Chapitre 1

Introduction

1.1. Qu'est-ce que la vision ?

L'espace qui nous entoure a une structure tri-dimensionnelle (3D). Lorsque l'on demande à une personne de décrire ce qu'elle *voit*, elle n'éprouve aucune difficulté à nommer les objets qui l'entourent : téléphone, table, livre... Et pourtant l'information qui est réellement disponible sur la rétine des ses yeux n'est, ni plus ni moins, une collection de points (environ un million !). En chaque point ou *pixel* (picture element) il y a tout simplement une information qui donne une indication quant à la quantité de lumière et la couleur qui proviennent de l'espace environnant et qui ont été projetées à cet endroit de la rétine. Le téléphone, la table ou le livre *n'existent pas* sur la rétine. Guidé à la fois par l'information codée dans l'image (ou la rétine) et par ses propres connaissances, le processus visuel *construit* des percepts. Le téléphone ou le livre sont le réponses finales, résultant d'un processus *d'interprétation* qui fait partie intégrante du système de vision. De plus, il n'y a pas de correspondance terme à terme entre l'information sensorielle (la lumière et la couleur) et la réponse finale (des objets 3D). Le système de vision doit *fournir* les connaissances nécessaires afin de permettre une interprétation non ambiguë.

1.2. Comprendre la vision

Il n'est pas suffisant de constater qu'un problème est complexe. Encore faut-il essayer de le comprendre dans ces moindres détails et de proposer une solution.

La vision a suscité l'intérêt de nombreux scientifiques et philosophes depuis déjà très longtemps. Parmi ceux-ci, les neurobiologistes mènent des recherches théoriques et expérimentales afin d'essayer de comprendre l'anatomie et le fonctionnement du cerveau dans son ensemble. Ils ont découvert une structure très complexe qui est loin de leur avoir révélé tous ses secrets. La tâche des neurobiologistes semble être à la fois grandiose et illusoire. Grandiose, parce que le cerveau est une des plus complexes *inventions* de la nature. Il reste et restera pour longtemps le bastion encore inconnu que les sciences humaines se proposent de conquérir. Illusoire, car on ne connaît pas ses limites. Ces limites ne sont-elles par repoussées à chaque découverte ? David Hubel [99] a merveilleusement bien exprimé ce paradoxe : *Le cerveau peut-il comprendre le cerveau ?*

Avec la naissance de machines de calcul de plus en plus sophistiquées, un certain nombre de scientifiques se sont attaqués au problème de la vision d'un point de vue quantitatif : est-il possible de construire un *modèle computationnel* pour la perception visuelle ? Attention : il ne s'agit pas de fournir une explication de comment marche la vision biologique mais de **créer un modèle** qui, vu de l'extérieur, possède des propriétés semblables.

Ce modèle *artificiel* peut-il être d'une utilité quelconque quant à la vision biologique ? Peut-il constituer la base d'une nouvelle technologie – *des machines qui voient* ?

Il est certainement trop tôt pour répondre à ces questions et pour tirer des conclusions. Malgré les efforts non négligeables, il y a très peu de résultats convaincants. Nous pensons que deux démarches doivent être suivies simultanèment :

- essayer d'élaborer une théorie de la **vision par ordinateur** qui doit nous guider à long terme ;
- tenter de résoudre des problèmes spécifiques dans le cadre de cette théorie : de tels résultats partiels permettraient de confirmer ou au contraire de mettre en cause certains aspects de la théorie.

1.3. Une théorie de la vision

L'élaboration d'une théorie scientifique demande trois étapes :

1. énoncer la théorie, spécifier et élaborer les concepts de base : ces concepts doivent exprimer le cadre formel qui est à la base de la théorie,

2. exprimer ces concepts sous forme mathématique,

3. réaliser un ensemble expérimental qui permette de vérifier la théorie.

Voici comment la vision par ordinateur peut s'énoncer brièvement dans les termes de ce paradigme. La vision est un processus de traitement de l'information. Elle utilise des stratégies bien définies afin d'atteindre ses buts. L'entrée d'un système de vision est constituée par une séquence d'images. Le système lui-même apporte un certain nombre de connaissances qui interviennent à tous les niveaux. La sortie est une description de l'entrée en termes d'objets et de relations entre ces objets.

Deux types de stratégies sont mises en jeu : ascendantes et descendantes. Les stratégies ascendantes tentent de construire à partir de l'information sensorielle une représentation la plus abstraite possible (par exemple, un ensemble de primitives géométriques 3D). Les stratégies descendantes déduisent à partir de l'ensemble d'objets connus par le système une description compatible avec les primitives extraites de l'image. Il est alors possible de mettre en correspondance la représentation extraite de l'image avec les descriptions des objets afin de décrire les données sensorielles en termes de ces objets.

Les connaissances mises en jeu peuvent être de trois types : physiques, géométriques et sémantiques. Les lois physiques imposent des contraintes aux signaux lumineux qui partant d'une source, traversent la scène et se projettent sur l'image. La gravitation impose à la scène (et donc à l'image) une structure hétérogène : prépondérance de lignes verticales et horizontales pour ne citer qu'un exemple. La forme des objets (l'ensemble de ses surfaces) et la géométrie de la formation de l'image imposent des contraintes très strictes quant aux structures susceptibles d'être présentes dans l'image. A un niveau plus élevé, un objet peut être décrit par sa fonction dans le contexte d'un raisonnement symbolique. Cette fonction n'est pas directement mesurable dans l'image. On devrait pouvoir dériver des contraintes sur la forme et l'emplacement d'un objet à partir de sa fonction. Par exemple, le mot *chaise* désigne une classe d'objets réels (un objet réel est un objet qui occupe une place dans l'espace physique). Cependant il y a une grande variété de chaises quant à la forme et à la couleur pour ne citer que deux propriétés. Quelles sont les propriétés communes à toutes les chaises, **mesurables** dans l'image ?

L'étape suivante consiste à exprimer ces stratégies et connaissances dans le cadre d'un formalisme mathématique et à construire les algorithmes correspondants. Les performances de ces algorithmes doivent correspondre aux qualités exigées d'un tel système : *la reconnaissance visuelle doit être fiable et rapide.*

1.4. Le paradigme de David Marr

Vers la fin des années 70, David Marr [121] a proposé un modèle calculatoire pour le traitement et la représentation de l'information visuelle. Voici quels sont les principaux traits de ce paradigme :

• à partir d'une ou de plusieurs images un processus d'extraction de caractéristiques produit un description en termes d'attributs bi-dimensionnels ; ce niveau de représentation est appelé *première ébauche* (primal sketch) ;

• la première ébauche constitue l'entrée d'un certain nombre de processus plus ou moins indépendants qui calculent des propriétés tri-dimensionnelles locales relatives à la scène ; il s'agit d'une représentation centrée sur l'observateur, appelée *ébauche 2.5D* ; ces processus opèrent sur une séquence d'images (analyse du mouvement) sur une paire d'images (stéréoscopie) ou sur une seule image. Dans ce dernier cas il s'agit de processus d'inférence qui utilisent des connaissances géométriques (analyse des contours), géométriques et statistiques (analyse des textures), photométriques (analyse des ombrages) ou colorimétriques (analyse des reflets) ;

• l'ébauche 2.5D est mise en correspondance avec des connaissances 3D afin de construire une description de la scène en termes d'objets et de relations entre les objets ; il s'agit maintenant d'une représentation *centrée sur la scène* (la description ne dépend plus de la position de l'observateur).

1.5. Segmentation, reconstruction, reconnaissance

En pratique, le paradigme de David Marr se traduit par trois étapes de traitement : segmentation, reconstruction et reconnaissance.

La segmentation d'images étant la pierre de base de tout système de vision, de nombreux travaux lui ont été consacrés. La diversité des images, la difficulté du problème, les origines variées des chercheurs, l'évolution de la puissance de calcul des ordinateurs, et un certain empirisme dans l'évaluation des résultats ont conduit à l'introduction d'une multitude d'algorithmes.

Quelle que soit son origine, une image constitue une représentation d'un univers composé d'entités : objets dans une scène d'intérieur, cellules, surfaces sismiques, organes du corps humain ... Le but de toute méthode de segmentation est l'extraction d'attributs caractérisant ces entités. Les attributs étudiés correspondent à des points d'intérêt ou à des zones caractéristiques de l'image : contours et régions. La détection des contours (chapitre 2) implique la recherche des discontinuités locales de la fonction des niveaux de gris de l'image. La segmentation des contours (chapitre 3) consiste à approximer les contours par des représentations analytiques, telles que des droites ou des coniques. L'extraction de régions (chapitre 4) revient à déterminer des zones homogènes en niveaux de gris de l'image. Par exemple, dans le cas d'images réelles, les contours correspondent aux frontières des objets et les régions à leurs surfaces. Ces deux approches "contour" et "région" sont duales en ce sens qu'une région définit une ligne par son contour, et qu'une ligne fermée définit une région. Elles amènent cependant à des algorithmes complètement différents et ne fournissant pas les mêmes résultats. Cette dualité est cependant peu exploitée dans la plupart des méthodes existantes.

Un autre aspect de la segmentation est celui qui consiste à retrouver la géométrie des objets à partir des images. On obtient ainsi des représentations intrinsèques, aisément manipulables et utilisables, à partir de la réalité physique induite par l'image. De manière à obtenir ces caractéristiques géométriques, on est souvent conduit à définir une suite hiérarchique de représentations de l'information image permettant finalement d'obtenir des indices visuels servant à résoudre une tâche donnée. Dans les chapitres 10 et 11 nous illustrons ces principes dans le cas des images bi-dimensionnelles et des images volumiques.

La calibration est la première étape indispensable pour toute méthode de reconstruction (à moins que la calibration ait lieu en même temps que la reconstruction). Le chapitre 5 décrit en détail les modèles géométriques de plusieurs capteurs basés sur une caméra ainsi que plusieurs techniques de détermination des paramètres de ces capteurs (calibration). On étudiera ainsi la caméra matricielle, la caméra linéaire ainsi que les capteurs stéréoscopiques passifs et actifs. Le chapitre 12 aborde le problème de la caractérisation des cartes de profondeur obtenues avec un capteur stéréoscopique actif (caméra et faisceau laser).

Le chapitre 6 décrit en détail les principes de reconstruction tri-dimensionnelle à partir d'un système stéréoscopique. Plus particulièrement, le problème de mise en correspondance stéréo est abordé d'un point de vue géométrique et algorithmique. La reconstruction de surfaces polyédriques par vision stéréoscopique à partir d'images 2D peut être réalisée par une approche régions ou par une approche contours. Le chapitre 10 présente une approche de type géométrie algorithmique pour résoudre le problème de reconstruction polyédrique.

La reconnaissance consiste essentiellement à comparer des indices visuels bi- ou tri-dimensionnels avec les indices des objets à reconnaître. Les méthodes de reconnaissance sont souvent couplées avec des méthodes de localisation. Le chapitre 7 décrit quelques méthodes de localisation (position et orientation avec six degrés de liberté) à partir d'indices visuels 3D mis en correspondance avec des indices d'objets 3D. Le chapitre 8 décrit une méthode de localisation à partir d'indices visuels 2D mis en correspondance avec des indices d'objets 3D. Enfin le chapitre 9 montre comment on peut combiner les méthodes de localisaton avec des méthodes de recherche arborescente pour pouvoir reconnaître des objets rigides.

1.6. Quelques références bibliographiques

Le premier ouvrage consacré partiellement à la vision par ordinateur est celui de Duda et Hart, datant de 1973 [50]. A une première partie consacrée à la reconnaissance des formes, fait suite une deuxième partie qui introduit les bases théoriques d'une approche géométrique de l'interprétation d'une image.

Les ouvrages de Gonzales et Wintz (1977) et de Rosenfeld et Kak (1982, seconde édition) passent en revue l'état de l'art de ce qu'on appelle aujourd'hui la "vision bas niveau" [73], [160].

Pendant longtemps, l'ouvrage de référence en vision par ordinateur a été celui de Ballard et Brown (1982), [15]. Sans rentrer dans les détails mathématiques, ce texte fournit une vue synthétique des travaux de recherche dans les années 80.

L'ouvrage de Horn, publié en 1986, aborde quelques aspects de la vision d'un point de vue plus fondamental [97]. Les bases mathématiques de la formation d'une image, de la détection de contours et de régions, des propriétés photométriques ainsi que de la perception du mouvement sont clairement présentées.

L'utilisation de la vision par ordinateur pour la navigation des robots est le thème de l'ouvrage d'Ayache, publié en 1989 [8] et en 1991 [9] (version anglaise). On y trouve notamment les détails de l'utilisation du filtre de Kalman étendu pour intégrer l'information provenant de plusieurs cartes stéréoscopiques.

Enfin, l'ouvrage de Faugeras est, avec le nôtre, le plus récent [55]. Il propose une approche géométrique (géometrie projective et euclidienne) pour résoudre notamment le problème de reconstruction. Très clair et très détaillé, contenant de nombreux exemples ainsi que des exercices, ce texte rend compte de 10 années de travaux de recherche effectués par l'auteur et par son équipe de l'INRIA.

Chapitre 2

Détection de contours

Les contours des objets dans des images de dimensions quelconques (images naturelles 2D, images médicales 3D ...) correspondent le plus souvent aux extréma locaux du gradient ou aux zéros du laplacien de la fonction des niveaux de gris. Les difficultés de la détection des contours proviennent du bruit important présent dans les images (bruit du capteur, bruit d'échantillonnage, irrégularités de la surface des objets ...). On est donc confronté au problème de la différenciation d'un signal bruité. Pour le résoudre, il est d'abord nécessaire de définir des critères de performance d'un opérateur incluant une modélisation des contours recherchés et la notion de bruit. Dans une deuxième phase, on déduit de ces critères une famille de filtres optimaux. Dans une troisième phase, on résout le problème posé par l'implantation de ces filtres.

Nous mettrons l'accent dans cette partie sur l'utilisation de filtres linéaires. En effet, dans le cas où on ne dispose pas de connaissances a priori sur l'image, ce type de méthode fournit des algorithmes de faible complexité dont les résultats sont satisfaisants sur la plupart des types d'images. Nous nous attacherons plus particulièrement aux filtres séparables récursifs car ils peuvent se généraliser à une dimension quelconque (séparabilité) , et permettent une implantation de faible coût algorithmique pour des opérateurs de réponse impulsionnelle infinie (récursivité).

2.1. Généralités

2.1.1. Filtrage linéaire d'une image

Rappelons rapidement que filtrer signifie convoluer une image $I(x,y)$ avec une fonction $f(x,y)$ qui s'appelle réponse impulsionnelle du filtre. Dans le cas continu l'image filtrée est donnée par :

$$\begin{aligned} I_f(x,y) &= (f*I)(x,y) && [2.1] \\ &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x',y')I(x-x',y-y')dx'dy' && [2.2] \\ &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x-x',y-y')I(x',y')dx'dy' && [2.3] \end{aligned}$$

Dans le cas discret les domaines de I et de f sont bornés. Le domaine de I est $[-N/2,+N/2]$ et le domaine de f est $[-K/2,+K/2]$. On a nécessairement $K \leq N$, N étant la taille de l'image. Dans le cas discret la convolution s'écrit :

$$\begin{aligned} I_f(x,y) &= (f*I)(x,y) && [2.4] \\ &= \sum_{i'=-K/2}^{i'=+K/2}\sum_{j'=-K/2}^{j'=+K/2} f(i-i',j-j')I(i',j') && [2.5] \end{aligned}$$

On notera que le filtrage linéaire consiste simplement à remplacer chaque niveau de gris par une combinaison linéaire des niveaux de gris des points voisins ; les cœfficients de cette combinaison linéaire sont définis par la réponse impulsionnelle du filtre. Cette réponse impulsionnelle est la réponse du filtre à la fonction impulsion (d'où son nom!).

2.1.2. Le gradient d'une image

Le gradient d'une image se calcule comme suit :

$$\begin{aligned} I_x(x,y) &= \frac{\partial I_f(x,y)}{\partial x} && [2.6] \\ I_y(x,y) &= \frac{\partial I_f(x,y)}{\partial y} && [2.7] \end{aligned}$$

En chaque point (x, y) de l'image nous pouvons donc calculer le vecteur *gradient*. Le module et la direction de ce vecteur sont donnés par :

$$\begin{aligned} G &= (2_x + 2_y)^{\frac{1}{2}} & [2.8] \\ &\simeq \max(I_x, I_y) & [2.9] \\ &\simeq \mid I_x \mid + \mid I_y \mid & [2.10] \\ \phi &= \arctan(\frac{I_y}{I_x}) & [2.11] \end{aligned}$$

La direction du gradient maximise la dérivée directionnellle, et la norme du gradient est la valeur de cette dérivée. On obtient la dérivée de I dans une direction quelconque $\vec{d}$ à partir des deux dérivées directionnelles définissant le gradient I_x et I_y de la manière suivante : $L_{\vec{d}}(I) = \nabla I \cdot \vec{d}$.

2.1.3. Interprétation géométrique du gradient

Afin de mieux comprendre la signification du gradient, considérons une image constituée de deux régions "plates" séparées par une marche rectiligne (voir figure 2.3.). Essayons d'exprimer analytiquement cette image. Nous avons besoin pour cela de la fonction marche :

$$u(t) = \begin{cases} 1 & \text{si} \quad t \geq 0 \\ 0 & \text{si} \quad t < 0 \end{cases} \qquad [2.12]$$

Nous avons également besoin de la fonction impulsion :

$$\delta(t) = \begin{cases} 1 & \text{si} \quad t = 0 \\ 0 & \text{sinon} \end{cases} \qquad [2.13]$$

La fonction marche peut être écrite comme l'intégrale de la fonction de Dirac :

$$u(z) = \int_{-\infty}^{z} \delta(t) dt \qquad [2.14]$$

La dérivée de $u(t)$ est :

$$u'(t) = \delta(t) \qquad [2.15]$$

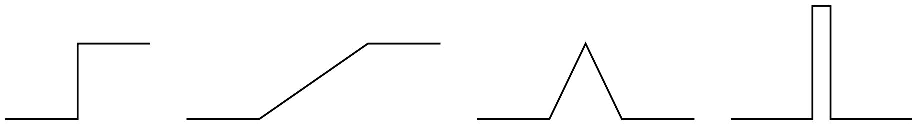


Figure 2.1. *Quelques profils de contours : marche, rampe, toit, pic.*

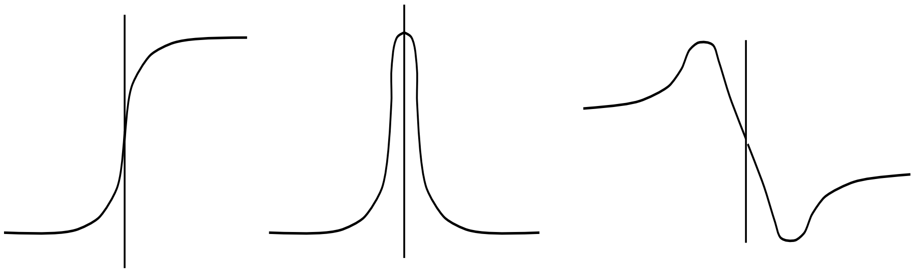


Figure 2.2. *Les deux principes de la détection de contours : dérivée première et dérivée seconde (discontinuité du signal, dérivée première, dérivée seconde).*

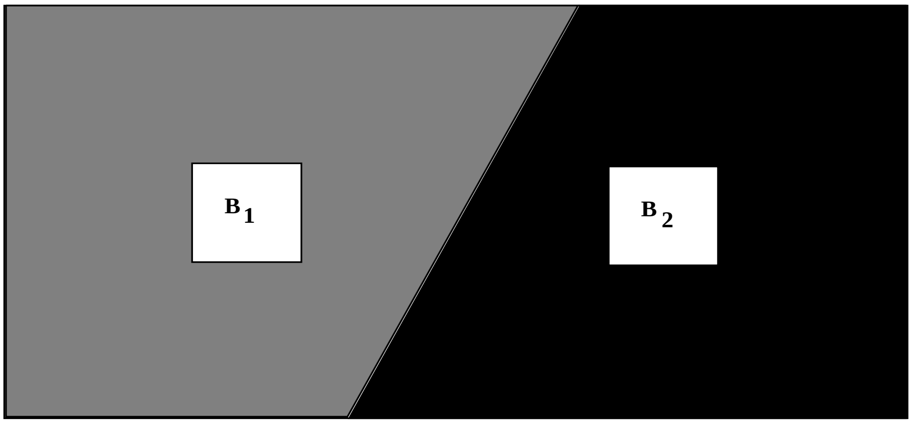


Figure 2.3. *Deux régions plates séparées par une marche d'escalier.*

Ecrivons maintenant l'équation de la droite décrivant la marche de l'image. Cette droite fait un angle θ avec l'axe des x, son équation s'écrit donc :

$$-x\sin\theta + y\cos\theta + \rho = 0 \qquad [2.16]$$

Nous pouvons maintenant écrire la fonction image :

$$I(x,y) = B_1 + (B_2 - B_1)u(-x\sin\theta + y\cos\theta + \rho) \qquad [2.17]$$

Les composantes du vecteur gradient au point image (x, y) sont donc :

$$\begin{aligned} I_x &= \cos(\theta + \frac{\pi}{2})(B_2 - B_1)\delta(-x\sin\theta + y\cos\theta + \rho) \qquad [2.18] \\ I_y &= \sin(\theta + \frac{\pi}{2})(B_2 - B_1)\delta(-x\sin\theta + y\cos\theta + \rho) \qquad [2.19] \end{aligned}$$

Le gradient est nul partout dans l'image sauf le long du contour en marche. Le vecteur gradient se trouve dans le plan image et il est *perpendiculaire* à la direction du contour :

$$\phi = \theta + \frac{\pi}{2} \qquad [2.20]$$

La norme du gradient est égale à la dérivée de l'image le long de la normale au contour soit : $B_2 - B_1$.

