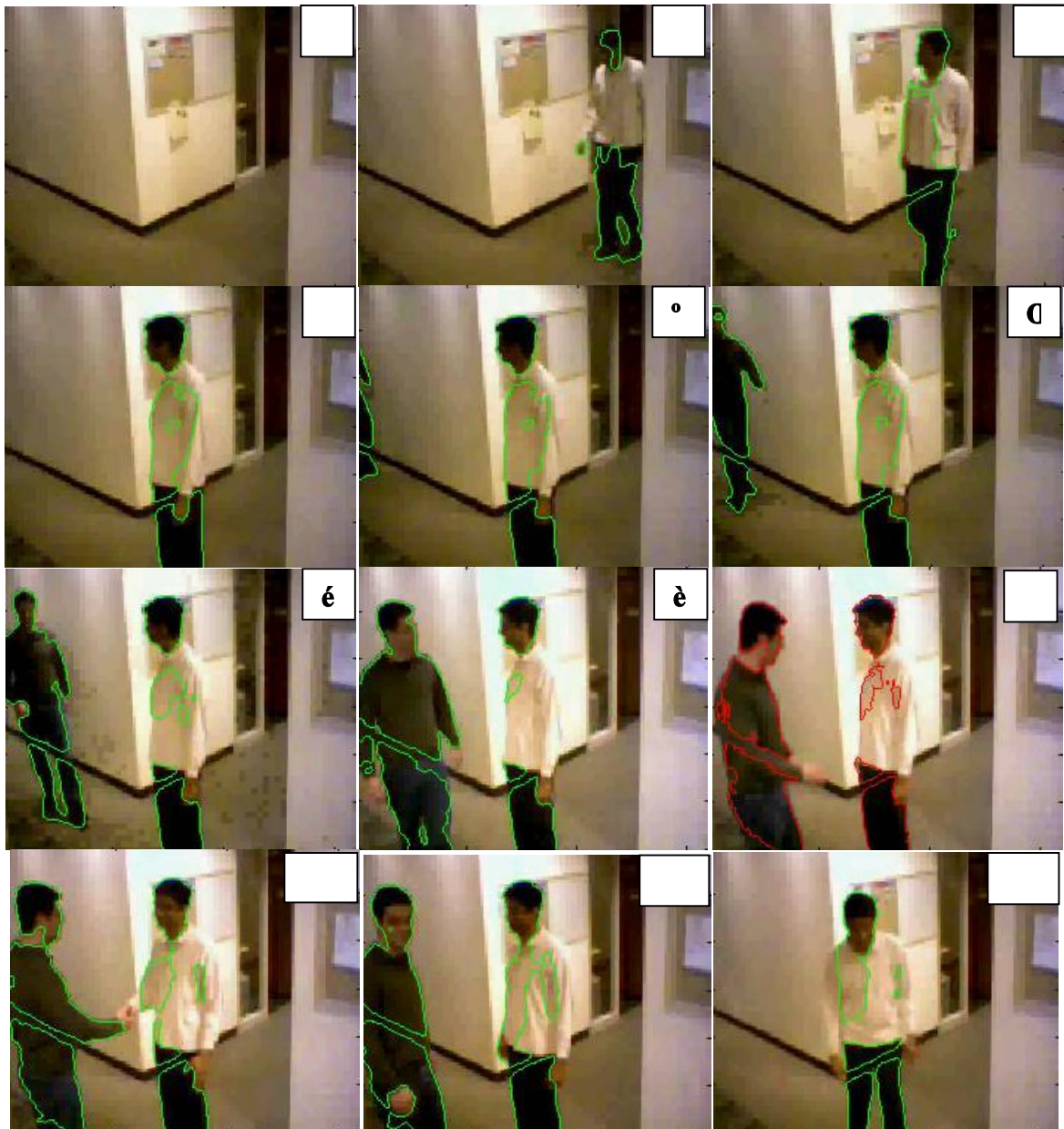


Figure IV.15 Application directe sur vidéo.

Discussion

L'image (1) est utilisée comme arrière plan du fait qu'elle ne contient pas d'objet en mouvement. Pour (2, 3 et 4), l'apparition de la première personne a été détectée et suivi par le contour en vert. L'arrivée de la deuxième personne (en 5, 6, 7 et 8) aussi a été détectée et suivi avec succès. Cependant, on constate des défauts de segmentation qui s'expliquent par :

- o Changement d'intensité : il est dû à l'entrée des deux personnes dans la scène. Cela peut se voir en 5, 8 et 12. Les deux silhouettes sont partiellement segmentées.

- o Fond complexe : La chemise blanche n'a pas été bien détectée du fait qu'elle possède les mêmes intensités que le fond. Aussi les pantalons noirs avec la faïence noire.
- o Mouvement brusque : illustré par l'image 9, où le contour rouge signifie que la convergence n'est pas encore atteinte.
- Correction des défauts : Un simple traitement additionnel sur l'image résultante de la soustraction (correction par morphologie mathématique (Dilatation) [21]), nous a permis d'obtenir de résultats meilleurs (Figure IV.16).



Figure IV.16 Correction des défauts avec l'application d'une dilatation avec élément structurant à 6-connexité.

Discussion

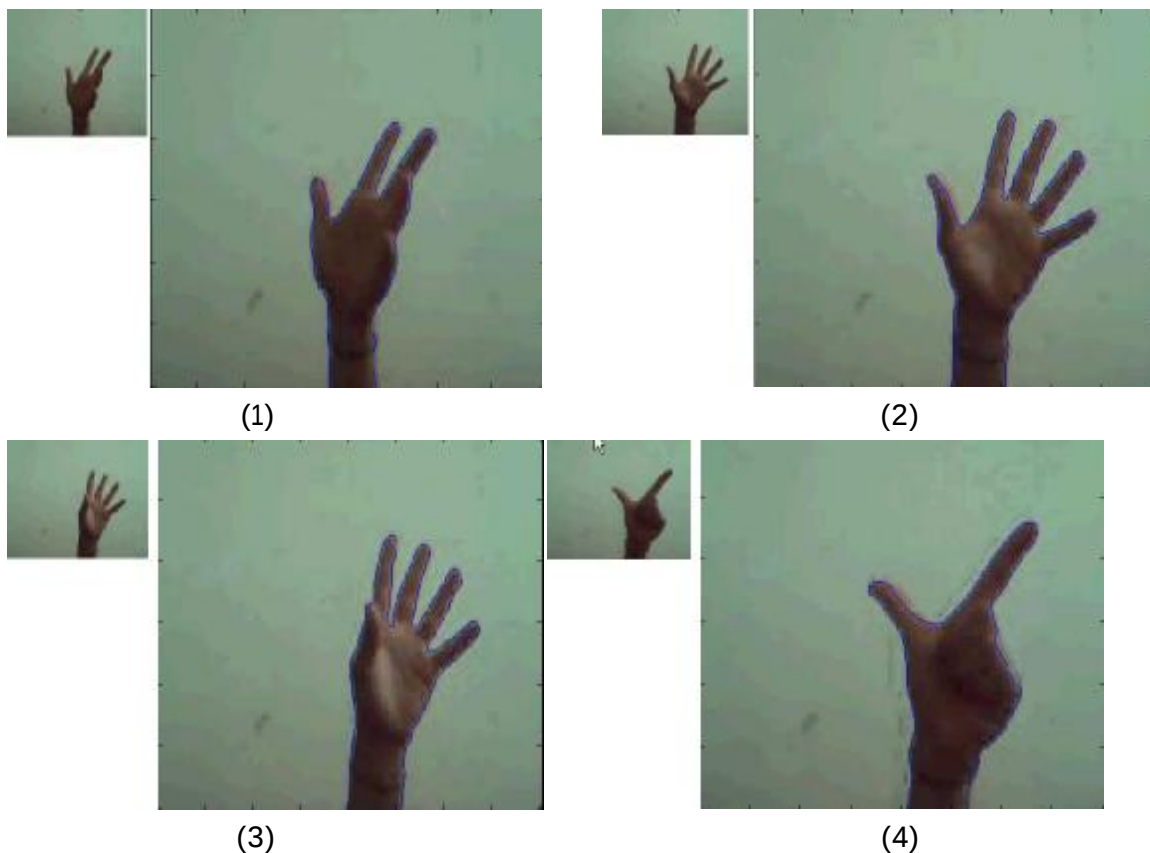
Nous constatons une nette amélioration par rapport aux deux premiers défauts de segmentation des personnages du test précédent. Cependant, nous constatons aussi la persistance du défaut lié au mouvement brusque.

