

# Technique de détection de contours flous et bruités

M. Sarifuddin, Rokia Missaoui et Hayet Laggoune

*Laboratoire LARIM, Département d'informatique et d'ingénierie,  
Université du Québec en Outaouais  
C.P. 1250, Succ. B Gatineau (Qc)  
Canada, J8X 3X7*

[m.sarifuddin@uqo.ca](mailto:m.sarifuddin@uqo.ca),  
[rokoa.missaoui@uqo.ca](mailto:rokoa.missaoui@uqo.ca),  
[laggoune@labunix.uqam.ca](mailto:laggoune@labunix.uqam.ca).

**Résumé** La détection des contours d'une image numérique est un traitement très délicat, car une image peut comporter un ensemble de zones avec des caractéristiques différentes en matière de bruit, flou, homogénéité, niveaux de détails, etc. Il existe un grand nombre de détecteurs et chaque détecteur a des performances différentes suivant les caractéristiques de l'image. Certains détecteurs sont très efficaces pour les images nettes mais sensibles au bruit, alors que d'autres détecteurs sont très adaptés aussi bien aux images nettes qu'aux images bruitées (selon le choix des paramètres du filtre) mais sensibles au flou.

Dans cet article, nous proposons deux nouveaux filtres de lissage et de détection de contours dont le développement est basé sur un modèle de contour flou. Nous prenons une approche particulière pour que ces filtres soient identiques à la fonction Gaussienne et sa première dérivée respectivement et implantables sous forme récursive. Ces filtres présentent des avantages pour l'analyse de divers types de contours (nets, bruités et flous). Une analyse expérimentale montre que les résultats obtenus par ces filtres donnent nettement une meilleure qualité de détection de contours par rapport aux résultats des filtres existants. Ils offrent également une meilleure détection et une bonne localisation des contours.

Nous proposons également des filtres circulaires bidimensionnels qui peuvent être utilisés pour la détection des contours directionnels (horizontaux, verticaux, diagonaux, coin et arc circulaire) d'une image en niveaux de gris et image couleur.

**Mots clés:** contour flou, filtres, détection de contours.

## 1 Introduction

La qualité ou la netteté d'une image lors de sa prise dépend de l'ouverture du diaphragme, de la vitesse d'obturation de la caméra et de l'effet de lumière dans la scène. Lorsque la vitesse d'obturation est lente, les objets en mouvement dans la scène seront flous et le reste de l'image sera net. En outre, la clarté de la scène peut entraîner du bruit dans l'image. Le bruit se manifeste plus précisément dans les zones sombres de l'image. On distingue ici deux types de bruit : le bruit de chrominance (les petites tâches de couleurs disparates dans les zones sombres de l'image) et le bruit de luminance (les petites tâches blanches ou noires qui donnent à l'image une apparence granuleuse dans les zones sombres ou certaines plages colorées).

Ainsi, les images réelles prises par une caméra photo/vidéo peuvent contenir naturellement des zones nettes, floues et/ou bruitées. Les techniques d'analyse de l'image doivent alors tenir compte de ces caractéristiques.

L'objectif de cet article est de proposer deux filtres : l'un pour le lissage et l'autre pour la détection de contours. Le premier a pour but d'atténuer ou d'éliminer les bruits alors que le second est destiné à détecter les contours d'une image. Étant donné que l'application du lissage sur une image donne souvent un effet de contours flous, le filtre de détection doit tenir compte de cet état de fait. Pour cette raison, le développement des deux filtres proposés se base sur un modèle de contour flou.

Le reste de cet article est organisé comme suit. Les modèles de contours et le développement des

filtres basés sur un modèle de contour flou sont présentés dans la section 2. La section 3 décrit l'implantation des filtres selon deux formes distinctes : non récursive et récursive. L'analyse expérimentale et la conclusion sont décrites dans les sections 4 et 5 respectivement.

## 2 Développement de filtres

Nous présentons dans cette section de nouveaux filtres qui se basent sur un modèle de contour flou. La première partie de cette section présente les modèles de contours tandis que la seconde traite des filtres proposés.

### 2.1 Modèle de contours flous

La détection de contours au sein d'une image est une caractéristique très importante dans le processus de reconnaissance d'objets et de la recherche d'images basée sur le contenu. Le développement d'un détecteur de contours est souvent basé sur une caractéristique spécifique de l'image. A titre d'exemple, les opérateurs du premier ordre (le gradient de Roberts, les masques de Prewitt et de Sobel) et l'opérateur du second ordre (le Laplacien) sont bien adaptés aux images nettes (sans bruit ni flou). Les filtres de Canny (Canny, 1986) et Deriche (Deriche, 1990) sont développés en se basant sur un modèle de contour de type marche (échelon), alors que les filtres de Bourennane (Bourennane et al., 2002) et de Laggoune (Laggoune, 1996), (Gouton et al., 2000) utilisent le contour de type rampe et pic respectivement. Chacun de ces chercheurs propose une paire de filtres : passe bas et passe bande. Le filtre passe bas permet de lisser l'image (et donc atténuer le bruit) alors que le filtre passe bande permet de détecter des contours. Les détecteurs de Canny, Deriche, Bourennane et Laggoune sont bien adaptés aux images nettes et bruitées par un réglage d'un ou plusieurs paramètres.

Les figures 1-a à 1-c représentent les modèles de contours échelon, pic et impulsion respectivement, tandis que les figures 2-a à 2-c montrent les modèles de contours flous. La figure 2-a est un contour échelon rendu flou et correspond à un modèle de contour rampe. Les figures 2-b et 2-c représentent respectivement les contours pic et impulsion rendus flous. Nous pouvons constater que le schéma 2-b est un bon modèle de contour flou car il couvre les deux autres modèles de contours flous. Nous pouvons ensuite modéliser ce contour par une fonction  $C(x)$  comme montré ci-dessous (voir équation [1]), où  $k$  et  $\lambda$  sont deux constantes de normalisation,  $\beta$  représente le paramètre de flou et  $\alpha$  correspond au paramètre d'échelle qui est lié au bruit. La figure 3 illustre l'allure de cette fonction pour  $\alpha=1$ ,  $\lambda=1$ ,  $k=1$  et  $\beta=0,8$ .

$$C(x) = k \cdot (\lambda + \sin(\beta \cdot x)) e^{-\alpha \cdot |x|} \quad (1)$$

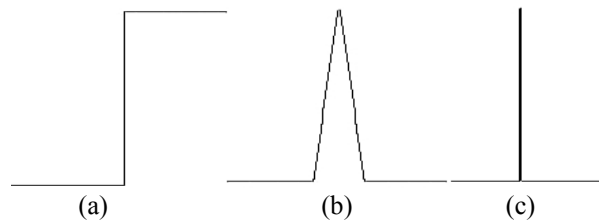


Figure 1. (a) contour marche, (b) contour pic et (c) contour impulsion.

