

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOU MAMMERI, TIZI-OUZOU
FACULTE DE GENIE ELECTRIQUE ET DE L'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE

Mémoire de fin d'études
Présenté en vue de l'obtention
du Diplôme d'Ingénieur d'Etat en Electronique

Option : Communication

Thème:

***Segmentation d'image approche
contour en utilisant les détecteurs de
Deriche et Shen :
application aux images MSG***

Proposé et dirigé par :

M^r : S.Ameur

Conçu et réalisé par :

M^r: Aimeur Ghania

M^r: Aissat Naima

Année universitaire **2008/2009**

SSOMMAIRE

Introduction générale	1
------------------------------------	---

CHAPITRE I : Généralités sur le traitement d'image.

1. Préambule	2
2. Définition de l'image	2
3. Image numérique	2
4. Caractéristiques d'une image numérique	2
4.1. Le pixel	2
4.2. Dimension	3
4.3. Résolution.	3
4.4. Bruit	3
4.5. Contours	3
4.6. Luminance	3
4.7. Contraste	3
4.8. Images binaires	4
4.9. Image en niveaux de gris	4
4.10. Image en couleurs	4
4.11. Histogramme	5
4.12. Voisinage d'un pixel	6
5. Qualité de l'image numérisée	6
6. Image bitmap et image vectorielle	6
7. Les systèmes de traitement d'images	6
7.1. Acquisition des données images	7
7.2. Dispositif de numérisation d'images	7

7.3. Prétraitement et post traitement	8
7.3.1. Modification d'histogramme	8
7.3.1.1. Egalisation d'histogramme	8
7.3.1.2. Etirement d'histogramme	9
7.3.1.3. Le seuillage	9
7.3.2. Filtrage numérique	10.
7.3.2.1. Filtres linéaires	10
a-. Filtre global	11
b-. Filtre local	11
b.1. Filtres passe-bas (lissage)	11
b.1.1. Le filtre moyen	12
b.1.2. Le filtre Gaussien	12
b.2. Filtres passe hauts (accentuation).....	12
b.3. Filtres directionnels	13
7.3.2.2. Filtres non linéaires	13
a. Filtres d'ordre.....	13
a.1. Filtre médian	13
a.2. Le plus proche voisin radiométrique	13:
b. Filtres de position	14
c. Filtres morphologiques	15
7.4. Traitement numérique des images	15
7.4.1. La segmentation	16
7.5. Visualisation	16
8. Domaines d'application	16

CHAPITRE II : Méthodes de détection de contours

1. Préambule	17
2. Définition d'un contour	17
3. Méthodes de détections de contours	18
3-1.Méthodes dérivatives.....	18
A. Approche Gradient.....	18
a-1. Calculer la norme du gradient.....	20

a-2. Sélection des points de fort gradient.....	20
a-2.1 Extraction des extrema locaux	20
a-2.2 Seuillage par hystérésis des extrema locaux.....	21
A .1 Opérateurs dérivatifs du premier ordre	21
Opérateur de Robert	22
Opérateur de Prewitt et Sobel	22
Opérateur gradient directionnel de Kirsh.....	23
Opérateur MDIF	23
B. Approche Laplacien	24
b-1. Détection des passages par zéro	25
b-2. Seuillage des passages par zéro	25
B.1 Opérateurs dérivatifs de second ordre.....	25
C. les filtres optimaux	26
1- Critère de Canny	26
2- Le filtre de Shen	29
3- Le filtre de Deriche	29
3.2 Les méthodes surfaciques	29
a-Méthode de HUECKEL	30
b- Méthode de Haralick	31
3.3 Méthodes morphologique	32
a- Gradient morphologique	32
4. suivi de contours	33
5. Modélisation des contours	33
5.1 Chaînage des points de contours	33
5.2 Modélisation de contour	34

CHAPITRE III : Opérateur optimal de DERICHE

1. Préambule	35
2. Opérateur monodimensionnel de Deriche	35
3. Opérateurs bidimensionnels de dérivation et de lissage de Deriche	37
3.1 Lissage	37
3.2 Calcul du gradient	38
3.3 Opérateur Laplacien de Deriche :.....	39
4. Implantation des opérateurs de Deriche	40
4.1 Implantation des opérateurs monodimensionnels de Deriche	40

4.1.1 Dérivation	40
4.1.2 Lissage	43
4.2 Implémentation de l'opérateur gradient bidimensionnel de Deriche	44

CHAPITRE IV : opérateur optimal de SHEN

1. Préambule	48
2. Modèle de contour et de bruit	48
3. Critère de performance	48
4. Filtre optimal de Shen.....	50
4.1 Cas monodimensionnel.....	50
4.1.1 Opérateur de lissage.....	50
4.1.2 Opérateur de dérivation	50
4.2 Extension bidimensionnelle	52
4.2.1 Filtre de lissage	52
4.2.2 Calcul du gradient	52
4.2.3 Calcul du Laplacien	53
5. Implémentation des opérateurs de Shen	53
5.1 Cas monodimensionnel	53
5.1.1 Opérateur de lissage	53
5.1.2 Opérateur de dérivation.....	54
5.2 Cas bidimensionnel	55
5.2.1 Opérateur de lissage.....	55
5.2.2 Opérateur de dérivation	55

CHAPITRE V : Testes et résultats

1. Préambule	58
2. Présentation du logiciel pandore	58
2.1. Fonctionnement du logiciel ' Pandore'	58
2.2. Exemple de fonctions sous format « Pandore ».....	59
Conversion de format de fichiers.....	59
Conversion de type Pandore	59
Détection de contours	60
Détection de contour par Deriche	61

3. Testes et résultats	62
3.1 Description sur les images	62
Le canal visible	62
Le canal infrarouge	62
Le canal vapeur d'eau	62
Maison	62
Bureau	62
Ville.....	63
4- interprétation des résultats	73
Discussion.....	73
Conclusion générale.....	75
Annexe.....	
Bibliographie.....	

Introduction

Le traitement d'image désigne une discipline des mathématiques appliquées qui étudie des images numériques et leur transformation, dans le but d'améliorer leur qualité ou d'en extraire de l'information. Il s'agit donc d'un sous-ensemble du traitement du signal dédié aux images et aux données dérivées comme la vidéo.

Notre mémoire porte sur la segmentation d'images par approche frontière. Ce problème a été largement traité dans la littérature et il ressort qu'il est difficile de choisir une méthode car les résultats dépendent notamment du type d'images à traiter. En effet, pour valider correctement une segmentation d'objets naturels, comme en imagerie médicale par exemple, il faut disposer de la « vérité terrain ». De plus dans le cas où les contours sont flous ou mal définis il est souvent nécessaire d'utiliser des méthodes hybrides ou des techniques de segmentation guidée par l'interprétation.

La connaissance des contours des objets dans une image est très utile pour la reconnaissance des formes. De plus les contours sont aussi utilisés comme primitives en analyse d'images et en vision par ordinateur. La qualité et la précision des contours détectés jouent donc un rôle très important chaque fois que l'on doit mettre en correspondance des primitives robustes issues d'images différentes. Citons par exemple : la stéréovision, l'analyse du mouvement, etc.

Dans le cadre de notre travail, nous nous sommes intéressées aux méthodes dérivatives. Plus particulièrement aux approches développées par Deriche et de Shen. Ces techniques basées sur le filtrage optimal permettent en effet d'avoir de bons résultats lorsque les images à traiter ne sont pas texturées.

La rédaction de notre mémoire a été organisée en cinq chapitres

Dans le premier chapitre, nous présentons des généralités sur le traitement d'image.

Dans le second chapitre, nous donnons un bref aperçu sur les méthodes de détection de contour et nous détaillerons les méthodes dérivatives.

Le troisième chapitre et le quatrième sont consacrés à l'étude et l'implémentation des opérateurs optimaux de Deriche et de Shen.

Dans le cinquième chapitre, nous donnerons les résultats obtenus ainsi que leurs interprétations.

Enfin nous terminerons notre travail par une conclusion.

1. Préambule

La compréhension du traitement d'images commence par la compréhension de ce qu'est une image .le mode et les conditions d'acquisition et de numérisation. Dans ce chapitre, nous donnons les définitions essentielles ainsi que les outils que nous utiliserons dans notre étude.

2. Définition de l'image

Une image est définie comme une représentation d'une personne, d'une scène ou d'un spectacle dans le monde réel par la peinture, la photo ...
L'image ou les signaux de l'image provenant des capteurs ont été numérisés pour pouvoir être stockés sur l'ordinateur.

Elle peut être décrite sous la forme d'une fonction $I(x, y)$ de brillance analogique continue, définie dans un domaine borné, tel que x et y sont les coordonnées spatiales d'un point de l'image et I est une fonction d'intensité lumineuse et de couleur, sous cet aspect, l'image est inexploitable par la machine, ce qui nécessite sa numérisation.

3. Image numérique

Une image est constituée d'un ensemble de points appelés pixels. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions. Une image est donc représentée par un tableau dont chaque case est un pixel. La valeur stockée dans une case est codée sur un certain nombre de bits, le nombre de bits détermine la couleur ou le niveau de gris du pixel.

4. Caractéristiques d'une image numérique

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants :

4.1. Le pixel

Le pixel est une abréviation de « Picture x element », c'est-à-dire qu'il s'agit du plus petit élément constitutif d'une image. La qualité d'information que véhicule chaque pixel donne des nuances entre images monochromes et image couleurs.

4.2. Dimension

C'est la taille de l'image. Cette dernière représente le nombre total de pixels contenus dans l'image. Cette taille est obtenue en effectuant le produit entre le nombre de pixels d'une ligne par celui d'une colonne.

4.3. Résolution

C'est la clarté ou la finesse des détails atteinte par un moniteur ou une imprimante dans la production d'image sur les moniteurs d'ordinateurs. La résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour désigner le nombre totale de pixels affichables horizontalement ou verticalement sur un moniteur, plus grand est ce nombre, meilleur est la résolution.

4.4. Bruit

Un bruit (parasite) dans une image est considéré comme un phénomène de variation brusque de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur.

4.5. Contours

Les contours représentent les frontières entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative.

4.6. 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.

4.7. Contraste

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images.

Si L_1 et L_2 sont les degrés de luminosité respectivement de deux zones voisines A_1 et A_2 d'une image, le contraste est définie par rapport :
$$C = \frac{L_1 - L_2}{L_1 + L_2} \quad (1-1)$$

4.8. Images binaires

Une image binaire est une matrice dont le nombre de niveaux de gris est réduit à deux éléments 0 et 1, ou le niveau 0 représente le noir et le niveau 1 représente le blanc. On n'utilise dans ce cas qu'un seul bit pour coder le niveau de gris. La figure 1-c illustre ce type d'images.

4.9. Image en niveaux de gris

Le niveau de gris est la valeur de l'intensité lumineuse de chaque pixel. Sa couleur peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images en niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la qualité de lumière renvoyée. Lorsque les pixels sont codés sur 8 bits, cette valeur est comprise entre 0 et 255 soit 256 niveaux de gris. Chaque pixel n'est donc plus représenté par un bit, mais par un octet. Pour cela il faut que le matériel utilisé pour afficher l'image soit capable de produire les différents niveaux de gris correspondant. Le nombre de niveaux de gris dépend donc du nombre de bits utilisés pour coder les pixels de l'image.

La plupart des opérateurs de traitement d'image ne travaillent que sur les images en niveau de gris. La figure 1-b illustre ce type d'images.

4.10. Image en couleurs

Ces images sont codées en utilisant le codage des trois couleurs fondamentales (rouge, vert, bleu), on parle alors d'images RVB. Chaque couleur est codée sur 1 octet, d'où :

Une composante rouge d'intensité de 0 à 255

Image RVB Une composante verte d'intensité de 0 à 255

Une composante bleue d'intensité de 0 à 255

On code ainsi $2^{24} = 16777216$ couleurs différentes. Cela est cependant théorique, car aucun écran n'est capable d'afficher 16 millions de points. Dans la plus haute résolution (1600x 1200), l'écran n'affiche que 1920000 points. Par ailleurs, l'œil humain n'est pas capable de distinguer autant de couleurs. La figure 1 illustre ce type d'images.



a- Image couleur b- Image à niveaux de gris c- Image binaire

Figure1 : exemple d'images

4.11. Histogramme

On appelle histogramme de l'image I, la fonction H définie sur l'ensemble des entiers naturels est donnée par l'équation :

$$H(x) = \text{Card} \{P - I(P) = x\}. \quad (1-2)$$

C'est à dire que H(x) traduit le nombre d'apparitions du niveau de gris x dans l'image I. Cette définition se généralise aux images multi-bandes, l'histogramme est alors une fonction de p variables où p désigne le nombre de canaux. L'histogramme est un outil privilégié en analyse d'images car il représente un résumé simple, mais souvent suffisant du contenu de l'image. Un exemple de l'histogramme de l'image « rose » est donné par la figure 2.

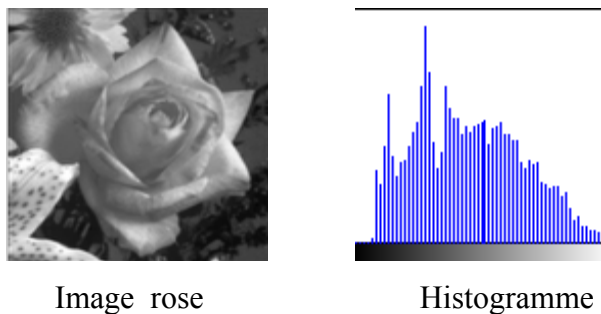


Figure 2 : exemple d'histogramme

On peut distinguer 3 types d'histogrammes d'images :

Histogramme Unimodal : Ce type d'histogramme n'a qu'un pic, il représente alors soit un objet soit un fond.

Histogramme bimodal : il est formé de deux pics, séparés par une vallée et l'on déduit ainsi qu'il existe un objet sur un fond.

Histogramme multimodal : il est formé de plusieurs modes séparés (plusieurs pics séparés par plusieurs vallées) qui nous renseignent sur la présence de plusieurs objets.

4.12. Voisinage d'un pixel :

Tout pixel d'une image est caractérisé par un couple de coordonnées spatiales (i, j). Le voisinage du pixel est défini comme étant l'ensemble des pixels qui l'entourent dans toutes les directions possibles (8 directions : 2 horizontales, 2 verticales, 4 diagonales), ces 8 voisins forment la fenêtre 3 x 3 des pixels (voir Fig.3).

$(i-1, j-1)$	$(i-1, j)$	$(i-1, j+1)$
$(i, j-1)$	(i, j)	$(i, j+1)$
$(i+1, j-1)$	$(i+1, j)$	$(i+1, j+1)$

Figure 3 : les 8 voisins d'un pixel (i, j) .

5. Qualité de l'image numérisée

Elle dépend d'une part, de la qualité des images d'origine et d'autre part, des moyens mis en œuvre pour convertir un signal analogique en signal numérique. Elle dépend aussi de la qualité des périphériques de numérisation de l'image, du nombre de niveau de gris ou de couleurs enregistrées, de la qualité de l'affichage de l'écran (définition de l'écran, nombre de teintes disponibles simultanément, calibrage de l'écran, etc.).

Les critères d'appréciation de la qualité d'une image, tels que cités succinctement, dépendent largement de la structure même de l'image et de son mode de représentation (bitmap ou vectorielle).

6. Image bitmap et image vectorielle

Les images appartiennent à deux grandes familles : bitmap (image bit) et vectorielle.

- Une image vectorielle est décrite à l'aide de courbes et d'équation mathématique.
- Une image bitmap est constituée de pixels et se réduit donc à une matrice de points.

7. Les systèmes de traitement d'images

Tout système de traitement numérique d'images est composé de plusieurs parties à savoir l'acquisition de l'image, son prétraitement, son analyse son post-traitement, puis sa visualisation ou son stockage ou tout simplement sa transmission. Un schéma général d'un système de traitement d'image est donné par la figure 4.