2.2. Dérivation et séparabilité des filtres

L'objectif de cette partie est de montrer que sous réserve d'hypothèses raisonnables, on peut ramener la détection de contours au lissage et à la dérivation par filtrage linéaire d'un signal monodimensionnel. On notera que les méthodes décrites dans les développements suivants s'appliquent pour une image de dimension quelconque.

Soit $I(x, y)$ une image de dimension 2 (le même développement peut être réalisé pour un signal de dimension quelconque).

On admet communément que les contours correspondent aux discontinuités d'ordre 0 de I . On peut noter que cette hypothèse peut être mise à mal par certains contours associés aux discontinuités d'ordre 1 (contours en toit). Quoi qu'il en soit peu de travaux ont été consacrés à la détection de ce deuxième type de contours.

La détection de contours s'effectue classiquement de deux manières :

1. calcul du gradient et extraction des extréma locaux de la norme du gradient dans la direction du gradient : **approche gradient**.

2. calcul du laplacien et détermination des passages par zéro : **approche laplacien**.

On remarque que la première approche se ramène à extraire les zéros de la dérivée seconde dans la direction du gradient, ce qui est différent des passages par zéro du laplacien. Pratiquement ces deux types de méthodes fournissent cependant des résultats très proches.

Soit G le gradient de I :

$$G(x,y) = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})^t$$

Soit L le laplacien de I :

$$L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

Il s'agit donc de trouver des méthodes permettant de calculer $G(I)$ et $L(I)$. Une manière commode de poser ce problème est de rechercher des filtres linéaires permettant l'approximation du gradient ou du laplacien. Si on suppose le bruit de moyenne nulle, le filtrage linéaire est susceptible d'amener des solutions satisfaisantes. Pour la plupart des images traitées l'hypothèse d'un bruit blanc gaussien est raisonnable. Cependant dans certains cas un bruit impulsionnel nécessite au préalable un filtrage de lissage non linéaire par exemple un filtrage médian [122]. Le principe du filtre médian est de calculer en un point non pas une combinaison linéaire des niveaux de gris de ses voisins, mais une valeur médiane dépendant d'un tri des niveaux de gris des points voisins. Les figures 2.4. à 2.7. illustrent cette remarque en dimension 1 :

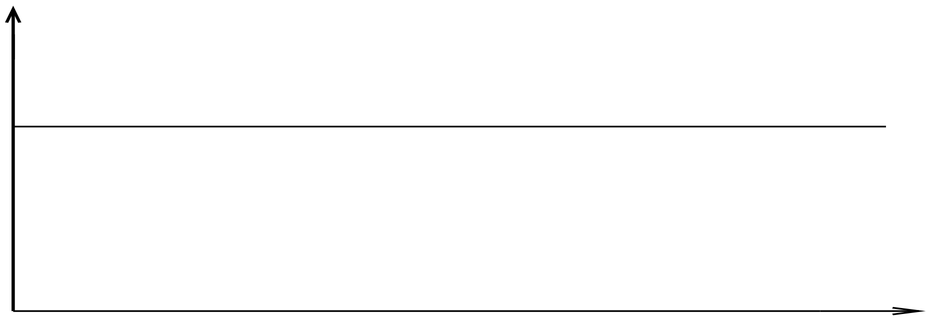


Figure 2.4. *Signal idéal.*

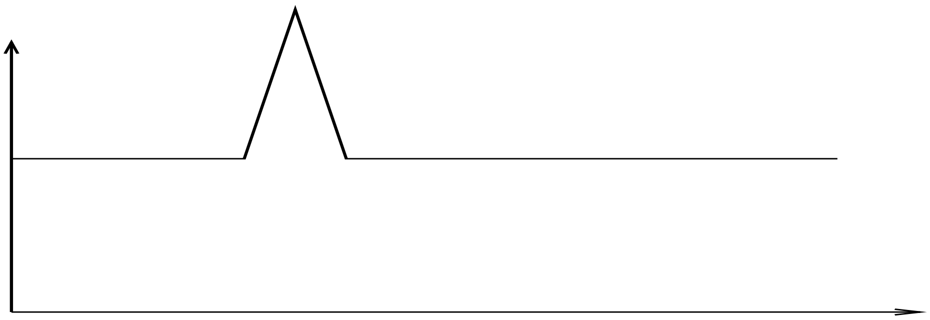


Figure 2.5. *Signal bruité par un bruit impulsionnel.*

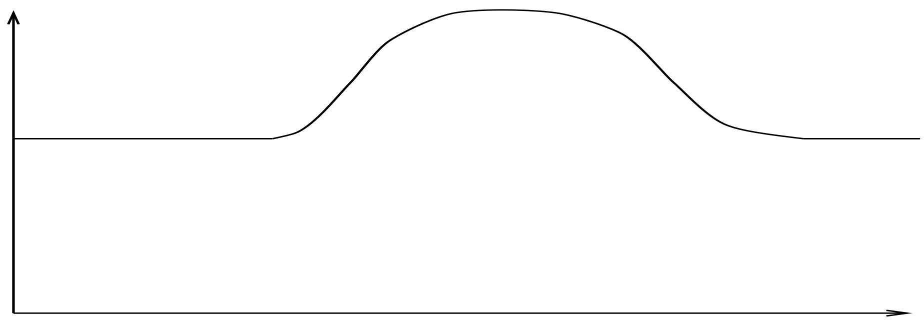


Figure 2.6. *Signal lissé par un filtre linéaire : pas d'issue !*

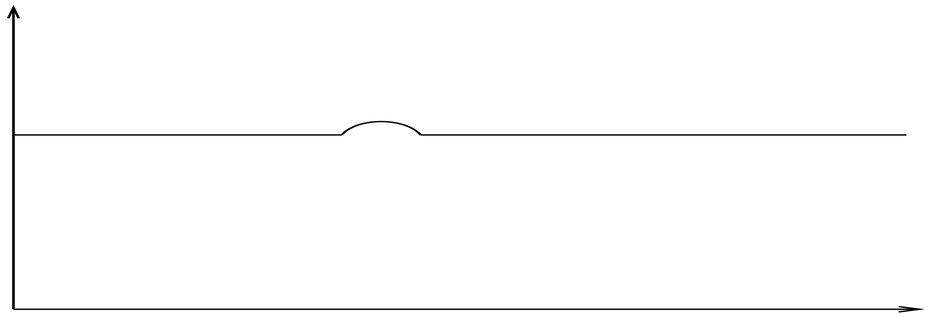


Figure 2.7. *Signal après un filtrage médian de dimension 3*
Soit $(i_1, i_2, \dots i_n)$ *la suite des valeurs de* I
Soit $(j_1, j_2, \dots j_n)$ *la suite des valeurs de* I *après filtrage*
j_p *: valeur médiane de* (i_{p-1}, i_p, i_{p+1})
$j_p = i_p$ *si* $i_p \leq i_{p+1}$ *et* $i_{p-1} \leq i_p$
$j_p = i_{p-1}$ *si* $i_{p+1} \leq i_{p-1}$ *et* $i_{p-1} \leq i_p \dots$

On a donc ramené la détection de contours à la détermination de filtres linéaires permettant d'approximer le gradient ou le laplacien de l'image (le gradient ou le laplacien permettent de définir les discontinuités d'ordre 0 de la fonction des niveaux de gris). Soit un filtre linéaire de réponse impulsionnelle f et I un signal, la dérivée du signal convolué avec f est égale à l'image convoluée avec la dérivée de f :

$$(I * f)' = I * f'$$

$$(I * f)'' = I * f''$$

A cause de cette propriété les approches "dérivée première" et "dérivée seconde" se ramènent à déterminer un filtre de lissage de réponse $f(x, y)$. On calcule les dérivées premières et secondes par rapport aux variables x, y en convoluant avec $\frac{\partial f(x,y)}{\partial x}$, $\frac{\partial^2 f(x,y)}{\partial x^2} \cdots$ On remarque que le calcul du laplacien peut s'effectuer directement par convolution avec le filtre de réponse :

$$\frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}$$

On obtient :

$$\begin{aligned} I_x(x,y) &= \frac{\partial(I*f)}{\partial x} = I * \frac{\partial f}{\partial x} = I * f_x \\ I_y(x,y) &= \frac{\partial(I*f)}{\partial y} = I * \frac{\partial f}{\partial y} = I * f_y \\ \Delta I(x,y) &= I * \Delta(f) \end{aligned} \quad [2.21]$$

On remarquera que parfois l'expression $\Delta(f)$ se simplifie bien. Une autre solution pour le calcul du laplacien consiste à l'approximer par différence entre l'image lissée par deux filtres de lissage de largeurs différentes [165]. Cependant les résultats obtenus avec cette deuxième solution sont nettement moins satisfaisants que ceux fournis par un calcul direct du laplacien. Un des premiers algorithmes de détection de contours dans des images bidimensionnelles introduits par Prewitt [129] revient à approximer le gradient par convolution de l'image avec des masques de convolution (les masques de convolution représentent la fonction discrète f définie dans la section 2.2.1 : "filtrage d'une image") :

$$L = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \qquad f_x = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}$$

$$f_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Figure 2.8. *Masques de Prewitt.*

Dans le cas de l'opérateur de Prewitt, l'expression analytique de la réponse impulsionnelle de L n'étant pas dérivable, on dérive alors par différences finies.

Les premiers algorithmes de détection de contours par approximation du laplacien consistent à calculer le laplacien par convolution de l'image avec un masque (par exemple voir figure 2.9.). Il est important de remarquer que dans le cas des approches reposant sur le calcul de la dérivée première, le but du calcul du gradient est de déterminer la direction selon laquelle la variation locale des niveaux de gris est la plus forte (direction du gradient) ainsi que l'intensité de cette variation (norme du gradient). Dans le cas d'un signal continu (déri-

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} \qquad \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} -1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & -1 \end{pmatrix}$$

Figure 2.9. *Masques d'approximation du laplacien.*

vable) , il suffit de dériver selon deux directions non colinéaires pour obtenir la norme et la direction du gradient invariants selon les directions de dérivation choisies. Ainsi, dans des images comprenant des points où se rejoignent plusieurs contours, et où le bruit est important, il est intéressant de calculer le gradient par une suite de dérivations directionnelles. Le principe de ces méthodes consiste à discrétiser l'espace des orientations possibles pour un contour, et à définir un masque de dérivation pour chaque direction. On sélectionne ensuite en chaque point l'orientation selon laquelle le gradient est maximum. La figure 2.10. présente les masques proposés par Kirsch [129]. Ces masques sont obtenus par rotation d'un masque de dérivation selon 8 orientations (la figure 2.10. présente les masques obtenus pour 4 orientations). Cependant, le coût élevé de l'implantation de ce type de méthodes et l'amélioration souvent assez ténue des résultats qu'elles permettent font qu'elles sont assez peu utilisées. Quoi qu'il en soit, ces algorithmes se révélent très efficaces dans les zones de l'image où plusieurs contours se croisent. Cependant, même lorsque l'on

$$N \begin{pmatrix} 5 & 5 & -3 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix} \qquad N-W \begin{pmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix}$$

$$W \begin{pmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{pmatrix} \qquad S-W \begin{pmatrix} 5 & 5 & -3 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix}$$

Figure 2.10. *Masques de Kirsch correspondant aux directions $N, N-W, W, S-W$.*

choisit d'approximer le gradient par dérivation selon n directions (n : dimension de l'image, $n = 2$ dans le cas traité), des problèmes de coût algorithmique se posent. En effet, par exemple en dimension 2 pour une image de dimensions $d_x \times d_y$, le coût d'une convolution $p \times p$ est $p^2 d_x d_y$. Si on approxime les dérivées partielles à l'aide d'un filtre s'implantant avec un masque de convolution $p \times p$ il en coûte : $2p^2 d_x d_y$. Ces remarques ont amené l'introduction de filtres à réponse impulsionnelle séparable selon les directions x, y :

$f(x,y) = f_1(x) f_2(y)$

Le filtrage séparable comporte plusieurs avantages :

- réduction du temps de calcul d'une convolution $p \times p$ de p^2 à $2p$ (pour le cas d'une image de dimension 2),
- possibilité de prise en compte d'un bruit de caractéristiques différentes selon chaque direction (par exemple dans certaines images le bruit est différent selon les deux axes X et Y),
- généralisation directe d'un filtre à une dimension quelconque,
- utilisation du filtrage récursif.

L'inconvénient majeur du filtrage séparable est que l'on peut déboucher sur des filtres anisotropiques selon les directions différentes de X, Y (l'anisotropie étant définie par rapport à la distance euclidienne). On provoque alors une légère délocalisation des contours fournis. On peut noter que les filtres gaussiens sont parmi les rares filtres séparables et isotropes (voir paragraphe 2.6.).

Dans le cas où f est un filtre séparable on obtient :

$$I * f(x,y) = I * (f_1(x) f_2(y)) \qquad [2.22]$$

Or la convolution d'un signal par le produit de deux réponses impulsionnelles séparables peut s'écrire comme la convolution par la deuxième réponse du signal convolué par la première réponse :

$$I * (f_1(x) f_2(y)) = (I * f_1(x)) * f_2(y)$$

f peut donc s'implanter par la cascade des filtres f_1 et f_2.

Si on suppose le bruit homogène selon toutes les directions on peut poser :

$$f_1 = f_2 = S$$

On supposera dans la suite que le bruit est isotrope, mais l'approche que nous décrirons s'adapte directement au cas d'un bruit anisotropique mais homogène selon X, Y. On verra en effet que le filtrage récursif ne peut être utilisé que si le bruit est homogène selon chaque variété de dimension 1 : $x = cste$ ou $y = cste$.

On obtient :

$$I * f(x,y) = I * (S(x)S(y)) = I * S(x) * S(y)$$

On pose :

$$\begin{aligned} D &= S' \\ P &= S'' \end{aligned}$$

d'où, par exemple, si on note I_x et I_{xx} les dérivées première et seconde de l'image par rapport à x on obtient :

$$\begin{aligned} I_x &= \frac{\partial I(x,y)}{\partial x} = I * (D(x)S(y)) \\ I_{xx} &= \frac{\partial^2 I(x,y)}{\partial x^2} = I * (P(x)S(y)) \\ \Delta I &= I * (S(x)P(y) + S(y)P(x)) \end{aligned}$$

Exemple :
Le premier opérateur utilisant le filtrage séparable a été introduit par Sobel (voir figure 2.11.) [160] :

$$H1 = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \qquad H2 = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Figure 2.11. *Masques de Sobel (à gauche gradient en X, à droite gradient en Y).*

Si on pose :

$$S = (1,2,1) \quad ; \quad D = (1,0,-1)$$

On obtient :

$$S(x) = (1,2,1) \quad ; \quad S(y) = (1,2,1)^t$$

$$D(x) = (1, 0, -1) \quad ; \quad D(y) = (1, 0, -1)^t$$

Soit :

$$H1 = D(x) * S(y) \quad ; \quad H2 = S(x) * D(y)$$

En effet :

$$\begin{aligned}
I_{D(x)*S(y)}(i,j) &= \sum_{i'=-1}^{i'=1} \sum_{j'=-1}^{j'=1} D(i-i')S(j-j')I(i',j') \\
&= I(i-1,j-1) + 2I(i-1,j) + I(i-1,j+1) \\
&- I(i+1,j-1) - 2I(i+1,j) - I(i+1,j+1) \\
&= I_{H_1}(i,j) \\
I_{S(x)*D(y)}(i,j) &= \sum_{i'=-1}^{i'=1} \sum_{j'=-1}^{j'=1} S(i-i')D(j-j')I(i',j') \\
&= I(i-1,j-1) + 2I(i,j-1) + I(i+1,j-1) \\
&- I(i-1,j+1) - 2I(i,j+1) - I(i+1,j+1) \\
&= I_{H_2}(i,j)
\end{aligned}$$

On a donc réduit le problème de la dérivation et du lissage d'un signal d'une dimension quelconque au cas monodimensionnel. On est donc ramené à la recherche de filtres de lissage 1D.

2.3. Du gradient ou du laplacien vers les points de contours

On remarque que le calcul du gradient ou du laplacien bien que constituant la partie essentielle de la détection des contours ne fournit pas directement les points de contour. En effet dans les deux cas deux étapes supplémentaires sont nécessaires. Nous décrivons brièvement ces traitements **communs** à tous les opérateurs avant d'introduire des opérateurs de filtrage 1D.

Soit $I(x,y)$ une image

Soit $G(x,y)$ le gradient de I au point (x,y) :

$$G(x,y) = (I_x(x,y), I_y(x,y))^t$$

Soit $\Delta(I)$ le laplacien de I au point (x,y)

2.3.1. Approche gradient

Dans le cas d'une approche dérivée première, on dispose donc de la valeur du gradient en tout point de l'image soit de la fonction G. Dans les premières approches [160] l'extraction des points de contour s'effectuait par sélection des points de norme de gradient élevée grâce aux deux étapes suivantes :

1. **calcul de la norme du gradient**

$$N(x, y) = (I_x(x, y)^2 + I_y(x, y)^2)^{1/2}$$

2. **sélection des points de fort gradient**

On détermine les points tels que :

$$N(x, y) > s$$

s : seuil fixé a priori

Dans le cas d'images où la norme du gradient aux points de contour varie fortement selon les parties de l'image cette méthode se révèle inefficace. En effet il n'existe alors pas de seuil s permettant d'obtenir les vrais points de contours sans sélectionner aussi ceux dus au bruit.

Un moyen de tourner cette difficulté est d'extraire non pas les points de norme de gradient élevée mais les extréma locaux de la norme du gradient. Une méthode efficace consiste à déterminer les maxima de la norme du gradient dans la direction du gradient [32]. Ces extréma correspondent aux passages par zéro de la dérivée de la norme du gradient dans la direction du gradient. Dans une deuxième étape on élimine les points de norme de gradient faible avec un seuillage par hystérésis [32]. Ce type de seuillage permet l'obtention de points de contour bien connectés entre eux. Il faut cependant noter qu'il utilise une propriété topologique : la connexité. On obtient donc les deux traitements suivants :

1. **extraction des extréma locaux du gradient**

Soit un point M de gradient $G(\vec{M})$ et d une distance seuil (par exemple $d = 1$). Soient M_1 et M_2 deux points de la droite passant par M et de vecteur directeur $G(\vec{M})$ situés à une distance d de M ; M_1 est pris dans le sens du gradient et M_2 dans le sens inverse.

On détermine une approximation du gradient aux points M_1 et M_2 par exemple par interpolation linéaire avec les points voisins.

Le point M est sélectionné si $N(M) > N(M_2)$ et $N(M) \geq N(M_1)$; le fait d'imposer que le maxima soit strict dans un sens revient à choisir si on localise le point de contour dans la zone de plus faible ou de plus forte valeur de la fonction des niveaux de gris. Cela revient à se déplacer le long de la normale au contour (approximée par le gradient), et à détecter le point de plus fort gradient.

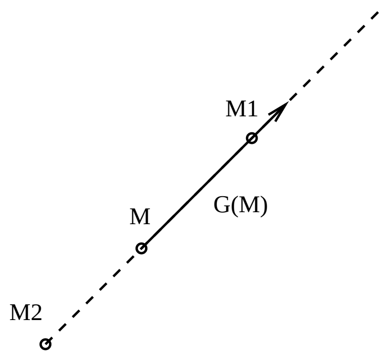


Figure 2.12. *Extraction des extréma locaux : cas 2D.*

On remarquera que les points obtenus vérifient l'équation $\nabla(\|\nabla I\|).\nabla I = 0$.

