



UNIVERSITE MOHAMED BOUDIAFDE M'SILA

Faculté des Mathématiques et de l'Informatique

Département de Mathématiques



MEMOIRE DE FIN D'ETUDE

Présenté pour l'obtention du Diplôme de **MASTER**

Domaine : Mathématiques et Informatique

Filière: Mathématiques

Option : EDPS et application

Par

Sadem Zerouak et Mouhamed Khaldoune

Sujet

Calcul fractionnaire et détection de contours d'image

Soutenu publiquement le :

Devant le jury :

B.Noura

Univ de M'sila

Président

N. Benhamidouche

Univ de M'sila

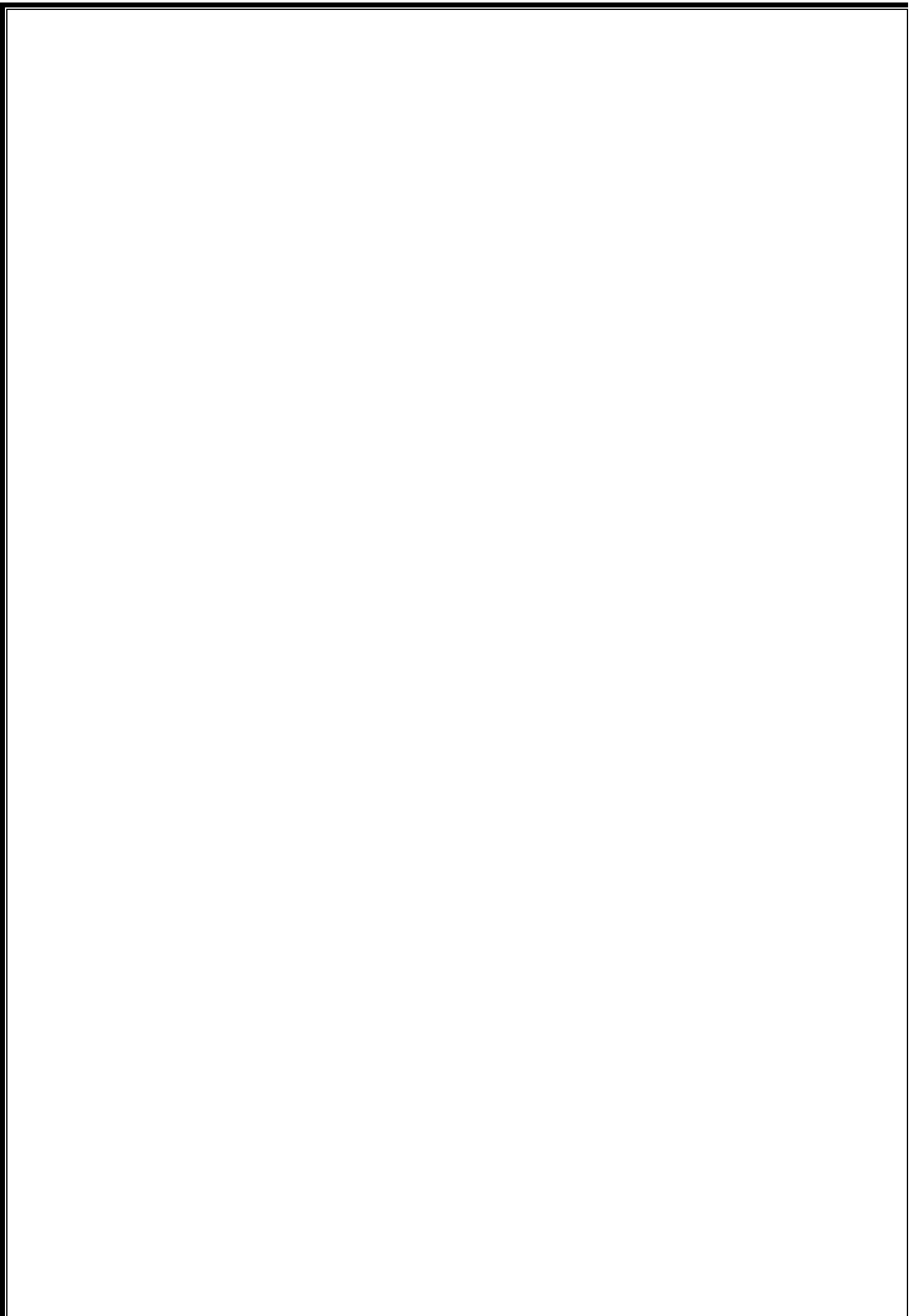
Rapporteur

Y. Arioua

Univ de M'sila

Examineur

Promotion : 2016 / 2017



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

"وَقُلِ اعْمَلُوا فَسَبَّحُوا اللَّهَ عَمَلَكُمْ وَرَسُولَهُ
وَالْمُؤْمِنُونَ وَسُجَّدُوا إِلَىٰ عَالِمِ الْغَيْبِ وَالشَّهَادَةِ
فَيُنَبِّئُكُمْ بِمَا كُنْتُمْ تَعْمَلُونَ" [التوبة: 105]

ديقول العباد الأصفهاني

(إني رأيت أنه لا يَلْتَبُ أحد كتابا في يومه إلا قال
فيه عنه : لو غير هذا لكان أحسن ، ولو زيد لكان
بسنحسن ، ولو قدم هذا لكان أفضل ، ولو ترك هذا
لكان أجمل ، وهذا من أعظم العبر ، وهو دليل على
استيلاء النفس على جملة البشر).



Dédicace

Je dédie ce travail :

Avant tout à mes chers parents

Qui m'ont soutenue durant toutes ces années de formation.

A mes chère frères et

A mes chères sœurs.

A toute la famille zerouak et la famille khaldoune

*A tous mes amis et toute ma famille de département de
Mathématiques*

Et a toute la promotion 2016/2017 de l'EDP

*A toutes mes adorables que j'ai connu pendant toute
ma vie ...*

Sadem Zerouak,..... Mouhamed Khaldoune

Notations

$D^{(1)}f(x)$:Dérivées du premier ordre d'une fonction f .

$D^{(2)}f(x)$:Dérivées du second ordre d'une fonction f .

$D^\alpha f(x)$: Le dérivé d'une fonction f d'ordre réel.

$\nabla u = \left(\frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}, \dots, \frac{\partial u}{\partial x_n} \right)^t$, $(x_1, x_2, \dots, x_n) \in \mathbb{R}^n$, le vecteur gradient d'une fonction u .

$\Delta u = \left(\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + \dots + \frac{\partial^2 u}{\partial x_n^2} \right)$, $(x_1, x_2, \dots, x_n) \in \mathbb{R}^n$, le Laplacien d'une fonction u .

Table des matières

Introduction	1
1 Quelques rappels et définitions	3
1.1 Traitement d'images	3
1.1.1 Pourquoi le traitement d'images ?	3
1.1.2 Définitions générales des images	4
1.1.3 Définitions générales des contours	6
1.1.4 Approximation par différences finies	7
1.2 Le calcul fractionnaire	8
1.2.1 Définition générale	8
1.2.2 Fonctions Spéciales(Gamma d'Euler et Beta...)	8
1.2.3 Dérivées fractionnaires de Grünwald-Letnikov	9
2 Détection des contours d'images	11
2.1 Méthode du maximum du gradient	12
2.1.1 Principe	12
2.1.2 Opérateurs gradient	15
2.1.3 Filtrage de Canny	22
2.2 Méthode du passage à zéro Laplacien	24
2.2.1 Principe	24
2.2.2 Opérateurs Laplacien	25
2.2.3 Filtrage de Marr et Hildreth	28
2.3 Comparaison (Gradient/Laplacien)	31

2.3.1	Avantages et inconvénients	32
2.3.2	Résultats	33
3	Détection des contours et calcul fractionnaire	34
3.1	Calcul fractionnaire d'ordre non entier $0 \leq v \leq 2$	34
3.1.1	Principe	34
3.1.2	Opérateurs gradient fractionnaire	35
3.1.3	Opérateurs Laplacien fractionnaire	38
3.2	Structure d'opérateur gaussien d'ordre non entier $1 \leq v \leq 2$	41
3.2.1	Principe	41
3.2.2	Opérateurs gaussien fractionnaire	42
3.3	Comparaison générale	44
3.3.1	L'image artificielle(sans bruit/bruit)	44
3.3.2	L'image naturelle(sans bruit/bruit)	45
3.3.3	Résultats	46
	Conclusion	47
	Annexe	48
	Bibliographie	73

Introduction

Le calcul fractionnaire joue un rôle très important dans différents domaines de recherche, y compris l'image et le traitement du signal.

Dans le traitement d'image, le calcul fractionnaire peut être assez intéressant pour le filtrage et la détection des bords (contours d'images), donnant une nouvelle approche pour améliorer la qualité des images. Le calcul fractionnaire est la généralisation de la dérivée et l'intégration d'une fonction à l'ordre non entier. Ce calcul a évolué dans le domaine de la recherche des mathématiques pures jusqu'en 1920, quand la science appliquée a commencé à l'utiliser. Récemment, le calcul fractionnaire a été impliqué dans les méthodes de traitement d'image.

Dans cette recherche on va expliquer et développer les deux travaux réalisés par: HAN Qirui / LIU Ke et par: Zhuzhong YANG / Fangnian LANG / Xiaohong YU / Yu ZHANG

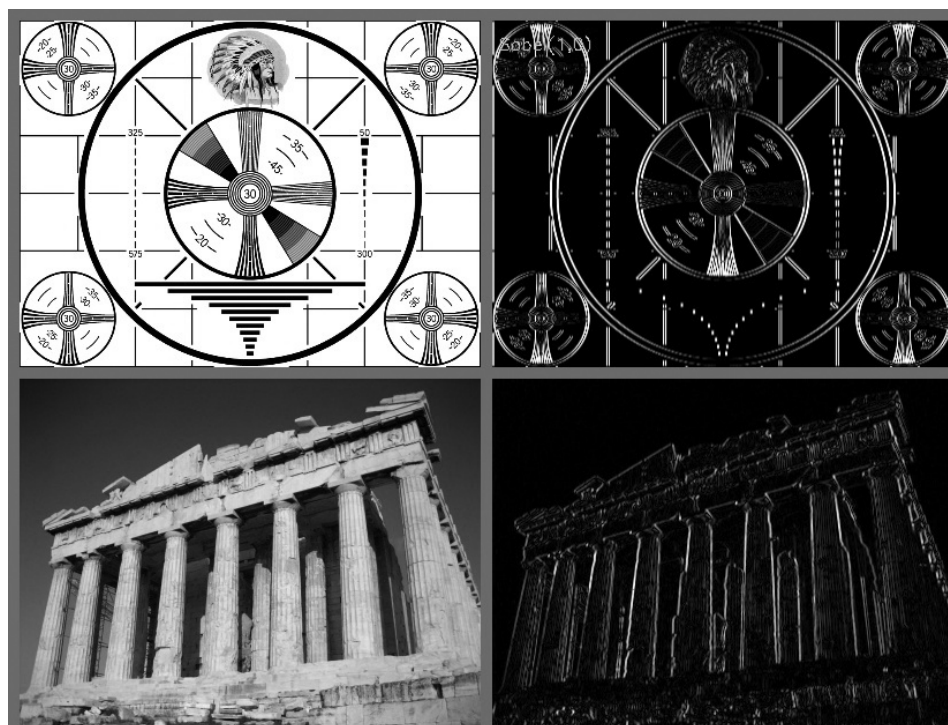
- *A Novel Edge Detection Operator Based On Fractional Gaussian Differential*[1]
- **The Construction of Fractional Differential Gradient Operator** [16]

On va montrer comment le calcul fractionnaire peut fournir des avantages au traitement d'image. En particulier, nous verrons qu'il est utile dans la détection des contours d'images et pour améliorer la qualité de l'image.

Le but de la détection des contours est de repérer les points d'une image numérique qui correspondent à un changement brutal de l'intensité lumineuse. Ces changements de propriétés de l'image traduisent en général des événements importants. Ils incluent des discontinuités dans la profondeur, dans l'orientation d'une surface, dans les propriétés d'un matériau

et dans l'éclairage d'une scène. La détection des contours est un champ de recherche qui appartient au traitement d'image et à la vision par ordinateur, particulièrement dans le domaine de l'extraction de caractéristiques. Donc on va répartir cette recherche comme suit:

- Dans le premier chapitre nous donnons des définitions et rappels du traitement d'image (définition d'image mathématique et types d'images, contours d'images, approximation par différences finies ...etc) et le calcul fractionnaire (définitions et dérivée fractionnaire.....etc).
- Dans le deuxième chapitre nous allons étudier et détailler la détection des contours d'images (principe et algorithme et méthode de détection.....etc) avec les méthodes simples(du maximum du gradient et du passage par zéro du laplacienetc), à la fin, une étape comparative et réalisé.
- Finalement, dans le dernier chapitre nous allons étudier et détailler l'utilisation du calcul fractionnaire pour la détection des contours d'images et nous allons donner les résultats de ce travail. Avec un travail numérique réalisé sous matlab.



Chapitre 1

Quelques rappels et définitions

1.1 Traitement d'images

1.1.1 Pourquoi le traitement d'images ?

Le domaine de traitement d'image est important pour l'étude pour plusieurs raisons. Le futur est au multimedia : les images sont partout!. Les applications sont multiples:

- 1. Télédétection : météo, cartographie, astronomie.*
- 2. Imagerie médicale : aide au diagnostic, tomographie, suivi automatique, reconstruction 3D.*
- 3. Applications militaires : guidage de missile, reconnaissance terrestre.*
- 4. Robotique : reconnaissance/assemblage de pièces, véhicules autonomes, contrôle de qualité.*
- 5. Sécurité : identification de visages, reconnaissance d'empreintes digitales, tatouage d'image (watermarking), data hiding.*
- 6. Divertissement : HDTV, images haute qualité (DVD), compression (normes JPEG et MPEG).*
- 7. Amélioration : augmenter la qualité de la perception visuelle qu'on a d'une image.*

8. *Restoration* : compenser les dégradations (bruit, flou, ...).
9. *Compression* : stocker et transférer efficacement.
10. *Segmentation* : délimiter les “objets”.
11. *Reconstruction 3D* : obtenir un volume à partir de plans(images).
12. *Représentation* : modéliser
 - (a) *Bas niveau* : texture, couleur, forme, etc.
 - (b) *Haut niveau* : caractéristiques (features), apprentissage statistiques, graphes
13. *Analyse* : convertir en informations.
14. *Reconnaissance / Compréhension* : identifier le contenu.

1.1.2 Définitions générales des images

Définition 1.1.1 *l'image est une fonction 2D.*

$$I : (x, y) \rightarrow I(x, y)$$

Définition 1.1.2 (Le pixel) *Le pixel est l'abréviation de “Picture élément” élément d'image. C'est la primitive de bas niveau la plus pauvre en information car ses seuls attributs sont la position dans la matrice image (n de ligne, n de colonne) et la valeur numérique indiquant sa couleur, ou son niveau de gris. Il peut être représenté en mémoire sur :*

Un bit (0 ou 1) pour les images monochromes : 0 pour le noir et 1 pour le blanc.

Un octet, soit 256 niveaux de gris pour une image à niveaux de gris : 0 noir et 255 blanc.

Types d'images

Définition 1.1.3 *Les images binaires(noir ou blanc)*

Exemple, images les plus simples, un pixel peut prendre uniquement les valeurs noir ou blanc. C'est typiquement le type d'image que l'on utilise pour scanner du texte quand celui-ci est composé d'une seule couleur.

Définition 1.1.4 Les images aux niveaux du gris

En général, les images aux niveaux du gris renferment 256 teintes de gris. Par convention la valeur zéro représente le noir (intensité lumineuse nulle) et la valeur 255 le blanc (intensité lumineuse maximale). Le nombre 256 est lié à la quantification de l'image. En effet chaque entier représentant un niveau du gris est codé sur 8 bits. Il est donc compris entre 0 et $2^8 - 1 = 255$. C'est la quantification la plus courante. On peut coder une image en niveaux du gris sur 16 bits ($0 \leq n \leq 2^{16} - 1$) ou sur 2 bits : dans ce dernier cas le « niveau de gris » vaut 0 ou 1 : il s'agit alors d'une image binaire (Noir et Blanc).

Définition 1.1.5 Les images couleurs

L'espace couleur est basé sur la synthèse additive des couleurs, c'est à dire que le mélange de trois composantes (par exemple (R, V, B)) donne une couleur. On garde l'information couleur, ou intensité lumineuse et chromaticité. Un pixel est codé par trois valeurs numériques. La signification de ces valeurs dépend du type de codage choisi. Le plus utilisé pour le maniement des images numériques est l'espace couleur Rouge, Vert, Bleu (R,V,B) (RGB en anglais). La restitution des couleurs sur écran utilise cette représentation (synthèse additive).