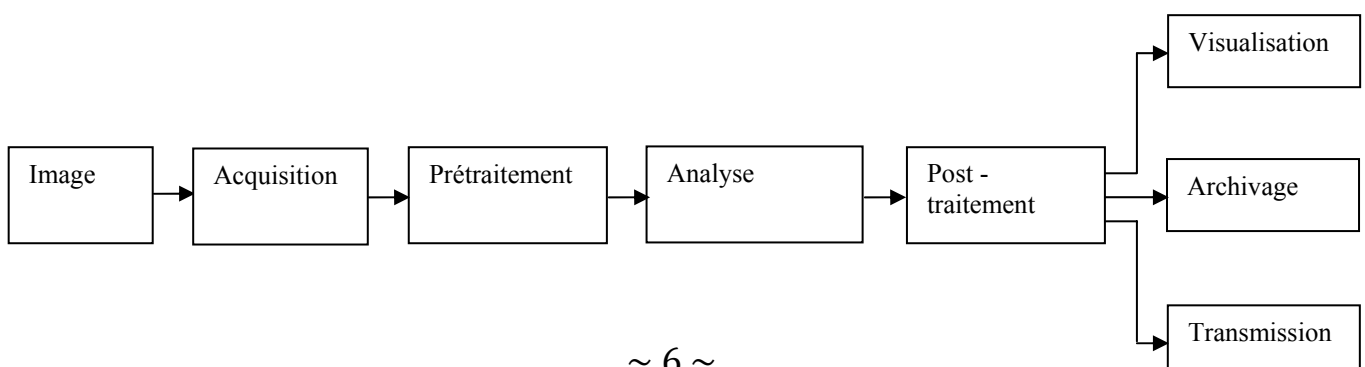


Figure 4 : Schéma d'un système de traitement d'image

7.1. Acquisition des données images

L'acquisition d'une image est l'opération qui permet le passage de l'information réelle à une représentation numérique. Ce passage appelé aussi numérisation qui est une conversion d'un signal analogique en un signal numérique.

Elle est réalisée en plusieurs étapes :

- Le transfert des images optiques qui est une opération permettant de transformer le signal optique en une information électrique (signal analogique).
- L'échantillonnage de ce signal analogique qui consiste à multiplier ce dernier par une série d'impulsions dans le but d'avoir des échantillons.
- La quantification qui est une traduction des échantillons en valeurs numériques selon une règle de codage choisi.

7.2. Dispositif de numérisation d'images

La procédure de numérisation [15] d'une image est schématisée par la figure 5.

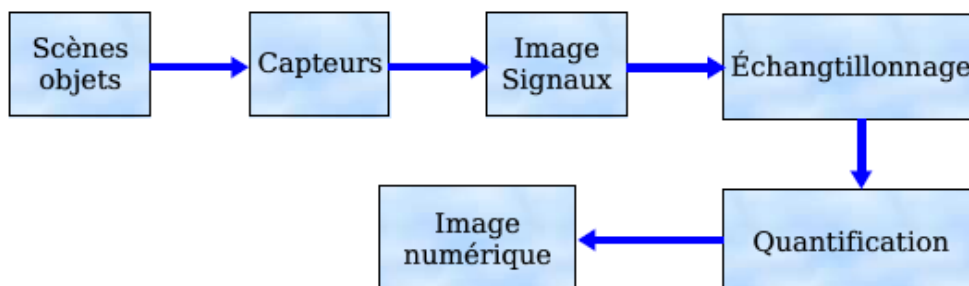


Figure 5 : Numérisation d'une image

7.3. Prétraitement et post traitement

Le prétraitement est la première étape du traitement de l'information visuelle, son but est d'améliorer la perception de certains détails et de réduire le bruit induit notamment par les capteurs. Il permet aussi de faciliter l'analyse d'une image en renforçant la ressemblance entre pixels appartenant à une même région, ou en accentuant la dissemblance entre pixels appartenant à des régions différentes.

Le prétraitement est nécessaire notamment dans les cas suivants:

- image est bruitée,
- faible contraste,
- Eclaircissement de l'image non uniforme.

Les méthodes de prétraitement les plus utilisées sont :

- Modification de l'histogramme.
- Réduction de bruit (filtrage).

7.3.1. Modification d'histogramme

7.3.1.1. Egalisation d'histogramme

L'égalisation d'histogramme est un outil qui permet d'améliorer la qualité visuelle des images (mauvais contraste, image trop sombre ou trop claire, mauvaise répartition des niveaux d'intensité, etc...). Cette transformation consiste à rendre le plus plat possible l'histogramme des niveaux de gris, ce qui donne une meilleure dynamique et un fort contraste à l'image. La figure 6-b illustre ce type de traitement.

Interprétation mathématique :

Soit $R(i)$ la fonction de répartition du niveau de gris et N_g : le nombre de niveaux de gris.

$R(i)$ alors définie par :

$$R(i) = \sum_{k=0}^i P_k \quad (1-3)$$

Avec

$$i=0, 1, \dots, N_g-1$$

P_k : la probabilité d'apparition du niveau k .

$R(i)$ croit de 0 à 1, elle est normalisée en la multipliant par le nombre maximum des niveaux de gris. On aura finalement :

$$R_c(i) = (N_g-1) \sum_{k=0}^i P_k \quad (1-4)$$

$R_c(i)$ sera la nouvelle valeur du niveau de gris i .

7.3.1.2. Etirement d'histogramme

L'étirement d'histogramme (aussi appelé linéarisation d'histogramme) consiste à répartir les fréquences d'apparition des pixels sur la largeur de l'histogramme. Ainsi il s'agit d'une opération consistant à modifier l'histogramme de telle manière à répartir au mieux les intensités sur l'échelle des valeurs disponibles. Ceci revient à étendre l'histogramme afin que la valeur d'intensité la plus faible soit à zéro et que la plus haute soit à la valeur maximale.

Ce cette façon, si les valeurs de l'histogramme sont très proches les unes des autres, l'étirement va permettre de fournir une meilleure répartition afin de rendre les pixels claires encore plus claires et les pixels foncés proche du noir. La figure 6-c illustre ce type de transformation.

Toute fois cela ne donne pas toujours de bons résultats.

7.3.1.3. Le seuillage

L'opérateur dit de « simple seuillage » consiste à mettre à zéro '0' tous les pixels ayant un niveau de gris inférieur à une certaine valeur (appelée seuil) et à un '1', les pixels ayant une valeur supérieure à ce seuil. Ainsi le résultat du seuillage est une image binaire contenant les pixels noirs et blancs, c'est la raison pour laquelle le terme de binarisation est employé. Le seuillage permet de mettre en évidence des formes ou des objets dans une image. La figure 6-d illustre ce type transformation.

Toutefois, la différence réside dans le choix du seuil à adopter.



a- Image initiale "Maison"



b- Egalisation d'histogramme



c- Etirement d'histogramme

d- Seuillage binaire

Figure 6 : Images obtenues en modifiant les histogrammes des niveaux de gris de l'image "Maison"

7.3.2. Filtrage numérique

Pour améliorer la qualité visuelle de l'image, on doit éliminer les effets du bruit, en lui faisant subir un traitement appelé filtrage.

Il existe un grand nombre de filtres, parmi lesquelles on peut citer :

- Les filtres linéaires.
- Les filtres non linéaires.

7.3.2.1. Filtres linéaires

Ils sont appelés ainsi car ils possèdent la propriété de linéarité du produit de convolution. Filtrer une image $I(x, y)$, consiste à la convoluer avec une fonction $f(x, y)$ qui est la réponse impulsionnelle du filtre.

Dans le cas continu, l'image filtrée est donnée par :

$$I'(i, j) = (f * I)(i, j) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(i_1, j_1) \cdot I(i - i_1, j - j_1) di_1 dj_1 \quad (1-5)$$

Dans le cas discret, les domaines de définition de I et f sont bornés. Le domaine de I est $[-N/2, +N/2]$ et le domaine de f est $[-K/2, +K/2]$, avec $K \leq N$ et N est la taille de l'image

La convolution s'écrit alors:

$$I'(i, j) = (f * I)(i, j) = \sum_{i'=-\frac{K}{2}}^{+\frac{K}{2}} \sum_{j'=-\frac{N}{2}}^{+\frac{N}{2}} f(i-i', j-j') \cdot I(i', j') \quad (1-6)$$

Il existe deux méthodes pour ce type de filtrage : le filtrage linéaire global et le filtrage linéaire local.

a-. Filtre global

Le filtrage global consiste à effectuer le produit de convolution de la transformée de Fourier de l'image par une fonction F , qui est la fonction de transfert du filtre, appelée « gain complexe du filtre »

b-. Filtre local

Ce filtrage consiste à effectuer le produit de convolution de l'image par une fonction de voisinage (masque).

La convolution est effectuée en appliquant un masque à tous les pixels de l'image.

Les filtres locaux les plus utilisés sont les filtres passe bas et les filtres passe haut.

b.1. Filtres passe-bas (lissage)

Ils sont utilisés pour atténuer le bruit et les irrégularités de l'image. En pratique, il faut choisir un compromis entre l'atténuation du bruit et la conservation des détails et contours significatifs.

Pour réaliser ce filtrage, on effectue le produit de convolution de l'image par une fonction de voisinage définie d'une façon générale par :

$$H = \left[\frac{1}{b+2} \right]^2 \begin{bmatrix} 1 & b & 1 \\ b & b^2 & b \\ 1 & b & 1 \end{bmatrix}$$

$$\text{Où } \left[\frac{1}{b+2} \right]^2 : \text{ est un facteur de normalisation.}$$

Parmi les filtres passe-bas les plus utilisés, on trouve :

b.1.1. Le filtre moyen :

Il consiste à affecter au pixel traité la moyenne des niveaux de gris de ses voisins. Son masque est donné par :

$$H = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

La figure 7-a illustre le résultat obtenu en appliquant le filtre moyenneur.

b.1.2. Le filtre Gaussien

C'est un filtre de lissage très utilisé pour sa facilité de mise en œuvre. L'expression de la gaussienne en deux dimensions est donnée par :

$$g(x, y) = \frac{1}{2\sigma^2} \exp\left(\frac{-x^2 + y^2}{2\sigma^2}\right). \quad (1-7)$$

Pour filtrer une image par le filtre gaussien, il faut faire une approximation de la fonction gaussienne. Puis, nous faisons une convolution entre l'approximation de la fonction gaussienne et l'image à filtrer. Cette approximation est donnée par le masque

suivant :

$$H = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

L'intérêt du filtrage gaussien est que l'on règle très facilement le degré de filtrage à travers le paramètre σ . On constate par contre que ce filtrage dégrade les contours en introduisant du flou.

Nous utilisons souvent le filtre gaussien pour diminuer le bruit avant d'appliquer un opérateur de détection de contours. La figure 7-b illustre le résultat obtenu en appliquant le filtre gaussien.

b.2. Filtres passe hauts (accentuation)

Pour ce type de filtres, on attribue à chaque point de la fenêtre d'analyse, des poids statistique différents, mais choisis de telle sorte à ce que leur somme soit toujours égale à zéro. Les filtres passe-haut n'éliminent pas le bruit, mais permettent la mise en évidence de contours entre les plages de niveaux de gris différents. Ce filtrage est réalisé par le produit de convolution de l'image par les masques suivants :

$$H1 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad H2 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

b.3. Filtres directionnels

Dans certains cas, on cherche à faire apparaître des détails de l'image dans une direction bien déterminée. Pour cela, on utilise des filtres qui s'opèrent selon des directions choisies (horizontales, verticales et diagonales).

7.3.2.2. Filtres non linéaires

Ils constituent une alternative au filtrage linéaire, la nouvelle valeur du pixel n'est plus une combinaison linéaire des valeurs des pixels voisins, mais une autre forme de calcul excluant toute relation linéaire entre le pixel et son voisinage.

Parmi ces filtres, nous citerons :

a. Filtres d'ordre

Ils classent les niveaux de gris d'un voisinage et sélectionnent parmi ces quantités une ou plusieurs valeurs. Ils sont de deux types :

a.1. Filtre médian :

Cette méthode est très souvent exploitée en traitement d'image. Elle consiste à remplacer la valeur des pixels d'une fenêtre de l'image, par la valeur qui par classement croissant se trouve au milieu.

Ex. considérons la configuration bidimensionnelle suivante :

$$\begin{bmatrix} 1 & 5 & 1 \\ 4 & 6 & 3 \\ 1 & 2 & 7 \end{bmatrix}$$

Le classement croissant 111234567, la valeur du pixel central (qui était de 6) prendra la valeur 3.

Ce genre de filtre produit un adoucissement de l'image puisque la valeur du point central d'une fenêtre est affectée par celle de ces voisins. Ce filtre sert surtout à réduire le bruit impulsionnel dans l'image. La figure 7-c illustre ce type d'images.

a.2. Le plus proche voisin radiométrique :

Dans ce cas, on attribue au pixel central la valeur de la moyenne des K pixels voisins dont les valeurs (en niveaux de gris) sont les plus proches de la sienne.

Prenons l'exemple de la fenêtre 3x3 suivantes :

$$\begin{bmatrix} 1 & 1 & 2 \\ 2 & 3 & 1 \\ 3 & 4 & 2 \end{bmatrix}$$

La moyenne sur les K voisins avec K=4 est 3, Dans cet exemple, le point central garde sa valeur.

La figure 7-d illustre ce type de traitement.

b. Filtres de position

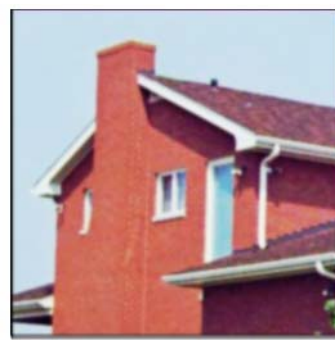
Pour ce filtrage, on utilise des voisinages 5*5. Le sou voisinage du point le plus homogène est sélectionné, et son niveau de gris moyen est affecté au point central.

Exemple :

3	3	4	5	6
3	4	3	6	7
3	3	4	8	3
9	2	3	2	1
8	5	3	2	6

La valeur du pixel central passera de 4 à 3

Ce filtre présent l'inconvénient d'avoir recours à un voisinage très grand (en nombre de pixels) du point considéré, ce qui nécessite un temps d'exécution long.



a- Application d'un filtre médian moyenneur b- Application d'un filtre gaussien



c- Application d'un filtre médiane d- Application d'un filtre basé sur le choix des voisins

Figure 7 : Application des filtres linéaires et non linéaires sur l' image "Maison"

c. Filtres morphologiques

Ces filtres ont été conçus en vue d'être appliqués sur des images en niveaux de gris et sur des images binaires.

Le concept de ce filtrage fait appel à des notions mathématiques morphologiques qui permettent de conserver les propriétés algébriques et géométriques, topologiques et ensemblistes.

Parmi ces filtres nous pouvons citer : l'érosion, la dilatation, l'ouverture et la fermeture.

Soit A l'image de départ et C_x l'élément structurant.

- **L'érosion** de A par l'élément structurant C_x est l'ensemble Y composé de tous les points x tel que C_x inclus dans A .

$$Y = A \ominus C_x = \{x / C_x \subset A\} \quad (1-8)$$

Sur les images au niveau de gris ce filtre permet de supprimer toutes les hautes fréquences claires de l'image, sur les images binaires il permet de cacher les petits points blancs.

- **La dilatation** de A par C_x est l'ensemble Y composé de tous les points x tel que C_x est un voisin de A .

$$Y = A \oplus C_x = \{x / C_x \neq \text{ensemble vide}\} \quad (1-9)$$

Sur les images au niveau de gris ce filtre permet de supprimer toutes les hautes fréquences sombres de l'image, sur les images binaires il permet de cacher les petits points noirs.

- **L'ouverture** comprend l'érosion suivie de la dilatation.
- **La fermeture** comprend la dilatation suivie de l'érosion.

L'ouverture et la fermeture morphologiques filtrent le bruit quelle que soit son amplitude.

7.4. Traitement numérique des images

Le traitement numérique des images est l'ensemble des méthodes qui permettent de décrire quantitativement le contenu d'une image. Les traitements applicables aux images sont nombreux et dépendent du domaine d'application. La segmentation est le traitement le plus utilisé.

