

Support de cours pour le congrès PIXIM 89. Version révisée. Segmentation d'images : ou en sommes nous ?

Olivier Monga

► **To cite this version:**

Olivier Monga. Support de cours pour le congrès PIXIM 89. Version révisée. Segmentation d'images : ou en sommes nous ?. [Rapport de recherche] RR-1216, INRIA. 1990, pp.62. inria-00075342

HAL Id: inria-00075342

<https://hal.inria.fr/inria-00075342>

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INRIA

UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.: (1) 39 63 55 11

Rapports de Recherche

N° 1216

Programme 6
Robotique, Image et Vision

SUPPORT DE COURS POUR LE
CONGRES PIXIM 89

Version révisée

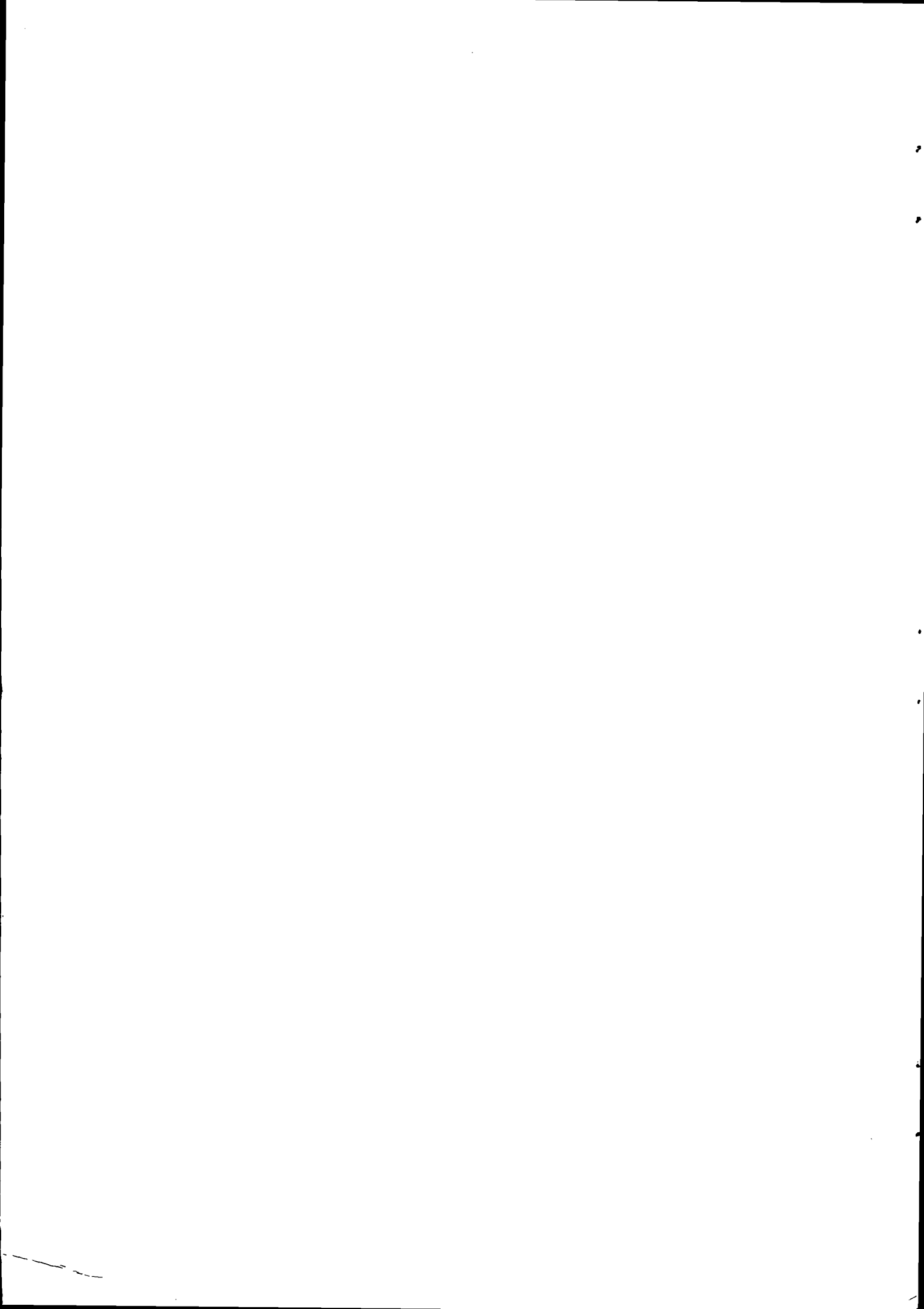
SEGMENTATION D'IMAGES :
Où EN SOMMES NOUS ?

Olivier MONGA

Avril 1990



★ R R - 1 2 1 6 ★



1 Introduction

L'objectif de la vision artificielle est d'extraire automatiquement des informations d'une image. Les images traitées proviennent de domaines d'application variés : robotique, imagerie médicale, images aériennes ou satellites, sismique Il ne s'agit pas toujours d'images bidimensionnelles (images d'intérieur ou d'extérieur ...) mais aussi d'images tridimensionnelles (tomographie, imagerie sismique, séquence temporelle d'images 2d ...) et parfois même d'images de dimension 4 (séquence temporelle d'images tomographiques 3d ...). La richesse et la diversité du contenu informationnel d'une image pose le problème du choix de l'information pertinente pour résoudre une tâche donnée : navigation d'un robot mobile, localisation de réseaux routiers, identification de failles sismiques David Marr a proposé de décrire un système de vision selon trois niveaux [18] :

1. Quelle est l'information présente dans l'image qui est réellement nécessaire pour résoudre cette tâche : Que calculer ?
2. Choix de méthodes pour calculer l'information : Avec quels algorithmes ; comment calculer ?
3. Sur quel matériel faut-il implanter ces méthodes : Avec quoi calculer ; ordinateur biologique ou ordinateur informatique ?

On peut noter que les trois niveaux de Marr s'appliquent autant à un système complet de vision qu'à chacune de ses composantes. La segmentation d'images constitue la première étape des systèmes de vision et permet d'engendrer une description compacte de l'image (contours, régions) plus exploitable que l'ensemble des pixels. Un problème de segmentation d'images se spécifie donc par la caractérisation de l'information à extraire : modélisation analytique et par les algorithmes mis en œuvre pour extraire cette information : solution algorithmique.

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.

L'objectif de cet article est de faire le point sur les méthodes existantes et d'identifier les problèmes ouverts.

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

...the ...
...the ...
...the ...
...the ...
...the ...
...the ...
...the ...
...the ...
...the ...
...the ...
...the ...
...the ...

...the ...
...the ...
...the ...
...the ...
...the ...
...the ...

...the ...
...the ...
...the ...
...the ...
...the ...
...the ...
...the ...
...the ...
...the ...
...the ...
...the ...
...the ...

...the ...
...the ...
...the ...
...the ...
...the ...
...the ...

...the ...
...the ...
...the ...
...the ...
...the ...
...the ...
...the ...

...the ...
...the ...
...the ...
...the ...
...the ...
...the ...

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 implique la recherche des discontinuités locales de la fonction des niveaux de gris de l'image. L'extraction de régions nécessite la détermination 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 n'est cependant pas exploitée dans la plupart des méthodes existantes.

Cet article se partage naturellement en deux parties : détection de contours et extraction de régions. Son objectif n'est pas d'être exhaustif mais de présenter parmi les approches les plus récentes celles qui ont été incluses dans des systèmes de vision opérationnels.

2 Détection de contours

2.1 Introduction

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 différentiation 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).

2.2 Problématique : de la détection de contours nd au filtrage $1d$

L'objectif de cette partie est de montrer que sous réserve d'hypothèses raisonnables, on peut ramener la détection de contours dans des images de dimensions quelconques au lissage et à la dérivation par filtrage linéaire d'un signal monodimensionnel.

Soit $I(x_1, x_2, \dots, x_n)$ un signal bruité de dimension n .

Par exemple $n = 2$ dans le cas d'une image naturelle ou $n = 3$ pour une image médicale obtenue par scanner.

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). Quoiqu'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 $G(I)$ le gradient de I :

$$G(I) = \left(\frac{dI}{dx_1}, \frac{dI}{dx_2}, \dots, \frac{dI}{dx_n} \right)^t$$

Soit $L(I)$ le laplacien de I :

$$L(I) = \frac{d^2 I}{dx_1^2} + \frac{d^2 I}{dx_2^2} + \dots + \frac{d^2 I}{dx_n^2}$$

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 [19]. 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 1 à 4 illustrent cette remarque en dimension 1 :

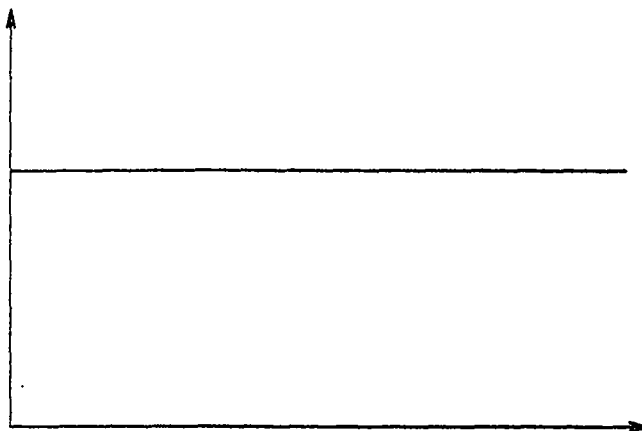


Figure 1 : Signal idéal

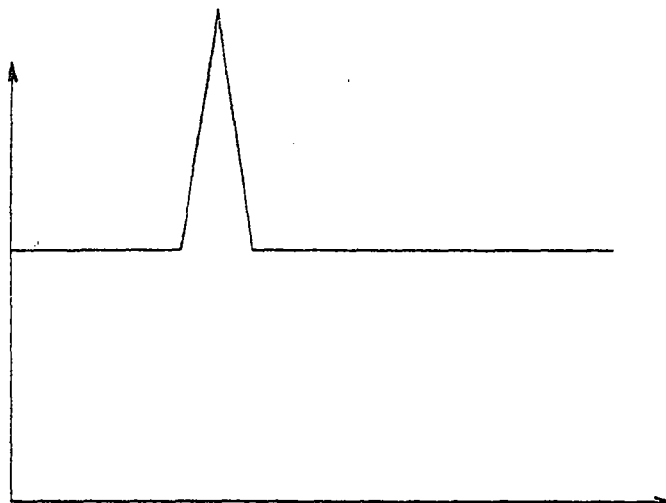


Figure 2 : Signal bruité par un bruit impulsionnel

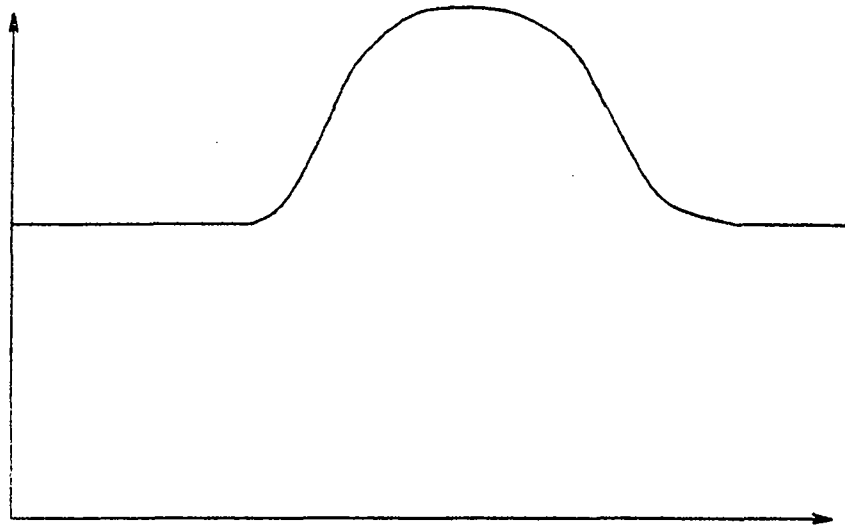


Figure 3 : Signal lissé par un filtre linéaire : pas d'issue !

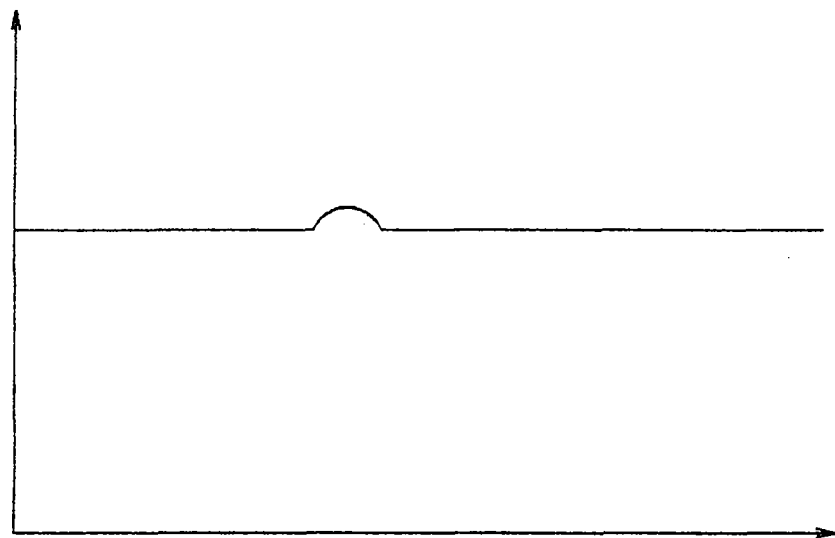


Figure 4 : Signal après un filtrage médian de dimension 3

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 $(f(i_{p-1}), f(i_p), f(i_{p+1}))$

$j_p = i_p$ si $f(i_p) \leq f(i_{p+1})$ et $f(i_{p-1}) \leq f(i_p)$

$j_p = i_{p-1}$ si $f(i_{p+1}) \leq f(i_{p-1})$ et $f(i_{p-1}) \leq f(i_p) \dots$

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.

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 $L(x_1, x_2, \dots, x_n)$. On calcule les dérivées premières et secondes par rapport aux variables x_i en convoluant avec $\{\frac{dL(x_1, x_2, \dots, x_n)}{dx_i}\}$ et $\{\frac{d^2L(x_1, x_2, \dots, x_n)}{dx_i^2}\}$. On remarque que le calcul du laplacien peut s'effectuer directement par convolution avec le filtre de réponse : $\frac{d^2L(x_1, x_2, \dots, x_n)}{dx_1^2} + \dots + \frac{d^2L(x_1, x_2, \dots, x_n)}{dx_n^2}$. On verra dans la section suivante que parfois cette expression 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 caractéristiques différentes [33]. 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 [23] revient à poser :

$$L = \begin{vmatrix} 1 & 1 \\ 1 & 1 \end{vmatrix} \quad dL/dx = \begin{vmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{vmatrix}$$

$$dL/dy = \begin{vmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{vmatrix}$$

Figure 5 : 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.

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 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 ou 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 6 présente les masques proposés par Kirsch [23]. Ces masques sont obtenus par rotation d'un masque de dérivation selon 8 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. Quoiqu'il en soit ce type d'algorithme peut se révéler très efficace dans les zones de l'image où plusieurs contours se croisent.

$$\begin{array}{cc}
 N \begin{vmatrix} 5 & 5 & -3 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{vmatrix} & N - W \begin{vmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{vmatrix} \\
 \\
 W \begin{vmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{vmatrix} & S - W \begin{vmatrix} 5 & 5 & -3 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{vmatrix}
 \end{array}$$

Figure 6 : Masques de Kirsch correspondant aux directions $N, N - W, W, S - W$

Cependant lorsque l'on choisit d'approximer le gradient par dérivation selon n directions (n : dimension de l'image), 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 $p^2 \times d_x d_y$. 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 : $2 \times p^2 \times d_x d_y$. Ces remarques ont amené l'introduction de filtres à réponse impulsionnelle séparable selon les directions x_1, x_2, \dots, x_n :

$$L(x_1, x_2, \dots, x_n) = L_{x_1}(x_1) \cdot L_{x_2}(x_2) \dots L_{x_n}(x_n)$$

Le filtrage séparable comporte plusieurs avantages :

- réduction du temps de calcul d'une convolution $\overbrace{p \dots p}^n$ de p^n à $n \cdot p$.
- 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, z . 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.