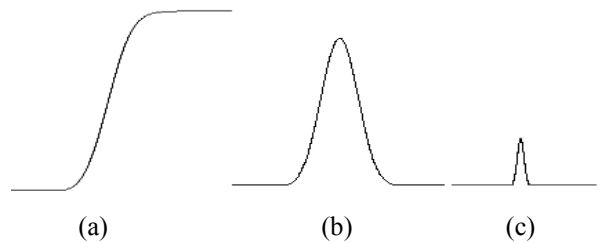


Figure 2. (a) contour marche rendu flou (contour rampe), (b) contour pic rendu flou et (c) contour impulsion rendu flou.

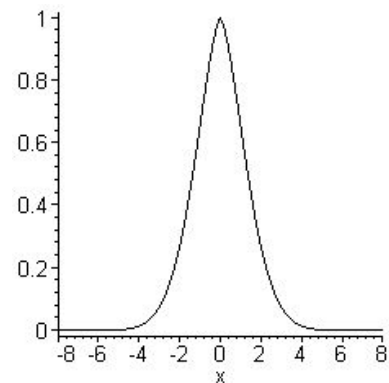


Figure 3. Modèle de contour pic et flou.

### 2.2 Filtres de lissage et de détection de contours

Soit  $C(x)$  un modèle de contour dans une image (modèle de la variation d'intensité de contours de type pic tout au long de l'axe  $x$ ). Le contour de l'image peut être calculé par le Laplacien (détecteur du second ordre) de cette fonction  $C(x)$  tel que décrit par la formule ci-après :

$$f(x) = \frac{\partial^2 C(x)}{\partial x^2} \quad (2)$$

où  $f(x)$  correspond à une fonction de détection de contours et peut être calculée comme suit en remplaçant  $C(x)$  par l'équation [1] :

$$f(x) = \frac{\partial^2 (k(\lambda + \sin(\beta \cdot x)) e^{-\alpha \cdot x})}{\partial x^2} \quad (3)$$

$$f(x) = k \cdot e^{-\alpha \cdot x} (\lambda \cdot \alpha^2 - 2\alpha \beta \cdot \cos(\beta \cdot x) + (\alpha^2 - \beta^2) \sin(\beta \cdot x)) \quad (4)$$

Afin que  $f(x)$  soit une fonction impaire et convergente, c'est-à-dire  $f(x) = -f(-x)$ ,  $f(0) = 0$ ,  $f(-\infty) = 0$  et  $f(\infty) = 0$ , on peut calculer la valeur  $\lambda$  à la position  $x = 0$  par :

$$f(0) = 0 = (\lambda \alpha^2 - 2\alpha\beta) \rightarrow \lambda = 2\beta / \alpha .$$

Pour garantir que la fonction soit impaire, la formule [4] peut être généralisée par :

$$f(x) = \text{sgn}(-x)K_1.e^{-\alpha|x|}\left(1 - \cos(\beta|x|) + \left(\frac{\alpha^2 - \beta^2}{2\alpha\beta}\right)\sin(\beta|x|)\right) \quad (5)$$

où  $K_1 = k.2\alpha\beta$  et  $\text{sgn}(-x)$  est négatif quand  $x$  positif et positif quand  $x$  négatif. Pour éviter l'oscillation le long de l'axe  $x$ , le choix de la valeur  $\beta$  doit être limité par  $0 < \beta \leq \alpha.\sqrt{\frac{\pi}{4}}$ . Si  $\beta$  tend vers zéro,  $f(x)$  peut être utilisée pour détecter les contours dans les images peu ou non floues et si  $\beta$  est grand,  $f(x)$  est bien adaptée pour détecter les contours dans les images floues.

Ce filtre a deux cas particuliers : le premier est qu'il est identique au filtre de Deriche lorsque  $\beta \ll \alpha$  (c'est-à-dire lorsque  $\beta$  tend vers zéro), alors que le second est qu'il est identique à la première dérivée de la Gaussienne lorsque  $\beta$  vaut environ  $\alpha/1,125$  et la relation entre  $\alpha$  et  $\sigma$  est donnée par  $\alpha.\sigma = 1,59139$ . Ce dernier cas peut simplifier la formule  $f(x)$  comme illustrée par l'équation [6] où  $K_1$  est une nouvelle constante de normalisation.

$$f(x) = K_1 \left(1 - \cos\left(\frac{\alpha|x|}{1,125}\right) + 0.118056 \sin\left(\frac{\alpha|x|}{1,125}\right)\right) e^{-\alpha|x|} \quad (6)$$

Dans le cas de l'analyse du signal  $S(x)$  et d'images  $I(x,y)$  bruités, il est nécessaire d'atténuer ou de lisser d'abord le bruit et ensuite détecter les contours. Donc la détermination du filtre de lissage peut être formulée comme suit :

$$\text{Signal lissé} : S'(x) = h(x) * S(x) \quad (7)$$

$$\text{Contour} : f(x) * S(x) = \frac{d(S'(x))}{dx} = \frac{dh(x)}{dx} * S(x) \quad (8)$$

où  $f(x)$  correspond à la première dérivée d'une fonction de lissage. Ainsi, la fonction de lissage peut être calculée comme ci-après où  $K$  est une constante de normalisation :

$$h(x) = \int f(x) dx$$

$$h(x) = K \left(1 - 0.4999998 \cos\left(\frac{\alpha|x|}{1,125}\right) + 0.56250025 \sin\left(\frac{\alpha|x|}{1,125}\right)\right) e^{-\alpha|x|} \quad (9)$$

Nous avons finalement les deux fonctions  $h(x)$  et  $f(x)$  qui se comportent comme la fonction Gaussienne  $g(x)$  et sa première dérivée  $g'(x)$  respectivement (équations [12] et [13]). Les figures 4 et 5 représentent respectivement la courbe de  $h(x)$  (et de  $g(x)$ ), et la courbe de  $f(x)$  (et de  $g'(x)$ ), pour  $\alpha = 0.795695$  et  $\sigma=2$  avec  $K = 0.3992$  et  $K_1 = 0.3085$ . Les courbes en ligne solide représentent les filtres  $g(x)$  et  $g'(x)$ , et celles en ligne pointillée sont les filtres

proposés dans cet article. Nous constatons que les courbes  $h(x)$  et  $g(x)$  sont quasiment identiques, Il en est de même pour  $f(x)$  et  $g'(x)$ . Ainsi donc, les filtres  $h(x)$  et  $f(x)$  peuvent être considérés respectivement comme l'approximation de  $g(x)$  et  $g'(x)$  avec une erreur maximum de l'ordre de  $10^{-3}$ .

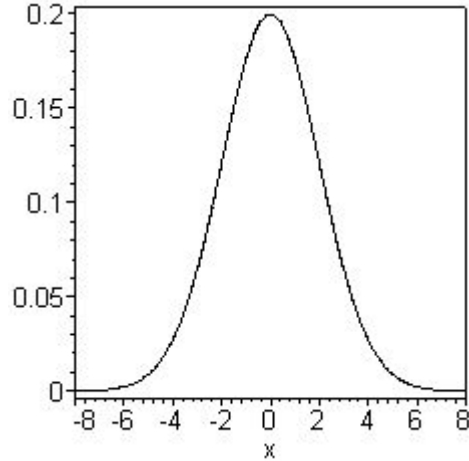


Figure 4. Courbe unique de  $h(x)$  (pour  $\alpha = 0.795695$  et  $K = 0.3992$ ) et de  $g(x)$  (pour  $\sigma = 2$ ).