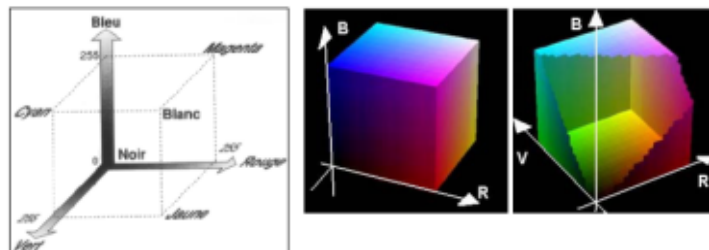
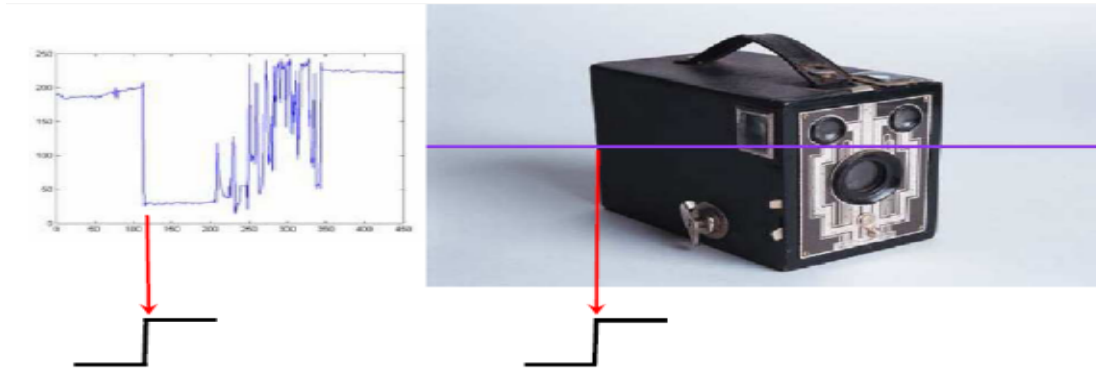


FIG. 1.2 – L'espace couleur Rouge Vert Bleu (RGB)

1.1.3 Définitions générales des contours

Le contour

Définition 1.1.6 *contour d'image c'est la frontière entre deux objets dans une image ou la discontinuité de l'image (variation brusque d'intensité)*



tout discontinuité n'est pas nécessairement située a la frontière entre deux objets.

de profondeur



d'orientation
de surface

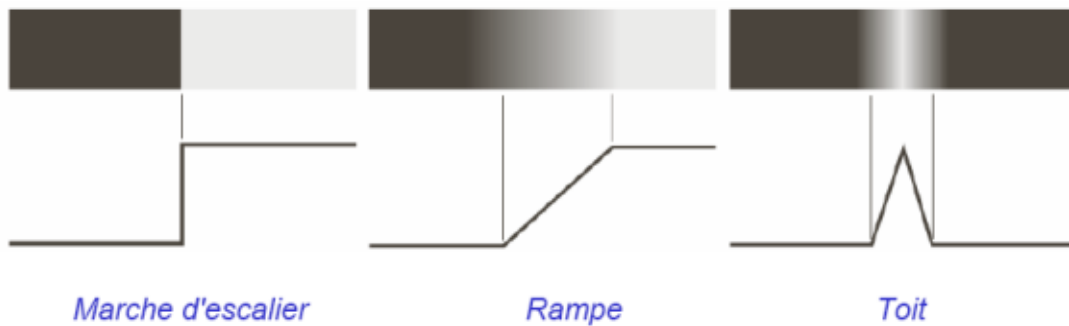
de réflectance

d'illumination

Définition 1.1.7 *Dans l'approche " contour ", on considère que les primitives à extraire sont les lignes de contrastes séparant des régions de niveaux de gris différents et relativement homogènes, ou bien des régions de texture différentes. En pratique, il s'agit de reconnaître les zones de transition et de localiser au mieux la frontière entre les régions. Donc un contour peut approximativement être défini comme une zone de l'image où l'intensité des pixels change brusquement, cette discontinuité dans l'image est le passage d'un niveau de gris à un autre, de manière plus ou moins rapide, donnant lieu à trois interprétations d'échelon, de rampe, de toit ou de ligne. On distingue trois types de contour simple.*

Types de contours

- **Marche d'escalier** : le contour est net (contour idéal).
- **Rampe** : le contour est plus flou.
- **Toit** : il s'agit d'une ligne sur un fond uniforme.



1.1.4 Approximation par différences finies

Définition 1.1.8 On note u l'image considérée, et $u_{i,j}$ sa valeur au pixel (i, j) . Le pas d'espace h est fixé égal à 1.

Il existe la fonction gradient dans Matlab. Cependant, il sera utile de coder plusieurs types d'approximations du gradient. L'approche par différences finies est la plus simple. Elle se révèle bien adaptée pour les équations paraboliques.

Différences finies centrées :

$$\delta_x u_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2}$$

$$\delta_y u_{i,j} = \frac{u_{i,j+1} - u_{i,j-1}}{2}$$

Différences finies décentrées à droite :

$$\delta_x^+ u_{i,j} = \frac{u_{i+1,j} - u_{i,j}}{2}$$

$$\delta_y^+ u_{i,j} = \frac{u_{i,j+1} - u_{i,j}}{2}$$

Différences finies décentrées à gauche :

$$\begin{aligned}\delta_x^- u_{i,j} &= \frac{u_{i,j} - u_{i-1,j}}{2} \\ \delta_y^- u_{i,j} &= \frac{u_{i,j} - u_{i,j-1}}{2}\end{aligned}$$

1.2 Le calcul fractionnaire

1.2.1 Définition générale

Nous connaissons bien les notations $D^{(1)}f(x)$ et $D^{(2)}f(x)$ pour les dérivées du premier et du second ordre, mais beaucoup de gens peuvent douter de la signification de la notation $D^{(1/2)}f(x)$, décrivant la dérivée d'ordre $1/2$. Les gens, à qui la notation semble plutôt inconnue, pourraient obtenir une idée trompeuse qu'il s'agit d'un calcul récemment développé. En fait, le problème des dérivées fractionnaires est assez ancien. Leibnitz en a déjà discuté au XVIII^e siècle et d'autres noms célèbres du passé ont étudié et contribué au développement du calcul fractionnaire dans le domaine des mathématiques pures [7]. Nous voyons les premières applications du calcul fractionnaire en 1920 et seulement récemment, il a été appliqué au traitement d'image [14].

Il existe plusieurs définitions de dérivées fractionnaires, comme définitions de Riemann-Liouville, Liouville, Caputo ainsi que Grunwald-letnikov qui sont les plus utilisées. Nous présentons dans cette partie la définition de Grunwald-letnikov qui est utilisée dans ce travail.

1.2.2 Fonctions Spéciales (Gamma d'Euler et Beta...)

Définition 1.2.9 On appelle fonction **Gamma** la fonction définie par:

$$\Gamma(z) = \int_0^{+\infty} t^{z-1} e^{-t} dt, \quad (z \in \mathbb{C}, \operatorname{Re}(z) > 0)$$

avec: $t^{z-1} = e^{(z-1)\ln t}$.

Définition 1.2.10 La fonction de **Beta** est un type d'intégrale d'Euler définie par:

$$B(p, q) = \int_0^1 t^{p-1} (1-t)^{q-1} dx, \quad (p, q \in \mathbb{C}, \operatorname{Re}(p) > 0, \operatorname{Re}(q) > 0)$$

pour tout $p, q \in \mathbb{C}$, avec $\text{Re}(p) \succ 0$, $\text{Re}(q) \succ 0$ on a:

$$B(p, q) = \frac{\Gamma(p) \Gamma(q)}{\Gamma(p+q)}$$

Définition 1.2.11 La fonction de **Mittag-Leffter** est définie par:

$$E_\alpha(x) = \sum_{k=0}^{+\infty} \frac{x^k}{\Gamma(\alpha k + 1)}, \quad \alpha \succ 0$$

et la fonction de Mittag-Leffter généralisée est définie par:

$$E_{\alpha, \beta}(x) = \sum_{k=0}^{+\infty} \frac{x^k}{\Gamma(\alpha k + \beta)}, \quad \alpha, \beta \succ 0$$

1.2.3 Dérivées fractionnaires de Grünwald-Letnikov

Définition 1.2.12 Cette définition se base sur l'obtention de dérivées par différences finies.

Soit $f : \mathbb{R} \rightarrow \mathbb{R}$, pour $h \succ 0$ on a:

$$f'(t) = \lim_{h \rightarrow 0} \frac{1}{h} [f(t) - f(t-h)] \quad (1.1)$$

et la dérivée seconde :

$$\begin{aligned} f''(t) &= \lim_{h \rightarrow 0} \frac{1}{h} [f'(t) - f'(t-h)] \\ &= \lim_{h \rightarrow 0} \frac{1}{h} \left[\frac{1}{h} (f(t) - f(t-h)) - \frac{1}{h} (f(t-h) - f(t-2h)) \right] \\ &= \lim_{h \rightarrow 0} \frac{1}{h^2} [f(t) - 2f(t-h) + f(t-2h)] \end{aligned} \quad (1.2)$$

Plus généralement, la dérivée $n^{\text{ième}}$ de f est donnée par :

$$f^{(n)}(t) = \lim_{h \rightarrow 0} \frac{1}{h^n} \sum_{k=0}^n (-1)^k C_k^n f(t - kh) \quad (1.3)$$

où

$$C_k^n = \frac{n!}{k!(n-k)!} = \frac{n(n-1)\dots(n-(k-1))}{k!} \quad (1.4)$$

Il est possible d'étendre C_k^n à $k \succ n$, en posant $C_k^n = 0$. La formule devient alors:

$$f^{(n)}(t) = \lim_{h \rightarrow 0} \frac{1}{h^n} \sum_{k=0}^{\infty} (-1)^k C_k^n f(t - kh) \quad (1.5)$$

Là encore, on peut généraliser le terme de droite grâce à la fonction **Gamma**, en posant pour $\alpha \in \mathbb{R}^+ - \mathbb{N}$, et $k \in \mathbb{N}$,

$$C_k^\alpha = \frac{\Gamma(\alpha + 1)}{\Gamma(k + 1)\Gamma(\alpha - k + 1)} \quad (1.6)$$

Notons cette fois que $C_k^\alpha \neq 0$ même si $k > n$.

Définition 1.2.13 Soit $\alpha > 0$. La dérivée fractionnaire de **Grünwald-Letnikov** à gauche d'ordre α est définie par :

$$\forall t \in \mathbb{R}, {}^{GL}D_+^\alpha f(t) = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{k=0}^{\infty} (-1)^k C_k^\alpha f(t - kh) \quad (1.7)$$

Définition 1.2.14 Soit $\alpha > 0$. La dérivée fractionnaire de **Grünwald-Letnikov** à droite d'ordre α est définie par :

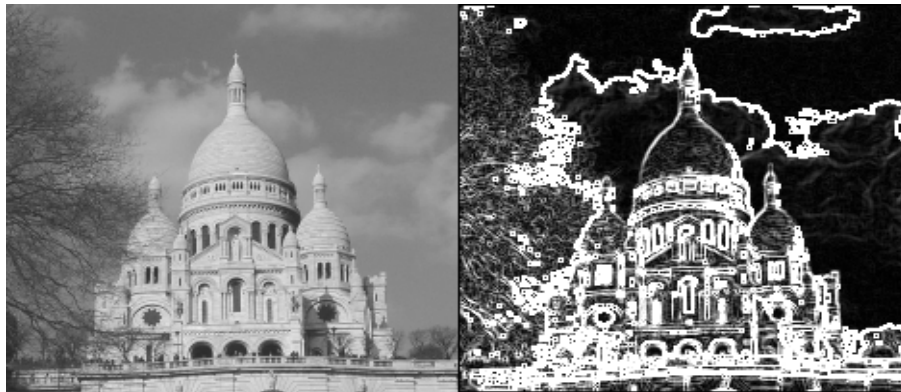
$$\forall t \in \mathbb{R}, {}^{GL}D_-^\alpha f(t) = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{k=0}^{\infty} (-1)^k C_k^\alpha f(t + kh) \quad (1.8)$$

La dérivée de **Grünwald-Letnikov** présente un intérêt numérique évident. Si h est assez petit, l'évaluation discrète de $\frac{1}{h^\alpha} \sum_{k=0}^{\infty} (-1)^k C_k^\alpha f(t - kh)$ permet d'approximer la dérivée fractionnaire (**de Liouville**) sur \mathbb{R} .

Chapitre 2

Détection des contours d'images

La détection des contours d'une image réduit de manière significative la quantité de données et élimine les informations qu'on peut juger moins pertinentes, tout en préservant les propriétés structurelles importantes de l'image. Il existe un grand nombre de méthodes de détection des contours des images mais la plupart d'entre elles peuvent être regroupées en deux catégories. La première recherche les extremums de la dérivée première, en général les maximums locaux de l'intensité du gradient. La seconde recherche les annulations de la dérivée seconde, en général les annulations du laplacien ou d'une expression différentielle non linéaire.



2.1 Méthode du maximum du gradient

2.1.1 Principe

Le principe de la détection de contours par l'utilisation du gradient consiste à calculer d'abord le gradient de l'image dans deux directions orthogonales puis le module du gradient. Il s'agira ensuite d'effectuer une sélection des contours les plus marqués, c'est-à-dire les points de plus fort contraste par un seuillage adéquat, les directions des contours étant orthogonales à la direction α_0 déterminée en tout pixel de l'image.

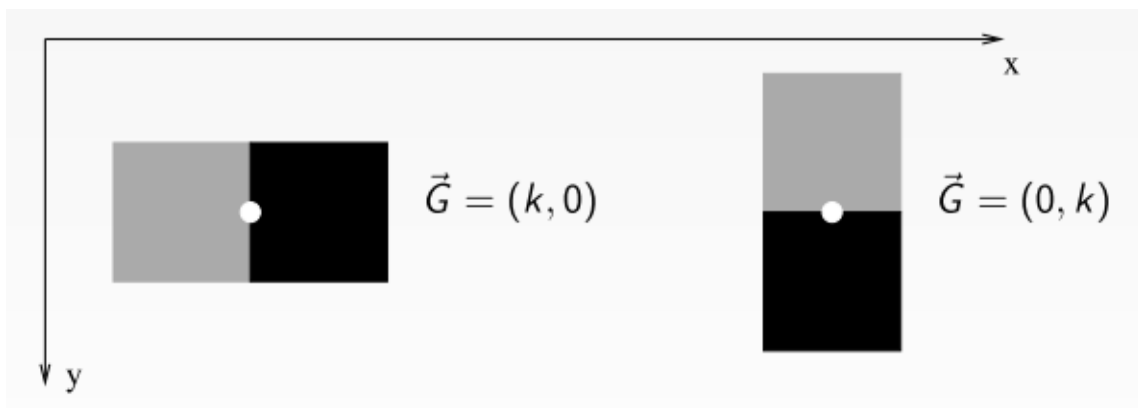
On rappelle que l'image est une fonction 2D.

$$f : (x, y) \longrightarrow f(x, y)$$

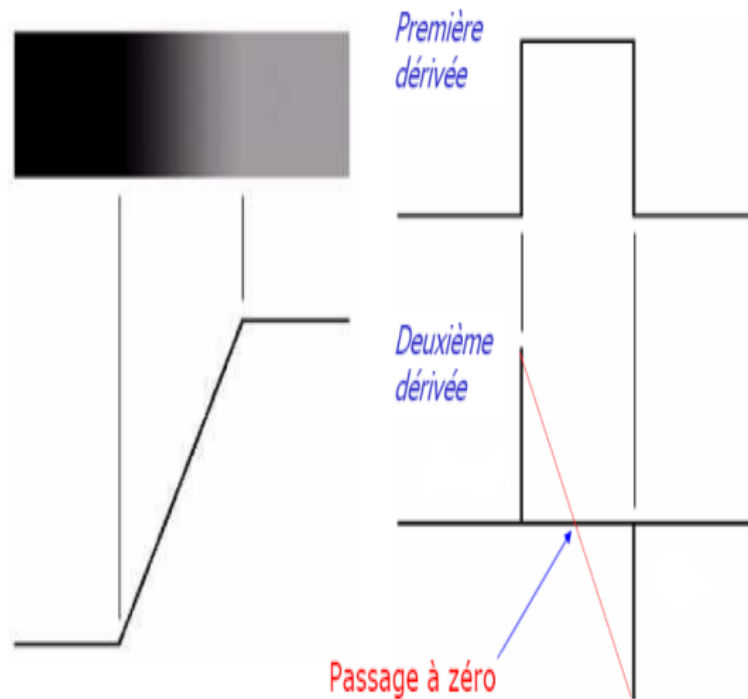
En considérant l'image dans un repère orthogonal (Oxy) tel que (Ox) désigne l'axe horizontal et (Oy) l'axe vertical,

dérivée première de l'image

$$\vec{G} = (G_x, G_y) = \left(\frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y} \right)$$



Dérivées d'un contour



le Gradient de l'image (ou plutôt de la luminance f) en tout point ou pixel de coordonnées (x, y) est désigné par :

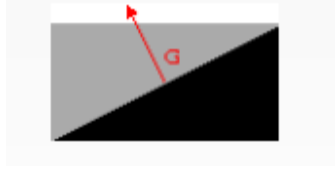
$$\overrightarrow{\text{Grad}} f = \vec{\nabla} f = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}$$

Le **module du gradient** permet de quantifier l'importance du contour mis en évidence, c'est-à-dire l'amplitude du saut d'intensité relevé dans l'image :

$$\|\vec{\nabla} f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

La **direction du gradient** permet de déterminer l'arête présentée dans l'image. En effet, la direction du gradient est orthogonale à celle du contour :

$$\alpha_0 = \arctan\left(\frac{\partial f / \partial y}{\partial f / \partial x}\right)$$



La direction du gradient de f :

$$\vec{g} = \frac{\vec{\nabla} f}{\|\vec{\nabla} f\|}$$

Lieu des maxima du gradient dans la direction \vec{g} du gradient :

$$\frac{\partial f}{\partial \vec{g}} \text{ est un extrema et } \frac{\partial^2 f}{\partial \vec{g}^2} = 0$$

$$\text{avec } \frac{\partial}{\partial \vec{g}} = \vec{g} \cdot \vec{\nabla}$$

L'équation obtenue est complexe et non linéaire :

$$\frac{\partial f}{\partial x} \frac{\partial}{\partial x} \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} + \frac{\partial f}{\partial y} \frac{\partial}{\partial y} \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} = 0$$

Définition 2.1.1 *Le gradient, en un pixel d'une image numérique, est un vecteur caractérisé par son amplitude et sa direction :*

- *L'amplitude est liée à la quantité de variation locale des niveaux de gris.*
- *La direction est orthogonale à la frontière qui passe au point considéré.*

Remarque 2.1.1 *Le gradient est orthogonal et normal aux courbes de niveaux de l'image.*

- **Remarque 2.1.2** *La direction du gradient est orthogonale à la frontière qui passe au point considéré et maximise la dérivée directionnelle.*

Sa norme est la valeur de cette dérivée.