Conclusion

Nous avons testé notre algorithme sur une vidéo enregistrée, jugée assez compliquée (Changement d'intensité, fond complexe et mouvement brusque), ce qui a permis la mise en évidence des limites de cette méthode, à savoir, la vitesse du mouvement, les changements importants d'intensité et le fond complexe. Ce dernier peut être amélioré par l'introduction d'outils de correction comme la morphologie mathématique [21].

b. Tracking d'une main en temps réel

Dans ce qui va suivre, nous présenterions les résultats obtenus lors des expériences faites pour le suivi du mouvement des doigts de la main en temps réel.



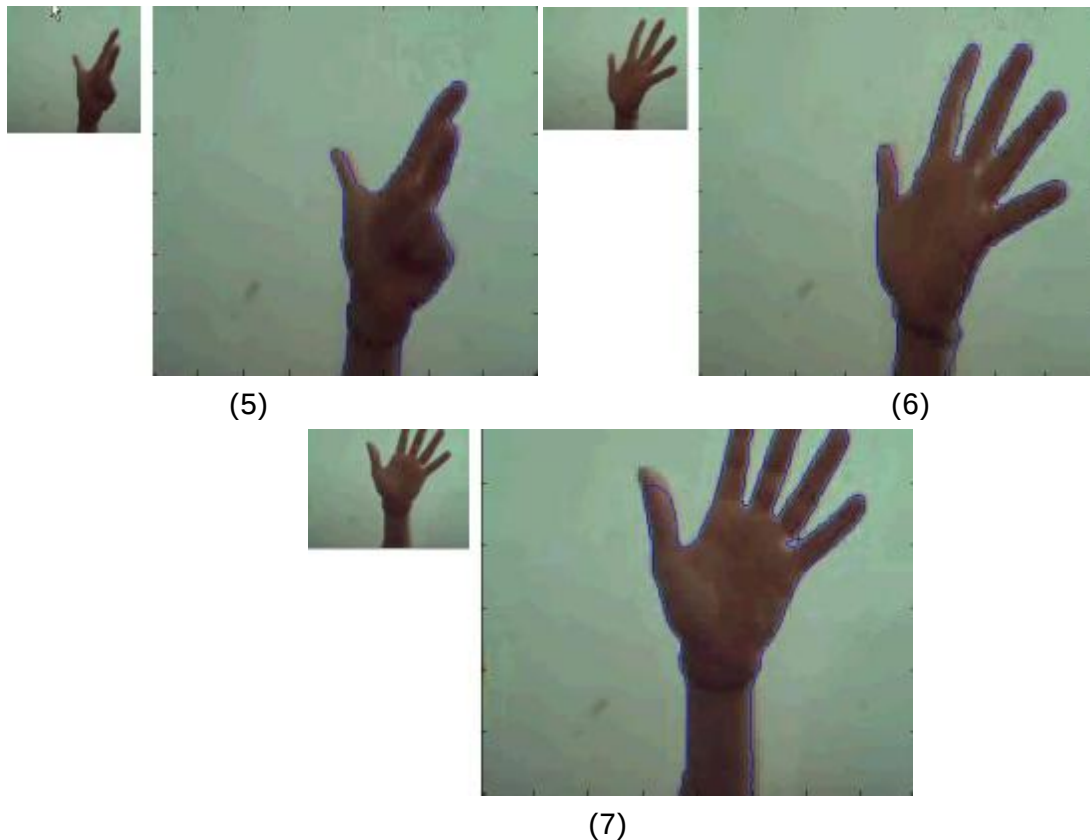


Figure IV.17 Séquence tracking en temps réel de la main.

Discussion

Les images de (1) à (6) de la Figure IV.15, sont des échantillons pris d'une séquence acquise en temps réel. Le choix de « α » pour cette expérience est basé sur l'étude faite précédemment.

La segmentation est assez robuste pour gérer les changements d'intensité au sein de l'objet, comme nous pouvons le constater dans les images (1), (2) et (3). Pour l'image (7) le pouce est mal détecté, ce qui est dû au changement important d'intensité à son extrémité. Les doigts sont rapidement segmentés dès leurs apparitions. Au cours de l'évolution du contour, nous avons remarqué que le contour prend une autre couleur (rouge) quand il y a un mouvement rapide, ce qui signifie que la convergence n'est pas atteinte.

c) Tracking d'objet rigide spécifique avec critère de taille en temps réel

Nous avons vu dans le chapitre précédent que pour réaliser une segmentation locale, le contour doit être initialisé tout près de l'objet désiré, ce qui n'est pas facile à réaliser dans une vidéo. L'expérience suivante s'en est affranchi en utilisant une information additionnelle à priori sur l'objet à suivre.

En effet, une modification a été apportée au niveau de notre algorithme, juste après l'étape de soustraction de chaque image acquise avec l'image de référence. Le but de cette modification étant d'affecter une couleur pour chaque objet en mouvement, compter le nombre d'objets en mouvement et calculer leurs surfaces. Nous choisissons un objet selon sa taille. D'ailleurs, pour évaluer la modification apportée, nous avons pu faire une petite expérience.

Description de l'expérience :

Nous avons conçu à l'aide : d'un carton d'emballage, des pièces métalliques d'un vieux transformateur de récupération sous forme de « E » et de « I » et d'un aimant, un environnement de travail adéquat.

En effet, à l'aide de l'aimant placé de l'autre côté du carton, nous faisons bouger les deux pièces métalliques qui se trouvent de l'autre côté, selon la trajectoire voulue.

Les résultats obtenus sont illustrés par les figures IV.18 et IV.19.