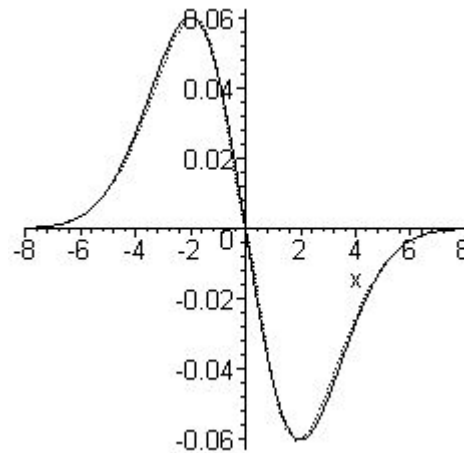


Figure 5. Courbe unique de  $f(x)$  (pour  $\alpha = 0.795695$  et  $K_1 = 0.3085$ ) et de  $g'(x)$  (pour  $\sigma = 2$ ).

### 3 Implantation non récursive et récursive de filtres

Beaucoup de chercheurs travaillant sur l'analyse de contours multi-échelles pour des images en niveaux de gris évitent d'utiliser des filtres à réponse impulsionnelle finie (RIF) ou un masque de convolution de taille variable, car cette approche donne un temps de calcul très élevé et entraîne un nombre variable d'opérations (multiplication et addition) selon le choix du paramètre d'échelle. Pour cette raison, ils s'intéressent plus aux approches de filtres à réponse impulsionnelle infinie (RII) où la structure d'implantation est récursive. Cette dernière réduit énormément le temps de calcul et le nombre d'opérations indépendamment du choix de la valeur

d'échelle.

Dans cette partie, nous présentons et discutons les intérêts d'implantation des filtres  $h(x)$  et  $f(x)$  en structure RII pour l'analyse des images en niveaux de gris et d'implantation des filtres  $g(x)$  et  $g'(x)$  en structure RIF pour l'analyse des images en niveaux de gris et couleur.

**3.1 Implantation non récursive 2-D**

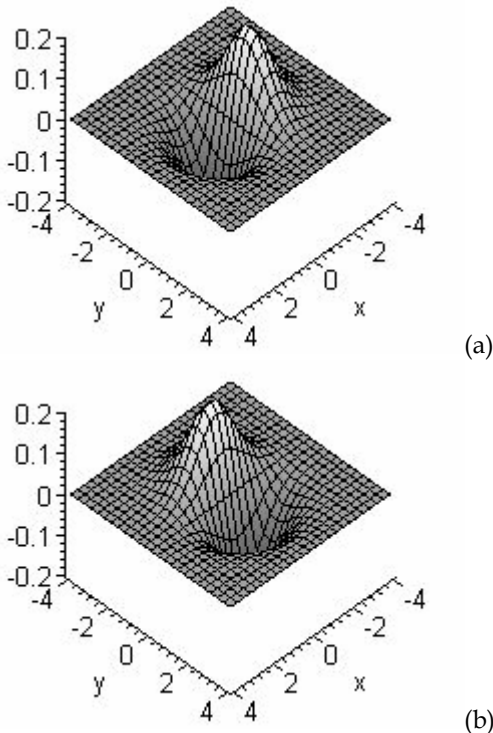
L'application du filtre de lissage sur une image permet d'atténuer les bruits mais rend certains contours flous. Donc l'utilisation d'un détecteur de contour de type flou est très importante, ce qui est le cas de nos filtres, du filtre Gaussien et de sa première dérivée. Nous avons vu dans la section 2 que nos filtres  $h(x)$  et  $f(x)$  sont quasiment identiques aux filtres de la Gaussienne  $g(x)$  et  $g'(x)$  respectivement. Aussi, pour l'implantation non récursive, nous utilisons les filtres Gaussiens à cause de la simplicité de leur formulation.

Afin de détecter les contours dans une image 2-D, on définit deux détecteurs bidimensionnels de la façon suivante :

- avec les filtres proposés :

$$G_{s1}(x, y) = h(x).f(y) \tag{10}$$

$$G_{s2}(x, y) = f(x).h(y) \tag{11}$$



**Figure 6.** Les courbes de détecteurs proposés en 2-D, (a) le détecteur de contours verticaux  $G_{s2}$  et (b) le détecteur de contours horizontaux  $G_{s1}$ .

- avec les filtres Gaussiens :

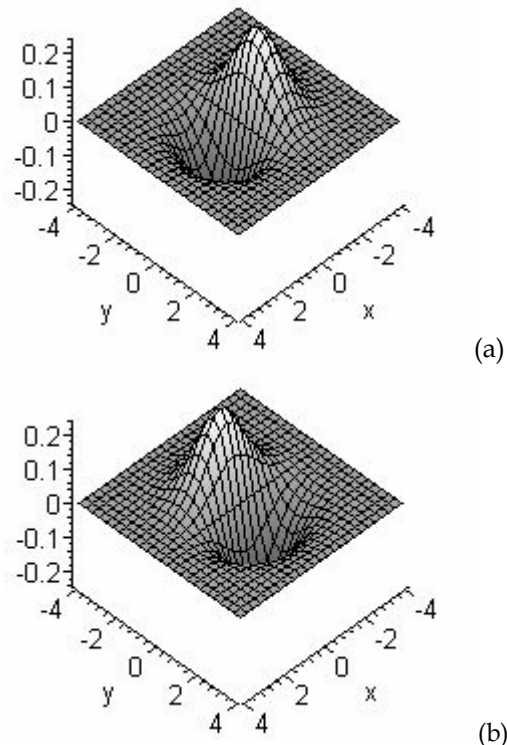
$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{12}$$

$$g'(x) = \frac{-x}{\sigma^3\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{13}$$

$$G_1(x, y) = g(x).g'(y) = C_1.y.\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{14}$$

$$G_2(x, y) = g'(x).g(y) = C_1.x.\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{15}$$

où  $x$  et  $y$  représentent les coordonnées d'un pixel de l'image,  $G_{s1}$  et  $G_1$  représentent le lissage selon  $x$  suivi par la détection de contours dans la direction  $y$  (cas des contours horizontaux),  $G_{s2}$  et  $G_2$  réalisent la détection de contours selon  $x$  suivie par le lissage selon  $y$  (cas des contours verticaux). Les figures 6 et 7 illustrent les courbes 2-D des détecteurs proposés  $G_{s1}$  et  $G_{s2}$  et ceux du Gaussien  $G_1$  et  $G_2$  respectivement. Elles illustrent clairement la correspondance quasi-parfaite entre  $G_{s1}$  et  $G_1$  d'une part, et celle entre  $G_{s2}$  et  $G_2$  d'autre part.