Le gradient est orienté dans le sens de croissance du champ, cette croissance étant d'autant plus forte que la normale du gradient est importante (d'après l'inégalité de Cauchy-Schwarz).

2.1.2 Opérateurs gradient

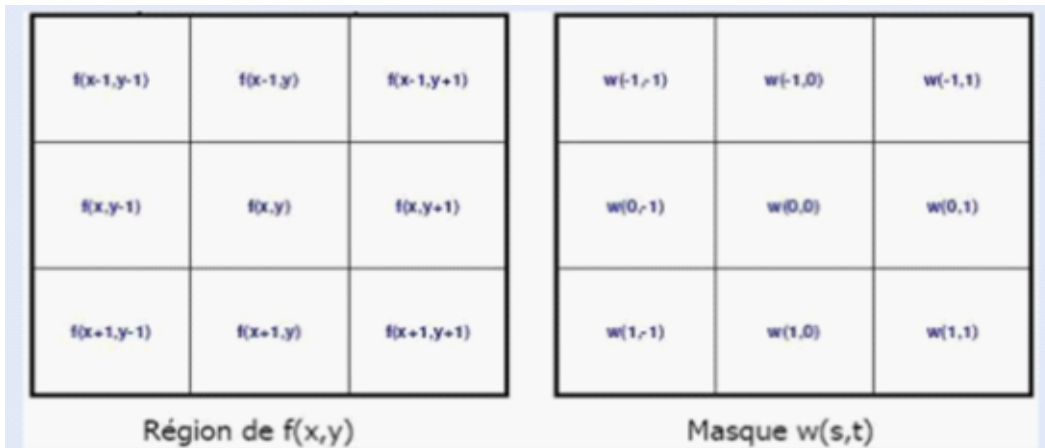
Ces opérateurs sont à considérer comme des filtres qui vont être corrélés à l'image. Les réponses impulsionnelles de ces filtres peuvent se présenter sous la forme de fonctions analytiques souvent d'une seule variable ou bien sous la forme de masques bi-dimensionnels. Dans les deux cas le filtrage a lieu en deux étapes : un filtrage suivant les lignes de l'image puis suivant les colonnes dans le cas d'une expression monodimensionnelle de la réponse impulsionnelle du filtre, une corrélation bi-dimensionnelle de l'image avec deux masques modélisant deux contours dans des directions orthogonales dans l'autre cas.

En pratique, il faut **approcher les gradients** pour travailler avec des gradients discrets. Les approximations les plus simples des dérivées directionnelles se font par **différences finies**. On peut les calculer par exemple à l'aide d'une **convolution** avec des noyaux très simples: par exemple, l'approximation de $\frac{\partial f}{\partial x}$ et $\frac{\partial f}{\partial y}$ se fait par convolution . En effet, dans ce cas, la formule générale de convolution discrète donne:

$$g(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 w(i, j) f(x + i, y + j) \quad (2.1)$$

tel que : $\frac{\partial f}{\partial x} \simeq (f * w_x)(x, y)$ et $\frac{\partial f}{\partial y} \simeq (f * w_y)(x, y)$

Relation à la notion de convolution...



Pour chaque opérateur, deux masques sont utilisés de façon à déterminer le gradient de l'image dans deux directions orthogonales.

Le masque le plus intuitif à mettre en œuvre est un masque à deux éléments(x, y)

Approximation de la dérivée [14]

l'approximation de $\frac{\partial f}{\partial x}$ se fait par convolution entre la fonction f et le masque w donnée par la formule suivante:

$$w = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

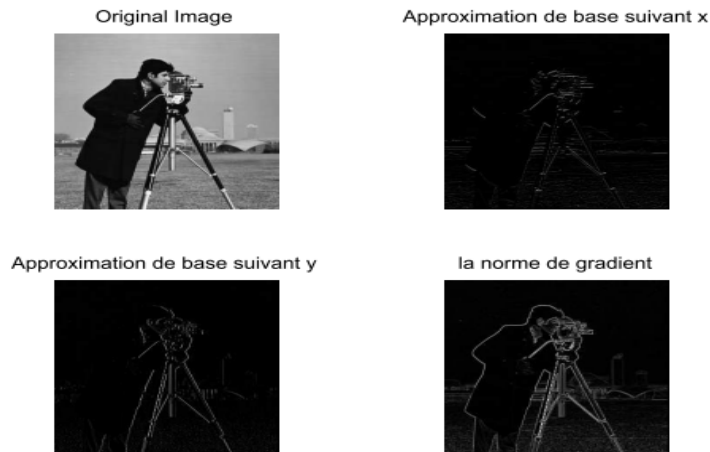
$$g(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 w(i, j) f(x + i, y + j) = f(x + 1, y) - f(x, y) \simeq \frac{\partial f}{\partial x}$$

l'approximation de $\frac{\partial f}{\partial y}$ se fait par convolution entre la fonction f et le masque w donnée par la formule suivante:

$$w = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$g(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 w(i, j) f(x + i, y + j) = f(x, y + 1) - f(x, y) \simeq \frac{\partial f}{\partial y}$$

La figure suivante propose une illustration de l'application de ces masques $\frac{\partial f}{\partial x}$ et $\frac{\partial f}{\partial y}$ et la norme du gradient $\sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$: **voire Annexe (1.1)**



Exemple d'application de l'approximation de base

les masques de Sobel [14]

l'approximation de $\frac{\partial f}{\partial x}$ se fait par convolution entre la fonction f et le masque w donnée par la formule suivante:

$$w = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

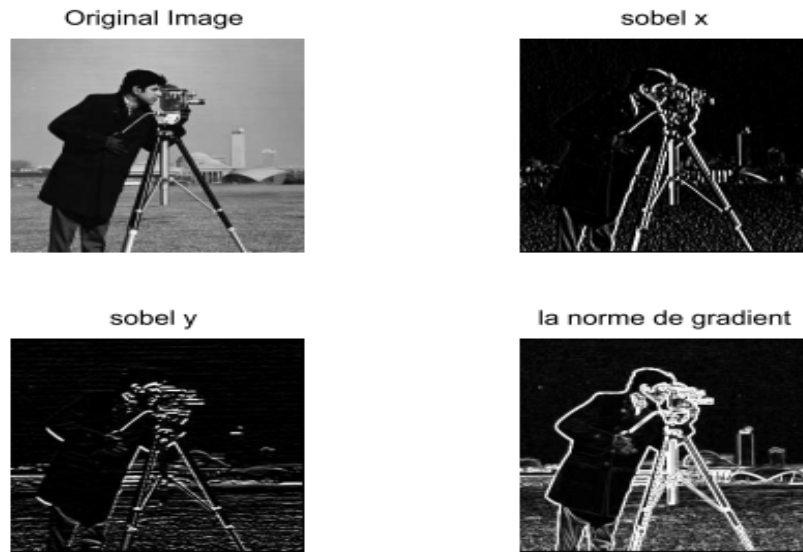
$$\begin{aligned} g(x, y) &= \sum_{i=-1}^1 \sum_{j=-1}^1 w(i, j) f(x+i, y+j) = \\ &= -f(x-1, y-1) - 2f(x, y-1) - f(x+1, y-1) + \\ &+ f(x-1, y+1) + 2f(x, y+1) + f(x+1, y+1) \end{aligned}$$

l'approximation de $\frac{\partial f}{\partial y}$ se fait par convolution entre la fonction f et le masque w donnée par la formule suivante:

$$w = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$\begin{aligned} g(x, y) &= \sum_{i=-1}^1 \sum_{j=-1}^1 w(i, j) f(x+i, y+j) = \\ &= -f(x-1, y-1) - 2f(x-1, y) - f(x, y+1) + \\ &+ f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1) \end{aligned}$$

La figure suivante propose une illustration de l'application de ces masques: **voire Annexe (1.2)**



Exemple d'application de l'approximation des masques de Sobel

les masques de Prewitt [14]

l'approximation de $\frac{\partial f}{\partial x}$ se fait par convolution entre la fonction f et le masque w donnée par la formule suivante:

$$w = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$g(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 w(i, j) f(x+i, y+j) =$$

$$-f(x-1, y-1) - 2f(x, y-1) - f(x+1, y-1) +$$

$$f(x-1, y+1) + 2f(x, y+1) + f(x+1, y+1)$$

l'approximation de $\frac{\partial f}{\partial y}$ se fait par convolution entre la fonction f et le masque w donnée par la formule suivante:

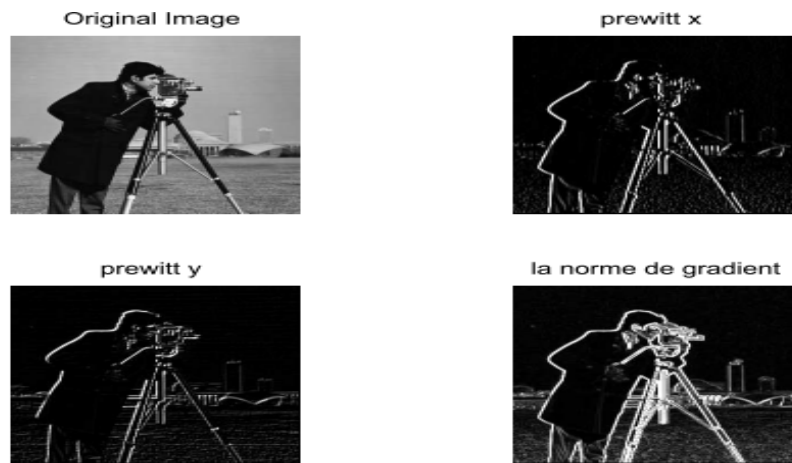
$$w = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

$$g(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 w(i, j) f(x + i, y + j) =$$

$$-f(x - 1, y - 1) - 2f(x - 1, y) - f(x, y + 1) +$$

$$f(x + 1, y - 1) + 2f(x + 1, y) + f(x + 1, y + 1)$$

La figure suivante propose une illustration de l'application de ces masques :voire **An-nexe (1.3)**



Exemple d'application de l'approximation des masques de prewitt

les masques de roberts [14]

l'approximation de $\frac{\partial f}{\partial x}$ se fait par convolution entre la fonction f et le masque w donnée par la formule suivante:

$$w = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

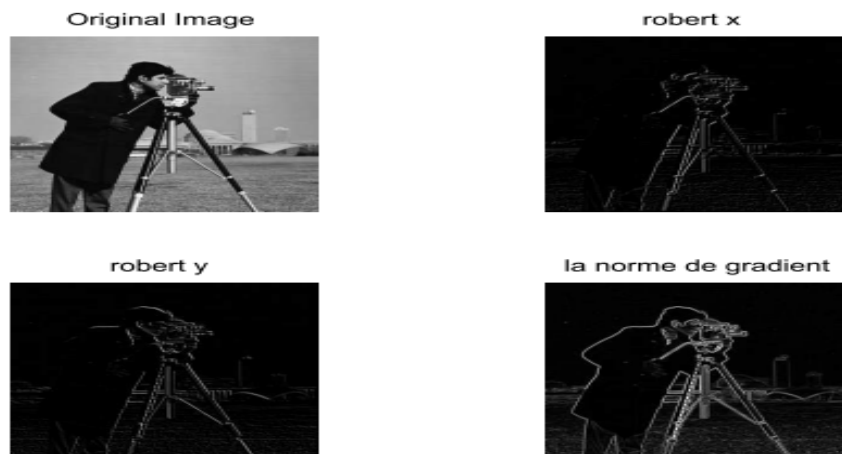
$$g(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 w(i, j) f(x + i, y + j) = -f(x, y) + f(x + 1, y + 1)$$

l'approximation de $\frac{\partial f}{\partial y}$ se fait par convolution entre la fonction f et le masque w donnée par la formule suivante:

$$w = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$g(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 w(i, j) f(x + i, y + j) = -f(x + 1, y) + f(x, y + 1)$$

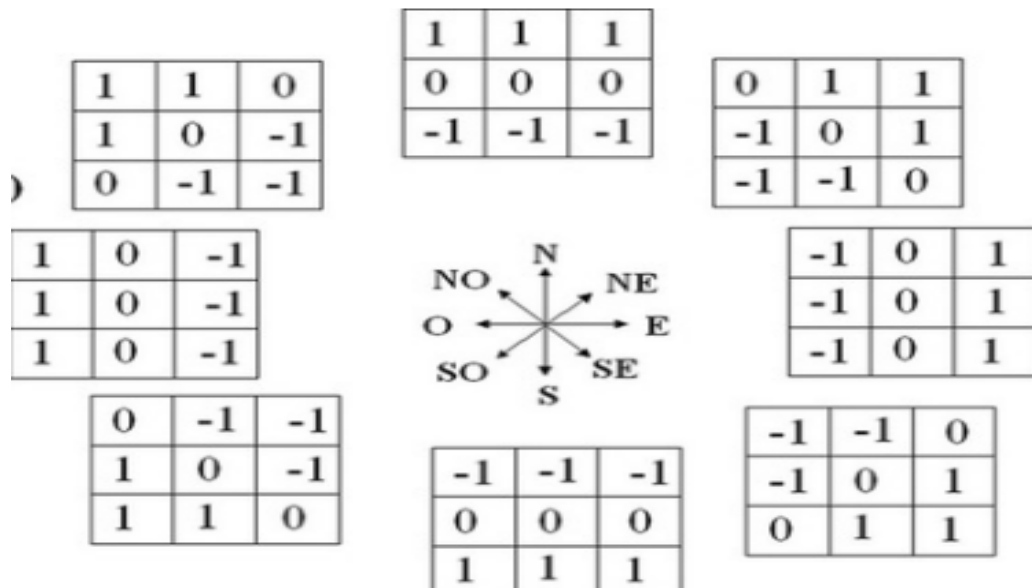
La figure suivante propose une illustration de l'application de ces masques :voire **An-**
nexe (1.4)



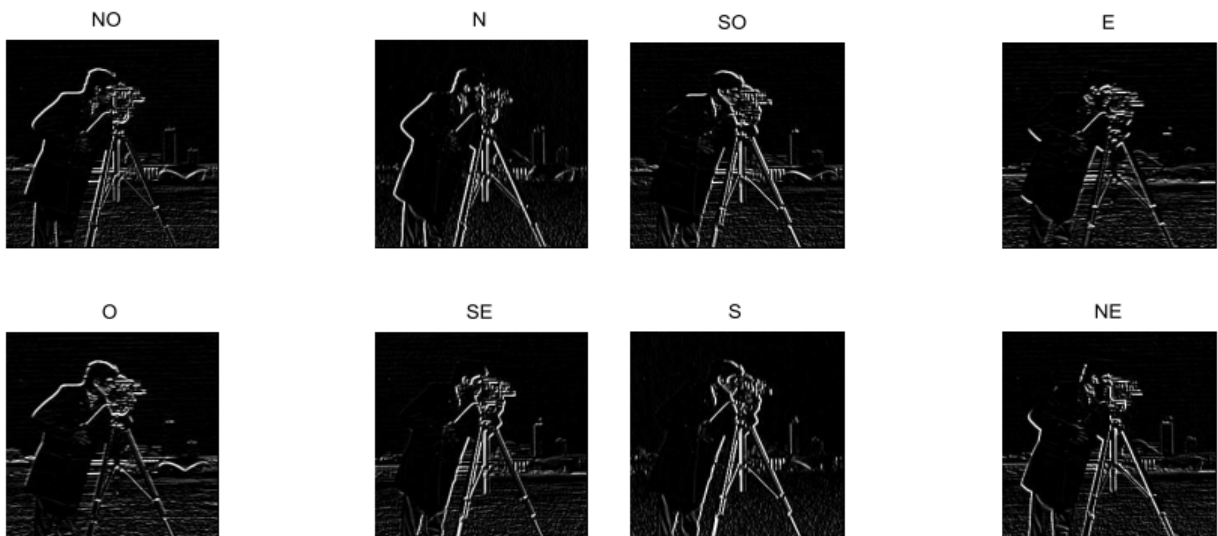
Exemple d'application de l'approximation des masques de robert

les masques de robinson [14]

les masques de robinson de niveau 3 sont :



