الجمهورية الجزائرية الديمقراطية الشعبية

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DR. YAHIA FARES DE MEDEA

FACULTE DES SCIENCES ET DE LA TECHNOLOGIE

Département du Génie Électrique et de l'Informatique



Projet de Fin d'Etudes

Présenté pour l'obtention du Diplôme
de Master en Génie Électrique

Spécialité : *Systèmes de Télécommunications*

THEME

Segmentation d'image par la méthode de contour actif

Proposé et dirigé par,

M. BAILICHE Mohamed Amine.

Préparé et présenté par,

M. BENGUERMIT Mohamed.

M. SOFTA Billal.

Devant le jury composé de

Pré :M. MAAZOUZ

Exa : M. LAIDI

Dédicaces

A ma mère et mon père pour leur appui, leur amour, leur compréhension, leur patience et pour tout ce qu'ils ont fait pour moi.

A ma sœur Nadjat, a mes frères et sœurs.

A toute ma famille.

A mes amis, Messoud SOFTA, Mohamed B, Elhadi M, Amin D, Hakim A.

A mes collègues de l'université de Medea.

A tous ceux qui m'aiment, et ceux que j'aime.

A surtout mon frère Bouzid.

Je dédie ce mémoire.

SOFTA Billal



DEDICACE

Je dédie ce modeste travail

A mes très chers parents, pour leurs patience, leurs soutien et Leurs confiance ; mon père qui s'est sacrifié afin que rien n'entrave le déroulement de mes études, ma mère qui m'encourager, et me soutenir dans les moments difficiles ; qui a sa su m'entourer de tout son amour et son affectation pour que je ne puisse manquer de rien.

A mes très chères sœurs et frères.

A toute ma famille.

A mon binôme ***Mr. SOFTA Billal*** *et toute sa famille.*

A tous mes enseignants sans exception.

A mes collègues de l'université de Médéa.

A tous les étudiants de l'université de Médéa on particulier les étudiants de télécommunication.

A tous ceux qui me sont chères, en particulier :

Stembouli M , Stembouli A ,safar remali R ,Mekki al hadi , debdabi mohamed amine, aouina hakim, oulha mohamed, bouhla ismail, Bndjebesse A ,Albane S ,saoudi A.

BENGUERMIT Mohamed

Remerciements

*Nous tenons tout d'abord à remercier **ALLAH** le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce Modeste travail.*

*En second lieu, nous tenons à remercier tout le corps enseignant qui a contribué à notre formation, plus particulièrement notre encadreur **M. BAILICHE Mohamed Amine** pour son aide et les précieux conseils qu'il a bien voulu nous donner.*

Nos remerciements s'étendent également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail.

ملخص:

في هذا العمل أردنا انجاز كاشف الحواف الذي يعرف باسم كاشف الحافة النشيط، حيث أن الهدف من ذلك هو برمجة خوارزمية باستعمال برنامج MATLAB للكشف عن الاجسام في الصور.

للوصول لهذ الغرض اتبعنا منهجية والتي تتمثل في: حساب طاقة نقاط الكاشف ومقارنتها مع نقاط أخرى في الصورة وتحويل موضعها. فمن الضروري استعمال طريقة فعالة لكشف الأجزاء المهمة التي تحتويها الصورة حيث نستنتج دقة النتيجة من جهة ومدة تشغيل طويلة نوعا ما من جهة أخرى.

الكلمات المفتاحية:

تجزئة، حافة فعالة، طاقة الحافة، العتبة، الرسم البياني.

Résumé :

Dans ce travail nous avons essayé d'implémenter un détecteur de contour connu sous le nom contour actif. Le but est programmer un algorithme en utilisent le programme MATLAB pour détecter le contour des objets dans les images.

Pour atteindre efficacement cet objectif nous avons suivis une démarche qui consiste à : calcul de l'énergie du contour, comparaison et déplacement des pixels du contour. Il était en question de définir une méthode de segmentation en contour d'image capable de mettre en évidence cet algorithme qui est caractérisé par la précision des résultats et en effet un long temps d'exécution.

Mots clés :

Segmentation, contour actif, énergie du contour, seuillage, histogramme.

Abstract :

In this work we tried to implement an edge detector known as the active snake, the goal is to program an algorithm using the MATLAB to detect the edges of objects in images.

To effectively achieve this goal we followed an approach that is :calculating contour energy, comparison and displacement of edge pixels , it was a question of defining a method of image segmentation can do this work , this last characterized by the precision of the results and long time execution.

Keywords :

Segmentation, actif snack, contour energy, thresholding, histogram.

Table des matières

LISTE DES FIGURES ..... **I**

Introduction general **1**

ChapitreI : Generalite sur le traitement d'image

1. Introduction **2**

2. Notions élémentaires **3**

2.1. Image 3
2.2. Image binaire 3
2.3. Les images à niveau de gris 3
2.4. Les images en couleurs 4
2.5. Pixel 4
2.6. Point de contour 5
2.7. Point d'intérêt 5
2.8. Partition 5
2.9. Connectabilité 5
2.10. Connexite 5
2.11. Histogramme 6
2.12. Contraste 7
2.13. Luminance 7

3. Formats de fichier image **7**

3.1. Format TIF (.tif) Tagged Image File 7
3.2. Format JPEG (.jpg) Joint Photographie Experts Groupe 8
3.3. Format MPEG (.mpg) Moving Picture Experts Group 8
3.4. Format GIF (.gif) Graphiques Interchange Format 8
3.5. Format BMP (.bmp) bitmap Picture 8

4. Les etapes du traitement d'image **8**

5. Le filtrage **10**

5.1. Filtrage linéaire 10
5.2. Filtrage non linéaire 11
5.3. Filtrage adaptatif 11
5.4. Comparaison des méthodes de filtrage 12

6. Seuillage d'une image **12**

6.1. Seuillage global 13
6.2. Seuillage adaptatif ou local 13
6.3. Seuillage par hystérésis 14
6.4. Comparaison des techniques de seuillage 14

7. Système de traitement d'image **14**

8. Quelques notes importantes **15**

9. Conclusion **15**

ChapitreII : La segmentation d'image

1. Introduction 16

2. Objectifs de la segmentation 16

3. définition de la segmentation 16

4. Approche région (extraction de région) 18

4.1. Partition topologique 18
4.2. Prédicats d'homogénéité 18

5. Approche contour 20

5.1. Notion de contour 20
5.2. Les étapes d'extraction de contour 21
5.2.a La détection des points de contours 21
5.2.b Affinage de points de contours 21
5.2.c Suivi et la prolongation (fermeture des contours) 21
5.2.d Chainage des contours 21
5.3. Détecteur de contour 22
5.3.1 Méthode par masque 22
A. Opérateur de roberts 22
B. Opérateur de sobel 22
C. L'opérateur de Prewitt 24
D. L'opérateur de Kirsch 25
5.3.2 Méthode par optimisation basées sur modélisation des conteurs 26
A Le détecteur de contour de Canny 26
5.3.3 Méthode différentielles 27
A. Opérateurs basée sur la dérivée seconde 27
B. Opérateur du LAPLACIEN 27

6. Approche mixte (dualite contour/region) 28

7. Comparaison des methodes de segmentation 29

8. Conclusion 30

ChapitreIII : Segmentation d'image par la méthode de contour actif

1. Introduction 31

2. Principe et fonctionnement intuitif de la methode des contours actifs 31

2.1 Énergie globale 32
2.2 Énergie interne 33
2.3 Énergie externe 34
2.4 Utilisation des deux énergies 35
2.5 Modifications du contour 35
2.6 Faiblesses des snakes, cas de non convergence totale 36

3. Mise en œuvre 37

3.1 Discrétisation 37
3.2 Minimisation de l'énergie par algorithme dynamique 39

4. Conclusion 41

Chapitre IV : Mise en marche et discussion de résultat

1. Introduction 42

2. Conception 42

3. Résultats et interprétation 44

3.1 *Résultats* 44

3.2 *Discussion* 48

4. Conclusion 49

Conclusion général 50

Bibliographie 51

Annexe

LISTE DES FIGURES

Figure I.1*: Les couleurs additives RVB* 4
Figure I.2*: Voisinage 4 et 8 connexté* 6
Figure I.3*: Exemples d'histogramme* 6
Figure I.4*: Exemple d'application du filtre médian* 11
Figure II.1*: Variétés des formes d'un contour* 20
Figure II.2.a*: masque de Roberts* 22
Figure II.2.b*: masques de Sobel* 23
Figure II.3*: détection de contour à l'aide l'opérateur de Sobel (a) l'image d'origine, la détection de contour en utilisant la valeur de seuil (b) 110, (c) 90, (d) 70* 24
Figure II.4*: masques Prewitt en 90 'rotations successives* 25
Figure II.5*: détection de contour à l'aide l'opérateur Prewitt avec des suilles (a) 110, (b) 90, (c) 70 (d) 50* 25
Figure II.6*: Masques de Kirsch pour calculer des gradients dans les quatre directions* 26
Figure II.7*: masques de Laplace (a) dans la direction X, et (b) dans la direction Y* 28
Figure II.8*: Coopération Séquentielle Région-Contour* 28
Figure II.9*: Coopération associative Région-Contour* 29
Figure III.1 *: Initialisation du contour actif* 32
Figure III.2 *: Paramétrisation des déformations du snake* 33
Figure III.3 *: illustrations des grandes étapes du déroulement d'un snake* 36
Figure III.4 *: les lignes de fort gradient et le flot* 37
Figure III.5 *: Vecteurs* $(u_i - u_{i-1}$ *et* $u_{i+1} - u_i)$ *et variation* $(u_{i-1} - 2u_i + u_{i+1})$ 38
Figure III.6 *: Le voisinage de neuf points* 40
Figure III.7 *: Minimisation par algorithme dynamique* 40
Figure IV.1 *: Principe de contour actif* 43
Figure IV.2 *: contour actif (image de synthèse)* 44
Figure IV.3 *:* Test (image 1) 45
Figure IV.4 *:* Test (image 2) 46
Figure IV.5 *:* Test (image 3) 47
Figure IV.6 *:* Test (image 4) 48

Introduction générale

Il est parfois demandé à l'homme d'être minutieux dans son travail dans lequel l'exactitude des résultats obtenus est primordiale, et même parfois vitale. dès lors ; l'introduction du traitement d'images dans différents secteurs où l'homme opère, est nécessaire pour remplacer l'observateur humain par la machine, ou au moins l'aider dans quelques tâches, étant donné que l'ordinateur permet une meilleure qualité et une excellente visibilité.

La segmentation d'images est un problème souvent posé avant toute phase de mesure ou d'interprétation. Deux approches sont envisageables : l'approche contour ou l'approche région.

Dans l'approche contour, on trouve les méthodes dérivatives qui sont les plus immédiates pour détecter et localiser les variations du signal .L'avantage des opérateurs de détection des contours est la simplicité d'utilisation. Par contre, ils sont très sensibles aux bruits et donnent des contours ouverts. Contrairement aux approches classiques de détection de contours, les modèles déformables incorporent de la connaissance à priori sur la forme de l'objet recherché. Elle peut cependant être plus générale et se limiter à des propriétés comme la continuité ou la régularité qui caractérise le contour d'un objet. Dans ce cas, on parlera plutôt de contour actif.

Les contours actifs sont largement utilisés en segmentation d'images pour leur capacité à intégrer les processus de détection et de chaînage des contours en un seul processus de minimisation d'énergie. Toutefois l'estimation des paramètres et les problèmes d'initialisation font des contours actifs une méthode difficile à calibrer.

Notre projet intitulé la segmentation d'image par la méthode de contour actif, au fête Le modèle de contours actifs se présente sous la forme d'une courbe fermée ou ouverte dont L'initialisation est située à proximité du contour qu'on veut obtenir.

Il est important de noter que ce type d'approche est particulièrement bien adapté à la segmentation d'un objet dans une image. C'est pourquoi, ces dernières années, les contours actifs sont devenus très populaires.

Ce mémoire s'articule autour de quatre chapitres, dans le plan de travail sera comme suit :

Le premier chapitre est consacré à la présentation de traitement d'image en général, en situons quelques notions sur le traitement d'image.

Le second chapitre introduit les méthodes de segmentation d'image.

Le troisième chapitre est dédié à l'étude et la mise en œuvre de la méthode de détection de contour, il s'agit de la méthode de contours actif qui basé sur la minimisation de l'énergie.

Le chapitre IV contient les résultats de notre programme et discutions de ces derniers.

Et finalement on termine par une conclusion générale.

Chapitre I

Généralités sur le traitement d'image

1. Introduction

Le traitement d'images touche à l'heure actuelle de nombreux domaines, avec des objectifs aussi variés comme l'aide au diagnostic pour les images médicales, la télédétection et la poursuite de cible (domaine militaire) ou l'analyse des ressources terrestres à partir des images satellitaires .le traitement de ces images est à la fois simple dans son concept et difficile dans sa réalisation. Simple en effet, puisqu'il s'agit de reconnaitre des objets que notre système visuel perçoit rapidement. Difficile cependant, à cause de la grande qualité d'informations contenues dans l'image, il faut extraire des éléments pertinents pour l'application visée et ceci indépendamment de la qualité de l'image.

Le traitement d'images consiste à améliorer le contenu d'une image pour en extraire de l'information (lui ajouter du sens par un procédé automatique). Pour cela il peut être nécessaire de supprimer les facteurs de bruit, de rehausser les signaux utiles contribuant à une meilleure lisibilité .le traitement d'images s'est donc doté d'outils et des méthodes puissants issues de domaines aussi variés que les mathématiques, l'informatique qu'a réduit considérablement le temps de calcul consacré aux applications complexes.

On peut distinguer trois grandes classes faisant appel au traitement d'image :

- La retouche des images photographiques, les arts graphiques, les applications multimédia.
- La reconnaissance de formes, la reconstruction 3D et vision en robotique.
- Le contrôle non destructif (identification et/ou comptage de cellules dans un tissu biologique) l'analyse de matériaux et de surfaces, l'imagerie médicale, la télédétection, etc.

2. Notion élémentaires

2.1. *Image*

L'image est une représentation planaire d'une scène ou un objet situé en général dans un espace tridimensionnel [1]. C'est une reproduction totale de la représentation analogique d'un objet ou d'une scène réelle.

L'image est représentée par une matrice de points (pixel) à deux valeur numérique f(x, y). La matrice est composée de N ligne et P colonnes. Chaque élément est caractérisé par sa localisation dans la matrice (x, y), et l'intensité de ce dernier correspond au niveau de gris. L'image numérique est la base du traitement numérique d'image [2].

2.2. *Image binaire*

Une image binaire est une image pour laquelle chaque pixel ne peut avoir pour valeur que 0 ou 1. La manipulation d'une de telles images contient beaucoup de théories mathématiques.

Pour l'obtention d'une image binaire à partir d'une image en niveaux de gris, se reporter aux techniques de seuillage [3].

2.3. *Les images à niveau de gris*

Pour représenter de façon plus efficace les images monochromes, il faut attribuer à chaque cellule de 1'image une valeur correspondant à la quantité de lumière renvoyée [4]. C'est à dire la couleur d'un pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Chaque point n'est donc plus représenté par un bit, mais par un octet ce qui donne 2^8 =256 niveaux de gris.

2.4. *Les images en couleurs*

Même s'il est parfois utile de pouvoir représenter des images en noir et blanc, les applications multimédias utilisent le plus souvent des images en couleurs.

La représentation des couleurs est basée sur les mêmes techniques que celles employées pour les images monochromes, avec quelques particularités. On peut représenter les couleurs à l'aide de leurs composantes primaires. Les systèmes émettant de la lumière, comme les écrans d'ordinateur, sont basés sur le principe de la synthèse additive, dans lequel les couleurs sont composées d'un mélange de rouge, de vert et de bleu (modèle RVB). La représentation en couleurs réelles consiste à utiliser 24 bits pour chaque point de l'image 8 bits par couleurs, il est donc possible de représenter environ 16,7 millions de couleurs différentes. Par ailleurs, l'œil humain n'est pas capable de distinguer autant de couleurs [4].

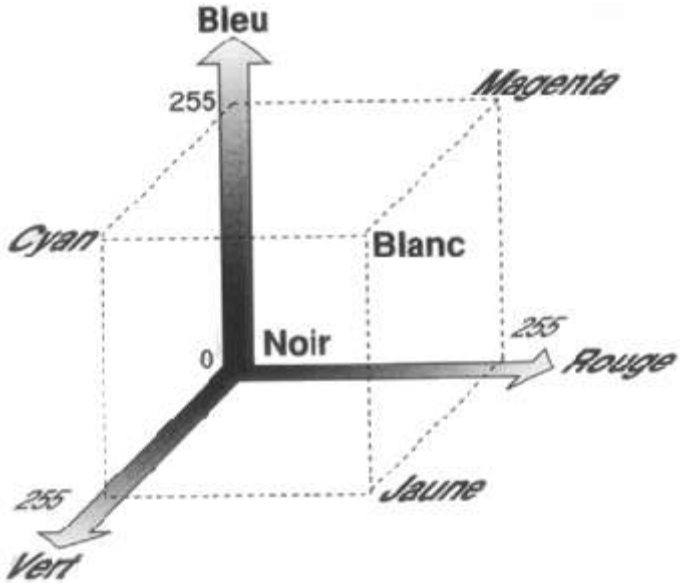


Figure I.1 : Les couleurs additives RVB.

2.5. *Pixel*

Le pixel (Picture Cell) est un point de l'image (ou du tableau bidimensionnel), c'est à dire la donnée brute, issue du capteur après conversion analogique / numérique [5]. II est caractérisé par :

- Une position dans la matrice d'image.
- Une valeur numérique indiquant sa couleur.