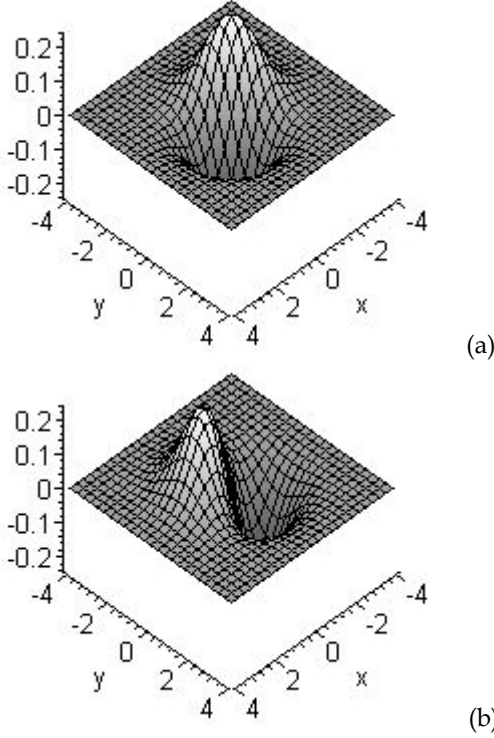
**Figure 7.** Les courbes de détecteurs Gaussiens en 2-D, (a) le détecteur de contours verticaux, (b) le détecteur de contours horizontaux.

Pour l'implantation non récursive, nous utilisons les filtres gaussiens en raison de la simplicité de leur formulation en 2-D et de la déduction des filtres circulaires. Ces derniers donnent l'avantage pour la détection des contours sous forme de ligne directionnelle, coin ou de forme circulaire. Pour ce faire, il suffit d'introduire une rotation dans les coordonnées des filtres  $G_1$  et  $G_2$ , ce qui nous donne deux filtres circulaires comme illustrés par les équations [16] et [17].

$$G_{1,\theta}(x, y) = C_1(-x \cdot \sin \theta + y \cdot \cos \theta) e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (16)$$

$$G_{2,\theta}(x, y) = C_1(x \cdot \sin \theta + y \cdot \cos \theta) e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (17)$$

où  $\theta$  représente un angle entre 0 et 90 degrés, et  $C_1$  est une constante de normalisation des filtres.



**Figure 8.** Les courbes de détecteurs Gaussiens en 2-D, (a) le détecteur de contours diagonaux  $45^\circ$  et (b) le détecteur de contours diagonaux  $135^\circ$ .

Ainsi, nous avons  $G_1$  comme masque de détection de contours en direction  $\theta$  et  $G_2$  comme masque de détection de contours en direction  $\theta+90^\circ$ . La largeur du masque dépend du choix de la valeur d'échelle  $\sigma$ . Les masques ci-dessus représentent un exemple des deux paires de masques  $G_{1,\theta}$  et  $G_{2,\theta}$  pour  $\sigma=1$  et  $\theta=0^\circ$  et pour  $\sigma=1$  et  $\theta=45^\circ$ . Dans ce cas, certains éléments du masque sont tronqués et remis à zéro. Mais dans les expérimentations, la troncature des éléments du masque se fait avec une précision de  $10^{-4}$  (Missaoui et al., 2005).

$$G_{1,\theta=0^\circ} = \begin{bmatrix} 0.0168 & 0.0754 & 0.1242 & 0.0754 & 0.0168 \\ 0.0377 & 0.1689 & 0.2784 & 0.1689 & 0.0377 \\ 0 & 0 & 0 & 0 & 0 \\ -0.0377 & -0.1689 & -0.2784 & -0.1689 & -0.0377 \\ -0.0168 & -0.0754 & -0.1242 & -0.0754 & -0.0168 \end{bmatrix}$$

$$G_{2,\theta=90^\circ} = \begin{bmatrix} -0.0168 & -0.0377 & 0 & 0.0377 & 0.0168 \\ -0.0754 & -0.1689 & 0 & 0.1689 & 0.0754 \\ -0.1242 & -0.2784 & 0 & 0.2784 & 0.1242 \\ -0.0754 & -0.1689 & 0 & 0.1689 & 0.0754 \\ -0.0168 & -0.0377 & 0 & 0.0377 & 0.0168 \end{bmatrix}$$

$$G_{1,\theta=45^\circ} = \begin{bmatrix} 0.1242 & 0.0754 & 0.0377 & 0.0168 & 0 \\ 0.0754 & 0.2784 & 0.1689 & 0 & -0.0168 \\ 0.0377 & 0.1689 & 0 & -0.1689 & -0.0377 \\ 0.0168 & 0 & -0.1689 & -0.2784 & -0.0754 \\ 0 & -0.0168 & -0.0377 & -0.0754 & -0.1242 \end{bmatrix}$$

$$G_{2,\theta=135^\circ} = \begin{bmatrix} 0 & -0.0168 & -0.0377 & -0.0754 & -0.1242 \\ 0.0168 & 0 & -0.1689 & -0.2784 & -0.0754 \\ 0.0377 & 0.1689 & 0 & -0.1689 & -0.0377 \\ 0.0754 & 0.2784 & 0.1689 & 0 & -0.0168 \\ 0.1242 & 0.0754 & 0.0377 & 0.0168 & 0 \end{bmatrix}$$

Nous pouvons maintenant appliquer les filtres dans toutes les directions souhaitées afin de mieux détecter les contours. La détection de contours  $E(m,n)$  se fait par la convolution de chaque masque avec l'image  $I(m,n)$  de taille  $M \times N$  comme proposé par l'algorithme qui suit :

```

for n = 0 to N - 1 do
  for m = 0 to M - 1 do
     $E_\theta(m,n) = G_{1,\theta} * I(m,n)$ 
     $E_{\theta+90^\circ}(m,n) = G_{2,\theta} * I(m,n)$ 
     $E(m,n) = \sqrt{E_\theta(m,n)^2 + E_{\theta+90^\circ}(m,n)^2}$ 
  end m
end n

```

Nous constatons ici que pour le calcul au niveau de chaque pixel de l'image, cet algorithme a besoin de  $P \cdot (O^2 - O)$  multiplications et  $P \cdot ((O^2 - O) - 1)$  additions, où  $O$  est la largeur/hauteur du masque et  $P$  est le nombre de masques (ici  $P=4$ ). Cela rend l'implantation de l'algorithme coûteuse. Cependant l'algorithme n'utilise que  $P$  lignes d'image de mémoire de stockage pendant le traitement. Celles-ci sont beaucoup moins coûteuses que dans l'algorithme récursif qui a besoin de sept lignes d'image de mémoire plus deux mémoires d'image complètes pour le stockage pendant le traitement (Sariffuddin et al., 1999), (Sariffuddin et al., 2000).

Par ailleurs, cet algorithme non récursif est très intéressant pour la détection de contours d'une image couleur, sans division de l'image en trois images selon les composantes rouge, vert et bleu.