2. **seuillage par hystérésis des extréma**

Le principe du seuillage par hystérésis est de sélectionner parmi tous les extréma dont la norme du gradient est supérieure à un seuil bas s_b, ceux tels qu'il existe un chemin composé de points dont la norme du gradient est plus élevée que le seuil bas entre l'extremum considéré et un extremum de norme de gradient plus élevé qu'un seuil haut s_h. Le seuil haut s_h et le seuil bas s_b, peuvent par exemple être déterminés à partir de l'histogramme cumulé des valeurs de la norme du gradient aux extréma [129]. L'algorithme se décompose donc en deux étapes :

- détermination de deux images I_h et I_b telles que :

$$\begin{aligned} I_h(M) &= 0 \text{ si } N(M) < s_h \\ I_h(M) &= 1 \text{ si } N(M) \geq s_h \end{aligned}$$

$$
\begin{aligned}
I_b(M) &= 0 \text{ si } N(M) < s_b \\
I_b(M) &= 1 \text{ si } N(M) \geq s_b
\end{aligned}
$$

- expansion en composantes connexes à partir de tous les points tels que $I_h(M) = 1$ sur tous les points tels que $I_b(M) = 1$. Cette étape revient à déterminer le graphe d'adjacence des points de I tels que $I_b(M) = 1$, puis à sélectionner les composantes connexes des nœuds tels que $I_h(M) = 1$.

Le seuillage par hystérésis peut être amélioré en effectuant une expansion en composantes connexes non pas dans toutes les directions, mais simplement dans la direction perpendiculaire au gradient (direction du contour). Ceci permet de diminuer la valeur du seuil bas sans introduire de contours parasites. Cette idée se révèle particulièrement intéressante pour la détection de contours 3D où une expansion dans toutes les directions marque parfois de faux points de contours [129]. Il est important de noter que ce type de traitement introduit de manière implicite des contraintes sur la morphologie des contours, et ne repose pas uniquement sur l'information du "signal image" comme les étapes précédentes.

2.3.2. Approche laplacien

Dans le cas d'une approche dérivée seconde, on dispose donc de la valeur du laplacien en chaque point de l'image soit de la fonction Δ. On considère que les points de contours sont localisés aux passages par zéro du laplacien. Si le calcul du laplacien était exact il suffirait de sélectionner les points M tels que $\Delta(M) = 0$. Mais comme généralement l'approximation du laplacien est assez bruitée, on détecte les points où il change de signe. Une dernière étape de seuillage est là encore nécessaire afin d'éliminer les points de trop faible gradient. L'extraction de ces passages par zéro s'effectue classiquement en trois étapes :

1. **détermination d'une image de polarité**

On calcule une image I_p telle que :

$$
I_p(M) = 0 \text{ si } \Delta(M) > 0
$$

$$I_p(M) \quad = \quad 1 \;\; \text{si} \;\; \Delta(M) \leq 0$$

2. **détection des passages par zéro**
On calcule une image I_z telle que :

$I_z(M) = 1$ si M correspond à une transition 0-1 ou 1-0 dans I_p,

$I_z(M) = 0$ sinon.

On remarque que le choix de la localisation du passage par zéro au point de laplacien positif ou négatif revient, comme pour l'extraction des extréma locaux, à définir les points de contour dans la région la plus claire ou la plus foncée.

3. **seuillage des passages par zéro**
L'élimination des passages par zéro de faible norme de gradient peut s'effectuer par un algorithme de seuillage quelconque. L'algorithme de seuillage par hystérésis décrit pour l'approche dérivée première peut par exemple être utilisé. On peut aussi se servir du fait que les passages par zéro extraits définissent des lignes fermées délimitant les régions de points connexes où le laplacien est positif ou négatif. Des méthodes reposant sur le suivi de ces frontières et sur un calcul local du gradient peuvent aussi être utilisées [165].

Il faut noter que la méthodologie que nous avons développée dans cette partie, tant pour le filtrage que les traitements complémentaires, s'applique à l'extraction de contours dans des images de dimensions quelconques. Cette approche nous a donc permis de ramener ce problème à celui du lissage d'un signal 1D bruité. La partie suivante est donc consacrée au filtrage 1D . Ensuite nous décrirons les algorithmes que nous obtenons dans le cas 2D.

2.4. Mise en oeuvre du filtrage 1D

Nous présentons dans cette partie des critères de performance pour un filtre de lissage 1D, et le filtre de dérivation correspondant. Nous déduisons de ces critères des filtres optimaux. Il faut cependant noter que des liens de causalité existant entre les filtres et les critères d'optimalité qu'ils vérifient peuvent être vus dans les deux sens. Nous montrons enfin que le filtrage récursif amène, pour certains de ces filtres, des implantations de faible complexité algorithmique.

2.4.1. Modèles de contours

Une première étape est de spécifier le type de contours auxquels on s'intéresse ainsi que les caractéristiques du bruit.

- *type de contours*

Plusieurs types de contour existent dans une image. Horn en a définit une classification : marche, arête, toit ...

- *bruit*

Des bruits de caractéristiques diverses sont présents dans les images : bruit convolutif, bruit additif, bruit blanc, bruit impulsionnel ...

- *modèle classique de contours*

Les contours de type "marche" étant très fréquents dans les images et le bruit blanc facile à modéliser, la plupart des méthodes se focalisent sur la détection de contours marche avec un bruit blanc additif. Néanmoins, cette approche s'applique pour d'autres types de contours.

Soit A l'amplitude de la marche et η_0^2 la variance du bruit blanc. Le signal d'entrée $I(x)$ peut être représenté par :

$$I(x) = Au_{-1}(x) + \eta(x)$$

avec

$$\eta_0^2 = E[\eta^2(x)]$$

$$u_{-1}(x) = \begin{cases} 0 & \text{pour x dans}[-\infty, 0] \\ 1 & \text{pour x dans}]0, +\infty] \end{cases}$$

2.4.2. Critères de performances

Il est généralement difficile d'apprécier le résultat d'une détection de contours et ceci pour plusieurs raisons :

- les résultats sont assez souvent estimés à "l'œil", ce qui ne fournit pas de jugement objectif.

- les images traitées sont en général en nombre et en type assez limités.

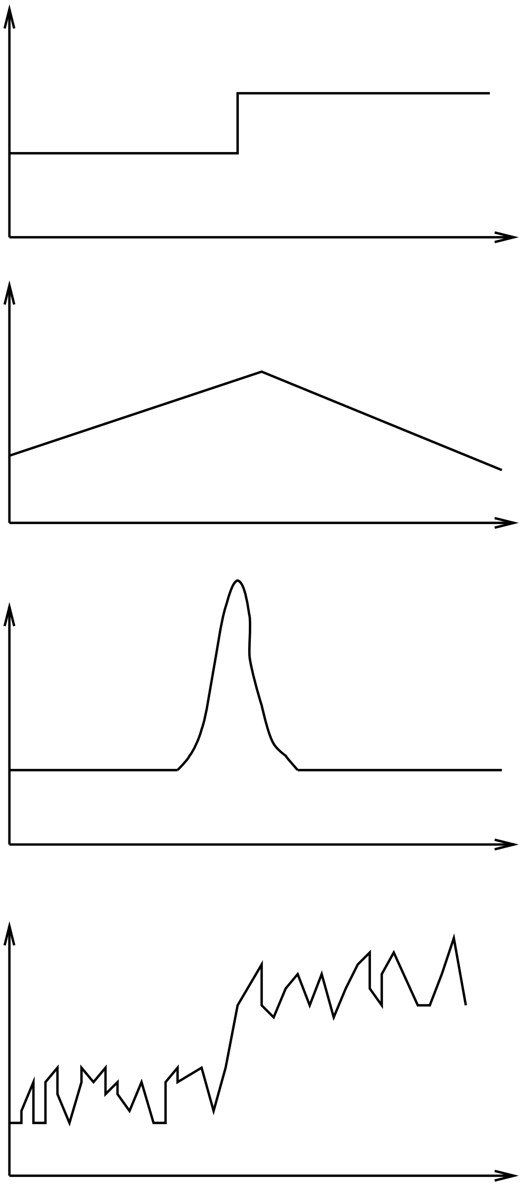


Figure 2.13. *Différents types de contours : contour idéal, contour en toit, contour en pointe, contour idéal bruité.*

Aussi, il est nécessaire de définir au préalable des critères de performance d'un détecteur utilisant évidemment le modèle de contour et le modèle de bruit choisis. Les performances d'un détecteur se caractérisent essentiellement par :

- *détection* : l'opérateur doit donner une réponse au voisinage d'un contour,
- *localisation* : le contour doit être localisé avec précision,
- *réponse unique* : un contour doit provoquer une seule réponse de l'opérateur d'extraction.

On suppose classiquement que la détection est effectuée en convoluant le contour bruité avec une fonction spatiale antisymétrique $f(x)$ et que l'on marque les contours avec des maxima de la sortie $\theta(x_0)$ de cette convolution :

$$\theta(x_0) = \int_{-\infty}^{+\infty} I(x) f(x_0 - x) dx$$

A partir de cette modélisation Canny [32] définit des critères évaluant :

• *la détection des points de contours* : faible probabilité de ne pas détecter un vrai point de contour, et faible probabilité de marquer de faux points de contours ; ce critère correspond à maximiser le rapport signal sur bruit RSB (sortie du filtre au point de discontinuité pour le contour marche / écart type de la réponse du filtre du bruit)

$$RSB = \frac{A}{\eta_0} \frac{\int_{-\infty}^{0} f(x) dx}{\left(\int_{-\infty}^{+\infty} f^2(x) dx\right)^{\frac{1}{2}}} \qquad [2.23]$$

Si on note

$$\sigma = \frac{\int_{-\infty}^{0} f(x) dx}{\left(\int_{-\infty}^{+\infty} f^2(x) dx\right)^{\frac{1}{2}}} \qquad [2.24]$$

on obtient :

$$RSB = \frac{A}{\eta_0} \sigma \qquad [2.25]$$

• *la localisation des points de contours* : les points marqués comme contours par le détecteur doivent être aussi près que possible du centre du contour véritable. Ce critère correspond à maximiser l'écart type de la position des passages par zéro (voir 2.6. pour le calcul détaillé), et correspond à l'inverse de l'espérance de la distance entre le vrai point de contour, et le point de contour détecté (maximum de la sortie de l'opérateur).

$$L = \frac{A}{\eta_0} \frac{\mid f'(0) \mid}{(\int_{-\infty}^{+\infty} f'^2(x)dx)^{\frac{1}{2}}} \qquad [2.26]$$

Si on note

$$\lambda = \frac{\mid f'(0) \mid}{(\int_{-\infty}^{+\infty} f'^2(x)dx)^{\frac{1}{2}}} \qquad [2.27]$$

on obtient :

$$L = \frac{A}{\eta_0}\lambda \qquad [2.28]$$

• une réponse à un contour : le détecteur ne doit pas fournir de multiples réponses à un seul contour ; la distance moyenne entre les pics dans la réponse au bruit de f, noté $xmax$, est posée comme une fraction k de la largeur de l'opérateur W (la largeur d'un opérateur est la taille du masque de convolution nécessaire pour l'implanter). Cette expression ($xmax$) se calcule aisément car on connait la distance moyenne entre les passages par zéro de la dérivée seconde de la réponse d'un filtre à un bruit blanc gaussien (Rice 1944).

$$x_{max} = kW = 2\Pi \left(\frac{\int_{-\infty}^{+\infty} f'^2(x)dx}{\int_{-\infty}^{+\infty} f''^2(x)dx} \right)^{\frac{1}{2}} \qquad [2.29]$$

On notera que l'introduction de ce troisième critère est dû au fait que le critère de détection ne prend en compte que la réponse du filtre au point de contour et non dans son voisinage. Ce critère est le seul utilisant l'hypothèse d'un bruit gaussien.

Les critères de détection et de localisation étant antinomiques, on les combine d'une manière significative en maximisant le produit $\sigma\lambda$ sous la contrainte du 3ème critère. (d'autres solutions sont envisageables, par exemple maximiser le produit des trois termes...). On obtient ainsi une équation différentielle dont la solution est :

$$\begin{aligned} f(x) &= a_1 e^{\alpha x} \sin \omega x + a_2 e^{\alpha x} \cos \omega x \\ &+ a_3 e^{-\alpha x} \sin \omega x + a_4 e^{-\alpha x} \cos \omega x + c \end{aligned} \qquad [2.30]$$

avec les conditions initiales :

$$f(0) = 0 \quad f(W) = 0 \quad f'(0) = S \quad f'(-W) = 0$$

W étant la taille du filtre défini dans $[0, W]$.
Si on fait tendre W vers $+\infty$ on obtient

$$a_1 = a_2 = a_3 = c = 0$$

Soit le filtre optimal devient :

$$f(x) = a_3 e^{\alpha x} \sin wx$$

Le filtre étant impair on obtient pour le filtre de dérivation optimal :

$$f(x) = a_3 e^{\alpha |x|} \sin wx \qquad [2.31]$$

α définit la largeur du filtre soit le compromis détection-localisation désiré (la localisation est une fonction croissante de α et la détection une fonction décroissante de α).

Shen et Castan [165] proposent de déterminer les opérateurs de lissage optimisant un critère incluant la détection et la localisation. Les critères qu'ils obtiennent correspondent aux critères de détection et de localisation de Canny et les filtres obtenus assez similaires dans la pratique.

2.4.3. Filtres optimaux de lissage/dérivation

Les filtres de dérivation correspondant aux filtres de lissage sont obtenus simplement par dérivation des réponses impulsionnelles et renormalisation. Il est important de remarquer que l'utilisation de filtres de dérivation pour la détection des contours revient à supposer que les contours correspondent aux discontinuités d'ordre 0. Si on s'attachait à la recherche des contours en toit, il faudrait mettre en œuvre des opérateurs de détection des discontinuités d'ordre 1 (celà revient à déterminer la courbure du signal).

1. **filtre Shen-Castan**

Le filtre de lissage Shen-Castan (voir figure 2.14) s'écrit :

$$s_1(x) = ce^{-\alpha.|x|} \qquad [2.32]$$

c est choisi de manière à obtenir un filtre normalisé (dans l'espace discret) soit :

$$\sum_{-\infty}^{+\infty} s_1(n) = 1$$

d'où :

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

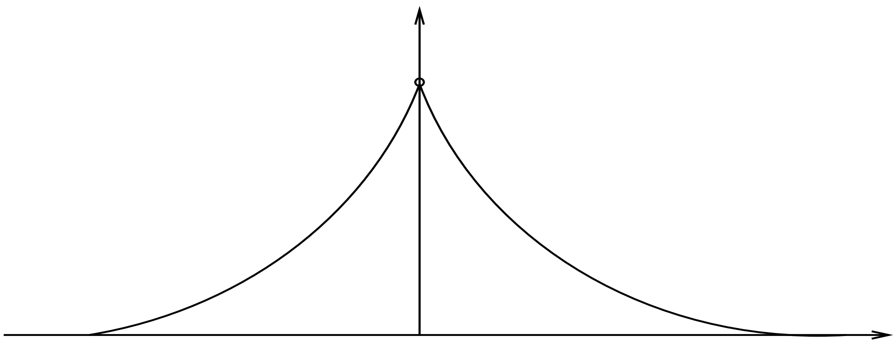


Figure 2.14. *Réponse impulsionnelle de s_1.*

le filtre de dérivation correspondant (d_1) est obtenu par normalisation de la dérivée du filtre de lissage.

$$d_1(x) = \begin{cases} d.e^{-\alpha|x|} \text{ si } x \geq 0 \\ -d.e^{-\alpha|x|} \text{ si } x \leq 0 \end{cases}$$

d est choisi de manière à obtenir un filtre normalisé dans l'espace discret soit :

$$\sum_{0}^{+\infty} d_1(n) = 1$$

d'où

$$d = 1 - e^{-\alpha}$$

Le paramètre α définit la "largeur" du filtre ; plus α est petit, plus le lissage effectué par le filtre est important.

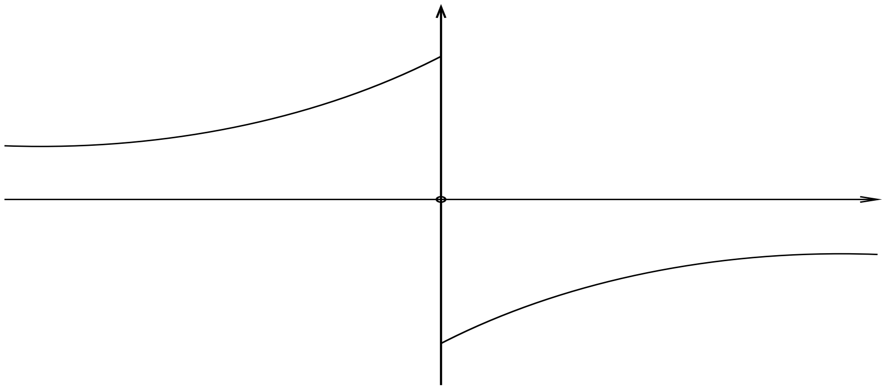


Figure 2.15. *Réponse impulsionnelle de d_1.*

Ce filtre a été introduit par Shen et Castan pour l'approximation du laplacien par différence entre l'image originale et l'image lissée [165]. Le filtre obtenu par dérivation constitue une solution optimale pour la première partie des critères de Canny (voir équations [2.23] et [2.26]). Il correspond à une valeur optimale pour le produit $\sigma\lambda$, soit au meilleur compromis détection-localisation. La discontinuité d'ordre 1 au point 0 permet d'éviter une délocalisation importante des contours dans l'image lissée même avec des valeurs faibles de α. Cependant cette discontinuité peut entrainer la détection de contours multiples [32].

2. **filtre moyenne** Le filtre moyenne s'écrit :

$$s_2(x) = \begin{cases} \frac{1}{(2n+1)} & \text{si} \quad x \quad \text{dans} \quad [-n, n] \\ 0 & \text{sinon} \end{cases}$$

Dans ce cas, le filtre dérivé d_2 est approximé par :

$$d_2(x) = \begin{cases} \frac{-1}{(2n+1)} \text{pour} \quad x \quad dans \quad [n, 0] \\ \frac{1}{(2n+1)} \quad \text{pour} \quad x \quad \text{dans} \quad]0, n] \\ 0 \quad \text{sinon} \end{cases} \qquad [2.33]$$

Ce filtre est le plus simple à mettre en œuvre et est utilisé depuis longtemps. Il peut par exemple être mis en œuvre pour approximer le laplacien par différence entre les images obtenues par lissage avec deux filtres de largeurs différentes [104]. Il est équivalent au filtre s_1 lorsque α tend vers zéro. Il est

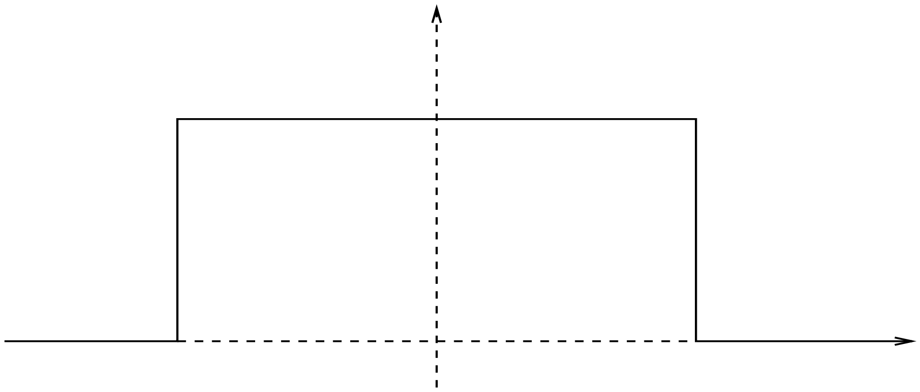


Figure 2.16. *Réponse impulsionnelle de* s_2.