2.6. *Point de contour*

Le point de contour est un pixel reconnu appartenant à un contour de l'image, c'est à dire qu'il est le siège de variations importantes d'une propriété (par exemple de l'intensité) de l'image. [5]

2.7. *Point d'intérêt*

Le point d'intérêt est un pixel reconnu comme extremum local d'une propriété de 1'image (extremum local de l'intensité). II peut être obtenu soit directement (à bas niveau de traitement) soit après un traitement plus long [5]

2.8. *Partition*

Effectuer une partition de 1'image consiste à regrouper des pixels entre eux ou découper une image de manière à ce que les ensembles de points résultants vérifient la propriété de connectabilité. De plus, l'image doit être totalement segmentée et les éléments de la partition doivent être deux à deux disjoints [5].

2.9. *Connectabilité*

Xi et X j sont deux points connectables si et seulement si :

- ils appartiennent tous deux à un élément de la partition.
- II existe au moins un chemin permettant de les réunir, constitué de pixels deux a deux voisines et appartenant tous à cet élément de la partition [5],

2.10. *Connexité*

Un ensemble de points est connexe si et seulement si ses points sont deux a deux connectables. II existe deux types de connexité (figure 1.2), la 4-connexité qui ne tient compte que 4 pixels horizontaux et verticaux dans le voisinage 3*3 d'un pixel central, et la 8- connexité qui tient compte en plus des 4 pixels horizontaux et verticaux les 4 pixels diagonaux.

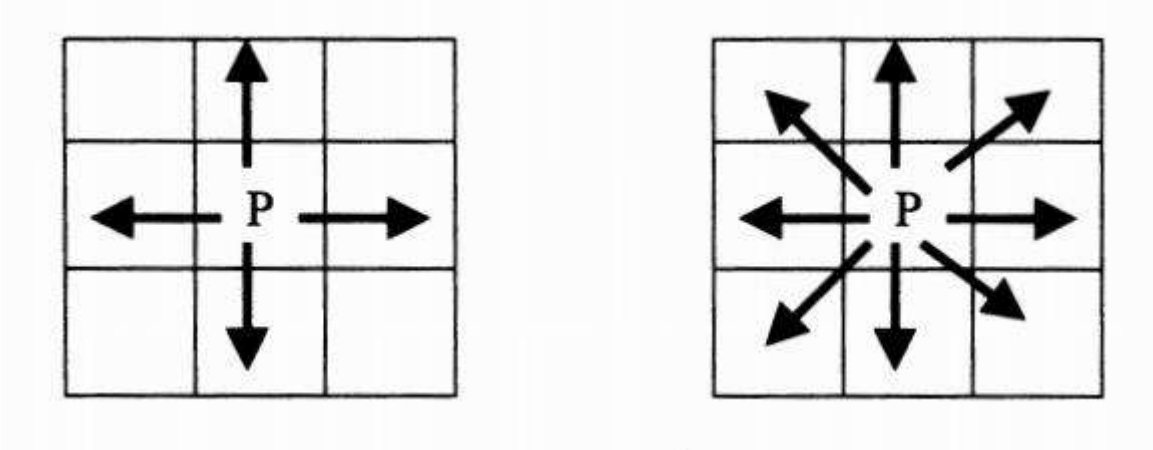


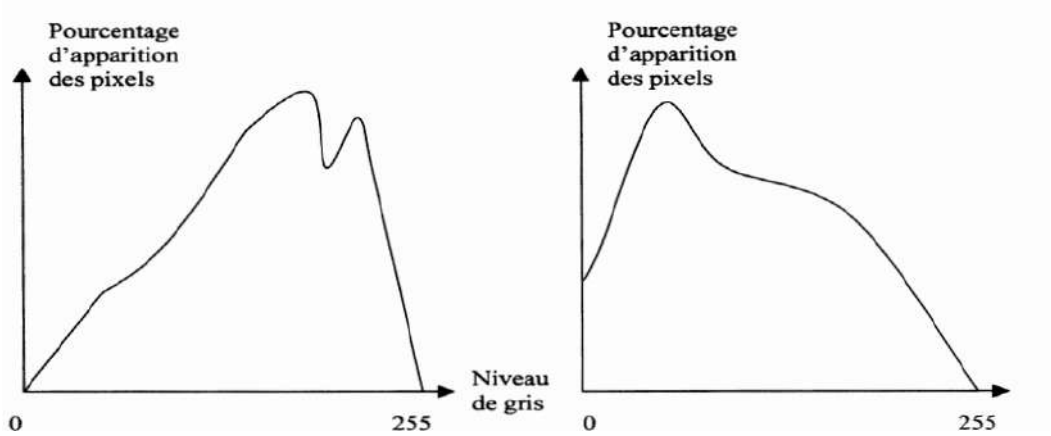
Figure I.2 : Voisinage 4 et 8 connexité.

2.11. Histogramme

L'histogramme des niveaux de gris d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris.

On appelle aussi «La dynamique de 1'image ». L'histogramme permet de tirer des informations sur la distribution des niveaux de gris.

II permet aussi d'avoir les bornes entre lesquelles la majorité sont repartis des niveaux de gris dans le cas d'une image trop claire ou trop fonce comme le représentent la **figure I.3** :



a)Histogramme d'une image claire. b) Histogramme d'une image sombre.

Figure I.3 : Exemples d'histogrammes.

- Dans l'histogramme **a** la plupart des valeurs sont proches de la couleur blanche (255).
- Dans l'histogramme **b** la plupart des valeurs sont proches de la couleur noire (0).

2.12. *Contraste*

C'est l'opposition marquée entre deux régions d'une image. Une image contrastée présente une bonne dynamique de la distribution des valeurs de gris sur tout l'intervalle des valeurs possibles, avec des blancs bien clairs et des noirs profonds. Au contraire une image peu contrastée a une faible dynamique, la plupart des pixels ayant des valeurs de gris très proches.

Si *L1* et *L2* sont les degrés de luminosité respectivement de deux zones voisines *A1* et *A2* d'une image, le contraste est défini par le rapport :

$$\mathrm{C} = \frac{L1 - L2}{L1 + L2}$$

2.13. *Luminance*

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

3. formats de fichier image

Il y a un certain nombre de formats de fichiers dans laquelle on peut stocker les images dans les fichiers et les récupérer. Ils sont connus comme des normes de format de fichier d'image ici, nous allons présenter quelques normes de format d'images les plus couramment utilisés.

3.1. *Format TIF (.tif) Tagged Image File*

ce format n'est pas lié à une application particulière ,un fichier tiff contient une série d'annotations (tags)qui décrivent les caractéristique de l'image :résolution ,taille ,couleur,

etc .les spécification de ce format sont ouvertes ,produisant ainsi une multitude de type de fichier tiff . De ce fait, il est impossible à une routine de conversion de lire tous ces différents format. Ce format existe sur les systèmes pc.

3.2. *Format JPEG (.jpg) Joint Photographie Experts Groupe*

Ce format utilisé pour comprimer des fichiers en mode point, en éliminant l'information considérée non perceptible par la vision humaine.si le ficher est comprimé il fut le décomprimer, mais les différences ne doivent pas être perceptibles.

3.3. *Format MPEG (.mpg) Moving Picture Experts Group*

Ce format est largement utilisé à travers le Web et est utilisé uniquement pour les images animées. Il utilise la compression, ce qui donne uniquement les vidéos avec perte.

3.4. *Format GIF (.gif) Graphiques Interchange Format*

Le format gif est principalement utilisé pour des échanges de graphiques entre systèmes d'information électronique. Car il offre des options de compression.

3.5. *Format BMP (.bmp) bitmap Picture*

C'est le format de base dans la programmation de Windows pour des images en mode point, et les accessoires standards de Windows. Les images et les icones de Windows sont stockés dans ce format BMP.

4. Les étapes du traitement d'image

La plupart des applications de traitement d'image passent par les étapes suivantes :

- **Acquisition**

L'acquisition a pour objet le passage de la scène physique à l'aide d'un capteur d'image(Camera) à une forme numérique observée .par le biais d'un convertisseur analogique/numérique (Carte d'acquisition). Qu'a pour rôle de sélectionner la camera active

(dans le cas de plusieurs cameras) et de sélectionner l'écran d'affichage (dans le sens numérique/analogique).

- **Prétraitement**

L'étape de prétraitement a pour but de faciliter la segmentation en renforçant la ressemblance entre pixels appartenant à une même région ou en accentuant la dissemblance entre pixels appartenant à des régions différentes. Les images comportent généralement une grande quantité de données. Pour éviter des temps de traitement trop élevés, on cherche donc à leur appliquer des opérateurs locaux, qui ne font appel pour le calcul, qu'a un nombre limité de pixels, situés dans un proche voisinage du pixel courant [1].

- **Analyse**

L'analyse est une suite d'opérations permettant l'extraction de l'information essentielle contenue dans une image. Elle est essentiellement composée par la phase de segmentation qui fournit une partition de l'image. A partir de cette segmentation, des techniques de description et de modélisation permettent d'obtenir la description structurelle de l'image.

- **Interprétation**

L'interprétation a pour but le passage de la description structurelle à la description sémantique en regard à certains objectifs. Cet objectif peut être très simple (mesure de certains paramètres sur des formes) ou beaucoup plus complexe (description du contenu de la scène en termes de concepts non mathématiques). Dans le cadre de la vision industrielle pour un objectif de qualité, les objectifs sont le plus souvent (en ordre croissant de complexité) :

- Contrôle dimensionnel (mesures simples).
- Contrôle d'aspect (mesure de texture).
- Contrôle structurel (analyse des composants d'un objet).
- Tri (reconnaissance).

5. Le filtrage

Les régions formant l'image sont caractérisées par leur intensité moyenne. Les fluctuations autour de cette intensité moyenne peuvent provenir soit du dispositif d'acquisition (camera, amplificateurs,...) soit de la scène elle-même. Elles constituent des perturbations qui affectent la qualité de la segmentation et sont généralement désignées sous le terme de «Bruit D'image». Le bruit est donc un phénomène de brusque variation d'intensité d'un pixel par rapport à ses voisins [2].

L'objectif du filtrage est donc :

- Réduire l'amplitude des variations d'intensité dans chaque région.
- Conservation des transitions entre régions adjacentes.

Cette préoccupation apparait dans les différentes approches de la segmentation.

Les méthodes les plus simples et les plus faciles à implanter sont fondées sur le filtrage linéaire, compte tenu des limitations de cette approche, on présentera également des méthodes basées sur des opérateurs non linéaires et adaptatifs.

5.1. *Filtrage linéaire*

Un filtre linéaire consiste à effectuer la convolution de l'image originale I(X, Y) par un masque M (i, j) de taille (2m+1)*(2m+1). La formule de la convolution à employer à chaque pixel est :

$$\boldsymbol{F(x,y) = M \sum_{i=-m}^{m} \sum_{j=-m}^{m} I(x+i, y+j)} \ . \qquad \textbf{(I.1)}$$

Le filtre moyen est un filtre linéaire, il s'implante donc comme une convolution, c'est à dire comme une combinaison des valeurs des intensités des pixels de son voisinage avec

M = 1 / [(2m+l)*(2m+l)], donnant l'operateur «de moyenne ». L'inconvénient majeur de filtrage linéaire est qui ne préserve pas les transitions, un tel filtrage altérera l'information intéressante c'est à dire les contours.

5.2. *Filtrage non linéaire*

Le filtre non linéaire le plus connu est le filtre médian. Son principe consiste à classer les valeurs du voisinage du pixel courant (3*3 ou 5*5) de plus petite *à* la plus grande, puis lui affecter la valeur médiane.

Ce filtre non linéaire permet d'éliminer des pixels isoles ou les trainées fines noyées dans des zones assez homogènes.

Plus le filtre médian est grand plus le filtrage peut paraitre efficace, mais il déforme l'image sans pour autant adoucir son contraste.

8	6	7	8	9
10	11	12	1	2
4	5	6	7	8
10	11	12	14	15

Image 5*4 pixels

8	6	7	8	9
10	7	7	7	2
4	10	11	12	8
10	11	12	14	15

Image après le filtrage

Figure 1.4 : Exemple d'application du filtre médian

5.3. *Filtrage adaptatif*

- ***Filtre de Nagao***

  Ce filtre est conçu afin de préserver les contours tout en éliminant le bruit. En fait, il mélange subtilement les procédures linéaires et non linéaires, il opère sur des fenêtres 5*5 et fait appel aux moyenne (ou médiane) et variance.

  L'opération de filtrage se fait en plusieurs étapes [2] :

1. Déterminer la position du masque qui donne la variance minimale du niveau de gris.

2. Affecter au point (x, y) (point central du voisinage) la moyenne du masque de variance minimale.

3. jusqu'à ce ne pas changer les niveaux de gris de l'image.

Lorsqu'un voisin contient un contour, la variance du niveau de gris devient importante, ce qui explique 1'utilisation de cette variance comme mesure de non-homogénéité du voisinage et de préserver ainsi les contours. L'inconvénient de ce filtre est la «Lourdeur » des calculs.

5.4. *Comparaison des méthodes de filtrage*

Le filtrage linéaire altère l'information intéressante c'est à dire modification des contours. Les méthodes de filtrage adaptatif donnent des résultats impressionnants concernant l'élimination du bruit, leur inconvénient majeur est la lourdeur des calculs.

Le filtrage médian a comme avantage la diminution des composantes du bruit, sans pour autant altérer l'information issue des contours, avec un temps de calcul plus ou moins raisonnable.

6. Seuillage d'une image

Le seuillage est l'une des approches les plus anciennes et importantes de détection d'objet à partir d'histogramme des niveaux de gris **f(x, y**), elle est appliquée pour des images simples.

Soit une image composée d'objets gris clair sur un fond sombre. Cette image a un histogramme **p = p (f(x, y))** forme de deux groupes de niveaux de gris dominants (bimodal).

Une méthode évidente d'extraction des objets à partir du fond est de sélectionner une valeur seuil **s** séparant les deux groupes dominants en deux bandes B1 et B2.

Alors l'image souillée est appelée **g(x, y**) et satisfait le critère suivant :

$$\mathbf{g(x,y)} = \begin{cases} \mathbf{1}\ \boldsymbol{si}\ \mathbf{f(x,y)} \geq \boldsymbol{s} \\ \mathbf{0}\ \boldsymbol{si}\ \mathbf{f(x,y)} < \boldsymbol{s} \end{cases} . \qquad \textbf{(I.2)}$$

- ✓ Si **s** ne dépend que de **f(x, y),** alors le seuillage est dit **«global ».**
- ✓ Si **s** dépend de **f(x, y)** et de **p,** alors le seuillage est dit **«local ».**
- ✓ Si **s** dépend du point, de **f(x, y)** et de **p,** alors le seuillage est dit **«dynamique ».**
- ✓ Si l'histogramme est multimodal, deux seuils successifs délimitent un mode.

6.1. *Seuillage global*

C'est une technique simple et souvent utilisée pour segmenter des images. Elle consiste à diviser l'espace des intensités en bandes et à déterminer des seuils afin d'obtenir des régions ou des points frontières.

Considérons 1'histogramme bimodal **p (f(x, y))** (zone sombre et zone claire) défini ci-dessous. La détection des frontières dans les directions horizontal et verticale nécessite deux balayages de **f(x, y)**.

- Un balayage horizontal : Pour chaque ligne de l'image **f(x, y**). On crée une ligne dans une image intermédiaire $\mathbf{g_1(x,}$ y) telle que :

$$\mathbf{g_1(x,y)} = \begin{cases} \boldsymbol{L_e\ si}\ \mathbf{f(x,y)}\ \boldsymbol{et}\ \mathbf{f(x,y-1)} \text{ sont des bandes défferentes} \\ \boldsymbol{L_b\ sinon} \end{cases} \quad \textbf{(I.3)}$$

- Un balayage vertical : Pour chaque colonne de l'image **f(x, y)**. On crée une ligne dans une image intermédiaire $\mathbf{g_2(x, y)}$ telle que :

$$\mathbf{g_2(x,y)} = \begin{cases} \boldsymbol{L_e\ si}\ \mathbf{f(x,y)}\ \boldsymbol{et}\ \mathbf{f(x-1,y)} \text{ sont des bandes défferentes} \\ \boldsymbol{L_b\ sinon} \end{cases} \quad \textbf{(I.4)}$$

L'image seuillée g(x, y) est obtenue par :

$$\mathbf{g(x,y)} = \begin{cases} \boldsymbol{L_e\ si}\ \mathbf{g_1(x,y)}\ \boldsymbol{ou}\ \mathbf{g_2(x,y)} \text{ sont égaux à } \boldsymbol{L_b} \\ \boldsymbol{L_b\ sinon} \end{cases} \quad \textbf{(I.5)}$$

Cette technique peut être généralisée à plusieurs bandes de niveaux de gris.

6.2. *Seuillage adaptatif ou local*

Dans certaines images, l'intensité du fond n'est pas constante et le contraste des objets varie dans l'image. Dans de telles images, un seuil donnant des résultats satisfaisants pour une partie ne convient pas aux autres zones de l'image. Dans ces situations, il est plus intéressant d'utiliser un seuil variant en fonction de la position dans l'image. Donc l'image est partitionnée en différentes sous-régions et un seuil est déterminé pour chacune des sous-régions.

Nous devons avoir un ensemble de seuils $\{t_1, t_2, t_{3,\ldots} t_k\}$ tels que tous les points-image satisfont la condition suivante : $f(x,y) \in \{t_i, t_{i+1}\}\left(\text{la } i^{éme} \text{ région}\right), i = 0, l, \ldots, k$.

6.3. *Seuillage par hystérésis*

D'autres techniques de seuillage existent et sont employées dans quelques méthodes de détection des contours, par exemple le seuillage par hystérésis

On se donne deux seuils T_H et T_B (haut et bas) et une image G (i, j). Le seuillage par hystérésis est obtenu de la manière suivante :

- Les points-image de G (i j) dont la valeur de niveau de gris est supérieure à T_H sont détectés.
- Les points-image de G (i, j) dont la valeur de niveau de gris est comprise entre T_H et T_B sont détectes s'ils sont connexes a un point détecté.
- Les points-image dont la valeur de gris est inférieure à T_B ne sont pas détectes.