7.4.1. La segmentation :

La segmentation est une étape importante dans une chaîne de traitement d'image par ordinateur. Elle consiste en une séparation des différents éléments de l'image en régions connexes présentant une homogénéité selon un critère spécifique à l'image, comme par exemple la couleur, la distribution des niveaux de gris ou la texture. L'union de ces régions redonne alors l'image initiale. Ces régions peuvent être caractérisées directement par les pixels qui les composent, il s'agit de la segmentation en régions homogènes. Ou bien par leurs frontières.

7.5. Visualisation

Tout système de traitement d'image est doté d'un dispositif de visualisation. Plusieurs types de supports peuvent être employés à cet effet (moniteur vidéo, cliché, photographique, impression sur papier, etc....).

8. Domaines d'application

Le traitement d'image possède l'aspect multidisciplinaire. On trouve ses applications dans des domaines très variés tels que les télécommunications (T.V, vidéo, publicité, ...) la médecine (radiographie, ultrasons, ...), la biologie, l'astronomie, la géologie, l'industrie (robotique, sécurité), la météorologie, l'architecture, l'imprimerie, l'armement (application militaire), les métiers du spectacle, la radiodiffusion, les créations artistiques, la CAO, la réalité augmentée, les spots publicitaires, l'EAO, etc.....

1. Préambule

Une image est généralement composée de plusieurs zones de niveaux de gris différents correspondants aux différents objets. Pour repérer ces objets dans l'image, deux grandes approches peuvent être envisagées à savoir, l'approche région qui consiste à rechercher des zones de niveaux de gris homogènes et l'approche contour basée sur la recherche des discontinuités entre les différentes régions de la scène.

Dans ce chapitre, nous définirons la notion de contour et nous exposerons brièvement les différentes méthodes de détection de contours.

2. Définition d'un contour

Par définition [15], un contour est une brusque variation de niveau de gris dans une image d'une amplitude A et avec une pente p . Un contour peut être défini comme une "marche d'escalier" si le contour est net, comme une "rampe" si le contour est flou ou comme un "toit" s'il s'agit d'une ligne sur un fond uniforme. La figure 1 illustre les différents cas contours.

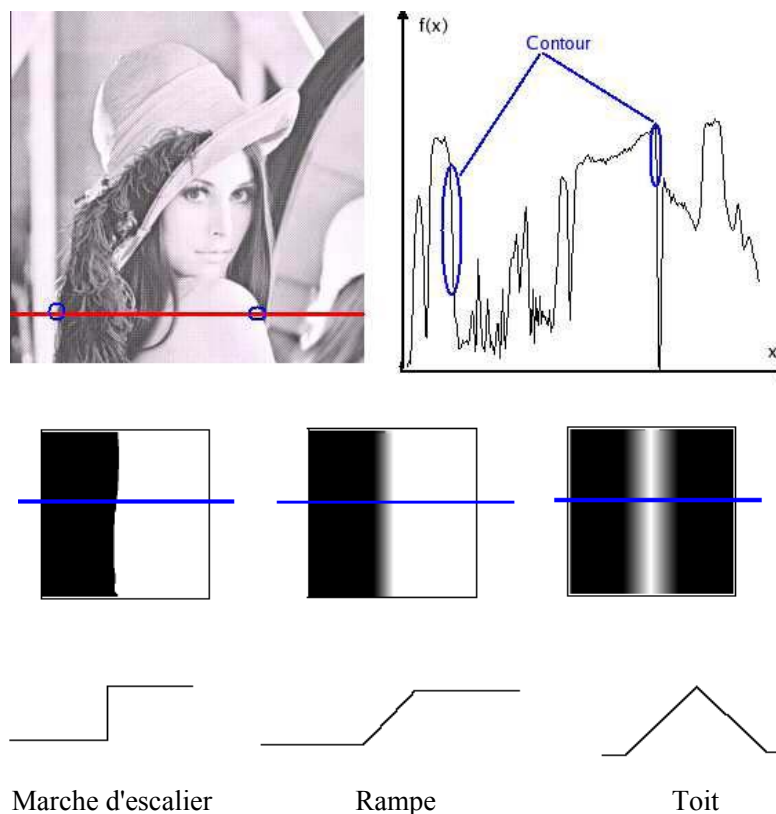


Figure 1 : contours et ses différents types

3. Méthodes de détections de contours

L'efficacité d'une méthode de détection de contours dépend de l'application visée. En effet, un algorithme de détection de contours est conçu, en fonction de l'utilisation ultérieure des contours obtenus et de la précision requise (images de télévision, problèmes de cartographie, de radiographie médicale, reconnaissance des formes ou analyse de scènes élémentaire...),

Plusieurs méthodes de détection de contours ont été publiées dans la littérature [1], [4], [6], [7], [8], [12], [15] dans le cadre de notre mémoire nous nous intéressons uniquement aux méthodes dérivatives.

3-1)-Méthodes dérivatives

Ces méthodes sont basées sur l'étude des dérivées de la fonction d'intensité de l'image. Les points du contour sont obtenus en maximisant les extréma locaux de la norme du Gradient ou en détectant les passages par zéro du Laplacien. Notons que le calcul de la dérivée nécessite un pré filtrage des images (filtre Gaussien pour les bruits de moyenne nulle tel que le bruit blanc, filtre médian pour les bruits impulsionnels et filtre morphologique pour les bruits de grande amplitude)

Les approches dérivatives [1] sont les plus rapides pour détecter et localiser un signal. On distingue les méthodes basées sur le gradient et celles basées sur le Laplacien.

A. L'approche Gradient

La première approche possible pour détecter les variations locales de la fonction $I(x, y)$ représentant l'intensité est d'utiliser une transformation du type gradient [2].

Un contour d'orientation θ au point (x, y) voir figure 3 est détecté par un maximum de la dérivée directionnelle, dans la direction Φ du gradient $\nabla I(x, y)$, c'est-à-dire par le maximum de la fonction (2-1) :

$$G(\Phi) = \nabla I(x, y) \cdot \vec{n} \quad (2-1)$$

\vec{n} Étant le vecteur unitaire dans la direction du gradient : $\vec{n} (\cos(\Phi), \sin(\Phi))$

La dérivée première directionnelle de $I(x, y)$ suivant n s'écrit alors comme suit :

$$G(\Phi) = \nabla I(x, y) \cdot \vec{n} = \cos(\Phi) \frac{\partial I}{\partial x} + \sin(\Phi) \frac{\partial I}{\partial y} \quad (2-2)$$

En dérivant $G(\Phi)$ et en annulant l'expression (2-2), on trouve:

L'orientation du gradient Φ tel que :

$$\Phi = \arctang \left(\frac{\frac{\partial I}{\partial y}}{\frac{\partial I}{\partial x}} \right) \quad (2-3)$$

De même l'orientation du contour θ :

$$\theta = \frac{\pi}{2} - \Phi \quad (2-4)$$

La direction du gradient maximise la dérivée directionnelle, elle est normale au contour et s'oriente de la partie claire vers la partie foncée.

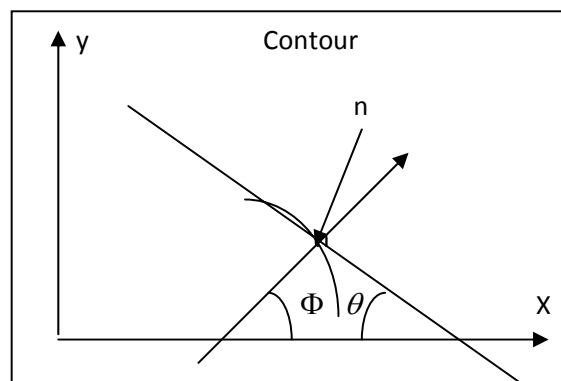


Figure 2: orientation d'un contour

Les points de contour dans une image sont caractérisés par des extrema locaux du gradient.

L'approche gradient consiste donc à :

a-1. Calculer la norme du gradient en tout point de l'image

$$N(x; y) = [G_1(x, y)^2 + G_2(x, y)^2]^{1/2} \quad (2-5)$$

a-2. Sélection des points de fort gradient

On détermine les points tels que :

$$N(x, y) > S \quad S : \text{seuil fixé a priori}$$

Dans le cas où la norme du gradient au point de contour varie fortement [14], il n'existe pas de seuil S permettant d'obtenir les vrais points de contours sans sélectionner aussi ceux dus au bruit.

Pour contourner cette difficulté, on utilise l'extraction des extrema locaux. Ces derniers correspondent au passage par zéro de la dérivée de la norme du gradient dans la direction du gradient.

a-2.1 Extraction des extrema locaux de la norme du gradient

Soit M un point de gradient $G(M)$, et M_1 et M_2 deux points de la droite passante par M , de vecteur directeur $G(M)$. le point M_1 est pris dans le sens du gradient et M_2 dans le sens inverse, la figure 3 nous donne le maximum local selon la direction du gradient.

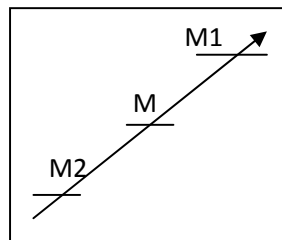


Figure 3 : Maximum local selon la direction du gradient

Le point M est sélectionné si $N(M) > N(M_1)$ et $N(M) \geq N(M_2)$. Le fait d'imposer que le maximum soit limité dans un sens, revient à choisir un point de contour dans la zone de faible ou de forte valeur de la fonction des niveaux de gris. Ce la revient à se déplacer le long de la normale au contour, et à détecter le point de plus fort gradient. L'algorithme d'extraction

d'extrémas locaux est décrit en annexe. La figure (4-b) montre cette extraction des maxims locaux.

a-2.2 Seuillage par hystérésis des extrema locaux

Ce type de seuillage nécessite de calcul de deux seuils S_b et S_h . Ces seuils sont déterminés à partir de l'histogramme des niveaux de gris de l'image représentant la norme du gradient aux extremas.

Cette étape permet d'éliminer les faux points de contours caractérisés soit par une norme de gradient faible, soit par points isolés. La figure (4-c) montre ce type de seuillage.

Conditions de détections des points contours

On détermine deux seuils sh (seuil haut) et sb (seuil bas) tels que :

si $M(x,y) < sb$, (x,y) n'est pas un point contour.

si $M(x,y) > sh$, (x,y) n'est pas un point c.

si $sb \leq M(x,y) \leq sh$ alors (x,y) est un point contour.

L'algorithme de seuillage par hystérésis est décrit en annexe.

A .1 Opérateurs dérivatifs du premier ordre

Dans le cas discret les dérivées directionnelles suivant les directions horizontales et verticales sont approchées par de simples différences finies [1].

Soit $I[i,j]$ l'image à traiter (i indice ligne , j indice colonne), les images $I_i[i,j]$ et $I_j[i,j]$ sont de la forme :

$$\frac{\partial I}{\partial y} \approx \frac{\Delta I}{\Delta i} = I_i[i, j] = I[i+1, j] - I[i, j] = (I * G_i)(i, j) \quad (2-6)$$

$$\frac{\partial I}{\partial x} \approx \frac{\Delta I}{\Delta j} = I_j[i, j] = I[i, j+1] - I[i, j] = (I * G_j)(i, j) \quad (2-7)$$

Avec

$$G_i = [-1, 1] \quad \text{Et} \quad G_j = \begin{bmatrix} -1 \\ +1 \end{bmatrix}$$

La norme de gradient est alors donnée par :

$$|\nabla I[i, j]| = \sqrt{I_i^2[i, j] + I_j^2[i, j]} \quad (2-8)$$

Où encore par :

$$|\nabla I[i, j]| = \max \{ |I_i[i, j]|, |I_j[i, j]| \} \quad (2-9)$$

Opérateur de Robert :

Roberts approxime les dérivées directionnelles suivant les axes orienté à 45 degrés à l'aide d'une convolution de l'image $I[i, j]$ avec les masques suivants :

$$H_0 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad \text{et} \quad H_1 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

Soient $I_j[i, j] = H_0 * I[i, j]$ et $I_i[i, j] = H_1 * I[i, j]$

L'image avec les contours accentués I_a est alors obtenue comme étant l'image approximant la norme du gradient. L'application de cet opérateur sur une image réelle nous donne le résultat illustré sur la figure (5-c). I_a a pour expression :

$$I_a = |\nabla I[i, j]| \quad (2-10)$$

Opérateur de Prewitt et Sobel

Prewitt et Sobel approximent les dérivées directionnelles suivant les axes O_x et O_y à l'aide d'une convolution de l'image $I[i, j]$ avec les masques 3×3 :

$$H_i = \begin{bmatrix} 1 & c & 1 \\ 0 & 0 & 0 \\ -1 & -c & -1 \end{bmatrix} \quad \text{et} \quad H_j = \begin{bmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{bmatrix}$$

Les masques de Prewitt sont définis pour $c=1$

Les masques de Sobel sont définis pour $c=2$

On remarque que les deux masques combinés à la fois l'opération de lissage par $[1,0,-1]$ suivi de l'opération de dérivation par $[1,0,-1]$ dans la direction orthogonale. Les figures (5-b) et (5-d) montrent l'application de ces opérateurs sur une image réelle.

Opérateur gradient directionnel de Kirsh

L'opérateur de Kirsh est très intéressant dans les images comprenant des points où se rejoignent plusieurs contours et où le bruit est très important. C'est un opérateur à 8 masques correspondant chacun à une direction préférentielle et obtenu par rotation $\frac{\pi}{4}$ de l'opérateur de base H_0 .

$$H_0 = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}, H_1 = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}, \dots, H_8 = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$$

Le gradient retenu est donné par l'équation :

$$|\nabla I[i, j]| = \max \{H_i * I\}_{i=0..7}$$

De même pour l'orientation Φ qui a pour expression :

$$\Phi = \frac{\pi}{4} \arg \max \{ |H_i * I| \}_{i=0..7}$$

Opérateur MDIF

Il s'agit de la combinaison d'un filtre moyenneur M et d'un dérivateur utilisant les masques directionnels de Prewitt H_i et H_j .

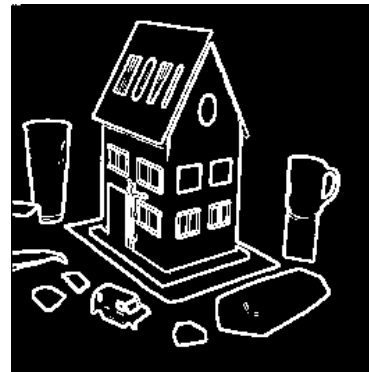
$$M = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}, H_i = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}, H_j = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix},$$

Les approximations des composantes du gradient peuvent être calculées après le lissage ou directement en appliquant les masques suivants de taille 5*5 :

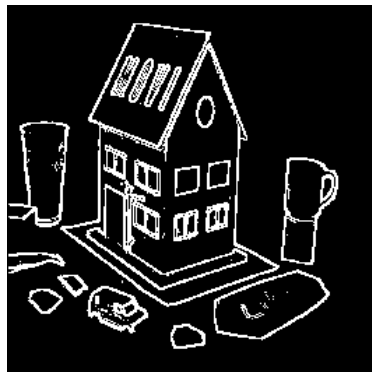
$$M_x = M * H_j = \begin{bmatrix} 0 & 1 & 0 & -1 & 0 \\ 1 & 2 & 0 & -2 & -1 \\ 1 & 3 & 0 & -3 & -1 \\ 1 & 2 & 0 & -2 & -1 \\ 0 & 1 & 0 & -1 & 0 \end{bmatrix}, \quad M_y = M * H_i = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -2 & -3 & -2 & -1 \\ 0 & -1 & -1 & -1 & 0 \end{bmatrix}$$



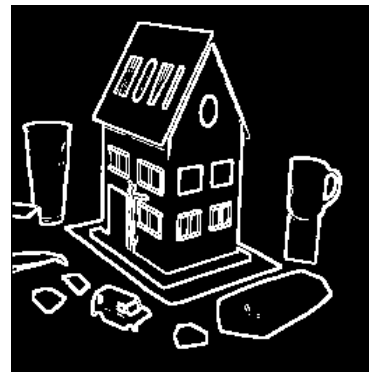
a- Image originale



b- opérateur de Prewitt,



c- Opérateur de Roberts



d- opérateur de Sobel

Figure 5 : opérateurs de dérivation

B. Approche Laplacien :

Les points de contour sont localisés au passage par zéro de Laplacien.

$$\nabla^2 = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

L'extraction de ces passages par zéro s'effectue classiquement en deux étapes[14] :

b-1. Détection des passages par zéro :

On calcule une image I_z des passages par zéro tel que :

$$I_z(M) = 1 \text{ si } M \text{ correspond à une transition } 0-1 \text{ ou } 1-0.$$

$$I_z(M) = 0 \text{ sinon.}$$

b-2. Seuillage des passages par zéro :

L'élimination des passages par zéro de faible norme de gradient peut s'effectuer par l'algorithme de seuillage par hystérésis décrit dans l'approche gradient.