En vue de comparer la couleur de deux pixels, on peut utiliser la distance euclidienne. Dans l'espace de couleur RVB, cette distance s'exprime par :

$$D = \sqrt{R^2 + V^2 + B^2} \quad (18)$$

Les formules de  $R^2$ ,  $V^2$  et  $B^2$  sont données respectivement par :

$$R^2 = (R(m + j, n + i) - R(m - j, n - i))^2,$$

$$V^2 = (V(m + j, n + i) - V(m - j, n - i))^2 \text{ et}$$

$$B^2 = (B(m + j, n + i) - B(m - j, n - i))^2.$$

Les variables  $R(m+j, n+i)$ ,  $V(m+j, n+i)$  et  $B(m+j, n+i)$  représentent respectivement l'intensité de la couleur rouge, vert et bleu d'un pixel de l'image à la position  $m+j$  et  $n+i$ .

Ainsi l'algorithme précédent peut être réécrit comme suit en remplaçant l'opérateur de convolution (notée par '\*') par l'opérateur de multiplication (notée par '.') faisant intervenir la distance  $D$  :

```

for n = 0 to N - 1 do
  for m = 0 to M - 1 do
    for i = 1 to (O - 1) / 2 do
      for j = 1 to (O - 1) / 2 do
         $E_{\theta}(m, n) = G_1(m + j, n + i) \cdot D$ 
         $E_{\theta+90^{\circ}}(m, n) = G_2(m + j, n + i) \cdot D$ 
         $E(m, n) = \sqrt{E_{\theta}(m, n)^2 + E_{\theta+90^{\circ}}(m, n)^2}$ 
      end j      end i
    end m      end n.

```

### 3.2 Implantation récursive 1-D

L'implantation récursive d'un filtre est très importante pour la réduction du temps de calcul et du nombre de multiplications et d'additions dans le traitement en temps réel. Il est clair que la mise sous forme récursive de la fonction Gaussienne est très difficile. Pour remédier à ce problème, certains chercheurs tels que (Deriche, 1993) et (Young et al., 2002) ont procédé par une approximation. Deriche a proposé une fonction d'approximation de la gaussienne qui est implantable sous forme récursive. Cette fonction a une récursivité d'ordre quatre et a besoin de 16 multiplications et 15 additions. Young a fait une approximation de la transformée de Fourier de la Gaussienne avec une troncature. Cette dernière perd de la qualité de détection selon le choix de  $\sigma$ . Lorsque  $\sigma$  est très petit, l'allure de la courbe est très serrée et la troncature ne donne pas un grand effet. Quand  $\sigma$  devient de plus en plus grand, la courbe s'élargit et la troncature produit une détérioration de la qualité de la détection.

Les filtres  $h(x)$  et  $f(x)$  produisent un effet quasiment identique à la gaussienne  $g(x)$  et la première dérivée de  $g(x)$  respectivement. Nous avons montré que les filtres  $h(x)$  et  $f(x)$  peuvent être considérés comme une approximation de la gaussienne et sa première dérivée respectivement. Ainsi, l'implantation sous forme récursive de  $h(x)$  et  $f(x)$  peut être aussi considérée comme l'implantation récursive des filtres gaussiens.

L'application de ces filtres pour l'analyse d'image numérique nécessite la transformation de  $h(x)$  et  $f(x)$  en filtres numériques. La numérisation de ces filtres est assurée par la transformée en Z ci-après.

**Filtre lissage  $h(x)$  :**

$$H(Z) = \sum_{n=-\infty}^{\infty} h(n)Z^{-n} = \sum_{n=-\infty}^{-1} h(n)Z^{-n} + \sum_{n=0}^{\infty} h(n)Z^{-n} \quad (19)$$

avec  $\sum_{n=-\infty}^{\infty} h(n) = 1$ .

La partie causale du filtre de lissage est donnée par :

$$H^+(Z) = \sum_{n=0}^{\infty} h(n)Z^{-n} = K \left( \begin{array}{l} \sum_{n=0}^{\infty} e^{-\alpha|x|} Z^{-n} \\ -0.4999998 \cdot \sum_{n=0}^{\infty} \cos\left(\frac{\alpha|x|}{1.125}\right) e^{-\alpha|x|} Z^{-n} \\ + 0.56450026 \sum_{n=0}^{\infty} \sin\left(\frac{\alpha|x|}{1.125}\right) e^{-\alpha|x|} Z^{-n} \end{array} \right) \quad (20)$$

et sa transformée en Z est donnée par :

$$H^+(Z) = K \cdot \left( \begin{array}{l} \frac{1}{1 - e^{-\alpha} Z^{-1}} \\ - \frac{0.4999998 \left(1 - \cos\left(\frac{\alpha}{1.125}\right) e^{-\alpha} Z^{-1}\right)}{1 - 2 \cos\left(\frac{\alpha}{1.125}\right) e^{-\alpha} Z^{-1} + e^{-2\alpha} Z^{-2}} \\ + \frac{0.56450025 \cdot \sin\left(\frac{\alpha}{1.125}\right) e^{-\alpha} Z^{-1}}{1 - 2 \cos\left(\frac{\alpha}{1.125}\right) e^{-\alpha} Z^{-1} + e^{-2\alpha} Z^{-2}} \end{array} \right) \quad (21)$$

soit

$$H^+(Z) = \frac{a_0 + a_1 \cdot Z^{-1} + a_2 \cdot Z^{-2}}{1 + b_1 \cdot Z^{-1} + b_2 \cdot Z^{-2} + b_3 \cdot Z^{-3}} \cdot K \quad (22)$$

où les coefficients  $a_i$  et  $b_i$  sont donnés par :

$$a_0 = 0.5,$$

$$a_1 = \left( \begin{array}{l} 0.5 + 0.5645 \sin\left(\frac{\alpha}{1.125}\right) \\ -1.5 \cos\left(\frac{\alpha}{1.125}\right) \end{array} \right) e^{-\alpha},$$

$$a_2 = \left( \begin{array}{l} 1 - 0.5645 \sin\left(\frac{\alpha}{1.125}\right) \\ -0.5 \cos\left(\frac{\alpha}{1.125}\right) \end{array} \right) e^{-2\alpha},$$

$$b_1 = -(1 + 2 \cos\left(\frac{\alpha}{1.125}\right)) e^{-\alpha},$$

$$b_2 = (1 + 2 \cos\left(\frac{\alpha}{1.125}\right)) e^{-2\alpha} \quad \text{et} \quad b_3 = -e^{-3\alpha}.$$