6.4 *Comparaison des techniques de seuillage*

Les méthodes de seuillage local el hystérésis peuvent être vues comme un problème de classification c'est à dire simplifier l'image à analyser, tout en conservant le maximum d'informations sur la géométrie des objets qui y sont représentés. Elles permettent d'effectuer cette classification même si les non-stationnarités de l'image rendent 1'histogramme global *inexploitable. Ces méthodes utilisent des informations* locales (voisinage) ou semi-locales (bloc) Elles nécessitent d'y ajouter des opérations en aval (suppression de petites régions) ce qui implique un cout élevé en temps d'exécution et une complexité du traitement assez importante.

Le but du seuillage globale est de rechercher un seuil divisant l'histogramme en deux bandes distinctes de niveaux de gris, en vue de déterminer les frontières entre les objets et le fond. Cette technique conserve son intérêt et elle reste simple et rapide.

D'une manière générale le choix des seuils est crucial pour que les résultats soient de bonne qualité dans toutes les techniques de seuillage.

7. Système de traitement d'image

Dans le contexte de la vision artificielle, le traitement d'images se place après les étapes d'acquisition et de numérisation, assurant les transformations d'images et la partie de calcul permettant d'aller vers une interprétation des images traitées. Cette phase d'interprétation est

d'ailleurs, de plus en plus intégrée dans le traitement d'images, en faisant appel notamment à l'intelligence artificielle pour manipuler des connaissances, principalement sur les informations dont on dispose à propos de ce que représentent les images traitées.

8. Quelques notes importantes

Il y a quelques notes importantes qui doivent se rappeler lors de la conception d'un système de traitement d'image. Si le niveau de l'éclairage est faible, elle peut conduire à une sous-exposition du capteur alors que trop d'éclairage peut conduire à une surexposition. Les deux sous et images surexposées besoin prétraitement pour détecter des objets dans une scène. Il est également important de choisir la résolution correcte du capteur de sorte que les images capturées ont une résolution de l'objet adéquat qui aide à la reconnaissance automatique.

9. Conclusion

Le traitement d'images a pour vocation l'étude, la conception et la réalisation de systèmes d'exploitation d'images considérés comme véhicules de l'information. Son but ultime consiste à extraire le contenu informationnel (ou information pertinente) des images en vue d'une prise de décision ou d'une action en passant par plusieurs étapes.

Les différentes techniques de filtrage ont pour but d'améliorer l'homogénéité des régions qui la composent, en renforçant le contraste entre régions adjacentes.

Les méthodes de seuillage visent à simplifier l'image à traiter en conservant toutes les informations sur les objets contenus dans cette image. Elles ne donneront donc de bons résultats que si les pixels sont proche géographiquement (situes dans une même région).

Chapitre II

La segmentation des images

1. Introduction

La segmentation est un traitement de bas-niveau qui consiste à créer une partition de l'image **A** en sous-ensembles **Ri** appelés régions, ayant des propriétés communes (intensité, texture,...) qui les différencient des pixels des régions voisines. [l]

II n'y a pas une méthode unique de segmentation d'une image, le choix d'une technique est lié :

- A la nature de l'image :
  - Eclairage non homogène.
  - Présence de bruit, de zones texturées.
  - Contours flous.
- Aux primitives à extraire :
  - Contours, segments droite, angle ...
  - Région, Formes.
  - textures.
- Aux contraintes d'exploitation
  - Complexité algorithmique, fonctionnement en temps réel.
  - Taille de la mémoire disponible en machine.

2. Objectifs de la segmentation

La segmentation d'image sert à :

- fournir des régions homogènes (selon un critère donné)

- réduire le bruit.

- localiser de manière précise les contours des régions.

3. Définition de la segmentation

Une segmentation complète d'une image **R** implique l'identification d'un ensemble fini de régions (R1, R2, R3, ..., RN) telle que :

$$\begin{cases} \mathbf{1}.\boldsymbol{R} = \boldsymbol{R_1} \cup \boldsymbol{R_2} \cup \dots \boldsymbol{R_N} \\ \mathbf{2}.\boldsymbol{R_i} \cap \boldsymbol{R_j} = \emptyset, \qquad \forall \boldsymbol{i} \neq \boldsymbol{j} \\ \mathbf{3}.\boldsymbol{P}(\boldsymbol{R_i}) = \boldsymbol{Vrai}\,, \qquad \forall \boldsymbol{i} \\ \mathbf{4}.\boldsymbol{P}(\boldsymbol{R_i} \cup \boldsymbol{R_j}) = \boldsymbol{Faux}\,, \qquad \boldsymbol{i} \neq \boldsymbol{j} \end{cases} \quad \textbf{(II.1)}$$

La première condition indique que l'union des régions nous ramène à l'image de départ.

La deuxième indique que deux régions différentes sont disjointes(les régions n'ont pas des similarités entre eux).

La troisième exprime que les pixels appartenant à une région doivent satisfaire le critère d'homogénéité.

La quatrième exprime le critère d'homogénéité pour la segmentation en régions disjointes.

Il peut exister un certain nombre de partitions possibles, mais la sélection d'un ensemble approprié de régions dépend du choix de la propriété d'association dans la région P.

En outre, la connectivité entre les pixels dans une région doit être considérée.

Les algorithmes de segmentation sont basés sur l'une des deux propriétés de base, les niveaux de gris des valeurs de discontinuité et la similitude entre les pixels.

Dans la première catégorie des algorithmes, on partitionne une image basée sur des changements brusques de niveau de gris. Les principaux domaines d'intérêt dans cette catégorie sont la détection des lignes et des contours dans une image. Ainsi, si l'on peut extraire les contours dans une image et de les relier, la région est décrite par le contour de bordure qui le contient.

Nous pouvons voir le processus de segmentation d'un autre point de vue. À partir de ce point de vue, l'ensemble connexe de pixels ayant plus ou moins la même intensité homogène former les régions. Les principales approches de la seconde catégorie sont calculées sur la similitude entre les pixels à l'intérieur d'une région. [7]

Du fait de cette diversité, il est difficile de définir de manière absolue, une bonne segmentation. Ceci donne naissance à des approches couramment qualifiées :

- Approche Région.
- Approche Contour (frontière).
- Approche Mixte (Coopérative, dualité contour/région).

4. Approche région (extraction de région)

Une région est un ensemble connexe de points images (pixels) ayant des propriétés communes (intensité texture,...) qui les différencient des pixels des régions voisines[l], Une région est limitée par un contour.

La segmentation en région vise à extraire les zones homogènes de l'image supposées être la projection de l'intérieur des objets [6].elle comporte beaucoup moins d'étapes classiquement différenciées. Généralement, elles sont constituées par une image d'étiquettes, ou d'une structure de données dans laquelle sont contenus les divers attributs géométriques et photométriques.

Une chaine d'extraction des régions est composée des étapes suivantes :

- Première étape constitue d'une division ou d'une fusion
- Croissance isotrope à partir de régions embryons (rebouchage des régions).
- Fusion des pixels adjacents (regroupements des pixels).

4.1. *Partition topologique*

La segmentation en régions vérifie les trois propriétés suivantes :

1. Les régions sont des zones connexes de l'image. Une zone est connexe si et seulement si quel que soit le couple de points choisis appartenant à la région, il existe un chemin de points adjacents les reliant et totalement inclus dans la zone connexe.
2. L'image est totalement segmentée c'est-à-dire l'union des régions est égale à l'image. en d'autres termes, tout pixel de l'image appartient au moins à une région.
3. Quel que soit le couple de régions choisi, l'intersection entre ces deux régions est

1'ensemble vide. En d'autres termes, tous les points de l'image appartiennent au plus à une région.

4.2. *Prédicats d'homogénéité*

Définissons tout d'abord la notion de critère **C(R)**, qui est une valeur réelle quantifiant une propriété d'homogénéité d'une zone de l'image candidate à être une région. Un prédicat **P(R)** associe au critère **C(R)** est une fonction booléenne. [6]

Telle que :

$$\begin{cases} \boldsymbol{P(R) = VRAI\ \ Si\ \ C(R)} > \text{Seuil } (\textbf{ou Si C(R)} < \text{Seuil}). \\ \textbf{P(R) = FAUX\ \ Sinon}. \end{cases} \quad \textbf{(II.2)}$$

Une segmentation en région est donc une partition topologique de l'image qui doit satisfaire aux contraintes sur le prédicat ci-dessous.

Soit **Rk** un élément de la partition ou région :

$$\begin{cases} \textbf{P(Rk) = VRAI pour tout k} \\ \textbf{P (Ri U Rj) = FAUX pour tout couple Ri, Rj de régions adjacentes}. \end{cases} \quad \textbf{(II.3)}$$

Cette dernière contrainte impose à la partition d'être minimale, ce qui veut dire que plus aucune fusion (respectant le prédicat prédéfini) n'est possible.

Le prédicat porte sur des attributs d'homogénéité de la fonction intensité. Deux exemples courants sont les prédicats associes aux critères Max moins Min et d'écart type (variance).

- **Critère Max Moins Min :**

On pose : **Min (Rk) = Min {I (x) ∈ Rk}** avec **I (x)** l'intensité du pixel **X**, pour tout pixel **X** appartenant à la région.

Et **Max (Rk) = Max {I (x) ∈ Rk}.**

La valeur du critère est alors la différence entre le Maximum et le Minimum :

C (Rk) = Max (Rk) - Min (Rk).

Le prédicat est une simple fonction de seuil sur ce critère :

$$\begin{cases} \textbf{P(Rk) = VRAI Si C (Rk) = Max (Rk) − Min (Rk) < Seuil} \\ \textbf{P(Rk) = FAUX Sinon} \end{cases} \quad \textbf{(II.4)}$$

- **Critère sur L'écart type ou la variance :**

$\boldsymbol{\sigma}$ Etant L'écart type (ou la variance) sur la région, C(Rk) = $\boldsymbol{\sigma}$ et comme précédemment le prédicat est constitué par simple fonction de seuil :

$$\begin{cases} \textbf{P(Rk) = VRAI Si } \boldsymbol{\sigma} \textbf{ < Seuil} \\ \textbf{P(Rk) = FAUX Sinon} \end{cases} \quad \textbf{(II.5)}$$

5. Approche contour

Une image, qu'elle soit en noir et blanc ou en couleur, quelle que soit son origine physique (visible, infra rouge, rayon X, scanner ...) est dans la plupart des cas constituée de zones homogènes que nous appellerons régions, n'est pas uniforme ; par exemple, un mur de brique est une surface homogène, mais non uniforme. En effet, les joints sont de couleurs différentes des briques .ce qui induit de nombreux contours. Deux régions adjacentes sont séparées par une zone de transition d'extension faible par rapport à l'étendue des régions, appelée contours. Pour simplifier, la région est l'intérieur d'un contour fermé, et le contour fermé est la frontière d'une région [5]. Cette approche est associée à une variation d'intensité ou à une discontinuité entre les propriétés de deux ensembles connexes de points [1].
Le but de la segmentation en contours est de repérer dans l'image ces discontinuités pour tenter d'inférer par la suite une hypothèse relative au phénomène que l'image subit [5].

5.1. *Notion de contour*

II semble difficile de donner une définition suffisamment générale pour couvrir tous les types de contours dans l'image. La complexité de la détection de contour provient de la grande diversité des transitions de luminance qui marquent la présence d'un contour. La figure suivante illustre ce fait.

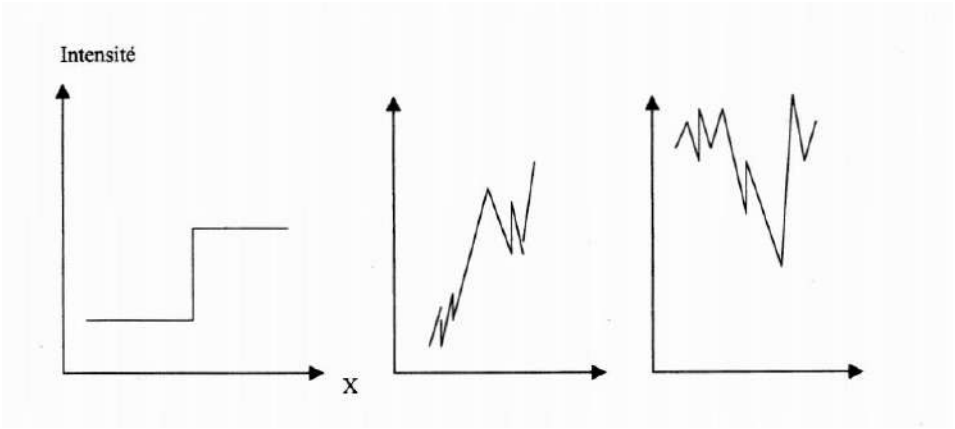


Figure II.1 : Variétés des formes d'un contour(x : ligne d'image).

Un contour est la transition entre deux régions d'intensité constante, Les paramètres du modèle (chercher dans l'image) sont :

- La différence d'intensité entre les deux régions.
- L'orientation du contour (angle).
- La longueur du contour.
- La largeur de la zone de transition.

Un contour est un ensemble de pixels formant une frontière entre deux ou plusieurs régions voisines. L'épaisseur du contour est d'un pixel ou plus (un pixel dans le cas idéal) [2].

5.2. *Les étapes d'extraction de contour*

Une chaine simple d'extraction de contours est classiquement constituée des étapes suivantes : [5]

5.2.a. *La détection des points de contours*

Permet de sélectionner à partir des pixels de l'image le siège de transitons importants «l'ensemble de points de contour ».

5.2.b. *Affinage de points de contours*

Les discontinuités des propriétés physiques ne génèrent pas forcément des variations bien localisées, ainsi la détection précédente engendre très souvent des contours d'épaisseur supérieure à un pixel. L'affinage permet d'éliminer des points de contours en trop, et de ramener l'épaisseur des contours à un pixel. [8]

5.2.*c. Suivi et la prolongation (fermeture des contours)*

Le suivi consiste à parcourir les points de contours de manière à détecter les extrémités. Ensuite, la prolongation qui consiste à détecter de nouveaux points de contour de manière à combler les lacunes précédentes, afin d'obtenir un contour fermé c'est à dire un ensemble de points dont chacun est connecte à au moins deux voisins.

5.2.d. *Chainage des contours*

Permet de passer de l'ensemble de points connexes mais non connectes précédents a une liste ordonnée de points de contours appelées contour. En général ils relient deux points de jonction de l'image (points de contour ayant au moins trois points de contour voisine). [8]

5.3. *Détecteur de contour*

Un détecteur de contour idéal est nécessaire pour détecter un point de contour précisément, dans ce sens que le véritable point de contour dans une image ne doit pas être manqué, La décision relative à l'existence d'un point de contour est basée sur un seuil. Ainsi, si l'ampleur de la pente est supérieure à un seuil, alors on déduit que le point de contour existe à ce moment-là, Si le seuil choisi est grand, il n'y a pas d'autre point de contour. Alors il y a une possibilité que les vrais points de contour peuvent être détectés.

Si le seuil est faible, de nombreux points bruyants peuvent être faussement détectés comme points de contour [7].il existe plusieurs méthodes pour détecter le contour tell que :

5.3.1. *Méthode par Masque*

Certain nombre de détecteurs de contour ont été développés par différents chercheurs. Parmi les opérateurs les plus importants sont l'opérateur Robert, opérateur de Sobel, Prewitt, l'opérateur Krisch ….etc, Ci-dessous, nous discutons de certains de ces opérateurs

A. *Opérateur de Roberts*

L'opérateur Robert Cross est un opérateur simple basé sur un opérateur de gradient 2 x 2. Cet opérateur fournit l'approximation la plus simple de l'amplitude de gradient l'opérateur du Robert est illustré ci-dessous. Comme le noyau Robert n'est qu'un masque 2 x 2, il est très sensible au bruit.

$$\boldsymbol{G[f(i,j)] = [f(i,j) - f(i+1,j+1)] + [f(i+1,j) - f(i,j+1)]} \qquad \textbf{(II.6)}$$

1	1
-1	-1

Figure II.2.a : masque de Roberts

Les nombre (1,1,-1,-1) ce sont des constants.

B. *Opérateur de Sobel*

L'Opérateur de Sobel est un opérateur de gradient à base de voisinage 3 x 3. Les masques de convolution pour l'opérateur de Sobel sont définis par les deux noyaux représentés sur la figure II.2.b Les deux masques sont appliqués séparément sur l'image d'entrée pour obtenir

deux composantes de gradient Gx, et Gy, dans les orientations horizontale et verticale respectivement

1	2	1
0	0	0
-1	-2	-1

(a)

1	0	-1
2	0	-2
1	0	-1

(b)

Figure II.2.b : masques de Sobel

$$\boldsymbol{G_x = [f(i-1,j-1) + 2f(i-1,j) + f(i-1,j+1)] -}$$

$$\boldsymbol{[f(i+1,j-1) + 2f(i,j+1) + f(i+1,j+1)]} \quad \textbf{(II.7)}$$

et

$$\boldsymbol{G_y = [f(i-1,j-1) + 2f(i,j-1) + f(i+1,j-1)] -}$$

$$\boldsymbol{[f(i-1,j+1) + 2f(i,j+1) + f(i+1,j+1)]} \quad \textbf{(II.8)}$$

L'ampleur de gradient est généralement calculée comme :

$$\boldsymbol{G[f(x,y)] = \sqrt{G_x^2 + G_y^2}} \quad \textbf{(II.9)}$$

Les deux formules dans l'équation II.10 peut être utilisé pour calculer l'amplitude de gradient. La direction du gradient est calculée comme dans **l'équation II.11**.

$$\left.\begin{array}{l} \boldsymbol{G[f(x,y)] = \sqrt{G_x^2 + G_y^2}} \\ \boldsymbol{G[f(x,y)] = |G_x| + |G_Y|} \\ \boldsymbol{G[f(x,y)] = max\{|G_x|, |G_Y|\}} \end{array}\right\} . \quad \textbf{(II.10)}$$

$$\boldsymbol{\theta(x,y) = tan^{-1}(G_y/G_x)} \quad \textbf{(II.11)}$$

Le résultat d'une image de contour générée par l'opérateur de Sobel est représenté sur la figure II.3. Les images de contours ont été calculées en utilisant les valeurs de seuil 110, 90 et 70. Il peut être observé à partir des images en tant que la valeur seuil est diminuée, de plus en plus le nombre de points de contour non deviennent des points de contour.

