

Segmentation

Segmentation

- Une image est souvent composée d'objets, cellules, surfaces. La segmentation consiste à extraire les attributs caractérisant ces entités.

- Deux approches :

Approche contour : les frontières entre les entités peuvent être matérialisées par un contour.

Approche région : les entités sont représentées par des zones « homogènes » de l'image (couleur, nvg, textures)

Détection de contours – principe (1)

- Soit une image I continue, un contour est une ligne qui représente les très fortes variations de niveaux de gris. Considérons le gradient de l'image :

$$\vec{G} = \vec{\nabla} I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right]^T$$

- On crée une image du gradient de I en calculant le module et l'argument :

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

- Un contour est le lieu des points où le gradient est maximum dans la direction du gradient :

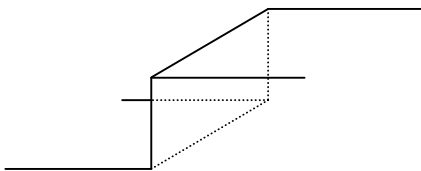
$$\frac{\partial G}{\partial g} = 0 \quad \text{et} \quad \frac{\partial^2 G}{\partial g^2} = \max \quad \text{avec} \quad \frac{\partial}{\partial g} = \vec{g} \vec{\nabla}$$

Détection de contours – principe (2)

- L'équation à résoudre est non linéaire et ne peut pas être résolue analytiquement :

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

- ➡ • On calcule l'image du gradient puis on recherche les extrémums dans la direction du gradient
- En présence d'un contour de type échelon, la double dérivation est impossible. On prétraite alors l'image en réalisant une convolution avec un filtre au moins deux fois dérivable (ex : la gaussienne)



Contour en échelon

Détection de contours – principe (3)

- On peut aussi se placer dans le repère formé par la tangente et la normale à l'image, ce repère fait un angle Φ par rapport au repère image :

$$\Phi = \text{Arctg} \left[\frac{\frac{\partial I}{\partial x}}{\frac{\partial I}{\partial y}} \right]$$

- On a alors : $\frac{\partial I}{\partial g} = \frac{\partial I}{\partial x} \cos(\Phi) + \frac{\partial I}{\partial y} \sin(\Phi)$

- Ce qui conduit en reprenant la définition d'un contour a :

$$\frac{\partial^2 I}{\partial g^2} = \frac{\partial^2 I}{\partial x^2} \cos^2(\Phi) + 2 \frac{\partial^2 I}{\partial x \partial y} \cos(\Phi) \sin(\Phi) + \frac{\partial^2 I}{\partial y^2} \sin^2(\Phi)$$

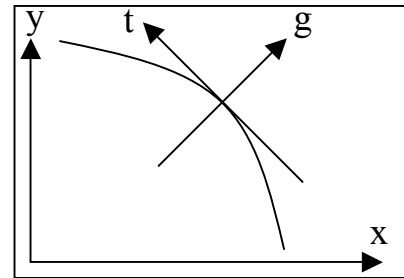
Détection de contours – principe (4)

- L'équation précédente n'est pas plus simple à résoudre, cependant elle fait apparaître les termes du Laplacien :

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

- Dans le repère local défini par la normale et la tangente, le laplacien vaut :

$$\Delta I = \frac{\partial^2 I}{\partial g^2} + \frac{\partial^2 I}{\partial t^2}$$



- si la courbure est faible alors : $\frac{\partial^2 I}{\partial t^2} \approx 0$ et $\Delta I = 0 \Rightarrow \frac{\partial^2 I}{\partial g^2} \approx 0$

Recherche des passage par zéro du Laplacien

Détection de contours (1)

Approches classiques

- Recherche des extrema de la dérivée première ou des zéros de la dérivée seconde.
- Le calcul est souvent réalisé par application d'un filtrage linéaire passe-haut précédé d'un filtrage passe-bas pour limiter l'effet du bruit.

Soit I l'image dont on veut déterminer les contours et f le filtre passe-bas d'élimination du bruit.