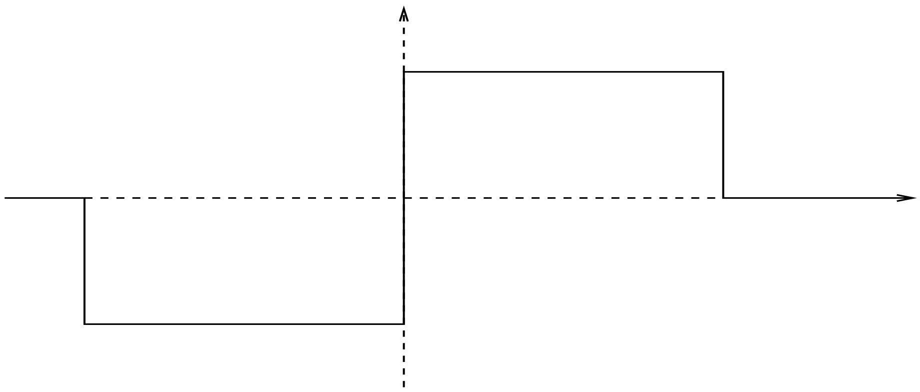


Figure 2.17. *Réponse impulsionnelle de* d_2.

recommandé de l'utiliser avec des valeurs du paramètre α faibles par rapport à la distance minimale entre deux contours, car sinon la dégradation des contours dans l'image lissée est importante.

3. **filtre gaussien** Le filtre gaussien s'écrit :

$$s_3(x) = ce^{-\frac{x^2}{2\sigma^2}}$$

c est choisi de manière à obtenir un filtre normalisé.
Le filtre de dérivation correspondant s'écrit :

$$d_3(x) = c\frac{-x}{\sigma^2}e^{-\frac{x^2}{2\sigma^2}} \qquad [2.34]$$

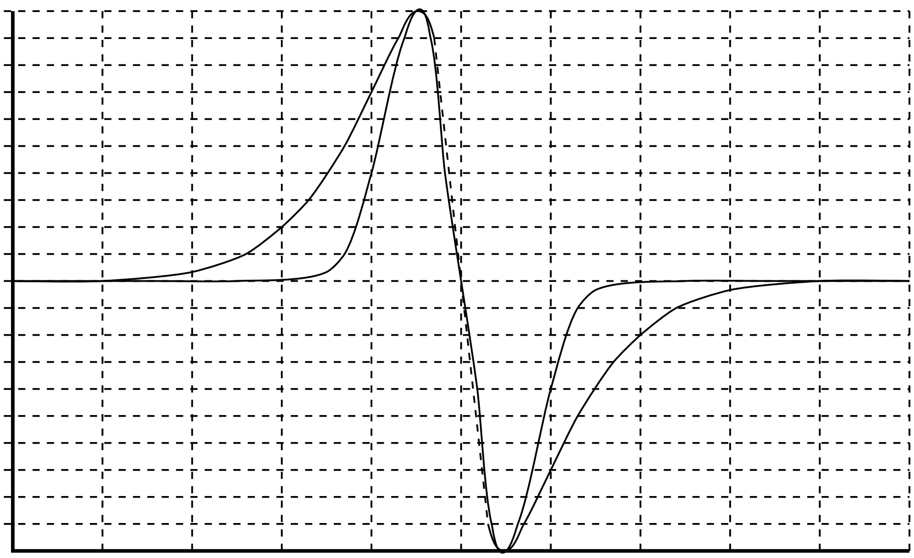


Figure 2.18. *Réponses impulsionnelles de d_3 et de d_4 ($d_4 \leq d_3$).*

Ce filtre a été introduit pour la détection de contours par Marr et Hildreth pour le calcul du laplacien [122]. Il correspond à la valeur optimale pour le produit de la variance temporelle et de la variance spectrale. Le filtre de dérivation correspondant $d_3(x)$ est une solution approchée à l'équation de Canny. En effet par optimisation numérique Canny trouve que la plus grande valeur de k qui peut être obtenue est 0.58 et que sa performance est donnée par $\sigma\lambda = 1.12$ (voir équations [2.23], [2.26], [2.29]). Il propose l'utilisation de la première dérivée d'une gaussienne pour laquelle : $\sigma\lambda = 0.92$ et $k = 0.51$ (voir équation [2.34]).

4. **filtre Deriche** Le filtre Deriche s'écrit :

$$s_4(x) = k(\alpha \mid x \mid +1)e^{-\alpha|x|}$$

k est choisi de manière à obtenir un filtre discret normalisé soit :

$$\sum_{-\infty}^{+\infty} s_4(x) = 1 \Longleftrightarrow k = \frac{(1-e^{-\alpha})^2}{1+2\alpha e^{-\alpha} - e^{-2\alpha}}$$

Ce filtre a été proposé récemment par R. Deriche [43] et sa dérivée est une solution exacte à l'équation de Canny étendue à des filtres infinis. Il est obtenu en étendant l'équation différentielle [2.30] aux opérateurs infinis et antisymétriques et constitue ainsi une solution exacte à l'équation de Canny.

En effet, soit $f(x)$ le filtre solution de l'équation de Canny étendue aux filtres à réponses infinies :

$$f(x) = -de^{-\alpha|x|} \sin wx$$

α, ω, c réels positifs.

Si on pose :

$$\alpha = m\omega$$

on obtient :

(a) $m >> 1$ $\quad \lambda = (2\alpha)^{\frac{1}{2}}$ $\quad \sigma \approx (\frac{2}{\alpha})^{\frac{1}{2}}$ $\quad \sigma\lambda \approx 2$ $\quad k \approx .44$

(b) $m << 1$ $\quad \lambda = (2\alpha)^{\frac{1}{2}}$ $\quad \sigma \approx \frac{\lambda}{m}$ $\quad \sigma\lambda \approx 2m$ $\quad k \approx 1$

(c) $m = 1$ $\quad \lambda = (2\alpha)^{\frac{1}{2}}$ $\quad \sigma = (\frac{1}{\alpha})^{\frac{1}{2}}$ $\quad \sigma\lambda = 1.414$ $\quad k = .58$

(d) $m = (3)^{\frac{1}{2}}$ $\quad \lambda = (2\alpha)^{\frac{1}{2}}$ $\quad \sigma = (\frac{3}{2\alpha})^{\frac{1}{2}}$ $\quad \sigma\lambda = 1.732$ $\quad k = .5$

Le cas (d) montre que pour une même valeur de k la performance de la première dérivée d'une gaussienne est moins bonne que celle de l'opérateur f de plus de 90 %.

Le cas (c) montre que la forme finale utilisée par Canny pour son opérateur optimal (la dérivée d'une gaussienne, $\sigma\lambda = 1.112, k = .58$) est moins bonne que l'opérateur f de plus de 25 %.

Le cas (b) présente une réponse idéale pour k mais le produit $\sigma\lambda$ devrait être beaucoup plus petit que l'unité.
Le cas (a) présente le meilleur compromis. On peut alors écrire ce filtre optimal de dérivation comme :

$$\begin{aligned} d_4(x) &= -kxe^{-\alpha|x|} \\ k &= \frac{(1-e^{-\alpha})^2}{e^{-\alpha}} \end{aligned}$$

Le filtre de lissage s'obtient par intégration du filtre de dérivation.

Les filtres de dérivée seconde s'obtiennent en dérivant deux fois les filtres de lissage correspondants (ou une fois les filtres de dérivation). Ils permettent d'approximer la dérivée seconde du signal. Nous remarquerons que pour le filtre moyenne et le filtre Shen il est nécessaire d'effectuer une approximation à cause de la discontinuité au point 0. De toute manière, même s'il n'existe pas d'expression analytique des réponses impulsionnelles, il est toujours possible de les implanter par cascade de deux filtres de dérivation. Les filtres de dérivée seconde et troisième de Deriche sont décrits dans le chapitre “Reconstruction et modélisation de surfaces” (partie : “Des dérivées partielles des images 3D aux courbures des surfaces”).

2.4.4. Récursivité et implantation des filtres

La détermination de filtres ayant des propriétés intéressantes pour la détection de contours répond à la question : Que calculer (voir introduction). Maintenant il nous faut trouver avec quels algorithmes. En d'autres termes, comment implanter efficacement la convolution d'un signal 1D par une réponse impulsionnelle infinie.

L'implantation classique consiste à tronquer les réponses de manière à ne conserver que les cœfficients significatifs (les filtres de lissage et de dérivation première ou seconde considérés décroissent vers 0 à l'infini). Ensuite on procède par implantation directe d'un produit de convolution. On obtient ainsi par exemple pour une gaussienne d'écart type σ un masque de convolution de taille 8σ, et pour un filtre Shen ou un filtre Deriche de paramètre α un masque de taille $\frac{8}{\alpha}$. Pour des étendues de filtres assez importantes on obtient donc des complexités prohibitives et ceci d'autant plus que la dimension de l'image est

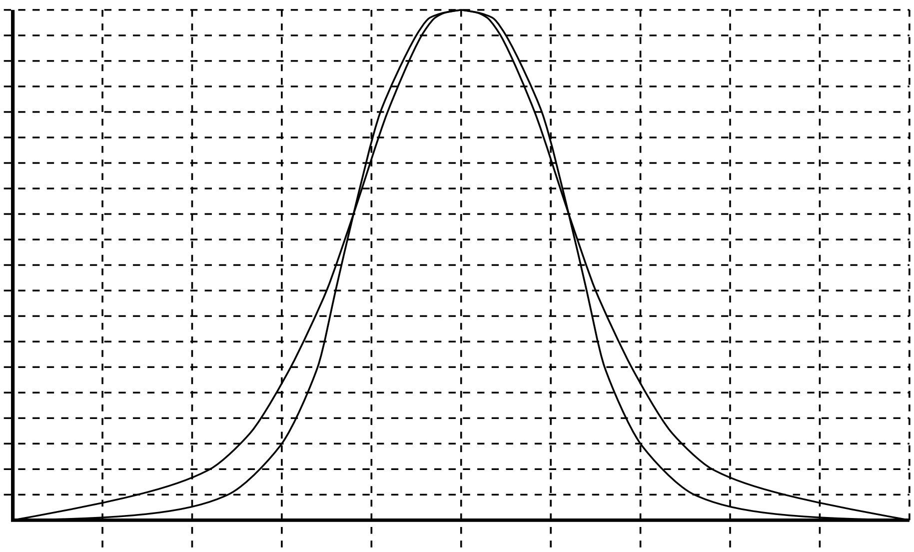


Figure 2.19. *Réponses impulsionnelles de s_3 et de s_4 ($s_4 \leq s_3$).*

importante. De plus la troncature des réponses introduit un effet "cut off" qui nuit à la qualité des contours détectés.

Une solution pour éviter ce compromis entre la taille des masques et la qualité des résultats consiste à utiliser le filtrage récursif. Le filtrage récursif est une technique bien connue en traitement du signal et a été introduite récemment en vision artificielle pour la détection des contours [43, 165]. elle permet d'implanter des filtres de réponse impulsionnelle infinie avec un coût faible. Il se trouve que le filtre Shen et le filtre Deriche admettent des réalisations récursives respectivement d'ordres 1 et 2. Le filtre gaussien peut être approximé par des filtres récursifs d'ordre 4 [42]. Dans cette partie nous montrerons comment obtenir la mise en œuvre récursive des filtres de Canny, Deriche et Shen.

Nous allons montrer, par exemple, que le filtre optimal de Canny ($f(x)$) se réalise par un filtrage récursif d'ordre 2 [43].

$$f(x) = -ce^{-\alpha|x|} \sin wx$$

Soit $f(n)$ la fonction d'échantillonnage de $f(x)$ et $F(Z)$ sa transformée en

Z :

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

On réécrit $f(n)$ comme :

$$\begin{aligned}
f(n) &= f_{-}(n) + f_{+}(n) \quad \text{où} \\
f_{-}(n) &= \begin{cases} 0 & n \geq 0 \\ \alpha_1 (p_2)^n + \alpha_1^* (p_2^*)^n & n < 0 \end{cases} \\
f_{+}(n) &= \begin{cases} \alpha_1 (p_1)^n + \alpha_2^* (p_1^*)^n & n \geq 0 \\ 0 & n < 0 \end{cases} \\
\alpha_1 &= \tfrac{-c}{2ip_1} = e^{-\alpha+i\omega} \qquad p_2 = e^{\alpha+i\omega} \\
\alpha^* &: \ \text{conjugué de} \ \ \alpha
\end{aligned}$$

Les équations précédentes reviennent à séparer la réponse en une partie positive (f_{+}) et une partie négative (f_{-}). En utilisant la transformée en Z, on reconnaît que chaque facteur $\alpha_i (p_i)^n$ a une transformée en Z égale à $\frac{\alpha_i}{(1-p_i Z^{-1})}$ et on obtient :

$$\begin{aligned}
F(Z) &= F_{-}(Z) + F_{+}(Z^{-1}) \quad \text{où} \\
F_{+}(Z^{-1}) &= \frac{aZ^{-1}}{(1 + b_1 Z^{-1} + b_2 \times Z^{-2})} \\
F_{-}(Z) &= \frac{-aZ}{(1 + b_1 Z + b_2 Z^2)}
\end{aligned}$$

avec :

$$\begin{aligned}
a &= -c.e^{-\alpha} \sin \omega \\
b1 &= -2.e^{-\alpha} \cos \omega \\
b2 &= e^{-2\alpha}
\end{aligned}$$

Toutes les singularités de $F_{+}(Z^{-1})$ (resp. $F_{-}(Z)$) sont à l'intérieur (resp. à l'extérieur) du cercle unité pour une valeur de α positive, ces deux transformées en Z correspondent donc à deux fonctions de transfert rationnelles de filtres stables du second ordre récursifs de la gauche vers la droite (F_{+}) et de la droite vers la gauche (F_{-}).

En particulier la séquence de sortie $y(m)$ en réponse à l'entrée $x(m)$ pour un système de réponse impulsionnelle $f(n)$ peut être obtenue récursivement de la manière suivante :

$$\begin{aligned} y^+(m) &= ax(m-1) - b_1 y^+(m-1) - b_2 y^+(m-2) \qquad m = 1, \ldots, N \\ y^-(m) &= -ax(m+1) - b_1 y^-(m+1) - b_2 y^-(m+2) \quad m = N, \ldots, 1 \\ y(m) &= y^+(m) + y^-(m) \qquad m = 1, \ldots, N \end{aligned}$$

Les cœfficients de la récursion se déduisent de la transformée en Z de la manière suivante :

$$F(Z) = \frac{a_0 + a_1 Z^{-1} + \ldots\ldots a_{n-1} Z^{-(n-1)}}{1 + b_1 Z^{-1} + \ldots\ldots b_n Z^{-n}} \text{ si } |z| > 1$$

caractérise un système récursif causal d'ordre n de réalisation :

$$y(i) = \sum_{l=0}^{l=n-1} a_l x(i-l) - \sum_{k=1}^{k=n} b_k y(i-k)$$

$$F(Z) = \frac{a_0 + a_1 Z^{1} + \ldots\ldots a_{n-1} Z^{(n-1)}}{1 + b_1 Z^{1} + \ldots\ldots b_n Z^{n}} \text{ si } |z| < 1$$

caractérise un système récursif anticausal d'ordre n de réalisation

$$y(i) = \sum_{l=0}^{l=n-1} a_l x(i+l) - \sum_{k=1}^{k=n} b_k y(i+k)$$

La récursivité du filtre de Canny permet une réalisation qui requiert seulement 6 multiplications et 5 additions par point, et cela avec un paramètre de filtre (α) quelconque.

En appliquant la même technique on peut montrer que l'échantillonnage h de l'intégrale $h(x)$ du filtre précédent $f(x)$ s'implante aussi récursivement .

En effectuant le même type de calcul on montre que les filtres de lissage et de dérivation de Shen s'implantent par filtrage récursif d'ordre 1. s_1 (filtre de lissage Shen) est un filtre récursif du premier ordre de réalisation [165] :

$$s_1(x) = ce^{-\alpha . |x|}$$

$$\begin{aligned}
y^+(m) &= ax(m) + by^+(m-1) \\
&\quad \text{pour } m = 1, \ldots, N \\
y^-(m) &= abx(m+1) + by^-(m+1) \\
&\quad \text{pour } m = N, \ldots, 1 \\
y(m) &= y^-(m) + y^+(m) \\
&\quad \text{pour } m = 1, \ldots, N
\end{aligned}$$

avec

$$a = c = \frac{1 - e^{-\alpha}}{1 + e^{-\alpha}} \; ; \; b = e^{-\alpha}$$

d_1 est un filtre récursif du premier ordre de réalisation :

$$d_1(x) = \begin{cases} d.e^{-\alpha|x|} & \text{si } x \geq 0 \\ -d.e^{-\alpha|x|} & \text{si } x \leq 0 \end{cases}$$

$$\begin{aligned}
y^+(m) = &\quad dx(m) + ey^+(m-1) && \text{pour } m = 1, \ldots, N \\
y^-(m) = &\quad -dx(m) + ey^-(m+1) && \text{pour } m = N, \ldots, 1 \\
y(m) = &\quad y^-(m) + y^+(m) && \text{pour } m = 1, \ldots, N
\end{aligned}$$

avec

$$d = e = 1 - e^{-\alpha} \; ; \; f = e^{-\alpha}$$

s_4 (filtre de lissage Deriche) est un filtre récursif du second ordre de réalisation :

$$s_4(x) = k(\alpha \mid x \mid +1)e^{-\alpha|x|}$$

$$\begin{aligned}
y^+(m) &= a_0x(m) + a_1x(m-1) - b_1y^+(m-1) - b_2y^+(m-2) \\
&\quad \text{pour } m = 1, \ldots, N \\
y^-(m) &= a_2x(m+1) + a_3x(m+2) - b_1y^-(m+1) - b_2y^-(m+2) \\
&\quad \text{pour } m = N, \ldots, 1 \\
y(m) &= y^-(m) + y^+(m) \\
&\quad \text{pour } m = 1, \ldots, N
\end{aligned}$$

avec

$$a_0 = k \ ; \ a_1 = k(\alpha - 1)e^{-\alpha} \ ; \ a_2 = k(\alpha + 1)e^{-\alpha} \ ; \ a_3 = -ke^{-2\alpha} \ ;$$

$$b_1 = -2e^{-\alpha} \ ; \ b_2 = e^{-2\alpha} \ ; \ k = \frac{(1 - e^{-\alpha})^2}{1 + 2\alpha e^{-\alpha} - e^{-2\alpha}}$$

d_4 (filtre de dérivation Deriche) est un filtre récursif du second ordre de réalisation :

$$d_4(x) = -cxe^{-\alpha|x|}$$

$$\begin{array}{rcl} y^+(m) = & ax(m-1) - b_1 y^+(m-1) - b_2 y^+(m-2) & \text{pour } m = 1, \ldots, N \\ y^-(m) = & -ax(m+1) - b_1 y^-(m+1) - b_2 y^-(m+2) & \text{pour } m = N, \ldots, 1 \\ y(m) = & y^-(m) + y^+(m) & \text{pour } m = 1, \ldots, N \end{array}$$

avec

$$a = ce^{-\alpha} \ ; \ b_1 = -2e^{-\alpha} \ ; \ b_2 = e^{-2\alpha}$$

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

l_4 (filtre dérivée seconde Deriche) est un filtre récursif du second ordre de réalisation :

$$l_4(x) = c_2(1 - c_3\alpha \mid x \mid)e^{-\alpha|x|}$$

$$\begin{array}{rcl} y^+(m) & = & a_0 x(m) + a_1 x(m-1) - b_1 y^+(m-1) - b_2 y^+(m-2) \\ & & \text{pour } m = 1, \ldots, N \\ y^-(m) & = & a_2 x(m+1) + a_3 x(m+2) - b_1 y^-(m+1) - b_2 y^-(m+2) \\ & & \text{pour } m = N, \ldots, 1 \\ y(m) & = & y^-(m) + y^+(m) \\ & & \text{pour } m = 1, \ldots, N \end{array}$$

avec

$$a_0 = -1 \ ; \ a_1 = (1 + k\alpha)e^{-\alpha} \ ; \ a_2 = -(1 - k\alpha)e^{-\alpha} \ ; \ a_3 = -e^{-2\alpha} \ ;$$

$$b_1 = -2e^{-\alpha} \ ; \ b_2 = e^{-2\alpha} \ ; \ k = \frac{1 - e^{-2\alpha}}{2\alpha e^{-\alpha}}$$

2.5. Résumé : algorithmes de calcul du gradient et du laplacien

Nous présentons dans cette section les différentes étapes des algorithmes de détection de contours issus de la méthodologie précédemment décrite.

1. **choix d'un filtre de lissage 1D :** $s(x)$

On recommande de choisir un filtre qui s'implante récursivement soit par exemple :

$$s(x) = s_4(x) = k(\alpha \mid x \mid +1)e^{-\alpha|x|} \quad \text{filtre de lissage Deriche}$$

où

$$s(x) = s_1(x) = ce^{-\alpha|x|} \quad \text{filtre de lissage Shen}$$

Théoriquement le filtre de dérivation $s_4'(x)$ est meilleur par rapport au critère de réponse multiple, mais $s_1'(x)$ présente le meilleur compromis détection-localisation. Pour des α petits la forme de $s_4'(x)$ induit des problèmes de délocalisation des contours comme pour la plupart des filtres continus au point 0 [160]. Néanmoins, sur une grande variété d'images les résultats obtenus sont comparables. On notera que les deux filtres ne sont pas de la même espèce (discontinu ou continu au point 0) donc ne sont pas théoriquement comparables.