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

$$I * L(x_1, x_2, \dots, x_n) = I * (L_{x_1}(x_1) \times L_{x_2}(x_2) \dots L_{x_n}(x_n))$$

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 :

$$I * (f.g) = (I * f) * g$$

L peut donc s'implanter par la cascade des filtres $L_{x_1}, L_{x_2} \dots L_{x_n}$.

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

$$L_{x_1} = L_{x_2} \dots = L_{x_n} = S$$

On supposera dans la suite que le bruit est isotrope mais l'approche que nous décrivons s'adapte directement au cas d'un bruit anisotropique mais homogène selon $X_1, X_2 \dots X_n$. On verra en effet que le filtrage récursif ne peut être utilisé que si le bruit est homogène selon chaque variété de dimension 1 : $x_i = \text{cste}, i \neq j$.

On obtient :

$$I * L(x_1, x_2, \dots, x_n) = I * (S(x_1).S(x_2) \dots S(x_n))$$

On pose :

$$D = S'$$

$$P = S''$$

d'où

$$\frac{dI(x_1, x_2, \dots, x_n)}{dx_i} = I * (S(x_1).S(x_2) \dots D(x_i) \dots S(x_n))$$

$$\frac{d^2I(x_1, x_2, \dots, x_n)}{dx_i^2} = *I * (S(x_1).S(x_2) \dots P(x_i) \dots S(x_n))$$

on pose :

$$Q_i(x_1, x_2, \dots, x_n) = S(x_1) \cdot S(x_2) \dots P(x_i) \dots S(x_n)$$

d'où

$$Lap(I) = I * (Q_1(x_1, x_2, \dots, x_n) + Q_2(x_1, x_2, \dots, x_n) \dots + Q_n(x_1, x_2, \dots, x_n))$$

Le premier opérateur utilisant le filtrage séparable a été introduit par Sobel (voir figure 7) [32] :

$$H1 = \begin{vmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{vmatrix} \quad H2 = \begin{vmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{vmatrix}$$

Figure 7 : 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 :

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

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.

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

brèvement ces traitements commun à tous les opérateurs avant d'introduire des opérateurs de filtrage 1D.

Soit $I(x_1, x_2, \dots, x_n)$ une image de dimension n

Soit $G(x_1, x_2, \dots, x_n)$ le gradient de I au point (x_1, x_2, \dots, x_n)

$$G(x_1, x_2, \dots, x_n) = (G_{x_1}(x_1, x_2, \dots, x_n), \dots, G_{x_n}(x_1, x_2, \dots, x_n))$$

Soit $Lap(x_1, x_2, \dots, x_n)$ le laplacien de I au point (x_1, x_2, \dots, x_n)

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 [32] 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

$$N(x_1, x_2, \dots, x_n) = (G_{x_1}(x_1, x_2, \dots, x_n)^2 + \dots + G_{x_n}(x_1, x_2, \dots, x_n)^2)^{1/2}$$

2. Sélection des points de fort gradient

On détermine les points tels que :

$$N(x_1, x_2, \dots, x_n) > s$$

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 élevé 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 [5]. Ces extréma correspondent aux passages par zéro de la dérivée seconde dans la direction du gradient. Dans une deuxième étape on élimine les points de norme de gradient faible avec un seuillage par hystérésis [5]. 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 $G(M)$ et d une distance seuil (par exemple $s = 1$).

Soient M_1 et M_2 les deux points de la droite passant par M et de vecteur directeur $G(M)$ 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.

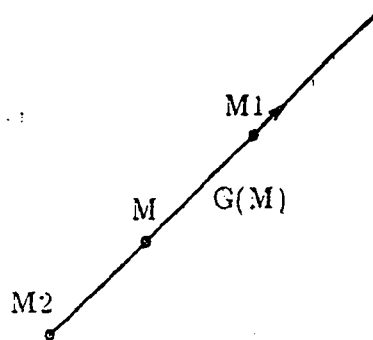


Figure 8 : Extraction des extréma locaux : cas 2d

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 supérieur 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'extréma considéré et un extréma de norme de gradient plus élevée qu'un seuil haut s_h . Le seuil haut s_h et le seuil bas s_b peuvent par exemple être déterminé à partir de l'histogramme cumulé des valeurs de la norme du gradient aux extréma [23]. L'algorithme se décompose donc en deux étapes :

- détermination de deux images I_h et I_b telles que :

$$I_h(M) = 0 \text{ si } N(M) < s_h$$

$$I_h(M) = 1 \text{ si } N(M) \geq s_h$$

$$I_b(M) = 0 \text{ si } N(M) < s_b$$

$$I_b(M) = 1 \text{ si } N(M) \geq s_b$$

- 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 [23]. 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.

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 Lap . 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 $Lap(M) = 0$. 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 :

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

$$I_p(M) = 1 \text{ si } Lap(M) \leq 0$$

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 fermés 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 ont été aussi proposées [33].

Il faut noter que la méthodologie que nous avons développée dans cette partie 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 consacré au filtrage 1D. Ensuite nous décrirons les algorithmes que nous obtenons dans le cas 2D et dans le cas 3D.

2.3 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 vues dans les deux sens. Nous montrons enfin que le filtrage récursif amène pour certains de ces filtres des implantations de faible complexité algorithmique.

2.3.1 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 défini 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 :

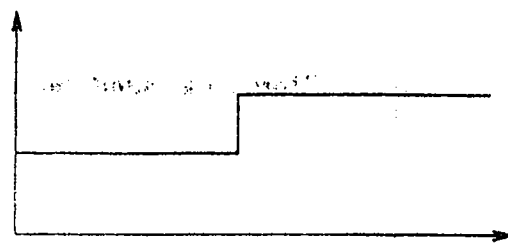
$$I(x) = Au_{-1}(x) + \eta(x)$$

avec

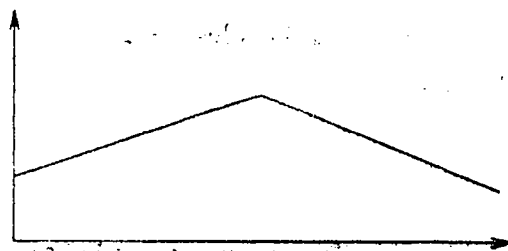
$$\eta_0^2 = E[\eta^2(x)]$$

$$u_{-1}(x) = 0 \text{ pour } x \text{ dans }]-\infty, 0[$$

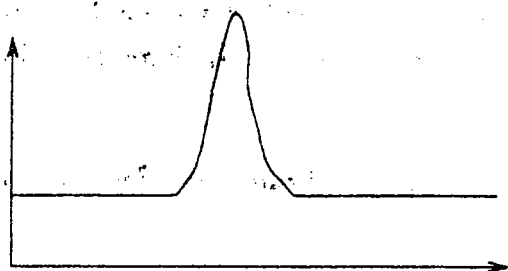
$$u_{-1}(x) = 1 \text{ pour } x \text{ dans }]0, +\infty[$$



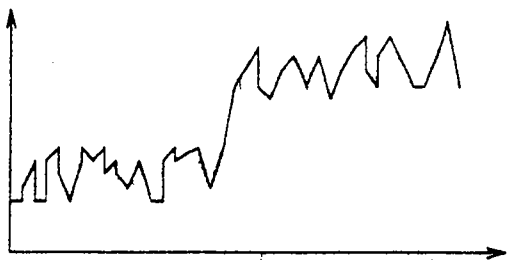
contour idéal



contour en toit



contour en pointe



contour idéal bruité

Figure 9 : Différents types de contours

2.3.2 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

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

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 [5] 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 SNR :

$$SNR = \frac{A}{\eta_0} \cdot \frac{\int_{-\infty}^0 f(x)dx}{(\int_{-\infty}^{+\infty} f^2(x)dx)^{\frac{1}{2}}} = \frac{A}{\eta_0 \cdot \Sigma}$$

- 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 :

$$L = \frac{A}{\eta_0} \cdot \frac{|f'(0)|}{(\int_{-\infty}^{+\infty} f'^2(x)dx)^{\frac{1}{2}}} = \frac{A}{\eta_0 \cdot \lambda}$$

- une réponse à un contour : le détecteur ne doit pas fournir de multiples réponses à un seul contour ; la distance entre les pics dans la réponse au bruit de f , noté x_{max} , est posée comme une fraction k de la longueur de l'opération W :

$$x_{max} = k \cdot W = 2\pi \left(\frac{\int_{-\infty}^{+\infty} f'^2(x) dx}{\int_{-\infty}^{+\infty} f''(x) dx} \right)$$

Ces critères sont combinés d'une manière significative : on maximise le produit $\sum \lambda$ sans la contrainte du 3ème critère. On obtient ainsi une équation différentielle dont la solution est :

$$f(x) = a_1 \cdot e^{\alpha x} \cdot \sin \omega x + a_2 \cdot e^{\alpha x} \cdot \cos \omega x + a_3 \cdot e^{-\alpha x} \cdot \sin \omega x + a_4 \cdot e^{-\alpha x} \cdot \cos \omega x + c \quad (1)$$

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 $[0, W]$.

Shen et Castan [33] proposent de déterminer les opérateurs de lissage optimisant un critère E incluant la détection et la localisation :

$$E = ((E_N + E'_N) / E_S^2)^{\frac{1}{2}}$$

où

$$E_N = \eta_0^2 \cdot \int_{-\infty}^{+\infty} f^2(x) dx$$

E_N est la réponse du filtre au bruit

$$E'_N = \eta_0^2 \cdot \int_{-\infty}^{+\infty} f'^2(x) dx$$

E'_N est la réponse de la dérivée du filtre au bruit

$$E_S = A^2 \cdot f(0)$$

E_S est la réponse du filtre à la frontière échelon
on remarque :

$$E = c \left(\frac{1}{\sum \lambda} \right)^{\frac{1}{2}}$$

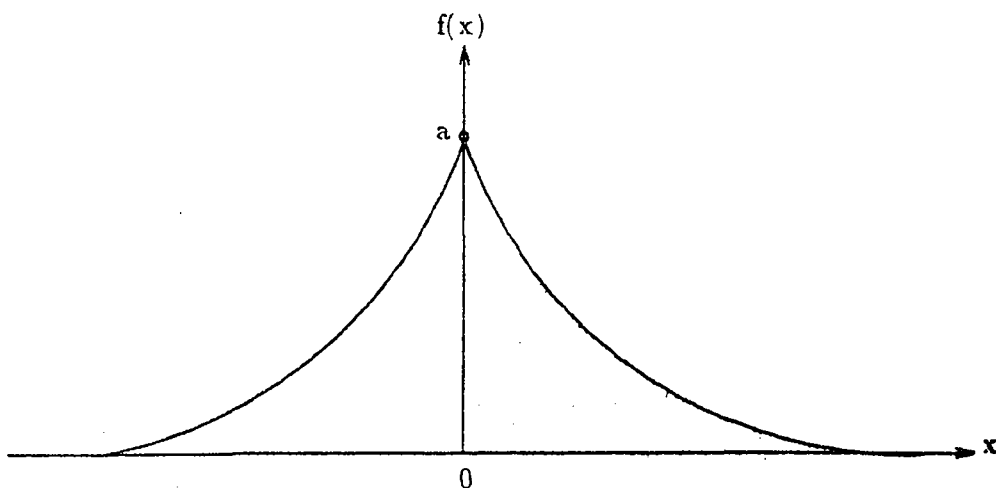


Figure 10 : Réponse impulsionnelle de s_1

La minimisation du critère défini par E est donc équivalente à la maximisation du produit des deux premiers critères de Canny. On obtient aussi une équation différentielle dont la résolution donne [34] :

$$f(x) = c_1 \times e^{p \cdot x} + c_2 \times e^{-p \cdot x}$$

2.3.3 Filtres optimaux de lissage

1. Filtre Shen

$$s_1(x) = c \cdot e^{-\alpha|x|}$$

c est choisi de manière à obtenir un filtre normalisé soit tel que :

$$\int_{-\infty}^{+\infty} s_1(x) dx = 1$$

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

Ce filtre a été introduit par Shen et Castan pour l'approximation du laplacien par différence entre l'image originale et l'image lissée [33, 34]. Le filtre obtenu par dérivation (voir partie : filtres optimaux de dérivation) constitue une solution optimale pour la première partie des critères de Canny (voir section suivante). Il correspond à une valeur optimale pour le produit $\sum \times \lambda$ soit au meilleur compromis détection-localisation. La discontinuité d'ordre 1 au point 0 permet d'éviter une délocalisation importante des

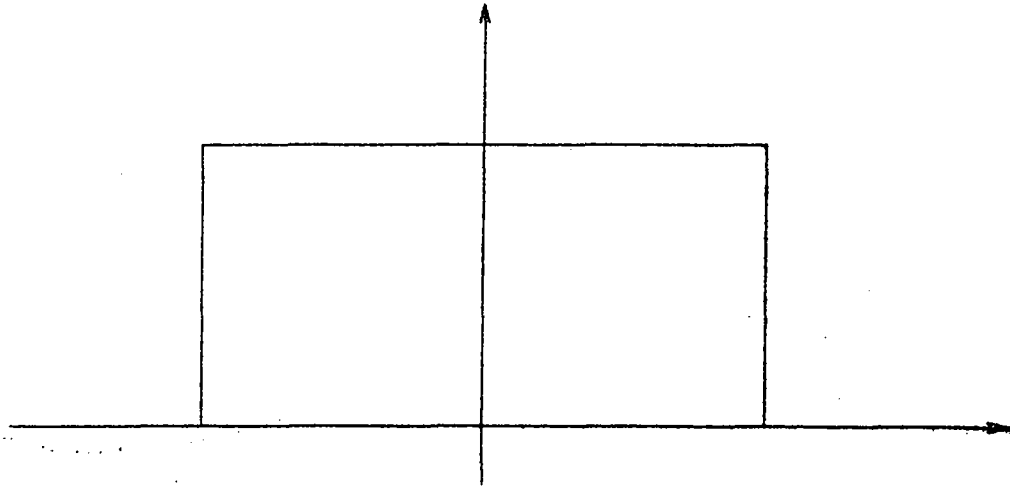


Figure 11 : Réponse impulsionnelle de s_2

contours dans l'image lissée même avec des valeurs faibles de α . Cependant cette discontinuité peut entraîner la détection de contours multiples [5].

2. Filtre moyenne

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

Ce filtre est le plus simple à implémenter et est utilisé depuis longtemps. Il a par exemple été utilisé pour approximer le laplacien par différence entre les images obtenu par lissage avec deux filtres de dimension (n) différentes [16]. Il est équivalent au filtre s_1 lorsque α tend vers zéro. Il est recommandé de l'utiliser avec des valeurs de n 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

$$s_3(x) = c \cdot e^{-\alpha x^2}$$

c est choisi de manière à obtenir un filtre normalisé

Ce filtre a été introduit par Marr et Hildreth pour le calcul du laplacien [19]. Il correspond à la valeur optimale pour le produit de la variance temporelle et de la variance spectrale. Le filtre de dérivation correspondant $s'_3(x)$ 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 $\sum \cdot \lambda = 1.12$. Il propose l'utilisation de la première dérivée d'une gaussienne pour laquelle : $\sum \cdot \lambda = 0.92$ et $k = 0.51$.