B.1 Opérateurs dérivatifs de second ordre

$$\nabla^2 I(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (2-11)$$

Dans le cas discret, l'opérateur $\frac{\partial^2}{\partial x^2}$ est approximé par :

$$\Delta^2 x = I[i+1, j] - 2I[i, j] + I[i-1, j] \quad (2-12)$$

De même pour l'opérateur $\frac{\partial^2}{\partial y^2}$ donné par :

$$\Delta^2 y = I[i, j+1] - 2I[i, j] + I[i, j-1] \quad (2-13)$$

Le Laplacien sera donc représenté par les masques suivants :

$$L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{ou} \quad L = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

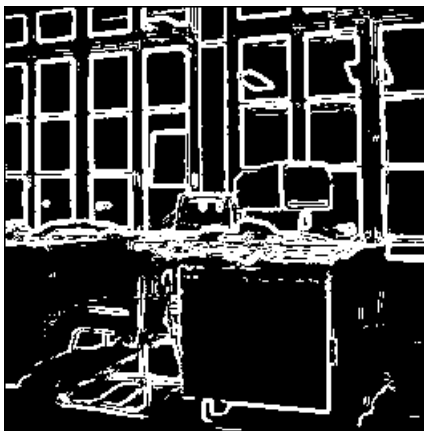
L'algorithme de détection de contour par l'opérateur Laplacien est décrit en annexe.



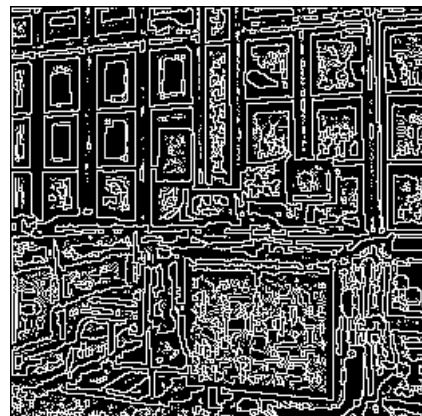
a- Image initiale



b- extraction des extrema locaux



c- Seuillage par hystérésis



d- Passage par zéro de Laplacien

Figure 5: Approche de détection

C. les filtres optimaux

Dans ces approches[7] qui font partie des méthodes de dérivation, un contour est modélisé par un échelon d'amplitude U_0 noyé dans un bruit blanc. Il s'agit alors de calculer le filtre optimal pour détecter et localiser la transition.

1- Critère de Canny

Dans son approche, Canny considère le cas d'une image monodimensionnelle, un contour est une variation brutale du niveau de gris, ce qui peut être représenté par la fonction suivante :

$$U(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases} \quad (2-14)$$

Nous trouvons fréquemment ce type de contours (type marche avec un bruit blanc de moyenne nulle), donnant aussi la formule suivante :

Le signal d'entrée est donné par l'expression :

$$I(x) = U_0 U(x) + b(x) \quad (2-15)$$

$U(x)$: échelon unité

U_0 : amplitude de la marche

$b(x)$: bruit blanc gaussien de moyenne nulle et de variance σ^2 constante.

Le signal de sortie est donné par l'expression :

$$C(x) = I * h(x) = \int_{-\infty}^{+\infty} I(t)h(x-t)dt \quad (2-16)$$

Où $h(x)$ est la réponse impulsionnelle du filtre optimal.

La détection de contour est effectuée en convoluant l'image d'entrée $I(x)$ avec la fonction spatiale $h(x)$. Les contours seront détectés par les maxima de la sortie $C(x)$.

Le problème posé est de trouver $h(x)$ tel que $C(x)$ soit maximum au point $x_0 = 0$ en respectant les trois contraintes suivantes :

- **Bonne détection :**

L'opérateur doit donner une réponse au voisinage d'un contour, ce critère revient à chercher $h(x)$ tel que le rapport signal sur bruit RSB soit maximum.

$$\Sigma = \text{RSB} = \frac{\int_{-\infty}^0 h(x) dx}{\left[\int_{-\infty}^{+\infty} h^2(x) dx \right]^{1/2}} \quad (2-17)$$

- **Bonne localisation :**

Le contour doit être localisé avec précision. Ce critère correspond à la maximisation de l'inverse de la variance $\frac{1}{\sigma^2}$ de la distance entre le maximum de la réponse et la position réelle de la transition. Le critère de localisation λ est donné par :

$$\lambda = \frac{|h'(0)|}{\left[\int_{-\infty}^{+\infty} h^2(x) dx \right]^{1/2}} \quad (2-18)$$

Le produit $\Sigma\lambda$ est un critère qui combine une bonne détection et une bonne localisation, appelé indice de performance.

- **Une réponse unique**

Un contour doit provoquer une seule réponse pour l'opération d'extraction. Ce critère correspond à la limitation du nombre de maxima locaux détectés en réponse à un seul contour. La distance moyenne entre les maxima locaux notée X_{\max} est alors contrainte à l'égalité suivante :

$$X_{\max} = 2\pi \left[\frac{\int_{-M}^{+M} h^2(x) dx}{\int_{-M}^{+M} h''^2(x) dx} \right]^{1/2} = K_s \cdot M \quad (2-19)$$

Maximiser le produit $\Sigma\lambda$ sous la contrainte que le 3^{ème} critère soit fixé à une constante, revient alors à trouver la fonction $h(x)$.

Le filtre de dérivation optimale $h(x)$ proposé par Canny est un filtre à réponse impulsionnelle fini (RIF) ce filtre a pour expression :

$$h(x) = a_1 e^{\alpha x} \sin(wx) + a_2 e^{\alpha x} \cos(wx) + a_3 e^{-\alpha x} \sin(wx) + a_4 e^{-\alpha x} \cos(wx) + c \quad (2-20)$$

Cherchant l'opérateur $h(x)$ sous forme de filtre RIF défini sur l'intervalle $[-M, +M]$ et présentant une pente S à l'origine. Canny a imposé les conditions aux limites suivantes :

$$h(0) = 0, \quad h(+M) = 0, \quad h'(0) = S, \quad h'(+M) = 0. \quad (2-21)$$

Ces quatre conditions aux limites permettent de déterminer les coefficients a_1 à a_4 .

$h(x)$ étant impaire, la solution est étendue aux x négatifs avec $h(x) = -h(-x)$.

Utilisant une optimisation numérique sous contrainte, Canny trouva que la plus grande valeur pouvant être obtenue pour k_s est de 0.58 et qu'elle correspond à un indice de performance $\Sigma\lambda = 1.12$.

L'opérateur trouvé ne présentant pas de simplicité au niveau de la mise en œuvre, Canny l'approxima au vu de sa forme par l'opérateur 1^{ère} dérivée d'une Gaussienne

$$h(x) = -\frac{x}{\sigma^2} \cdot e^{-\frac{x^2}{2\sigma^2}} \quad (2-22)$$

Cette dérivée présente un indice de performance, $\Sigma\lambda = 0.92$ et $k_s = 0.51$, dégradée de 20%.

2- Le filtre de Shen

L'approche développée par Shen consiste en l'élaboration d'un filtre passe-bas optimale pour la détection de contours. Le formalisme mathématique développé s'inspire fortement de celui de Canny.

3- Le filtre de Deriche

L'approche développée par Deriche a consisté en la recherche de l'opérateur optimale sous forme de filtre à réponse impulsionnelle infini (RII). Utilisant la même démarche que Canny.

Les approches développées par Deriche et Shen seront étudiées en détail dans les chapitres suivants.

3.2 Les méthodes surfaciques

En considérant la fonction d'intensité d'une image comme une surface, le contour est maintenant considéré comme une courbure dans un espace à deux dimensions. Puisque cette méthode prend en compte les propriétés géométriques des contours à deux dimensions, elle peut donner la position précise de ces derniers.

La modélisation est souvent réalisée par approximation de l'image par une fonction continue obtenue par filtrage de l'image.

a- Méthode de HUECKEL :

Le principe de l'approche proposé par Huechel [1] est de se donner un modèle idéal de transition représenté par un échelon bidimensionnel U dans un voisinage circulaire D

$$U(x, y, \theta, \rho, b, h) = \begin{cases} b & \text{si } x \cos \theta + y \sin \theta < \rho \\ b+h & \text{si } x \cos \theta + y \sin \theta > \rho \end{cases} \quad (2-23)$$

Soit $I(x, y)$ la fonction image en niveau de gris, le problème est de calculer les paramètres du modèle qui approchent au mieux l'image en un domaine circulaire D centré en (x_0, y_0) . Pour cela, on minimise l'écart quadratique entre l'image et le modèle représenté ci-après :

$$\varepsilon^2 = \iint_D (U(x, y, \theta, \rho, b, h) - I(x, y))^2 dx dy \quad (2-24)$$

ε^2 : L'écart quadratique entre l'image et le modèle

Pour réaliser cette minimisation, Huechel décompose U et $I(x, y)$ sur une base de fonctions orthogonale défini dans le domaine de Fourier. Afin de diminuer le temps de calcul requis par cette technique assez lourde, la présence ou la absence de contour dans le disque de recherche est d'abord testé par un opérateur de type gradient. Si un contour potentiel est détecté l'algorithme de Huechel est mis en œuvre sur une base limitée à seulement huit termes. Une étape finale utilise alors un opérateur de décision basé sur des techniques de seuillage afin de choisir les contours valables.

b- Méthode de Haralick

Cette méthode[8] consiste à approcher la surface représentant une transition par un polynôme dont l'équation sera utilisée pour obtenir l'orientation du contour.

La fonction proposée par Haralick est la suivante :

$$P(x, y) = K_1 + K_2x + K_3y + K_4x^2 + K_5xy + K_6y^2 + K_7x^3 + K_8x^2y + K_9xy^2 + K_{10}y^3 \quad (2-25)$$

Pour un pixel en (x_0, y_0) , on choisit un voisinage carré $S \times S$ pixels et on cherche les 10 paramètres (k_1, \dots, k_{10}) de l'équation ci-dessus en minimisant la fonction suivante :

$$\sum_{i=0}^{S-1} \sum_{j=0}^{S-1} (P(i, j) - I(i + x_0, j + y_0))^2 \quad (2-26)$$

Où $I(i, j)$ est la fonction 'image' en niveaux de gris.

Le gradient de fonction $P(x, y)$ en $(0,0)$ peut être calculé par l'équation :

$$\vec{\nabla}P(0,0) = (k_2, k_3)^T \quad (2-27)$$

L'amplitude du gradient est alors donnée par :

$$|\nabla P(0,0)| = (k_2^2 + k_3^2)^{\frac{1}{2}} \quad (2-28)$$

$$\text{La direction du gradient est : } \Phi = \arctg \frac{k_3}{k_2} \quad (2-29)$$

Pour trouver la position du contour, Haralick calcule le passage par zéro dans la direction du gradient :

$$\frac{\partial^2 P}{\partial n^2} = \frac{\partial^2 P}{\partial x^2} \cos^2 \Phi + 2 \frac{\partial^2 P}{\partial x \partial y} \cos \Phi \sin \Phi + \frac{\partial^2 P}{\partial y^2} \sin^2 \Phi \quad (2-30)$$

Où n est la direction du gradient : $\vec{n} = x \cos \Phi + y \sin \Phi$ (2-31)

$$\cos \theta = \frac{k_2}{\sqrt{k_2^2 + k_3^2}} \quad \text{et} \quad \sin \theta = \frac{k_3}{\sqrt{k_2^2 + k_3^2}} \quad (2-32)$$

Le pixel (x_0, y_0) est alors déclaré point de contour si $\frac{\partial^2 P}{\partial n^2}$ passe par zéro en $(0,0)$ et si $|\nabla P(0,0)|$ est supérieur à un seuil.

Cette méthode donne une position de contour d'une manière plus précise que les méthodes de filtrage. Néanmoins elle demande trop de calculs. Pour cette raison, Haralick a proposé une composition de la fonction $P(x, y)$ par une base de polynôme orthogonaux discrets, ce qui permet de calculer les paramètres du modèle avec des masques spécifiques.

3.3 Méthodes morphologique

a-Gradient morphologique

L'approche morphologique [1] consiste à considérer l'image non comme une fonction bidimensionnelle mais plutôt comme un espace de représentation d'objet ayant les propriétés ensemblistes.

Ces méthodes travaillent sur les maximums et les minimums des intensités de voisinage, par exemple un contour de type « marche » sera détecté si la différence entre le maximum et le minimum est important.

Donc après l'opération de dilatation et d'érosion qu'on a déjà expliquée dans le chapitre précédent, on calcule le gradient morphologique qui donne une approximation du module du vecteur gradient en chaque point.

Le gradient dans l'espace discret est donné par :

$$|\nabla A[i, j]| = \frac{(A \oplus C_x) - (A \ominus C_x)}{2} \quad (2-33)$$

La procédure consiste à dilater et éroder l'image en niveaux de gris puisque le gradient s'interprète comme l'étendue, c'est-à-dire la différence entre le max et le min pour chaque pixel.

4. suivi de contours

Dans une image, le choix du seuil [13] définit un compromis entre les vrais et les faux points de contours. La raison de cette limitation des algorithmes de détection de contours réside dans la nature locale des opérateurs. En effet lorsque le bruit d'une image est très important il est nécessaire de définir aussi des critères globaux. Ces critères s'expriment généralement par une fonction à optimiser un ensemble d'attributs locaux. On choisit de rechercher dans l'image des chemins optimisant la somme des normes de gradient calculés en chaque point à l'aide d'un opérateur local. On ajoute aussi un critère de forme de contour minimisant les variations de courbure et la longueur des contours (minimisation des courbures).

5. Modélisation des contours

Il est nécessaire de regrouper les points de contours obtenus après seuillage en chaînes de contours. En effet les points de contours représentent une grande quantité d'information qui n'est pas structurée et donc inutilisable directement. Une description plus intéressante est structurée par l'ensemble des segments de contours représentant une partie du contour de l'objet

Pour obtenir cette description, deux étapes sont nécessaires :

- Chaînage des points de contours
- Modélisation des chaînes par approximation généralement polygonale.

5.1 Chaînage des points de contours

Le chaînage des points de contours [13] consiste à regrouper les points de contours dans des chaînes de contours. Il permet de diminuer la quantité d'information puisque l'on passe d'une représentation matricielle des contours à une représentation sous forme de listes chaînées.

5.2 Modélisation de contour

Afin de diminuer la quantité d'information, les chaînes de contours sont généralement modélisées en segments par un algorithme d'approximation polygonale. D'autre type de méthodes consiste à modéliser les chaînes par des arcs de cercle.

Discussions

Nous avons présenté dans ce chapitre de façon succincte les méthodes de détection de contours les plus utilisées. Nous constatons que les méthodes dérivatives font appel au produit de convolution, alors que les méthodes surfaciques utilisent les propriétés géométriques des contours. Comparaison en fonction de leurs résultats. Les premières permettent d'obtenir des contours fermés, peu bruités et des lignes fines. Cela en faisant un choix en fonction de la propriété (rapport signal sur bruit, localisation, unicité de la réponse), adaptés aux images non texturées et les secondes conduisent à des contours moins précis en faisant un choix en fonction des critères topologiques pour définir des contours de région dans une image, donc moins adaptés aux images non texturées.

Dans les prochains chapitres, nous étudierons de façon détaillée les techniques de Deriche et Shen qui sont l'objet de notre mémoire.