2. **choix du type d'approche : gradient, laplacien** En général, on constate que le calcul de la dérivée seconde est plus sensible au bruit. D'un autre côté, la complexité du calcul du gradient est généralement plus élevée que pour le laplacien. Ceci est dû aux simplifications qui apparaissent lors du calcul de la réponse du filtre d'approximation du laplacien par multiplication et addition de filtres de lissage et de dérivée seconde. Il faut cependant noter que le plus souvent une étape de seuillage des passages par zéro du laplacien est utile, et qu'elle nécessite le calcul de la norme du gradient (aux points de passage par zéro). La localisation des contours obtenus est pratiquement la même. On remarque que l'approche laplacien a tendance à arrondir les angles tout en maintenant une localisation exacte pour les coins. On choisira donc plutôt une approche gradient si des angles aigus sont présents dans les images (scènes d'intérieur par exemple), et plutôt une approche laplacien dans le cas de contours à faible variation de courbure.

3. **détermination d'une cascade de filtres pour approximer le gradient ou le laplacien**

Une fois que l'on a choisi le type d'approche : gradient ou laplacien, il nous faut déterminer un moyen de les calculer.

- **calcul du gradient**

  Détermination d'une implantation des filtres 1D de réponses : $s(x)$ (lissage) et $d(x)$ (dérivation). On choisira de préférence des implantations récursives. Des exemples sont donnés dans la section précédente.

  Soit $I(x, y)$ une image,

  Soit $G(x, y)$ le gradient de I

$$G(x,y) = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})^t = (I_x, I_y)^t$$

  Le calcul des composantes du gradient $\frac{dI}{dx_i}$ s'effectue en calculant les images correspondant aux dérivées partielles des points par rapport à x et à y par filtrage séparable récursif :

$$I_x(x,y) = (I * s(y)) * d(x)$$

$$I_y(x,y) = (I * s(x)) * d(y)$$

  Pour une image de dimension d^2 le calcul de chaque composante du gradient nécessite donc le calcul de 2 convolutions par point ; on obtient pour le calcul total du gradient $2d^2$ convolutions par point. Si on utilise l'implantation directe d'un masque de convolution (1D) de taille k on obtient une complexité de $2d^2k$. Le filtrage récursif d'ordre r permet d'obtenir une complexité de l'ordre de $2d^2r$. Par exemple pour le filtre Deriche on obtient 13 multiplications et 12 additions par points et ceci quelle que soit la valeur de α. Si on utilise un filtre récursif le calcul de $I * s(x)$ s'effectue en filtrant successivement les droites d'équation $y = cste$.

- **calcul du laplacien**

  On peut le calculer en déterminant d'abord les dérivées secondes puis en les additionnant. Le calcul des dérivées secondes s'effectue avec la structure algorithmique précédente dans laquelle on remplace l'opérateur de dérivée première $d(x)$ par l'opérateur de dérivée seconde $l(x)$. Cependant, dans certains cas, la réponse impulsionnelle du filtre permettant d'approximer le laplacien L se simplifie et se réécrit sous une forme séparable. On obtient alors :

$$L(x,y) = s(x)s''(y) + s''(x)s(y)$$

4. **extraction des extréma locaux du gradient ou des passages par zéro du laplacien**

Bien que ne constituant pas la partie essentielle de la détection des contours cette étape doit être effectuée avec attention car elle conditionne fortement le résultat final. La méthode proposée dans la section précédente permet d'obtenir des résultats satisfaisants pour des images 2D [43] ou 3D [128].

5. **seuillage des extréma locaux du gradient ou des passages par zéro du laplacien**

Le plus souvent les filtres utilisés ne suffisent pas à éliminer complètement le bruit présent dans les images. On obtient alors des extréma locaux ou des passages par zéro du laplacien qui ne correspondent pas à de vrais points de contour. Fréquemment ces points parasites ont une norme de gradient faible. Ainsi un seuillage selon une estimation de la norme du gradient permet le plus souvent d'éliminer la plupart de ces faux points de contour. L'algorithme de seuillage par hystérésis décrit précédemment semble être une bonne réponse à ce problème de seuillage.

2.6. Analyse des performances des filtres

On traitera dans cette section le cas 3D afin de montrer l'applicabilité à une dimension quelconque des développements précédents et d'expliciter dans un cas particulier le calcul des critères de Canny. On remarquera que ces calculs sont réalisés avec des filtres normalisés dans le domaine continu.

2.6.1. Introduction

Le filtrage séparable permet de dériver des opérateurs d'approximation du gradient ou du laplacien 2D ou 3D (mêmes développements que précédemment en rajoutant une dimension) à partir d'opérateurs 1D optimaux au sens des critères de Canny (voir partie 1.2.5). Dans cette partie nous nous intéressons au calcul des performances des filtres 2D ou 3D correspondant pour le filtre

Deriche (voir partie 1.2.5). Nous verrons que l'anisotropie des réponses impulsionnelles obtenues nécessite la prise en compte de l'orientation du contour pour le calcul des performances. Nous remarquons que dans le cas du filtre gaussien, l'isotropie de la réponse impulsionnelle fait que les performances du filtre 1D sont les performances des filtres 2D, 3D à une constante multiplicative près.

2.6.2. Modèle de contour 2D

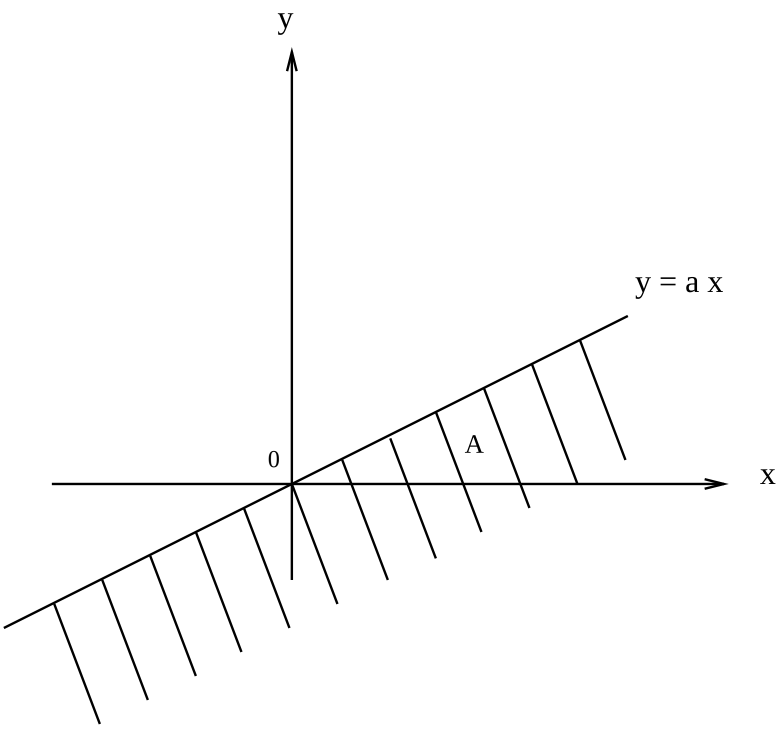


Figure 2.20. *Contour idéal 2D.*

Un contour parfait est défini par une droite séparant l'espace en 2 régions homogènes (voir figure 2.20.) :

$$H(x, y) = Au_{-1}(x, y)$$

avec

$$u_{-1}(x,y) = \begin{cases} 0 & \text{si} \quad y > ax \\ 1 & \text{si} \quad y \leq ax \end{cases}$$

avec $a \geq 0$

Un contour bruité est défini par :

$$C(x,y) = H(x,y) + \eta(x,y)$$

où $\eta(x,y)$ représente le bruit.

On suppose que le bruit est décorrélé en x, y soit :

$$\eta(x,y) = \eta_1(x) + \eta_2(y)$$

où η_1 et η_2 sont des bruits blancs gaussiens indépendants de moyenne nulle et de variance η_0^2. On supposera dans la suite par commodité que les variances de η_1 et η_2 sont égales.

2.6.3. Modèle de contour 3D

Dans le cas 3D le contour parfait est défini par un plan divisant l'espace en deux régions de niveaux de gris uniformes (voir figure 2.6.3.) :

$$H(x,y,z) = Au_{-1}(x,y,z)$$

avec

$$u_{-1}(x,y,z) = \begin{cases} 0 & \text{si} \quad z > ax + by, \\ 1 & \text{si} \quad z \leq ax + by \end{cases}$$

avec $a, b \geq 0$ Un contour bruité est défini par :

$$C(x,y,z) = H(x,y,z) + \eta(x,y,z)$$

avec :

$$\eta(x,y,z) = \eta_1(x) + \eta_2(y) + \eta_3(z)$$

où les η_i sont des bruits blancs gaussiens indépendants de variances η_0^2.

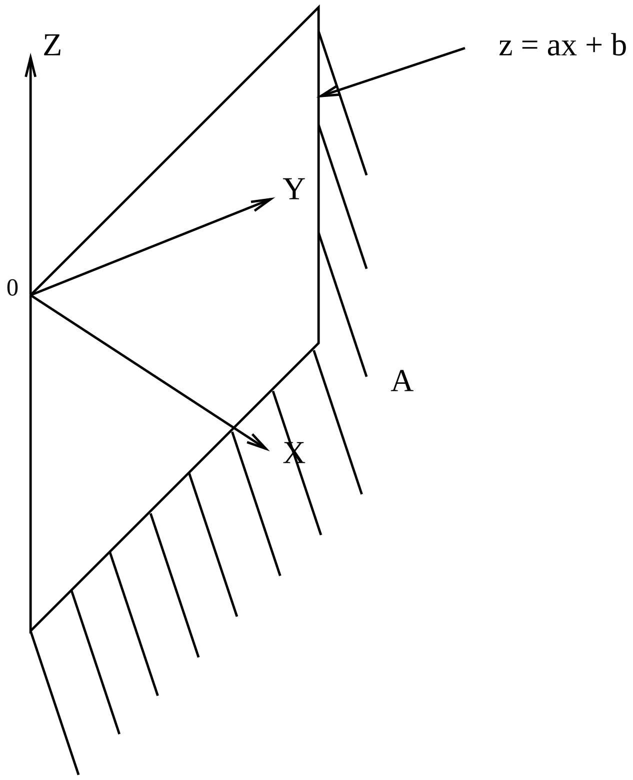


Figure 2.21. *Contour idéal 3D.*

2.6.4. Performances du filtre Deriche 2D, 3D

Dans cette partie on calcule les performances des filtres 2D et 3D de Deriche en fonction de l'orientation du contour. On commencera par le calcul dans le cas 3D, le cas 2D étant déduit directement.

2.6.5. Opérateurs

$$\begin{aligned}
s_4(x) &= k(\alpha \mid x \mid +1)e^{-\alpha|x|} \text{ : composante de lissage} \\
d_4(x) &= -cxe^{-\alpha|x|} \text{ : composante de dérivée première} \\
l_4(x) &= f(1-\alpha \mid x \mid)e^{-\alpha|x|} \text{ : composante de dérivée seconde} \\
D(x,y,z) &= l_4(x)s_4(y)s_4(z)s_4(x)l_4(y)s_4(z) + \\
& \quad s_4(x)s_4(y)l_4(z) \text{ : filtre de calcul du laplacien}
\end{aligned} \quad [2.35]$$

et:

$$\begin{aligned}
g_1(x,y,z) &= d_4(x)s_4(y)s_4(z) \\
g_2(x,y,z) &= s_4(x)d_4(y)s_4(z) \\
g_3(x,y,z) &= s_4(x)s_4(y)d_4(z)
\end{aligned}$$

sont les filtres de calcul des dérivées par rapport à x, y, z. Pour les calculs suivants, les constantes de normalisation k, c, f sont déterminées dans l'espace continu.

2.6.6. Performances

Détection pour l'approche gradient

De manière à déterminer RSB, on calcule la réponse du filtre en un point du contour pour le contour idéal $(H(x,y,z))$ soit :

$$\begin{aligned}
G_{x_0} &= I_x(x_0,y_0,z_0) = [H * g_1]_0 \\
&= \int\int\int H(x,y,z)g_1(x_0-x,y_0-y,z_0-z)dxdydz, \\
G_{y_0} &= [H * g_2]_0, \\
G_{z_0} &= [H * g_3]_0,
\end{aligned}$$

Soit le vecteur gradient $G_0 = (G_{x_0}, G_{y_0}, G_{z_0})^t$, dont la direction est orthogonale au contour défini par $z = ax + by$. Après calcul, on obtient la norme du gradient qui dépend de l'orientation du contour et est donnée par :

$$||G_0|| = Ag(a, b)$$

où $g(a, b)$ est le rapport de deux polynômes en a et b.

$$\begin{aligned} g(a,b) &= \frac{(ab+b+a)\sqrt{a^2+b^2+1}(b^4(a+1)^2+b^3(3a^3+8a^2+8a+3))}{(b+1)^3(a+b)^3(a+1)^3} \\ &+ \frac{(ab+b+a)\sqrt{a^2+b^2+1}(b^2(a^4+8a^3+15a^2+8a+1))}{(b+1)^3(a+b)^3(a+1)^3} \\ &+ \frac{(ab+b+a)\sqrt{a^2+b^2+1}(b(2a^4+8a^3+8a^2+2a)+a^4+3a^3+a^2)}{(b+1)^3(a+b)^3(a+1)^3} \end{aligned}$$

Il est intéressant d'étudier comment varie $g(a, b)$ avec la direction (a, b). On atteint la valeur maximale de 1 pour une normale au contour alignée avec un des axes du repère, soit si a et b tendent tous les deux vers 0 ou si un des deux tend vers l'infini. On atteint la valeur minimale 0.85 pour $(a, b) = (1, 1)$ soit pour un plan orienté à 45 degrés.

De manière à obtenir le cas 2D on pose $b = 0$ et dans ce cas :

$$g(a, 0) = \frac{(a^2 + 3a + 1)\sqrt{a^2 + 1}}{(a + 1)^3}$$

De nouveau la valeur maximale 1 est atteinte pour $a = 0$ ou $d \to \infty$. La valeur minimum $g(1, 0) \simeq 0.88$ est obtenue pour $a = 1$ ce qui correspond à un contour orienté à 45 degrés (voir figure 2.22.).

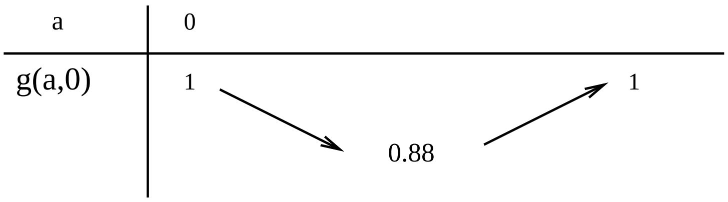


Figure 2.22. *Tableau des variations de $g(a, 0)$.*

Calcul de l'écart type de la réponse au bruit :

En utilisant le fait que $\int_{-\infty}^{+\infty} l_4(x)dx = 0$ (la réponse du filtre dérivée seconde à un signal constant est nulle), et $\int_{-\infty}^{+\infty} s_4(x)dx = 1$ (la réponse du filtre de lissage à un signal constant est ce signal) on obtient :

$$
\begin{aligned}
N_0 &= \int\int\int(\eta_1(x)+\eta_2(y)+\eta_3(z))D(x_0-x,y_0-y,z_0-z)dxdydz \\
&= \int \eta_1(x)l_4(x)dx + \int \eta_2(y)l_4(y)dy + \int \eta_3(z)l_4(z)dz
\end{aligned}
$$

La valeur du critère de détection de Canny (voir équation 2.23) s'obtient en faisant le rapport de la norme du gradient en un point du contour idéal et de l'écart type de la réponse au bruit :

$$
\mathrm{RSB} = \frac{||G_0||}{\sqrt{E(N_0^2)}} = \frac{Ag(a,b)}{\eta_0\sqrt{\alpha}\sqrt{\frac{3}{2}}}
$$

Localisation pour l'approche laplacien

Un point (x_0, y_0, z_0) est considéré comme point de contour si et seulement si :

$$
\theta_0 = \theta(x_0,y_0,z_0) = \int\int\int C(x,y,z)D(x-x_0,y-y_0,z-z_0)dxdydz = 0
$$

Ce produit de convolution peut être réécrit comme :

$$
\theta_0 = P_0 + N_0
$$

où $P_0 = [H * D]_0$ et $N_0 = [\eta * D]_0$.

On calcule d'abord la réponse au contour idéal $(H(x,y,z))$ soit :

$$
P_0 = A\int\int\int_{z\leq ax+by} D(x_0-x,y_0-y,z_0-z)dxdydz
$$

Et ensuite on développe P_0 au premier ordre autour de l'origine (ou de n'importe quel point du plan d'équation $z = ax + by$). Ce développement limité se justifie car on suppose que le point de contour détecté est au voisinage du vrai point de contour.

$$
P_0 \simeq A(ax_0 + by_0 - z_0)\alpha^2 m(a,b)
$$

où

$$\begin{aligned} m(a,b) &= \frac{(a^3b^3+3a^2b^3+3ab^3+b^3+7a^2b^2)(a^2+b^2+1)}{(a+b)^3\,(a+1)^3\,(b+1)^3} \\ &+ \frac{(3a^3b^2+3ab^2+3a^3b+3a^2b+a^3)(a^2+b^2+1)}{(a+b)^3\,(a+1)^3\,(b+1)^3} \end{aligned}$$

Etant donné que $\theta_0 = 0$ aux points de contours, le point (x_0, y_0, z_0) est un point de contour si $P_0 + N_0 = 0$. Ainsi pour chaque point de contour détecté on a $E(P_0^2) = E(N_0^2)$.

On obtient :

$$\begin{aligned} E\left[N_0^2\right] &= \frac{3}{2}\alpha^3\eta_0^2, \\ E\left[P_0^2\right] &= E\left[(ax_0+by_0-z_0)^2\right]2\alpha^4 m^2(a,b) \end{aligned}$$

L'écart type de la distance du point de contour détecté au plan contenant les vrais points de contour (inverse du critère de localisation de Canny) est donc :

$$\sigma\left[\frac{ax_0+by_0-z_0}{(a^2+b^2+1)^{\frac{1}{2}}}\right] = \sqrt{\frac{3}{2\alpha}}\frac{\eta_0}{A}h(a,b)$$

où :

$$h(a,b) = \frac{1}{m(a,b)\sqrt{a^2+b^2+1}}$$

Par définition le critère de localisation de Canny (voir équation [2.26]) correspond à l'inverse de cet écart type soit :

$$L = \sqrt{\frac{2\alpha}{3}}\frac{A}{\eta_0 h(a,b)}$$

On obtient ainsi une estimation quantitative de la qualité de la localisation des contours. Comme dans le cas 1D, la localisation est une fonction croissante de α. La fonction $h(a,b)$ apparaît comme un facteur de correction dépendant de l'orientation du contour, et qui est minimum quand la normale au contour est parallèle à un des trois axes, c.a.d. quand a et b tendent vers 0 ou quand l'un des deux tend vers l'infini. Dans ces trois cas, $h(a,b) \to 1$. La valeur maximale de $h(a,b)$ est $h(1,1) = 128/21\sqrt{3} \simeq 3.52$, obtenue pour un plan de contour dont la normale est parallèle au vecteur $(1,1,-1)$. Cela montre clairement l'importance de la prise en compte de l'orientation du contour pour la localisation.

Le cas 2D est obtenu en posant $b = 0$ ce qui entraine :

$$h(a,0) = \frac{1}{m(a,b)\sqrt{a^2+1}} = \frac{(a+1)^3}{(a^2+1)^{\frac{3}{2}}}$$

Dans ce cas, la valeur minimale de $h(a,0)$ est 1 pour $a = 0$ ou $a \to \infty$ (contours parallèles aux axes de coordonnées). La valeur maximale (produisant la pire localisation) est $2\sqrt{2} \simeq 2.83$ et est de nouveau obtenue pour $a = 1$ (voir figure 2.23.)

a	0		1		
h(a,0)	1	$\nearrow$	2.83	$\searrow$	1

Figure 2.23. *Tableau des variations de $h(a,0)$.*

2.6.7. Isotropie et invariants euclidiens pour les filtres linéaires

Le développement précédent illustre dans le cas du filtre de Deriche et de la norme du gradient les problèmes d'invariance (ou plutot de non-invariance) engendrés par des filtres de lissage non-isotropes. Dans cette partie, nous allons aborder ces problèmes de façon plus générale en montrant que l'invariance euclidienne est conservée si les dérivées partielles sont calculées à partir de filtres issues d'un filtre de lissage isotrope. On notera que l'isotropie est en théorie une condition suffisante mais non nécessaire et en pratique une condition aussi nécessaire.