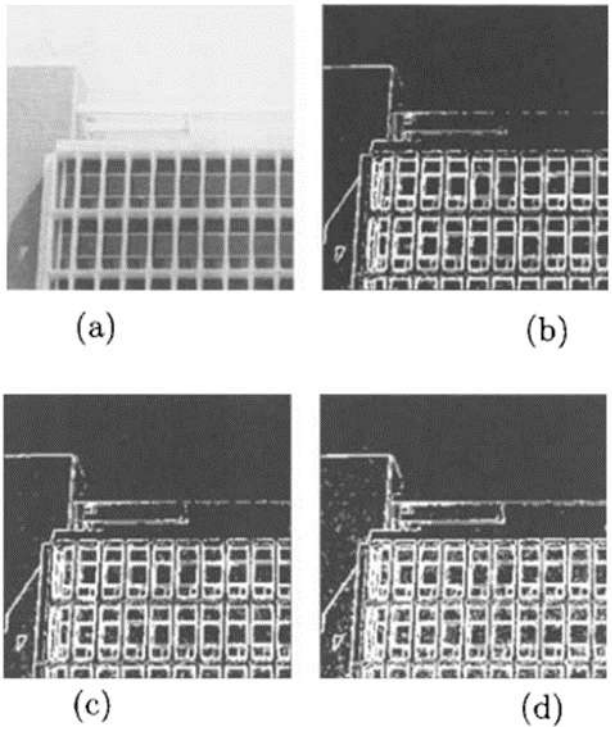


Figure II.3 : détection de contour à l'aide l'opérateur de Sobel (a) l'image d'origine, la détection de contour en utilisant la valeur de seuil (b) 110, (c) 90, (d) 70

C. *L'opérateur de Prewitt*

L'opérateur de Prewitt est défini par un ensemble de huit masques, dont quatre sont représentés sur la figure II.4. D'autres peuvent être générés par la rotation de 90' successivement. Le masque qui produit la réponse maximale donne la direction du gradient. L'amplitude et la direction des gradients de contour peuvent être calculées d'une manière similaire à celle de l'opérateur de Sobel. Le résultat d'une image de contour générée par l'opérateur de Prewitt est représenté sur la figure II.5 :

1	1	1
0	0	0
-1	-1	-1

(a)

0	1	1
-1	0	1
-1	-1	0

(b)

-1	0	1
-1	0	1
-1	0	1

(c)

-1	-1	0
-1	0	1
0	1	1

(d)

Figure II.4 : masques Prewitt en 90 ' rotations successives

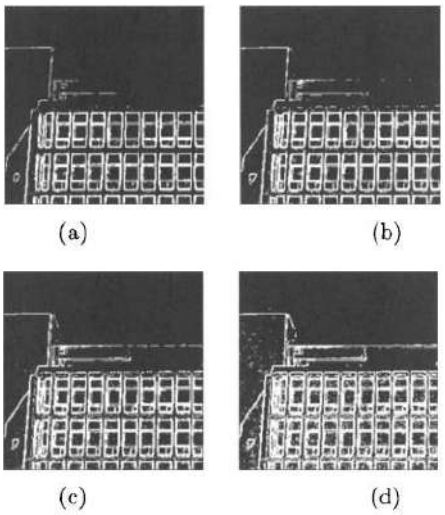


Figure II.5 : détection de contour à l'aide l'opérateur Prewitt avec des suilles (a) 110, (b) 90, (c) 70 (d) 50

D. *L'opérateur de Kirsch*

Comme pour les opérateurs Prewitt, les masques peuvent être définis dans huit directions, ce qui donne les gradients estimés dans ces directions. Sur la figure II.6, nous montrons les masques de Kirsch qui détectent les gradients en seulement quatre directions.

3	3	3
3	0	3
5	5	5

(a)

-5	3	3
-5	0	3
-5	3	3

(b)

5	-3	-3
5	0	-3
5	-3	-3

(c)

-3	-3	-3
-3	0	-3
5	5	5

(d)

Figure II.6 : Masques de Kirsch pour calculer des gradients dans les quatre directions

5.3.2 *Méthode par optimisation basées sur modélisation des conteurs*

A. *Le détecteur de contour de Canny*

Le détecteur de contour de Canny assure une bonne immunité au bruit et en même temps détecte véritables points de contour avec l'erreur minimum. Canny a optimisé le processus de détection des contours par :

1. En maximisant le rapport signal-sur-bruit du gradient :

Le rapport signal-sur-bruit du gradient est maximale lorsque les contours sont détectés vrais et les contours faux sont évités.

2. En maximisant les réponses multiples à un seul contour :

En éliminant les fausses réponses quand il y a un nombre multiple de réponses à un seul contour, les contours de bruit corrompu peuvent être enlevés. Dans ce procédé, l'image est d'abord convolée avec un filtre de lissage gaussien avec un écart type 0. Cette opération est suivie par le calcul de gradient sur l'image lissée résultante [7].

5.3.3. *Méthode différentielles*

A. *Opérateurs basée sur la dérivée seconde*

Le principe de détection de contour base sur la dérivé seconde consiste à détecter les points pour les points de contour qui possèdent des maxima locaux dans les valeurs de gradient.

Dans ce cas, nous obtenons un pic dans la dérivée première et un passage par zéro à la dérivée seconde aux points de contour. Par conséquent, les points où la dérivée seconde possède un passage par zéro sont traités comme des points de contour. Laplacien est le deuxième opérateur de contour le plus couramment utilisé.

B. *Opérateur du LAPLACIEN*

Il approxime de façon très simple la dérivée seconde de l'image. Il est souvent utilisé pour déterminer dans quel côté se trouve un pixel au voisinage d'un contour. En effet, la dérivée seconde est nulle sur ce contour, et prend des signes opposés d'autre part.

C'est un opérateur différentiel à coefficients constants, invariants par rotation [7]. Cet opérateur estime que les points de contours représentent les points où la deuxième dérivée est égale à zéro. Donc l'opérateur LAPLACIEN est exprimé par :

$$\frac{\boldsymbol{d^2 f}}{\boldsymbol{dx^2}} = \frac{\boldsymbol{dG_x}}{\boldsymbol{dx}} = \frac{\boldsymbol{d[f(i,j)-f(i,j-1)]}}{\boldsymbol{dx}} = \frac{\boldsymbol{df(i,j)}}{\boldsymbol{dx}} - \frac{\boldsymbol{df(i,j-1)}}{\boldsymbol{dx}}$$

$$= [\boldsymbol{f(i,j+1) - f(i,j)}] - [\boldsymbol{f(i,j) - f(i,j-1)}]$$

$$= \boldsymbol{f(i,j+1) - 2f(i,j) + f(i,j-1)} \qquad \textbf{(II.12)}$$

De même

$$\frac{\boldsymbol{d^2 f}}{\boldsymbol{dy^2}} = \boldsymbol{f(i+1,j) - 2f(i,j) + f(i-1,j).} \qquad \textbf{(II.13)}$$

L'opérateur de Laplace est exprimée par Les masques de convolution(les matrices des filtres) correspondant selon des directions X et Y, la réponse impulsionnelle de Laplacien est montré dans la figure II.7.

0	0	0
1	-2	1
0	0	0

(a)

0	1	0
0	-2	0
0	1	0

(b)

Figure II.7 masques de Laplace (a) dans la direction X, et (b) dans la direction Y.

6. Approche mixte (dualité contour/région)

Cette méthode combine 1'approche par régions et l'approche par contours. Elle exploite conjointement l'information contour et l'information région afin de pallier les insuffisances de chacune des deux approches. La coopération peut être associative ou séquentielle (par contraintes). [8]

Dans le cas de la coopération associative, la compatibilité des résultats fournie par chaque approche est recherchée, et dans 1'autre cas, les résultats de la seconde méthode mise en œuvre fournissent des contraintes à la première.

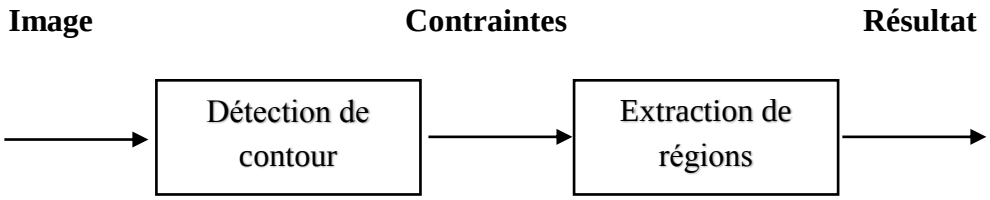


Figure II.8 : Coopération Séquentielle Région-Contour.

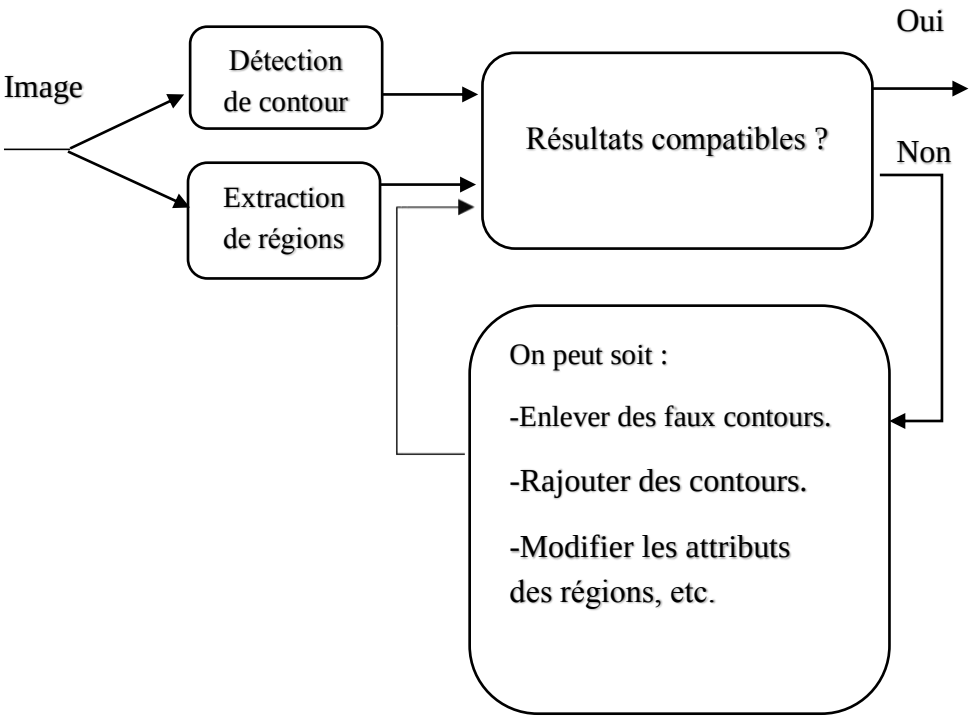


Figure II.9 : Coopération associative Région-Contour.

7. Comparaison des méthodes de segmentation

Les méthodes permettant 1'extraction directe des contours sont relativement rapides dans le sens où elles ne nécessitent que des informations locales. C'est à dire le traitement d'un volume restreint de données est généralement plus rapide que le traitement d'un flot important. Les inconvénients de ce type de primitive sont :

- La non sensibilité d'une image à l'autre résultant de la forte composante de bruit, C'est-à-dire qu'un même point d'une image à 1'autre (s'il existe dans les deux images) ne corresponds pas forcement au même point physique de 1'objet ou un point d'une image n'aura pas de correspondant dans l'image suivante.
- L'ambiguïté d'appariement liée au caractère particulièrement local de l'information. C'est à dire que rien ne ressemble autant à un point qu'un autre de la même image. D'où les ambiguïtés d'appariement pouvant surgir lors de l'association avec les points de l'image suivante.

Nous allons examiner maintenant les primitives de type global (région), par son caractère global au niveau de la phase de segmentation, la primitive région est beaucoup moins sensible au bruit contenu dans les images que les primitives locales, tel que contour ou point d'intérêt. La région est une primitive beaucoup plus stable d'une image à l'autre donc mieux adaptée à l'appariement entre images ou entre modèle et image [8]. De plus du fait de la globalité les attributs des régions sont plus discriminants d'une région à l'autre d'une même image, ce qui permet de lever de nombreuses ambiguïtés d'appariement.

Les inconvénients de ce type de primitive sont :

- De manière générale, une segmentation en régions d'une image est longue à obtenir.
- En ce qui concerne I 'utilisation des propriétés globales, il ne faut pas chercher leur précisons locale. En d'autres mots, les frontières des régions sont généralement peu précisées.

D'un point de vue efficacité sur le plan rapidité d'extraction et compatibility des informations recueillies, il s'avère très avantageux de mélanger les deux types d'approches et d'avoir recours, à un type de segmentation particulier : La segmentation mixte (coopérative ou dualité contour/région).

8. Conclusion

Nous avons présenté dans ce chapitre les différentes méthodes et techniques de prétraitement et de segmentation d'images, ce dernier est loin d'être un problème résolu.

Les méthodes de segmentation étudiées présentent quelques inconvénients. L'approche contour, par exemple nécessite souvent un algorithme de fermeture du contour. L'approche région est quant à elle généralement plus gourmande en temps calcul. Actuellement, des nouvelles méthodes sont apparues dans les deux types d'approches. Dans ce qui suit nous nous intéresserons à la segmentation par contours actifs qui permettent d'améliorer la qualité de la segmentation.

Chapitre III

La segmentation d'image par la méthode de contour actif

1. Introduction

La détection du contour d'un objet bien déterminé dans une image joue un rôle décisif dans bien des cas en traitement d'image et en vision artificielle. Plusieurs tentatives ont été faites pour accomplir cette tâche et améliorer la qualité de la segmentation. Une solution usuelle consiste à utiliser une détection locale à l'aide de détecteurs d'arêtes, comme Sobel (1970), Canny (1986), Une autre technique prometteuse dans le domaine de la segmentation des images bidimensionnelles, (détection du contour plus précisément) et considérablement étudiée durant les dernières années, est celle du contour actif proposée par Kass et al (1988). Le contour est représenté comme un modèle élastique déformable contrôlé par une contrainte de continuité dont les mouvements de glissement, se produisant lors de la déformation lui ont valu le nom de snake (ou serpent).

La segmentation est réalisée à travers un processus de minimisation d'une énergie basé sur deux forces de contrainte : force externe, qui dépend des propriétés du contour à détecter et qui est responsable de mettre le contour proche d'un minimum local en termes d'énergie, et une force interne responsable de la courbure et de la continuité du modèle.

2. Principe et fonctionnement intuitif de la méthode des contours actifs

Un contour actif est un ensemble de points qu'on va tenter de déplacer pour leur faire épouser une forme. Il s'agit d'une technique d'extraction de données utilisée en traitement d'images. L'idée de cette méthode est de déplacer les points pour les rapprocher des zones de fort gradient tout en conservant des caractéristiques comme la courbure du contour ou la répartition des points sur le contour ou d'autres contraintes liées à la disposition des points.
Au démarrage de l'algorithme, le contour est disposé uniformément autour de l'objet à détourer puis il va se rétracter pour en épouser au mieux ses formes. De la même manière, un contour actif peut aussi se dilater et tenter de remplir une forme, il sera alors situé à l'intérieur de celle-ci au démarrage de l'algorithme.

A chaque itération, l'algorithme va tenter de trouver un meilleur positionnement pour le contour pour minimiser les dérives par rapport aux contraintes utilisées.
L'algorithme s'arrêtera lorsqu'il ne sera plus possible d'améliorer le positionnement ou simplement quand le nombre maximum d'itérations aura été atteint. On utilise les notions d'énergies interne et externe pour caractériser respectivement la forme du contour et tous les

éléments qui lui sont propres, et le positionnement du contour sur l'image en tenant compte des lignes de gradient.

Voici un exemple de déroulement d'une recherche en contour actif pour détourer un objet :

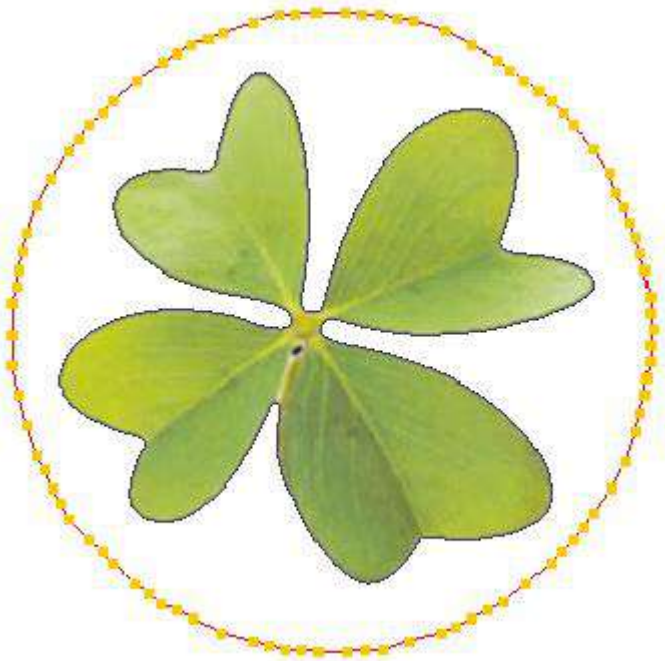


Figure III.1 : initialisation du contour actif

Chaque itération peut se représenter de la manière suivante :

- calcul des énergies interne et externe, caractérisant le contour lui-même et son positionnement sur l'image.
- pour chaque point du contour, détermination d'une nouvelle position, sur laquelle le contour devrait mieux minimiser les écarts de contraintes.
- Arrangement du contour pour qu'il respecte des contraintes d'écartement entre les points, de régularité de points.