La figure suivante propose une illustration de l'application de ces masques: **voire Annexe (1.5)**



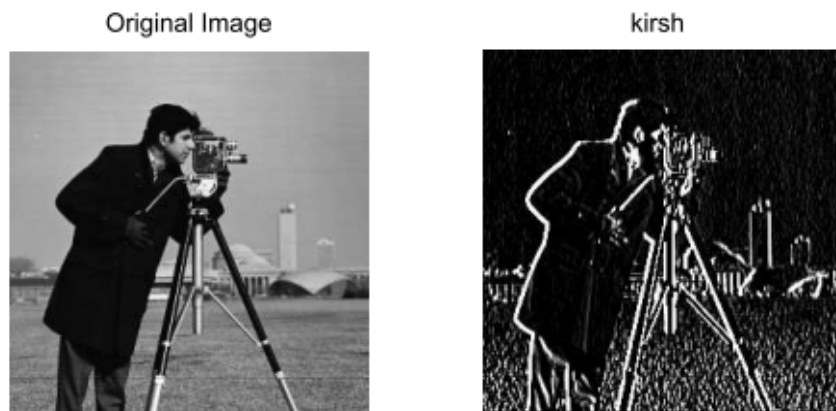
Exemple d'opérateurs Gradient boussole avec les masques de Robinson de niveau3

les masques de kirsh [14]

un exemple sur le masque de kirsh

$$w = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

La figure suivante propose une illustration de l'application de ce masque: **voire Annexe (1.6)**



Exemple d'application de l'approximation de masque de kirsh

2.1.3 Filtrage de Canny

L'objectif est bien de calculer le module du gradient de l'image analysée. Souvent avant d'appliquer le filtre de Canny, un filtrage préalable est opéré sur l'image au moyen d'un filtre gaussien. Les différentes étapes sont énumérées ci-après :

1-Convolution de l'image initiale avec un filtre passe-bas gaussien bi-dimensionnel (ou convolution 1D dans chacune des deux directions)

ie:

$$g(x, y) = f(x, y) * w(i, j)$$

$f(x, y)$: l'image initiale

tel que : $w(x, y)$: filtre passe-bas gaussien

$g(x, y)$: l'image lissée

2-Convolution de l'image lissée avec le filtre de Canny ou la dérivée de gaussienne dans les directions horizontale et verticale.

ie:

$$k(x, y) = g(x, y) * v(x, y)$$

$k(x, y)$: l'image de sortie

tel que : $g(x, y)$: l'image lissée

$v(x, y)$: filtre de canny ou la dérivée de gaussienne

3-Calcul du module du gradient à partir des deux images représentant les gradients de l'image filtrée passe-bas dans les directions horizontale et verticale.

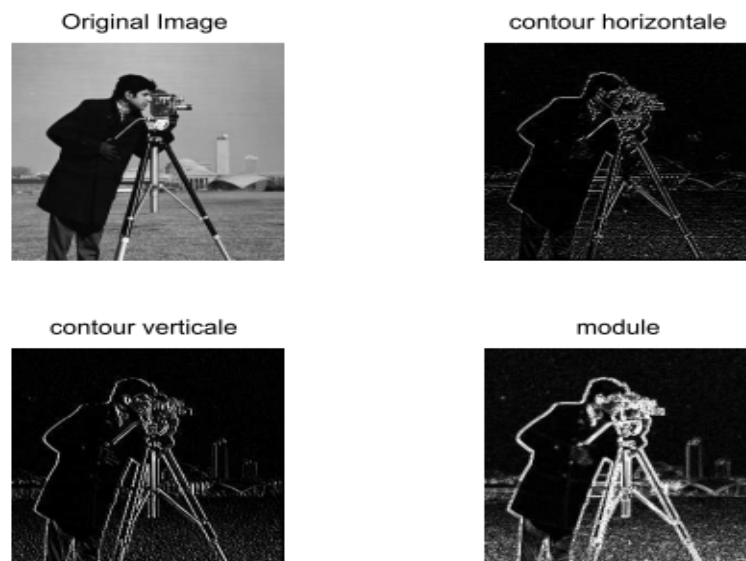
ie:

$$\|k(x, y)\| = \sqrt{k_1(x, y)^2 + k_2(x, y)^2}$$

tel que : $k_1(x, y)$: l'image dans les directions horizontales

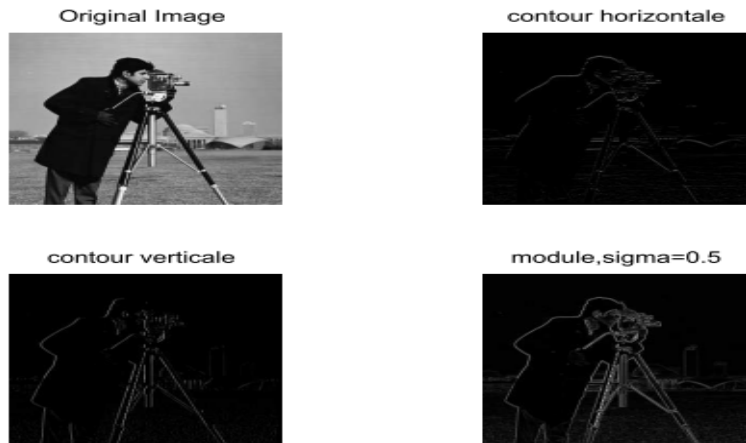
$k_2(x, y)$: l'image dans les directions verticales

La figure suivante propose une illustration de l'application du filtrage de Canny1: **voire Annexe (1.7a)**



Exemple d'application du filtrage de Canny

La figure suivante propose une illustration de l'application du filtrage de Canny2: **voire Annexe (1.7b)**



Obtention du module du gradient par filtrage de Canny

2.2 Méthode du passage à zéro Laplacien

2.2.1 Principe

De la même façon, l'approximation par différences finies la plus simple de la dérivée seconde est la convolution par le noyau $[1 \ -2 \ 1]$ pour l'approximation de $\frac{\partial^2 f}{\partial x^2}$ et $[1 \ -2 \ 1]^t$ pour l'approximation de $\frac{\partial^2 f}{\partial y^2}$. Le laplacien $\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$ peut donc être approché par l'un opérateurs linéaires suivants :

<i>Laplacien discret - 4</i>	<i>Laplacien discret - 8</i>	<i>Laplacien de Robinson</i>																											
<table border="1"> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>-4</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> </table>	0	1	0	1	-4	1	0	1	0	<table border="1"> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>-8</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	1	1	1	1	-8	1	1	1	1	<table border="1"> <tr><td>1</td><td>-2</td><td>1</td></tr> <tr><td>-2</td><td>4</td><td>-2</td></tr> <tr><td>1</td><td>-2</td><td>1</td></tr> </table>	1	-2	1	-2	4	-2	1	-2	1
0	1	0																											
1	-4	1																											
0	1	0																											
1	1	1																											
1	-8	1																											
1	1	1																											
1	-2	1																											
-2	4	-2																											
1	-2	1																											

Les opérateurs de gradient vus précédemment exploitent le fait qu'un contour dans une image correspond au maximum du gradient dans la direction orthogonale au contour. Ou le passage par zéro de la dérivée seconde d'une rupture d'intensité permet également de mettre en évidence le contour.

2.2.2 Opérateurs Laplacien

La dérivée seconde est donc déterminée par le calcul du Laplacien :

$$\Delta f = \nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{\partial f}{\partial y} \right)$$

qui peut s'écrire :

$$\nabla^2 f = \nabla_x (f(x+1, y) - f(x, y)) + \nabla_y (f(x, y+1) - f(x, y))$$

on peut définir :

$$\begin{aligned} \nabla_x (f(x+1, y) - f(x, y)) &= f(x+1, y) - f(x, y) - (f(x, y) - f(x-1, y)) \\ &= f(x+1, y) + f(x-1, y) - 2f(x, y) \end{aligned}$$

et

$$\begin{aligned} \nabla_y (f(x, y+1) - f(x, y)) &= f(x, y+1) - f(x, y) - (f(x, y) - f(x, y-1)) \\ &= f(x, y+1) + f(x, y-1) - 2f(x, y) \end{aligned}$$

Ainsi le Laplacien peut s'écrire :

$$\Delta f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

Cette opération de calcul de Laplacien peut alors être appliquée à une image par l'intermédiaire d'un filtrage avec le masque 3*3 suivant :

l'approximation de $\frac{\partial^2 f}{\partial x^2}$ se fait par convolution entre la fonction f et le masque w_1 donnée par la formule suivante:

$$w_1 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$g(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 w(i, j) f(x+i, y+j) = f(x-1, y) - 2f(x, y) + f(x+1, y)$$

l'approximation de $\frac{\partial^2 f}{\partial y^2}$ se fait par convolution entre la fonction f et le masque w_2 donnée par la formule suivante:

$$w_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$g(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 w(i, j) f(x + i, y + j) = f(x, y - 1) - 2f(x, y) + f(x, y + 1)$$

la norme du gradient $\left(\frac{\partial^2 f}{\partial x^2}\right) + \left(\frac{\partial^2 f}{\partial y^2}\right)$ se fait par convolution entre la fonction f et le masque w_3 donnée par la formule suivante:

$$w_3 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$g(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 w(i, j) f(x + i, y + j) = f(x, y - 1) + f(x, y + 1) - 4f(x, y) + f(x - 1, y) + f(x + 1, y)$$

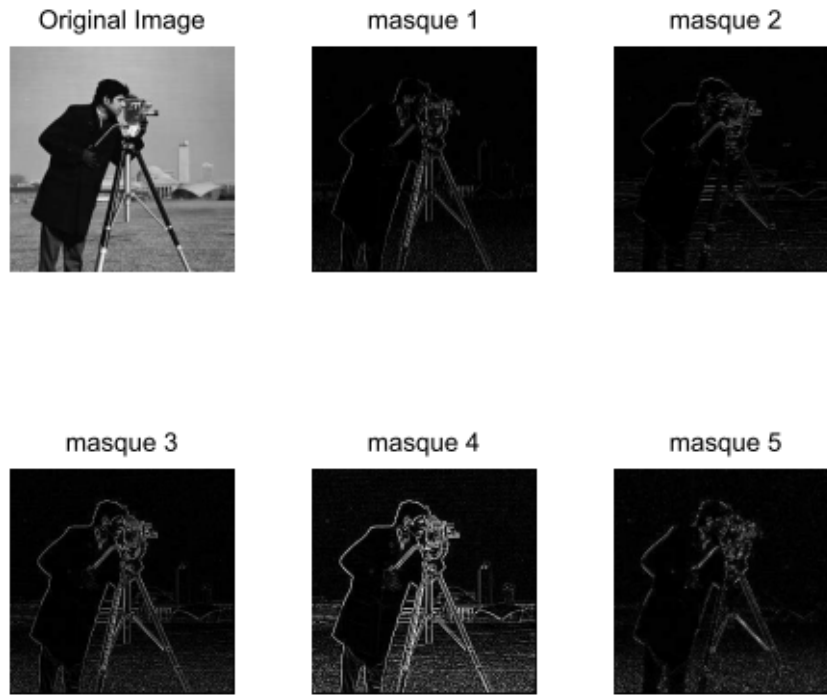
Ou se fait par convolution entre la fonction f et les masques w_1 et w_2 donnée par la formule suivante:

$$w_4 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad w_5 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

$$g_1(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 w(i, j) f(x + i, y + j) = f(x - 1, y - 1) + f(x, y - 1) + f(x + 1, y - 1) \\ + f(x - 1, y) - 8f(x, y) + f(x + 1, y) + f(x - 1, y + 1) + f(x, y + 1) + f(x + 1, y + 1)$$

$$g_2(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 w(i, j) f(x + i, y + j) = f(x - 1, y - 1) - 2f(x, y - 1) + f(x + 1, y - 1) \\ - 2f(x - 1, y) + 4f(x, y) - 2f(x + 1, y) + f(x - 1, y + 1) - 2f(x, y + 1) + f(x + 1, y + 1)$$

La figure suivante propose une illustration de l'application de ces masques voire **Annexe (1.8)**:



Exemple d'application de l'approximation des masques 1-2-3-4-5

Définition 2.2.2 (Le bruit) *Le bruit caractérise les parasites ou interférences d'un signal, c'est-à-dire les parties du signal déformées localement. Ainsi le bruit d'une image désigne les pixels de l'image dont l'intensité est très différente de celle des pixels voisins.*

Définition 2.2.3 (Seuillage global) *La première solution pour fixer le seuil est de procéder par tâtonnement à partir de l'image en noir et blanc résultante censée mettre en évidence les contours. Le seuil correspond à une valeur réelle dans l'intervalle $[0, 1]$. Au préalable, il est nécessaire de normaliser l'image du module du gradient afin que tous les pixels se trouvent également dans l'intervalle $[0, 1]$. Cette méthode est très simple mais peu efficace.*

Après le filtrage de l'image au moyen d'un de ces filtres, il s'agit de détecter les passages par zéro en conservant uniquement les passages les plus marqués. En effet, la technique est particulièrement sensible au bruit en raison de la double dérivation. Il s'agit donc de ne pas

considérer le bruit, qui peut très bien se traduire par des oscillations autour de zéro, comme un contour. C'est le rôle du seuil S qui va être utilisé dans cette approche pour ne prendre en compte que les passages par zéro d'amplitude relativement élevée correspondant à des vrais contours de l'image.

La figure suivante propose une illustration du laplacien avec un seuillage 0.2 **voire Annexe (1.8)**:



Exemple d'obtention des contours par utilisation du Laplacien suivi d'un seuillage 0.2

2.2.3 Filtrage de Marr et Hildreth

Le filtre de Marr ou Laplacien de Gaussienne ou chapeau mexicain

Cet opérateur est la combinaison d'un filtre Gaussien de paramètres de variance σ_X et σ_Y avec un opérateur Laplacien (4-connecté).

Il s'agit d'un détecteur de contours permettant de limiter les amplifications des hautes fréquences des dérivées secondes par une gaussienne de variance ajustable.

En vision informatique, l'algorithme de Marr-Hildreth est une méthode de détection des bords dans les images numériques. C'est, courbes continues où il existe des variations fortes et rapides de la luminosité de l'image. La méthode de détection de bord de Marr-Hildreth est simple et fonctionne en convolution de l'image avec le Laplacien de la fonction gaussienne, ou, comme une approximation rapide par la différence des Gaussiens. Alors, des passages à zéro sont détectés dans le résultat filtré pour obtenir les bords. L'opérateur d'image Laplacien-of-Gaussian est parfois aussi appelé l'ondelette du chapeau mexicain en

raison de sa forme visuelle lorsqu'il est renversé. David Marr et Ellen C. Hildreth sont deux des inventeurs. [8].

Le remède contre la sensibilité au bruit de la double dérivation est à la base des travaux de Marr qui ont débouché sur l'utilisation d'un filtre gaussien en guise de filtre passe-bas pour $h(x, y)$. Le choix de ce filtre a été établi en raison du principe d'Heisenberg qui stipule que l'on ne peut pas améliorer en même temps la résolution spatiale Δx et la résolution fréquentielle Δf :

$$\Delta x \Delta f \geq \frac{1}{4\pi}$$

Dans le cas d'une détection d'un contour, celui-ci étant localisé spatialement et en fréquence, il est clair que les intervalles Δx et Δf doivent être petits. Détecter une rupture d'intensité implique une analyse des pixels voisins mais le voisinage ne doit pas être trop étendu. Filtrer l'image a pour conséquence de réduire les variations d'intensité, ce qui doit être raisonnable. Le filtre doit en effet avoir une largeur spectrale limitée. Or c'est la fonction gaussienne qui permet d'obtenir l'égalité dans le principe d'Heisenberg, d'où ce choix qui se traduit par :

$$h(x, y) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}$$

La dérivée de la fonction gaussienne bi-dimensionnelle s'écrit alors :

$$h'(x, y) = -\frac{(x+y)}{\pi\sigma^4} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}$$

Et la dérivée seconde s'exprime :

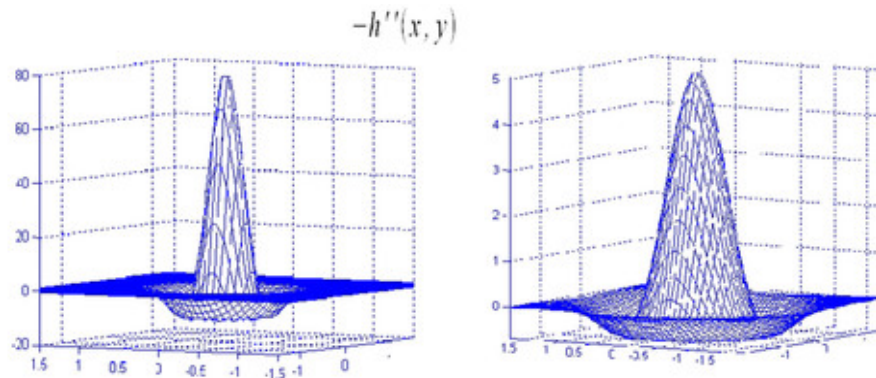
$$h''(x, y) = -\frac{1}{\pi\sigma^4} \left(1 - \frac{x^2+y^2}{2\sigma^2}\right) e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}$$

L'allure de $h''(x, y)$, réponse impulsionnelle du filtre Laplacien de Gaussienne (*LOG*), lui a valu l'appellation « chapeau mexicain ». La mise en œuvre d'une détection de contours avec le filtre *LOG* nécessite les étapes suivantes :

Convolution de l'image avec le filtre *LOG* (*filtre2D*) ;

Détection des passages par zéro de l'image résultante ;

Seuillage de l'image afin de ne considérer que les passages par zéro d'amplitude suffisante.