1. Préambule

Deux raisons au moins conduisent à utiliser le filtre de Deriche en détection de contours. D'une part, sa qualité de détection en présence du bruit et d'autre part, la possibilité de traitement multi résolutions. Dans ce chapitre, nous décrirons de façon détaillée les opérateurs monodimensionnels et bidimensionnels de Deriche ainsi que leurs implémentations.

2. Opérateur monodimensionnel de Deriche

Rappelons que Deriche [1] a déterminé un filtre dérivateur optimal vis-à-vis des trois critères définis par Canny (détection, localisation, non multiplicité de réponses) étendus au cas des filtres à réponse impulsionnelle infini RII. La forme récursive de ce type de filtre est largement utilisée pour les implantations logicielles.

Deriche a utilisé la même démarche que Canny pour chercher une réalisation de l'opérateur sous la forme d'un filtre à réponse impulsionnelle infinie (RII). Il a abouti à la même équation différentielle. Seules les conditions aux limites sont différentes et deviennent :

$$(M \rightarrow +\infty): h(0) = 0 \quad ; h(+\infty) = 0 \quad ; h'(0) = 0 \quad ; h'(+\infty) = 0.$$

La solution est alors :

$$h(x) = ce^{-\alpha|x|} \sin wx \quad (3-1)$$

En évaluant, pour cet opérateur, les différentes intégrales intervenant dans les calculs des critères de performance, on obtient les résultats suivants :

$$\lambda = \sqrt{2\alpha} \quad (\text{localisation}) \quad \Sigma = \sqrt{\frac{2\alpha}{\alpha^2 + w^2}} \quad (\text{détection}) \quad (3-2)$$

$$\Sigma\lambda = \frac{2\alpha}{\sqrt{\alpha^2 + w^2}} \quad (\text{Indice de performance}) \quad ks = \sqrt{\frac{\alpha^2 + w^2}{5\alpha^2 + w^2}} \quad (3-3)$$

En posant $\alpha = mw$, on obtient les trois cas suivants :

$$a) \quad m \gg 1 \quad ; \quad \lambda = \sqrt{2\alpha} \quad \Sigma = \sqrt{\frac{2}{\alpha}} \rightarrow \lambda\Sigma = 2 \quad ; \quad ks = 0.44 \quad (3-4)$$

$$b) \quad m=1 \quad ; \quad \lambda=\sqrt{2\alpha} \quad \Sigma = \sqrt{\frac{1}{\alpha}} \rightarrow \lambda\Sigma = \sqrt{2} \quad ; \quad ks = 0.58 \quad (3-5)$$

$$c) \quad m=\sqrt{3} \quad ; \quad \lambda=\sqrt{2\alpha} \quad \Sigma = \sqrt{\frac{3}{2\alpha}} \rightarrow \lambda\Sigma = \sqrt{3} \quad ; \quad ks = 0.5 \quad (3-6)$$

➤ Le cas (c) montre que pour $ks = 0.5$, l'indice de performance $\lambda\Sigma$ est de 1.73 , l'opérateur de Deriche est nettement meilleur que celui obtenu avec la dérivée première de la gaussienne qui donnait $\lambda\Sigma = 0.92$ et $ks = 0.51$.

➤ Le cas (b) montre que pour $ks = 0.58$, l'opérateur de Deriche présente un indice de performance $\lambda\Sigma = 1.41$ amélioré de près de 25% par rapport à l'opérateur optimal de Canny qui donnait $\lambda\Sigma = 1.12$ et $ks = 0.58$.

➤ Le cas (a) présente le meilleur indice de performance $\lambda\Sigma = 2$ pour $ks = 0.44$, il correspond à la limite de l'opérateur de Deriche pour w tendant vers zéro. Il est facile de vérifier que cette limite correspond à l'opérateur $h(x)$ donné par :

$$h(x) = c x e^{-\alpha|x|} \quad (3-7)$$

Avec :

α : étant le paramètre de résolution.

c : Constante de normalisation calculée pour avoir un maximum de la réponse égale à 1 en $x=0$. Pour un échelon $U(x)$. On obtient :

$$c \int_{-\infty}^0 x e^{-\alpha x} dx = 1 \rightarrow c = -\alpha^2 \quad (3-8)$$

En effet, cet opérateur est la solution de l'équation différentielle :

$$2h(x) - 2\lambda_1 h'(x) + 2\lambda_2 h''(x) + \lambda_3 = 0 \quad (3-9)$$

Dans le cas où le discriminant de son équation caractéristique est nul.

L'opérateur $h(x)$ présente donc un indice de performance bien meilleur que celui donnée par Canny. Il ne présente qu'un seul paramètre à régler correspondant à la résolution α à laquelle les contours sont à détecter. Diminuer α permet de favoriser la détection au détriment de la localisation et vice versa .De plus, il possède une particularité de mise en œuvre très intéressante. En effet, il peut être implémenté de manière récursive avec un

nombre d'opérations extrêmement réduit et indépendant de la résolution α à laquelle on opère.

3. Opérateurs bidimensionnels de dérivation et de lissage de Deriche

En utilisant l'opérateur optimal de Deriche, on peut mettre en œuvre les critères énoncés précédemment. Pour ce faire, on opère un lissage pour améliorer l'immunité au bruit puis on calcule le gradient de chaque point de l'image.

3.1 Lissage

Le filtre utilisé est la combinaison de deux filtres monodimensionnels dans les directions x et y .

Le filtre de lissage monodimensionnel $f(x)$ retenu par Deriche [6] est la dérivée du filtre optimal $h(x)$ équation (3-7). Le filtre de lissage est donné par :

$$f(x) = b(\alpha|x| + 1)e^{-\alpha|x|} \quad (3-10)$$

b est calculé de manière à avoir une réponse constante de valeur 1 pour un signal d'entrée constant de niveau 1. On obtient dans ce cas $b = \frac{\alpha}{4}$.

➤ L'expression du filtre bidimensionnel séparable de lissage $f(x, y)$ est illustrée par la figure 7-a.

$$\text{Sachant que } f(x, y) = f(x).f(y) \quad (3-11)$$

Le filtre de lissage s'écrit alors sous forme :

$$f(x, y) = b^2(\alpha|x| + 1)e^{-\alpha|x|} . (\alpha|y| + 1)e^{-\alpha|y|} \quad (3-12)$$

Si l'image originale est notée $A(x, y)$, l'image lissée aura pour expression :

$$B(x, y) = A(x, y) * f(x, y) \quad (3-13)$$

Ou encore

$$B(x, y) = A(x, y) * f(x, y) = A(x, y) * (f(x).f(y)) = (A(x, y) * f(x)) * f(y) \quad (3-14)$$

3.2 Calcul du gradient :

Le calcul du gradient se fait à partir des dérivées selon x et y du produit de convolution de l'image par le filtre de lissage $f(x,y)$:

Compte tenu des règles de dérivation, de l'opération de convolution et de la séparabilité du filtre $f(x,y)$, nous avons :

$$\frac{\partial B}{\partial x}(x, y) = B_x(x, y) = \frac{\partial \{A * f\}}{\partial x}(x, y) = A * \frac{\partial f}{\partial x}(x, y) \quad (3-15)$$

Les opérateurs de dérivation suivant x et y sont illustrés par la figure 7-b, ils sont de la forme :

$$f_x(x, y) = \eta x e^{-\alpha|x|} \cdot (\alpha|y| + 1)e^{-\alpha|y|} \quad (3-16)$$

$$f_y(x, y) = \eta x e^{-\alpha|y|} \cdot (\alpha|x| + 1)e^{-\alpha|x|} \quad (3-17)$$

Où η est une constante de normalisation calculée, par exemple : pour l'équation (13) de façon à fournir un maximum d'amplitude 1 en réponse à une transition unitaire. On trouve alors que

$$\eta = \frac{-\alpha^3}{4}; \text{ on en conclut que } f(x, y) = h(x) \cdot f(y)$$

Par conséquent, les images des dérivées directionnelles en x et y s'écrivent :

$$B_x(x, y) = [A * f(y)] * h(x) \quad (3-18)$$

$$B_y(x, y) = [A * f(x)] * h(y) \quad (3-19)$$

C'est-à-dire que la dérivée directionnelle selon x est le résultat d'un lissage suivant la direction y, suivi par une dérivation suivant x.

De même, la dérivée directionnelle selon y est le résultat d'un lissage suivant la direction x, suivi par une dérivation suivant y.

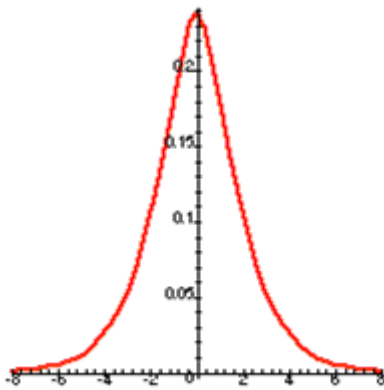
Les filtres sont monodimensionnels, l'implémentation récursive est alors facile à mettre en œuvre.

L'extraction de contours par l'opérateur gradient de Deriche se fait selon les étapes suivantes :

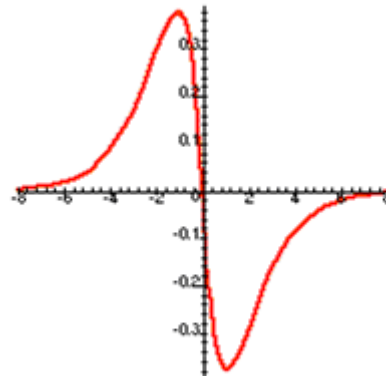
- Lissage.
- Calcul du gradient en chaque point de l'image.
- Création de l'image de la norme du gradient.
- Extraction des maxima locaux (obtention de contours fins), pour chaque point de l'image de la norme, dans la direction exacte du gradient.

- Seuillage par Hystérésis de l'image des maxima locaux : on seuille d'abord avec un seuil haut et on conserve ensuite toutes les chaînes ayant au moins un point dans la norme du gradient est supérieur au seuil bas.

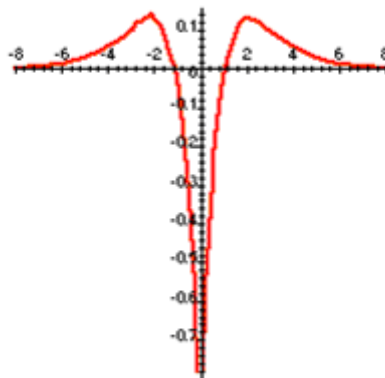
Notons que le seuillage par Hystérésis est utilisé pour éliminer les transitions de faible gradient pouvant correspondre à un bruit.



a - Filtre de lissage de Deriche



b- filtre dérivée première de Deriche



c- Filtre dérivé seconde de Deriche

Figure 7 : lissage et dérivation par le filtre de Deriche

3.3 Opérateur Laplacien de Deriche :

Il est possible de calculer le Laplacien [1] pour la détection de contours. En effectuant le calcul sur l'image lissée, on obtient :

$$\Delta B(x, y) = A * \Delta f(x, y) \quad (3-20)$$

Comme :
$$\Delta f(x, y) = \frac{\delta f^2_x}{\delta^2 x} + \frac{\delta f^2_y}{\delta^2 y} \quad (3-21)$$

En utilisant les formules (3-16) et (3-17), l'opérateur Laplacien de Deriche s'exprime sous la forme d'un opérateur bidimensionnel qui est la différence de deux fonctions de transfert séparables.

$$\Delta f(x, y) = e^{-\alpha|x|} e^{-\alpha|y|} - k\alpha|x|e^{-\alpha|x|} \cdot \alpha|y|e^{-\alpha|y|} \quad (3-22)$$

Le premier terme correspond à un filtre de lissage, le deuxième, à un filtre dérivateur.

➤ Le calcul du Laplacien d'une image $A(x,y)$ est obtenu par une simple soustraction entre deux images « traitées ». il peut alors être mis en œuvre de manière totalement récursive. Cette technique est très efficace car elle permet d'obtenir le Laplacien d'une image à n'importe quelle résolution spécifiée par le paramètre α .

➤ Pour l'opérateur Laplacien, la démarche utilisée est la suivante :

- Calcul du Laplacien.
- Chercher les passages par zéro du Laplacien.
- Création de l'image des passages par zéro affectés à la norme du gradient.
- Seuillage par hystérésis (élimination des passages par zéro non significatifs).

Cette technique nécessite le calcul du gradient.

4. Implémentation des opérateurs de Deriche :

Le filtre de Deriche possède une réponse impulsionnelle infinie dont l'expression analytique permet une implémentation récursive exacte [11]. Ces implémentations sont de type « parallèle ».

4.1 Implantation des opérateurs monodimensionnels de Deriche

4.1.1 Dérivation

On développe dans cette partie la procédure permettant d'implémenter de manière récursive l'opérateur de détection, ainsi que sa dérivée qui va servir comme opérateur de lissage dans les opérateurs bidimensionnels.

Une mise en œuvre récursive du filtre optimal de dérivation peut être obtenue en employant la transformée en Z:

Soit $h[i]$ la séquence discrète résultant de l'échantillonnage de $h(x)$ et $h(z)$ sa transformée en Z.

$$h(x) = c x e^{-\alpha|x|} \tag{3-23}$$

$$h(z) = \sum_{i=-\infty}^{+\infty} h[i]Z^{-i} \quad i = -\infty \dots \dots \dots + \infty \tag{3-24}$$

La séquence $Z[i]$ est mise sous la forme d'une somme d'une séquence causale $h_-[i]$ et d'une séquence non causale $h_+[i]$.

$$h[i] = h_-[i] + h_+[i] \tag{3-25}$$

Avec :

$$h_-[i] = \begin{cases} c i e^{-\alpha i} \text{ pour } i \geq 0 \\ 0 \text{ pour } i < 0 \end{cases} \tag{3-26}$$

$$h_+[i] = \begin{cases} 0 \text{ pour } i \leq 0 \\ c i e^{\alpha i} \text{ pour } i < 0 \end{cases} \tag{3-27}$$

La transformée $h(z)$ a pour expression :

$$h[z] = h_-[z^{-1}] + h_+[z] \tag{3-28}$$

Avec :

$$h_-[z^{-1}] = \frac{c e^{-\alpha} z^{-1}}{1 - 2e^{-\alpha} z^{-1} + e^{-2\alpha} z^{-2}} \tag{3-29}$$

$$h_+[z] = \frac{-c e^{-\alpha} z}{1 - 2e^{-\alpha} z + e^{-2\alpha} z^2} \tag{3-30}$$

$$h_-[z^{-1}] \text{ Converge pour } |e^{-\alpha} z^{-1}| < 1$$

$$h_+[z] \text{ Converge pour } |e^{-\alpha} z| < 1$$

Comme toutes les singularités de $h_+(z)$ sont à l'intérieur et les singularités $h_-(z^{-1})$ sont à l'extérieur du cercle unité pour α réel positif, ces deux transformations en z correspondent à 2 fonctions de transfert de filtres récursifs stables du second ordre [5].

Le premier terme $h_-(z^{-1})$ opérant de gauche à droite et le deuxième $h_+(z)$ de droite à gauche.

Le coefficient c est choisi pour obtenir une réponse d'amplitude maximale égale à 1 en réponse à un échelon unitaire, ce qui donne :

$$B[i] = \sum_{m=-\infty}^{+\infty} U[I-m] h[m] = 1 \quad \text{pour } i=1 \quad (3-31)$$

$$B[0] = -c \sum_{m=0}^{+\infty} m e^{-\alpha m} = 1 \quad \rightarrow \quad c = -\frac{(1-e^{-\alpha})^2}{e^{-\alpha}} \quad (3-32)$$

On peut remarquer que cette valeur de la constante de « normalisation » est différente de celle calculée pour un filtre connu. $c = -\alpha$

On obtient alors les équations aux différences suivantes :

$$B_1[i] = c e^{-\alpha} A[i-1] + 2e^{-\alpha} B_1[i-1] - e^{-2\alpha} B_1[i-2] \quad (3-33)$$

pour $i = 1, \dots, M$

$$B_2[i] = -c e^{-\alpha} A[i+1] + 2e^{-\alpha} B_2[i+1] - e^{-2\alpha} B_2[i+2] \quad (3-34)$$

pour $i = M, \dots, 1$

M : étant la longueur de la séquence d'entrée $A[i]$.