- Suivi du plus grand objet :

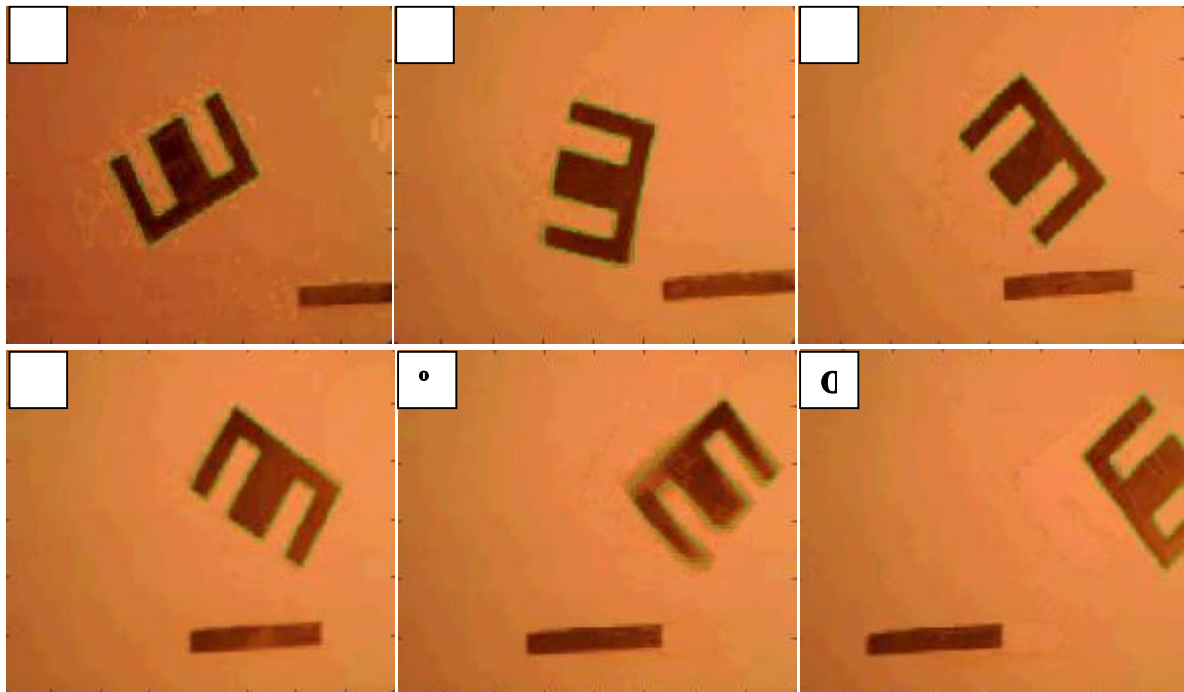


Figure IV.18 Séquence de suivi du plus grand objet.

- Suivi du plus petit objet :

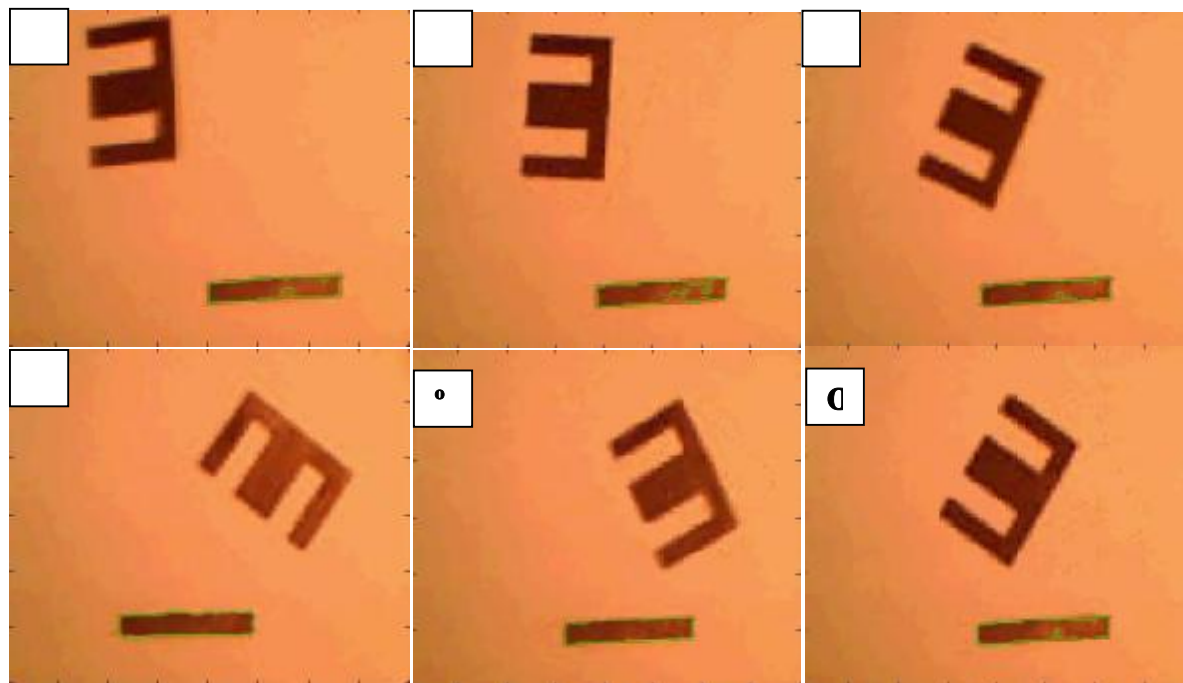


Figure IV.19 Séquence de suivi du plus petit objet.

Discussions :

Dans la Figure IV.19 nous avons pu détecter le plus petit objet. Nous constatons bien que seul le « I » a été détecté du fait que sa surface est moins importante que celle de « E ».

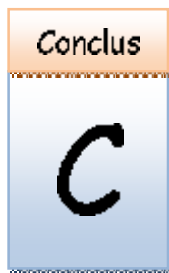
Remarque : La réussite de cette expérience exige la rigidité de tous les objets présents dans la scène.

IV.5 Conclusion

Dans le but de prendre connaissance de tous les avantages de l'approche que nous avons adoptée, nous avons effectué une diversité de tests assez riche, où, nous avons commencé par un ensemble de tests sur différentes images statiques, qui nous démontrent clairement ces principales qualités. A travers ces tests, nous avons aussi pu mettre en évidence la capacité de segmentation locale et globale, mais aussi l'indépendance du résultat par rapport à l'emplacement du contour initial.

En second lieu, nous l'avons appliqué au suivi de deux personnes en mouvement, dans une séquence d'image enregistrée. Dans cette application, l'algorithme a essentiellement pu gérer automatiquement les changements de topologie, et ignorer les variations de l'intensité dans l'image. Sauf qu'il a failli, quant à la gestion de l'occlusion entre les objets en mouvement et le fond, ce qui est considérablement amélioré grâce à un simple traitement additionnel que nous avons rajouté (La morphologie mathématique).

Enfin, pour aller au bout des performances de notre algorithme, nous avons réalisé les applications de suivi d'objet en mouvement en temps réel. Relativement au matériel que nous avons utilisé, nous estimons avoir eu de très bons résultats, sauf que, malgré tout les efforts fournis pour mieux accélérer l'algorithme, il reste incapable de suivre les objets ayant un mouvement à vitesse relativement élevée.



Conclusion générale

« La logique des passions renverse l'ordre traditionnel du raisonnement et place la conclusion avant les prémisses »

Albert camus.