Allure de la fonction Laplacien de Gaussienne pour deux valeurs de σ

Le traitement repose sur cinq paramètres :

- N représente la taille du masque (matrice carrée) implantant le filtre LOG. N est impair.
- σ permet d'ajuster la taille du chapeau mexicain.
- Δx et Δy sont les pas d'échantillonnage utilisés pour discrétiser $h''(x, y)$. Généralement $\Delta x = \Delta y$
- S est le seuil qui permet de sélectionner les contours les plus marqués

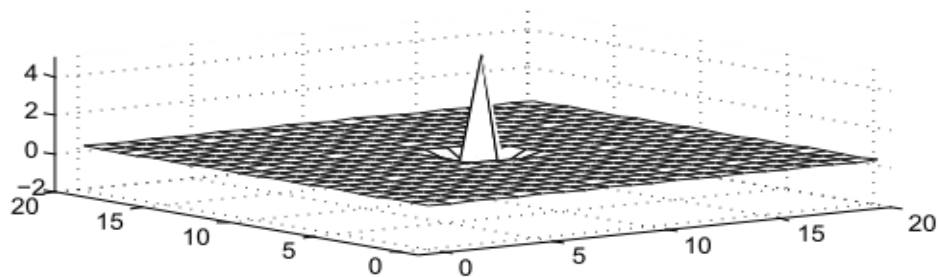
Remarque 2.2.3 *Il est à noter que le choix des paramètres N , σ et Δx ne doit pas se faire de façon indépendante. En effet, le masque, même de taille réduite, doit ressembler à un chapeau mexicain. Le problème ici est le même que celui que l'on rencontre lors de l'échantillonnage d'une fonction gaussienne. Le nombre de points N à considérer doit être tel que l'étendue occupe l'intervalle $[-3\sigma, 3\sigma]$.*

En fonction du pas d'échantillonnage, l'étendue spatiale vaut : .

Cette étendue peut aussi s'écrire en fonction de σ : avec k entier.

En prenant par exemple , il s'agit de choisir N et σ de sorte que l'étendue du chapeau mexicain soit pertinente. Pour le chapeau mexicain, la valeur de k doit être au moins de 4.

La figure suivante propose une illustration le **filtre de Marr et Hildreth** (**Laplacien de Gaussienne**) voire **Annexe (1.9)**:



Application du filtre *LOG* à la détection de contours sigma=0.6

2.3 Comparaison (Gradient/Laplacien)

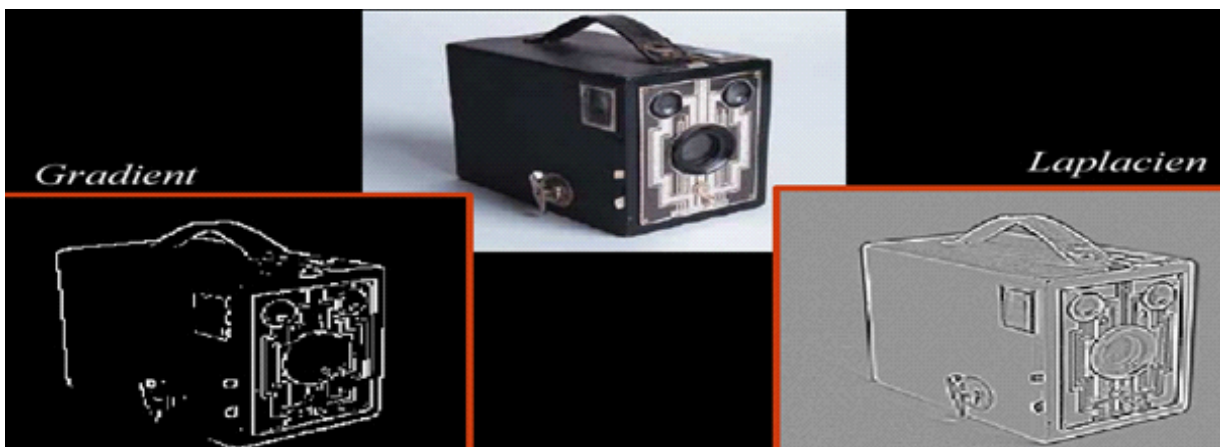
Propriétés de gradient

- Le gradient est orthogonal et normal aux courbes de niveaux de l'image.
- La direction du gradient est orthogonale à la frontière qui passe au point considéré et maximise la dérivée directionnelle.
- Sa norme est la valeur de cette dérivée.
- Le gradient est orienté dans le sens de croissance du champ, cette croissance étant d'autant plus forte que la normale du gradient est importante (d'après l'inégalité de Cauchy-Schwarz)

Propriété de laplacien

- L'opérateur laplacien est linéaire.

Illustrations:



2.3.1 Avantages et inconvénients

Le problème est que la dérivation accentue le bruit (pixels parasites de répartition aléatoire). Des filtres dérivés plus robustes, ont donc été proposés parmi lesquels Roberts, Prewitt, Sobel, Kirsch, Robinson.

a. Roberts

L'avantage de cet opérateur est sa rapidité; le calcul de l'amplitude consistant à faire la somme des valeurs absolues de la différence de deux pixels. De plus, il conserve les détails car il utilise des matrices $2 * 2$ et donc agit localement.

L'inconvénient de cet opérateur est son extrême sensibilité au bruit du fait de sa petite taille.

b. Prewitt

Cet opérateur est sensible aux bruit

c. Sobel

Le détecteur de Sobel donnera de meilleurs résultats au prix d'une complexité de calcul plus grande que celle de Roberts.

d. Kirch

Le coût d'implémentation des masques directionnels est élevé.

e. Laplacien

Bien que la méthode du laplacien respecte les contours formés, elle reste moins utilisée car elle est plus sensible au bruit.

2.3.2 Résultats

Les deux techniques sont assez proches. Globalement, le laplacien est plus sensible au bruit mais sa complexité est moindre surtout par rapport aux variantes directionnelles du gradient de type Sobel ou Prewitt. En effet, le laplacien nécessite une convolution, le gradient deux.

Au final, les points obtenus par ces deux techniques sont non-structurés, d'où la nécessité de post-traitements. Des filtrages ultérieurs simples comme le seuillage permettent de compenser le bruit mais leur enchainement est délicat.

Chapitre 3

Détection des contours et calcul fractionnaire

Aujourd'hui, Il existe des méthodes de détection de bord bien meilleures, Tel que le détecteur de bord par calcul fractionnaire .Il joue un rôle important dans l'amélioration de l'image et de montrer la frontière de l'image Nous examinerons cette amélioration conformément aux deux parties lorsque le dérivée d'ordre v (tel que v est réel) $0 \leq v \leq 1$ et $1 \leq v \leq 2$.Nous développerons aussi les travaux dans [1] [16]

3.1 Calcul fractionnaire d'ordre non entier $0 \leq v \leq 2$

3.1.1 Principe

On utilise le même principe pour le calcul fractionnaire entre 0 et 2.

ie: Une convolution entre l'image et un filtre pour trouver la dérivée fractionnaire entre 0 et 2.

Implémentation du filtre différentiel fractionnaire (FIR)

Le filtre différentiel fractionnaire peut être déduit du filtre de différenciation d'ordre entier .

Fonction de transfert du filtre par impulsion finie différentielle fractionnaire (*FIR*) comme suit:

$$D^v(z) = \left(\frac{1 - z^{-1}}{T}\right)^v$$

Référencement de l'extension de la série binomiale $(1+x)^v = 1 + vx + \sum \frac{v(v-1)\dots(v-k+1)}{k!} x^k$,

Et en utilisant z^{-1} au lieu de x , l'équation ci-dessus peut s'écrire:

$$\begin{aligned} D^v(z) &= \frac{1}{T^v} (1 - vz^{-1} + \sum \frac{v(v-1)\dots(v-i+1)}{i!} (-z^{-1})^{-i}) \\ &= \frac{1}{T^v} \sum (-1)^i \frac{T(v+1)}{T(i+1)T(v-i+1)} z^{-1} \end{aligned}$$

Il: T et la période d'échantillonnage, z est l'opérateur de déplacement et $\Gamma(*)$ est la fonction Gamma.

Afin de clarifier une plus étape par étape voir [14]

3.1.2 Opérateurs gradient fractionnaire

Selon l'impact limité de la fonction de transfert de filtre différentiel fractionnaire (FIR), la sélection de la N appropriée, la formule de différence finie approximative du premier ordre approximative obtenue

$$D^v(z) = \left(\frac{1 - z^{-1}}{T}\right)^v = \frac{1}{T^v} \sum (-1)^i \frac{T(v+1)}{T(i+1)T(v-i+1)} z^{-1}$$

Il peut obtenir une équation différentielle de signal

$$\frac{d^v f(t)}{dt^v} = f(t) + (-v)f(t-1) + \frac{v(v-1)}{2}f(t-2) + \dots + (-1)^n \frac{T(v+1)}{n!T(v-n+1)}f(t-n)$$

Pour les images numériques, sur la base de l'équation des différences de signaux, la formule de gradient différentiel fractionnaire peut être obtenue dans différentes directions

Direction horizontale ($D_{XL \leftrightarrow XR}^V = D_{XL}^V - D_{XR}^V$)

$$\frac{d^v f(x, y)}{dx^v} = a_1 f(x-1, y) - a_1 f(x+1, y) + \dots + a_n f(x-n, y) - a_n f(x+n, y)$$

$$\text{tel que: } a_1 = -v, a_2 = \frac{v(v-1)}{2}, a_3 = \frac{-v(v-1)(v-2)}{2}, a_4 = \frac{v(v-1)(v-2)(v-3)}{2}, \dots, a_n = (-1)^n \frac{T(v+1)}{n!T(v-n+1)}$$

En sélectionnant les éléments précédents n les quatre directions du gradient de masque différentiel fractionnaire peuvent être obtenues par le tronc.

Pour ne pas rendre trop grandes les erreurs de filtrage, nous choisissons les trois éléments précédents de la définition de différence d'ordre fractionnaire pour construire le masque de gradient différentiel fractionnel à directions différentes de $5 * 5$.

$$w_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ (v^2 - v)/2 & -v & 0 & v & (v^2 - v)/2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Le figure suivante de l'application de masque w_1 :voire **Annexe (1.10)**

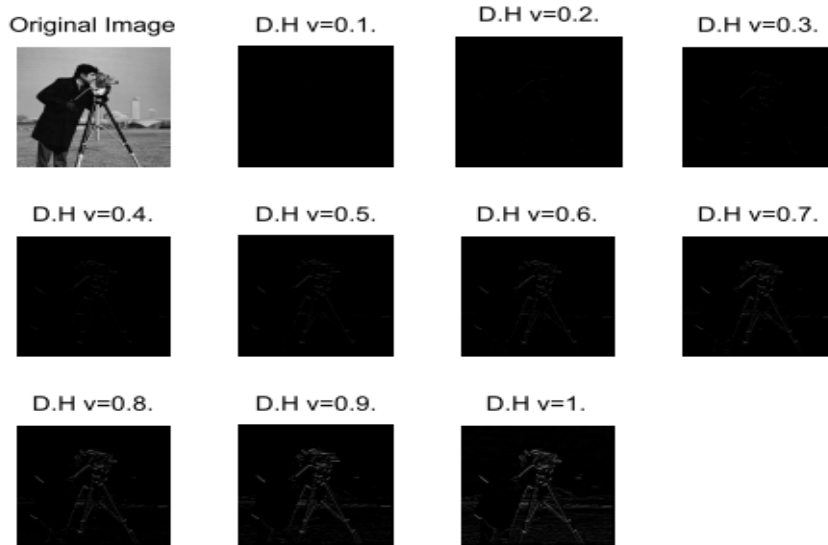


figure1

Direction verticale($D_{YU \leftrightarrow YD}^V = D_{YU}^V - D_{YD}^V$)

$$\frac{d^v f(x, y)}{dy^v} = a_1 f(x, y - 1) - a_1 f(x, y + 1) + \dots + a_n f(x, y - n) - a_n f(x, y + n)$$

le masque de gradient différentiel fractionnel à directions différentes de $5 * 5$:

$$w_2 = \begin{bmatrix} 0 & 0 & (v^2 - v)/2 & 0 & 0 \\ 0 & 0 & -v & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & v & 0 & 0 \\ 0 & 0 & (v - v^2)/2 & 0 & 0 \end{bmatrix}$$

Le figure suivante de l'application de masque w_2 :voire **Annexe (1.11)**

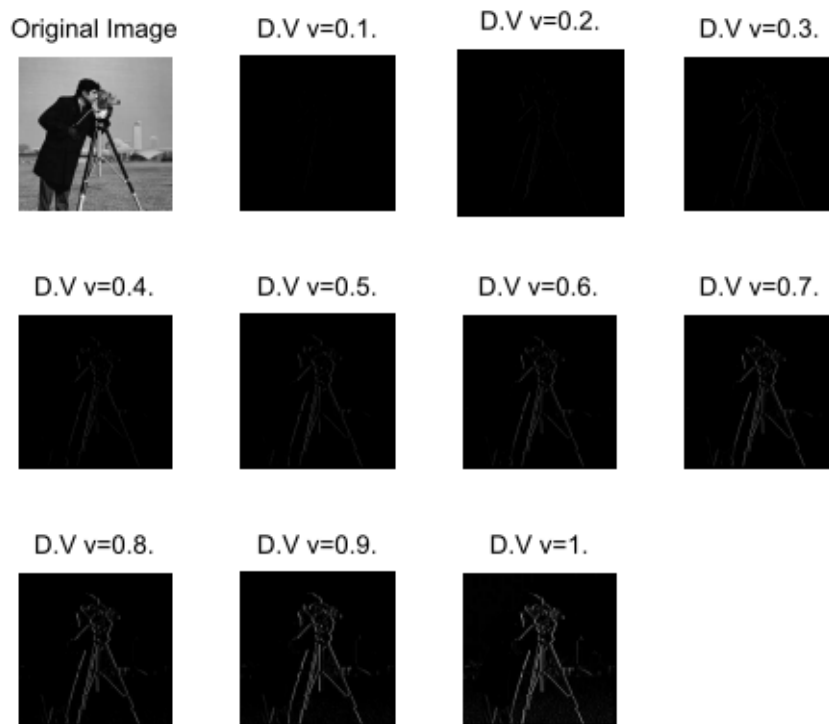


figure 2

Le figure suivante de l'application de norme des masques w_1 et w_2 : **voire Annexe (1.12)**

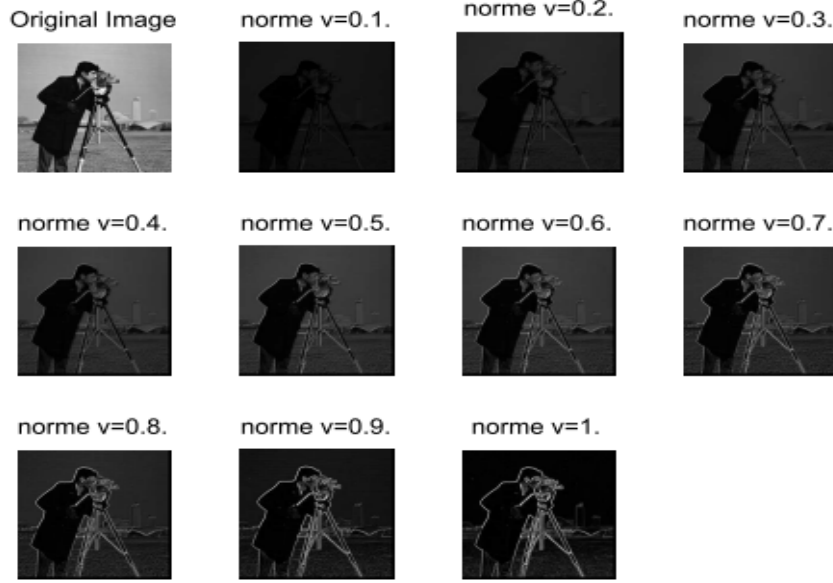


figure 3

3.1.3 Opérateurs Laplacien fractionnaire

Selon l'impact limité de la fonction de transfert de filtre différentiel fractionnaire (FIR), la sélection de la N appropriée, la formule de différence finie approximative du premier ordre approximative obtenue

$$D^v(z) = \left(\frac{1 - z^{-1}}{T}\right)^v = \frac{1}{T^v} \sum (-1)^i \frac{T(v+1)}{T(i+1)T(v-i+1)} z^{-1}$$

Il peut obtenir une équation différentielle de signal

$$\frac{d^v f(t)}{dt^v} = f(t) + (-v)f(t-1) + \frac{v(v-1)}{2}f(t-2) + \dots + (-1)^n \frac{T(v+1)}{n!T(v-n+1)}f(t-n)$$

Pour les images numériques, sur la base de l'équation des différences de signaux, la formule de gradient différentiel fractionnaire peut être obtenue dans différentes directions

Direction horizontale ($D_{XL \leftrightarrow XR}^V = D_{XL}^V - D_{XR}^V$)

$$\frac{d^v f(x, y)}{dx^v} = a_1 f(x-1, y) - a_1 f(x+1, y) + \dots + a_n f(x-n, y) - a_n f(x+n, y)$$

3.1. Calcul fractionnaire d'ordre non entier $0 \leq v \leq 2$

tel que: $a_1 = -v, a_2 = \frac{v(v-1)}{2}, a_3 = \frac{-v(v-1)(v-2)}{2}, a_4 = \frac{v(v-1)(v-2)(v-3)}{2}, \dots, a_n = (-1)^n \frac{T(v+1)}{n!T(v-n+1)}$

En sélectionnant les éléments précédents n les quatre directions du gradient de masque différentiel fractionnaire peuvent être obtenues par le tronç.