La séquence de sortie est alors donnée par :

$$B[i] = B_1[i] + B_2[i] \quad (3-35)$$

➤ L'intérêt de cette mise en œuvre récursive est le faible nombre d'opérations : 5 multiplications et 5 additions pour l'opérateur dérivé première.

➤ Le nombre d'opérations est indépendant de la valeur de la résolution (α) utilisée pour détecter les contours. En effet, la forme du filtre liée à α peut varier mais le nombre d'opérations par point reste identique.

➤ A titre de comparaisons, une implantation RIF de l'opérateur $h[n]$ à l'aide de $(2N+1)$ coefficients non nuls, requiert $(2N+1)$ opérations par point de sortie. Un gain de $\frac{2N+1}{5}$ en résulte en faveur de la mise en œuvre récursive qui présente en plus l'avantage de ne pas introduire de déformation par troncature de la réponse implutionnelle [4]. par exemple, une implantation de cette opération de détection sous forme RIF pour $\alpha = 0.5$ et $\alpha = 0.25$ nécessite respectivement de l'ordre de 50 et 100 opérations par point.

4.1.2 Lissage

En adoptant la même démarche que dans le paragraphe précédent, le filtre de lissage :

$$f(x) = b(\alpha|x| + 1)e^{\alpha|x|} \quad (3-36)$$

Peut être implanté de manière récursive.

La transformée en Z du filtre numérique correspondant décomposé en partie causale et anti causale s'écrit :

$$f[z] = f_-[z^{-1}] + f_+[z] \quad (3-37)$$

Avec :

$$f_-[z^{-1}] = b \frac{1 + e^{-\alpha}(\alpha - 1)z^{-1}}{1 - 2e^{-\alpha}z^{-1} + e^{-2\alpha}z^{-2}} \quad (3-38)$$

$$f_+[z] = b \frac{e^{-\alpha}(\alpha + 1)z - e^{-2\alpha}z^2}{1 - 2e^{-\alpha}z + e^{-2\alpha}z^2} \quad (3-39)$$

Ayant le même dénominateur que $h_+(z)$ et $h_-(z)$ ces deux transformées en z correspondent de la même manière à deux fonctions de transfert de filtre récursif stable du second ordre, le premier opérant de gauche à droite (f_+) et le second de la droite vers la gauche (f_-). La sortie $B[i]$ résulte de la convolution de la séquence d'entrée $A[i]$ avec la réponse impulsionnelle $h[i]$.

Les domaines de convergence sont les mêmes que précédemment.

Les équations aux différences correspondantes ont pour expressions :

$$B_1[i] = bA[i] - be^{-\alpha}(\alpha - 1)A[i - 1] + 2e^{-\alpha}B_1[i - 1] - e^{-2\alpha}B_1[i - 2] \quad (3-40)$$

pour $i = 1, \dots, M$

$$B_2[i] = be^{-\alpha}(\alpha + 1)A[i + 1] - be^{-2\alpha}A[i + 2] + 2e^{-\alpha}B_2[i + 1] - e^{-2\alpha}B_2[i + 2] \quad (3-41)$$

pour $i = M, \dots, 1$

$$B[i] = B_1[i] + B_2[i] \quad (3-42)$$

- Pour réaliser un lissage il faut donc 8 multiplications et 7 additions, indépendamment du facteur α .

• La constante b est calculée pour obtenir une réponse d'amplitude 1 à une entrée constante unitaire :

$$b = \frac{(1 - e^{-\alpha})^2}{1 + 2\alpha e^{-\alpha} - e^{-2\alpha}} \quad (3-43)$$

Cette valeur est différente de celle trouvée en continu : $b = \frac{\alpha}{4}$.

4.2 Implémentation de l'opérateur gradient bidimensionnel de Deriche

Dans le cas bidimensionnel, on crée deux masques $h_x(x, y)$ et $h_y(x, y)$ Selon les directions horizontale et verticale. Une implémentation totalement récursive en résulte. En appliquant la relation (3-10) on obtient alors:

$$h_x(x, y) = -b \alpha^2 x e^{-\alpha|x|} \cdot b(\alpha|y| + 1)e^{-\alpha|y|} \quad (3-44)$$

$$h_y(x, y) = -b \alpha^2 y e^{-\alpha|y|} \cdot b(\alpha|x| + 1)e^{-\alpha|x|} \quad (3-45)$$

Du fait de leurs séparabilité les masques $h_x(x, y)$ et $h_y(x, y)$ peuvent être efficacement implémentés de manière récursive. Pour ce faire, on applique un filtre de lissage à l'image $A(x, y)$ dans la direction verticale puis, on convolu l'image lissée avec le masque $h_x(x, y)$. On opère de la même manière pour la direction horizontale en utilisant un filtre de lissage horizontal et on convolu avec le masque $h_y(x, y)$.

Cette mise en œuvre [4] permet de convoluer l'image d'entrée $A(x, y)$ avec le masque infini $h_x(x, y)$ en seulement 13 multiplications et 12 additions par point indépendamment de la taille du filtre spécifié par le paramètre α . De même pour la convolution de $A(x, y)$ avec le masque $h_y(x, y)$

En pratique pour une image numérique $A(i, j)$ où i représente l'indice ligne et j l'indice colonne, Les images résultats $B_i[i, j]$ et $B_j[i, j]$ B sont donnée par les relation (3-41) et (3-42).

$$B_i[i, j] = [A * f(j)] * h(i) \quad (3-46)$$

$$B_j[i, j] = [A * f(i)] * h(j) \quad (3-47)$$

Les étapes à suivre pour extraire les contours d'une image $A(i, j)$ sont résumées ci après :

- a). mise en œuvre récursive de $A(i, j) * h_x(x; y)$ pour obtenir l'image gradient directionnel selon X : $B_i(i, j)$.
- b). mise en œuvre récursive de $A(i, j) * h_y(x; y)$ pour obtenir l'image gradient directionnel selon Y : $B_j(i, j)$.
- c). calcul de la norme $B(i, j)$ et de la direction du gradient $D(i, j)$ en fonction $B_i(i, j)$ et $B_j(i, j)$:

$$B(i, j) = \left[B_i(i, j)^2 + B_j(i, j)^2 \right]^{1/2} \quad (3-48)$$

$$D(i, j) = \arctg\left(\frac{B_j(i, j)}{B_i(i, j)}\right) \quad (3-49)$$

- d). Extraction des maxima locaux pour chaque point de l'image $B(i, j)$ dans la direction exacte du gradient $D(i, j)$. Pour l'obtention de l'image des maxima locaux $R(i, j)$, on effectue une interpolation linéaire entre le pixel précédent et le pixel suivant selon la direction choisie.
- e). Seuillage par hystérésis de l'image des maxima locaux $R(i, j)$, obtenue à l'étape d). On seuille $R(i, j)$ avec un seuil S_b et on élimine tous les segments n'ayant pas au moins un point dont la norme est supérieur à un plus grand seuil, s_h .

Algorithme de détection de contour pour le filtre optimale de Deriche

Variabes utilisées

- A : L'image en entrée.
- B : L'image en sortie.
- B_x^+ : La réponse positive en X.
- B_y^+ : La réponse positive en Y.
- B_x^- : La réponse négative en X.
- B_y^- : La réponse négative en Y.
- L_x : La réponse totale en X (lissage).
- L_y : La réponse totale en Y (lissage).
- H : la longueur de l'image.
- W : la largeur de l'image.
- B_1 : La réponse totale en X (dérivation).
- B_2 : La réponse totale en Y (dérivation).

Filtre de dérivation de Deriche :Début Lire (α) ;

$$c = -\frac{(1-e^{-\alpha})}{e^{-\alpha}}$$

Calcul de (c) ;

Pour i allant de 0 à H-1 faire

$$B_x^+[i, 1] := ce^{-\alpha} * L_y[i, 0] ;$$

Pour j allant de 2 à W-1 Faire

$$B_x^+[i, j] := ce^{-\alpha} * L_y[i, j-1] - 2e^{-\alpha} * B_x^+[i, j-1] - e^{-2\alpha} * B_x^+[i, j-2] ;$$

Fait

Fait

Pour i allant de 0 à H-1 faire

$$B_x^-[i, W-1] := 0 ;$$

$$B_x^-[i, W-2] := -e^{-\alpha} * L_y[i, W-1] ;$$

Pour j allant de W-3 à 0 Faire

$$B_x^-[i, j] := -ce^{-\alpha} L_x[i, j+1] + 2e^{-\alpha} * B_x^-[i, j+1] - e^{-2\alpha} * B_x^-[i, j+2] ;$$

Fait

fait

Pour i allant de 0 à H-1 faire

Pour j allant de 0 à W-1 Faire

$$B_1[i, j] := B_1^+[i, j] + B_1^-[i, j]$$

Fait

Fait

Pour j allant de 0 à W-1 Faire

$$B_y^+[1, j] := ce^{-\alpha} * L_y[0, j] ;$$

Pour i allant de 2 à H-1 Faire

$$B_y^+[i, j] := ce^{-\alpha} * L_y[i-1, j] + 2e^{-\alpha} * L_y[i-1, j] - e^{-2\alpha} * B_y^+[i-2, j] ;$$

Fait

Fait

Pour i allant de 0 à H-1 faire

$$B_y^-[H-1, j] := 0 ;$$

$$B_y^-[H-2, j] := -ce^{-\alpha} * L_y[H-1, j] ;$$

Pour i allant de H-3 à 0 Faire

$$B_y^-[i, j] := -ce^{-\alpha} * L_y[i, j+1] + 2e^{-\alpha} * B_y^-[i, j+1] - e^{-2\alpha} * B_y^-[i, j+2] ;$$

Fait

Fait

Pour i allant de 0 à H-1 Faire

pour i allant de 0 à W-1 Faire

$$B_2[i, j] := B_2^+[x, y] + B_2^-[x, y]$$

Fait

Fait

Pour i allant de 0 à H-1 Faire

pour i allant de 0 à W-1 Faire

$$B[i, j] := [(B_1[i, j])^2 + (B_2[i, j])^2]^{1/2}$$

Fait

Fait

Fin

Filtre de lissage de Deriche :Début Lire (α) ;

$$b = -\frac{(1-e^{-\alpha})^2}{1+2e^{-\alpha}-e^{-2\alpha}}$$

Calcul de b ;

Pour i allant de 0 à H-1 faire

$$B_x^+[i, 0] := b * A[i, 0] ;$$

$$B_x^+[i, 1] := b * A[i, 1] + be^{-\alpha}(\alpha - 1) * A[i, 0];$$

Pour j allant de 2 à W-1 Faire

$$B_x^+[i, j] := -b * A[i, j] + be^{-\alpha}(\alpha - 1) * A[i, j - 1] + 2e^{-\alpha} * B_x^+[i, j - 1] - 2e^{-\alpha} * B_x^+[i, j - 2];$$

Fait

Fait

Pour i allant de 0 à H-1 faire

$$B_x^-[i, W - 1] := be^{-\alpha}(\alpha + 1) * A[i, W - 2];$$

$$B_x^-[i, W - 2] := be^{-\alpha}(\alpha + 1) * A[i, W - 2] - be^{-\alpha} * A[i, W - 1] ;$$

Pour j allant de W-3 à 0 Faire

$$B_x^-[i, j] := be^{-\alpha}(\alpha + 1) * A[i, j + 1] - be^{-\alpha} * A[i, j + 2] + 2e^{-\alpha} * B_x^-[i, j + 1] - e^{-2\alpha} * B_x^-[i, j + 2];$$

Fait

fait

Pour i allant de 0 à H-1 faire

Pour j allant de 0 à W-1 Faire

$$L_x[i, j] := B_x^+[i, j] + B_x^-[i, j]$$

Fait

Fait

Pour j allant de 0 à W-1 Faire

$$B_y^+[0, j] := b * A[0, j] ;$$

Pour i allant de 2 à H-1 Faire

$$B_y^+[i, j] := b * A[1, j] + be^{-\alpha}(\alpha - 1) * A[0, j];$$

Pour i allant de 2 à H-1 Faire

$$B_y^+[i, j] := b * A[i, j] + be^{-\alpha}(\alpha - 1) * A[i - 1, j] + 2e^{-\alpha} * B_y^+[i - 1, j] - 2e^{-\alpha} * B_y^+[i - 2, j];$$

Fait

Fait

Pour i allant de 0 à H-1 faire

$$B_y^-[H - 1, j] := 0;$$

$$B_y^-[H - 2, j] := be^{-\alpha}(\alpha + 1) * A[H - 1, j];$$

Pour i allant de H-3 à 0 Faire

$$B_y^-[i, j] = be^{-\alpha}(\alpha + 1) * A[i + 1, j] - be^{-2\alpha} * A[i + 2, j] + e^{-2\alpha} * B_y^-[i + 1, j] - e^{-2\alpha} * B_y^-[i + 2, j];$$

Fait

Fait

Pour i allant de 0 à H-1 Faire

pour i allant de 0 à W-1 Faire

$$L_y[i, j] := B_y^+[x, y] + B_y^-[x, y]$$

Fait

Fait

1. Préambule

Shen a adopté la même démarche que celle de Deriche, à savoir un lissage par des filtres passe-bas avant l'application d'un opérateur différentiel pour mettre en évidence les transites. Dans ce chapitre, nous décrirons de façon détaillée les opérateurs monodimensionnels et bidimensionnels de Shen ainsi que leurs implémentations.

2. Modèle de contour et de bruit

Shen modélise la frontière par un échelon $U(x)$ d'amplitude U_0 noyé dans un bruit blanc stationnaire additionnel $N(x)$ de moyenne nulle ($E\{N(x)\} = 0$) et de densité spectrale N_0^2 ($E\{N^2(x)\} = N_0^2$).

Le signal d'entrée $A(x)$ a donc pour équation :

$$A(x) = U_0 U(x) + N(x) \quad (4-1)$$

3. Critère de performance

D'une façon similaire à celle de Canny, Shen a défini un critère permettant de quantifier les performances d'un détecteur de contours en termes de détection et de localisation.

Il utilise la réponse impulsionnelle étendue infinie $h(x)$ du filtre de lissage associé au détecteur de contours pour déterminer ce critère.

Le signal de sortie du filtre de lissage est donné par l'équation (4-2) :

$$B(x) = A(x) * h(x) = U_0 U(x) * h(x) + N(x) * h(x) \quad (4-2)$$

Le premier terme de l'équation (4-2) est la réponse du filtre au signal d'entrée non bruité, et le second la réponse du filtre au bruit seul.

Pour déterminer l'énergie du contour dans le signal de sortie, Shen a utilisé la dérivée de la réponse du filtre de lissage qui équivaut à la réponse du détecteur de contour appliqué au signal. Cette dérivée est donnée par l'équation (4-3) :

$$C(x) = \frac{dB(x)}{dx} = U_0 \delta(x) * h(x) + N(x) * \frac{dh(x)}{dx} = U_0 h'(0) + N(x) * h'(x) \quad (4-3)$$

Où δ est la distribution de Dirac.

Pour bien détecter la transition Shen a choisi les trois contraintes suivantes : [5]

a-Minimiser l'énergie du bruit E_N du signal de sortie lissée $B(x)$. Cette énergie est donnée par l'équation (4-4) :

$$E_N = E \left\{ \left[N(x) * h(x) \right]^2 \right\} = N_0^2 \cdot \int_{-\infty}^{\infty} h^2(x) dx \quad (4-4)$$

Où N est relatif au bruit.

b- Minimiser l'énergie du bruit de la dérivée première $C(x)$. Cette énergie est donnée par l'équation (4-5) :

$$E_{N'} = E \left\{ \left[\left(\frac{d}{dx} \right) N(x) * h(x) \right]^2 \right\} = N_0^2 \cdot \int_{-\infty}^{\infty} h'^2(x) dx \quad (4-5)$$

Où N' est relatif au bruit de la dérivée première $C(x)$.

c- Maximiser l'énergie E_U à la position du contour ($x=0$) de la dérivée première $C(x)$.