4. Filtre Deriche

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

c est choisi de manière à obtenir un filtre normalisé

Ce filtre a été proposé récemment par R. Deriche [7] 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 (1) aux opérateurs infinis et antisymétriques et en dérivant ainsi une solution exacte à l'équation de Canny :

$$f(x) = -c \times e^{-\alpha|x|} \times \sin \omega x$$

α, ω, c réels positifs.

Si on pose :

$$\alpha = m \cdot \omega$$

on obtient :

$$(a) \quad m \gg 1 \quad \lambda = (2\alpha)^{\frac{1}{2}} \quad \sum \approx \left(\frac{2}{\alpha}\right)^{\frac{1}{2}} \quad \sum \cdot \lambda \approx 2 \quad k \approx .44$$

$$(b) \quad m \ll 1 \quad \lambda = (2\alpha)^{\frac{1}{2}} \quad \sum \approx \frac{\lambda}{m} \quad \sum \cdot \lambda \approx 2 \cdot m \quad k \approx 1$$

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

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

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 ($\sum \cdot \lambda = 1.112, k = .58$) est moins bonne que l'opérateur f de plus de 25 %.

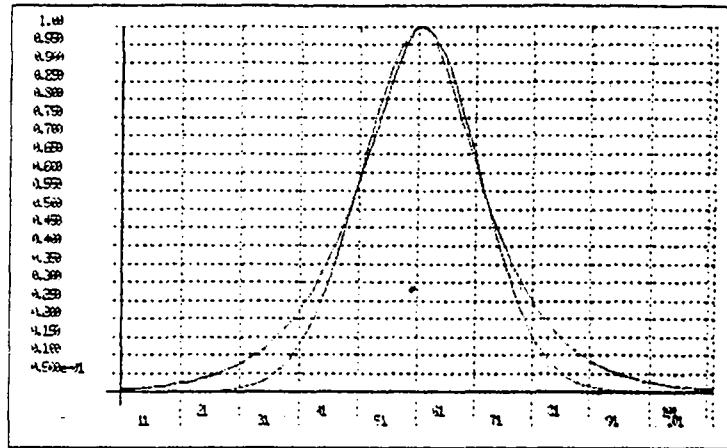


Figure 12 : Réponses impulsionnelles de s_3 et de s_4

Le cas (b) présente une réponse idéale pour k mais le produit $\sum \lambda$ devrait être beaucoup plus petit que l'unité.

Le cas (a) présente le meilleur compromis. On peut alors écrire ce filtre optimal comme :

$$g(x) = -kx e^{-\alpha|x|}$$

2.3.4 Filtres de dérivation

Les filtres de dérivation correspondant aux filtres de lissage sont obtenus simplement par dérivation des réponses impulsionnelles. 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 soi il faudrait mettre en œuvre des opérateurs de détection des discontinuités d'ordre 1.

1. Filtre Shen

$$\begin{aligned} s_1'(x) &= c \cdot e^{-\alpha|x|} \text{ si } x \geq 0 \\ &= -c \cdot e^{-\alpha|x|} \text{ si } x \leq 0 \end{aligned}$$

2. Filtre moyenne

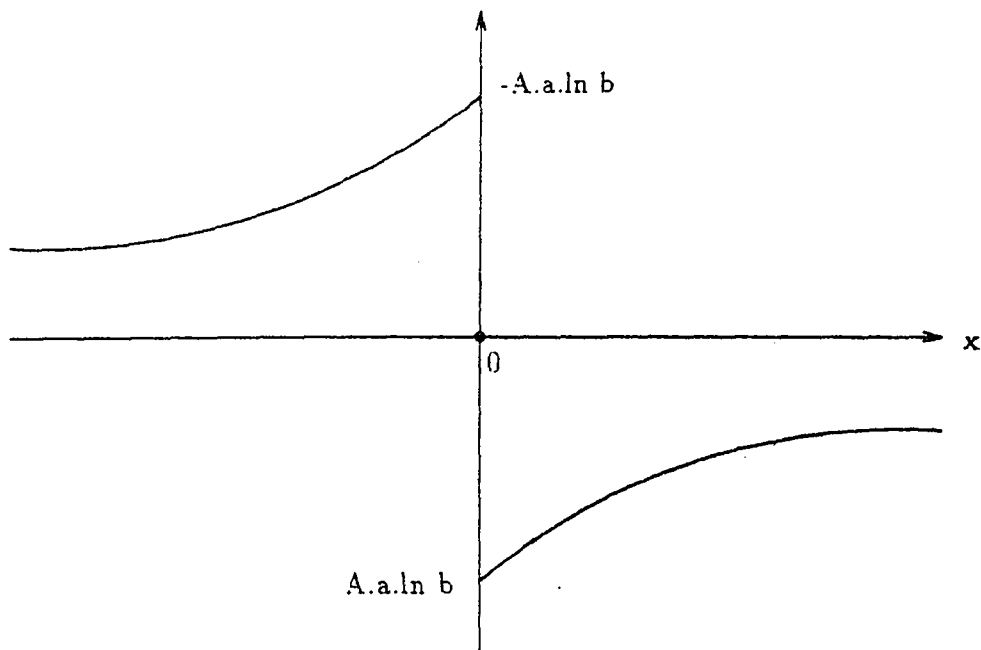


Figure 13 : Réponse impulsionnelle de s'_1

$$\begin{aligned}
 s'_2(x) &= \frac{-1}{(2.n+1)} \quad \text{pour } x \text{ dans } [-2.n, 0] \\
 &= \frac{1}{(2.n+1)} \quad \text{pour } x \text{ dans } [0, 2.n] \\
 &= 0 \quad \text{sinon}
 \end{aligned}$$

Dans ce cas la dérivée de s_2 est obtenue par différences finies.

3. Filtre gaussien

$$s'_3(x) = \frac{-x}{\sigma_2} \times \frac{e^{-x^2}}{2\sigma^2}$$

4. Filtre Deriche

$$s'_4(x) = -c \times x e^{-a.a.|x|}$$

2.3.5 Filtres de calcul de la dérivée seconde

Ces filtres 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 remarquons 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 si il n'existe pas

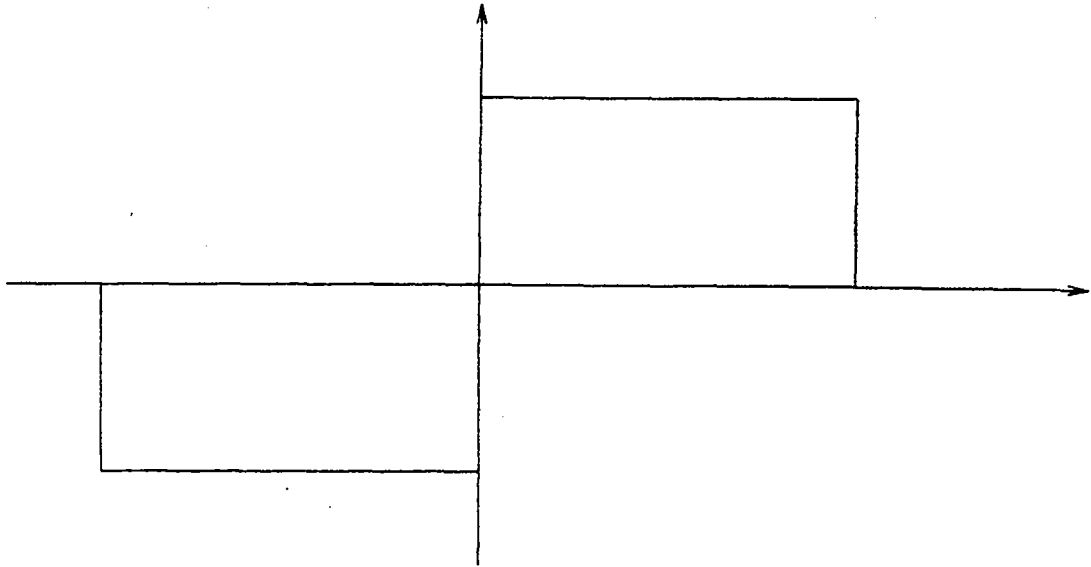


Figure 14 : Réponse impulsionnelle de s'_2

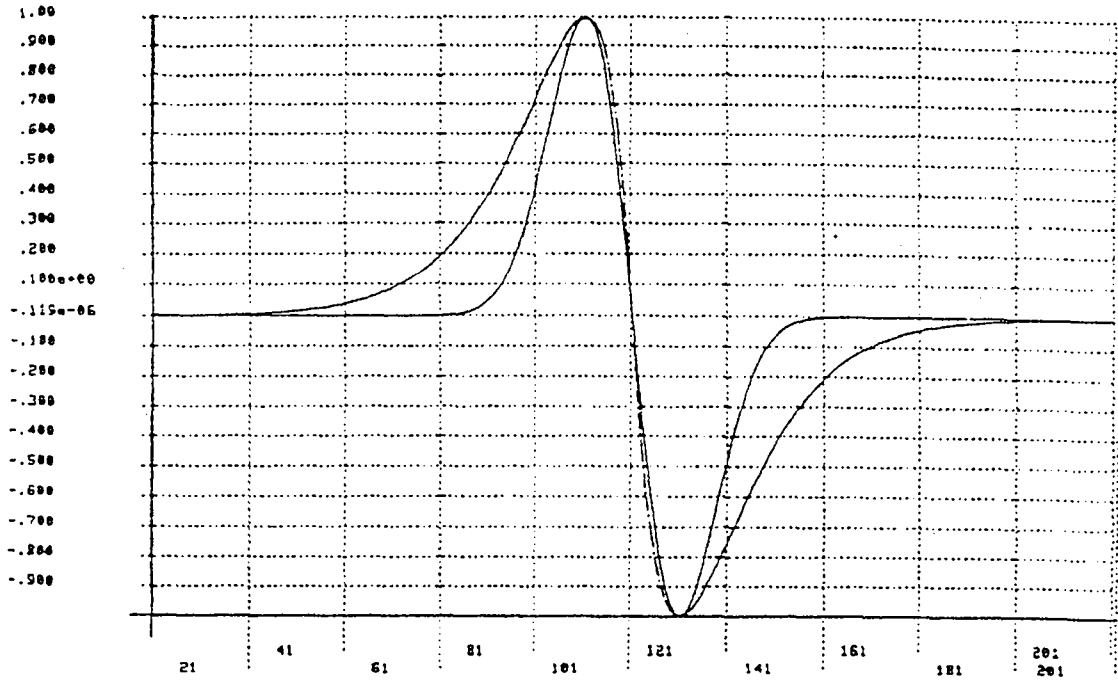


Figure 15 : Réponses impulsionnelles de s'_3 et s'_4

d'expression analytique des réponses impulsionnelles, il est toujours possible de les implanter par cascade de deux filtres de dérivation.

2.3.6 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'implémentation classique consiste à tronquer les réponses de manière à ne conserver que les coefficients significatifs (les filtres de lissage et de dérivation 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 $8 \times \sigma$ et pour un filtre Shen ou un filtre Deriche de paramètre 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 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 introduit récemment en vision artificielle pour la détection des contours [7, 33, 34]. Il permet d'implémenter des filtres de réponse impulsionnelle infinie avec un coût faible. Il se trouve que le filtre Shen et le filtre Deriche (filtres de lissage et de dérivation) admettent des implantations récursives respectivement d'ordres 1 et 2. Le filtre gaussien peut être approximé par des filtres récursif d'ordre 4 [8]. Dans cette partie nous montrerons comment obtenir la mise en œuvre récursive du filtre de dérivation Deriche et du filtre de lissage Shen.

Nous allons montrer que le filtre Deriche peut s'implanter par un filtrage récursif d'ordre 2 [7].

$$\text{Soit } s'_4(x) = f(x) = -c \cdot x \cdot e^{-a|x|}$$

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 :

$$f(n) = f_-(n) + f_+(n) \text{ où}$$

$$f_-(n) = \begin{cases} 0 & n \geq 0 \\ \alpha_1 \cdot (p_2)^n + \alpha_1^* \cdot (p_2^*)^n & n, 0 \end{cases}$$

$$f_+(n) = \begin{cases} \alpha_1 \cdot (p_1)^n + \alpha_2 \cdot (p_1^*)^n & n \geq 0 \\ 0 & n, 0 \end{cases}$$

$$\alpha_1 = \frac{-c}{2ip_1} = e^{-\alpha+i\omega} \quad p_2 = e^{\alpha+i\omega}$$

α^* : conjugué de α

En utilisant la transformée en Z , on reconnait que chaque facteur $\alpha_i(p_i)^n$ a une transformée $\frac{\alpha_i}{(1-p_i Z^{-1})}$ et on obtient :

$$F(Z) = F_-(Z) + F_+(Z-1) \text{ où}$$

$$F_-(Z-1) = \frac{a \cdot Z^{-1}}{(1+b_1 \cdot Z^{-1} + b_2 \cdot Z^{-2})}$$

$$F_+(Z) = \frac{-aZ}{(1+b_1 \cdot Z + b_2 Z^2)}$$

avec :

$$a = -c \cdot e^{-\alpha} \cdot \sin \omega$$

$$b_1 = -2 \cdot e^{-\alpha} \cdot \cos \omega$$

$$b_2 = e^{-2 \cdot \alpha}$$

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 :

$$y^+(m) = a \cdot x(m-1) - b_1 y^+(m-1) - b_2 y^+(m-2) \quad m = 1, \dots, N$$

$$y^-(m) = -a \cdot x(m-1) - b_1 y^-(m+1) - b_2 y^-(m+2) \quad m = N, \dots, 1$$

$$y(m) = y^+(m) + y^-(m) \quad m = 1, \dots, N$$

La récursivité du filtre permet une implantation qui requiert seulement 6 multiplications et 5 additions par point et cela avec une taille 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'implémente par filtrage récursif d'ordre 1 :

$$s_1(x) = f(x) = a \cdot e^{-\alpha \cdot |x|} = c_1 \cdot e^{|\alpha| \cdot \text{Log}(c_2)}$$

$$f(n) = f^+(n) + f^-(n)$$

$$y^+(n) = y^+(n-1) + a_0 \times (x(n) - x(n-1))$$

$$y^-(n) = y^-(n+1) + a_0 \times (x(n) - x(n+1))$$

$$f^+(x) = \begin{cases} a_0(1 - a_0)^u & u \geq 0 \\ 0 & u < 0 \end{cases}$$

$$f^-(x) = \begin{cases} 0 & u > 0 \\ a_0(1 - a_0)^{-u} & u \leq 0 \end{cases}$$

$$c_1 = \frac{a_0}{2 - a_0}$$

$$c_2 = 1 - a_0$$

2.4 Algorithme de détection de contours *nd*

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

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

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

$$s(x) = s_4(x) = (c \cdot |x| + 1) \cdot e^{-\alpha|x|} \quad \text{filtre Deriche}$$

où

$$s(x) = s_1(x) = a \cdot e^{-\alpha|x|} \quad \text{filtre Shen}$$

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 [32]. 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 (discontinue ou continue 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 lorsqu'on calcule 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 exactement la même. On remarque que l'approche laplacien a tendance à arrondir les angles. 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 : $s(x)$ et $s'(x)$. On choisira de préférence des implantations récursives. Des exemples sont donnés dans la section précédente.

Soit $I(x_1, \dots, x_n)$ une image de dimension n

Soit $G(x_1, \dots, x_n)$ le gradient de I

$$G(I) = \left(\frac{dI}{dx_1}, \frac{dI}{dx_2}, \dots, \frac{dI}{dx_n} \right)^t$$

Le calcul des composantes du gradient $\frac{dI}{dx_i}$ s'effectue en calculant les image (D_i) correspondant aux dérivées partielles des points par rapport à x_i de la manière suivante :

pour i allant de 1 à i faire

$$D_i = I;$$

pour k allant de 1 à $i - 1$ faire

$$D_i = D_i * s(x_k);$$

finpour

$$D_i = D_i * s'(x_i)$$

pour k allant de $i - 1$ à n faire

$$D_i = D_i * s(x_k)$$

finpour

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

- **Calcul du laplacien**

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

$$\begin{aligned} L(x_1, x_2, \dots, x_n) &= \sum_{i=1, n} (s(x_1) \dots s(x_{i-1}) \times s''(x_i) \cdot s(x_{i+1}) \dots s(x_n)) \\ &= \prod_{i=1, n} L_i(x_i) \end{aligned}$$

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 [7] ou 3d [22].