La partie anti-causale du filtre de lissage est donnée par :

$$H^-(Z) = \sum_{n=-\infty}^{-1} h(n)Z^{-n} = K \left( \begin{array}{l} \sum_{n=0}^{\infty} e^{-\alpha(n+1)} Z^{(n+1)} \\ -0.4999998 \cdot \sum_{n=0}^{\infty} \cos\left(\frac{\alpha(n+1)}{1.125}\right) e^{-\alpha(n+1)} Z^{(n+1)} \\ + 0.56450025 \sum_{n=0}^{\infty} \sin\left(\frac{\alpha(n+1)}{1.125}\right) e^{-\alpha(n+1)} Z^{(n+1)} \end{array} \right) \quad (23)$$

et sa transformée en Z est donnée par :

$$H^-(Z) = K \left( \begin{array}{l} \frac{e^{-\alpha} Z}{1 - e^{-\alpha} Z} \\ + \frac{0.4999998 \left( \cos\left(\frac{\alpha}{1.125}\right) e^{-\alpha} Z - e^{-2\alpha} Z^2 \right)}{1 - 2 \cos\left(\frac{\alpha}{1.125}\right) e^{-\alpha} Z + e^{-2\alpha} Z^2} \\ + \frac{0.56450025 \cdot \sin\left(\frac{\alpha}{1.125}\right) e^{-\alpha} Z}{1 - 2 \cos\left(\frac{\alpha}{1.125}\right) e^{-\alpha} Z + e^{-2\alpha} Z^2} \end{array} \right) \quad (24)$$

soit 
$$H^-(Z) = \frac{a_3 \cdot Z + a_4 \cdot Z^2 + a_5 \cdot Z^3}{1 + b_1 \cdot Z + b_2 \cdot Z^2 + b_3 \cdot Z^3} K \quad (25)$$

où les coefficients  $a_i$  sont donnés par :

$$a_3 = \left( \begin{array}{l} 1 + 0.5645 \sin\left(\frac{\alpha}{1.125}\right) \\ - 0.5 \cos\left(\frac{\alpha}{1.125}\right) \end{array} \right) \cdot e^{-\alpha},$$

$$a_4 = \left( \begin{array}{l} 0.5 - 0.5645 \sin\left(\frac{\alpha}{1.125}\right) \\ - 1.5 \cos\left(\frac{\alpha}{1.125}\right) \end{array} \right) \cdot e^{-2\alpha} \quad \text{et} \quad a_5 = 0.5 \cdot e^{-3\alpha}.$$

La réponse impulsionnelle  $y(m)$  d'un signal numérique  $S(m)$  correspondant aux filtres (22) et (25) s'écrit sous forme de :

$$y_h(m) = y^+(m) + y^-(m) \quad (26)$$

où  $y^+(m)$ ,  $y^-(m)$  et  $K$  sont donnés par :

$$y^+(m) = K(a_0 \cdot S(m-1) + a_1 \cdot S(m-2) + a_2 \cdot S(m-3) - b_1 \cdot y(m-1) - b_2 \cdot y(m-2) - b_3 \cdot y(m-3))$$

$$y^-(m) = K(a_3 \cdot S(m+1) + a_4 \cdot S(m+2) + a_5 \cdot S(m+3) - b_1 \cdot y(m+1) - b_2 \cdot y(m+2) - b_3 \cdot y(m+3))$$

$$K = \frac{1 + b_1 + b_2 + b_3}{a_0 + a_1 + a_2 + a_3 + a_4 + a_5}.$$

**Filtre détecteur  $f(x)$  :**

La transformée en  $Z$  du filtre détecteur se réalise selon la même démarche que précédemment.

$$F(Z) = \sum_{-\infty}^{\infty} f(n) Z^{-n} = \sum_0^{\infty} f(n) Z^{-n} + \sum_{-\infty}^{-1} f(n) Z^{-n} \quad (27)$$

avec 
$$\sum_{n=-\infty}^0 f(n) = \sum_{n=0}^{\infty} f(n) = 0.$$

La partie causale se calcule comme suit :

$$F^+(Z) = \sum_{n=0}^{\infty} f(n) Z^{-n} = K_1 \left( \begin{array}{l} \sum_{n=0}^{\infty} e^{-\alpha n} Z^{-n} \\ - \sum_{n=0}^{\infty} \cos\left(\frac{\alpha n}{1.125}\right) e^{-\alpha n} Z^{-n} \\ + 0.118056 \sum_{n=0}^{\infty} \sin\left(\frac{\alpha n}{1.125}\right) e^{-\alpha n} Z^{-n} \end{array} \right) \quad (28)$$

et sa transformée en  $Z$  est donnée par :

$$F^+(z) = K_1 \left( \begin{array}{l} \frac{1}{1 - e^{-\alpha} Z^{-1}} \\ - \frac{1 - \cos\left(\frac{\alpha}{1.125}\right) e^{-\alpha} Z^{-1}}{1 - 2 \cdot \cos\left(\frac{\alpha}{1.125}\right) e^{-\alpha} Z^{-1} + e^{-2\alpha} Z^{-2}} \\ + \frac{0.118056 \sin\left(\frac{\alpha}{1.125}\right) e^{-\alpha} Z^{-1}}{1 - 2 \cdot \cos\left(\frac{\alpha}{1.125}\right) e^{-\alpha} Z^{-1} + e^{-2\alpha} Z^{-2}} \end{array} \right) \quad (29)$$

soit 
$$F^+(Z) = \frac{c_0 \cdot Z^{-1} + c_1 \cdot Z^{-2}}{1 + b_1 \cdot Z^{-1} + b_2 \cdot Z^{-2} + b_3 \cdot Z^{-3}} K_1 \quad (30)$$

La partie anti-causale se calcule par :

$$F^-(Z) = \sum_{n=-\infty}^{-1} f(n) Z^{-n} = K_1 \left( \begin{array}{l} \sum_{n=0}^{\infty} e^{-\alpha(n+1)} Z^{-(n+1)} \\ - \sum_{n=0}^{\infty} \cos\left(\frac{\alpha(n+1)}{1.125}\right) e^{-\alpha(n+1)} Z^{-(n+1)} \\ + 0.118056 \sum_{n=0}^{\infty} \sin\left(\frac{\alpha(n+1)}{1.125}\right) e^{-\alpha(n+1)} Z^{-(n+1)} \end{array} \right) \quad (31)$$

et sa transformée en  $Z$  est donnée comme suit :

$$F^-(Z) = K_1 \left( \begin{array}{l} \frac{e^{-\alpha} Z}{1 - e^{-\alpha} Z} \\ - \frac{\cos\left(\frac{\alpha}{1.125}\right) e^{-\alpha} Z - e^{-2\alpha} Z^2}{1 - 2 \cdot \cos\left(\frac{\alpha}{1.125}\right) e^{-\alpha} Z + e^{-2\alpha} Z^2} \\ + \frac{0.118056 \sin\left(\frac{\alpha}{1.125}\right) e^{-\alpha} Z}{1 - 2 \cdot \cos\left(\frac{\alpha}{1.125}\right) e^{-\alpha} Z + e^{-2\alpha} Z^2} \end{array} \right) \quad (32)$$

soit 
$$F^-(Z) = \frac{c_0 \cdot Z^{-1} + c_1 \cdot Z^{-2}}{1 + b_1 \cdot Z^{-1} + b_2 \cdot Z^{-2} + b_3 \cdot Z^{-3}} K_1 \quad (33)$$

avec

$$c_0 = \left( 1 + 0.118056 \cdot \sin\left(\frac{\alpha}{1.125}\right) - \cos\left(\frac{\alpha}{1.125}\right) \right) e^{-\alpha}$$

et 
$$c_1 = \left( 1 - 0.118056 \cdot \sin\left(\frac{\alpha}{1.125}\right) - \cos\left(\frac{\alpha}{1.125}\right) \right) e^{-2\alpha}.$$