2.1. *Énergie globale*

Dans sa forme continue, le snake est modélisé par une courbe paramétrée $v(s) = (x(s), y(s))$ sur le domaine unité($s \in [0,1]$). Son énergie est fonction de sa configuration géométrique (énergie interne) et de l'image (énergie externe). Elle s'écrit sous la forme :

$$\boldsymbol{E_{Globale}(v) = E_{Interne}(v) + E_{Externe}(v)} \quad \textbf{(III.1)}$$

Chaque point de la courbe du snake contribue au calcul de cette énergie, ce qui se traduit par :

$$\boldsymbol{E_{Global}(v) = \int_0^1 E_{Global}(v,s).ds} \quad \textbf{(III.2)}$$

Soit :

$$\boldsymbol{E_{Global}(v) = \int_0^1 \left(E_{Interne}(v,s) + E_{Externe}(v,s)\right).ds} \quad \textbf{(III.3)}$$

Il s'agit maintenant de construire des énergies internes et externes qui procurent au snake le comportement désiré.

2.2. *Énergie interne*

L'énergie interne est conçue de façon à autoriser des déformations longitudinales (élongations ou contractions) et des déformations de courbure. On la modélise par :

$$\boldsymbol{E_{Interne}(v,s) = \frac{1}{2}\left(\alpha(s)\cdot\left|\frac{\partial v}{\partial s}(s)\right|^2 + \beta(s)\cdot\left|\frac{\partial^2 v}{\partial^2 s}(s)\right|^2\right)} \quad \textbf{(III.4)}$$

Le terme $\frac{\partial v}{\partial s}(s)$ donne la valeur de l'élongation au point $\boldsymbol{s}$ du snake. Il est contrôlé par la fonction $\alpha(s)$ qui règle la tension en fonction de la position sur le snake. Le terme $\frac{\partial^2 v}{\partial^2 s}(s)$ quantifie la courbure du snake dont la rigidité est également paramétrable (par$\beta(s)$). En particulier, la figure III.2 montre que l'affectation $\alpha(s_1) = \beta(s_1) = 0$ autorise une discontinuité du snake au point *s*1 (ce qui n'a pas grand intérêt), et que spécifier $\beta(s_2) = 0$ uniquement lui permet de développer un angle au point S_2.



Figure III.2 : Paramétrisation des déformations du snake.

L'énergie interne permet donc de définir la tension et la rigidité du snake. Par contre elle ne lui impose pas de forme particulière : si on laisse un snake se stabiliser uniquement sur le minimum de son énergie interne, c'est à dire lorsque $\frac{\partial v}{\partial s} = \frac{\partial^2 v}{\partial^2 s} = 0$, alors il se détend en une ligne droite dont les "points" sont régulièrement espacés. Il faut noter en particulier que le

snake n'a pas de longueur privilégiée. Ses déformations et déplacements résultant de son interaction avec l'image sont modélisés par son énergie externe.

2.3. *Énergie externe*

De manière générale, si $v(s) = (x(s), y(s))$, alors l'énergie externe est égale à :

$$\boldsymbol{E_{Externe}(v) = \lambda \cdot \int_0^1 E_{Image}(v, s).\, ds,\ \lambda \geq 0} \qquad \textbf{(III.5)}$$

Où *E* im*age* définit la force externe en chaque point de l'image. Les minimums de ce champ de force sont des attracteurs du snake puisque celui-ci cherche à minimiser son énergie. La modularité du concept des snakes, est ici mise en évidence puisque le comportement du snake vis-à-vis de l'image dépend uniquement de E_{Image}. Différentes formes d'énergies externes sont proposées dans [9]. Soit I(x, y) la fonction représentant l'intensité de l'image. La forme la plus simple de la fonction d'énergie externe s'écrit :

$$\boldsymbol{E\ Ligne\ (\ x, y\ ) =\ I\ (\ x, y\ )} \qquad \textbf{(III.6)}$$

Le comportement du snake dérivant de cette énergie est une attirance vers les points sombres de l'image car leur signal est de valeur faible (en pratique, ceux-ci sont souvent regroupés en ligne d'où le nom de *E Ligne*). L'attraction aux points lumineux est réalisée par une simple inversion du signe :

$$\boldsymbol{ELigne(x, y) = -I(x, y)} \qquad \textbf{(III.7)}$$

La deuxième forme de l'énergie externe proposée par [9] a pour objectif de créer une attraction du snake par les contours de l'image. Elle est réalisée en utilisant le gradient de l'image$(\nabla I\ (x, y))$, Qui représente les variations d'intensité du signal. L'énergie externe s'écrit alors :

$$\boldsymbol{E_{Contour}(x, y) = -|\nabla I(x, y)|^2} \qquad \textbf{(III.8)}$$

L'élévation au carré élimine le problème du signe dû au sens de variation du gradient et fournit une valeur toujours positive. De plus, elle amplifie les forts gradients. L'inversion du signe rend cette fonction d'autant plus petite (générant l'attirance du snake) que la norme du gradient est grande.

2.4. *Utilisation des deux énergies*

Chaque position du contour actif donne une énergie interne et une énergie externe dont la somme doit être minimisée et influencera les mouvements des points du contour actif. La grande difficulté de l'utilisation des snakes réside dans le choix des pondérations à donner à chaque énergie. Traditionnellement il est d'usage d'utiliser un paramètre pour l'énergie interne et un autre pour l'énergie externe, mais un réglage plus fin peut s'opérer en ajoutant des paramètres différents pour des aspects différents de l'énergie interne ou de l'énergie externe.

2.5. *Modifications du contour*

Après avoir calculé l'énergie globale dégagée par le contour et par son positionnement sur l'image, il convient de déterminer comment le faire évoluer pour minimiser cette énergie. Pour cela, une méthode simple et intuitive est d'observer les pixels voisins immédiats de chaque point du contour pour déterminer pour chacun d'eux l'énergie globale du snake, chaque meilleur voisin devenant un point du contour.

Il est nécessaire que le contour possède toujours suffisamment de points pour être sûr de bien calculer son énergie globale, en particulier son énergie externe qui sera plus précise en tenant compte de davantage de points. C'est la raison pour laquelle il peut s'avérer pertinent de rajouter ou de supprimer des points à chaque itération si des contraintes ne sont pas suffisamment respectées. Par exemple, on pourra rajouter un point au snake si ses voisins sont trop éloignés. A l'inverse, on pourra supprimer un point s'il est trop près de ses voisins.

Outre le fait de calculer plus efficacement les énergies, l'ajout de points est nécessaire dès lors que le snake change de taille. Nous ne connaissons pas a priori la taille de l'objet à extraire au début de l'algorithme, il n'est donc pas totalement absurde de revoir le nombre de points au cours de son déroulement. Comme il est illustré dans la figure III.3

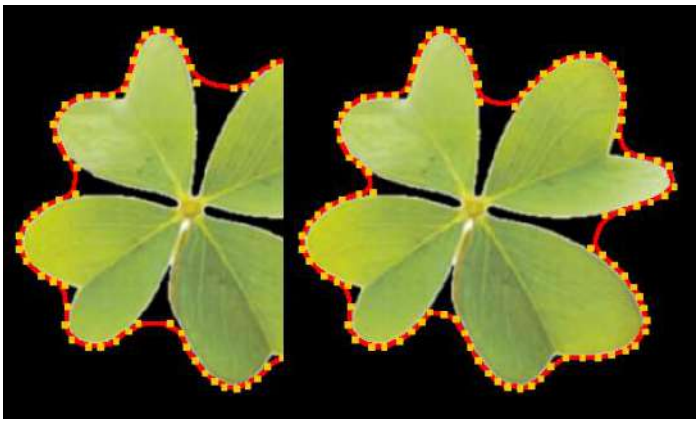


Figure III.3 : illustrations des grandes étapes du déroulement d'un snake

2.6. *Faiblesses des snakes, cas de non convergence totale*

Le positionnement des paramètres des différents éléments des énergies reste un point très délicat à prévoir, surtout si on commence à tenir compte de beaucoup d'aspects de chaque énergie. De plus, ces valeurs sont liées à la forme à extraire. La forme va beaucoup jouer, notamment dans ses concavités où les points du snakes ne peuvent s'aventurer sans pénaliser leur énergie interne en augmentant leur courbure. Les pondérations s'obtiennent le mieux via un tuning après déjà plusieurs exécutions pour affiner les résultats ou le comportement fin du snake. Les contours actifs s'avèrent très pertinents pour isoler des formes convexes régulières et idéales. Pour l'extraction de formes comportant beaucoup de concavités franches, il faudra diminuer l'impact du non-respect de cette contrainte pour privilégier l'écartement entre les points ou même l'énergie externe. C'est ce qu'on peut remarquer sur cette image résultat où l'on aperçoit la divergence du snake à la base des feuilles du trèfle. Et comme on pouvait s'y attendre, le snake converge parfaitement sur les contours externes des feuilles, là où les contraintes de son énergie interne sont les mieux respectées.

Un autre aspect non négligeable mais non directement lié à la méthode des snake réside dans l'image à analyser. En effet, celle-ci doit être la plus pure possible pour effacer au maximum le bruit qui pourrait amener à un pic de gradient sur lequel le snake pourrait rester accroché. Une fois accroché, il n'aurait que son énergie interne pour s'en décrocher et continuer sa route. Voici un exemple d'image de travail nettoyée et mettant en évidence des données de guidage pour l'énergie externe : les lignes de fort gradient et le flot.

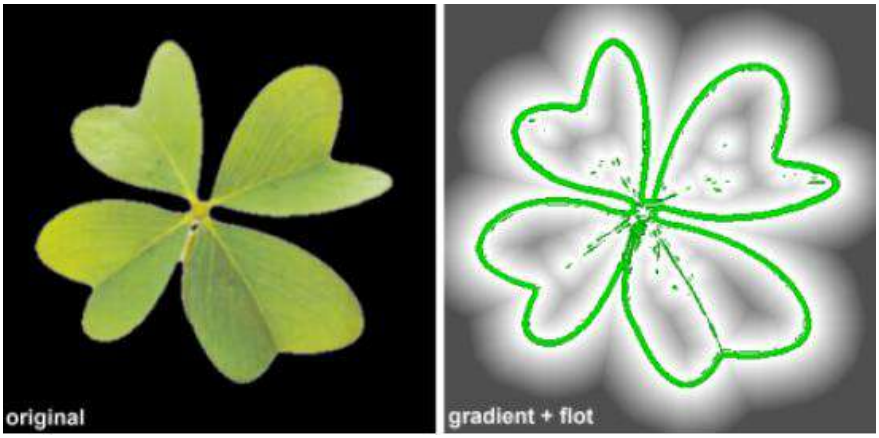


Figure III.4 : les lignes de fort gradient et le flot.

3. Mise en œuvre

L'adaptation du modèle mathématique à la simulation informatique passe en premier lieu par une discrétisation du modèle continu. Ensuite, il est nécessaire de trouver un algorithme qui simule le comportement du snake par minimisation de son énergie. Étant donné un snake de longueur *l*, il existe, dans une image fixe, une et une seule configuration géométrique du snake qui correspond à sa plus faible énergie. La minimisation de l'énergie du snake ne cherche pas à trouver exactement cette solution unique, d'abord parce que c'est un problème trop complexe, ensuite parce que l'intérêt porte sur un comportement dynamique du snake lié aux minimums locaux du système. En effet, que ce soit dans le cas d'une recherche de contours sur image fixe ou dans celui du suivi dans une séquence animée, la cible du snake a peu de chance de correspondre à l'unique solution minimale. La recherche est donc ramenée à une zone réduite, autour de la position précédente du snake, et le recours à un algorithme de plus haut niveau (ou même à l'utilisateur) est nécessaire pour initialiser la position du snake "près" du centre d'intérêt (contour à déterminer, ou cible à suivre). La minimisation tente alors de réduire l'énergie globale du contour par déplacements successifs de ses points.

Dans les sections qui vont suivre, nous définissons un modèle discret du snake, puis nous présentons une technique de minimisation. [12]

3.1. *Discrétisation*

Le principe de la discrétisation que nous présentons ici est tiré de [9], mais il a été repris par l'ensemble des publications sur le sujet. Rappelons que le snake est modélisé par une courbe paramétrée v(s)=(x(s), y(s)) ou $s \in [0,1]$ snake s'obtient naturellement en construisant un

vecteur u regroupant n nœuds du snake. Ces nœuds sont les points v(ih) = (x(ih), y(ih)) avec$\left(i = 0 \dots (n-1) \wedge h = \frac{1}{(n-1)}\right)$.

Dans la suite de ce paragraphe, $u_i = (x_i, y_i)$ représenté la position de i^{eme} nœud de snake.

L'énergie interne d'un nœud du snake, représentée dans sa version continue par l'équation (III.4) se présente en discret sous la forme :

$$\boldsymbol{E_{Interne} = \frac{1}{2}\left(\alpha(i) \cdot \left|\frac{u_i - u_{i-1}}{h}\right|^2 + \beta(i) \cdot \left|\frac{u_{i-1} - 2u_i + u_{+i+1}}{h^2}\right|^2\right)} \quad \textbf{(III.9)}$$

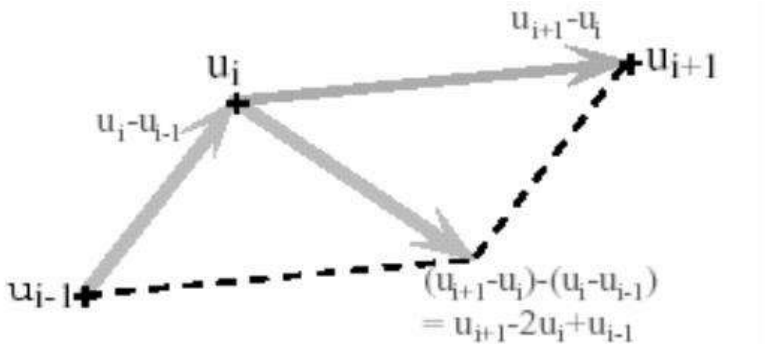


Figure III.5 : Vecteurs $(u_i - u_{i-1}\ et\ u_{i+1} - u_i)$ et variation $(u_{i-1} - 2u_i + u_{i+1})$

$(u_i - u_{i-1})$ représente le vecteur entre les nœuds $(i-1)$ et (i). On supprime un calcul de racine dans l'évaluation de l'énergie d'élongation en faisant intervenir le carré de la norme de ce vecteur qui est calculé ainsi :

$$\boldsymbol{|u_i - u_{i-1}|^2 = (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad \textbf{(III.10)}$$

La courbure du snake est représentée par le vecteur $(u_{i-1} - 2u_i + u_{i+1})$: c'est la déférence (variation) entre le vecteur $\overrightarrow{u_{i-1}\ u_i}$ et $\overrightarrow{u_i\ u_{i+1}}$ (voir figure III.5).

$$\boldsymbol{|u_i - u_{i-1} + u_{i+1}|^2 = (x_{i-1} - 2x_i + x_{i+1})^2 + (y_{i-1} - 2y_i + y_{i+1})^2} \quad \textbf{(III.11)}$$

En ce qui concerne l'énergie externe, la discrétisation est immédiate puisque le potentiel est calculé à partir de l'image qui est déjà disponible sous forme discrète (I (i, j)) et non pas sous forme continue (I (x, y)). On utilise par exemple :

$$\boldsymbol{E\ Externe\ (i) = E\ Contour\ (x\ i, y\ i)} \quad \textbf{(III.12)}$$

L'énergie globale du système peut finalement être présentée sous la forme :

$$\boldsymbol{E_{Global} = \sum_{i=0}^{n-1}(E_{Interne}(i) + E_{Externe}(i))} \quad \textbf{(III.13)}$$

La représentation discrète du snake et de son modèle d'énergie étant définie, on s'attache maintenant à trouver une méthode pour minimiser cette énergie.

3.2. *Minimisation de l'énergie par algorithme dynamique*

Dans cet algorithme, la minimisation se fait tour à tour sur chaque nœud du snake, indépendamment de celle des autres. Soit (u_i), le nœud considéré. Son énergie est donnée par la formule (III.13). Cette opération est effectuée pour toutes les positions du voisinage de (u_i).