Pour ne pas rendre trop grandes les erreurs de filtrage, nous choisissons les trois éléments précédents de la définition de différence d'ordre fractionnaire pour construire le masque de gradient différentiel fractionnel à directions différentes de $5 * 5$.

$$w_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ (v^2 - v)/2 & -v & 0 & v & (v^2 - v)/2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Le figure suivante de l'application de masque w_1 : **voire Annexe (1.15)**

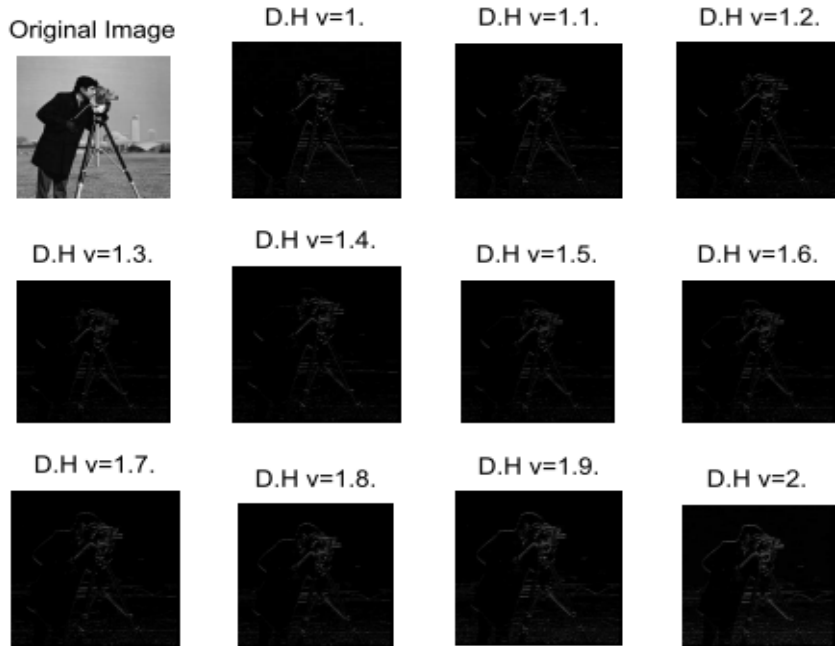


figure1

Direction verticale ($D_{YU \leftrightarrow YD}^V = D_{YU}^V - D_{YD}^V$)

$$\frac{d^v f(x, y)}{dy^v} = a_1 f(x, y - 1) - a_1 f(x, y + 1) + \dots + a_n f(x, y - n) - a_n f(x, y + n)$$

le masque de gradient différentiel fractionnel à directions différentes de $5 * 5$:

$$w_2 = \begin{bmatrix} 0 & 0 & (v^2 - v)/2 & 0 & 0 \\ 0 & 0 & -v & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & v & 0 & 0 \\ 0 & 0 & (v - v^2)/2 & 0 & 0 \end{bmatrix}$$

Le figure suivante de l'application de masque w_2 :voire **Annexe (1.16)**

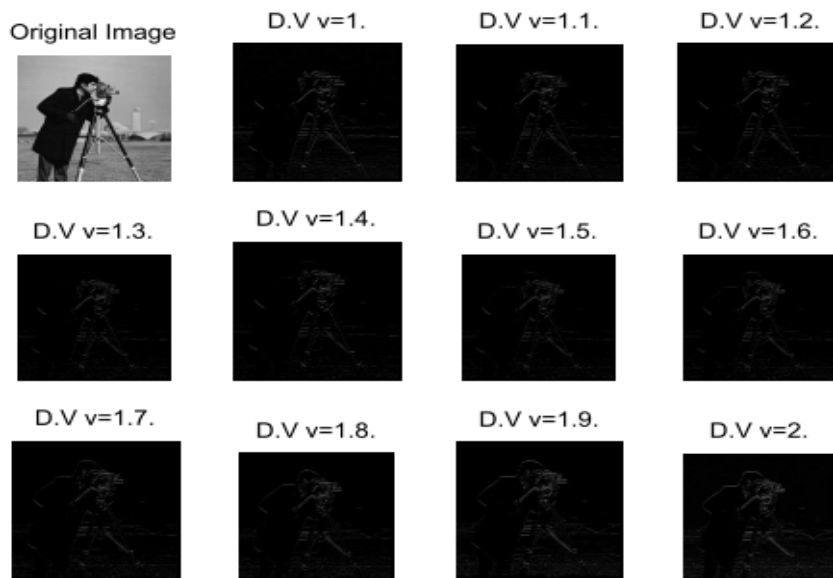


figure 2

Le figure suivante de l'application de norme des masques w_1 et w_2 :voire Annexe (1.17)

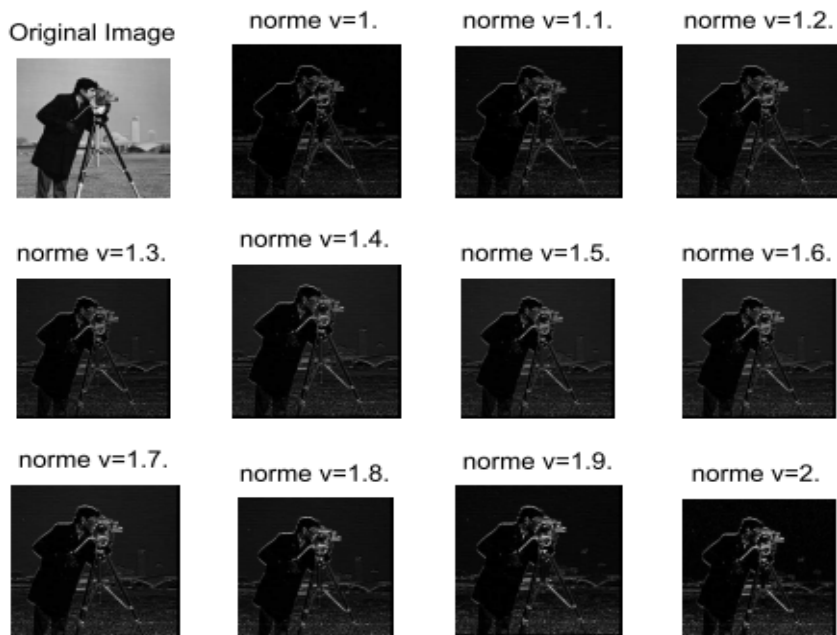


figure 3

3.2 Structure d'opérateur gaussien d'ordre non entier

$$1 \leq v \leq 2$$

3.2.1 Principe

La performance des algorithmes de détection des arêtes sera influencée par le bruit inévitablement. Par conséquent, un certain degré de lissage doit être utilisé avant l'opération de détection sur des images polluées par le bruit.

L'opérateur de moyenne gaussienne effectue de manière excellente le lissage d'image, et comme la surface de la fonction gaussienne bidimensionnelle a une forme de cloche, l'effet de lissage peut être ajusté en contrôlant la variance de la fonction gaussienne et de la taille du gabarit pour générer différents gabarits pour différentes images polluées par le bruit.

3.2.2 Opérateurs gaussien fractionnaire

Fonction noyau gaussien:

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Où σ dénote une variance qui assure la sommation des coefficients de gabarit est 1.

Supposons que $P(x, y)$ est une image de taille $M \times N$. Sa circonvolution gaussienne est:

$$F = g(x, y) * p(x, y)$$

Où F est la sortie de ce filtre gaussien. Nous effectuons un différentiel d'ordre non entier sur l'image filtrée F avec l'ordre v ($1 < v < 2$), à savoir:

$$\frac{\partial^v F}{\partial(x, y)^v} = \frac{\partial^v (g(x, y) * p(x, y))}{\partial(x, y)^v}$$

Le différentiel d'image sur la direction horizontale et verticale est:

$$\frac{\partial^v F}{\partial x^v} = \frac{\partial^v (g(x, y) * p(x, y))}{\partial x^v} = \frac{\partial^v g(x, y)}{\partial x^v} * p(x, y)$$

$$\frac{\partial^v F}{\partial y^v} = \frac{\partial^v (g(x, y) * p(x, y))}{\partial y^v} = \frac{\partial^v g(x, y)}{\partial y^v} * p(x, y)$$

Ici les opérateurs sur les directions horizontale et verticale sont:

$$\frac{\partial^v g(x, y)}{\partial x^v} \text{ et } \frac{\partial^v g(x, y)}{\partial y^v}$$

Plus loin, On déduit les fonctions du noyau dans deux directions à partir de l'équation 1 à Eq.7. Laissez KX et KY les fonctions du noyau dans X – direction et Y – direction.

Prenez la fonction de Gauss comme l'équation originale de deux directions 'Fonction de noyau fractionnaire:

$$KX = g(x, y) \left[1 + (-v) e^{\frac{2x+1}{2\sigma^2}} + \frac{(-v)(-v+1)}{2} e^{\frac{4x+4}{2\sigma^2}} + \frac{(-v)(-v+1)(-v+2)}{6} e^{\frac{6x+9}{2\sigma^2}} \right] \quad (3.1)$$

$$KY = g(x, y) \left[1 + (-v) e^{\frac{2y+1}{2\sigma^2}} + \frac{(-v)(-v+1)}{2} e^{\frac{4y+4}{2\sigma^2}} + \frac{(-v)(-v+1)(-v+2)}{6} e^{\frac{6y+9}{2\sigma^2}} \right] \quad (3.2)$$

3.2. Structure d'opérateur gaussien d'ordre non entier $1 \leq v \leq 2$

Le modèle différentiel peut être construit selon la fonction du noyau KX et KY . Pour les modèles différentiels, trois paramètres peuvent être ajustés pour contrôler les coefficients de modèle:

- (1) La taille de la fenêtre de modèle (de manière souhaitable 3, 5, 7, 9 et 11). Ce paramètre doit être ajusté en fonction d'images bruyantes. La valeur de la taille devrait être plus grande sur les images les plus polluées.
- (2) La variance de la fonction gaussienne (généralement 1.0).
- (3) v , l'ordre du différentiel fractionnaire ($1 < v < 2$).

Exemple 3.2.1 *Le choix du paramètre de modèle a un grand impact sur la performance pratique. Un modèle de taille = 3, $\sigma = 1,0$, $v = 1,4$ est illustré comme suit:*

Tab1. Le modèle de fonction du noyau KX

Tab2. Le modèle de fonction du noyau KY

<i>Modèle KX</i>			<i>Modèle KY</i>		
0.0406	0.0669	0.0406	0.0406	0.0669	0.0406
0.0183	0.0301	0.0183	0.0183	0.0301	0.0183
-0.0595	-0.0980	-0.0595	-0.0595	-0.0980	-0.0595

La figure suivante propose une illustration de l'application de ces masques **tab 1:voire Annexe (1.13)**:

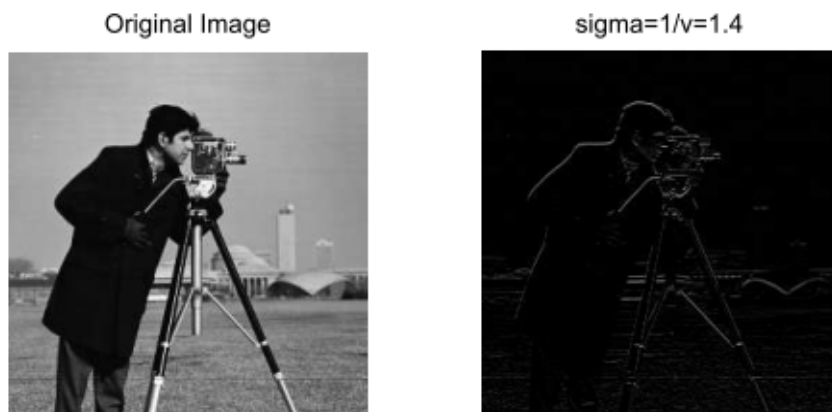


figure1

La figure suivante propose une illustration de l'application un modèle de taille= $3/\sigma = 1/v = \text{change:voire Annexe (1.14)}$:

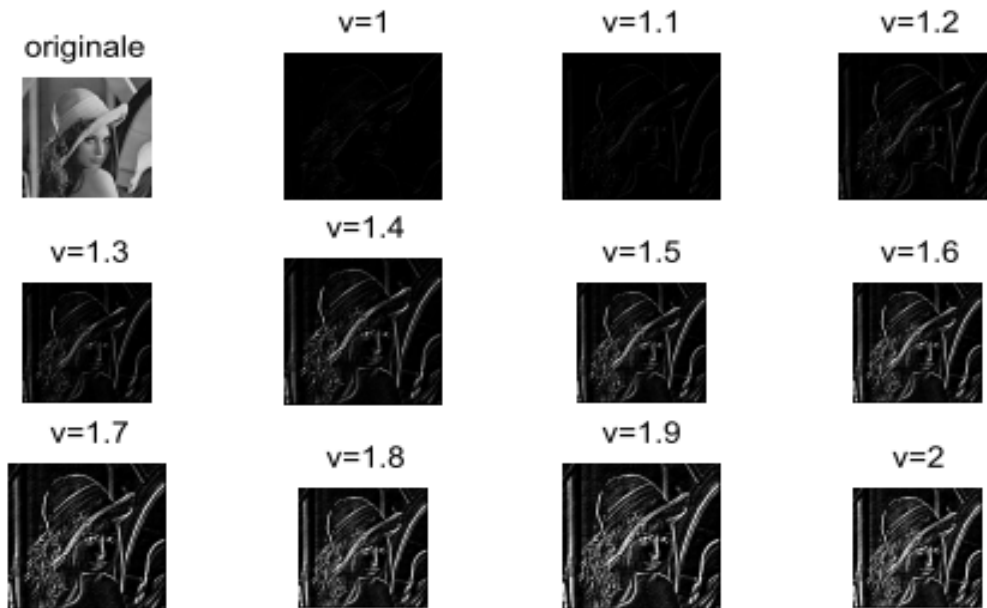


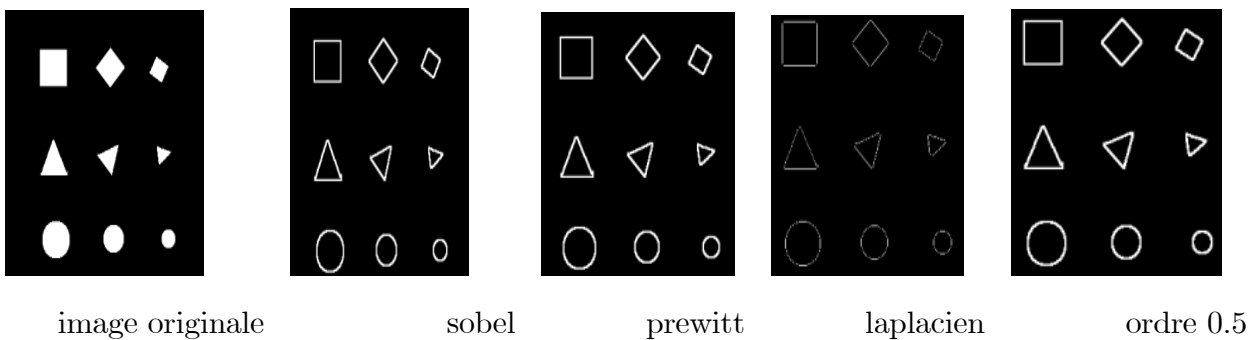
figure2

3.3 Comparaison générale

Dans la suite on peut présente une comparaisons entre les résultats obtenue avec le gradient fractionnaire et ceux obtenue par Laplacien fractionnaire.

3.3.1 L'image artificielle(sans bruit/bruit)

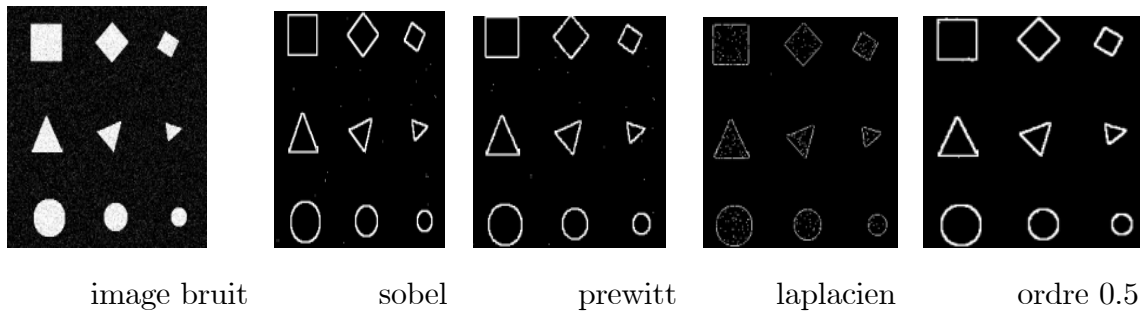
Image sans bruit



On peut voir à partir de cette figure que l'opérateur du gradient différentiel du premier ordre et l'opérateur du gradient différentiel fractionnaire génèrent des réponses plus larges dans la région proche du bord de l'image, ce qui affecte la précision de l'emplacement du bord.