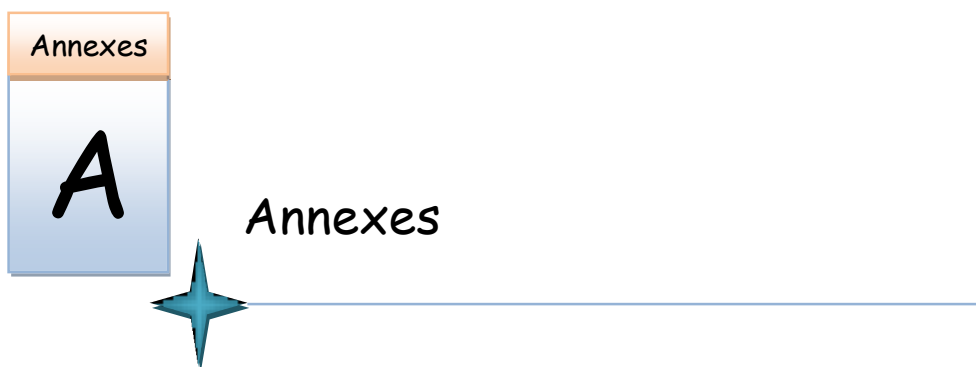
Le principal objectif de notre travail était d'appliquer un modèle de contour actif pour le suivi d'objets en temps réel. Le modèle choisi est implémenté avec les ensembles de niveaux (*level-set*), et combine les avantages d'une méthode de segmentation basée région (C-V), et une autre basée contour (CAG). L'apport principal de l'approche adoptée réside dans le temps de calcul, qui est considérablement réduit grâce à la nouvelle considération de l'initialisation de la fonction *level-set*, et les restrictions apportées à la fonctionnelle qu'on est sensé minimiser pour la convergence.

Les résultats obtenus sont jugés intéressants, relativement, au matériel d'expérimentation que nous avons utilisé.

Notons cependant, que la localisation des objets en mouvement dans la scène observée est contrainte de plusieurs paramètres tels que la vitesse des déplacements, la qualité des images de la séquence (variations d'éclairage, fond non uniforme, etc.).

Ainsi, afin de remédier à ces limites, et élargir le champ d'application de cette méthode, des perspectives sont envisageables, telles que, l'estimation du fond de la séquence qui pourrait être utile pour le suivi d'objets avec une caméra mobile, l'utilisation de plusieurs fonctions Level-set afin de pouvoir détecter plus d'une seule différence d'intensité dans l'image (segmentation de plusieurs objets différents).

Nous espérons que le travail que nous avons présenté, sera jugé utile et intéressant et qu'il sera exploité à bon escient et amélioré par les promotions à venir.



A.1 Menu principal :

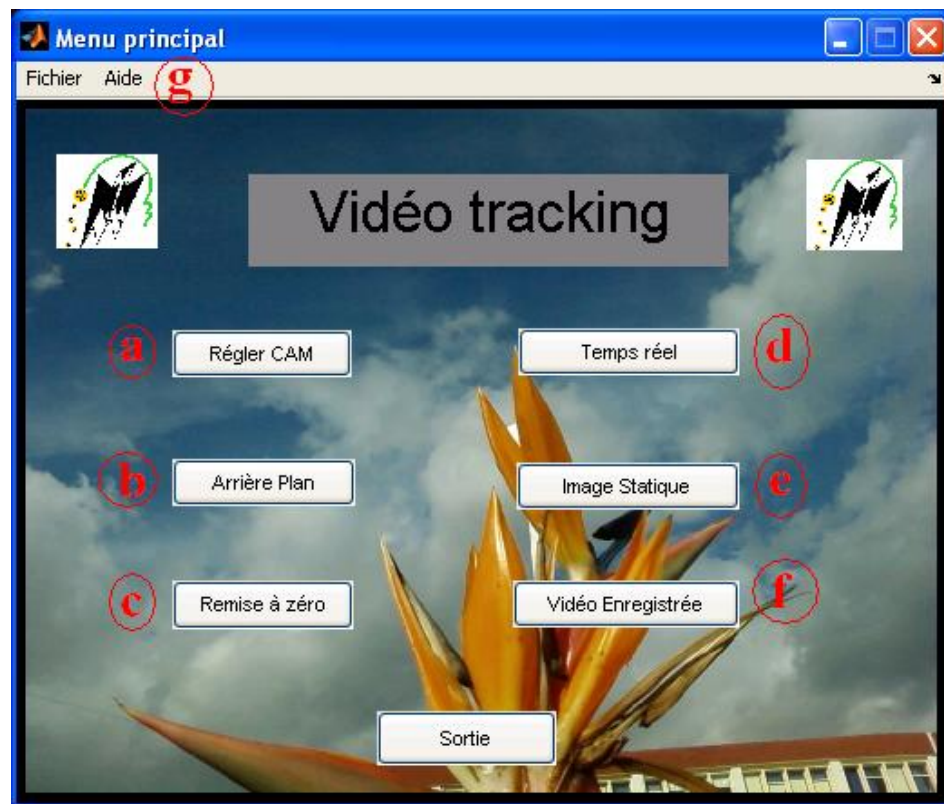
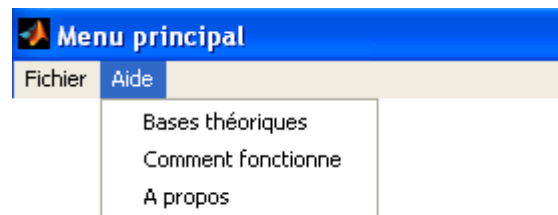
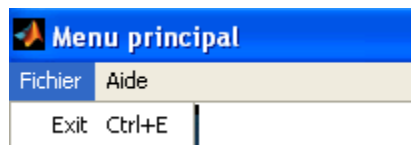


Figure A.1 « Menu principal ».

Ergonomique dans son genre. C'est la fenêtre d'accueil principale, qui nous donne accès aux différentes étapes nécessaires de notre projet. Les boutons 1, 2, 4, 5, et 6 nous permettent un accès à d'autres interfaces. Les fonctionnalités du menu principal sont résumées dans le détail dans ce qui suit :

- 1) Régler CAM : Configuration de la caméra utilisée.
- 2) Arrière plan : Une fois qu'on a terminé la configuration de la caméra, on peut s'en servir de cette étape pour capturer l'image d'arrière plan.
- 3) Remise à zéro : Permet d'effacer tous les fichiers qui sont créés à chaque utilisation de notre logiciel. Ces fichiers se résument à des images temporaires, des variables stockées en extension MAT et des fichiers Excel.
- 4) Temps réel : Pour entrer dans l'interface de suivi d'objets en temps réel.
- 5) Image Statique : Pour faire des tests sur des images statiques, que nous avons enregistrées sur disque dur, et étudier ainsi les performances de notre algorithme sur un type d'images tests.

- 6) Vidéo Enregistrée : Nous donne les mêmes fonctionnalités que 4, mais sur des vidéos enregistrées, c'est-à-dire que nous avons sur le disque dur.
- 7) Sortie : Ce bouton nous permet, comme son nom l'indique, de sortir de la fenêtre actuelle.
- 8) Fichier et Aide : c'est la barre de menu qui nous donne un accès à ce qui suit :



Figures A.2 « Défilement Fichier, Défilement Aide ».

- Fichier : permet la sortie du logiciel.
- Aide > Bases Théoriques : Ouvrir un document des rappels théoriques pour les techniques utilisées dans notre logiciel.
- Aide > Comment fonctionne : Accéder à des vidéo tutoriaux sur l'utilisation des différentes interfaces de notre logiciel.
- Aide > A propos : Présentation du projet.