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.

2.5 Résultats expérimentaux

La méthodologie que nous avons décrite dans la section précédente a été appliquée aux cas 2d [7], 3d [22], et 4d [31] 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 16 à 25). Sur une station de travail SUN-3 le temps de calcul pour une image 512.512 est de l'ordre de 30 secondes CPU.

2.6 Traitements en aval

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 de contours chaînés. Nous décrivons un exemple d'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 :

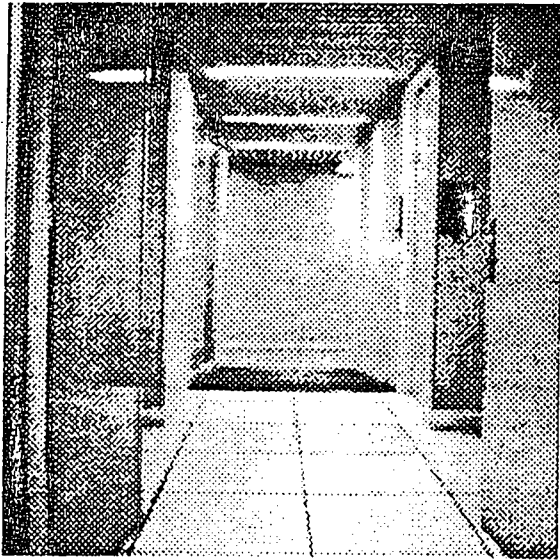


Figure 16 : Image originale

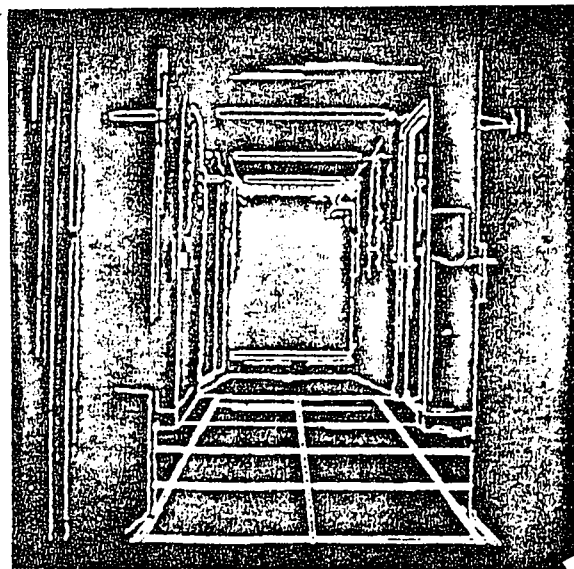


Figure 17 : Contours obtenus par le filtre Deriche

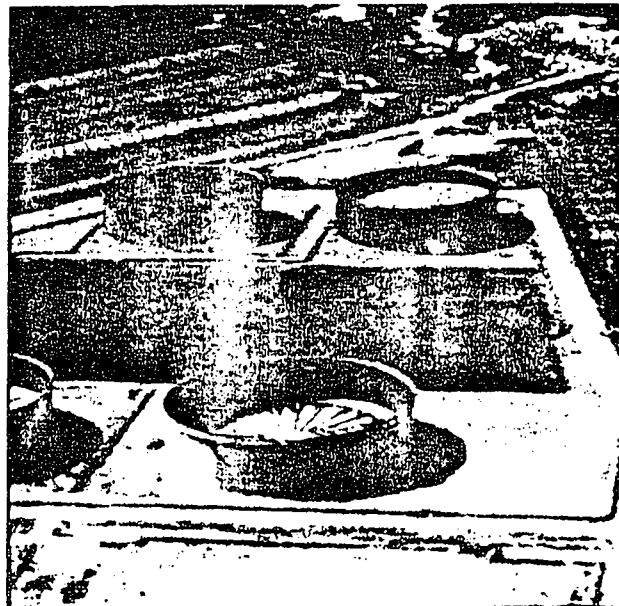


Figure 18 : Image originale

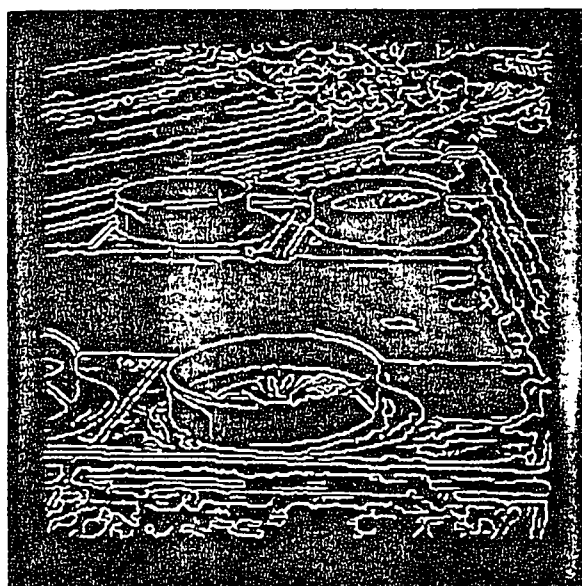


Figure 19 : Contours obtenus par le filtre Deriche

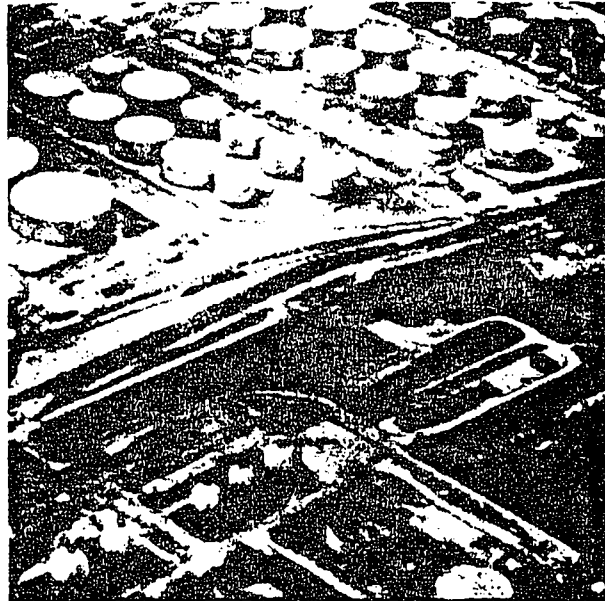


Figure 20 : Image originale

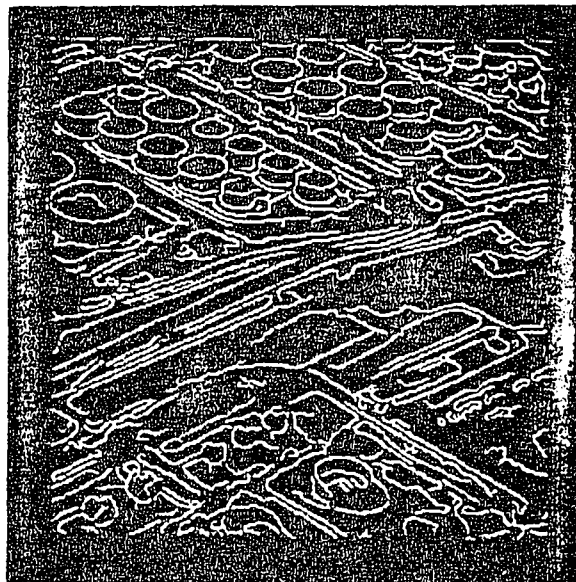


Figure 21 : Contours obtenus par le filtre Deriche

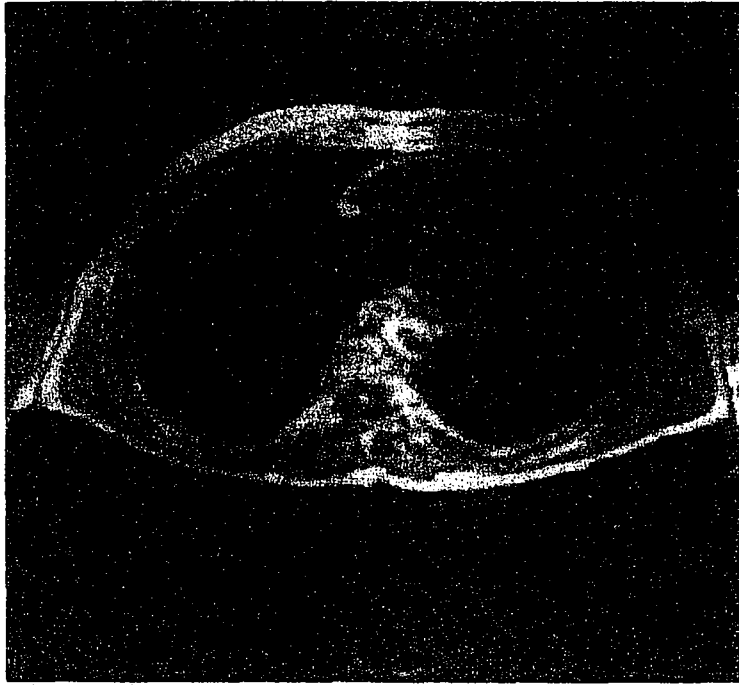


Figure 22 : Coupe d'une image RMN 3d de cœur

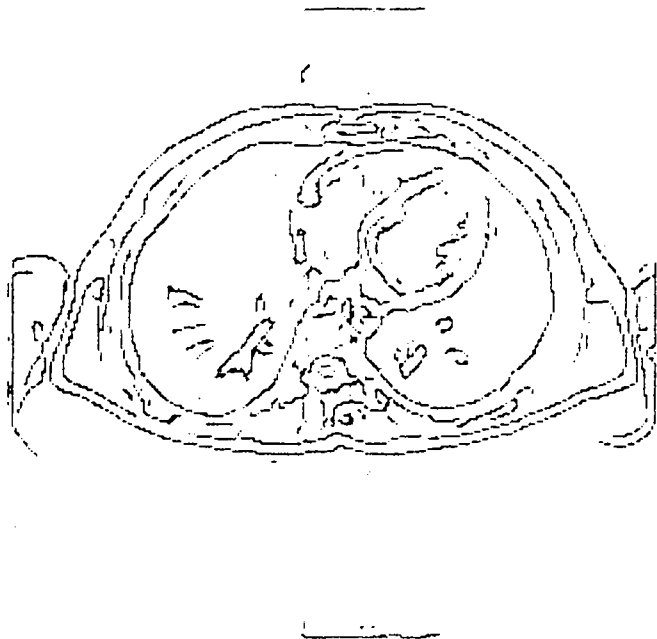


Figure 23 : Contours 3d obtenus par le filtre Shen



Figure 24 : Coupe d'une image RMN 3d de cerveau



Figure 25 : Contours obtenus par le filtre Shen

- 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.

G. Giraudon [11] a mis au point un algorithme de chaînage qui réduit 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, il a 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 des segments de contours [2].

Les figures 26 à 28 présentent un exemple de résultats obtenus sur des scènes d'intérieur.

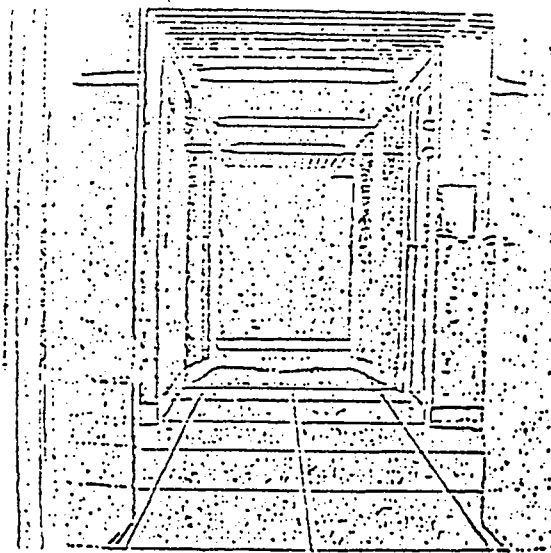


Figure 26 : Image de contours

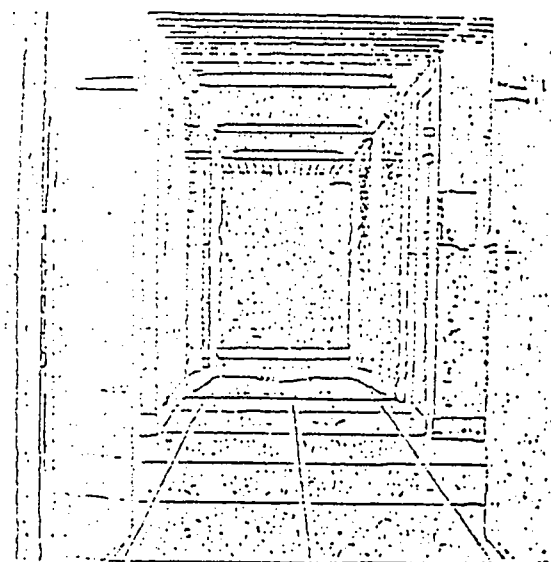


Figure 27 : Traitement chaînage seul
nombre de pixels traités = 20 855
nombre de chaînes créées = 4 263

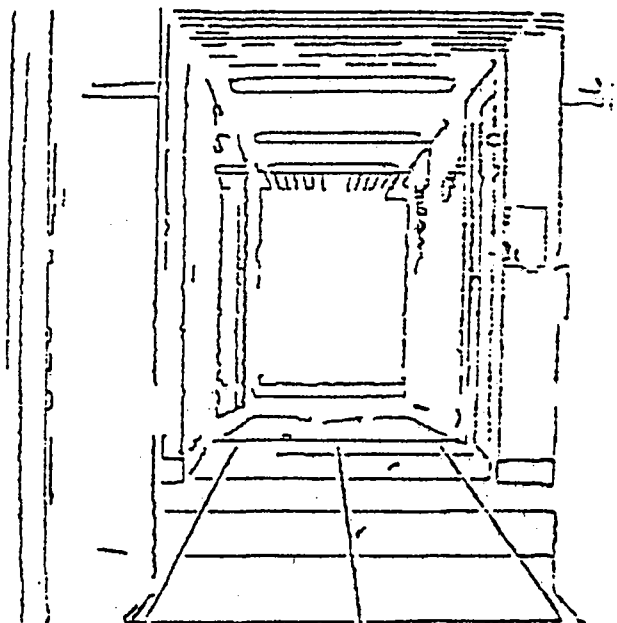


Figure 28 : Traitement complet
seuil de longueur des chaînes = 15
nombre de chaînes fusionnées = 1 071
nombre de chaînes éliminées = 2 900
nombre de chaînes restantes = 292

3 Extraction de régions

3.1 Introduction

L'objectif de la segmentation d'images en régions est de partitionner une image en zones d'intérêt correspondant à des objets de la scène d'où elle est issue. Elle peut être située dans le cadre plus général de la segmentation de données.

Le problème de base de la segmentation de données est le suivant :

On connaît :

- Un ensemble d'entités
- Un ensemble d'attributs caractérisant ces entités
- Des relations topologiques entre ces entités
- Des attributs relationnels entre ces entités

On cherche :

- Une (ou des) partition(s) de ces données ayant des propriétés intéressantes par rapport aux attributs et aux relations topologiques.

Les principaux problèmes qui se posent sont :

- Définir les **propriétés des partitions** que l'on cherche
- Concevoir des **algorithmes** permettant l'obtention de **partitions optimisant ces propriétés**.