Nous discuterons les propriétés des filtres dans l'espace continu et nous traiterons comme précédemment le cas 3D qui est plus général (une image se note $I(x,y,z)$ dans l'espace continu et $I(i,j,k)$ dans l'espace discret).

Soit $I(x,y,z)$ une image $3D$.

Supposons que nous cherchions les dérivées partielles de $I(x,y,z)$: $\dfrac{\partial^n (I(x,y,z))}{\partial x^m \partial y^p \partial z^q}$, $n = m+p+q$ que nous noterons avec la notation abrégée : $I_{x^m y^p z^q}$ (nous écrirons seulement les variables dont l'exposant est non nul, par exemple $I_{x^1y^0z^0}$ devient I_x).

Comme on l'a vu précédemment, si $f(x, y, z)$ est la réponse impulsionnelle du filtre de lissage, l'image restorée I_r est égale à $I * f$, où $*$ est le produit de convolution. On obtient donc :

$$\frac{\partial^n I_r}{\partial x^m \partial y^p \partial z^q} = \frac{\partial^n (I * f)}{\partial x^m \partial y^p \partial z^q} = I * (\frac{\partial^n f}{\partial x^m \partial y^p \partial z^q})$$

Ainsi, la réponse impulsionnelle du filtre qui calcule $I_{x^m y^p z^q}$ est $\frac{\partial^n f}{\partial x^m \partial y^p \partial z^q}$

Supposons que le filtre de lissage soit l'opérateur de Deriche :

$$f(x, y, z) = f_0(x) f_0(y) f_0(z)$$

avec

$$f_0(x) = c_0(1 + \alpha \mid x \mid) e^{-\alpha|x|}$$

où c_0 est une constante de normalisation. L'inconvénient de l'opérateur $f(x, y, z)$ est qu'il n'est pas isotrope (invariant par rotation). Par exemple, celà implique que la norme du gradient calculé avec les filtres de dérivation correspondants dépend de l'orientation du contour (voir paragraphe précédent). On pourrait aussi montrer la même chose pour le Laplacien ou d'autres invariants différentiels euclidiens.

On va maintenant montrer que si les dérivées partielles sont calculées avec des filtres dérivant d'un opérateur de lissage isotrope alors les invariants différentiels euclidiens "demeurent" invariants. On donnera la démonstration pour une image de dimension 3 mais elle s'applique pour une dimension quelconque.

Soit $I(x, y, z)$ une image $3D$.

Soit $J(x, y, z)$ une image I transformée par un déplacement rigide :

$$J(X_0, Y_0, Z_0) = I(x_0, y_0, z_0)$$

avec

$$(X_0, Y_0, Z_0)^t = R(x_0, y_0, z_0)^t + T$$

où R et T sont respectivement des matrices définissant une rotation autour de l'origine et une translation.

Soit f la réponse impulsionnelle d'un filtre de lissage isotrope, nous avons :

$$f(x_1, y_1, z_1) = f(x_2, y_2, z_2)$$

où

$$(x_2, y_2, z_2)^t = R(x_1, y_1, z_1)^t$$

ainsi

$$(I * f)(x_0, y_0, z_0) = (J * f)(X_0, Y_0, Z_0)$$

Soit E une fonction des dérivées partielles de I. Si E est un invariant différentiel euclidien de I nous avons :

$$E((I_x)(x_0, y_0, z_0), (I_y)(x_0, y_0, z_0), (I_z)(x_0, y_0, z_0), (I_{xx})(x_0, y_0, z_0)), (I_{xy})(x_0, y_0, z_0) \ldots) =$$
$$E((J_x)(X_0, Y_0, Z_0), (J_y)(X_0, Y_0, Z_0), (J_z)(X_0, Y_0, Z_0), (J_{xx})(X_0, Y_0, Z_0)), (J_{xy})(X_0, Y_0, Z_0) \ldots)$$

Si nous calculons les dérivées partielles en utilisant nos filtres, on obtient :

$$E((I * f_x)(x_0, y_0, z_0), (I * f_y)(x_0, y_0, z_0), (I * f_z)(x_0, y_0, z_0) \ldots.) =$$
$$E((I * f)_x(x_0, y_0, z_0), (I * f)_y(x_0, y_0, z_0), (I * f)_z(x_0, y_0, z_0) \ldots)$$

Etant donné que E est aussi un invariant différentiel euclidien de $I * f$, la dernière expression est égale à :

$$E((J * f)_x(X_0, Y_0, Z_0), (J * f)_y(X_0, Y_0, Z_0), (J * f)_z(X_0, Y_0, Z_0) \ldots) =$$
$$E((J * f_x)(X_0, Y_0, Z_0), (J * f_y)(X_0, Y_0, Z_0), (J * f_z)(X_0, Y_0, Z_0) \ldots)$$

Par conséquent, nous avons montré que :

$$E((I * f_x)(x_0, y_0, z_0), (I * f_y)(x_0, y_0, z_0), (I * f_z)(x_0, y_0, z_0) \ldots.) =$$
$$E((J * f_x)(X_0, Y_0, Z_0), (J * f_y)(X_0, Y_0, Z_0), (J * f_z)(X_0, Y_0, Z_0) \ldots)$$

L'équation précédente signifie que E reste invariant par changement de repère (invariant euclidien) si les dérivées partielles sont calculés à partir d'opérateurs dérivés de f.

On notera au passage qu'il existe deux types d'invariants différentiels euclidiens dans les images :

- Les invariants différentiels euclidiens pour chaque point de l'image : gradient, laplacien, courbures de la surface (hypersurface dans le cas $3D$) définie par l'image (voir chapitre : “Des images volumiques 3D à la géométrie des surfaces”).

- Les invariants différentiels euclidiens définis seulement sur les points de contour ; si on suppose que le gradient est paralléle à la normale à la surface (à la courbe dans le cas $2D$) les courbures s'expriment en fonction des dérivées partielles de l'image; les expressions correspondantes sont donc invariantes seulement sur les points de contour (voir chapitre "Des images volumiques 3D à la géométrie des surfaces").

La preuve précédente est aussi valide dans le second cas si on suppose que le gradient est paralléle à la normale pour $I * f$ ce qui est une hypothèse raisonnable.

Nous avons donc montré que si les dérivées partielles d'une image sont calculées avec des filtres dérivés d'un opérateur de lissage isotrope, alors les invariants différentiels euclidiens calculés en utilisant ces dérivées partielles, sont invariants par transformation rigide (invariance euclidienne).

La figure 2.24. illustre l'intérêt d'utiliser des filtres issus d'un filtre de lissage isotropique pour calculer les dérivées partielles ; on compare les courbures sur les contours d'un disque plein (voir chapitre "Des images volumiques 3D à la géométrie des surfaces" pour le calcul des courbures en fonction des dérivées). Les dérivées sont successivement calculées avec le filtre de Deriche et ensuite avec un filtre gaussien. On choisit la largeur des filtres de manière à ce que la courbure (inverse du rayon du cercle) ait la bonne valeur (en pixels). On discerne clairement les fausses variations de courbure induites par les filtres dérivant d'un opérateur anisotrope [136].

2.7. Filtres récursifs pour approximer les filtres gaussiens

La partie précédentes montrent clairement l'intérêt d'utiliser des opérateurs dérivés d'un filtre de lissage isotrope afin de calculer les dérivées partielles d'une image. D'un autre côté, on a aussi intérêt a mettre en œuvre des filtres récursifs séparables de manière à obtenir un temps de calcul raisonnable. Un moyen de joindre ces deux points antagonistes est d'utiliser des approximations récursives des filtres gaussiens et de leurs dérivées [45].

2.7.1. Calcul des dérivées d'un signal 1D

L'opérateur de lissage gaussien 1D est donné par :

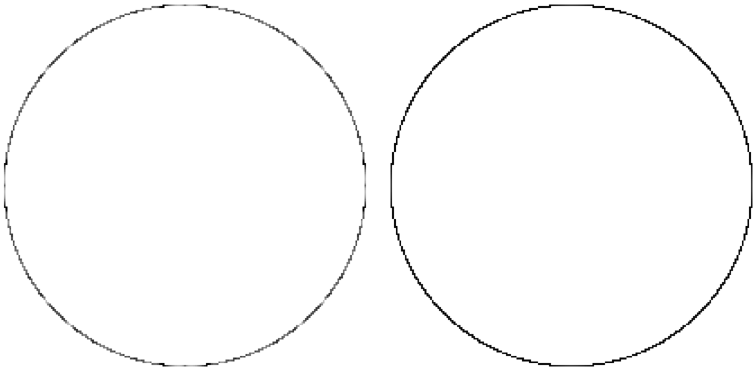


Figure 2.24. *Image gauche : Valeurs de la courbure obtenues avec les filtres dérivés d'un opérateur anisotrope (filtre de Deriche) ; Image droite : Valeurs de la courbure obtenues avec les filtres dérivés d'un opérateur isotrope (filtre gaussien)*

$$g(x) = e^{-\frac{x^2}{2\sigma^2}}$$

Les parties positives et négatives de g et de ses dérivées normalisées d'ordre $1, 2, 3$ peuvent être approximées (avec une erreur normalisée au sens des moindres carrées d'environ $\epsilon^2 = 10^{-6}$) par un filtre récursif (IIR) d'ordre 4 de la forme :

$$h(x) = (a_0 \cos(\frac{\omega_0}{\sigma}x) + a_1 \sin(\frac{\omega_0}{\sigma}x))e^{-\frac{b_0}{\sigma}x} + (c_0 \cos(\frac{\omega_1}{\sigma}x) + c_1 \sin(\frac{\omega_1}{\sigma}x))e^{-\frac{b_1}{\sigma}x} \quad [2.36]$$

On réalise ces approximations de la manière suivante.

$h(x)$ peut s'écrire :

$$h(x) = \sum_{i=1}^{m} \alpha_i e^{-\frac{\lambda_i x}{\sigma}}$$

où les cœfficients α_i et λ_i sont des nombres complexes conjuguées (dans le cas présent m est égal à 4). L'erreur au sens des moindres carrées est définie par :

$$\epsilon^2 = \frac{\sum_{k=0}^{10\sigma}(h(k) - u(k))^2}{\sum_{k=0}^{10\sigma} u(k)^2}$$

où u est la fonction que l'on veut approximer avec h.

Ce problème de minimisation peut être résolu en utilisant un algorithme classique de minimisation. Un bon point de départ pour une méthode itérative de minimisation peut être obtenu avec la méthode de Prony.

La méthode de Prony utilise un échantillonage équidistant de la fonction $((x_0, y_0 = f(x_0)), \ldots, (x_n, y_n = f(x_n)))$. On peut poser $x_r = x_0 + rh$, $c_r = \alpha_r e^{\lambda_r x_0}$ et $v_r = h\lambda_r$, pour $r = 0, \ldots m$, et écrire $f(x) = \sum_{i=1}^{m} \alpha_i \lambda_i x$. Ainsi, on obtient le système suivant :

$$\begin{cases} c_1 + c_2 + \ldots + c_m = y_0 \\ c_1 v_1 + c2 v_2 + \ldots + c_m v_m = y_1 \\ \vdots \\ c_1 v_1^m + c_2 v_2^m + \ldots + c_m v_m^m = y_m \end{cases} \qquad [2.37]$$

Si on construit les polynômes :

$$\prod_{i=1}^{m} (v - v_i) = \sum_{i=1}^{m} s_i v_{m-i} = \phi(v) \qquad [2.38]$$

avec $s_0 = 1$, en multipliant chaque équation de [2.37] par s_i, $i = 0 \ldots m$, et en additionant chacune de ces nouvelles équations, on obtient :
$\sum_{i=1}^{m} \phi(v_i) c_i = \sum_{i=0}^{m} s_i y_{m-i} = 0$, puisque $\phi(v_r) = 0$.
Si on choisit $n \geq m$, on peut aussi écrire :

$$\begin{cases} y_{m-1} s_1 + y_{m-2} s_2 + \ldots + y_0 s_m = -y_m \\ y_m s_1 + y_{m-1} s_2 + \ldots + y_1 s_m = -y_{m+1} \\ \vdots \\ y_{n-1} s_1 + y_{n-2} s_2 + \ldots + y_{n-m} s_m = -y_n \end{cases}$$

Ces $n - m + 1$ équations avec m inconnues s_i définissent un sytème au sens des moindres carrés. Ensuite, nous obtenons les inconnues v_i en résolvant [2.38]. Finalement, nous obtenons les cœfficients λ_i avec la formule $\lambda_i = \frac{\ln v_i}{h}$ les c_i à partir de [2.37], et les α_i avec $\alpha_i = c_i v_i^{-\frac{x_0}{h}}$.

L'avantage de cette méthode est qu'il n'est pas nécessaire d'affecter à σ une valeur particulière pour déterminer les cœfficients λ_i et α_i. Les cœfficients a_0, a_1, b_0, b_1, c_0, c_1, ω_0, ω_1 sont aisément obtenus à partir des α_i et λ_i.

Nous allons maintenant donner les résultats d'approximation obtenus pour g et ses dérivées normalisées d'ordre 1,2,3 que nous noterons : g_1, g_2, g_3. Ces opérateurs sont normalisés de maniére à ce que le résultat obtenu soit correct si on les applique à des polynômes. Soit :

$$\sum_{-\infty}^{+\infty} g(x)dx = 1;$$

$$\sum_{-\infty}^{+\infty} xg_1(x)dx = 1;$$

$$\sum_{-\infty}^{+\infty} g_2(x)dx = 0 \text{ et} \sum_{-\infty}^{+\infty} \frac{x^2}{2} g_2(x)dx = 1;$$

$$\sum_{-\infty}^{+\infty} \frac{x^3}{6} g_3(x)dx = 1 \text{ et } \sum_{-\infty}^{+\infty} xg_3(x)dx = 0.$$

On définit γ_1, γ_2, γ_3, γ_4 comme suit :

- $\gamma_1 = \sum_{k=0}^{\infty} -\frac{k^2}{2\sigma^2}$
- $\gamma_2 = \sum_{k=0}^{\infty} k^2 - \frac{k^2}{2\sigma^2}$
- $\gamma_3 = \sum_{k=0}^{\infty} k^4 - \frac{k^2}{2\sigma^2}$
- $\gamma_4 = \sum_{k=0}^{\infty} k^6 - \frac{k^2}{2\sigma^2}$.

On peut montrer qu'une fonction $F(\sigma) = \sum_{k=1}^{\infty} k^{\alpha} - \frac{k^2}{(2\sigma^2)}, \alpha > 0$ peut être approximée avec une très faible erreur par une fonction polynômiale $F_a(\sigma) = I(\alpha)\sigma^{\alpha+1}$ avec $I(\alpha) = \int_0^{\infty} t^{\alpha} - \frac{t^2}{2} dt$.
Nous avons $I(0) = \sqrt{\frac{\pi}{2}}, I(1) = 1$, et $I(\alpha+2) = (\alpha+1)I(\alpha)$.
Si $\alpha = 0$, on approxime $F(\sigma)$ avec $F_a(\sigma) = 1.25\sigma - 0.5$ pour $\sigma \geq 0.5$.
Ainsi nous obtenons les approximations suivantes :

- $\gamma_1 \approx 0.5 + 1.25\sigma$
- $\gamma_2 \approx \sigma^3 \sqrt{\frac{\pi}{2}}$
- $\gamma_3 \approx 3\sigma^5 \sqrt{\frac{\pi}{2}}$
- $\gamma_4 \approx 15\sigma^7 \sqrt{\frac{\pi}{2}}$

Les cœfficients de normalisation sont égaux à :

- $\beta_0 = \dfrac{1}{2\gamma_1 - 1} \approx \dfrac{0.4}{\sigma}$

- $\beta_1 = -\dfrac{1}{2\gamma_2} \approx -\dfrac{0.4}{\sigma^3}$
- $\beta_2 = -\dfrac{2\gamma_2}{-2\gamma_2^2 + 2\gamma_3\gamma_1 - \gamma_3} \approx -\dfrac{0.4}{\sigma^3}$
- $\beta_3 = \dfrac{2\gamma_1 - 1}{2\gamma_2} \approx \dfrac{1}{\sigma^2}$
- $\beta_4 = \dfrac{3}{\frac{\gamma_4\gamma_2}{\gamma_3\sigma^2} - \frac{\gamma_3}{\sigma^2}} \approx \dfrac{1.2}{\sigma^3}$
- $\beta_5 = \dfrac{3\gamma_2}{\gamma_3(\frac{\gamma_4\gamma_2}{\gamma_3\sigma^2} - \frac{\gamma_3}{\sigma^2})} \approx \dfrac{0.4}{\sigma^5}$

Nous obtenons les résultats suivants pour les cœfficients des filtres d'approximation :

	$g(x) = \beta_0 e^{-\frac{x^2}{2\sigma^2}}$	$g_1(x) = \beta_1 x e^{-\frac{-x^2}{2\sigma^2}}$
a_0	$0.657/\sigma$	$-0.173/\sigma^2$
a_1	$1.979/\sigma$	$-2.004/\sigma^2$
c_0	$-0.258/\sigma$	$0.173/\sigma^2$
c_1	$-0.239/\sigma$	$0.444/\sigma^2$
b_0	1.906	1.561
b_1	1.881	1.594
ω_0	0.651	0.700
ω_1	2.053	2.145

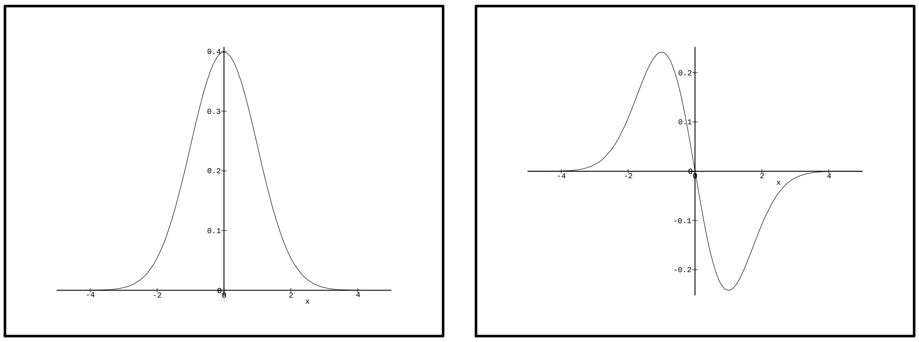
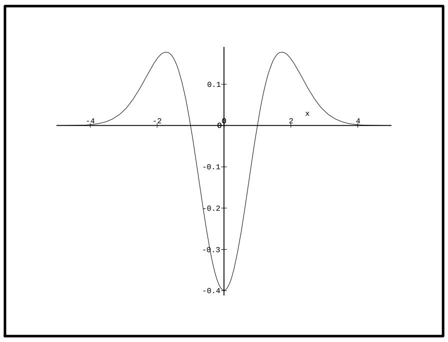


Figure 2.25. *Le filtre gaussien normalisé et sa première dérivée ($\sigma = 1$)*

Pour g et $\sigma = 1$, $\epsilon^2 = 9.7E-09$. Pour g_1, $\epsilon^2 = 1.2E-06$.

Pour $g_2(x) = \beta_2(1 - \beta_3 x^2)e^{-\frac{x^2}{2\sigma^2}}$ et $g_3(x) = \frac{1}{\sigma^2}(\beta_4 x - \beta_5 x^3)e^{-\frac{x^2}{2\sigma^2}}$, on obtient :

	$g_2(x)$	$g_3(x)$
a_0	-0.724/σ^3	1.286/σ^4
a_1	1.689/σ^3	-0.290/σ^4
c_0	0.325/σ^3	-1.286/σ^4
c_1	-0.721/σ^3	0.262/σ^4
b_0	1.295	1.012
b_1	1.427	1.273
ω_0	0.779	0.947
ω_1	2.234	2.338



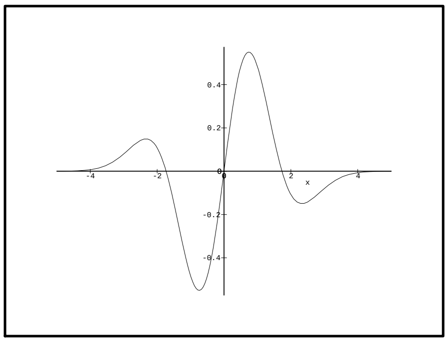


Figure 2.26. *Dérivées deuxième et troisième du filtre gaussien normalisées ($\sigma = 1$)*

Pour g_2 et $\sigma = 1$, $\epsilon^2 = 7.3E - 06$. Pour g_3 et $\sigma = 1$, $\epsilon^2 = 1.9E - 04$.