La réponse impulsionnelle  $y_f(m)$  d'un signal numérique  $S(m)$  correspondant aux filtres (30) et (33) s'écrit sous la forme :

$$y_f(m) = y^+(m) - y^-(m) \quad (34)$$

où  $y^+(m)$ ,  $y^-(m)$  et  $K_1$  sont donnés par :

$$y^+(m) = K_1 (c_0 \cdot S(m-1) + c_1 \cdot S(m-2) - b_1 \cdot y(m-1) - b_2 \cdot y(m-2) - b_3 \cdot y(m-3))$$

$$y^-(m) = K_1 (c_0 \cdot S(m+1) + c_1 \cdot S(m+2) - b_1 \cdot y(m+1) - b_2 \cdot y(m+2) - b_3 \cdot y(m+3))$$

$$K_1 = (1 + b_1 + b_2 + b_3) / (c_0 + c_1).$$

Nous avons vu que l'implantation numérique des filtres proposés correspond à des filtres récurrents d'ordre 3. Le filtre récurrent  $h(x)$  a besoin de 12 multiplications et 11 additions tandis que  $f(x)$  a besoin de 10 multiplications et 9 additions. Par ailleurs, Deriche (Deriche, 1990) a proposé une paire de filtres récurrents d'ordre 2 et une autre paire de filtres (Deriche, 1993) qui sont l'approximation de la Gaussienne et sa première dérivée (correspondant aux filtres récurrents d'ordre 4). Ces derniers ont besoin de 16 multiplications et 15 additions pour chaque filtre, ce qui les rend plus coûteux en terme temps de calcul que les filtres proposés dans cet article.

### 3.3 Implantation récurrente 2-D

Pour obtenir une implantation récurrente 2-D plus efficace, on peut considérer la séparabilité des filtres où une image peut être analysée séparément selon les deux directions  $x$  (ou  $m$ ) et  $y$  (ou  $n$ ), c'est-à-dire le lissage selon  $x$  suivi par une détection selon  $y$  (contours horizontaux) et la détection selon  $x$  suivie par le lissage selon  $y$  (contours verticaux). L'algorithme ci-dessous représente la détection de contours 2-D par la convolution de l'image  $I(m,n)$  par les filtres proposés.

for  $n = 0$  to  $N-1$  do

for  $m = 0$  to  $M-1$  do

$$E_1(m,n) = H(n) * (F(m) * I(m,n))$$

$$E_2(m,n) = F(n) * (H(m) * I(m,n))$$

$$E(m,n) = \sqrt{E_1(m,n)^2 + E_2(m,n)^2}$$

end  $m$  end  $n$ .

où  $F(n)$  et  $F(m)$  sont les filtres numériques de  $f(x)$  pour les axes  $n$  et  $m$  respectivement, alors que  $H(n)$  et  $H(m)$  sont les filtres numériques de  $h(x)$ .

## 4 Résultats expérimentaux

Dans cette section, nous présentons les résultats de contours obtenus par les filtres proposés et les filtres de Deriche. La figure 7 illustre des images synthétiques où 7-a représente une image contenant le contour marche (échelon), 7-b est l'image 7-a rendue floue par une Gaussienne avec  $\sigma=40$ , 7-c correspond à l'image 7-a qui est bruitée par un bruit blanc gaussien (SNR=2) et 7-d correspond à l'image 7-b qui est bruitée par un bruit blanc gaussien (SNR=2).

La figure 8 montre les contours de l'image floue (figure 7-b). La figure 9 illustre les contours de l'image bruitée (figure 7-c) alors que la figure 10 présente les contours de l'image floue et bruitée (figure 7-d). Nous constatons que les filtres proposés dans cet article donnent de meilleurs résultats que les filtres de Deriche. À titre d'exemple, pour l'image floue (figure 8-a) et pour  $\alpha=0.85$ , le détecteur de Deriche donne une réponse avec un triple contour et le détecteur proposé a une réponse double contour. De même, la figure 10, montre que pour les mêmes

valeurs de  $\alpha$ , quelques contours ont des réponses multiples dans les filtres de Deriche, alors qu'un simple contour est fourni par notre solution.

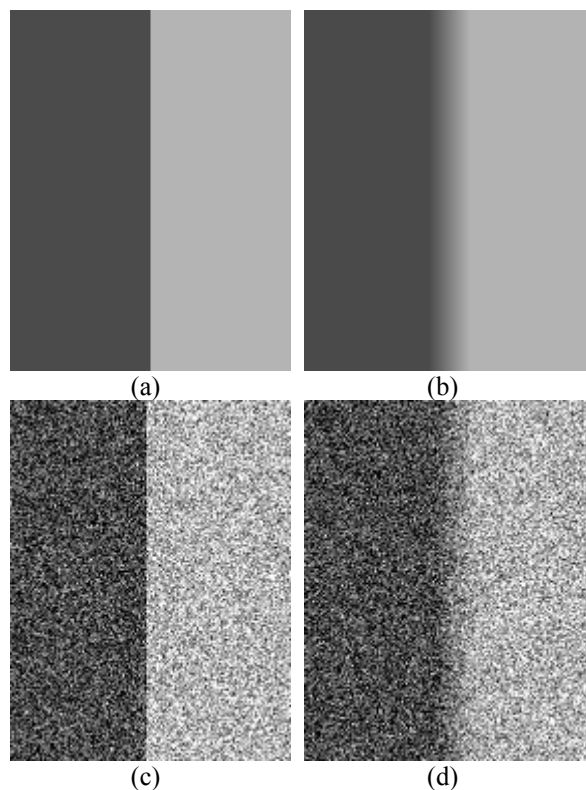


Figure 7. (a) Image synthétique d'un contour marche, (b) Image (a) rendue floue par une Gaussienne avec  $\sigma=40$ , (c) Image (a) bruitée avec SNR=2 et (d) Image (b) bruitée avec SNR=2.