Pour le problème spécifique de la segmentation d'images :

- Les **entités** sont des **points d'images** par exemple bidimensionnelles ou tridimensionnelles.
- Les **attributs** sont la **position dans l'espace** et les **luminances**.
- Les **relations topologiques** sont la **4-connexité** ou la **8-connexité**.
- Les **attributs relationnels** sont les caractéristiques des **frontières** entre deux ensembles d'entités.

On peut donc caractériser un problème de **segmentation d'images** par :

- Un **ensemble de critères d'homogénéité** déterminant les propriétés des partitions de l'image que l'on cherche : *modélisation analytique*.
- Un **algorithme** utilisant ces critères de manière à segmenter l'image : *modélisation algorithmique*.

On est donc confronté à deux problèmes de base distincts :

- La **définition de critères de segmentation** spécifiques à chaque problème particulier de segmentation.
- La **définition d'une stratégie d'utilisation des critères** de segmentation qui peut être définie de manière générale.

On peut noter que dans le cas où les propriétés d'homogénéité des régions s'expriment simplement, il est possible de dériver un opérateur d'extraction de contour qui détecte les frontières entre les régions. On peut ensuite déterminer les régions par fermeture des contours ainsi obtenus. Un exemple de ce type d'approche est présenté dans la section III.5. Mais souvent la recherche de filtres aisément implémentables permettant de détecter les ruptures d'une propriété d'homogénéité d'ordre supérieur à 0 reste complexe. Ce constat a motivé le nombre important de travaux consacrés à la segmentation d'images en régions.

Les méthodes existantes peuvent être rangées dans deux catégories : les méthodes de "classification", et les méthodes de "croissance de régions". La différence principale entre ces deux types d'approches réside dans la stratégie d'utilisation de l'espace des luminances et des relations spatiales existant entre les pixels. Les méthodes de classification déterminent d'abord une partition de l'espace des luminances et se servent ensuite des relations de connectivité pour déterminer les régions. Les méthodes de croissance de régions utilisent de manière simultanée ces deux types d'information.

3.2 Problématique : un problème d'optimisation NP difficile

L'objectif de cette partie est de montrer que la segmentation en régions dans des images de dimension quelconque se ramène à un problème d'optimisation. Nous verrons qu'il est le plus souvent NP difficile ce qui nécessite l'introduction d'heuristiques auxquelles est consacré la section suivante.

La segmentation d'une image I utilisant un prédicat d'homogénéité P est communément définie [39] comme une partition $S = R_1, R_2, \dots, R_n$ de I telle que :

1. $I = \cup R_i, i \in [1 \dots n]$
2. R_i est connexe, $\forall i \in [1 \dots n]$
3. $P(R_i) = \text{vrai}, \forall i \in [1 \dots n]$
4. $P(R_i \cup R_j) = \text{faux}, \forall i \neq j$, pour tout couple (R_i, R_j) de régions connexes.

Il est important de remarquer que les conditions 1, 2, 3, 4 ne définissent en général pas une segmentation unique. Les résultats de segmentation dépendent par conséquent de l'ordre et de la manière avec lesquelles les données sont traitées et non pas seulement de l'information contenue dans l'image. Par exemple, dans beaucoup de méthodes une segmentation calculée sur la transposée d'une image n'est pas similaire à la transposée de la segmentation de l'image.

Il est possible de réduire ce problème d'indétermination en ajoutant une contrainte d'optimisation d'une fonction C caractérisant la qualité d'une segmentation [20].

Soit Q une fonction caractérisant la qualité d'un élément de S (par exemple la variance des niveaux de gris des pixels d'une région).

Soit C une fonction monotone et symétrique par rapport à la qualité de chaque élément de S : $C(S) = C(Q(R_1), \dots, Q(R_n))$ (par exemple la moyenne arithmétique de $Q(R_1), \dots, Q(R_n)$).

Nous ajoutons aux axiomes 1 \rightarrow 4 le suivant :

5. Parmi toutes les segmentations possibles S vérifiant 1, 2, 3,

Nous recherchons la (ou une) segmentation S^* qui optimise la fonction de qualité c'est-à-dire telle que :

$$C(S^*) \leq C(S) \quad \forall S \in S_P(I)$$

où $S_P(I)$ est l'ensemble des partitions de I , et C une fonction décroissante.

Très souvent C est additive, elle peut être par exemple dans le cas d'images 2-D la fonction C_1 définie comme suit :

$$C_1(S) = \sum_{i=1, n} \sum_{(k, l) \in R_i} (I(k, l) - M_i)^2 ; n \text{ étant le nombre des régions};$$

$$M_i = \frac{\sum_{(k, l) \in R_i} I(k, l)}{\text{Card}(R_i)} ; \text{Card}(R_i) \text{ étant le nombre de points d'une région } R_i.$$

Il est important de remarquer que généralement les propriétés désirées pour les régions peuvent être décomposées hiérarchiquement de manière à simplifier leur optimisation [20]. Une suite $(P_1, C_1), \dots, (P_n, C_n)$ sera ainsi utilisée au lieu d'un simple couple (P, C) . Une telle décomposition permet de réduire, en le divisant le problème d'optimisation. Cette nécessité peut s'illustrer par un exemple simple. Supposons que nous voulions déterminer des angles droits à partir d'un ensemble de points. Dans une première étape, nous utiliserons un couple (P_1, C_1) permettant l'obtention de segments de droites à partir des points. Dans une deuxième étape, nous utiliserons un couple (P_2, C_2) permettant la formation d'angles droits avec les segments.

Il est clair que la condition 5 ne définit pas toujours une segmentation unique. En effet plusieurs segmentations ayant la même valeur minimale de C peuvent exister. Quoiqu'il en soit, cette condition permet d'obtenir une définition plus précise mais pose un problème algorithmique sous-tendant une complexité importante. C'est pourquoi la recherche d'algorithmes sous-optimaux est nécessaire.

3.3 Segmentation en régions par classification

Ces méthodes déterminent d'abord une partition de l'espace des luminances en utilisant les niveaux de gris présents dans l'image. On associe à chaque pixel la classe de niveaux de gris à laquelle il appartient. Les régions sont définies par les ensembles maximaux de pixels connexes appartenant à la même classe. Les prédicats d'homogénéité et les fonctions de qualité (voir paragraphe précédent) ne sont pas définis explicitement dans ce type d'algorithme.

Le plus souvent la classification des luminances s'effectue à partir du calcul de l'histogramme de répartition dans l'image. On recherche les différents modes de l'histogramme et les "vallées" correspondantes. Les classes sont déterminées par les intervalles entre les vallées. Cette procédure fonctionne bien pour des images comprenant un nombre peu important d'objets ayant des niveaux d'intensité différents. Olhander [29] a amélioré cette idée en introduisant une classification récursive permettant d'opérer quand un nombre important d'objets est présent dans l'image. On définit un masque sélectionnant tous les pixels de l'image (un masque définit une zone de l'image). Pour chaque masque un histogramme de l'image masquée est calculé. Les modes de l'histogramme sont détectés permettant ainsi de segmenter l'espace des niveaux de gris. Les points de l'image sont étiquetés avec la classe correspondante. Les ensembles maximaux de pixels connexes appartenant à une même classe sont déterminés. Si l'histogramme comprend plus qu'un mode alors le masque est terminé. Sinon tous les ensembles connexes précédemment calculés sont utilisés pour générer des masques qui sont placés dans une pile de masques. Durant les itérations successives le masque courant détermine les pixels utilisés pour le calcul d'histogrammes. La classification est répétée pour chaque nouveau masque jusqu'à ce que la pile soit vide. Le nombre des modes de l'histogramme déterminés à chaque étape est un paramètre de cette procédure ; par exemple si ce nombre est réduit à l'unité on détermine à chaque étape le meilleur pic. Ohta, Kanade et Sakai [28] ont adapté cet algorithme pour la segmentation d'images couleur.

Dans le cas où l'image contient un objet se détachant sur un fond des techniques de classification par seuillage adaptatif ont été développées [6], [35]- [37]. Ces méthodes combinent l'information spatiale de l'image avec l'information de luminance pour aider la détermination des seuils. L'itération de ce type d'algorithme peut permettre la segmentation d'images comprenant plusieurs objets.

Chow et Kaneko [6] calculent en chaque point un seuil dépendant de l'histogramme de répartition des luminances sur son voisinage. Des voisinages carrés 33.33 ou 65.65 sont par exemple utilisés pour la détermination de ces "histogrammes locaux". De manière à éviter le calcul d'un histogramme pour chaque point Chow et Kaneko partitionnent l'image en blocs, calculent l'histogramme pour chaque bloc, déterminent un seuillage approprié pour chaque histogramme, et ensuite interpolent spatialement les valeurs des seuils de manière à obtenir un seuil adaptatif en chaque pixel.

Weszca, Nagel et Rosenfeld [37] suggèrent de déterminer un histogramme en prenant seulement en compte les pixels de Laplacien faible. Seules les luminances à l'extérieur des zones de fort gradient donc a priori à l'intérieur de zones homogènes peu bruitées sont ainsi prises en compte.

Watanabe [35] propose de choisir une valeur de seuil qui maximise la somme des gradients calculée sur tous les pixels dont le niveau de gris est égal à la valeur du seuil.

Ces méthodes de "segmentation par classification" se révèlent efficaces si la classification des luminances permet de mettre en évidence les différentes régions homogènes de l'image. Dans le cas d'images comprenant des objets de luminances différentes se détachant sur un fond cette approche donne de bons résultats. Mais lorsque les images sont bruitées et contiennent un nombre important d'objets la classification se révèle peu utilisable. Dans ce cas il est nécessaire d'utiliser les relations spatiales tout au long du processus de segmentation. Cette idée a été développée dans les méthodes dites de "croissance de régions".

3.4 Segmentation par croissance de régions

3.4.1 Introduction

Ces techniques consistent à regrouper itérativement des ensembles de points connexes en régions plus importantes en utilisant des conditions dépendant de propriétés d'homogénéité. L'idée de base de ce type d'approche est de définir des critères de regroupement des pixels permettant l'obtention de régions homogènes. Les relations spatiales sont ainsi utilisées tout au long du processus de segmentation. Ces méthodes peuvent être subdivisées en deux classes : "agrégation de points" et "regroupement itératif d'ensembles de points".

3.4.2 Agrégation de points

Ces méthodes associent à chaque pixel un vecteur de propriétés. Deux pixels sont regroupés si leurs vecteurs de propriétés sont "suffisamment similaires". Le résultat de la segmentation est constitué par les composantes connexes déterminées. Divers vecteurs de propriétés et diverses mesures de similarité ont été proposées [1, 12, 13].

L'heuristique la plus simple consiste à réunir deux pixels si leur différence de niveau de gris est suffisamment faible. Bryant [4] normalise cette différence par la moyenne des différences calculées sur tous les couples de pixels voisins de l'image. Asano et Yokoya [1] regroupent deux pixels si leur différence de niveaux de gris est faible par rapport à la plus grande différence existant entre chacun des pixels et ses voisins pris dans un petit voisinage carré.

Nagao et Matsuyama utilise ce type d'algorithme pour la segmentation d'images couleur [27]. Un pixel est regroupé avec un noyau de région si la norme *SUP* entre son vecteur de composantes (dans R, V, B) et un vecteur de composantes quelconque d'un point du noyau est majorée par un seuil. Une expansion point à point permet d'obtenir un ensemble de régions dont l'amplitude de variation des intensités des points selon les composantes rouge (R), vert (V), bleu (B) est majorée par le seuil.

Des critères de regroupement plus sophistiqués attachant à chaque pixel un vecteur de propriétés dépendant d'un voisinage $k \times k$ autour du pixel ont été employés. Souvent

le vecteur propriété est issue d'un opérateur de détection de contour [13, 12]. Les pixels non séparés par un contour sont associés. Les résultats obtenus dépendent évidemment du détecteur de contour utilisé.

Les limitations de ce type d'approche par aggrégation de points sont liées au fait que les entités que l'on regroupe (les pixels) véhiculent peu d'informations. En effet ces méthodes consistent à déterminer des noyaux de régions et ensuite à regrouper itérativement des points à ces noyaux. Les seules informations dont on dispose pour décider d'un regroupement sont les caractéristiques du noyau de régions et la luminance du point. Les méthodes par regroupement itératif d'ensembles de points permettent d'utiliser des informations plus riches pour former les régions.

3.4.3 Regroupement itératif d'ensemble de points

Introduction

L'idée de ces méthodes est de définir une succession de partitions de l'image par regroupement itératif de régions connexes. Cette stratégie permet de définir des heuristiques de regroupement dépendant de la similarité des propriétés des régions.

La première approche de ce type a été proposée par Muerle et Allen [26]. Ils suggèrent de regrouper deux régions adjacentes si leurs distributions de niveaux de gris sont suffisamment similaires. Ils recommandent d'utiliser le test de Kolmogorov-Smirnov.

Brice et Fennema [3] effectuent la croissance de régions en partitionnant l'image en ensembles initiaux de points de même intensité. Ils regroupent ensuite séquentiellement les couples de régions adjacentes dont une fraction significative de la frontière a un contraste faible. Horowitz et Pavlidis utilisent un "quad tree" pour représenter l'image. Une stratégie de fusion et d'explosion ("split and merge") utilisant le quad-tree est ensuite mise en œuvre [15]. On obtient ainsi un ensemble de régions dont l'amplitude de variation des luminances est majorée par un seuil.

La méthode développée par Pong et al. [30] est basée sur l'utilisation séquentielle de plusieurs algorithmes : l'image est d'abord segmentée en utilisant un "sloped facet model" [14] ; ensuite la segmentation initiale sert d'entrée à un algorithme de croissance de régions. Deux régions adjacentes sont fusionnées si un critère de similarité est vérifié.

La plupart de ces méthodes regroupent itérativement tous les couples de régions adjacentes vérifiant certaines conditions. Le processus de croissance s'arrête quand plus aucun couple de régions adjacentes ne vérifie les conditions de fusion. Généralement le résultat obtenu dépend de l'ordre avec lequel les couples de régions sont regroupés. Ainsi le principe des méthodes les plus efficaces est de définir une hiérarchie de fusion.

Croissance hiérarchique de régions

Algorithme On décrit un algorithme sous-optimal illustrant la définition de la segmentation proposée précédemment. L'idée de base de cet algorithme est d'optimiser la qualité globale de la segmentation (S) par une optimisation locale [20]. Il consiste à regrouper parmi tous les couples de régions dont la réunion vérifie le prédicat (P) celui ayant la meilleure qualité locale (Q). Il peut être décrit comme suit :

$$S = I$$

tant qu'un couple de régions connexes dont l'union vérifie P existe faire

- regrouper parmi tous les couples (R_i, R_j) dont l'union vérifie P celui minimisant $Q(R_i \cup R_j)$
- mettre à jour S

tant que

Cet algorithme de regroupement optimise l'estimation globale de la qualité de la segmentation par optimisation locale de Q . Le fait que le coût de fusion (Q) soit optimisé sur l'image entière permet au processus de fusion d'être fortement guidé par les données. On ne définit pas de sens de parcours ou de traitement des données. L'ordre avec lequel l'image est traitée n'est pas déterminé a priori comme dans la plupart des algorithmes de segmentation. Cet algorithme peut être itéré pour utiliser une suite de critères de fusion : $(P_1, Q_1) \dots (P_n, Q_n)$.

Au premier abord cet algorithme semble sous-tendre une complexité importante. En effet le choix du meilleur couple de régions connexes par rapport à (P_i, Q_i) amène en général des implantations coûteuses. Or si certaines conditions relatives aux (P_i, Q_i) sont vérifiées, une implantation de faible complexité algorithmique est possible [20]. On va maintenant décrire ces propriétés. On donnera tout d'abord quelques définitions, et ensuite les hypothèses nécessaires sur les (P_i, Q_i) .