Williams utilise un voisinage de neuf points (Figure III.6). La position correspondant à l'énergie minimale est retenue comme nouvelle position de (u_i), et on passe au traitement du nœud suivant [10]. La minimisation globale se termine lorsqu'après un cycle de traitement sur tous les nœuds, le nombre de nœuds ayant bougé est inférieur à un certain seuil. Cette démarche est résumée par le pseudo-code suivant :

```
# n = nombre de nœuds
# déplacés = nombre de points déplacés au cours d'un cycle
# m = nombre de voisins
# V[0..m-1] = voisins
# Boucle de minimisation globale
Répéter
        déplacés = 0;
        # Cycle de minimisation (traitement de tous les nœuds
        # un par un
        Pour (i = 0 à n) # Le nœud 0 est traité en premier et dernier
                Emin = MAXIMUM
                V = les voisins de ui
                # Boucle sur le voisinage
                Pour (j = 0 à m-1)
                        Ej = EInterne(V[j]) + EExterne (V[j])
                        Si (Ej < Emin)
                                Emin = Ej
                                jmin = j
                Si (V[jmin] différent de la position courante)
                                Déplacer nœud courant vers V[jmin]
                                déplacés = déplacés + 1
Jusqu'à (déplacés < seuil)
```

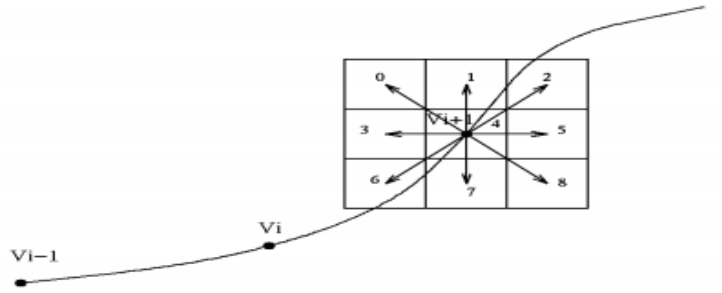
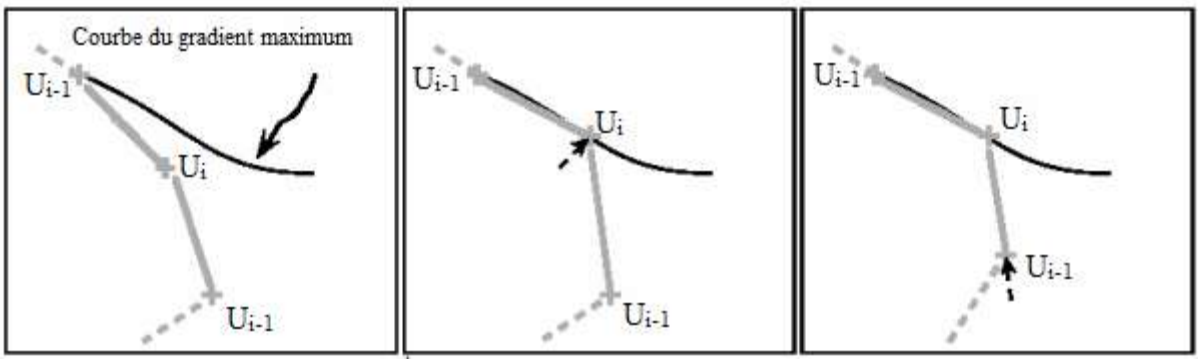


Figure III.6 : Le voisinage de neuf points.

La minimisation globale du système est obtenue au fur et à mesure des minimisations en chacun des nœuds du snake : par exemple un nœud peut se déplacer pour atteindre un pic du gradient sans prendre en compte l'augmentation d'énergie que cela provoque pour son voisin du fait de l'élongation du snake (figure.III.7.b). Au cycle suivant, le voisin se rapproche afin de ramener son énergie interne au niveau minimal (figure.III.7.c). Le bilan des deux cycles est une diminution de l'énergie globale du snake car un nœud a minimisé son énergie externe.



(a) (b) (c)

Figure III.7 : Minimisation par algorithme dynamique (a) état initial. (b) le nœud i diminue son énergie externe, l'énergie interne du nœud i+1 augmente.

(c) le nœud i+1 réduit son énergie interne.

4. Conclusion

Dans ce présent travail, nous nous sommes intéressés aux contours actifs qui consistent à initialiser une courbe et la faire évoluer de façon à minimiser l'énergie.

Cette approche consiste à initialiser un contour par une suite de noeuds soumis à des forces externes qui représentent les caractéristiques de l'image et à des forces internes d'élasticité et de rigidité qui assurent une cohérence pendant l'évolution du contour initial. Cette évolution permettra au contour de trouver un état d'équilibre tout en minimisant ces énergies.

Chapitre IV

Implémentation et discussion des résultats

1. Introduction

Ce chapitre présent les résultats obtenus, on va essayer de voir une présentation générale d'un algorithme Pour la détection de contour dans une image.

Pour mettre en place leur utilisation une méthode de résolution a été implantée pour le problème posé.

Le programme Développé permet la détection des Contours. Il permet aussi de générer des images binaires. Le programme est disponible dans l'annexe**.**

2. Conception

L'ensemble des nœuds joue un rôle important dans la déformation du contour. Localement, des portions de contours vont se contracter ou s'étirer pour s'adapter à la forme de l'objet sur la tranche courante. Cette déformation va entraîner des discontinuités au niveau de contour.

Lorsque le seuillage ne donne pas entière satisfaction, il faut utiliser l'information sur les arêtes. Une arête est définie comme une discontinuité abrupte entre deux régions ayant des intensités différentes mais uniformes. Il existe différents algorithmes de détection d'arêtes. Les plus simples et les plus rapides sont les opérateurs de gradient d'image.

Dans un processus itératif, le critère d'arrêt mérite une attention particulière. La méthode la plus simple consiste à proposer un nombre fixe d'itérations. Le temps de calcul étant un facteur à considérer, on ne peut se permettre un trop grand nombre de d'itérations inutiles.

L'énergie globale est décomposée en une somme d'énergies qui sont associées à tout point du contour courant qui évolue au cours des itérations dans le processus de convergence numérique. En chaque point M, une valeur d'énergie E peut être calculée. Elle s'exprime comme une somme pondérée d'énergies de différentes sortes qui interviennent dans le modèle du contour.

Les étapes de l'algorithme sont illustrées dans l'organigramme suivant **:**

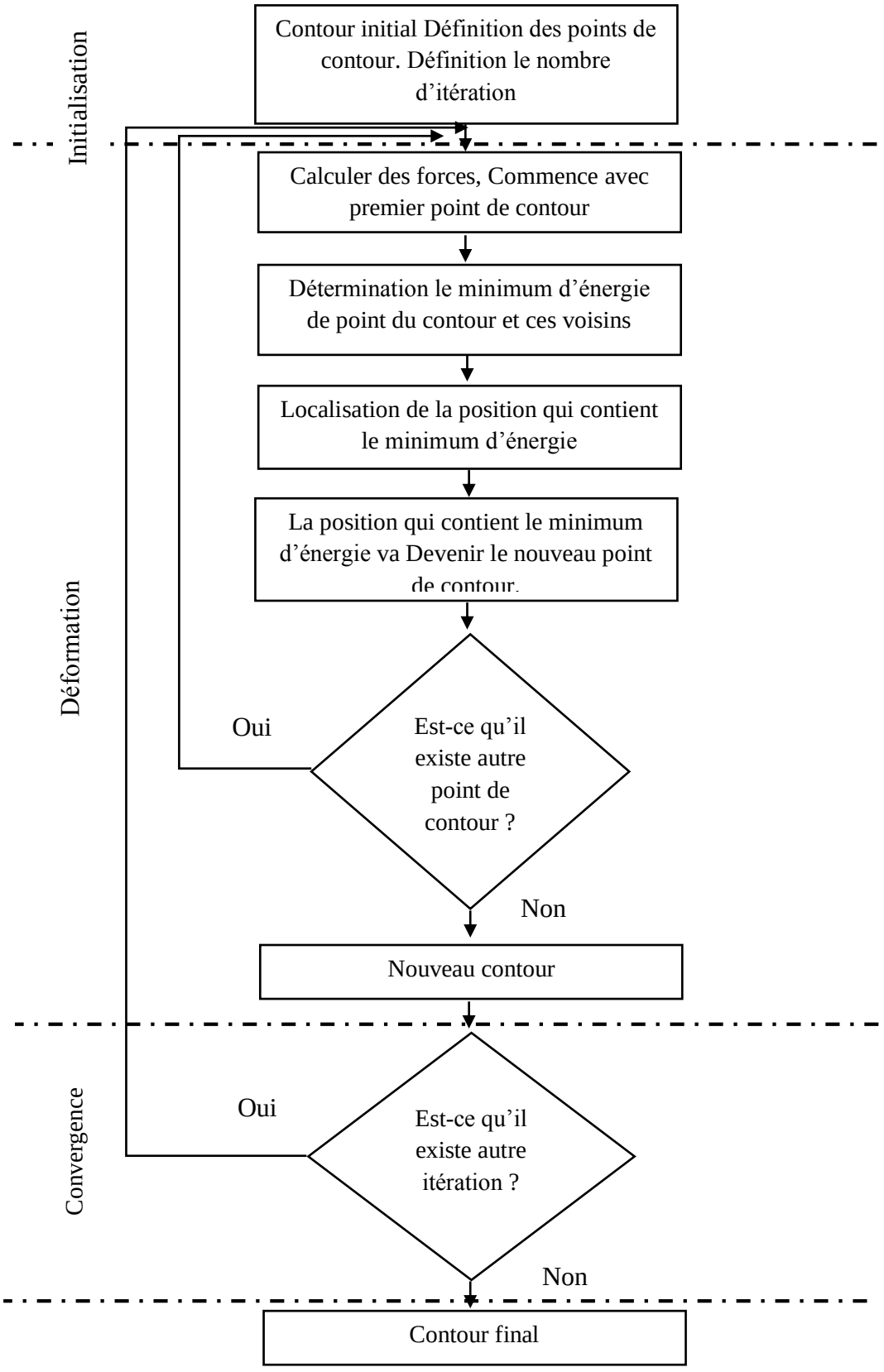
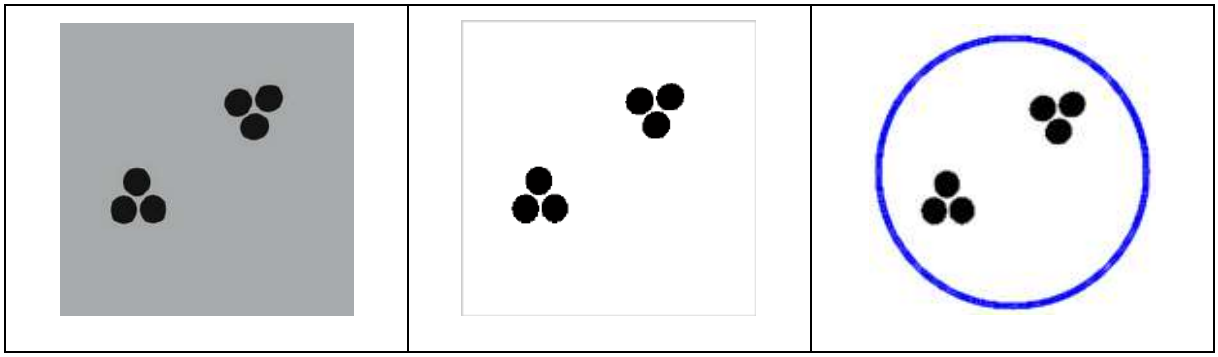


Figure IV .1: principe de contour actif.

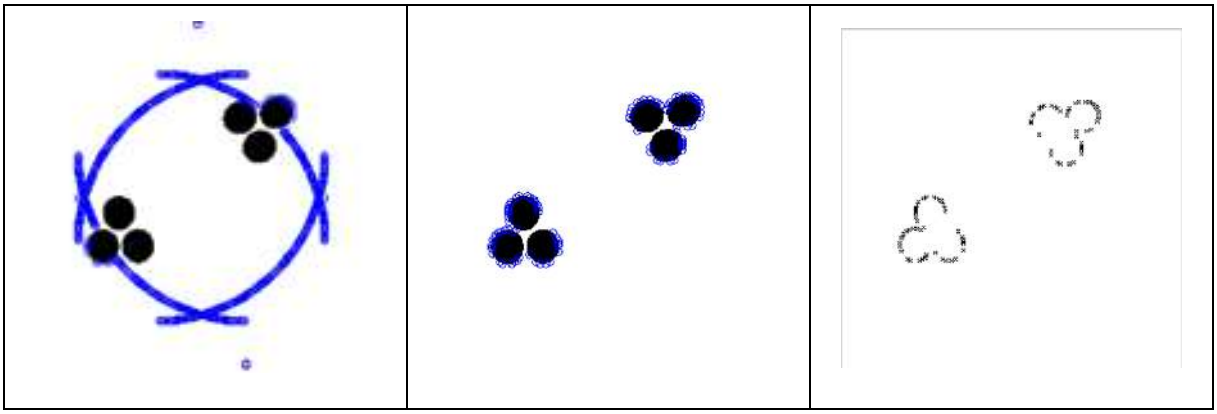
3. Résultats et interprétations

Nous avons choisi d'illustrer la méthode que nous venons de proposer à l'aide de quatre images, Après avoir proposé des illustrations mettant en évidence la qualité des résultats qui peuvent être obtenus sans aucune phase d'essai préalable pour adapter les forces à l'image, Nous examinerons aussi l'adaptabilité aux différentes formes, et on bien remarquer les etapes de déformation du contour. [11]

3.1. *Résultats*



(a) image original (b) L'opération de seuillage (c) Initialisation de contour

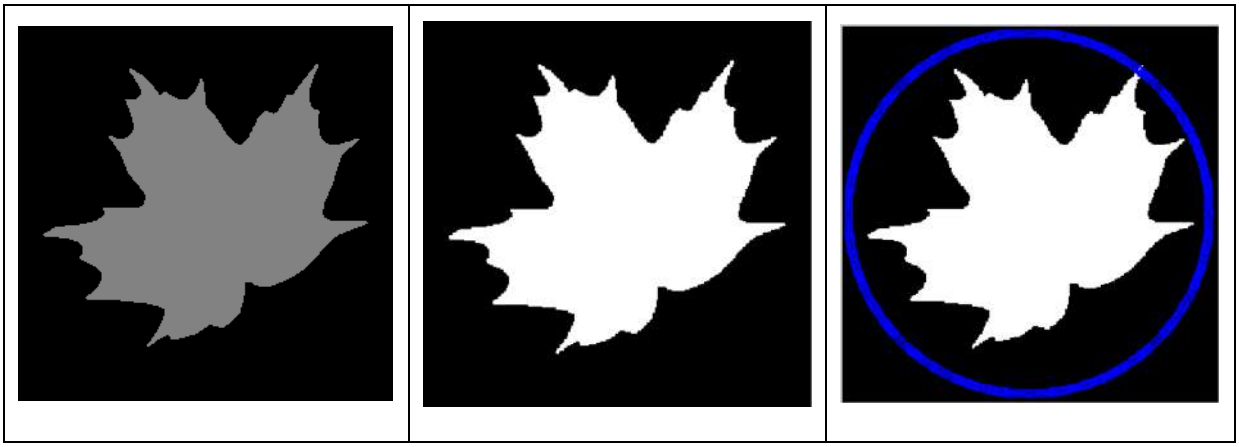


(d) Après 40 itérations (e) Après 236 itérations (f) Extraction et dessin de Contour final

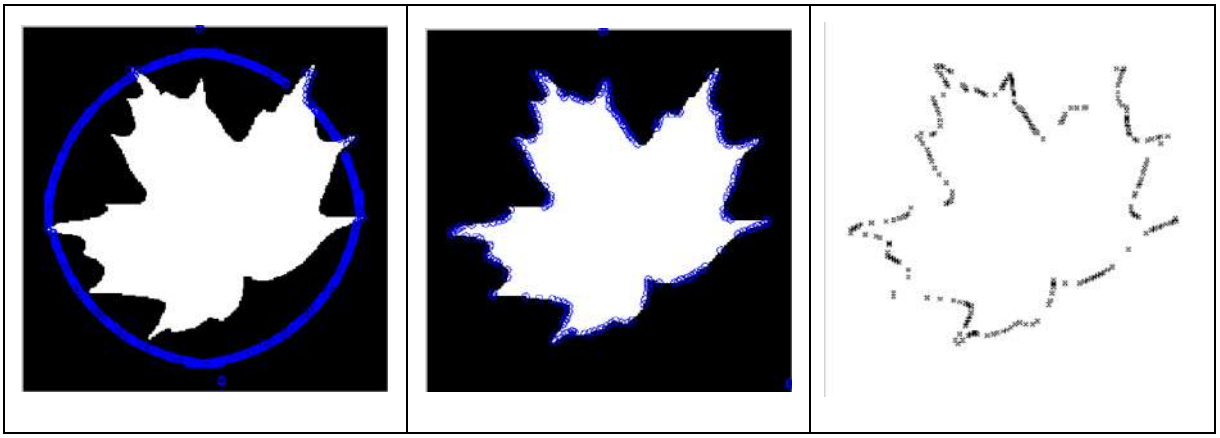
Figure IV.2. Contour actif (image de synthèse).

L'initialisation est faite par un cercle comme dans la Figure IV.1.c, les résultats présentés sur les Figures IV.1.d, IV.1.e sont obtenus pendant l'évolution du contour, la Figure VI.1.f représente le résultat après 236 itérations.

Dans ce que suit, comme dans la section précédente, nous présenterons l'initialisation, la binarisation, et deux étapes intermédiaires puis le résultat final.

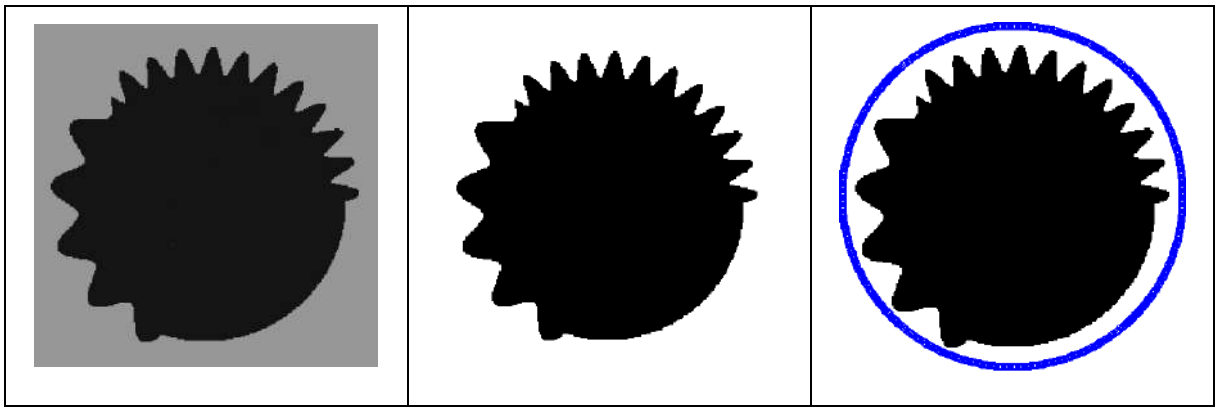


(a) image original (b) L'opération de seuillage (c) Initialisation de contour

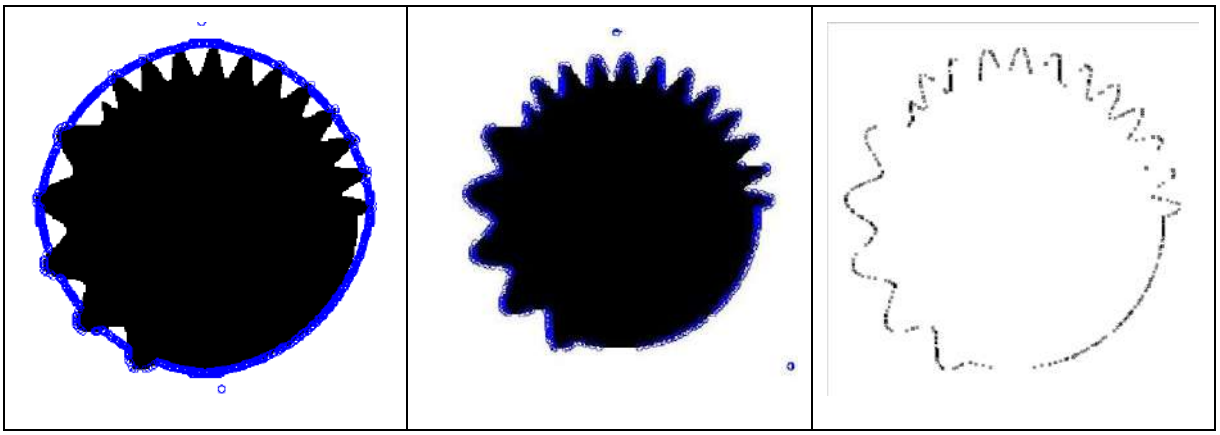


(d) Après 15 itérations (e) Après 236 itérations (f) Extraction et dessin de Contour final

Figure IV.3. Test (image 1).