E_U est donnée par la relation (4-6):

$$E_U = U_0^2 h^2(0) \quad (4-6)$$

Le critère de performance est alors donné par le rapport C tel que :

$$C = \sqrt{\frac{E_N E_{N'}}{E_U^2}} = \frac{N_0^2 \sqrt{\left(\int_{-\infty}^{\infty} h^2(x) dx \right) \left(\int_{-\infty}^{\infty} h'^2(x) dx \right)}}{U_0^2 h^2(0)} \quad (4-7)$$

4. Filtre optimal de Shen [1]

4.1 Cas monodimensionnel

4.1.1 Opérateur de lissage

En minimisant le rapport C donné en (4-7) et en tenant compte des conditions aux limites ($\lim_{x \rightarrow +\infty} h(x) = 0$). Shen a proposé un filtre optimal de lissage $h(x)$ sous forme d'un filtre exponentiel symétrique (voir figure 1-a):

$$h(x) = be^{-\beta|x|} \quad (4-8)$$

β : étant le paramètre de résolution.

b est une constante choisi de manière à obtenir un filtre normalisé définie par :

$$\int_{-\infty}^{\infty} h(x) dx = 1 \text{ soit } b = \frac{\beta}{2}$$

Le filtre de lissage normalisé est donné par l'équation (4-9)

$$h(x) = \frac{\beta}{2} e^{-\beta|x|} \quad (4-9)$$

$h(x)$ peut se mettre sous la forme :

$$h(x) = h_1(x) * h_2(x) \quad (4-10)$$

$$h_1(x) = \begin{cases} \beta e^{-\beta x} & \text{pour } x \geq 0 \\ 0 & \text{pour } x < 0 \end{cases} \quad (4-11)$$

$$h_2(x) = \begin{cases} 0 & \text{pour } x > 0 \\ \beta e^{\beta x} & \text{pour } x \leq 0 \end{cases} \quad (4-12)$$

4.1.2 Opérateur de dérivation [1]

Le problème est de détecter un contour après le lissage, ceci peut être fait, comme nous l'avons déjà énoncé, en cherchant les maximums de la dérivée première ou les passages par zéro de la dérivée seconde.

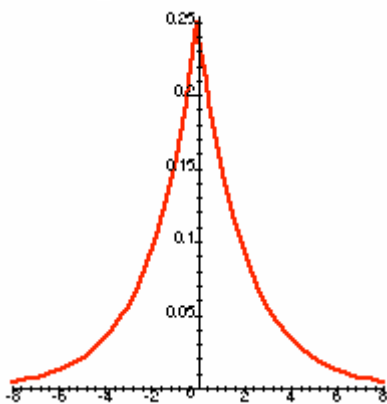
Les filtres de dérivation correspondants [11], voir les figures (1-b) et (1-c), sont obtenus par la dérivation du filtre de lissage.

Dérivée première :

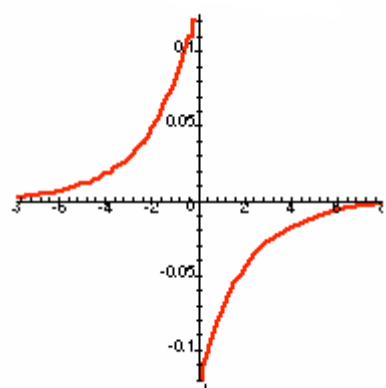
$$h'(x) = \begin{cases} -\frac{\beta^2}{2} e^{-\beta x} & \text{pour } x > 0 \\ \frac{\beta^2}{2} e^{\beta x} & \text{pour } x < 0 \end{cases} \quad (4-13)$$

Dérivée seconde :

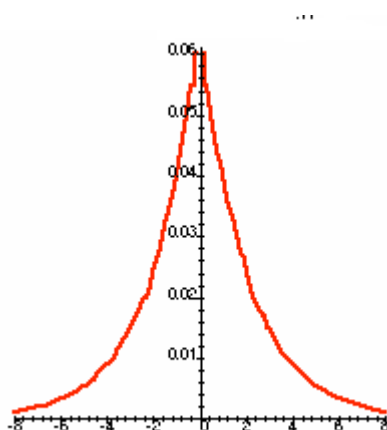
$$h''(x) = \frac{\beta^2}{2} (\beta e^{-\beta|x|} - 2\delta(x)) \quad (4-14)$$



a- Filtre de lissage



b- dérivée première



c- Dérivée seconde

Figure 1 : lissage et dérivés première et seconde du filtre de Shen

Avec les relations (4-13) et (4-14) nous pouvons écrire que :

$$h'(x) = \frac{\beta}{2}(h_2(x) - h_1(x)) \quad (4-15)$$

$$h''(x) = \beta^2(h(x) - \delta(x)) \quad (4-16)$$

Si $A(x)$ est l'image originale et $B(x)$ l'image lissée, nous avons alors :

$$B(x) = A * h(x) = (A * h_1(x)) * h_2(x) \quad (4-17)$$

$$B'(x) = \frac{\beta}{2}(A * h_2(x) - A * h_1(x)) \quad (4-18)$$

$$B''(x) = \beta^2(A * h(x) - A(x)) \quad (4-19)$$

- L'opérateur $h'(x)$ donné en équation (4-15) n'est pas normalisé. si l'on souhaite une réponse de valeur 1 en $x=0$ pour une entrée de type échelon unitaire, $h'(x)$ a pour nouvelle expression :

$$h'(x) = h_2(x) - h_1(x) \quad (4-20)$$

4.2 Extension bidimensionnelle

4.2.1 Filtre de lissage

Le filtre séparable de lissage bidimensionnel est une extension du filtre monodimensionnel précédent. Ce filtre est donné par $h(x, y)$ tel que :

$$h(x, y) = \frac{\beta^2}{4} e^{-\beta(|x|+|y|)} = h(x) \cdot h(y) \quad (4-21)$$

4.2.2 Calcul du gradient

Du fait de la séparabilité, les opérateurs normalisés de dérivation première s'écrivent simplement en utilisant l'équation (4-20) :

$$\frac{\partial h}{\partial x}(x, y) = h(y)(h_2(x) - h_1(x)) \quad (4-22)$$

$$\frac{\partial h}{\partial y}(x, y) = h(x)(h_2(y) - h_1(y)) \quad (4-23)$$

Chacun de ces opérateurs est le produit de la dérivation dans une direction (horizontale ou verticale) par le filtre de lissage dans l'autre direction.

Si $A(x, y)$ est l'image d'entrée. Les images d'estimation des dérivées premières suivant x et y auront pour expression :

$$B_x(x, y) = \left[(A * h(y)) * h_2(x) - (A * h(y)) * h_1(x) \right] \quad (4-24)$$

$$B_y(x, y) = \left[(A * h(x)) * h_2(y) - (A * h(x)) * h_1(y) \right] \quad (4-25)$$

4.2.3 Calcul du Laplacien

En appelant $B(x, y)$ l'image lissée par le filtre $h(x, y)$:

$$\Delta B(x, y) = A * \Delta h(x, y) = A * \frac{\partial^2 h}{\partial x^2}(x, y) + A * \frac{\partial^2 h}{\partial y^2}(x, y) \quad (4-26)$$

Les opérateurs de dérivation seconde ayant pour expressions :

$$\frac{\partial^2 h}{\partial x^2}(x, y) = \beta^2 h(y)(h(x) - \delta(x)) \quad (4-27)$$

$$\frac{\partial^2 h}{\partial y^2}(x, y) = \beta^2 h(x)(h(y) - \delta(y)) \quad (4-28)$$

$$\text{On peut alors écrire que : } \Delta B(x, y) = \beta^2 (2A * h(x, y) - A * h(x) - A * h(y)) \quad (4-29)$$

5. Implémentation des opérateurs de Shen [1]

Ce filtre possède une réponse impulsionnelle infinie dont l'expression analytique permet une implémentation récursive d'ordre 1. Les implémentations de ces opérateurs sont de type « cascade » ou « parallèle ».

Une mise en œuvre récursive du filtre optimal de Shen se fait de la même façon que celle de Deriche.

5.1 Cas monodimensionnel

5.1.1 Opérateur de lissage

Après la discrétisation des fonctions continues (4-11) et (4-12), on aura :

$$h_1[i] = \begin{cases} be^{-\beta i} & \text{pour } i \geq 0 \\ 0 & \text{pour } i < 0 \end{cases} \quad (4-30)$$

$$h_2[i] = \begin{cases} 0 & \text{pour } i > 0 \\ be^{\beta i} & \text{pour } i \leq 0 \end{cases} \quad (4-31)$$

Pour une séquence d'entrée $A[i]$, la réponse en sortie du filtre de lissage a pour expression :

$$B[i] = [A * h_1[i]] * h_2[i] \quad (4-32)$$

La transformée en Z de $h_1[i]$ et $h_2[i]$ est donnée par :

$$h_1(z^{-1}) = \frac{b}{1 - e^{-\beta} z^{-1}} \quad (4-33)$$

$$h_2(z^{+1}) = \frac{b}{1 - e^{-\beta} z} \quad (4-34)$$

Les équations aux différences ont la forme suivante :

$$B_1[i] = b A[i] + e^{-\beta} B_1[i-1] \quad \text{Pour } i = 1, \dots, M \quad (4-35)$$

$$B_2[i] = b B_1[i] + e^{-\beta} B_2[i+1] \quad \text{Pour } i = M, \dots, 1 \quad (4-36)$$

Où M est la longueur de la séquence d'entrée $A[i]$.

b : est une constante de normalisation différente de celle calculée en continu. Elle est obtenue en fixant une réponse unitaire pour un signal d'entrée unitaire à l'origine on trouve alors :

$$b = 1 - e^{-\beta}$$

En remplaçant b par sa valeur dans les équations (4-35) et (4-36), les équations aux différences s'écrivent alors :

$$B_1[i] = e^{-\beta} (B_1[i-1] - A[i]) + A[i] \quad \text{Pour } i = 1, \dots, M \quad (4-37)$$

$$B_2[i] = e^{-\beta} (B_2[i+1] - B_1[i]) + B_1[i] \quad \text{Pour } i = M, \dots, 1 \quad (4-38)$$

Il apparaît que ce traitement requiert 2 multiplications et 4 additions.

5.1.2 Opérateur de dérivation

On adopte la même démarche que précédemment, en partant de la formulation continue (4-18) pour obtenir une forme discrète donnée par :

$$B[i] = A * h_2[i] - A * h_1[i] \quad (4-39)$$

Avec

$$h_1[i] = \begin{cases} c e^{-\beta i} & \text{pour } i \geq 0 \\ 0 & \text{pour } i < 0 \end{cases} \quad (4-40)$$

$$h_2[i] = \begin{cases} 0 & \text{pour } i > 0 \\ c e^{\beta i} & \text{pour } i \leq 0 \end{cases} \quad (4-41)$$

Les filtres $h_i[i]$ de dérivation sont les même que les filtres de lissage .la constante de normalisation calculée pour obtenir la réponse d'amplitude 1 à l'origine est également la même ($c = 1 - e^{-\beta}$).

L'implémentation du dérivateur est directe sous une forme parallèle :

$$B_1[i] = e^{-\beta} (B_1[i-1] - A[i]) + A[i] \quad \text{Pour } i=1, \dots, M \quad (4-42)$$

$$B_2[i] = e^{-\beta} (B_2[i+1] - A[i]) + A[i] \quad \text{Pour } i= M, \dots, 1 \quad (4-43)$$

La séquence de sortie est alors donnée par :

$$B[i] = B_2[i] - B_1[i] \quad (4-44)$$

La dérivation nécessite 2 multiplications et 5 additions.

5.2 Cas bidimensionnel

Pour une image $A[i, j]$ et du fait de la séparabilité de l'opérateur de lissage,

L'implémentation bidimensionnelle est obtenue en exploitant les résultats de l'implémentation monodimensionnelle.

5.2.1 Opérateur de lissage

A partir de la relation (4-44) donnant l'expression dans le domaine continue de l'image lissée B.

$$B(x, y) = A * h(x) * h(y) \quad (4-45)$$

On peut écrire que : $B[i, j] = [A * h[i]] * h[j] = [[A * h_1[i]] * h_2[i]] * h_1[j] * h_2[j]$ (4-46)

Ce lissage comprend 4 multiplications et 8 additions.

5.2.2 Opérateur de dérivation

Pour le calcul des dérivées premières, on utilise la même démarche en étendant au domaine discret. Les formules données en (4-24) (4-25) deviennent alors :

$$B_i[i, j] = [A * h_2[i] - A * h_1[i]] * h_2[j] * h_1[j] \quad (4-47)$$

$$B_j[i, j] = [A * h_2[j] - A * h_1[j]] * h_2[i] * h_1[i] \quad (4-48)$$

Cette opération demande 4 multiplications et 10 additions.

Les étapes suivies par Shen pour extraire les contours d'une image A [i, j] par les maxima locaux, bien sur après un lissage qui s'implémente d'une manière récursive sous une forme cascade. Sont résumées ci après :

- Obtention de B_i [i, j] et B_j [i, j] suivant la mise en oeuvre récursive sous une forme parallèle.
- Calcul de la norme du gradient B [i, j].
- Extraction des maxima locaux (obtention de contours fins), pour chaque point de l'image de la norme du gradient.
- Seuillage par hystérésis de l'image des maxima locaux.

Le calcul de la norme du gradient, l'extraction des maxima locaux et le seuillage par hystérésis se fait de la même procédure énoncée dans le second chapitre (page 20 et 21).

Algorithme de Shen

Les variables utilisées sont définies dans le chapitre précédent.

Filtre de lissage de Shen

Début

Lire β ;

$$b = 1 - e^{-\beta}$$

Calcul de b ;

Pour i allant de 0 à H-1 faire

$$B_x^+[i, 0] := b * A[i, 0];$$

Pour j allant de 1 à W-1 faire

$$B_x^+[i, j] := e^{-\beta} * B_x^+[i, j-1] + b * A[i, j] ;$$

Fait

Fait

Pour i allant de 0 à H-1 faire

$$B_x^-[i, W-1] := b * B_x^+[i, w-1] ;$$

Pour j allant de W-2 à 0 faire

$$B_x^-[i, j] := b * B_x^+[i, j] + e^{-\beta} * B_x^-[i, j+1] ;$$

Fait

Fait

Pour i allant de 0 à H-1 faire

Pour j allant de 0 à W-1 faire

$$L_x[i, j] = B_x^-[i, j];$$

Fait

Fait

Pour j allant de 0 à W-1 faire

$$B_y^+[0, j] := b * B_x^-[0, j];$$

Pour i allant de 1 à H-1 faire

$$B_y^+[i, j] := b * B_x^-[i, j] + e^{-\beta} * B_y^+[i-1, j];$$

Fait

Fait

Pour j allant de 0 à W-1 faire

$$B_y^-[H-1, j] := b * B_y^+[H-1, j];$$

Pour i allant de H-2 à 0 faire

$$B_y^-[i, j] := b * B_y^+[i, j] + e^{-\beta} * B_y^-[i+1, j];$$

Fait

Fait

Pour i allant de 0 à H-1 faire

Pour j allant de 0 à W-1 faire

$$L_y[i, j] = B_y^-[i, j];$$

Fait

Fait

Filtre de dérivation de Shen

Début

Lire (β);

Pour i allant de 0 à H-1 faire

$$B_x^+[i, 0] := (1 - e^{-\beta}) * A[i, 0];$$

Pour j allant de 1 à W-1 faire

$$B_x^+[i, j] := e^{-\beta} * (B_x^+[i, j-1] - A[i, j]) + A[i, j];$$

Fait

Fait

Pour i allant de 0 à H-1 faire

$$B_x^-[i, W-1] := (1 - e^{-\beta}) * A[i, W-1];$$

Pour j allant de W-2 à 0 faire

$$B_x^-[i, j] := e^{-\beta} * (B_x^-[i, j+1] - A[i, j]) + A[i, j];$$

Fait

Fait

Pour i allant de 0 à H-1 faire

Pour j allant de 0 à W-1 faire

$$B_1[i, j] := B_x^-[i, j] - B_x^+[i, j];$$

Fait

Fait

Pour j allant de 0 à W-1 faire

$$B_y^+[0, j] := (1 - e^{-\beta}) * A[0, j];$$

Pour i allant de 1 à H-1 faire

$$B_y^+[i, j] := e^{-\beta} * (B_y^+[i-1, j] - A[i, j]) + A[i, j];$$

Fait

Fait

Pour j allant de 0 à W-1 faire

$$B_y^-[H-1, j] := (1 - e^{-\beta}) * A[H-1, j];$$

Pour i allant de H-2 à 0 faire

$$B_y^-[i, j] := e^{-\beta} * (B_y^-[i+1, j] - A[i, j]) + A[i, j];$$

Fait

Fait

Pour i allant de 0 à H-1 faire

Pour j allant de 0 à W-1 faire

$$B_2[i, j] := B_y^-[i, j] - B_y^+[i, j];$$

Fait

Fait

Pour i allant de 0 à H-1 faire

Pour j allant de 0 à W-1 faire

$$A_s[i, j] := ((B_2[i, j])^2 + (B_1[i, j])^2)^{1/2};$$

Fait

Fait

Fin.