Soit $E = e_1, e_2, \dots, e_n$ une partition initiale de l'image ; E peut être l'image ou une segmentation initiale.

Soit $P(E)$ l'ensemble des sous-ensembles de E .

Soit R l'ensemble des réels.

Soit k le nombre d'attributs associés à un sous-ensemble de E .

Soit A une fonction de $P(E)$ vers R^k qui associe un vecteur d'attributs à un sous-ensemble de E (par exemple pour une image de niveaux de gris : l'aire, la moyenne ou la variance de niveau de gris, ...).

Soit m le nombre d'attributs associés à un couple de sous-ensembles de E .

Soit G une fonction de $P(E)^2$ vers R^m qui associe un vecteur d'attributs à un couple de sous-ensembles de E (Par exemple pour une image de niveaux de gris ; le gradient moyen à la frontière).

Soit S une partition de E , $S = \{E_1, E_2, \dots, E_p\}$.

On associe à S l'ensemble des attributs de ses éléments et l'ensemble des attributs de ses paires d'éléments respectivement $V(S)$ et $W(S)$:

$$V(S) = \{A(E_1), A(E_2), \dots, A(E_p)\}$$

$$W(S) = \{G(E_r, E_s); r, s \in \{1 \dots p\}\}$$

Chaque triplet $(S, V(S), W(S))$ définit et caractérise une segmentation de E .

On définit l'action consistant à fusionner deux éléments de S E_i et E_j . Cette opération consiste à définir un nouveau triplet :

$$(S, V(S), W(S))_{(i,j)} = (R, V(R), W(R))$$

$$R = \{E_1, \dots, E_{i-1}, E_i \cup E_j, E_{i+1}, \dots, E_{j-1}, E_{j+1}, \dots, E_{p-1}\}$$

$$V(R) = \{A(E_1), \dots, A(E_{i-1}), A(E_i \cup E_j), \dots, A(E_{p-1})\}$$

$$W(R) = \{G(E_p, E_q), p \neq i, q \neq j, G(E_q, E_i \cup E_j) \dots\}$$

On suppose que les conditions (A_1) et (A_2) suivantes sont vérifiées :

(A_1) : La suite de couples : (Prédicat de fusion, Coût de fusion) = (P, Q) définissant les heuristiques de regroupement est telle que :

$\forall E, F \in S$ nous avons :

$$P(E \cup F) = P(A(E), A(F), G(E, F))$$

$$Q(E \cup F) = Q(A(E), A(F), G(E, F)).$$

Cela signifie que le prédicat d'homogénéité et la fonction de qualité de la réunion de deux régions peuvent être calculés à partir de leurs attributs.

(A_2) : Il existe deux fonctions T et U telles que :

T est une fonction de $(R^k)^2$ vers R^k telle que :

$$\forall S_1, S_2 \in P(E); A(S_1 \cup S_2) = T(A(S_1), A(S_2))$$

Cela signifie que les attributs de la réunion de deux régions peuvent être calculés à partir des attributs des deux régions.

U est une fonction de $(R^m)^2$ dans R^m telle que :

$$\forall S_1, S_2, S_3 \in P(E); G(S_1, S_2 \cup S_3) = U(G(S_1, S_2), G(S_1, S_3))$$

Cela signifie que les attributs du couple $(S_1, S_2 \cup S_3)$ peuvent être calculés à partir de ceux de (S_1, S_2) et (S_1, S_3) .

Les fonctions T et U permettent de mettre à jour aisément $(S, V(S), W(S))$ après une fusion. Ainsi si un triplet initial $(S, V(S), W(S))$ et les fonctions T et U sont données, nous pouvons effectuer une suite de fusions en mettant à jour itérativement le triplet courant.

Les conditions (A_1) et (A_2) permettent de réaliser une implantation de l' algorithme dont la complexité est faible. Dans beaucoup de problèmes de segmentation on peut trouver des critères de segmentation qui vérifient ces conditions.

Le problème principal posé par la condition (A_1) est que le prédicat ainsi que le coût de fusion sont uniquement des fonctions du couple de régions. En effet, aucune information contextuelle (concernant, par exemple, les attributs des régions voisines) ne peut être insérée dans un couple (P_i, Q_i) vérifiant (A_1) . Pour le cas de l'optimisation d'une fonction globale de qualité C_i monotone et symétrique par rapport aux qualités locales Q_i , cette restriction n'est pas gênante. En effet, une telle estimation de la valeur d'une segmentation ne nécessite pas pour être optimisée d'information contextuelle.

La caractérisation des attributs induite par (A_2) implique une restriction sur la nature des attributs utilisés. L'existence de la fonction T signifie que les attributs associés à une région peuvent être calculés à partir des attributs associés à un ensemble quelconque de régions formant une partition de la région. L'existence de la fonction U implique que les attributs associés à un couple de régions peuvent être déterminés à partir de ceux des couples du produit cartésien de deux partitions quelconques des deux régions. Pratiquement, quand un attribut intéressant pour un problème de segmentation ne vérifie pas la condition (A_2) nous cherchons à l'exprimer en fonction d'attributs la vérifiant. D'un point de vue théorique une telle décomposition existe toujours ; le pire des cas étant quand elle utilise tous les points de la région. Nous avons évidemment intérêt à employer des attributs qui peuvent être exprimés avec un nombre réduit d'attributs vérifiant (A_2) . Nous allons illustrer ces remarques par le cas simple de la variance des niveaux de gris souvent utilisée pour la segmentation des images naturelles. La variance des niveaux de gris de deux régions R_1 et R_2 n'est pas uniquement fonction de la variance de R_1 et de la variance de R_2 . Cependant elle peut être calculée si

on connaît l'aire, la somme des niveaux de gris, et la somme des carrés des niveaux de gris des deux régions R_1 et R_2 . Ces trois attributs vérifient la propriété (A_2) en effet le vecteur attribut incluant respectivement ces trois attributs est associé à la fonction T suivante : $T((x_1, x_2, x_3), (y_1, y_2, y_3)) = (x_1 + y_1, x_2 + y_2, x_3 + y_3)$. Ainsi si nous voulons utiliser la variance des niveaux de gris, pour segmenter des images naturelles 2-D par exemple, nous incluons dans les attributs associés aux régions : l'aire, la somme des niveaux de gris, et la somme des carrés des niveaux de gris. De même la matrice de covariance d'un ensemble de points 2-D ou 3-D peut être calculée aisément à partir des sommes des produits de coordonnées. Les mêmes remarques concernant la condition (A_2) peuvent être faites au sujet des attributs associés aux couples de régions tels que : nombre de points et gradient moyen le long de la frontière, coefficient de recouvrement.

Implantation On représente un triplet $(S, V(S), W(S))$ par un graphe valué. On associe à chaque nœud du graphe un élément de S et le vecteur attribut qui le caractérise. Nous associons à chaque arc du graphe un couple d'éléments de S et le vecteur attribut qui caractérise les relations existant entre les deux nœuds. Chaque nœud du graphe correspond à la réunion d'un ensemble de nœuds issus du graphe initial - le graphe initial étant obtenu à partir d'une segmentation initiale (dans le pire des cas l'image) - . Pratiquement des arcs sont créés seulement entre des régions vérifiant des propriétés intéressantes par exemple de connexité ou de recouvrement [20, 24].

Le processus de croissance de régions consiste donc à calculer le graphe initial et à itérer les deux actions suivantes :

1. Détermination du couple de nœuds à regrouper.
 2. Mise à jour du graphe après le regroupement des deux nœuds correspondant.
1. L'étape nécessite un accès rapide au couple de nœuds vérifiant le prédicat courant et optimisant la fonction de coût. Il est donc nécessaire de définir une structure de données permettant de représenter efficacement un ensemble d'objets muni d'une relation d'ordre et sur lequel doivent être appliqués les opérations suivantes :

- Insérer
- Oter le minimum

La structure de tas (**heap**) permet d'effectuer ces opérations avec un coût de l'ordre du logarithme du nombre d'éléments pris en compte. On trouvera une description du tas dans et de son implantation dans [38].

2. L'étape nécessite un **accès rapide** aux arcs incluant un nœud donné ainsi qu'**aux attributs** correspondant. On peut donc choisir une représentation du graphe permettant un accès direct à l'ensemble des arcs comprenant un nœud donné ainsi qu'**aux attributs** correspondant.

Le graphe est représenté par les **listes d'arcs associées à chaque nœud et des listes d'attributs associées aux nœuds et aux arcs.**

On spécifie ainsi un **problème de segmentation** par :

- Un **graphe initial** issu d'une partition initiale et les attributs associés à ses nœuds et à ses arcs.
- **Deux fonctions de mise à jour** attachées respectivement aux attributs-nœuds et aux attributs-arcs (T et U).
- **Une suite de couples (Predicat de fusion, Coût de fusion)** qui définit la stratégie de regroupement $((P_1, Q_1), \dots, (P_n, Q_n))$.

Complexité Discutons maintenant de la complexité algorithmique de cet algorithme. Les calculs provoqués par la fusion de deux nœuds sont donc les suivants:

- Calcul des attributs du nouveau nœud en utilisant la fonction de mise à jour T .
- Calcul des attributs des arcs incluant le nouveau nœud en utilisant la fonction de mise à jour U .

Nous devons ajouter aux calculs précédents ceux dus à la détermination du couple à fusionner et à la mise à jour du tas c'est-à-dire :

- Prendre la tête du tas jusqu'à ce qu'un arc valide soit extrait.
- Calcul du coût attaché aux nouveaux arcs et rangement des pointeurs correspondants dans le tas.

Plus formellement la complexité algorithmique peut être calculé comme suit :

Soient N et B respectivement le nombre de nœuds et le nombre d'arcs du graphe initial.

Soit F le nombre de fusions effectuées durant la segmentation.

Chaque nœud est connecté en moyenne avec $V = \frac{2 \cdot B}{N}$ nœuds ; on suppose que V reste à peu près constant durant le processus.

Soit T_1 le nombre d'opérations requises pour le calcul de la fonction T .

Soit T_2 le nombre d'opérations requises pour le calcul de la fonction U .

Soit T_3 le nombre d'opérations requises pour le calcul du prédicat et de la fonction de coût.

Soit M le nombre moyen de nœuds adjacents vérifiant le prédicat de fusion pour un nœud donné du graphe initial ; on suppose que M reste constant durant le processus de regroupement.

Le calcul nécessité par la mise à jour de graphe lors des fusions est : $F.T_1 + F.T_2.V$.

Pour mettre à jour tas lors d'une fusion nous insérons dans le tas en moyenne V nouveaux arcs et ainsi nous calculons V nouvelles valeurs de prédicats et de coût. De plus nous insérons dans le tas les nouveaux arcs qui vérifient le prédicat. Nous pouvons raisonnablement supposer que le nombre maximum d'éléments dans le tas est à peu près $V.M$.

Le nombre d'opérations requises pour la mise à jour du tas lors des fusions est donc approximativement :

$$F.T_3.V + F.M.V.O(\text{LOG}(V.M))$$

Le second terme de la somme ci dessus correspond au coût de l'insertion des valeurs des nouveaux arcs créés par le regroupement.

La complexité totale de notre algorithme est donc :

$$O(F.T_1 + F.T_2.V + F.T_3.V + F.M.V.\text{LOG}(V.M))$$

Ce résultat montre l'importance de V et M pour la complexité de notre algorithme. M caractérise l'état du graphe initial par rapport au prédicat de fusion. On a donc intérêt à utiliser une bonne méthode (!) pour déterminer une segmentation initiale. Nous remarquons que la complexité est une fonction de la "densité d'arc" du graphe. Nous aurons donc intérêt à limiter le nombre de relations entre les nœuds du graphe.

Cet algorithme de segmentation par croissance de régions s'appliquent pour des images de dimensions quelconques, monochromatiques et multichromatiques, ainsi que pour la segmentation de surfaces 3d [20, 24]. Il faut noter qu'il peut prendre en compte le résultat d'une détection de contours afin de contrôler la croissance des régions aussi avec des propriétés de discontinuité [25]. Nous donnons un exemple d'application de cette méthode pour des images 2D monochromatiques [20].

Application aux images monochromatiques De manière à appliquer cette méthode de segmentation aux images monochromatiques il faut établir une hiérarchie des propriétés désirées pour les régions. Ensuite, on attache à chaque propriété d'homogénéité un prédicat, une fonction globale de qualité, et une fonction locale de qualité. On détermine alors les attributs associés aux régions et aux couples de régions permettant de calculer le prédicat

d'homogénéité et la fonction locale de qualité. On montre enfin que ces prédicats et ces fonctions vérifient les propriétés définies dans le paragraphe précédent.

La détermination de critères de segmentation s'appliquant à une classe importante d'images naturelles monochromatiques n'est pas aisée. En effet la multiplicité des facteurs physiques intervenant dans la formation d'une image rend difficile le calcul de fonctions d'homogénéité permettant de différencier les projections des divers objets de la scène. L'expérimentation montre que des critères simples donnent le plus souvent des résultats comparables à ceux obtenus à l'aide de critères compliqués que de plus on maîtrise mal. Il est donc intéressant d'utiliser une suite de critères simples permettant de caractériser une région homogène au sens des niveaux de gris. Le premier critère basé sur l'amplitude des variations des niveaux de gris à l'intérieur d'une région permet de déterminer les régions de luminance quasiment uniforme (cette amplitude de variation est obtenue en effectuant la différence entre le niveau de gris le plus haut et le niveau de gris le plus bas). Le deuxième critère, basé sur la variance des niveaux de gris, engendre la formation de régions homogènes mais bruitées. Les régions de luminance à peu près constante sont ainsi regroupées afin de former des régions de variance faible. Le troisième critère permet de fusionner les couples de régions voisines de luminances moyennes sensiblement différentes, mais présentant une frontière avec un faible gradient. On remarquera que ces trois critères ont une complexité calculatoire croissante. De manière générale, plus on avance dans le processus de regroupement, plus les informations nécessaires à la croissance des régions deviennent complexes. Mais en contrepartie, le nombre de régions diminuant, le coût du processus de fusion n'est pas affecté. L'ordre d'application des critères est important car il correspond à une description hiérarchique des régions recherchées. Il serait par exemple illogique d'utiliser d'abord la variance des niveaux de gris et ensuite leur amplitude de variation. En effet, lorsque le processus de regroupement débute avec tous les pixels de l'image la variance des niveaux de gris d'une région est moins significative et plus coûteuse à calculer que l'amplitude de leurs variations. Lorsque les régions de luminance uniforme se sont formées, alors seulement la variance devient un critère intéressant. De même, lorsque les régions homogènes bruitées sont déterminées alors le gradient sur la frontière devient significatif. Les informations utilisées par le processus de croissance de régions évoluent donc au cours de celui-ci, correspondant à des niveaux de description différents de l'image.

Plus concrètement on suppose qu'une région peut être hiérarchiquement décrite comme un ensemble connexe de points tel que :

1. la somme des gradients calculés sur ses points intérieurs est faible ie des contours ne la traversent pas,
2. la région peut être partitionnée en sous-régions dont la variance des niveaux de gris

est faible,

3. chaque sous-région de faible variance de niveau de gris peut être elle-même décomposée en sous-régions dont les pixels ont à peu près la même luminance.