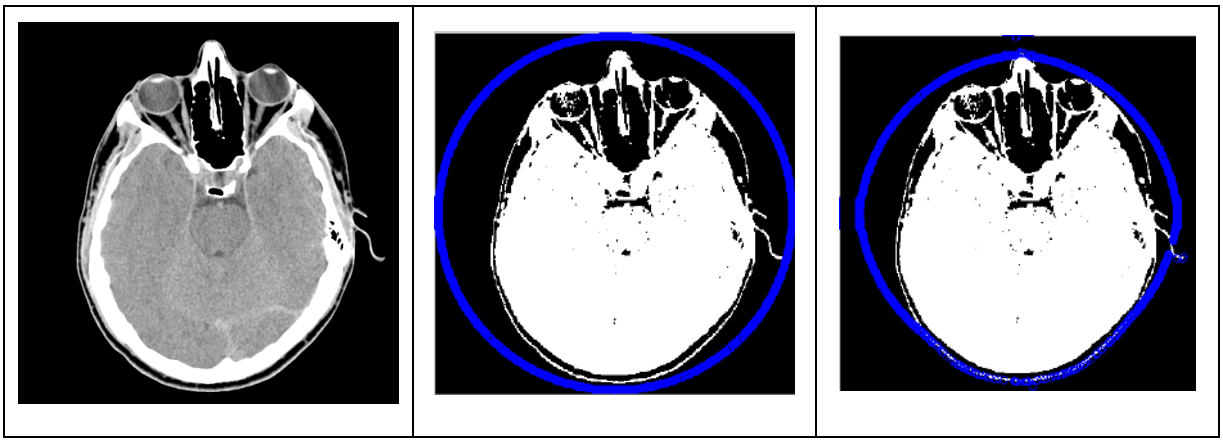


(a) image original (b) L'opération de seuillage (c) Initialisation de contour

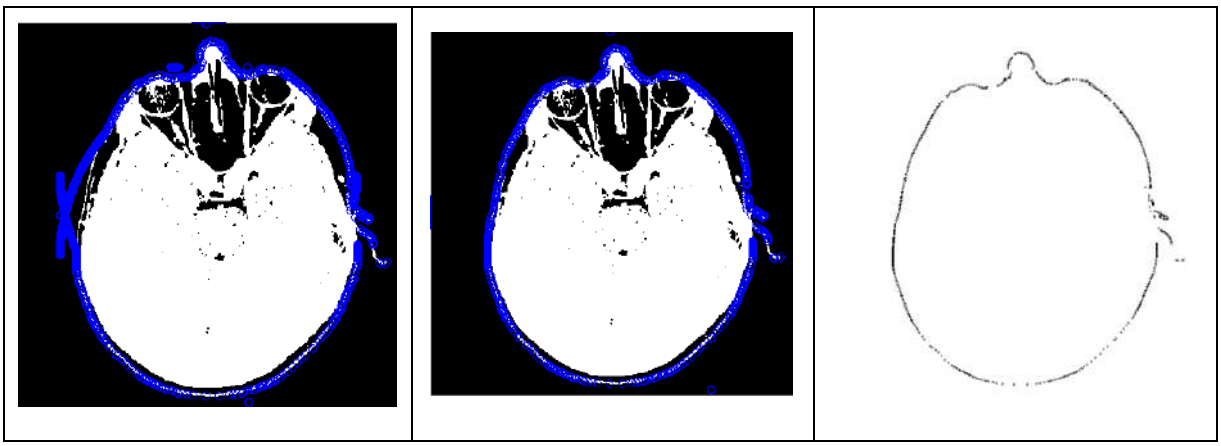


(d) Après 15 itérations (e) Après 336 itérations (f) Extraction et dessin de Contour final

Figure IV.4. Test (image 2).

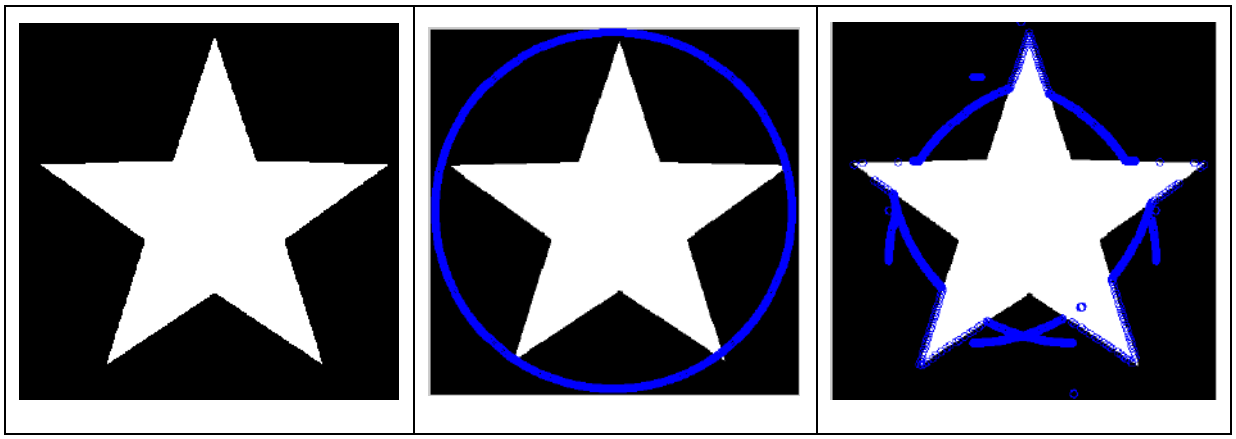


(a) image original (b) Initialisation de contour (c) Après 15 itérations

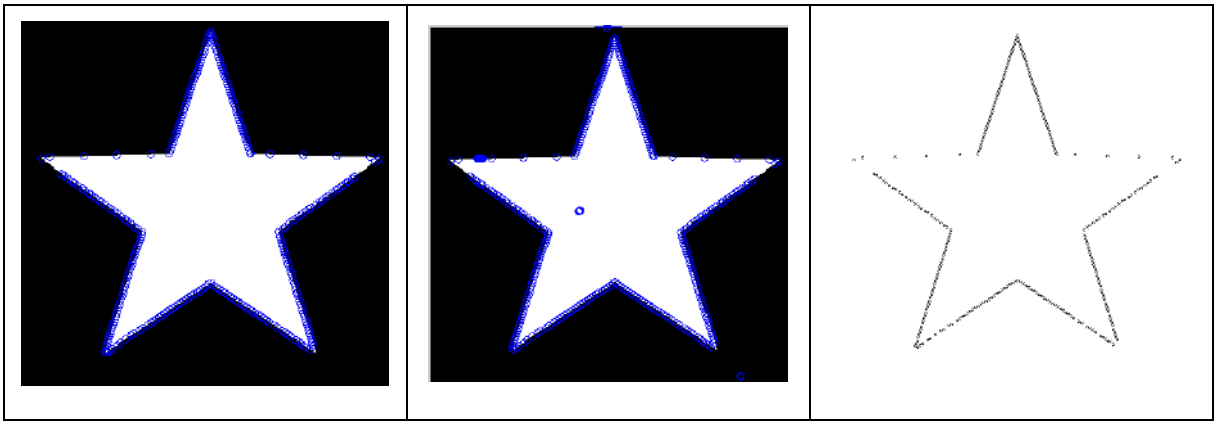


(d) Après 60 itérations (e) Après 492 itérations (f) Extraction et dessin de Contour final

Figure IV.5. Test (image 3).



(a) image original (b) Initialisation de contour (c) Après 50 itérations



(d) Après 140 itérations (e) Après 381 itérations (f) Extraction et dessin de Contour final

Figure IV.6. Test (image 4).

3.2. *Discussion*

Nous avons présenté le contour actif. Il est basé sur un algorithme, après l'avoir justifié théoriquement par une modélisation mathématique, nous avons présenté l'algorithme puis des résultats sur des images. Il est remarquable d'obtenir des résultats aussi satisfaisants sur les différentes images de test, et un temps de calcul évidemment long.

4. Conclusion

La méthode des contours actifs permet d'obtenir des résultats remarquables même dans les cas les plus difficiles. Nous avons obtenir des résultats acceptable donc on peut dire que la qualité des résultats dépend fortement du choix des nombreux paramètres qui interviennent dans la fonctionnelle d'énergie qui est à minimiser par le contour et le nombre de point de contour.

Conclusion générale

Ce projet nous a permis d'approfondir nos connaissances dans le domaine du traitement d'image et plus précisément dans le domaine de segmentation en contours d'image par la méthode de contour actif.

Nous avons commencé par un rappel sur le traitement d'image en général, puis on a parlé sur la segmentation par les méthodes classiques à savoir l'approche contour et l'approche région. Les contours actifs ou snakes, faisant l'objet de recherche depuis plus de deux décennies, et présentant des variantes dérivant des approches classiques ont été aussi présentées, Intéressés plus particulièrement par l'approche contour.

Nous avons implémenté un algorithme qui base sur la minimisation d'énergie des images, cet algorithme est classé parmi les meilleurs approches de minimisation, vue sa rapidité, sa précision et ainsi sa possibilité d'introduire des énergies de contexte.

Par la suite ; ce programme peut être utilise dans plusieurs domaine d'applications tel que l'imagerie médicale, la détection des objets et des obstacles, suivie des cibles…

Ce qui concerne la partie programmation, elle nous a permis d'apprendre et de comprendre de plus en plus les possibilités données en utilisant le logiciel de calcule Matlab.

En fin nous espérons que le programme que nous avons fait servira les étudiants des promotions futures et leur permettra d'atteindre des résultats plus performants dans ce domaine.

Bibliographie

[1] j.P.cocquerez, S.Philpe :« Analyse d'images filtrage et segmentation ».Edition, Masson Paris 1995.

[2] Rafik Berrou : « Poursuite Vidéo D'objets En Mouvement ».Mémoire D'ingénieur en Informatique, publication de l'école militaire polytechnique, 2001.

[3] salima nebti, «optimisation par écosystèmes artificiels : application à la segmentation d'images». Mémoire magistère Université Mentouri de Constantine, 2007.

[4] Hardancourt : « fou de Multimédia », Edition, Sybex, 1995.

[5] Patrick Bonnin : « Méthode Systématique de Conception et de Réalisation D'application en Vision Par Ordinateur »Thèse de Doctorat de L'Université de Paris VII. 1991.

[6] Laloui Derafa : « Etudes et Réalisation D'un dispositif de Poursuite Vidéo Application à L'orientation D'une Structure Deux Axes » publication de l'école militaire polytechnique ,2001.

[7] John Wiley & Sons : « Image Processing Principales and Applications » Department of Electrical Engineering Arizona State University Tempe, Arizona Copyright 2005.

[8] Sofiane hammouda « conception et réalisation d'un système de détection de mouvement et de poursuite video a l'aide d'une caméra motorisée. » publication de l'école militaire polytechnique, 2002.

[9] Kass, A. Witkin, D. Terzopoulos. "Snakes : Active Contour Modèls". Proc. 1 International Conf. on Computer Vision, 1987.

[10] D. J. Williams, M. Shah. "A Fast Algorithm for Active Contours". Proc. 1st European Conf. on Computer Vision, 1990.

[11] Jean-Jacques ROUSSELLE « les contours actifs, une méthode de segmentation Application à l'imagerie médicale » Université François Rabelais de Tours 2003.

[12] Méthode des contours actifs, Pierre Schwartz, 2006.

http://www.creatis.insalyon.fr/~bernard/AnalyseImage/sources/Detection_Contours/Snake/SNAKE.pdf

Annexe

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%Algorithme pour la détection de contour%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
close all
clc
nm=input('donner le nom de l''image d''entree /////////   ');
I=imread(nm);[z,q,c]=size (I);%%%la  Lecteur de l'image(z:les lignes,q:les
colonnes,c:nombre des matrice)
if c==3%%%Si l'image est en couleur on va le transféré en niveaux de gris
I=rgb2gray(I);end
M=max(max(I));%La valeur max de l'image
m=min(min(I));%La valeur min de l'image
se=(M+m)/2;%    le choix de Seuil
%%%%%%%%%%%%   fair la binarisation
for i=1:z
    for j=1:q
        if I(i,j)<se
            I(i,j)=0;
        else
            I(i,j)=255;end
    end
end
subplot(1,2,1);hold on;xlabel('Image original');imshow(I);pause
n=(2*(z+q));%%%%%%%%%% choix de nomber de point du contour
[x,y,xc,yc]=cont(n,I);%%% contour initial
plot(x,y,'o');II=double(I);subplot(1,2,2);title('Energie')
%%%%%%%%%%%%%%%%%%%%%%%%%%calcul d'energie
enrex=ExternalForceImage2D(II,0.1,0.1,0.1,0.1) ;mesh(1:q,1:z,enrex)
ite=(min(z,q)-20);%%choix de nomber d'itération
%%%%localisation de direction
  for j=1:n
        if (y(j)<yc)&(x(j)<xc) %Détermination la direction des points
            %vx: un vecteur contient les cordonnes x de point voisin,
            %vy: un vecteur contient les cordonnes y de point voisin,
            %dv: un vecteur associe au directions des points voisins avec les
% nœuds du contour
            vx(1)=x(j)  ;   vy(1)=y(j);    dv(1)=1;
            vx(2)=x(j)+1;   vy(2)=y(j)-1;  dv(2)=2;
            vx(3)=x(j)+1;   vy(3)=y(j);    dv(3)=7;
            vx(4)=x(j)-1;   vy(4)=y(j)+1;  dv(4)=3;
            vx(5)=x(j)  ;   vy(5)=y(j)+1;  dv(5)=5;
            vx(6)=x(j)+1;   vy(6)=y(j)+1;  dv(6)=1;
   %%%Détermination le minimum d'énergie de point du
   %contour et ces voisins
            for kl=1:6
            ev(kl)=enrex(vy(kl),vx(kl));end
            %%%La position qui contient le minimum
            %d'énergie va Devenir le
            %%%nouveau point de contour.
            mev=min(ev);
            for kl=1:6
                if ev(kl)==mev
                    pm=kl;end
            end
             d(j)=dv(pm);x(j)=vx(pm);y(j)=vy(pm);
        end
       if (y(j)>yc)&(x(j)<xc)%2
```

```
      vx(1)=x(j)   ;    vy(1)=y(j)   ;    dv(1)=2;
      vx(2)=x(j)+1;    vy(2)=y(j)   ;    dv(2)=7;
      vx(3)=x(j)+1;    vy(3)=y(j)+1;    dv(3)=1;
      vx(4)=x(j)   ;    vy(4)=y(j)+1;    dv(4)=5;
      vx(5)=x(j)-1;    vy(5)=y(j)+1;    dv(5)=3;
      vx(6)=x(j)+1;    vy(6)=y(j)-1;    dv(6)=2;
      for kl=1:6
      ev(kl)=enrex(vy(kl),vx(kl));end
      mev=min(ev);
      for kl=1:6
          if ev(kl)==mev
              pm=kl;end
      end
       d(j)=dv(pm);x(j)=vx(pm);y(j)=vy(pm);
end
if (y(j)<yc)&(x(j)>xc)%3
      vx(1)=x(j)   ;    vy(1)=y(j)   ;    dv(1)=3;
      vx(2)=x(j)-1;    vy(2)=y(j)-1 ;   dv(2)=4;
      vx(3)=x(j)-1;    vy(3)=y(j);      dv(3)=6;
      vx(4)=x(j)   ;    vy(4)=y(j)+1;    dv(4)=5;
      vx(5)=x(j)+1;    vy(5)=y(j)+1;    dv(5)=1;
      vx(6)=x(j)-1;    vy(6)=y(j)+1;    dv(6)=3;
      for kl=1:6
      ev(kl)=enrex(vy(kl),vx(kl));
      end
      mev=min(ev);
      for kl=1:6
          if ev(kl)==mev
              pm=kl;end
      end
       d(j)=dv(pm);x(j)=vx(pm);y(j)=vy(pm);
end
if (y(j)>yc)&(x(j)>xc)%4
      vx(1)=x(j)   ;    vy(1)=y(j)   ;    dv(1)=4;
      vx(2)=x(j)+1;    vy(2)=y(j)-1 ;   dv(2)=2;
      vx(3)=x(j);      vy(3)=y(j)-1;    dv(3)=8;
      vx(4)=x(j)-1   ; vy(4)=y(j)   ;    dv(4)=6;
      vx(5)=x(j)-1;    vy(5)=y(j)+1;    dv(5)=3;
      vx(6)=x(j)-1;    vy(6)=y(j)-1;    dv(6)=4;
      for kl=1:6
      ev(kl)=enrex(vy(kl),vx(kl));
      end
      mev=min(ev);
      for kl=1:6
          if ev(kl)==mev
              pm=kl;end
      end
       d(j)=dv(pm);x(j)=vx(pm);y(j)=vy(pm);
end
if (y(j)<yc)&(x(j)==xc) %5
      vx(1)=x(j)     ;    vy(1)=y(j)     ;    dv(1)=5;
      vx(2)=x(j)     ;    vy(2)=y(j)+1   ;    dv(2)=2;
      vx(3)=x(j)+1   ;    vy(3)=y(j)+1   ;    dv(3)=8;
      vx(4)=x(j)+1   ;    vy(4)=y(j) -1  ;    dv(4)=6;
      vx(5)=x(j)     ;    vy(5)=y(j)-1   ;    dv(5)=3;
      vx(6)=x(j)+1   ;    vy(6)=y(j)     ;    dv(6)=4;
      for kl=1:6
      ev(kl)=enrex(vy(kl),vx(kl));
      end
      mev=min(ev);
      for kl=1:6
          if ev(kl)==mev
              pm=kl;end
```