Le deuxième opérateur de détection de bord différentiel extrait seulement une largeur de bord de pixel, ce qui profite à un emplacement de bord précis

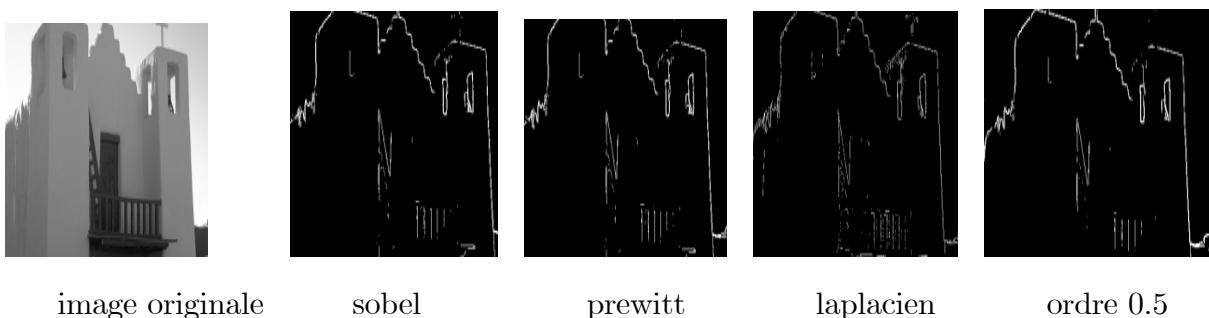
Image bruit



En comparant ces images, nous avons constaté que les opérateurs de gradient différentiel à ordre entier traditionnel, en particulier le second opérateur de gradient différentiel, sont sensibles au bruit. On peut le voir à partir de cette dernière figure, la capacité anti-bruit de l'opérateur de gradient différentiel fractionnaire est plus forte que l'opérateur de gradient différentiel à ordre entier.

3.3.2 L'image naturelle(sans bruit/bruit)

Image sans bruit



On peut voir à partir de cette figure, les résultats de détection de bord de l'opérateur de gradient différentiel de premier ordre ou l'opérateur de gradient différentiel fractionnaire est

approximatif. Le deuxième opérateur de gradient différentiel extrait seulement une largeur de bord de pixel, qui montre la précision de positionnement du bord.

Image bruit



image bruit

sobel

prewitt

laplacien

ordre0.5

On voit à partir de cette figure que l'opérateur de gradient différentiel à ordre entier classique est très sensible au bruit, où l'extraction de bord du second opérateur de gradient différentiel a été submergée dans le bruit et n'a pas été capable de distinguer.

Alors que l'opérateur différentiel de 0,5 ordre a extrait des informations de bord avec un bruit relativement faible, ce qui montre que, lorsque l'ordre différentiel plus petit ($0 < v < 1$), l'opérateur de gradient différentiel fractionné a beaucoup plus de performances anti-bruit que les opérateurs de gradient différentiel à ordre entier.

3.3.3 Résultats

si v entre 0 et 1 :

Chaque fois que la variable v change de 0 à 1 a augmenté la clarté des bords de l'image.

si v entre 1 et 2 :

Quelle que soit la variable v change 1 à 2. les Bords de l'image sont présentes, mais avec une grande précision à l'approche à 2 et augmenter l'éclairage de l'image.

Conclusion

Dans ce travail on a présenté les méthodes de détection des contours d'images. En utilisant la méthode du gradient et la méthode du passage à zéro du Laplacien, ces méthodes sont les plus utilisées dans ce domaine. On a étudié particulièrement dans ce mémoire l'amélioration du contour d'image par la méthode du gradient fractionnaire et la méthode du Laplacien fractionnaire et l'opérateur Gaussien fractionnaire, à la fin, on a présenté une étape comparative entre ces différentes méthodes. On déduit que les méthodes fractionnaires sont les plus efficace pour l'amélioration de l'image et la détection des contours.

Annexe

programme 1

(1.1)

```
clc;
close all;
clear all;
a=imread('f:\tt.tif'); %l'image d'entrée
b=im2double(a); %Convertit l'image d'intensité I en double précision, en rééchantillonnant
les données si nécessaire.
%Si l'image d'entrée est de classe double, l'image de sortie est identique.
[m,n]=size(a); %la taille de la matrice
L(1:m,1:n)=0; %matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
L(i,j)=b(i+1,j)-b(i,j);
end;
end;
P(1:m,1:n)=0;%matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
P(i,j)=b(i,j+1)-b(i,j);
end;
end;
R=sqrt(L.^2+P.^2);
```

```

figure;
subplot(2,2,1)
imshow(b)
title('Original Image');
subplot(2,2,2)
imshow(L)
title('Approximation de base suivant x');
subplot(2,2,3)
imshow(P)
title('Approximation de base suivant y');
subplot(2,2,4)
imshow(R)
title('la norme de gradient');

```

programme 2

(1.2)

```

clc;
close all;
clear all;
a=imread('f:\tt.tif'); %l'image d'entrée
b=im2double(a); %Convertit l'image d'intensité I en double précision, en réétalant
les données si nécessaire.
%Si l'image d'entrée est de classe double, l'image de sortie est identique.
[m,n]=size(a); %la taille de la matrice
L(1:m,1:n)=0; %matrice zéro d'ordr [m n]
for i=1:m-2;
for j=1:n-2;
L(i,j)=-1*b(i,j)-2*b(i+1,j)-1*b(i+2,j)+1*b(i,j+2)+2*b(i+1,j+2)+1*b(i+2,j+2);
end;
end;

```

```
P(1:m,1:n)=0;%matrice zéro d'ordr [m n]
for i=1:m-2;
for j=1:n-2;
P(i,j)=-1*b(i,j)+1*b(i+2,j)-2*b(i,j+1)+2*b(i+2,j+1)-1*b(i,j+2)+1*b(i+2,j+2);;
end;
end;
R=sqrt(L.^2+P.^2);
figure;
subplot(2,2,1)
imshow(b)
title('Original Image');
subplot(2,2,2)
imshow(L)
title('sobel x');
subplot(2,2,3)
imshow(P)
title('sobel y');
subplot(2,2,4)
imshow(R)
title('la norme de gradient');
```

programme 3

(1.3)

```
clc;
close all;
clear all;
a=imread('f:\tt.tif'); %l'image d'entrée
b=im2double(a); %Convertit l'image d'intensité I en double précision, en réétalant
les données si nécessaire.
%Si l'image d'entrée est de classe double, l'image de sortie est identique.
```

```
[m,n]=size(a); %la taille de la matrice
L(1:m,1:n)=0; %matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
L(i,j)=1*b(i,j)+1*b(i+1,j)+1*b(i+2,j)-1*b(i,j+2)-1*b(i+1,j+2)-1*b(i+2,j+2);
end;
end;
P(1:m,1:n)=0;%matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
P(i,j)=1*b(i,j)+1*b(i+1,j)+1*b(i+2,j)-1*b(i+2,j)-1*b(i+2,j+1)-1*b(i+2,j+2);
end;
end;
R=sqrt(L.^2+P.^2);
figure;
subplot(2,2,1)
imshow(b)
title('Original Image');
subplot(2,2,2)
imshow(L)
title('prewitt x ');
subplot(2,2,3)
imshow(P)
title('prewitt y');
subplot(2,2,4)
imshow(R)
title('la norme de gradient');
```

```
clc;
close all;
clear all;
a=imread('f:\tt.tif'); %l'image d'entrée
b=im2double(a); %Convertit l'image d'intensité I en double précision, en rééchantillonnant
les données si nécessaire.
%Si l'image d'entrée est de classe double, l'image de sortie est identique.
[m,n]=size(a); %la taille de la matrice
L(1:m,1:n)=0; %matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
L(i,j)=-1*b(i+1,j+1)+1*b(i+2,j+2);
end;
end;
P(1:m,1:n)=0;%matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
P(i,j)=-1*b(i+2,j+1)+1*b(i+1,j+2);
end;
end;
R=sqrt(L.^2+P.^2);
figure;
subplot(2,2,1)
imshow(b)
title('Original Image');
subplot(2,2,2)
imshow(L)
title('robert x ');
subplot(2,2,3)
imshow(P)
```

```

title('robert y');
subplot(2,2,4)
imshow(R)
title('la norme de gradient');

```

programme 5

(1.5)

```

clc;
close all;
clear all;
a=imread('f:\tt.tif'); %l'image d'entrée
b=im2double(a); %Convertit l'image d'intensité I en double précision, en rééchantillonnant
les données si nécessaire.
%Si l'image d'entrée est de classe double, l'image de sortie est identique.
[m,n]=size(a); %la taille de la matrice
NO(1:m,1:n)=0; %matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
NO(i,j)=b(i,j)+b(i+1,j)+b(i,j+1)-b(i+2,j+1)-b(i+1,j+1)-b(i+2,j+2);
end;
end;
N(1:m,1:n)=0;%matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
N(i,j)=b(i,j)+b(i+1,j)+b(i+2,j)-b(i,j+2)-b(i+1,j+2)-b(i+2,j+2);
end;
end;
O(1:m,1:n)=0;%matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:m-2;

```

```

O(i,j)=b(i,j)-b(i+2,j)+b(i,j+1)-b(i+2,j+1)+b(i,j+2)-b(i+2,j+2);
end;
end;
SE(1:m,1:n)=0; %matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
SE(i,j)=-b(i,j)-b(i+1,j)-b(i,j+1)+b(i+2,j+1)+b(i+1,j+2)+b(i+2,j+2);
end;
end;
SO(1:m,1:n)=0;%matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
SO(i,j)=-b(i+1,j)-b(i+2,j)+b(i,j+1)-b(i+2,j+1)+b(i,j+2)+b(i+1,j+2);
end;
end;
E(1:m,1:n)=0;%matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:m-2;
E(i,j)=-b(i,j)+b(i+2,j)-b(i,j+1)+b(i+2,j+1)-b(i,j+2)+b(i+2,j+2);
end;
end;
S(1:m,1:n)=0; %matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
S(i,j)=-b(i,j)-b(i+1,j)-b(i+2,j)+b(i,j+2)+b(i+1,j+2)+b(i+2,j+2);
end;
end;
NE(1:m,1:n)=0;%matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;

```

```
NE(i,j)=b(i+1,j)+b(i+2,j)-b(i,j+1)+b(i+2,j+1)-b(i,j+2)-b(i+1,j+2);
end;
end;
figure;
subplot(2,2,1)
imshow(NO)
title('NO ');
subplot(2,2,2)
imshow(N)
title('N');
subplot(2,2,3)
imshow(O)
title('O');
subplot(2,2,4)
imshow(SE)
title('SE');
figure;
subplot(2,2,1)
imshow(SO)
title('SO ');
subplot(2,2,2)
imshow(E)
title('E');
subplot(2,2,3)
imshow(S)
title('S');
subplot(2,2,4)
imshow(NE)
title('NE');
```

programme 6

(1.6)

```

clc;
close all;
clear all;
a=imread('f:\tt.tif'); %l'image d'entrée
b=im2double(a); %Convertit l'image d'intensité I en double précision, en rééchantillonnant
les données si nécessaire.
%Si l'image d'entrée est de classe double, l'image de sortie est identique.
[m,n]=size(a); %la taille de la matrice
L(1:m,1:n)=0; %matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
L(i,j)=5*b(i,j)+5*b(i+1,j)+5*b(i+2,j)-3*b(i,j+1)-3*b(i+2,j+1)-3*b(i,j+2)-3*b(i+1,j+2);
end;
end;
figure;
subplot(1,2,1)
imshow(b)
title('Original Image');
subplot(1,2,2)
imshow(L)
title('kirsh ')

```

programme 7a

(1.7a)

```

clc;
close all;
clear all;
a=imread('f:\tt.tif'); %l'image d'entrée

```

```
b=im2double(a); %Convertit l'image d'intensité I en double précision, en rééchantillonnant
les données si nécessaire.

%Si l'image d'entrée est de classe double, l'image de sortie est identique.
[m,n]=size(a); %la taille de la matrice
%approx de base
l(1:m,1:n)=0; %matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
l(i,j)=b(i+1,j)-b(i,j);
end;
end;
p(1:m,1:n)=0;%matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
p(i,j)=b(i,j+1)-b(i,j);
end;
end;
%canny
s(1:m,1:n)=0; %matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
s(i,j)=l(i,j)+l(i+1,j)+l(i+2,j)+l(i,j+1)-2*l(i+1,j+1)+l(i+2,j+1)-l(i,j+2)-l(i+1,j+2)-
end;
end;
x(1:m,1:n)=0;%matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
x(i,j)=p(i,j)+p(i+1,j)+p(i+2,j)-p(i,j+1)-2*p(i+1,j+1)+p(i+2,j+1)-p(i,j+2)-p(i+1,j+2)+
end;
end;
```

```

z=sqrt(s.^2+x.^2);
figure;
subplot(2,2,1)
imshow(b)
title('Original Image');
subplot(2,2,2)
imshow(s)
title('contour horizontale');
subplot(2,2,3)
imshow(x)
title('contour verticale');
subplot(2,2,4)
imshow(z)
title('module');

```

programme 7b

(1.7b)

```

clc;
close all;
clear all;
a=imread('f:\tt.tif'); %l'image d'entrée
b=im2double(a); %Convertit l'image d'intensité I en double précision, en réétalant
les données si nécessaire.
%Si l'image d'entrée est de classe double, l'image de sortie est identique.
[m,n]=size(a); %la taille de la matrice
%approx de base
l(1:m,1:n)=0; %matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
l(i,j)=b(i+1,j)-b(i,j);

```

```

end;
end;
p(1:m,1:n)=0;%matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
p(i,j)=b(i,j+1)-b(i,j);
end;
end;
%canny
sigma=0.5;
g=zeros(3);
for i=-1:1:1
for j=-1:1:1
g(i+2,j+2)=-1*((i+j)/(pi*sigma.^4))*exp(-(i.^2+j.^2)/(2*sigma.^2));
end;
end;
s(1:m,1:n)=0;%matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
s(i,j)=g(1,1)*l(i,j)+g(1+1,1)*l(i+1,j)+g(1+2,1)*l(i+2,j);
s(i,j)=g(1,1+1)*l(i,j+1)+g(1+1,1+1)*l(i+1,j+1)+g(1+2,1+1)*l(i+2,j+1);
s(i,j)=g(1,1+2)*l(i,j+2)+g(1+1,1+2)*l(i+1,j+2)+g(1+2,1+2)*l(i+2,j+2);
end;
end;
x(1:m,1:n)=0;%matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
x(i,j)=g(1,1)*p(i,j)+g(1+1,1)*p(i+1,j)+g(1+2,1)*p(i+2,j);
x(i,j)=g(1,1+1)*p(i,j+1)+g(1+1,1+1)*p(i+1,j+1)+g(1+2,1+1)*p(i+2,j+1);
x(i,j)=g(1,1+2)*p(i,j+2)+g(1+1,1+2)*p(i+1,j+2)+g(1+2,1+2)*p(i+2,j+2);

```

```
end;
end;
z=sqrt(s.^2+x.^2);
figure;
subplot(2,2,1)
imshow(b)
title('Original Image');
subplot(2,2,2)
imshow(s)
title('contour horizontale');
subplot(2,2,3)
imshow(x)
title('contour verticale');
subplot(2,2,4)
imshow(z)
title('module, sigma=0.5');
```

programme 8

(1.8)

```
clc;
clear all;
a=imread('f:\tt.tif'); %l'image d'entrée
b=im2double(a); %Convertit l'image d'intensité I en double précision, en réétalant
les données si nécessaire.
%Si l'image d'entrée est de classe double, l'image de sortie est identique.
[m,n]=size(a); %la taille de la matrice
%laplacien masque1:
L(1:m,1:n)=0 %matrice zéro d'ordr [m n]
for i=1:m-2;
for j=1:n-2;
```

```

L(i,j)=0+0+0+1*b(i+1,j)-2*b(i+1,j+1)+1*b(i+1,j+2)+0+0+0;
end;
end;
%laplacien masque2:
M(1:m,1:n)=0 %matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
M(i,j)=0+1*b(i,j+1)+0+0-2*b(i+1,j+1)+0+0+1*b(i+2,j+1)+0;
end;
end;
%laplacien masque 3:
N(1:m,1:n)=0 %matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
N(i,j)=0+1*b(i,j+1)+0+1*b(i+1,j)-4*b(i+1,j+1)+1*b(i+1,j+2)+0+1*b(i+2,j+1)+0;
end;
end;
%laplacien masque 4:
E(1:m,1:n)=0 %matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
E(i,j)=1*b(i,j)+1*b(i,j+1)+1*b(i,j+2)+1*b(i+1,j)-8*b(i+1,j+1)+1*b(i+1,j+2)+1*b(i+2,j)
end;
end;
%laplacien masque 5:
F(1:m,1:n)=0 %matrice zéro d'ordre [m n]
for i=1:m-2;
for j=1:n-2;
F(i,j)=1*b(i,j)-2*b(i,j+1)+1*b(i,j+2)-2*b(i+1,j)+4*b(i+1,j+1)-2*b(i+1,j+2)+1*b(i+2,j)
end;