La troisième propriété peut être caractérisée par le prédicat P_1 , la fonction de qualité globale C_1 et la fonction de qualité locale Q_1 :

Soit $I(i, j)$ l'image initiale Soit $S = \{R_1, R_2, \dots, R_n\}$ une segmentation de I

$P_1(R_i) = \{(MAX_i - MIN_i) < s_1\}$; s_1 est un seuil ; MIN_i et MAX_i sont respectivement la plus petite et la plus grande valeur prises par I dans la région R_i .

$$C_1(S) = \sum_{i=1, n} \sum_{(k, l) \in R_i} ((I(k, l) - MAX_i)^2 + (I(k, l) - MIN_i)^2)$$

$$Q_1(R_i) = MAX_i - MIN_i$$

La seconde propriété peut être caractérisée par le prédicat P_2 , la fonction de qualité globale C_2 et la fonction de qualité locale Q_2 :

$P_2(R_i) = [V(R_i) < s_2]$; s_2 est un seuil ; $V(R_i)$ est la variance des niveaux de gris des pixels de R_i .

s_2 est pris égal à s_1^2 de manière à ce que les régions issues de la première étape vérifient P_2 :

Notons $E(R_i)$ et $C(R_i)$ les moyennes respectives des niveaux de gris et du carré des niveaux de gris de la région R_i . Nous avons :

$MAX_i - MIN_i < s_1 \Rightarrow |I(k, l) - I(r, s)| < s_1 ; r(k, l)$ et $r(r, s)$ points de R_i (I est la fonction des niveaux de gris) $\Rightarrow (I(k, l) - I(r, s))^2 < s_1^2 \Rightarrow C(R_i) - E(R_i)^2 < s_1^2 \Rightarrow V(R_i) < s_1^2$

$C_2(S) = \sum_{(i, j) \in S} (I(i, j) - M_{i, j})^2$; $M_{i, j}$ est la moyenne des niveaux de gris de la région où (i, j) est inclu.

$$Q_2(R_i) = V(R_i)$$

La première propriété peut être exprimée par le prédicat P_3 , la fonction de qualité globale C_3 et la fonction de qualité locale Q_3 :

Soient R_1 and R_2 deux ensembles de points connexes et S une segmentation.

Soit $F(R_1, R_2)$ l'ensemble des couples de points définissant la frontière entre R_1 et R_2 (voir figure 29).

Soit $N_F(R_1, R_2)$ le nombre de couples de points de $F(R_1, R_2)$

Soit $D(R_1, R_2)$ le gradient moyen à la frontière entre R_1 et R_2 :

1													
1	1												
1	1	1	1	1	1	2	2	2	2	2			
1	1	1	1	1	2	2	2	2	2	2	2	2	
1	1	1	1	1	2	2	2	2	2	2	2	2	
1	1	1	1	1	2	2	2	2	2	2			
1	1												
1													

Figure 29 : Les couples de points déterminés par les extrémités des segments définissent la frontière entre la région "1" et la région "2" : $F("1", "2")$

$$D(R_1, R_2) = \left(\sum_{((i,j), (k,l)) \in F(R_1, R_2)} \frac{|I(i,j) - I(k,l)|}{(N_F(R_1, R_2))} \right)$$

$C_3(S) = \sum_{((i,j), (k,l)) \in X_s(I)} |I(i,j) - I(k,l)|$; $X_s(I)$ est l'ensemble des couples de points connexes de l'image I inclus dans une même région S .

Nous supposons que pendant le processus de fusion :

$$P_3(R_1 \cup R_2) = [D(R_1, R_2) < s_3] ; s_3 \text{ est un seuil.}$$

s_3 est pris supérieur à la valeur maximum de $D(R_1, R_2)$ sur les couples de régions adjacentes issues de l'étape précédente.

$$Q_3(R_1 \cup R_2) = D(R_1, R_2)$$

Le choix des différents seuils s_1, s_2, s_3 implique $P_1 < P_2 < P_3$.

$<$ est le symbole de la relation d'ordre partiel sur les prédicats :

Soient P et Q deux prédicats définis sur un ensemble E $P < Q$ signifie que si P est vrai alors Q est vrai.

On peut montrer que les conditions (A_1) et (A_2) définies précédemment sont vérifiées [20].

On spécifie ainsi ce problème de segmentation par les couples $(P_1, Q_1), (P_2, Q_2)$, et (P_3, Q_3) . Les attributs attachés à chaque région sont donc : l'aire, la somme des niveaux de gris, la somme des carrés des niveaux de gris, le maximum et le minimum des niveaux de gris ; et les attributs associés à chaque couple de régions : le nombre de couples de points définissant la frontière et la somme des différences de niveaux de gris entre ces points.

Lors d'une première étape on détermine une segmentation initiale afin de diminuer le coût du processus de regroupement. Nous utilisons la procédure MERGE de Pavlidis

avec le prédicat d'homogénéité P_1 . Ensuite on construit le graphe d'adjacence correspondant aux régions obtenues de manière à appliquer la stratégie de regroupement définie par (P_1, Q_1) , (P_2, Q_2) , et (P_3, Q_3) . La première étape de fusion permet d'obtenir des régions de luminance à peu près uniformes ; Lors de la seconde des régions de niveaux de gris homogènes mais bruités se forment, grâce à la troisième les régions de luminance sensiblement différentes mais dont le gradient à la frontière est faible sont fusionnées. On ajoute une dernière étape qui élimine les petites régions.

3.5 Détection de régions par fermeture de contours

3.5.1 Introduction

L'idée de R. Deriche et J.-P. Cocquerez [10] est de détecter les régions en identifiant les contours fermés constituant leurs frontières. On détermine alors les régions non pas en recherchant des zones homogènes mais en détectant les points frontières entre deux zones homogènes de caractéristiques différentes.

Ces points frontières correspondent aux points de contour et peuvent être calculés avec un algorithme de détection de contours.

Il faut noter que cette approche prend le contrepied de la méthode précédente fondée sur la détermination d'ensembles connexes de points possédant des propriétés d'homogénéité. Les contours issus d'un algorithme de détection de contours sont rarement fermés, aussi les difficultés des approches "détection des régions par fermeture des contours" résident justement dans la fermeture de ces contours. Le succès de ce genre de méthodes dépend donc généralement de la qualité du détecteur de contours utilisé.

R. Deriche et J.P. Cocquerez utilisent les résultats obtenus par le détecteur de contours de R. Deriche décrit précédemment [7]. On remarquera que ce choix conduit à l'obtention de régions dont la distribution des niveaux gris correspond à une fonction de Heavyside additionnée d'un bruit blanc.

L'originalité de cet algorithme est de proposer une méthode de fermeture basée sur un opérateur simple qui permet de fermer les contours en suivant les crêtes de gradient. Les régions sont ensuite obtenues par détection des composantes connexes maximales n'incluant pas de points de contours. Cette méthode a un coût algorithmique très faible (linéaire selon le nombre de points de l'image).

3.5.2 Description d'un l'algorithme de fermeture de contours

Le principe de cet algorithme [10] est de détecter, par balayage de l'image ligne par ligne avec un opérateur 3×3 , les extrémités des contours, puis de fermer ces contours en les

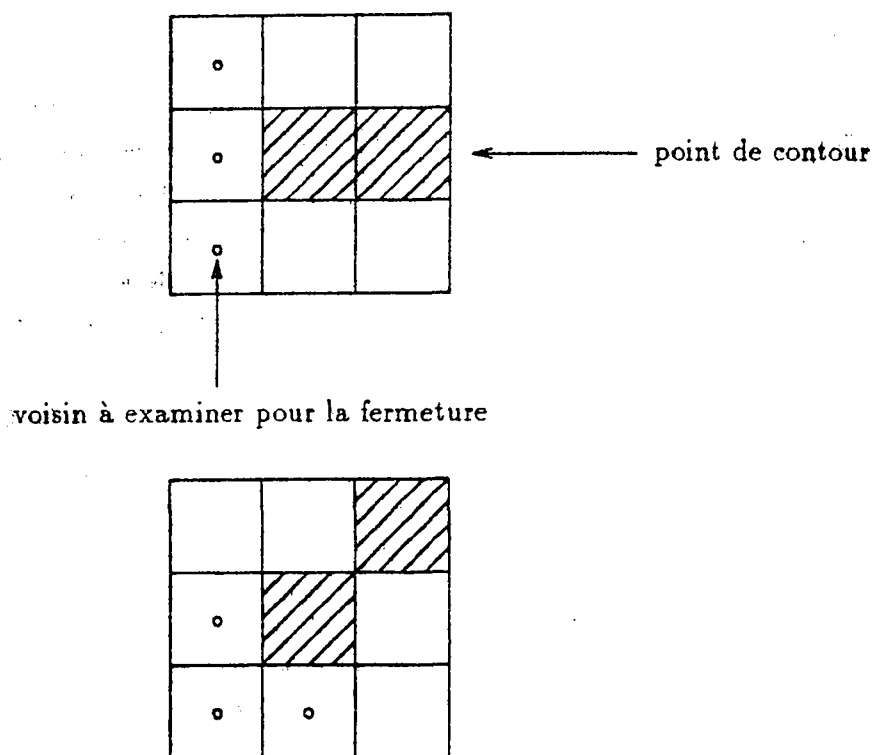


Figure 30 :

prolongeant par suivi des crêtes du gradient.

L'extraction des maxima locaux de la norme du gradient dans la direction du gradient permet l'obtention de contours d'épaisseur 1, le plus souvent.

Ainsi, une extrémité de contour peut être identifiée par un simple examen de son voisinage 3×3 .

La topologie de chaque extrémité permet de définir une direction d'exploration pour la fermeture et à chaque configuration d'extrémités est associée la liste des voisins à examiner. Pour élaborer le chemin de crête issu d'un point extrémité il suffit d'examiner 3 voisins en fonction de la topologie de l'extrémité.

L'image après détection de contours et fermeture comprend les pixels marqués à 1 (les contours) et d'autres pixels marqués à 0 qui représentent des régions.

Le problème est donc d'affecter une même étiquette aux points marqués 0 appartenant à la même composante connexe de l'image. Cette opération est effectuée par simple balayage de l'image par un opérateur en L. Les classes d'équivalence sont mises à jour à chaque nouvelle application de l'opérateur.

Exemple de voisins à examiner pour la fermeture des contours, dans deux cas de configurations différentes:

3.6 Résultats expérimentaux

Nous présentons à titre indicatif des résultats obtenus avec les algorithmes (figures 31 à 37). On remarque que les résultats obtenus avec ces deux types d'algorithmes sont comparables pour des images de scène d'intérieur. On notera aussi que les contours des régions correspondent le plus souvent aux contours extraits par détection de contours. Les temps de calcul sur une station de travail SUN-3 sont pour une image 256×256 de l'ordre de trois minutes CPU pour la croissance de régions et de l'ordre d'une minute CPU pour la fermeture de contours (détection de contours comprise).

Les figures 31 à 35 présentent des résultats obtenus par croissance optimale de régions et les figures 36 et 37 des résultats obtenus par fermeture de contours.

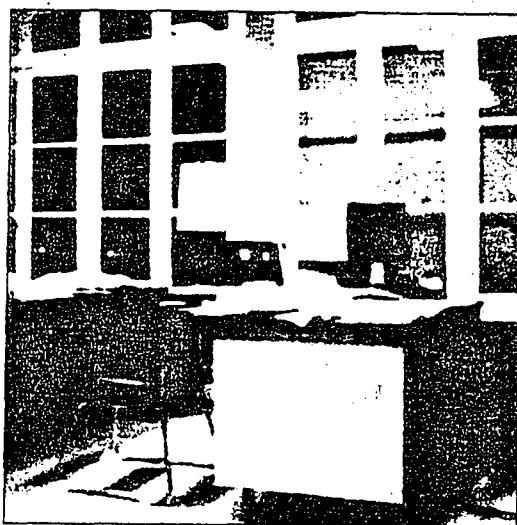


Figure 31 : Image originale

4 Conclusion : pourquoi segmenter des images ?

La segmentation est la première étape des systèmes de vision et est donc confrontée avec la réalité de l'image : une matrice de valeurs issue d'un mélange de phénomènes physiques et géométriques divers. De plus la représentation de l'information image qu'elle génère conditionne tous les traitements ultérieurs. Elle mérite donc que l'on s'y intéresse.

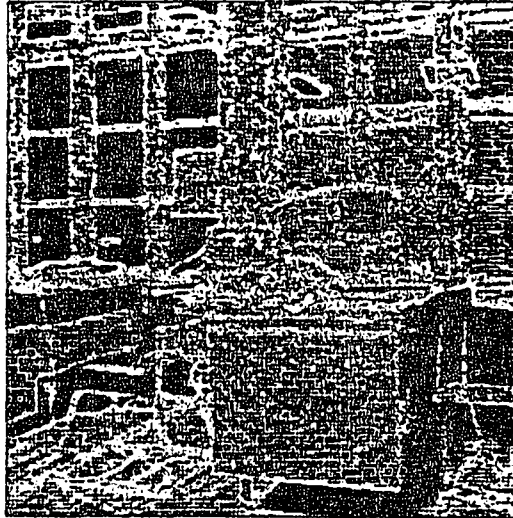


Figure 32 : Utilisation du critère max-min (croissance optimale de régions)

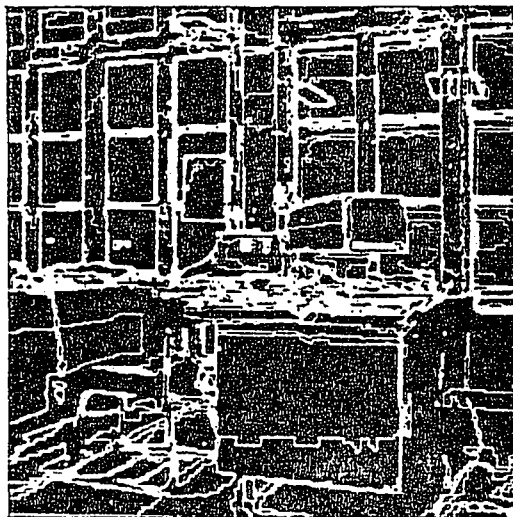


Figure 33 : Utilisation du critère moyenne (croissance optimale de régions)

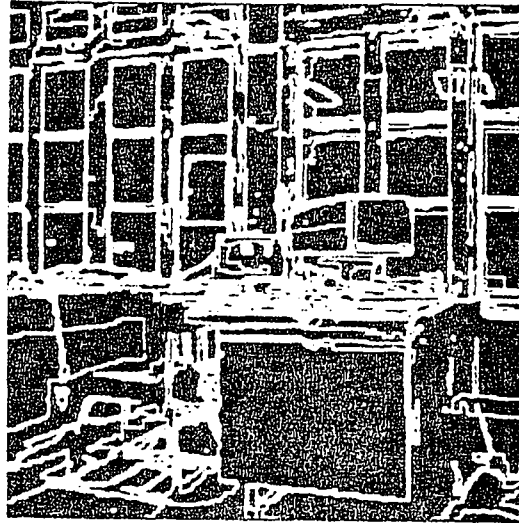


Figure 34 : Utilisation du critère gradient (croissance optimale de régions) :

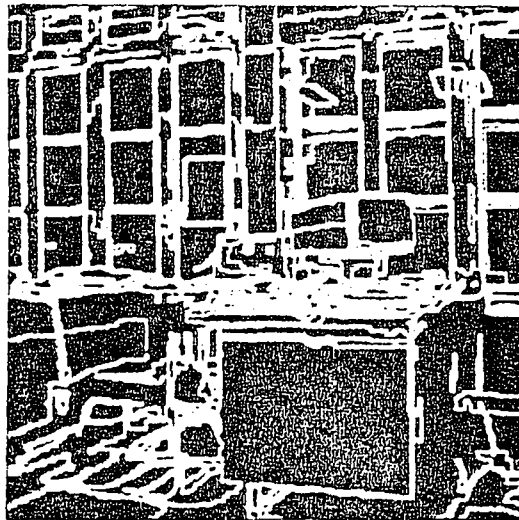


Figure 35 : Segmentation finale après élimination des petites régions (croissance optimale de régions)