Une fois que les cœfficients $a_0, a_1, c_0, c_1, b_0, b_1, \omega_0, \omega_1$ sont calculés, on peut aisément déterminer une réalisation récursive de $h(n)$. On calcule la transformée en Z : $H_1(z) = \sum_{n=0}^{\infty} h(n)z^{-n}$ pour la partie causale du filtre, et $H_2(z) = \sum_{n=-\infty}^{-1} h(n)z^{-n}$ pour la partie anticausale. Les détails d'implémentation peuvent être trouvés par exemple dans [45].

2.7.2. Conclusion

La prise en compte de modèles de contour réellement 2D ou 3D même aussi simples qu'un contour orienté remet en question l'optimalité des opérateurs. L'utilisation de modèles plus complets incluant des propriétés d'angles ou de courbures est sans doute intéressante d'un point de vue théorique. Cependant il paraît difficile d'obtenir des opérateurs de mise en œuvre aisée avec des modèles plus complexes.

Les algorithmes basés sur le filtrage séparable récursif apportent néanmoins des solutions intéressantes pour la détection de contours. Ces méthodes décrites précedemment permettent en effet d'obtenir des résultats satisfaisants sur la plupart des types d'images.

2.8. Détection de contours par lissage : Haralick

L'idée de ce modèle est d'approximer localement le signal discret formant l'image par une fonction dérivable, puis de dériver cette fonction [81].

Soit $I(r, c)$ l'image, $(r, c, I(r, c))$ forme une surface.

On suppose que localement l'image peut être approximée par une fonction polynômiale

$$I(r, c) = k_1 + k_2 r + k_3 c + k_4 r^2 + k_5 rc + k_6 c^2 + k_7 r^3 + k_8 r^2 c + k_9 rc^2 + k_{10} c^3$$

On recherche les cœfficients de ce polynôme en construisant une base de l'ensemble des fonctions polynômiales dans $R \times C$ (R et C étant des ensembles d'indices ; $r \in R \Rightarrow -r \in R$; $c \in C \Rightarrow -c \in C$). On va montrer que dans certains cas particuliers il existe un polynôme discret (défini dans l'espace des entiers relatifs) interpolant unique passant par tous les points de l'image. On montrera aussi que les cœfficients de ce polynôme peuvent être déterminés par des produits de convolution.

Soit $\{P_0(r), \ldots P_{N-1}(r)\}$ un ensemble de polynômes discrets orthogonaux entre eux.

$\{Q_0(r, c), Q_1(r, c) \ldots Q_{p-1}(r, c)\}$ peut être obtenu par produit tensoriel.

On prend pour $P_o(r), \ldots P_{n-1}(r)$ les polynômes discrets de Tchebycheff :

On définit : $P_o(r) = 1$.

On suppose : $P_0(r) \ldots P_{n-1}(r)$ définis par récurrence comme suit :

$$\left\{\begin{array}{l} P_n(r) = r^n + a_{n-1}r^{n-1} + \ldots + a_1 r + a_0 \\ P_n(r) \perp P_i(r) \qquad \forall i \in \{0, \ldots, n-1\} \end{array}\right.$$

Soit

$$\left\{\begin{array}{l} \sum_{r \in R} P_k(r)(r^n + a_{n-1}r^{n-1} + \ldots + a_1 r + a_0) = 0 \\ k = 0, \ldots, n-1 \end{array}\right.$$

On obtient ainsi n équations linéaires d'inconnues $a_0, \ldots, a_{n-1}$. Pour $n =$ 5 la résolution de ce système permet l'obtention d'une base orthogonale de l'ensemble des polynômes de degré 4 :

$$\left\{\begin{array}{l} P_0(r) = 1 \\ P_1(r) = r \\ P_2(r) = r^2 - \frac{u_2}{u_0} \\ P_3(r) = (r^3 - \frac{u_4}{u_2})r \\ P_4(r) = \frac{r^4 + (u_2 u_4 - u_0 u_6) r^2 + (u_2 u_6 + u_2^4)}{u_0 u_4 - u_2^2} \end{array}\right.$$

$\{P_0(r), \ldots P_N(r)\}$ est un ensemble de polynômes discrets et orthogonaux sur R.

$\{Q_0(c), \ldots Q_M(c)\}$ est un ensemble de polynômes discrets et orthogonaux sur C.

$$\left\{\begin{array}{l} r \in R \Rightarrow -r \in R \\ c \in C \Rightarrow -c \in C \end{array}\right. \Rightarrow \{P_0(r)Q_0(c), \ldots P_n(r)Q_n(c), \ldots P_N(r)Q_M(c)\}$$

est un ensemble de polynômes discrets et orthogonaux sur $R \times C$.

En effet

$$\sum_{r \in R} \sum_{c \in C} P_i(r)Q_j(c)P_n(r)Q_m(c) = (\sum_{r \in R} P_i(r)P_n(r))(\sum_{c \in C} Q_j(c)Q_n(c)) = 0$$

Si on pose : $R = C = \{-1, 0, 1\}$ on obtient une base orthogonale des polynômes discrets de degré 2 sur $R \times C$:

$$\{1, r, c, r^2 - \frac{2}{3}, rc, c^2 - \frac{2}{3}, r(c^2 - \frac{2}{3}), c(r^2 - \frac{2}{3}), (r^2 - \frac{2}{3})(c^2 - \frac{2}{3})\}$$

Au voisinage d'un point on a :

$$I(r,c) = \sum_{n=0}^{N-1} a_n P_n(r,c)$$

avec a_i tel que $e^2 = \sum_{(r,c)\in R\times C}[I(r,c) - \sum_{n=0}^{N-1} a_n P_n(r,c)]^2$ minimum

Soit

$$a_m = \frac{\sum_{(r,c)\in R\times C} P_m(r,c) I(r,c)}{\sum_{(r,c)\in R\times C} P_m^2(r,c)} \quad \text{car} \quad < P_m, P_n >= 0$$

On obtient dans ce *cas particulier* des cœfficients a_m tels que $e^2 = 0$ par produit scalaire (ce sont les composantes du polynôme d'interpolation dans la base (P_i)) D'où les cœfficients a_m du polynôme recherché s'expriment comme combinaison linéaire des valeurs de l'image dans un voisinage 3×3 autour du point. On peut donc générer des masques permettant le calcul des cœfficients du polynôme :

$$\frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad : \quad \text{cœfficient de } 1$$

$$\frac{1}{6}\begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad : \quad \text{cœfficient de } r$$

$$\frac{1}{6}\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad : \quad \text{cœfficient de } c$$

$$\frac{1}{6}\begin{pmatrix} 1 & 1 & 1 \\ -2 & -2 & -2 \\ 1 & 1 & 1 \end{pmatrix} \quad : \quad \text{cœfficient de } (r^2 - \tfrac{2}{3})$$

$$\frac{1}{4}\begin{pmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{pmatrix} \quad : \quad \text{cœfficient de } (rc)$$

$$\frac{1}{6}\begin{pmatrix} 1 & -2 & 1 \\ 1 & -2 & 1 \\ 1 & -2 & 1 \end{pmatrix} \quad : \quad \text{cœfficient de } (c^2 - \tfrac{2}{3})$$

$$\frac{1}{4}\begin{pmatrix} -1 & 0 & 1 \\ 2 & 0 & -2 \\ -1 & 0 & 1 \end{pmatrix} \quad : \quad \text{cœfficient de } (r^2 - \tfrac{2}{3})c$$

$$\frac{1}{4}\begin{pmatrix} -1 & 2 & -1 \\ 0 & 0 & 0 \\ 1 & -2 & 1 \end{pmatrix} \quad : \quad \text{cœfficient de } (c^2 - \tfrac{2}{3})r$$

$$\frac{1}{4}\begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix} \quad : \quad \text{cœfficient de } (r^2 - \tfrac{2}{3})(c^2 - \tfrac{2}{3})$$

On obtient ainsi neuf masques correspondant à une fenêtre 3×3 permettant de calculer par des produits de convolution un polynôme approximant localement l'image en chaque point (ce polynôme interpole les 9 points correspondant au point considéré et à ces 8 voisins). La base de fonctions polynômes utilisée définit implicitement un modèle du bruit. Le problème de ce type de méthodes est la sensibilité des approximations polynômiales au bruit.

2.9. Détection de contours par optimisation

Le principe de cette approche est de rechercher directement dans l'image un modèle de contour basé sur la séparation locale de l'image en deux zones homogènes [100].

Le modèle utilisé est une droite subdivisant le mieux possible un disque centré en un point en deux régions homogènes :

$$T(x, y, c, s, l, d, b) = \begin{cases} b & \text{si} \quad cx + sy \leq l \\ b + d & \text{si} \quad cx + sy > l \end{cases}$$

On considère qu'il existe une bijection entre les contours idéaux et les fonctions $T(x, y, c, s, l, d, b)$.

Soit $I(x, y)$ l'image :

On cherche (c, s, l, d, b) tel que :

$$\epsilon^2 = \int\int_D (I(x, y) - T(x, y, s, l, b, d))^2 dxdy = ||I - T|| \text{ minimum.}$$

De manière à minimiser cette fonction, on décompose selon une base orthogonale de l'espace hilbertien des fonctions sur D.

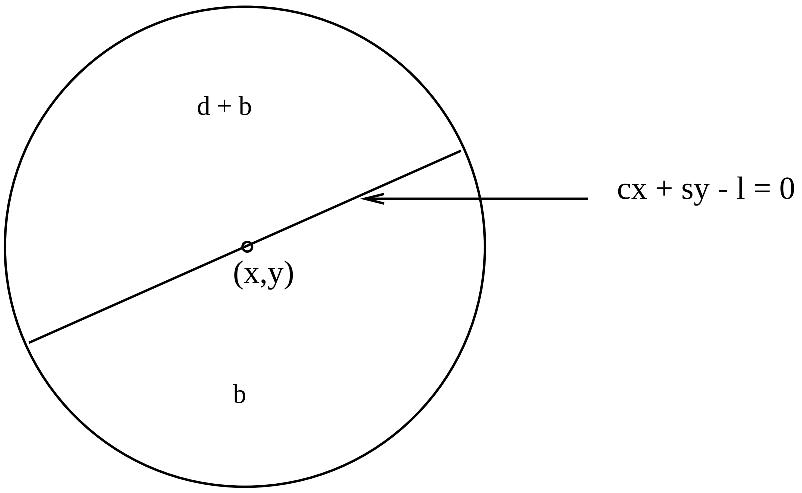


Figure 2.27. *Contour idéal au sens de Hueckel.*

$\{H_i\}_{i=0}^{\infty}$ base de l'espace hilbertien des fonctions sur D.
On obtient :

$$\begin{aligned} a_i &= \int H_i(x,y)I(x,y)dxdy = H_i \cdot I \\ s_i &= \int H_i(x,y)T(x,y,c,s,l,b,d)dxdy = H_i \cdot T \\ a_i &: \text{constantes} \\ s_i &: \text{variables} \end{aligned}$$

$$I(x,y) = \sum_i a_i H_i(x,y)$$
$$T(x,y,c,s,l,b,d) = \sum_i s_i H_i(x,y)$$

$$\begin{aligned} \int\int_D [I(x,y) - T(x,y,c,s,l,b,d)]^2 dxdy &= \\ \int\int_D [\sum_i (a_i - s_i)H_i(x,y)]^2 dxdy &= \sum_i (a_i - s_i)^2 \end{aligned}$$

On décompose les fonctions sur D en produits de composantes radiales et angulaires :

$$r^2 = x^2 + y^2, \quad Q(r) = (1 - r^2)^{\frac{1}{2}}$$

$$\begin{aligned}
H_0(x,y) &= (\frac{5}{6\pi})Q(r)(1+2r^2) \\
H_1(x,y) &= (\frac{8}{27\pi})^{\frac{1}{2}}Q(r)(5-2) \\
H_2(x,y) &= (\frac{3}{\pi})^{\frac{1}{2}}Q(r)x(4r^2-1) \\
H_3(x,y) &= (\frac{3}{\pi})^{\frac{1}{2}}Q(r)y(4r^2-1) \\
H_4(x,y) &= (\frac{3}{\pi})^{\frac{1}{2}}Q(r)x(3-4r^2) \\
H_5(x,y) &= (\frac{3}{\pi})^{\frac{1}{2}}Q(r)y(3-4r^2) \\
H_6(x,y) &= (\frac{32}{3\pi})^{\frac{1}{2}}Q(r)(x^2-y^2) \\
H_7(x,y) &= (\frac{32}{3\pi})^{\frac{1}{2}}Q(r)2xy
\end{aligned}$$

On obtient :

$$\begin{aligned}
H_1 &= (\frac{9}{2})^{\frac{1}{2}}H_1 \\
H_2 &= H_2 + H_4 \\
H_3 &= H_3 + H_5 \\
H_4 &= (\frac{3}{2})H_6 \\
H_5 &= (\frac{3}{2})H_7 \\
H_6 &= (\frac{5}{4})^{\frac{1}{2}}(H_2 - H_4) \\
H_7 &= (\frac{5}{4})^{\frac{1}{2}}(H_3 - H_5)
\end{aligned}$$

Ensuite on minimise par décomposition sur cette base de fonctions et on obtient (T, d, b) en fonction des luminances des points considérés.

Etudions maintenant le cas particulier de l'opérateur de Hueckel pour la détection de contour dans un voisinage 2×2 traité par Jean-François Abramatic.

$$s(x,y) = \begin{cases} a & \text{si} \quad x\sin\theta \geq y\cos\theta \\ b & \text{sinon} \end{cases}$$

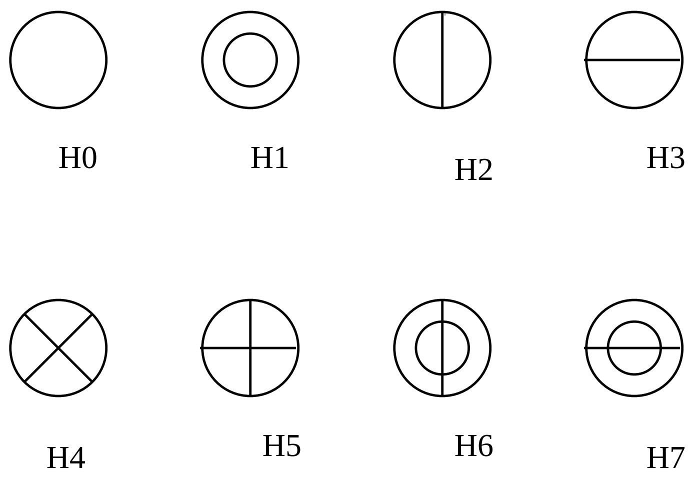


Figure 2.28. *Représentations graphiques des fonctions de la base de Hueckel.*

Soit (a^*, b^*) le couple de valeurs optimales, on va montrer que

$$| a^* - b^* | = \max(| B - C |, | A - D |)$$

Soit (H_1, H_2, H_3, H_4) la base canonique de l'espace des fonctions sur le voisinage 2×2 :

$$H_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad H_2 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} H_3 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \quad H_4 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

$$s(x, y, a, b, \theta) = \sum_{i=1}^{4} s_i H_i$$

$$\begin{aligned} f &= \sum_{i=1}^{4} f_i H_i \\ f &= AH_1 + BH_2 + CH_3 + DH_4 \end{aligned}$$

$$f_0 = A \ ; \ f_1 = b \ ; \ f_2 = C \ ; \ f_3 = D$$

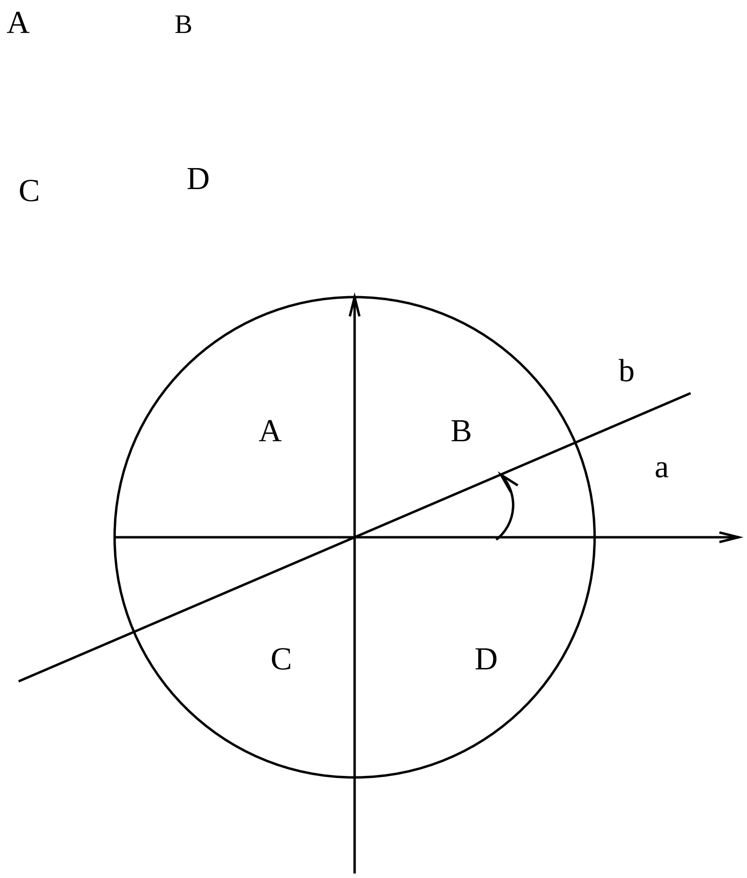


Figure 2.29. *Cas d'un voisinage 2×2 pour le modèle de Hueckel.*

$$s_0 = \begin{cases} b & \text{pour} \quad 0 \leq \theta \leq \frac{\pi}{2} \\ a(\frac{\theta - \frac{\pi}{2}}{\frac{\pi}{2}}) + b(\frac{\pi - \theta}{\frac{\pi}{2}}) & \text{pour} \quad \frac{\pi}{2} \leq \theta \leq \pi \\ a & \text{pour} \quad \pi \leq \theta \leq \frac{3\pi}{2} \\ a(\frac{2\pi - \theta}{\frac{\pi}{2}}) + b(\frac{\theta - \frac{3\pi}{2}}{\frac{\pi}{2}}) & \text{pour} \quad \frac{3\pi}{2} \leq \theta \leq 2\pi \end{cases}$$

$$s_1 = \begin{cases} a(\frac{\theta}{\frac{\pi}{2}}) + b(\frac{\frac{\pi}{2} - \theta}{\frac{\pi}{2}}) & \text{pour} \quad 0 \leq \theta \leq \frac{\pi}{2} \\ a & \text{pour} \quad \frac{\pi}{2} \leq \theta \leq \pi \\ a(\frac{\frac{3\pi}{2} - \theta}{\frac{\pi}{2}}) + b(\frac{\theta - \pi}{\frac{\pi}{2}}) & \text{pour} \quad \pi \leq \theta \leq \frac{3\pi}{2} \\ b & \text{pour} \quad \frac{3\pi}{2} \leq \theta \leq \pi \end{cases}$$

$$s_2 = \begin{cases} a(\frac{\frac{\pi}{2} - \theta}{\frac{\pi}{2}}) + b(\frac{\theta}{\frac{\pi}{2}}) & \text{pour} \quad 0 \leq \theta \leq \frac{\pi}{2} \\ b & \text{pour} \quad \frac{\pi}{2} \leq \theta \leq \pi \\ a(\frac{\theta - \frac{\pi}{2}}{\frac{\pi}{2}}) + b(\frac{\frac{3\pi}{2} - \theta}{\frac{\pi}{2}}) & \text{pour} \quad \pi \leq \theta \leq \frac{3\pi}{2} \\ a & \text{pour} \quad \frac{3\pi}{2} \leq \theta \leq \pi \end{cases}$$

$$s_3 = \begin{cases} a & \text{pour} \quad 0 \leq \theta \leq \frac{\pi}{2} \\ a(\frac{\pi - \theta}{\frac{\pi}{2}}) + b(\frac{\theta - \frac{\pi}{2}}{\frac{\pi}{2}}) & \text{pour} \quad \frac{\pi}{2} \leq \theta \leq \pi \\ b & \text{pour} \quad \pi \leq \theta \leq \frac{3\pi}{2} \\ a(\frac{\theta - \frac{3\pi}{2}}{\frac{\pi}{2}}) + b(\frac{2\pi - \theta}{\frac{\pi}{2}}) & \text{pour} \quad \frac{3\pi}{2} \leq \theta \leq 2\pi \end{cases}$$

Pour des raisons de symétrie on n'étudie que le premier et le second cadran.