A.2 Régler CAM :

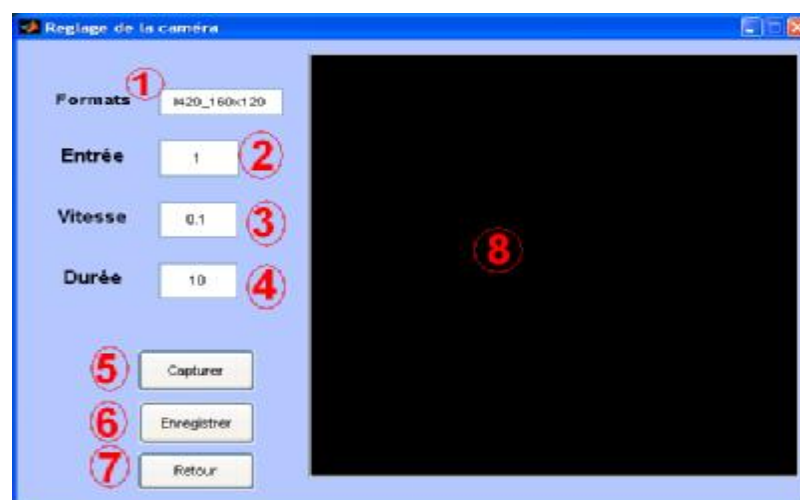


Figure A.3 « Interface réglage de la caméra ».

Cette interface est dédiée au réglage de la caméra. Elle est par défaut configurées par rapport au matériel utilisé que nous avons pris le soin de mentionner au début du chapitre « tests et résultats ». Elle comprend les zones : 1, 2, 3, et 4 où nous pouvons introduire, via le clavier ou par le bouton droit de la souris, quand il s'agit des valeurs les plus souvent utilisées. Pour avoir des informations sur notre matériel, nous pouvons aller à la ligne de commande MATLAB et taper 'imaqtool' suivi de la touche Entrée. Les différents éléments sont présentés dans ce qui suit :

- 1) **Formats** : Permet l'introduction du format de représentation des couleurs lors de l'acquisition, ainsi que la résolution. Pour le format, il est directement lié au type de caméra utilisée. Dans notre cas, une caméra supporte seulement le format I420 avec gamme de résolutions ; une autre supporte les deux formats qui sont : I420 et RGB, avec aussi différentes résolutions. Une clique avec le bouton droit sur la zone de texte devant 'Format' nous donne accès au menu contextuel suivant :

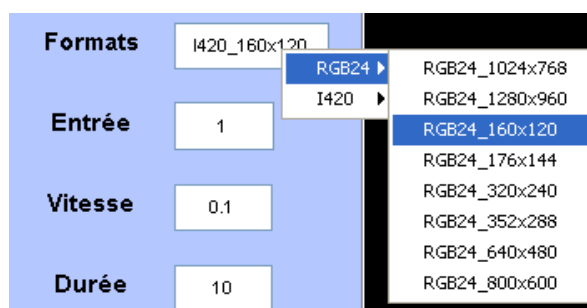


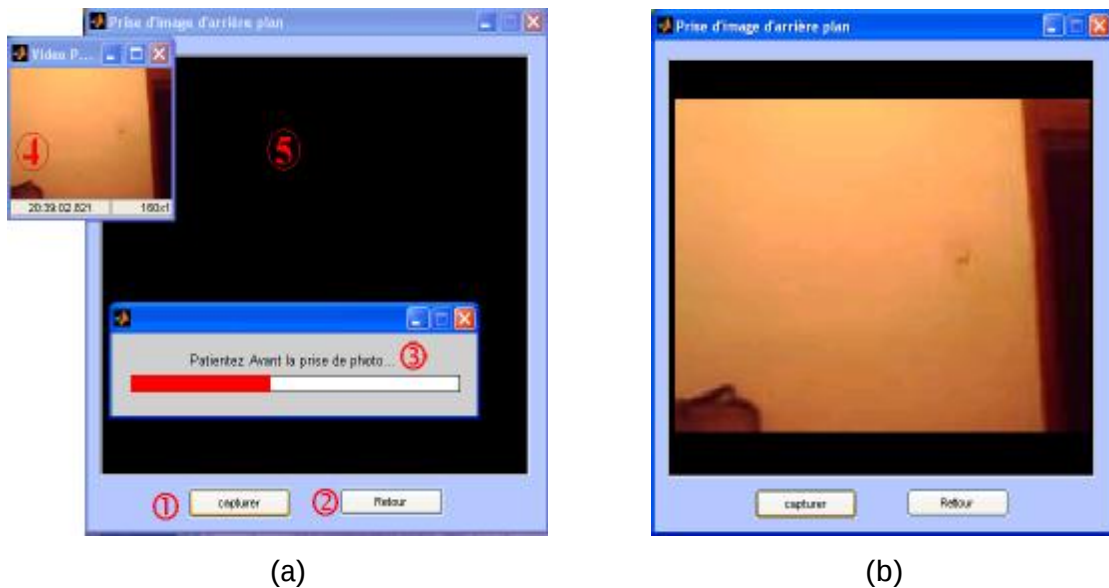
Figure A.4 « Exemple de Menu contextuel du bouton droit ».

Cette option du bouton droit ne sera pas réexpliquée pour les autres zones de textes 2,3 et 4.

- 2) **Entrée** : Dans le cas d'utilisation de plusieurs caméras, elle correspond à l'index de la caméra qu'on veut utilisée. Par défaut si on ne dispose que d'une seule caméra sa valeur est à 1. Si on dispose de plusieurs caméras on peut faire plusieurs acquisitions à la fois dans les limites de nos ressources matérielles, ce qui nous donne des idées en *perspective*.
- 3) **Vitesse** : C'est le contrôle de la fréquence de défilement des images dans l'écran de visualisation. Pour rappel, une vidéo est un défilement d'une séquence d'image à une fréquence donnée.

- 4) Durée : C'est pour limiter la durée du test de capture avec une configuration donnée.
- 5) Capturer : Pour lancer la capture et voir si la configuration est satisfaisante.
- 6) Enregistrer : Une fois qu'on a terminé la configuration. On peut enregistrer les paramètres de configuration pour un usage dans les autres interfaces de notre logiciel.
- 7) Retour : Permet le retour au menu principal.

A.3 Background :



Figures A.5 « Prise d'image de référence ».

- 1) Capturer : Pour prendre une image d'un arrière plan conditionné.
- 2) Retour : Pour revenir au menu principal.
- 3) Barre d'attente : Il est très important de donner un temps à la caméra pour qu'elle se stabilise lors de son démarrage ; ce qui justifie l'usage de cette barre d'attente d'environ 4 secondes (Figure A.5. (a)). Lorsque l'image se stabilise, on aura l'image capturée dans l'écran de visualisation 5 (Figure A.5. (b)).
- 4) Preview : Fenêtre qui permet la prévisualisation de l'acquisition en temps réel.
- 5) Ecran de visualisation : Affichage de l'image acquise.