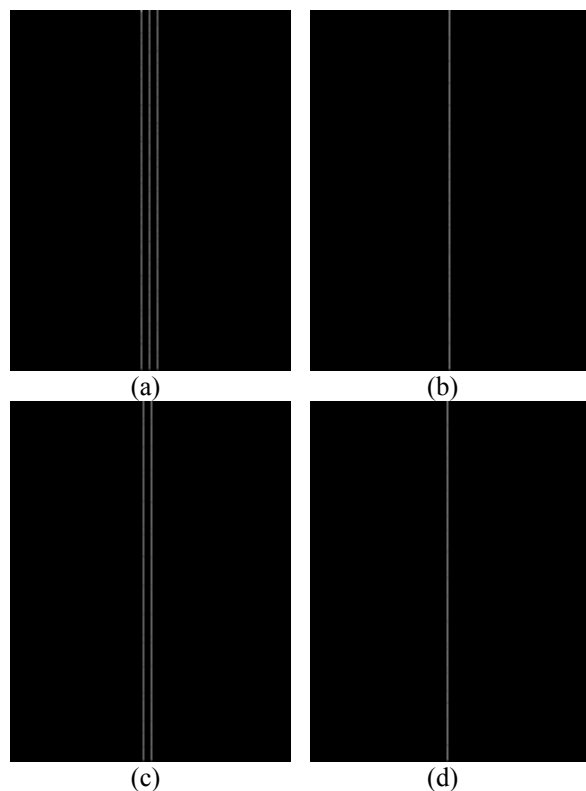
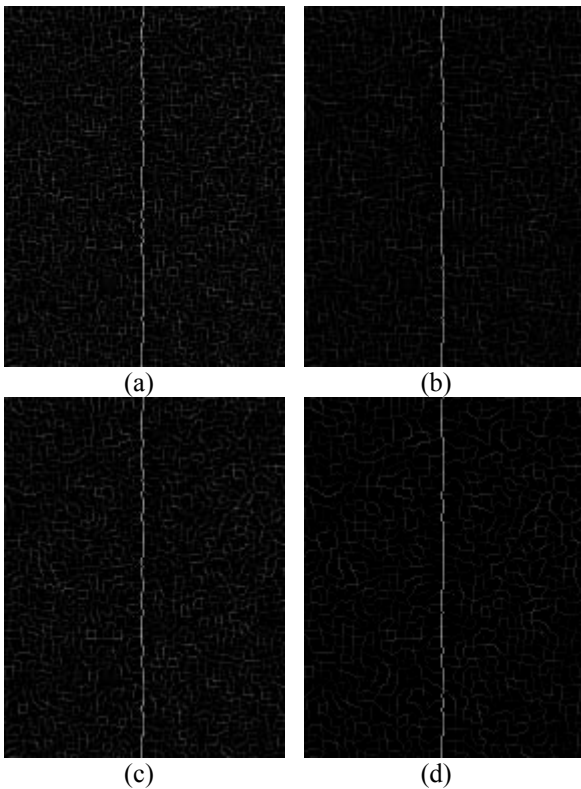
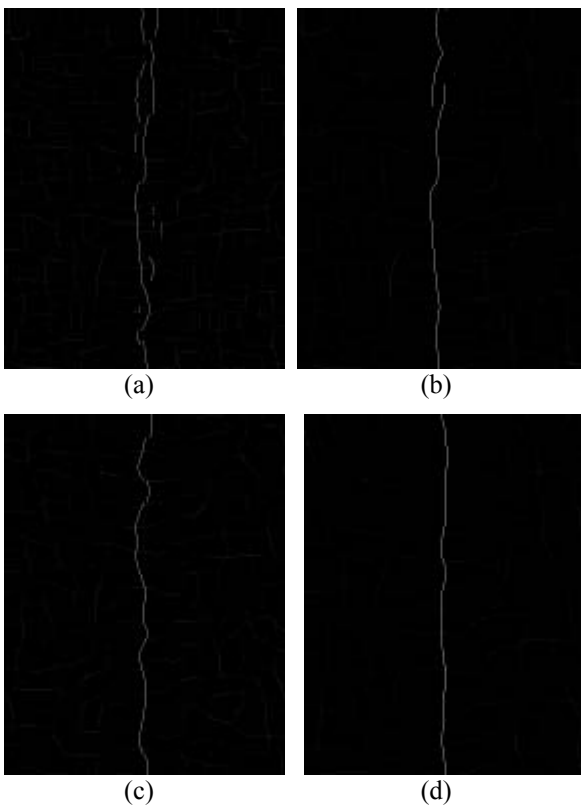


Figure 8. (a) et (b) sont des contours de l'image de la figure 7-b obtenus par les filtres de Deriche avec  $\alpha=0.85$  et  $\alpha=0.75$  respectivement. (c) et (d) sont des contours obtenus par nos filtres avec  $\alpha=0.85$  et  $\alpha=0.75$  respectivement.



**Figure 9.** (a) et (b) sont de contours de l'image de la figure 7-c obtenus par les filtres de Deriche avec  $\alpha=0.5$  et  $\alpha=0.25$  respectivement. (c) et (d) sont des contours obtenus par nos filtres avec  $\alpha=0.5$  et  $\alpha=0.25$  respectivement.



**Figure 10.** (a) et (b) sont de contours de l'image de la figure 7-d obtenus par les filtres de Deriche avec  $\alpha=0.35$  et  $\alpha=0.25$  respectivement. (c) et (d) sont de contours obtenus par nos filtres avec  $\alpha=0.35$  et  $\alpha=0.25$  respectivement.

## 4 Conclusion

Nous avons présenté dans cet article deux filtres : la fonction de lissage  $h(x)$  et le détecteur de contours  $f(x)$  qui sont mieux adaptés aux images : nettes, floues, bruitées ou floues et bruitées. Ces filtres peuvent être considérés comme l'approximation de la Gaussienne et sa première dérivée respectivement. Ils sont implantables sous forme récursive d'ordre 3. Les résultats obtenus en termes de qualité (rapport signal sur bruit) et de précision (localisation de contours) sont nettement meilleurs que ceux de Deriche. Ces filtres sont intégrés dans un système de recherche d'images basée sur le contenu, appelé MIRA (*Multimedia Information Retrieval Application*) et ont fait l'objet d'expérimentations (Missaoui et al., 2005).

## Reference

- E. Bourenane., P. Gouton, M. Paindavoine, F. Truchetet, "Generalization of Canny-Deriche Filter for detection of noisy exponential edge", *Signal Processing* 82 (2002) 1317 – 1328.
- R. Deriche, "Fast algorithms for low-level vision", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(12):78-88, January 1990.
- R. Deriche, Recursively implementing the Gaussian and its derivatives. Research Report 1893, INRIA, France, 1993.
- P. Gouton, H. Laggoune, Rk. Kouassi, M. Paindavoine, "Optimal Detector for Crest Lines", *Optical engineering*, June 2000.
- H. Laggoune, "Contributions à l'étude des filtres optimaux pour des contours 2-D et 3-D quelconques", thèse, 1996, Université de Bourgogne, Dijon, France.
- R. Missaoui, M. Sarifuddin et H. Laggoune, "Détection efficace de contours par décomposition de l'image en régions homogènes", Actes de la conférence SETIT 2005, Sousse-Tunise, Mars 2005.
- M. Sarifuddin, M. Paindavoine, C. Milan, J.C. Grapin, "Multiscale Edges Detection Algorithm Using FPGA Devices", *Third Workshop on Electro Communication and Information*, Institute of Technology of Bandung, Mars 3-4, 1999, Indonesia.
- M. Sarifuddin, H.Laggoune, "Real time edges detection using wavelet transform", *Proceedings 28th Visualization Symposium Japan*, Tokyo-Japan, 17-19 July 2000.
- I. T. Young, Lucas J. van Vliet, and Michael van Ginkel, "Recursive Gabor Filtering", *IEEE Transactions on Signal Processing*, Vol. 50, No. 11, November 2002.