Pour le premier cadran on obtient :

$$\epsilon^2 = (b - A)^2 + (a - D)^2 + (\frac{2\theta}{\pi}a + (1 - \frac{2\theta}{\pi})b - B)^2 + (\frac{2\theta}{\pi}b + (2 - \frac{2\theta}{\pi})a - c)^2$$

ϵ^2 minimum $\Longrightarrow \frac{\partial \epsilon^2}{\partial a} = \frac{\partial \epsilon^2}{\partial b} = \frac{\partial \epsilon^2}{\partial \theta} = 0$

Soit :

$$\begin{cases} a = \frac{3A + B + C - D}{4} \\ b = \frac{3D + B + C - A}{4} \\ \theta = \frac{\pi}{4} - \pi 4 \frac{C - B}{A - D} \end{cases}$$

Les calculs sont similaires pour le 2ème cadran. On obtient :

$$| a^* - b^* | = \max(| B - C |, | A - D |)$$

Ce qui correspond à l'opérateur de Roberts.

Comme dans le cas du modèle d'Haralick, le problème du modèle de Hueckel est la définition non explicite du bruit. Par contre, ces deux modèles sont réellement 2D ce qui n'est pas le cas du modèle de Canny.

2.10. Perspectives

Les propriétés essentielles d'un opérateur de détection de contour dans des images bidimensionnelles sont :

1. définition explicite du bruit,

2. prise en compte des propriétés bidimensionnelles d'un contour : courbures, angles,

3. coût calculatoire.

Les opérateurs issus du modèle de Canny apportent une réponse satisfaisante pour les points 1 et 3 mais négligent le point 2. Les opérateurs associés à des modèles de type Haralick ou Hueckel prennent en compte, du moins implicitement, le point 2 et conduisent à des algorithmes de coût raisonnable mais négligent le point 1.

Une autre alternative fournissant des résultats intéressants consiste à approximer la fonction des niveaux de gris de l'image par une fonction spline [35].

On peut raisonnablement penser que l'utilisation de mathématiques plus sophistiquées pourra permettre de définir des modèles prenant en compte à la fois une définition explicite du bruit et des propriétés géométriques locales. Cependant il semble peu vraisemblable que de tels modèles débouchent sur des algorithmes de coût raisonnable. On peut alors espérer que l'émergence des machines parallèles ou d'ordinateurs séquentiels plus puissants permettra leur implantation.

2.11. Résultats expérimentaux

La méthode reposant sur le filtrage séparable récursif que nous avons décrite a été appliquée aux cas 2D [43], 3D [128], et 4D [159] en utilisant les filtres de Shen et de Deriche. On obtient ainsi de bons résultats, au moins meilleurs que ceux des méthodes plus anciennes, sur une grande variété d'images : images 2D de scènes d'intérieur et d'extérieur, images aériennes, images médicales 3D obtenues par résonance magnétique nucléaire, images 4D constituées par des séquences temporelles d'images médicales 3D ...

Nous présentons dans cette section, à titre indicatif, quelques résultats expérimentaux (figures 2.30. à 2.11.). Sur une station de travail SUN-4 et pour une image $512 \times 512 \times 512$ le temps total de calcul incluant le filtrage, l'extraction des extréma et le seuillage par hystéresis est de l'ordre de 30 secondes CPU.

Figure 2.30. *Image originale d'une scène d'intérieur.*

Figure 2.31. *Valeur absolue du gradient en X de l'image précédente calculé avec le filtre Deriche* ($\alpha = 1$).

Figure 2.32. *Valeur absolue du gradient en Y de l'image précédente calculé avec le filtre Deriche ($\alpha = 1$).*

Figure 2.33. *Extréma locaux de la norme du gradient dans la direction du gradient (la valeur affichée est la norme du gradient).*

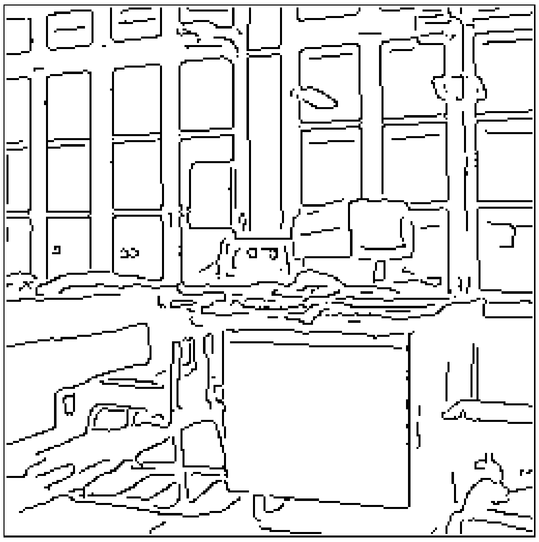


Figure 2.34. *Contours obtenus après seuillage des extréma locaux ($\alpha = 1$).*

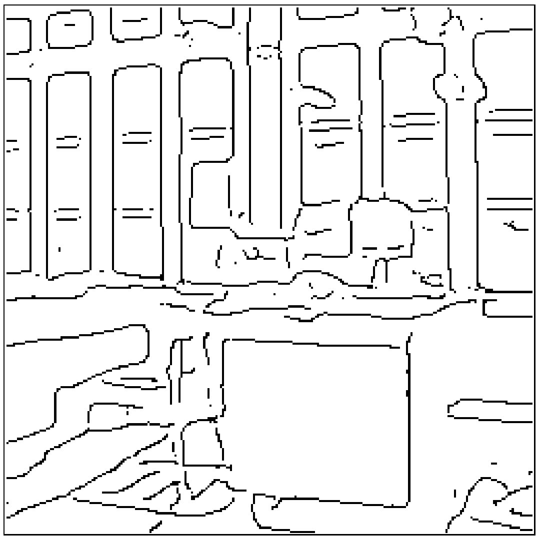


Figure 2.35. *Contours obtenus après seuillage des extréma locaux ($\alpha = 0.5$).*

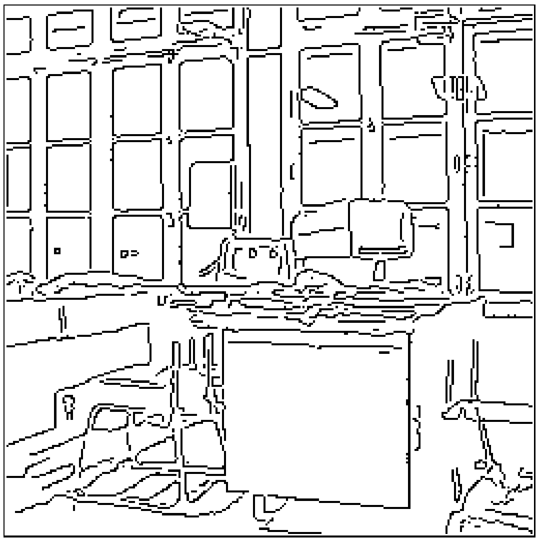


Figure 2.36. *Contours obtenus après seuillage des extréma locaux ($\alpha = 1.5$).*

Figure 2.37. *Image originale (coupe sagittale du cerveau obtenue par IRM).*

Figure 2.38. *Extréma locaux obtenus avec le filtre Deriche.*

Figure 2.39. *Contours obtenus par seuillage par hystéresis des extréma locaux.*

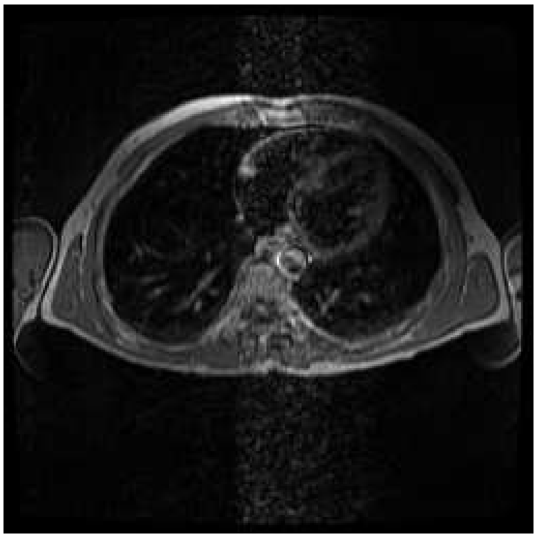


Figure 2.40. *Image originale (coupe d'une image IRM du thorax).*

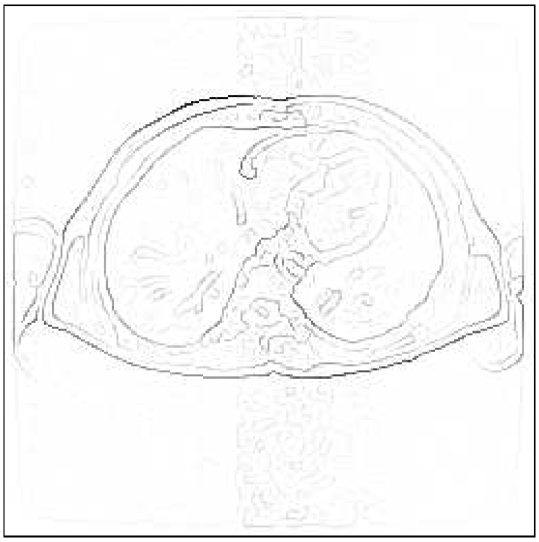


Figure 2.41. *Extréma locaux obtenus avec le filtre de Deriche.*

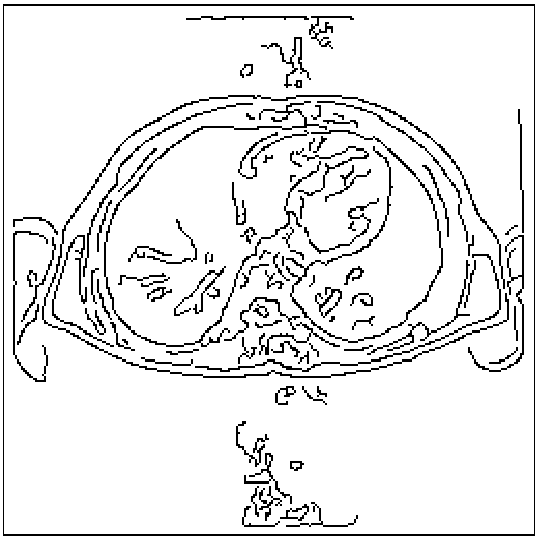


Figure 2.42. *Contours obtenus par seuillage par hystérésis des extréma locaux.*

Figure 2.43. *Image originale.*

Figure 2.44. *Extréma locaux obtenus avec le filtre de Deriche.*

Figure 2.45. *Contours obtenus par seuillage par hysteresis des extréma locaux.*

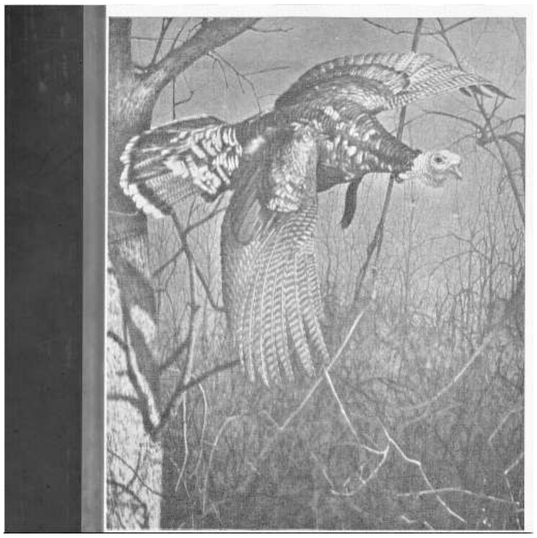


Figure 2.46. *Image originale.*

Figure 2.47. *Extréma locaux obtenus avec le filtre de Deriche.*

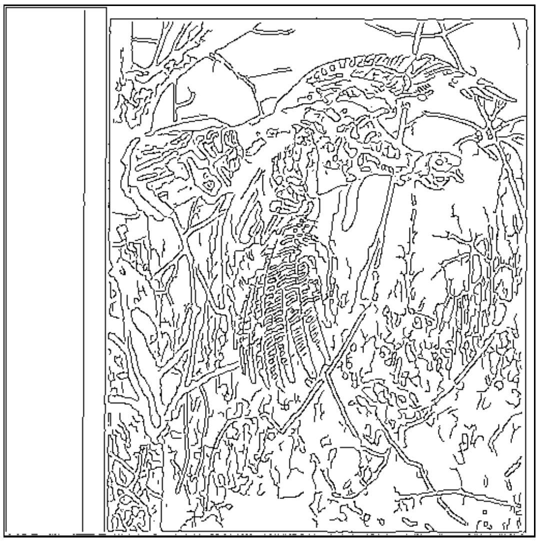


Figure 2.48. *Contours obtenus par seuillage par hysteresis des extréma locaux.*

2.12. Amélioration, suivi et chaînage des contours

Il est souvent difficile de sélectionner des seuils adéquats pour l'étape de seuillage. Des seuils élevés permettent de supprimer les points de contour dus au bruit mais aussi de vrais points de contour. Des seuils bas permettent d'obtenir tous les vrais points de contour mais aussi ceux dus au bruit. Le choix de seuils définit un compromis entre les vrais et les faux contours. La raison de cette limitation des algorithmes de détection de contours réside dans la nature locale des opérateurs. En effet lorsque le bruit de l'image est important il est nécessaire de définir aussi des critères globaux. Ces critères s'expriment généralement par une fonction à optimiser sur un ensemble d'attributs locaux. Le plus souvent on choisit de rechercher dans l'image des chemins optimisant la somme des normes du gradient calculé en chaque point à l'aide d'un opérateur local.

On peut aussi rajouter un critère de forme de contour minimisant les variations de courbure et la longueur des contours seconde (minimisation des courbures). Lorsqu'on prend simplement en compte un critère de normes de gradient, des algorithmes de recherche de chemins optimaux dans un graphe suffisent pour l'optimisation [123]. Si on considère aussi les courbures on utilise alors le calcul variationnel pour minimiser une fonctionnelle incluant la somme des normes du gradient sur le contour, la longueur et la somme des courbures [102, 38]. On ajuste alors un modèle déformable sur les données, ces algorithmes sont connus sous le nom de “snake” (serpent) mais leur danger réside dans le choix de la fonctionnelle à utiliser (cette fonctionnelle résulte le plus souvent de l'addition de quantités non comparables). Généralement on applique les méthodes de suivi non pas sur toute l'image mais simplement dans les zones où les opérateurs locaux ne donnent pas des résultats satisfaisants. Par exemple, si on suppose que les contours réels sont fermés, on peut mettre en œuvre un algorithme de suivi de contours à partir des extrémités des lignes de contours ouverts.

2.12.0.1. Recherche de chemin optimal dans un graphe

L'idée de ce type d'algorithme est de poser le problème de la détection de contour comme la recherche d'un chemin de coût minimum dans un graphe. Les données sont une carte incomplète des contours et les résultats une meilleure carte des contours. Les composantes critiques de ces méthodes sont :

1. prendre un bon point de départ ;

2. comment les résultats déjà obtenus doivent-ils influencer le choix du point suivant : fonction à optimiser ?

3. critère d'arrêt.

L'approche classique consiste à rechercher un chemin de coût minimum dans le graphe d'adjacence des points de l'image.

On définit une fonction d'évaluation :

- $F(n) = G(n) + H(n)$
- n : état
- F : fonction de coût
- G : coût pour aller de l'état n vers un état de départ
- H : estimation du coût pour aller de l'état n vers un état but.

Un exemple de ce type d'algorithme a été proposé par Martelli [123] :

But : Une ligne de contraste qui commence sur la ligne en haut de l'image et se termine sur la ligne du bas de l'image.

Etats : Eléments de contours définis par deux points :
$A = (i, j), \quad B = (i, j+1)$ définissant le contour orienté AB

Contour : Séquence d'éléments de contour commençant en haut de l'image et se terminant en bas, ne contenant pas de boucle, et n'ayant aucun élément de direction haut.

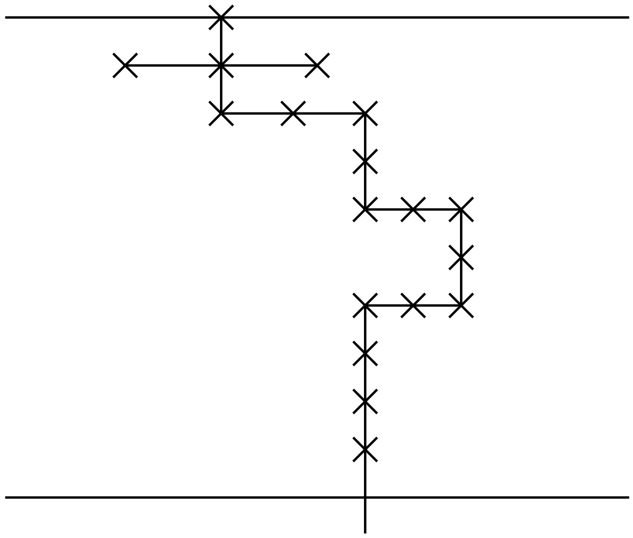


Figure 2.49. *Chemins recherchés.*

Un contour est donc un chemin dans le graphe représentant l'espace des états.

La détermination du meilleur contour se ramène à la recherche d'un chemin optimal dans le graphe.

Soit Max le niveau de gris maximum de l'image I

$$
\begin{aligned}
n &= A_r A_s \\
C(n) &= \mathrm{Max} - I(A_r) + I(A_s) \\
G(n) &= \text{coût pour aller de l'état } n \text{ à un état initial} = \sum C(s), \\
& s \quad \text{parcourant le chemin optimal reliant l'état de départ à l'état } n \\
H(n) &= \text{estimation du coût du chemin optimal reliant l'état } n \text{ à un état final} \\
F(n) &= G(n) + H(n)
\end{aligned}
$$

Pour optimiser F on peut utiliser des méthodes de programmation dynamique ou un algorithme de type A^*.

2.12.1. Chaînage de contours

Dans la plupart des applications la description matricielle des contours fournis par la détection de contours n'est pas suffisante. Il est alors nécessaire de passer à une description sous forme de listes chaînées de contours. Nous décrivons les principes d'un algorithme de chaînage opérant dans le cas 2D uniquement.

L'objectif du chaînage de contours est de passer d'une description matricielle des contours à une description sous forme de listes de contours chaînées. Cette étape qui semble aisée pose en fait des problèmes difficiles :

• le temps de calcul : cette étape n'étant que transitoire, il faut éviter des temps de calcul trop prohibitifs ;

• la représentation des relations entre les chaînes : les relations entre les chaînes doivent être gérées sans perdre d'informations par rapport à l'information matricielle ;

• la structuration de l'information : il faut bien structurer l'information pour que les manipulations de données effectuées par la suite soient aisées.

P. Garnesson et G. Giraudon [68] ont mis au point un algorithme de chaînage qui résout en partie ces problèmes en permettant :

• un chaînage rapide en une seule lecture de l'image ;

• une gestion efficace des liens de parenté ;

• la possibilité d'un stockage étendu pour divers types d'informations (niveau de gris, pixel, amplitude du gradient, ...) ;

• la possibilité d'éliminer des chaînes ;

• la minimisation du nombre de chaînes ;

• la possibilité de travailler avec des stockages d'images réduits ;

• une faisabilité temps réel envisageable pour un coût réduit.

Pour cela, ils ont repris l'idée de décomposition d'un pavé 3 x 3 centré autour d'un pixel, en passé, présent, futur, idée déjà utilisée pour le coloriage de taches ou sur des problèmes de restauration d'images. La méthode se divise en trois étapes principales :

- création des chaînes en parallèle en une passe par un balayage de l'image avec un pavé 3 x 3 ; les chaînes sont créées par mise à jour itérative par de simples jeux de pointeurs ;
- fusion des chaînes obtenues afin de minimiser leur nombre ;
- élimination de chaînes suivant un critère.

L'originalité de cette approche par rapport aux nombreux algorithmes de chaînage existant est de proposer une structure de données "propre" permettant d'effectuer le chaînage et de représenter son résultat. En effet il faut noter que dans les travaux antérieurs, l'aspect structure de données n'avait pas été abordé aussi clairement. Une méthode d'approximation polygonale peut permettre ensuite d'obtenir à partir des chaînes de contours une approximation polygonale des contours (voir le chapitre 3).