A.5 Tracking :

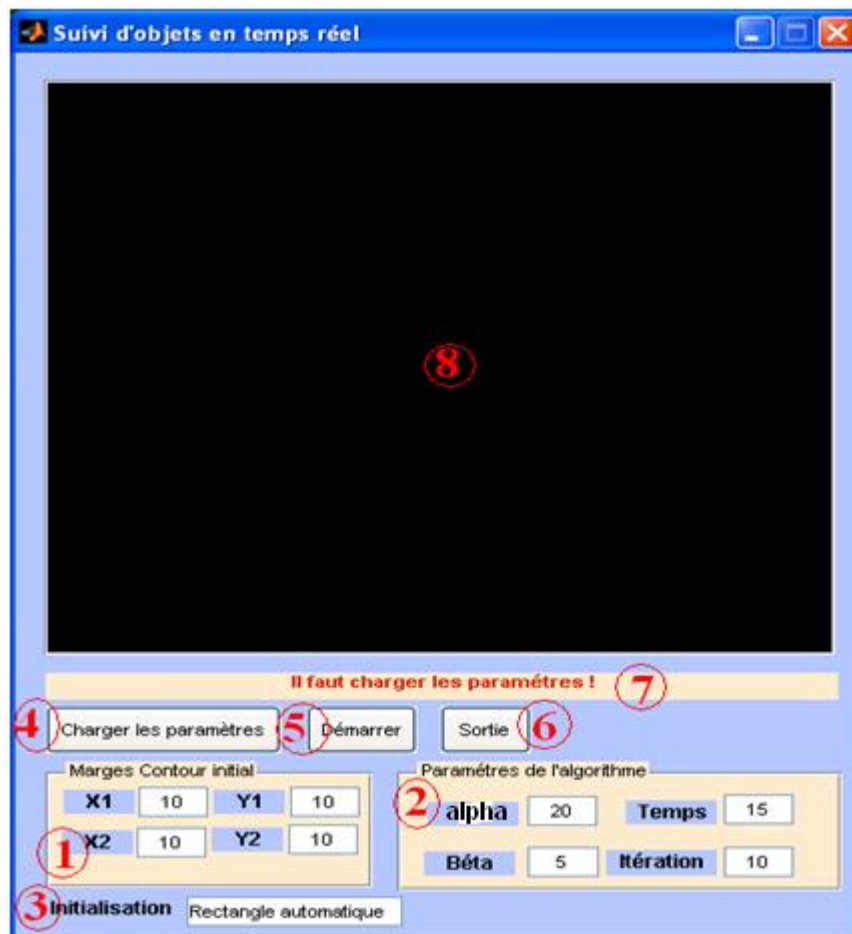


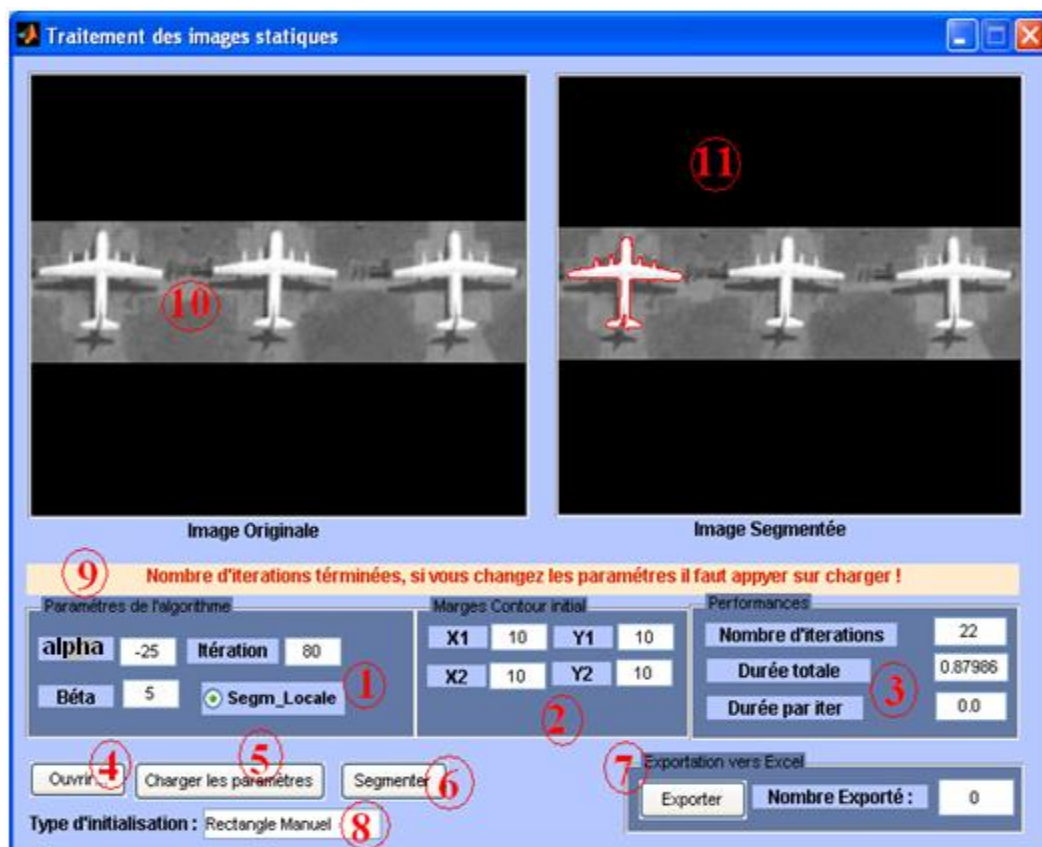
Figure A.6 « Interface de suivi d'objet en temps réel ».

Après avoir terminé la configuration du matériel et capturé l'image de référence qui est l'arrière plan, on peut passer à l'interface « Tracking » qui est destiné au suivi d'objet en temps réel qui se présente comme suit :

- 1) Marges contour initial : X1, X2, Y1 et Y2 représente les marges du contour initial dans le cas où l'initialisation se fait par un rectangle. C'est, en d'autres termes, le nombre de pixels entre les limites de l'image acquise et le rectangle d'initialisation.
- 2) Paramètres de l'algorithme :
 - Alpha : Paramètre très important qui varie selon l'image utilisée et selon la segmentation locale ou globale.
 - Temps : Durée fixée pour le Tracking.
 - Béta : Paramètre de la gaussienne.
 - Itération : Pour limiter le nombre d'itération pour chaque image, si on veut l'utiliser comme critère d'arrêt.

- 3) Initialisation : Pour définir le type du contour initial (*Rectangle automatique, Rectangle manuel, Cercle automatique, 4 Cercles automatiques, 16 Cercles automatiques, Ellipse manuel*).
- 4) Charger le paramètres : Une fois qu'on terminé l'introduction des paramètres expliqués dans 1, 2 et 3, ce bouton nous servira pour les charger en mémoire. Une étape indispensable pour le bon fonctionnement du logiciel avec les paramètres en cours.
- 5) Démarrer : Pour lancer le Tracking en temps réel.
- 6) Sortie : Pour revenir au menu principal.
- 7) Barre d'état : Nous permet de voir l'état d'avancement du logiciel. Nous conseil sur la succession logique des étapes principales lors de son utilisation.
- 8) Ecran de visualisation : Visualisation du Tracking.

A.6 Image Statique :



Pour expérimenter notre algorithme sur différentes images statiques, nous avons mis au point l'interface donnée par la Figure A.7.