```
            end
             d(j)=dv(pm);x(j)=vx(pm);y(j)=vy(pm);
        end
        if (y(j)==yc)&(x(j)>xc) %6
            vx(1)=x(j)    ;   vy(1)=y(j)    ;   dv(1)=6;
            vx(2)=x(j)    ;   vy(2)=y(j)-1  ;   dv(2)=8;
            vx(3)=x(j)-1  ;   vy(3)=y(j)-1  ;   dv(3)=4;
            vx(4)=x(j)-1  ;   vy(4)=y(j)+1  ;   dv(4)=3;
            vx(5)=x(j)    ;   vy(5)=y(j)+1  ;   dv(5)=5;
            vx(6)=x(j)-1  ;   vy(6)=y(j)    ;   dv(6)=6;
            for kl=1:6
            ev(kl)=enrex(vy(kl),vx(kl));
            end
            mev=min(ev);
            for kl=1:6
                if ev(kl)==mev
                    pm=kl;end
            end
             d(j)=dv(pm);x(j)=vx(pm);y(j)=vy(pm);
        end
        if (y(j)==yc)&(x(j)<xc) %7
            vx(1)=x(j)    ;   vy(1)=y(j)    ;   dv(1)=7;
            vx(2)=x(j)    ;   vy(2)=y(j)-1  ;   dv(2)=8;
            vx(3)=x(j)+1  ;   vy(3)=y(j)-1  ;   dv(3)=2;
            vx(4)=x(j)+1  ;   vy(4)=y(j)+1  ;   dv(4)=1;
            vx(5)=x(j)    ;   vy(5)=y(j)+1  ;   dv(5)=5;
            vx(6)=x(j)+1  ;   vy(6)=y(j)    ;   dv(6)=7;
            for kl=1:6
            ev(kl)=enrex(vy(kl),vx(kl));
            end
            mev=min(ev);
            for kl=1:6
                if ev(kl)==mev
                    pm=kl;end
            end
             d(j)=dv(pm);x(j)=vx(pm);y(j)=vy(pm);
        end
        if (y(j)>yc)&(x(j)==xc) %8
            vx(1)=x(j)    ;   vy(1)=y(j)    ;   dv(1)=8;
            vx(2)=x(j)+1  ;   vy(2)=y(j)    ;   dv(2)=8;
            vx(3)=x(j)+1  ;   vy(3)=y(j)-1  ;   dv(3)=2;
            vx(4)=x(j)-1  ;   vy(4)=y(j)    ;   dv(4)=1;
            vx(5)=x(j)-1  ;   vy(5)=y(j)-1  ;   dv(5)=5;
            vx(6)=x(j)    ;   vy(6)=y(j)-1  ;   dv(6)=7;
            for kl=1:6
            ev(kl)=enrex(vy(kl),vx(kl));
            end
            mev=min(ev);
            for kl=1:6
                if ev(kl)==mev
                    pm=kl;end
            end
             d(j)=dv(pm);x(j)=vx(pm);y(j)=vy(pm);
        end

  end
pause(0.001)%Espace de temps entre les plots
subplot(1,2,1);imshow(I);plot (x,y,' o');xlabel('Image binaire + contour initial')
for i=2:ite
    for j=1:n
     if d(j)==1
         if x(j)<q & y(j)<z & x(j)>1 & y(j)>1
            vx(1)=x(j)  ;   vy(1)=y(j);    dv(1)=1;
```

```
        vx(2)=x(j)+1;   vy(2)=y(j)-1;  dv(2)=2;
        vx(3)=x(j)+1;   vy(3)=y(j);    dv(3)=7;
        vx(4)=x(j)-1;   vy(4)=y(j)+1;  dv(4)=3;
        vx(5)=x(j)  ;   vy(5)=y(j)+1;  dv(5)=5;
        vx(6)=x(j)+1;   vy(6)=y(j)+1;  dv(6)=1;
        for kl=1:6
        ev(kl)=enrex(vy(kl),vx(kl));
        end
        mev=min(ev);
        for kl=1:6
            if ev(kl)==mev
                pm=kl;end
        end
         d(j)=dv(pm);x(j)=vx(pm);y(j)=vy(pm);
     end
end
if d(j)==2
    if x(j)<q & y(j)<z & x(j)>1 & y(j)>1
        vx(1)=x(j)  ;   vy(1)=y(j)  ;   dv(1)=2;
        vx(2)=x(j)+1;   vy(2)=y(j)  ;   dv(2)=7;
        vx(3)=x(j)+1;   vy(3)=y(j)+1;   dv(3)=1;
        vx(4)=x(j)  ;   vy(4)=y(j)+1;   dv(4)=5;
        vx(5)=x(j)-1;   vy(5)=y(j)+1;   dv(5)=3;
        vx(6)=x(j)+1;   vy(6)=y(j)-1;   dv(6)=2;
        for kl=1:6
        ev(kl)=enrex(vy(kl),vx(kl));
        end
        mev=min(ev);
        for kl=1:6
            if ev(kl)==mev
                pm=kl;end
        end
         d(j)=dv(pm);x(j)=vx(pm);y(j)=vy(pm);
    end
end
if d(j)==3
    if x(j)<q & y(j)<z & x(j)>1 & y(j)>1
        vx(1)=x(j)  ;   vy(1)=y(j)  ;   dv(1)=3;
        vx(2)=x(j)-1;   vy(2)=y(j)-1 ;  dv(2)=4;
        vx(3)=x(j)-1;   vy(3)=y(j);     dv(3)=6;
        vx(4)=x(j)  ;   vy(4)=y(j)+1;   dv(4)=5;
        vx(5)=x(j)+1;   vy(5)=y(j)+1;   dv(5)=1;
        vx(6)=x(j)-1;   vy(6)=y(j)+1;   dv(6)=3;
        for kl=1:6
        ev(kl)=enrex(vy(kl),vx(kl));
        end
        mev=min(ev);
        for kl=1:6
            if ev(kl)==mev
                pm=kl;end
        end
         d(j)=dv(pm);x(j)=vx(pm);y(j)=vy(pm);
    end
end
if d(j)==4
    if x(j)<q & y(j)<z & x(j)>1 & y(j)>1
        vx(1)=x(j)  ;   vy(1)=y(j)  ;   dv(1)=4;
        vx(2)=x(j)+1;   vy(2)=y(j)-1 ;  dv(2)=2;
        vx(3)=x(j);     vy(3)=y(j)-1;   dv(3)=8;
        vx(4)=x(j)-1  ; vy(4)=y(j)  ;   dv(4)=6;
        vx(5)=x(j)-1;   vy(5)=y(j)+1;   dv(5)=3;
        vx(6)=x(j)-1;   vy(6)=y(j)-1;   dv(6)=4;
        for kl=1:6
```

```
        ev(kl)=enrex(vy(kl),vx(kl));
        end
        mev=min(ev);
        for kl=1:6
            if ev(kl)==mev
                pm=kl;end
        end
         d(j)=dv(pm);x(j)=vx(pm);y(j)=vy(pm);
    end
end
if d(j)==5
    if x(j)<q & y(j)<z & x(j)>1 & y(j)>1
        vx(1)=x(j)    ;   vy(1)=y(j)    ;   dv(1)=5;
        vx(2)=x(j)    ;   vy(2)=y(j)+1  ;   dv(2)=2;
        vx(3)=x(j)+1  ;   vy(3)=y(j)+1  ;   dv(3)=8;
        vx(4)=x(j)+1  ;   vy(4)=y(j) -1 ;   dv(4)=6;
        vx(5)=x(j)    ;   vy(5)=y(j)-1  ;   dv(5)=3;
        vx(6)=x(j)+1  ;   vy(6)=y(j)    ;   dv(6)=4;
        for kl=1:6
        ev(kl)=enrex(vy(kl),vx(kl));
        end
        mev=min(ev);
        for kl=1:6
            if ev(kl)==mev
                pm=kl;end
        end
         d(j)=dv(pm);x(j)=vx(pm);y(j)=vy(pm);
    end
end
if d(j)==6
    if x(j)<q & y(j)<z & x(j)>1 & y(j)>1
        vx(1)=x(j)    ;   vy(1)=y(j)    ;   dv(1)=6;
        vx(2)=x(j)    ;   vy(2)=y(j)-1  ;   dv(2)=8;
        vx(3)=x(j)-1  ;   vy(3)=y(j)-1  ;   dv(3)=4;
        vx(4)=x(j)-1  ;   vy(4)=y(j)+1  ;   dv(4)=3;
        vx(5)=x(j)    ;   vy(5)=y(j)+1  ;   dv(5)=5;
        vx(6)=x(j)-1  ;   vy(6)=y(j)    ;   dv(6)=6;
        for kl=1:6
        ev(kl)=enrex(vy(kl),vx(kl));
        end
        mev=min(ev);
        for kl=1:6
            if ev(kl)==mev
                pm=kl;end
        end
         d(j)=dv(pm);x(j)=vx(pm);y(j)=vy(pm);
    end
end
if d(j)==7
    if x(j)<q & y(j)<z & x(j)>1 & y(j)>1
        vx(1)=x(j)    ;   vy(1)=y(j)    ;   dv(1)=7;
        vx(2)=x(j)    ;   vy(2)=y(j)-1  ;   dv(2)=8;
        vx(3)=x(j)+1  ;   vy(3)=y(j)-1  ;   dv(3)=2;
        vx(4)=x(j)+1  ;   vy(4)=y(j)+1  ;   dv(4)=1;
        vx(5)=x(j)    ;   vy(5)=y(j)+1  ;   dv(5)=5;
        vx(6)=x(j)+1  ;   vy(6)=y(j)    ;   dv(6)=7;
        for kl=1:6
        ev(kl)=enrex(vy(kl),vx(kl));
        end
        mev=min(ev);
        for kl=1:6
            if ev(kl)==mev
                pm=kl;end
```

```
            end
             d(j)=dv(pm);x(j)=vx(pm);y(j)=vy(pm);
        end
    end
    if d(j)==8
        if x(j)<q & y(j)<z & x(j)>1 & y(j)>1
            vx(1)=x(j)    ;   vy(1)=y(j)    ;   dv(1)=8;
            vx(2)=x(j)+1  ;   vy(2)=y(j)    ;   dv(2)=8;
            vx(3)=x(j)+1  ;   vy(3)=y(j)-1  ;   dv(3)=2;
            vx(4)=x(j)-1  ;   vy(4)=y(j)    ;   dv(4)=1;
            vx(5)=x(j)-1  ;   vy(5)=y(j)-1  ;   dv(5)=5;
            vx(6)=x(j)    ;   vy(6)=y(j)-1  ;   dv(6)=7;
            for kl=1:6
            ev(kl)=enrex(vy(kl),vx(kl));
            end
            mev=min(ev);
            for kl=1:6
                if ev(kl)==mev
                    pm=kl;end
            end
             d(j)=dv(pm);x(j)=vx(pm);y(j)=vy(pm);
        end
    end
    end
pause(0.001);
subplot(1,2,1);imshow(I);plot (x,y',' o ');
title('ite  ''i=');title(i);xlabel('déformation de contour')
end
%%%%Suppression des points indésirables
    i=1;
    hh=length(x);
       while i<=hh
           if (x(i)==1)   | (y(i)==1) |(x(i)==q)   | (y(i)==z)
               x(i)=[];
               y(i)=[];
               hh=hh-1;
           else
               i=i+1;
           end
       end
subplot(1,2,1);imshow(I);plot (x,y','o')
%%%Affichage de points de contour
n=length(x);
imr(1:z,1:q)=255;
for i=1:n
        imr(y(i),x(i))=0  ;
        imr(y(i)-1,x(i)-1)=0;
        imr(y(i)+1,x(i)-1)=0;
        imr(y(i)+1,x(i)+1)=0;
        imr(y(i)-1,x(i)+1)=0;
end
pause;
figure (2);imshow(imr)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%les fonctions%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Eextern = ExternalForceImage2D(I,Wline, Wedge, Wterm,Sigma)
% les entrées
% I : image
% Sigma : Sigma utilisé pour calculer des dérivées d'images
% Wline : Activité de lignes, (si wligne est négatif il s'agit dune lignes noires,en
revanche ligne blanc.
% Wedge : cet paramètre propre au contour
% Wterm : utilisé pour les déterminer de lignes (points finaux) et les coins
% les sorties
% Eextern : La fonction de l'énergie
Ix=ImageDerivatives2D(I,Sigma,'x');
Iy=ImageDerivatives2D(I,Sigma,'y');
Ixx=ImageDerivatives2D(I,Sigma,'xx');
Ixy=ImageDerivatives2D(I,Sigma,'xy');
Iyy=ImageDerivatives2D(I,Sigma,'yy');
Eline = imgaussian(I,Sigma);
Eterm = (Iyy.*Ix.^2 -2*Ixy.*Ix.*Iy + Ixx.*Iy.^2)./((1+Ix.^2 + Iy.^2).^(3/2));
Eedge = sqrt(Ix.^2 + Iy.^2);
Eextern= (Wline*Eline - Wedge*Eedge -Wterm * Eterm);
%%
function J=ImageDerivatives2D(I,sigma,type)
% Les dérivés de l'image ( fonction de Gauss)
% les sorties
%   J : Le dérivé d'image
[x,y]=ndgrid(floor(-3*sigma):ceil(3*sigma),floor(-3*sigma):ceil(3*sigma));
switch(type)
    case 'x'
        DGauss=-(x./(2*pi*sigma^4)).*exp(-(x.^2+y.^2)/(2*sigma^2));
    case 'y'
        DGauss=-(y./(2*pi*sigma^4)).*exp(-(x.^2+y.^2)/(2*sigma^2));
    case 'xx'
        DGauss = 1/(2*pi*sigma^4) * (x.^2/sigma^2 - 1) .* exp(-(x.^2 +
y.^2)/(2*sigma^2));
    case {'xy','yx'}
        DGauss = 1/(2*pi*sigma^6) * (x .* y)           .* exp(-(x.^2 +
y.^2)/(2*sigma^2));
    case 'yy'
        DGauss = 1/(2*pi*sigma^4) * (y.^2/sigma^2 - 1) .* exp(-(x.^2 +
y.^2)/(2*sigma^2));
end
J = imfilter(I,DGauss,'conv','symmetric');
%
function I=imgaussian(I,sigma,siz)
% les entrées
%   I: L'image d'entrée 2D en niveaux de gris / couleur
%   SIGMA: Le sigma est utilisé pour le noyau Gaussien
%   SIZE: Taille du noyau (valeur unique)
% les sorties
%   J: L'image gaussien filtré
if(~exist('siz','var')), siz=sigma*6; end
if(sigma>0)
    % Assurez-1D noyau gaussien
    x=-ceil(siz/2):ceil(siz/2);
    H = exp(-(x.^2/(2*sigma^2)));
    H = H/sum(H(:));
    % Filtrer chaque dimension avec les noyaux 1D gaussiennes
    if(ndims(I)==1)
        I=imfilter(I,H, 'same' ,'replicate');
    elseif(ndims(I)==2)
        Hx=reshape(H,[length(H) 1]);
        Hy=reshape(H,[1 length(H)]);
```

```
        I=imfilter(imfilter(I,Hx, 'same' ,'replicate'),Hy, 'same' ,'replicate');
    elseif(ndims(I)==3)
        if(size(I,3)<4) % Détecter si l'image 3D
            Hx=reshape(H,[length(H) 1]);
            Hy=reshape(H,[1 length(H)]);
            for k=1:size(I,3)
                I(:,:,k)=imfilter(imfilter(I(:,:,k),Hx, 'same' ,'replicate'),Hy, 'same'
,'replicate');
            end
        else
            Hx=reshape(H,[length(H) 1 1]);
            Hy=reshape(H,[1 length(H) 1]);
            Hz=reshape(H,[1 1 length(H)]);
            I=imfilter(imfilter(imfilter(I,Hx, 'same' ,'replicate'),Hy, 'same'
,'replicate'),Hz, 'same' ,'replicate');
        end
    else
        error('imgaussian:input','unsupported input dimension');
    end
end
%
function [x,y,xc,yc]= cont(n,I)
%%%%Nombre des points de contour
[z,q]=size(I) ;
%%%%choix de rayon(localisation)
if z<q
    ra=((z-10)/2);
else
    ra=((q-10)/2);
end
%%%%yc et xc ces sont les cordonnées de centre du cercle (contour initial)
yc=floor(z/2);
xc=floor(q/2);
%%%Tracer une cercle avec rayon ra et un centre (xc,yc)
for i=1:n
    x(i)=xc+floor(ra*cos(2*pi*i/n));
    y(i)=yc+floor(ra*sin(2*pi*i/n));end
```