1. Préambule

La mise en œuvre des algorithmes que nous avons présentés dans les chapitres 3 et 4 nécessite le choix d'un langage de programmation ou d'un logiciel de traitement d'images. Aussi, avant de présenter les résultats obtenus sur des images tests et des images Météosat prises dans les canaux visible, infrarouge et vapeur d'eau, nous présenterons le logiciel que nous avons utilisé.

2. Présentation du logiciel pandore

Pandore est une bibliothèque standardisée d'opérateurs de traitement d'images [16]. La version actuelle regroupe des opérateurs traitant d'images 1D, 2D et 3D, en niveaux de gris, en couleurs et multi spectrales. Elle se compose:

- d'une collection d'opérateurs exécutables ;
- d'un environnement de programmation en C++.

Pandore s'adresse en priorité aux traiteurs d'images, dans la mesure où ce logiciel nécessite des connaissances sur les opérations de traitement d'images et sur la façon de les combiner pour résoudre un problème posé. L'utilisation de la bibliothèque n'exige aucune compétence particulière en programmation puisque les opérateurs s'emploient comme des commandes exécutables du système d'exploitation.

Par contre, le développement de nouveaux opérateurs nécessite de bonnes connaissances en programmation C++.

2.1. Fonctionnement du logiciel 'Pandore'

Cette bibliothèque est conçue comme une collection de programmes exécutables travaillant directement sur des fichiers image. La construction d'une application de traitement d'images se fait par l'activation successive d'opérateurs, les images de sortie des uns servant d'images d'entrée aux autres. Les images sont au format Pandore, mais il existe des opérateurs de conversion des formats standards (e.g. bmp, tiff, giff...).

2.2. Exemple

➤ Conversion de format de fichiers

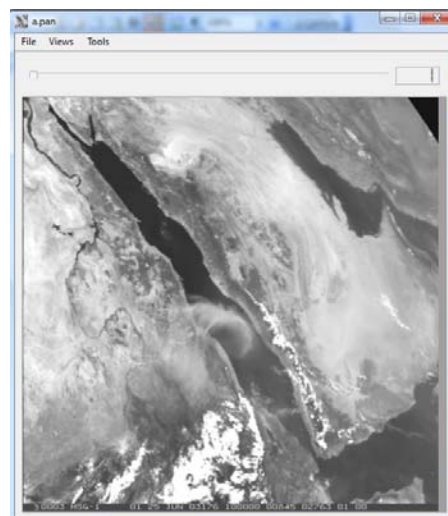
Conversions de format d'images (gif, bmp, tiff, pandore, etc).

```
C:\Windows\system32\cmd.exe
PANDORE Version
6.3.0 <2008-03-12>
GREYC-IMAGE © Caen France

Pandore5> pbmp2pan sat.bmp a.pan
Pandore5> pvisu a.pan
Pandore5>
```



Sat.bmp



a.pan

➤ Conversion de type Pandore

Conversion de l'image Nénuphars.bmp (couleur) en image.pan (niveaux de gris)

```
C:\Windows\system32\cmd.exe
PANDORE Version
6.3.0 <2008-03-12>
GREYC-IMAGE © Caen France

Pandore5> pbmp2pan Nénuphars.bmp nénuphars.pan
Pandore5> pimc2img 0 nénuphars.pan image.pan
Pandore5> pvisu image.pan
Pandore5>
```



Nénuphars.bmp



image.pan

➤ Détection de contours

Détection et localisation de contours dans les images.

- pgradient - Calcul du module et de la direction du gradient par convolution.
- plaplacian - calcul du Laplacien d'une image par convolution.
- pprewitt - Module du gradient de Prewitt.
- proberts - Module du gradient de Roberts.
- psobel - Module du gradient de Sobel.
- pderiche - Détection et localisation des contours de Deriche.
- pshen - Détection et localisation des contours de Shen-Castan.

Calcul du module et de la direction du gradient par convolution

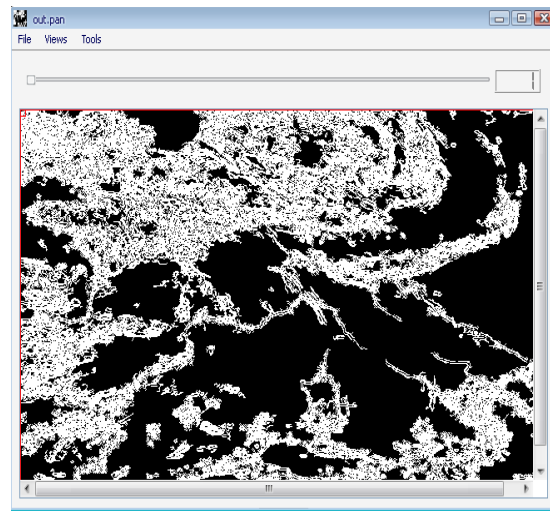
```

C:\WINDOWS\system32\cmd.exe
Pandore5> pversion
PANDORE 6.3.0 <2008-03-12>
Pandore5> pbmp2pan image.bmp a.pan
Pandore5> pgradient 1 a.pan i1.pan i2.pan
Pandore5> pbinarization 30 1e30 i1.pan i3.pan
Pandore5> pbinarization 60 1e30 i1.pan i4.pan
Pandore5> pgeodesicdilatation 1 1 -1 i4.pan i3.pan out.pan
Pandore5> pvisu a.pan
Pandore5> pvisu out.pan

```



a.pan

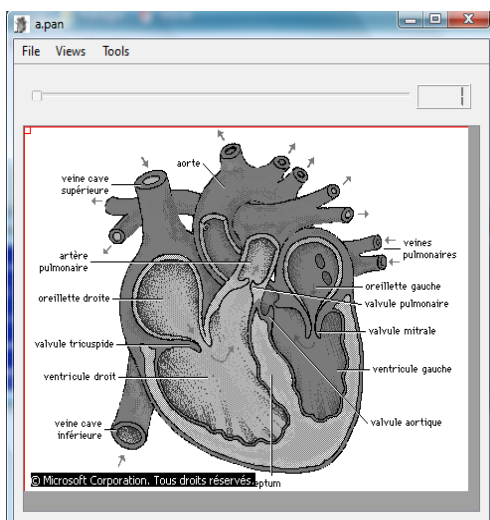


out.pan

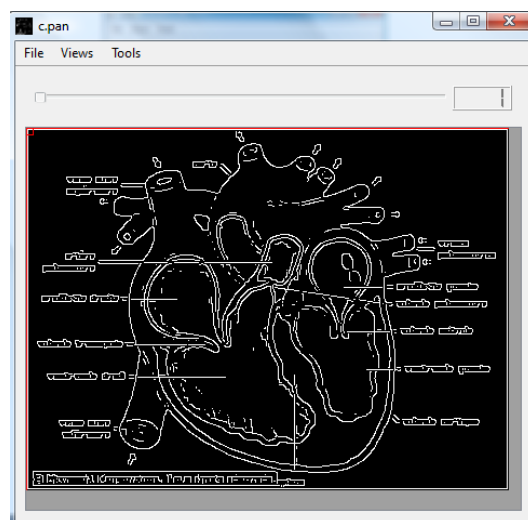
Détection de contour par Deriche

```

C:\Windows\system32\cmd.exe
Pandore5> pbmp2pan coeur.bmp a.pan
Pandore5> pderiche 1 a.pan b.pan
Pandore5> pbinarization 10 1e30 b.pan c.pan
Pandore5> pvisu a.pan
Pandore5> pvisu c.pan
    
```



a.pan



c.pa

J.P Cocquerez et S. philipe, « Analyse d'images : filtrage et segmentation », Edition Masson 1995.

R. Deriche, « techniques d'extraction de contours »,INRIA Sophia Antipolis, Support de cour.
Melle :B. Ouerdia et Melle :B.Farida, « Application des contours actifs sur les images satellitaires pour l'extraction des nuages »,mémoire d'ingénieurs,institut d'électronique,UMMTO 2006.

Document Internet : Rachid DERICHE, « Détection de contours avec une mise en œuvre récursive »

Document Internet « PANDORE -Une bibliothèque d'opérateurs de traitement d'images »

Document Internet « Amélioration du filtre de Canny Deriche pour la détection des contours sous forme de rampe » .

Document Internet « Détection de contours de l'image, étude des généralités de l'approche par filtre optimal »

Algorithme 1: extraction des extréma locaux :

Nous avons implémenté cette opération dans une fonction qui est appelée par toutes les méthodes de détection de contours qui calcule le gradient suivant X et suivant Y, une variante de cette fonction a été adapté par la méthode de Kirsch.

nx : largeur de l'image.

ny : hauteur de l'image.

gx : gradient au point (x,y).

gy : gradient au point (x,y).

Etape 1 : transmission des tableaux du gradient suivant X et Y ainsi que l'image de la norme du gradient.

Etape 2 : calcul de Gr1 et Gr2 et extraction des maximas :

Pour x allant de 0 à nx faire.

 Pour y allant de 0 à ny faire.

 Si gx=0 et gy=0 faire

 Gr1=0.

 Gr2=0.

 Fin si

 Si gx=0 faire.

 Si gy>0 faire.

 Gr1=Gradient|x||y+1|.

 Gr2=Gradient|x||y-1|.

 Fin si.

 Si gy<0 faire

 Gr1=Gradient|x||y-1|.

 Gr2=Gradient|x||y+1|.

 Fin si.

Si gy=0 faire

 Si gx>0 faire

 Gr1=Gradient|x+1||y|.

 Gr2=Gradient|x-1||y|.

 Fin si

 Si gx<0 faire.

 Gr1=Gradient|x-1||y|.

 Gr2=Gradient|x+1||y|.

 Fin si

Fin si

 Si gx>0 faire.

 Si gy>0 faire.

 Si gy>gx faire.

 C=gx/gy.

 Gr1=C*Gradient|x+1||y+1|+(1-C)*Gradient|x||y+1|.

$$\text{Gr2} = C * \text{Gradient } |x - 1| |y - 1| + (1 - C) * \text{Gradient } |x| |y - 1|.$$

Fin si

Sinon faire

$$C = gy/gx$$

$$\text{Gr1} = C * \text{Gradient } |x + 1| |y + 1| + (1 - C) * \text{Gradient } |x + 1| |y|.$$

$$\text{Gr2} = C * \text{Gradient } |x - 1| |y - 1| + (1 - C) * \text{Gradient } |x - 1| |y|.$$

Fin sinon

Fin si

Si $gy < 0$ faire

Si $-gy > gx$ faire

$$C = -gx/gy.$$

$$\text{Gr1} = C * \text{Gradient } |x + 1| |y - 1| + (1 - C) * \text{Gradient } |x| |y - 1|.$$

$$\text{Gr2} = C * \text{Gradient } |x - 1| |y + 1| + (1 - C) * \text{Gradient } |x| |y + 1|.$$

Fin si

Sinon faire

$$C = -gy/gx$$

$$\text{Gr1} = C * \text{Gradient } |x + 1| |y - 1| + (1 - C) * \text{Gradient } |x + 1| |y|.$$

$$\text{Gr2} = C * \text{Gradient } |x - 1| |y + 1| + (1 - C) * \text{Gradient } |x - 1| |y|.$$

Fin sinon

Fin si

Si $gx < 0$ faire

Si $gy > 0$ faire

Si $gy > -gx$ faire

$$C = -gx/gy.$$

$$\text{Gr1} = C * \text{Gradient } |x - 1| |y + 1| + (1 - C) * \text{Gradient } |x| |y + 1|.$$

$$\text{Gr2} = C * \text{Gradient } |x + 1| |y - 1| + (1 - C) * \text{Gradient } |x| |y - 1|.$$

Fin si

Sinon faire

$$C = -gy/gx$$

$$\text{Gr1} = C * \text{Gradient } |x - 1| |y + 1| + (1 - C) * \text{Gradient } |x - 1| |y|.$$

$$\text{Gr2} = C * \text{Gradient } |x + 1| |y - 1| + (1 - C) * \text{Gradient } |x + 1| |y|.$$

Fin sinon

Fin si

Si $gy < 0$ faire

Si $-gy < -gx$ faire

$$C = gy/gx$$

$$\text{Gr1} = C * \text{Gradient } |x - 1| |y - 1| + (1 - C) * \text{Gradient } |x - 1| |y|.$$

$$\text{Gr2} = C * \text{Gradient } |x + 1| |y + 1| + (1 - C) * \text{Gradient } |x + 1| |y|.$$

Fin si

Sinon faire

$$C = gx/gy.$$

```

Gr1=C*Gradient |x-1||y-1|+(1-C) *Gradient |x||y-1|.
Gr2= C*Gradient |x+1||y+1|+(1-C) *Gradient |x||y+1|.
    Fin sinon
  Fin si
Fin si

Si Gradient |x||y|>Gr2 et Gradient |x||y| ≥ Gr1 faire
  Extrem [x][y]=Gradient |x||y|.
Fin si
Sinon faire
  Extrem [x][y]=0.
Fin sinon
Faire
Faire
Etape 3 : affichage de l'image Extrem [x][y]

```

Algorithme 2 : souillage par hystérésis :

L'opération de souillage par hystérésis a été implémentée dans une fonction qui reçoit l'image des extrema locaux.

```

Extrem[x][y] : image des extrema locaux.
Seuillage [x][y]: image souillée.
SB =seuil bas.
SH =seuil haut.
Pour y allant de 0 à hauteur -1 faire.
  Pour x allant de 0 à largeur-1 faire.
    Si Extrem[x][y]> SB faire.
      Seuillage [x][y]= Extrem[x][y]
    Fin si
  Sinon faire
    Seuillage [x][y]=0
  Fin sinon
Faire

Iteration =0
Pour y allant de 0 à hauteur -1 faire
  Pour x allant de 0 à largeur -1 faire
    Si Seuillage [x][y]= 0 faire
      Sauter à l'iteration suivante
    Fin si
  Si Seuillage [x][y]> SH faire.
    Sauter à l'iteration suivante.

```

```

    Fin si
Recherche du Max sur un voisinage de 3*3 autour de Seuillee [x][y]
    Val=seuillee [x][y]
    Si val<Max faire
        Itération =1
        Seuillee Val=seuillee[x][y] = Max
    Fin si
    faire
faire
tant que itetation =1
Pour y allant de 0 à hauteur -1 faire.
    Pour x allant de 0 à largeur-1 faire.
        Si Extrem[x][y] > SH faire.
            Seuillee [x][y] = 255
        Fin si
        Sinon faire
            Seuillee [x][y] = 0
        Fin sinon
    Faire
Faire

```

Algorithme 3 détection de contours pour l'opérateur du Laplacien :

```

Début
Charger l'image ;
Pour X allant de 1 à longueur -1 Faire
    Pour Y allant de 1 à Largeur -1 Faire
        Pour i allant de -1 à +1 Faire
            Pour j allant de -1 à +1 faire
                Fenêtre [i,j] := image [X + i, Y + j] ;
            Fait
        Fait
        Pour i allant de -1 à +1 Faire
            Pour j allant de -1 à +1 faire
                Image-Laplac [X,Y] := Imag-Laplac [X,Y] + Fenêtre [i,j] * Masque [i,j] ;
            Fait
        Fait
    Fait
Fait
Fin

```