- 1) Segmentation Locale : Check box nous permettant de sélectionner la segmentation locale ou globale. Lorsqu'il est coché notre algorithme procède pour une segmentation locale. Dans l'autre cas, s'il n'est pas coché, l'algorithme procède pour une segmentation globale. Il faut aussi tenir compte du signe de alpha.
- 2) Marges contours initial : Expliqué dans le paragraphe A.5.1. de l'annexe.
- 3) Performances:
 - Ø Nombre d'itérations : Nous affiche à la fin de la segmentation le nombre d'itérations nécessaires pour une convergence finale.
 - Ø Durée Totale : Affiche la durée totale de la segmentation. Une donnée très importante du fait que le critère temps est très important pour notre travail qui s'oriente vers une application en temps réel.
 - Ø Durée Par itération : durée pour une seule itération.
- 4) Ouvrir: Nous permet l'insertion de l'image à segmenter.
- 5) Charger les paramètres: Pour charger les différents paramètres se trouvant dans les différents panneaux
- 6) Segmenter: Lancer la segmentation. Ce bouton ne doit pas être sollicité avant d'avoir chargé les paramètres en cours sinon, notre algorithme utilisera les paramètres précédents.
- 7) Exportation vers Excel: Pour une image donnée, ce panneau nous permet l'exportation de tout les paramètres vers un fichier Excel pour un usage ultérieur. Cette fonction nous a permis au chapitre IV de tracer, expérimentalement, les performances de notre algorithme en fonction de alpha.
- 8) Type d'initialisation: L'initialisation peut se faire par différentes cas : Un ou plusieurs cercles automatiques, un rectangle automatique ou manuel, Une ellipse manuelle.
- 9) Barre d'état: Afficher l'état de notre interface. Des messages textes nous permettent de voir si une opération donnée est terminée ou non.
- 10) Ecran 1: Visualisation de l'image insérée.
- 11) Ecran 2: Résultat de la segmentation.

A.7 Vidéo Enregistrée :

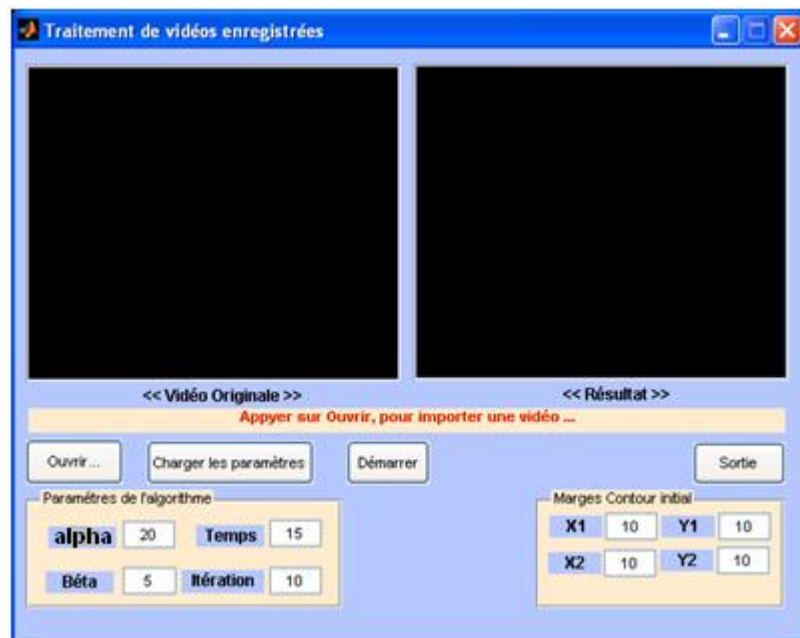


Figure A.e « Interface de traitement des vidéos enregistrées ».

A.8 à propos :



B.1 Notions sur le filtrage :

Le filtrage représente une opération cruciale en traitement d'images. En effet, il permet de manipuler les données d'une image numérisée afin d'améliorer sa qualité. Le terme qualité peut prendre plusieurs sens, à savoir :

- ü **Qualité visuelle** : chercher à avoir un meilleur aspect de l'image au niveau de la vision
- ü **Qualité du signal** : atténuation des distorsions liées aux techniques de numérisation.

Il revient donc à lier le principe du filtrage à la modification de la valeur des pixels de l'image. En effet, il s'agit de créer une nouvelle image en se servant de la valeur des pixels de l'image d'origine comme référence.

Deux types de filtrages peuvent être cités :

- ü **Filtrage global** : Le calcul de la valeur des pixels de la nouvelle image doit tenir compte de la valeur de tous les pixels de l'image d'origine.
- ü **Filtrage local** : La nouvelle valeur de chaque pixel de l'image filtrée est calculée en tenant en compte seulement un voisinage du pixel correspondant de l'image d'origine. Ce voisinage est souvent choisi comme étant carré et symétrique, sous forme de matrice et de taille impaire.

En introduisant la notion de linéarité, il y a lieu de citer les filtrages linéaire et non linéaire. On s'intéressera dans ce qui suit au filtrage local linéaire. (Exemple du filtre gaussien).

■ **Filtrage local non linéaire** : Le filtre, ne pouvant pas être exprimé par une combinaison linéaire est dit non linéaire. Les filtres linéaires ne sont point facile à mettre en œuvre mais ils sont très efficaces et ils donnent de meilleurs résultats que ceux obtenus par l'application des filtres linéaires.

■ **Filtrage local linéaire** : le filtrage local est dit linéaire si la nouvelle valeur de chaque pixel de l'image filtrée est donnée comme combinaison linéaire des valeurs des pixels du voisinage.

$$N(x, y) = \sum_{(i,j)} A_{(i,j)} * P_{(x+i,y+j)} \quad \text{B.1}$$

Avec $A_{i,j}$ est une valeur spécifique du filtre (entière ou réelle), et i, j variant entre $-h$ et $+h$, la demi taille du voisinage (pour 3×3 on a $h=1$, pour 5×5 on a $h=2$).

Ceci est une représentation d'une convolution d'une image avec un noyau (3×3) :

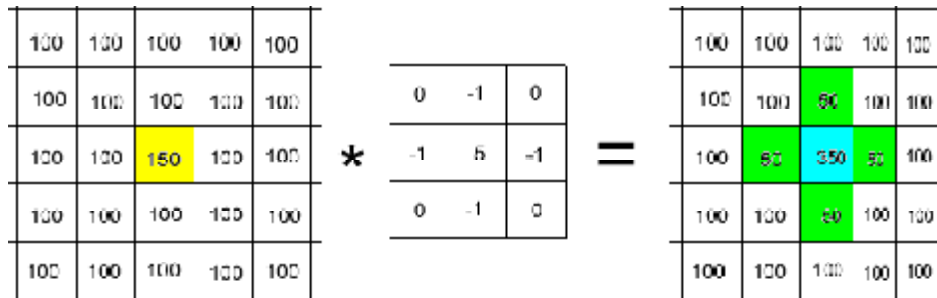


Figure B.1 Exemple de convolution d'une image avec un noyau.

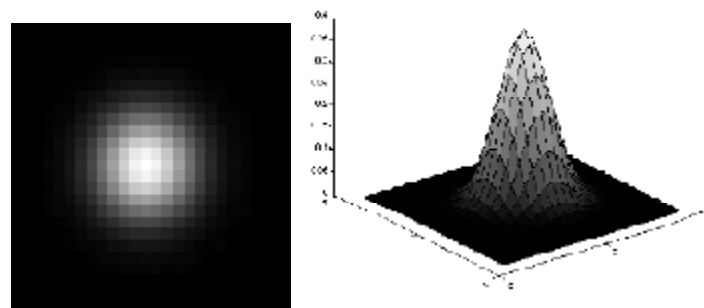
▼ Le filtre Gaussien :

Il représente un cas de filtrage local linéaire agissant comme un filtre à moyenne. En effet, son principe est de calculer la moyenne pondérée du voisinage de chaque pixel dans l'image originale. Une moyenne pondérée car les pixels du voisinage qui sont proches du pixel central ont un poids plus fort que ceux qui s'y situent loin. Les filtres gaussiens adoucissent les détails de l'image et élimine les bruits granuleux. Ces filtres ont pour fonction (B.2):

$$G(X, Y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{X^2+Y^2}{2\sigma}\right) \quad \text{B.2}$$

Le principe de ce filtre est de multiplier l'image par un masque (noyau) Gaussien de dimensions $[i] \times [j]$.