```

```
end;
%-----affichage-----
figure;
subplot(2,3,1)
imshow(b)
title('Original Image');
subplot(2,3,2)
imshow(L)
title('masque 1');
subplot(2,3,3)
imshow(M)
title('masque 2');
subplot(2,3,4)
imshow(N)
title('masque 3');
subplot(2,3,5)
imshow(E)
title('masque 4');
subplot(2,3,6)
imshow(F)
title('masque 5');
s=sqrt(L.^2+M.^2);
for i=1:m-2;
for j=1:n-2;
if (s(i,j)<0.2)
s(i,j)=0;
else s(i,j)=1;
end
end;
end;
```

```
figure;
imshow(s)
title('seuillage=0.2');
```

programme 9

(1.9)

```
clc;
close all;
clear all;
a=imread('f:\tt.tif'); %l'image d'entrée
b=im2double(a); %Convertit l'image d'intensité I en double précision, en rééchantillonnant
les données si nécessaire.
%Si l'image d'entrée est de classe double, l'image de sortie est identique.
[m,n]=size(a); %la taille de la matrice
L(1:m,1:n)=0; %matrice zéro d'ordre [m n]
zer=zeros(3);
e=0.6;
g=zer;
for i=-1:1:1
for j=-1:1:1
g(i+2,j+2)=(-1)/(pi*e.^4)*(1-(i.^2+j.^2)/(2*e.^2))*exp(-(i.^2+j.^2)/(2*e.^2));
end
end
for i=1:m-2;
for j=1:n-2;
L(i,j)=g(1,1)*b(i,j)+g(1,2)*b(i+1,j)+g(1,3)*b(i+2,j)+g(2,1)*b(i,j+1)+g(2,2)*b(i+1,j+1)+g(2,3)*b(i+2,j+1);
end
end
figure;
subplot(2,2,1)
```

```

imshow(b)
title('Original Image');
subplot(2,2,2)
imshow(L)
title('sigma=0.6/masque d'ordre 3');
zer=zeros(3);
e=0.6;
g=zer;
for i=-10:1:10
for j=-10:1:10
g(i+11,j+11)=-(-1)/(pi*e.^4)*(1-(i.^2+j.^2)/(2*e.^2))*exp(-(i.^2+j.^2)/(2*e.^2));
end
end
subplot(2,2,[3 4])
mesh(g);figure(gcf);
axis([-1 20 -1 20 -2 3])

```

programme 10

(1.10)

```

clc;
close all;
clear all;
a=imread('f:\tt.tif'); %l'image d'entrée
b=im2double(a); %Convertit l'image d'intensité I en double précision, en réétalant
les données si nécessaire.
%Si l'image d'entrée est de classe double, l'image de sortie est identique.
[m,n]=size(a); %la taille de la matrice
L(1:m,1:n)=0; %matrice zéro d'ordre [m n]
figure;
subplot(3,4,1)

```

```

imshow(b)
title('Original Image');
for v=0.1:0.1:1
L(1:m,1:n)=0; %matrice zéro d'ordre [m n]
for i=1:m-5;
for j=1:n-5;
L(i,j)=(v.^2-v)/2*b(i,j+2)-v*b(i+1,j+2)+v*b(i+3,j+2)+(v.^2-v)/2*b(i+4,j+2);
end;
end;
subplot(3,4,(v+0.1)*10)
imshow(L)
title(['D.H v=',num2str(v),'.'])
end

```

programme 11

(1.11)

```

clc;
close all;
clear all;
a=imread('f:\tt.tif'); %l'image d'entrée
b=im2double(a); %Convertit l'image d'intensité I en double précision, en rééchantillonnant
les données si nécessaire.
%Si l'image d'entrée est de classe double, l'image de sortie est identique.
[m,n]=size(a); %la taille de la matrice
L(1:m,1:n)=0; %matrice zéro d'ordre [m n]
figure;
subplot(3,4,1)
imshow(b)
title('Original Image');
for v=0.1:0.1:1

```

```

L(1:m,1:n)=0; %matrice zéro d'ordre [m n]
for i=1:m-5;
for j=1:n-5;
L(i,j)=(v.^2-v)/2*b(i+2,j)-v*b(i+2,j+1)+v*b(i+2,j+3)+(v.^2-v)/2*b(i+2,j+4);
end;
end;
subplot(3,4,(v+0.1)*10)
imshow(L)
title(['D.V v=',num2str(v),'.'])
end

```

programme 12

(1.12)

```

clc;
close all;
clear all;
a=imread('f:\tt.tif'); %l'image d'entrée
b=im2double(a); %Convertit l'image d'intensité I en double précision, en rééchantillonnant
les données si nécessaire.
%Si l'image d'entrée est de classe double, l'image de sortie est identique.
[m,n]=size(a); %la taille de la matrice
L(1:m,1:n)=0; %matrice zéro d'ordre [m n]
figure;
subplot(3,4,1)
imshow(b)
title('Original Image');
for v=0.1:0.1:1
L(1:m,1:n)=0; %matrice zéro d'ordre [m n]
for i=1:m-5;
for j=1:n-5;

```

```

L(i,j)=(v.^2-v)/2*b(i,j+2)-v*b(i+1,j+2)+v*b(i+3,j+2)+(v.^2-v)/2*b(i+4,j+2);
end;
end;
M(1:m,1:n)=0; %matrice zéro d'ordre [m n]
for i=1:m-5;
for j=1:n-5;
M(i,j)=(v.^2-v)/2*b(i+2,j)-v*b(i+2,j+1)+v*b(i+2,j+3)+(v.^2-v)/2*b(i+2,j+4);
end;
end;
K=sqrt(L.^2+M.^2)
subplot(3,4,(v+0.1)*10)
imshow(K)
title(['norme v=',num2str(v),'.'])
end

```

programme 13

(1.13)

```

clc;
close all;
clear all;
a=imread('f:\tt.tif'); %l'image d'entrée
b=im2double(a); %Convertit l'image d'intensité I en double précision, en rééchantillonnant
les données si nécessaire.
%Si l'image d'entrée est de classe double, l'image de sortie est identique.
[m,n]=size(a); %la taille de la matrice
L(1:m,1:n)=0; %matrice zéro d'ordre [m n]
% 1<v<2
%sigma=1,,,v=1.4
for i=1:m-2;
for j=1:n-2;

```

```

L(i,j)=1*b(i,j)+1*b(i+1,j+1)-1*b(i+1,j)-1*b(i+1,j+1);
end;
end;
m=b;
b=L;
for i=1:m-2;
for j=1:n-2;
L(i,j)=0.0406*b(i,j)+0.0669*b(i+1,j)+0.0406*b(i+2,j);
L(i,j)=L(i,j)+0.0183*b(i,j+1)+0.0301*b(i+1,j+1)+0.0183*b(i+2,j+1);
L(i,j)=L(i,j) +0.0595*b(i,j+2)+0.0980*b(i+1,j+2)+0.0980*b(i+2,j+2);
end
end
figure;
subplot(1,2,1)
imshow(m)
title('Original Image');
subplot(1,2,2)
imshow(L)
title('sigma=1/v=1.4');

```

programme 14

(1.14)

```

close all;
clear all;
a=imread('f:\mmmm.jpg'); %l'image d'entrée
b=im2double(a); %Convertit l'image d'intensité I en double précision, en réétalant
les données si nécessaire.

%Si l'image d'entrée est de classe double, l'image de sortie est identique.
[m,n]=size(a); %la taille de la matrice
L(1:m,1:n)=0; %matrice zéro d'ordre [m n]

```

```

% 1<v<2
%sigma=1,,,v=1.4
sigma =1;
figure;
subplot(4,4,1)
imshow(b)
title('originale');
for v=1:0.1:2.0
k=zeros(3);
for i=-1:1;
for j=-1:1;
g(i+2,j+2)=((1/(2*pi*sigma.^2))*exp(-(i.^2+j.^2)/(2*sigma.^2)));
k(i+2,j+2)=g(i+2,j+2)*(2+(-v)*(exp((2*i+1)/(2*sigma.^2)))+((-v)*(-v+1)/2)*(exp((4*i+4
end
end
for i=1:m-2;
for j=1:n-2;
L(i,j)=k(1,1)*b(i,j)+k(2,1)*b(i+1,j)+k(3,1)*b(i+2,j)+k(1,2)*b(i,j+1)+k(2,2)*b(i+1,j+1
end
end
for i=1:m-2;
for j=1:n-2;
L(i,j)=L(i+1,j)-2*L(i,j)+L(i,j+1);
end
end
subplot(4,4,(v-0.8)*10)
imshow(L)
title([' v=',num2str(v),]);
k
end

```

programme 15

(1.15)

```
-----  
clc;  
close all;  
clear all;  
a=imread('f:\tt.tif'); %l'image d'entrée  
b=im2double(a); %Convertit l'image d'intensité I en double précision, en rééchant  
les données si nécessaire.  
%Si l'image d'entrée est de classe double, l'image de sortie est identique.  
[m,n]=size(a); %la taille de la matrice  
L(1:m,1:n)=0; %matrice zéro d'ordre [m n]  
figure;  
subplot(3,4,1)  
imshow(b)  
title('Original Image');  
for v=1:0.1:2  
L(1:m,1:n)=0; %matrice zéro d'ordre [m n]  
for i=1:m-5;  
for j=1:n-5;  
L(i,j)=((-v)*(-v+1))/2*b(i,j)+0+0+(-v)*b(i+1,j)+b(i+2,j);  
end;  
end;  
subplot(3,4,(v-0.8)*10)  
imshow(L)  
title(['D.H v=',num2str(v),'.'])  
end  
-----
```

programme 16

(1.16)

```
clc;
close all;
clear all;
a=imread('f:\tt.tif'); %l'image d'entrée
b=im2double(a); %Convertit l'image d'intensité I en double précision, en rééchantillonnant
les données si nécessaire.
%Si l'image d'entrée est de classe double, l'image de sortie est identique.
[m,n]=size(a); %la taille de la matrice
L(1:m,1:n)=0; %matrice zéro d'ordre [m n]
figure;
subplot(3,4,1)
imshow(b)
title('Original Image');
for v=1:0.1:2
L(1:m,1:n)=0; %matrice zéro d'ordre [m n]
for i=1:m-5;
for j=1:n-5;
L(i,j)=((-v)*(-v+1))/2*b(i,j)+0+0+(-v)*b(i+1,j)+b(i+2,j);
end;
end;
subplot(3,4,(v-0.8)*10)
imshow(L)
title(['D.V v=',num2str(v),'.'])
end
```

programme 17

(1.17)

```
clc;
close all;
clear all;
```

```
a=imread('f:\tt.tif'); %l'image d'entrée
b=im2double(a); %Convertit l'image d'intensité I en double précision, en rééchantillonnant
les données si nécessaire.
%Si l'image d'entrée est de classe double, l'image de sortie est identique.
[m,n]=size(a); %la taille de la matrice
L(1:m,1:n)=0; %matrice zéro d'ordre [m n]
figure;
subplot(3,4,1)
imshow(b)
title('Original Image');
for v=1:0.1:2
L(1:m,1:n)=0; %matrice zéro d'ordre [m n]
for i=1:m-5;
for j=1:n-5;
L(i,j)=((-v)*(-v+1))/2*b(i,j)+0+0+(-v)*b(i+1,j)+b(i+2,j);
end;
end;
M(1:m,1:n)=0; %matrice zéro d'ordre [m n]
for i=1:m-5;
for j=1:n-5;
M(i,j)=((-v)*(-v+1))/2*b(i,j)+0+0+(-v)*b(i+1,j)+b(i+2,j);
end;
end;
K=sqrt(L.^2+M.^2);
subplot(3,4,(v-0.8)*10)
imshow(K)
title(['norme v=',num2str(v),'.'])
end
```

Bibliographie

- [1] A Novel Edge Detection Operator Based On Fractional Gaussian Differential.HAN Qirui, LIU Ke.
- [2] A. Sparavigna , Dipole Vectors in Images Processing, Computer Vision and Pattern Recognition (cs.CV), arXiv:0902.4663v1, 2009.
- [3] A. Sparavigna, Dipole and Quadrupole Moments in Image Processing, Computer Vision.
- [4] A.A. Kilbas, H.M. Srivastava and J.J. Trujillo, Theory and applications of fractional differential equations,Amsterdam, Elsevier, 2006.
- [5] B. Mathieu, P. Melchior, A. Oustaloup, Ch. Ceyral, Fractional differentiation for edge detections, Signal Processing, 83, pp.2421-2432, 2003.
- [6] Chunprng Wang and Jingxue, Shrinking self-similar solution of a nonlinear diffusion equation with nondivergence form, J. Math. Anal. Appl. 289 (2004), 387-404.
- [7] D. Cafagna, Fractional calculus: a methemathical toll from the past for present engineering, IEEE Indutrial Electronic Magazine, Summer 2007. pp.35-40.
- [8] G.I Barenblatt, On self-similar motoins of compressible fluids in porous media , Prikl.Mat.Mekh, 16 (1952) 679-698 (in Russian).
- [9] http://www.ummt0.dz/IMG/pdf/Memoire_.pdf.
- [10] Introduction au traitement d'images - simulation sous matlab19 octobre 2001de Gilles Burel.

-
- [11] Introduction au traitement mathématique des images - méthodes déterministes Broché – 24 février 2015.
- [12] Le traitement des images/tome 2/ I. Bloch, Y. Gousseau, H. Maitre, D. Matignon, B. Pesquet-Popescu, F. Schmitt, M. Sigelle, F. Tupin/version 5.0/8 decembre 2005.
- [13] M. Kleinz and T.J. Osler, A Child's Garden of Fractional Derivatives, The College Mathematics Journal,31(2), 2000, pp. 82–88.
- [14] M.D. Ortigueira and J.A. Tenreiro Machado, Fractional signal processing and applications, Signal processing, 83(2003) 2285-2286.
- [15] Marr, D.; Hildreth, E. (29 Feb 1980), "Theory of Edge Detection", Proceedings of the Royal Society of London. Series B, Biological Sciences, 207 (1167): 187–217, doi:10.1098/rspb.1980.0020
- [16] The Construction of Fractional Differential Gradient Operator
- [17] Traitement de leimage et de la video Broché – 4 mars 2010.
- [18] Traitement d'Images: Lissage de l'image, Détection de contours, Morphologie mathématique, Retouche d'image, Trucage photographique, Photomontage, Imagerie médicale9 novembre 2010 de Frederic P. Miller et Agnes F. Vandome.
- [19] Tseng C.C. Design of fractional order digital FIR differentiator [J]. Signal Processing Letters, 8 (3):77-79, 2001.
- [20] Tseng C.C. Improved Design of Digital Fractional-Order Differentiators Using Fractional Sample Delay [J]. IEEE Trans. On Circuits and Systems I: Regular Papers, 2006, 5(1):193-203.

Remerciements

*Nous remercions, en premier lieu, Mon Dieu qui nous a
.donné la force de rédiger ce modeste travail*

*Et nous remercions les membres du jury, qui ont accepté
.d'évaluer mon travail de mémoire*

*Et nous remercions le Maître B.Nouredine directeur de mon
mémoire, pour sa disponibilité et ses conseils judicieux tout
au long de ce travail*

*Et nous remercions tous les professeurs du département de
Mathématiques, sans oublier aussi mes collègues et amies,
ainsi tous ceux qui ont participé de loin ou de près à
l'élaboration de ce mémoire.*

تَسْمِيحًا بِحَمْدِ اللَّهِ

"الْحَمْدُ لِلَّهِ الَّذِي لَهُ مَا فِي السَّمَاوَاتِ وَمَا
فِي الْأَرْضِ وَلَهُ الْحَمْدُ فِي الْآخِرَةِ وَهُوَ
الْحَكِيمُ الْخَبِيرُ" [سبأ: 01]

المخلص:

الهدف من هذه الدراسة هو تبيان أن الحساب التفاضلي والتكامل الكسري له أهمية كبيرة في تحسين الصورة والكشف عن ملامحها، وذلك بتطبيقنا لعدة طرق وأكثرها إستعمالا على الصور والمقارنة بين النتائج لكل طريقة.

الكلمات المفتاحية: صورة، مشتق، مشتق جزئي، مصفوفة، المرشحة ،،،.

ABSTRACT:

The objective of this study is to show that the fractional calculation is of great importance to the improvement of the image, and detecting the contours. Applying several more widely used methods on images. And the comparison between the results of each method.

THE KEY WORKS: Image, derivative, partial derivative, matrix, filter

””

RÉSUMÉ :

L'objectif de cette étude est de montrer que la calcul fractionnaire est d'une grande importance à l'amélioration de l'image, et la detection des contours. En appliquant plusieurs méthodes les plus largement utilisé sur les images. Et la comparaison entre les résultats de chaque méthode.

LES MOTS CLÉS : Image, dérivé, dérivée partielle, matrice, filtre,,.,.