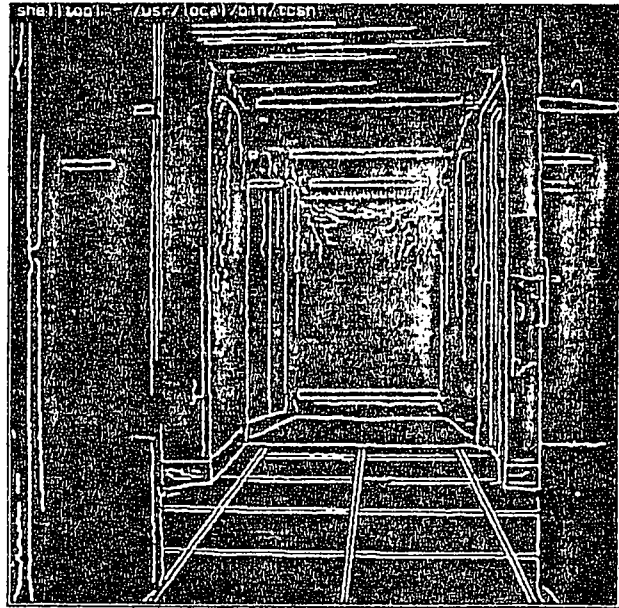


Figure 36 : Image originale des contours

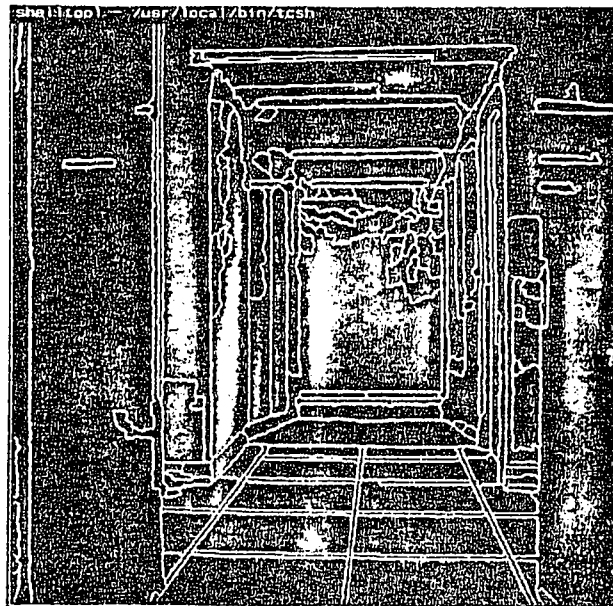


Figure 37 : Contours fermés détectés

L'objectif de la segmentation d'images est d'extraire des informations sur l'univers que représente l'image. Il est donc nécessaire d'établir des liens entre les propriétés des éléments de cet univers et l'image les représentant. On se trouve confronté au problème de mise en correspondance entre les indices images et les indices objets. Ceci implique de ne pas considérer l'image comme un signal mais comme la représentation d'une réalité physique. La complexité du processus de formation d'une image explique que peu d'algorithmes de segmentation la prennent en compte. Néanmoins les modélisations les plus répandues sont sous-tendues par une réalité physique. Par exemple la modélisation d'un contour par une marche d'escalier additionnée d'un bruit blanc ou d'une région par une zone de variance faible correspondent à une modélisation simplifiée.

Ainsi l'évaluation des résultats d'un algorithme de segmentation n'a de sens que par rapport à l'univers représenté par l'image. Aussi une manière objective d'estimer une méthode est son inclusion dans un système de vision complet : c'est-à-dire visant à accomplir une tâche spécifique : perception tridimensionnelle d'une scène d'intérieur pour la navigation d'un robot ou d'un organe humain pour guider un acte chirurgical.

Il est préférable d'éviter de juger à l'œil des résultats de segmentation. En effet la segmentation d'images n'est pas une fin en soi et ne doit pas être considérée comme un problème général de la vision artificielle pouvant être étudié indépendamment et engendrant lui-même ses propres questions. On ne segmente pas pour le plaisir !

Remerciements

Je tiens à remercier Rachid Deriche et Brigitte Wrobel pour l'aide indirecte qu'ils ont apportés à la rédaction de ce cours au travers d'articles écrits en commun.

Bibliographie

Références

- [1] T. Asano and N. Yokoya. Image segmentation schema for low level computer vision. In *International Conference on Pattern Recognition*, number 14, pages 267-273, 1981.
- [2] M. Berthod. Un nouvel algorithme d'approximation polygonale. Technical report, I.N.R.I.A., 1987.
- [3] C. Brice and C. Fenema. Scene analysis using regions. In *Artificial Intelligence*, number 1, pages 205-226, 1970.

- [4] J. Bryant. On the clustering of multidimensional pictorial data. In *Pattern Recognition*, number 11, pages 115-125, 1979.
- [5] J.F. Canny. Finding edges and lines in images. In *MIT*, number TR. 720, June 1983.
- [6] C.K. Chow and T. Kaneko. Boundary detection of radiographic images by a thresholding method. In S. Watanebe, editor. *Frontiers of pattern recognition*, pages 61-82. Academic Press, 1972. New York.
- [7] R. Deriche. Optimal edge detection using recursive filtering. *International Journal of Computer Vision*, 1987.
- [8] R. Deriche. Separable recursive filtering for efficient multi-scale edge detection. In *Proc. International Workshop on Machine Vision and Machine Intelligence*, Tokyo, December 1987. IEEE.
- [9] R. Deriche. Fast algorithms for low level vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1989.
- [10] R. Deriche and J.-P. Cocquerez. Extraction de composants connexes basée sur une détection optimale des contours. In *CESTA*, Paris, 1987.
- [11] G. Giraudon. An efficient edge chaining algorithm. In *5th Scandinavian Conference on Image Analysis*, June 1987. Stockholm.
- [12] R.M. Haralick and I. Dinstein. A spatial clustering procedure for multi-image data. *IEEE Trans. on Circuits and System*.
- [13] R.M. Haralick and S.G. Shapiro. Survey: image segmentation techniques. In *Computer Vision Graphics and Image Processing*, number 29, pages 100-132, 1985.
- [14] R.M. Haralick and L.T. Watson. A facet model for image data. In *Computer Graphics and Image Processing*, number 15, pages 113-129, 1981.
- [15] S.L. Horowitz and T. Pavlidis. Picture segmentation by a tree transversal algorithm. *Assoc. Comput. Machine*, 23:368-388, 1976.
- [16] N. Keskes. *Application des techniques d'analyse d'images aux signaux sismiques*. PhD thesis, Université Paris VI, 1984.
- [17] R. Kohler. A segmentation system based on thresholding. *Computer Graphics and Image Processing*, 15:319-338, 1981.

- [18] D. Marr. *Vision*. MIT Press, W.H. Freeman and Co., 1982.
- [19] D. Marr and E. Hildreth. Theory of edge detection. *Proc. Royal Soc. Lond.*, B 207:187-217, 1980.
- [20] O. Monga. An optimal region growing algorithm for image segmentation. *International Journal on Pattern Recognition and Artificial Intelligence*, 1(3):351-376, December 1987.
- [21] O. Monga, M. Berthod, S. Castan, G. Giraudon, J. Shen, and B. Wrobel. Segmentation d'images et robotique mobile. In *Journées Nationales du PRC-GRECO Communication Homme-Machine*. Actes des journées PRC e'dite's par EC2, 1988. Paris.
- [22] O. Monga and R. Deriche. 3d edge detection using recursive filtering. In *Computer Vision and Pattern Recognition*, San Diego, June 1989. IEEE.
- [23] O. Monga, R. Deriche, and G. Malandain. Recursive filtering: a primary tool for 3d edge detection. In *First European Conference on Computer Vision. ECCV 90*, 1990.
- [24] O. Monga and N. Keskes. A hierarchical algorithm for the segmentation of 3-d images. In *Proc. of eighth International Conference on Pattern Recognition*, Paris, October 1986.
- [25] O. Monga and B. Wrobel. Une méthode simple de contrôle d'une croissance de régions avec les contours. In *AFCET*, Antibes, 1987.
- [26] J. Muerle and D. Allen. Experimental evaluation of techniques for automatic segmentation of objects in a complex scenes. In G. Cheng et al., editor, *Pictorial Pattern Recognition*, pages 3-13. Thompson, 1968. Washington, D.C.
- [27] M. Nagao and M. Matsuyama. *Structural image analysis*, 1983.
- [28] Y. Ohta, T. Kanade, and T. Sakai. Color information for region segmentation. In *Computer Graphics and Image Processing*, pages 222-241, 1980. No. 13.
- [29] R. Olhander, K. Price, and D.R. Reddy. Picture segmentation using a recursive region splitting method. In *Computer Graphics and Image Processing*, number 8, pages 313-333, 1978.
- [30] T.C. Pong, L.G. Shapiro, L.T. Watson, and R.M. Haralick. Experiments in segmentation using a facet model region grower. In *Computer Vision Graphics and Image Processing*, number 25, 1980.

- [31] J.M. Rocchisani, O. Monga, J. Bittoun, and R. Deriche. Automatic spatio-temporal edge detection of the beating heart in mri. In *17ème Congrès International de Radiologie*, Paris, July 1989.
- [32] A. Rosenfeld and A. Kak. Digital image processing. *Academic Press: New York*, 2, 1982. Second Edition.
- [33] J. Shen and S. Castan. An optimal linear operator for edge detection. In *Conference on Vision and Pattern Recognition*. USA, June 1986. IEEE.
- [34] J. Shen and S. Castan. An optimal linear operator for edge detection. In *Conference on Vision and Pattern Recognition*. IEEE, 1986.
- [35] S. Watanabe and CYBEST Group. An automated apparatus for cancer prescreening: Cybest. In *Computer Graphics and Image Processing*, number 3, pages 350-358, 1974.
- [36] J.S. Weszka. A survey of threshold selection techniques. In *Computer Graphics and Image Processing*, number 7, pages 259-265, 1978.
- [37] J.S. Weszka, R.N. Nagel, and A. Rosenfeld. A threshold selection technique. In *IEEE Trans. Comput.*, number C-23, pages 1322-1326, 1974.
- [38] M. Williams. Algorithm 232 heapsort. In *Graphic and Image Processing Communication of the ACM*, volume 7, pages 347-348, June 1984.
- [39] S. Zucker. Region growing: childhood and adolescence. *Computer Graphics and Image Processing*, 5:1322-1326, 1976.

SUPPORT DE COURS POUR LE CONGRES PIXIM 89

Version révisée

SEGMENTATION D'IMAGES : où en sommes nous ?

Olivier MONGA

INRIA, Domaine de Voluceau, B.P. 105

78153 Le Chesnay Cédex

Résumé

L'objectif de cet article est d'effectuer un état de l'art des différentes méthodes de segmentation d'images en se focalisant sur les derniers développements.

Une étape fondamentale dans la plupart des systèmes de vision par ordinateur est d'engendrer une description compacte d'une image, plus exploitable que l'ensemble des pixels. De nombreuses techniques dites de "segmentation d'images" permettent d'atteindre cet objectif. Elles sont généralement fondées sur la recherche des discontinuités locales (détection de contours) ou sur la détection de zones de l'image présentant des caractéristiques d'homogénéité (extraction de régions). Ces deux approches sont duales en ce sens qu'une région définit une ligne par son contour et qu'une ligne fermée définit une région. Elles conduisent cependant à des algorithmes différents et ne fournissent pas les mêmes résultats.

On caractérisera les différentes classes de méthodes de segmentation d'images en contours ou en régions. Pour chaque type de méthode un algorithme sera décrit en détail. On s'attachera plus particulièrement aux techniques les plus récentes.

=====

Tutorial for the Conference PIXIM'89

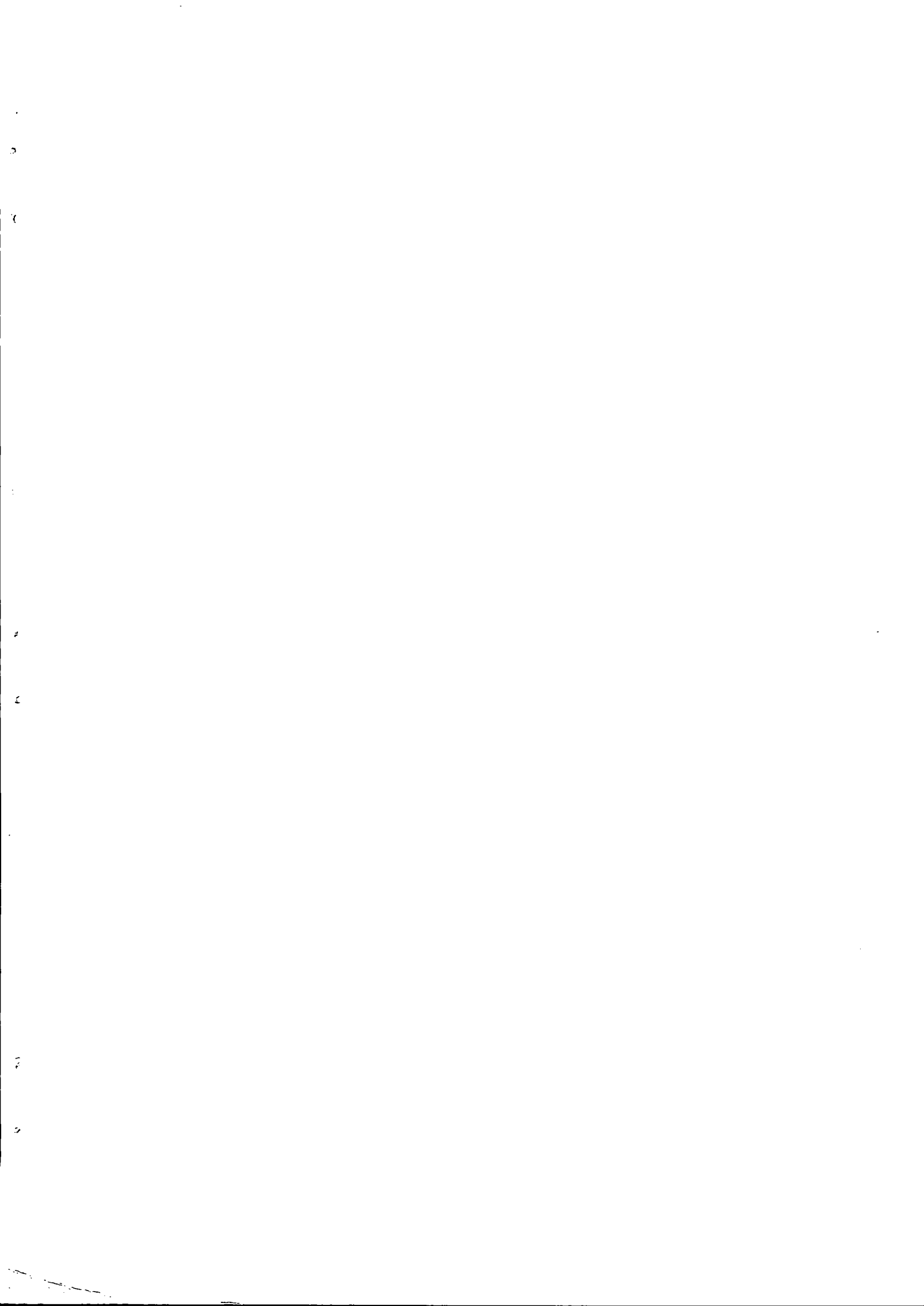
Image Segmentation: State of the Art

Abstract

This paper presents a state of the art of the various image segmentation methods by focusing on recent works.

A major problem in image analysis is to describe more compactly the visual information they contain. This problem can be solved either by finding local discontinuities of the grey level function: edge detection, either by searching areas of the image which present some homogeneous properties: region segmentation. These two approaches are dual because a region defines a line with its boundary and a closed line defines a region. Nevertheless they lead to completely different algorithms and do not provide the same results.

We characterize the different kinds of image segmentation algorithms. For each type of method an algorithm is precisely described. We focus on more recent techniques.



ISSN 0249 - 6399