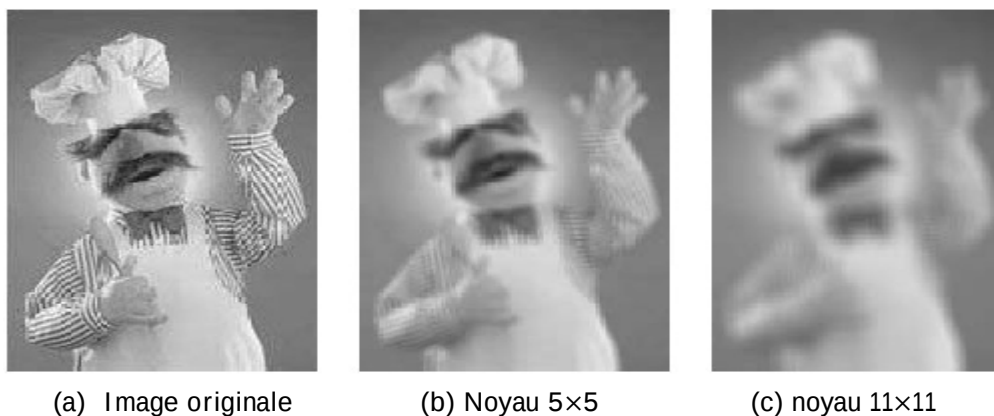
Exemple :



$$\frac{1}{98} \times \begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 6 & 8 & 6 & 2 \\ 3 & 8 & 10 & 8 & 3 \\ 2 & 6 & 8 & 6 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix}$$

Figure B.2 Exemple d'un noyau gaussien de dimension 5x5.

En général, un filtre gaussien avec un sigma < 1 est utilisé pour réduire le bruit. Si sigma est > 1 alors ceci est dans le but de créer une image qui sera utilisée pour créer « un masque flou » personnalisé, appliqué à de différents types images. Il est à noter que quand la valeur de sigma augmente, le noyau sera de dimension plus grande, le flou appliqué à l'image sera grand. (Figure B.3)



(a) Image originale

(b) Noyau 5x5

(c) noyau 11x11

Figure B.3 Résultat d'application du filtre gaussien avec différents noyaux

B.2 Le gradient :

Soit une image I . Le gradient de l'image, symbolisé par l'opérateur ∇ , se calcule par des dérivées partielles comme suit :

$$\nabla(I) = \begin{cases} I_x = \frac{\partial I}{\partial x} \\ I_y = \frac{\partial I}{\partial y} \end{cases} \quad \text{B. III}$$

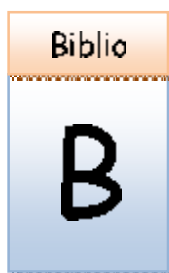
Ce qui veut dire que nous pouvons calculer le gradient en chaque point de l'image I .

Le gradient est caractérisé par son module et sa direction, qui sont calculés par les formules suivantes :

➤ Le module : $\nabla(I) = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$ B. IV

➤ L'argument : $\theta = \text{Arg}(\nabla I) = \tan^{-1}\left(\frac{\frac{\partial I}{\partial x}}{\frac{\partial I}{\partial y}}\right)$ B. V

Dans une image, le gradient détecte les changements brusques d'intensités d'une région à une autre, c'est-à-dire une forte variation du niveau de gris, alors nous pouvons dire qu'il y a présence de contour.



Bibliographie



« Il faut rendre à César ce qui appartient à César »

Citation latine.

- [1] C.J. Veenman, M.J.T. Reinders et E. Backer , « Resolving Motion correspondence for Densely Moving Points », *IEEE Transactions on Pattern*, 2001.
- [2] P. Wren et AP. Pentland, «A survey of computer vision-based human motion capture», *IMAGE'COM 96, Bordeaux, France, May 1996*.
- [3] A. Elgammal, D. Harwood, et *al.* « Non-parametric Model for Background Subtraction », *Computer Vision Laboratory University of Maryland, College Park, MD 20742, USA, 2000*.
- [4] J. Rittscher, J. Kato, S. Joga et A. Blake, « A probabilistic background model for tracking », *Computer Vision—ECCV 2000, 2000 – Springer*.
- [5] L. Li et M. leung , «Foreground object detection in changing background based on color co-occurrence statistics », *IEEE Trans. Image processing*, 11(2):105–112, February 2002.
- [6] Oliver, Rosario, «Coupled Hidden Markov Models for Modeling Interacting Processes », *Proceedings of IEEE Intl. Conference on Intelligent 2000*.
- [8] S. Birchfield, « Elliptical Head Tracking Using Intensity Gradients and Color Histograms », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Santa Barbara, California, pages 232-237, June 1998*.
- [9] H. Schweitzer, J. Bell et F. Wu, «Very fast template matching », *Computer Vision—ECCV 2002, 2006*.
- [10] H. Tao, Harpreet , S. Sawhney et R. Kumar, "Object tracking with Bayesian estimation of dynamic layer representations," *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, vol. 24, no. 1, pp. 75-89, 2002.
- [11] A. Yilmaz, X. Li et M. Shah, «Object tracking a survey» ,*ACM New York, NY, USA, 2006*.
- [12] N. Paragios et R. Deriche. « Geodesic Active Regions and Level Set Methods for Supervised Texture Segmentation». *International Journal of Computer Vision*, pp. 223-247, 2002.
- [13] Frédéric PRECIOUSO, «Contours actifs paramétriques pour la segmentation d'images et vidéos », thèse de doctorat, *Université de Nice - Sophia Antipolis*, septembre 2004.
- [14] V. Caselles, K. Ron et S. Guillermo, « Geodesic active Contour.», *International Journal of Computer Vision* 22(1), 61-79, 1997.

- [15] J-B. Stéphanie, « Modèle de contours actifs basés régions pour la segmentation d'images et de vidéo », thèse de doctorat, *Laboratoire I3S Sophia Antipolis*, Janvier 2003.
- [16] L. Cohen, E. Bardinet et N. Ayache, « Surface reconstruction using active contour models ». Dans SPIE Conference on Geometric Methods in Computer Vision, San Diego, CA, 1993.
- [17] R. Ronfard, «Region-based strategies for active contour models. », *International Journal of Computer Vision*, 13(2) :229_251, 1994.
- [18] Tony F. Chan et Luminita A. Vese, « Active contours without edges », *Image Processing, IEEE Transactions on*, FEBRUARY 2001.
- [19] Y. Shi et W-C. Karl, « a fast implementation of the level set method without solving partial differential equations », *Boston University, ECE-2005-02*.
- [20] K. Zhang, L. Zhang, H. Song et W. Zhou, « Active contours with selective local or global segmentation: A new formulation and level set method », *Image and Vision Computing* 28 (2010) 668–676.
- [21] « Traitement d'images binaires par Morphologie Mathématique », <http://webdav-noauth.unit-c.fr/files/perso/ythomas/cours/Videocommunications1%20/mm/E4mb/> .
dernière visite : le 27/06/2010.