On calcule :

$(I*f)' = I*f'$: dérivée première $(I*f)'' = I*f''$: dérivée seconde

Le laplacien se calcule directement en convoluant l'image avec :

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

Détection de contours (2)

- On calcule donc :

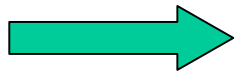
$$I_x(x,y)=\frac{\partial(I*f)}{\partial x}=I*\frac{\partial f}{\partial x}=I*f_x \quad I_y(x,y)=\frac{\partial(I*f)}{\partial y}=I*\frac{\partial f}{\partial y}=I*f_y$$

$$\Delta I(x,y)=I*\Delta(f)$$

- Exemple des masques de Roberts :

$$L=(1)$$

Le masque L choisit par Roberts s'obtient par différences finies



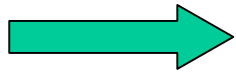
$$f_x = \begin{pmatrix} 1 & -1 \end{pmatrix} \quad f_y = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

Détection de contours (3)

- Exemple des masques de Prewitt

$$L = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Le masque L choisit par Prewitt étant non dérivable
 f_x et f_y sont obtenus par différences finies



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

- Masques d'approximation du Laplacien :

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

Détection de contours (4)


- Théoriquement le calcul du vecteur gradient dans deux directions non colinéaires suffit pour obtenir la norme et la direction du gradient. En pratique, en raison du bruit et de pixels se trouvant au croisement de plusieurs contours, il est parfois intéressant de calculer le gradient par une suite de dérivations directionnelles.

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

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

Masques de Kirsch : il y a 8 filtres issus de ces deux masques par rotation de 90°

Détection de contours (5)

- La convolution est coûteuse en temps de calcul, surtout si on répète plusieurs fois les calculs dans différentes directions.
- Pour limiter les coûts de calcul  masques 2D séparables.

$$I*f(x,y)=I*(f_x(x)f_y(y))=I*f_x(x)*f_y(y)$$

- Si on suppose le bruit homogène dans toutes les directions, on peut poser :

$$f_x=f_y=S \quad \text{ } \xrightarrow{\text{green arrow}} \quad I*f(x,y)=I*(s(x)s(y))=I*s(x)*s(y)$$

- Si de plus : $D = S'$ et $P = D''$ alors les dérivées premières I_x et I_y et secondes I_{xx} et I_{yy} s'écrivent :

$$I_x(x,y)=\frac{\partial(I*f)}{\partial x}=I*(D(x)s(y)) \quad I_y(x,y)=\frac{\partial(I*f)}{\partial y}=I*(s(x)D(y))$$

Détection de contours (6)

$$I_{xx}(x,y)=\frac{\partial^2(I*f)}{\partial x^2}=I*(P(x)s(y))$$

$$I_{yy}(x,y)=\frac{\partial^2(I*f)}{\partial y^2}=I*(s(x)P(y))$$

$$\Delta I=I*(s(x)P(y)+P(x)s(y))$$

- Exemple du masque de Sobel :

Soient $s = (1,2,1)$ et $D = (1,0,-1)$, on obtient les deux masques H1 et H2 :

$$H1=D(x)*s(y)=\begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \quad H2=s(x)*D(y)=\begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

H1 : Gradient en x, H2 : Gradient en y

Détection de contours (7)

- Retour sur le masque de Prewitt :

Si on pose $s = (1, \textcolor{red}{1}, 1)$ et $D = (1, 0, -1)$, on obtient :

$$H1 = D(x) * s(y) = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix} \quad H2 = s(x) * D(y) = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

- Les filtres de Prewitt et Sobel sont moins sensibles au bruit que ceux de Roberts car le fait d'introduire un moyennage local sur le domaine couvert par le masque diminue leur sensibilité. Le filtre de Sobel donne une meilleure estimation que celui de Prewitt car la série 1 2 1 est approximativement une gaussienne

Seuillage du gradient

- **Seuillage par hystérésis**

Le seuillage à hystérésis permet d'obtenir des contours fins et cohérents. Les pixels de l'image de gradient sont filtrés grâce à deux seuils distincts, $T1$ et $T2$ ($T1 < T2$). En dessous du seuil bas $T1$, on considère qu'il n'y a pas de contours. Au dessus du seuil haut, on décide qu'il y a contour : ces contours sont ensuite complétés par les pixels compris entre $T1$ et $T2$ si et seulement s'ils sont connexes entre eux.

Les étapes de l'algorithme sont les suivantes :

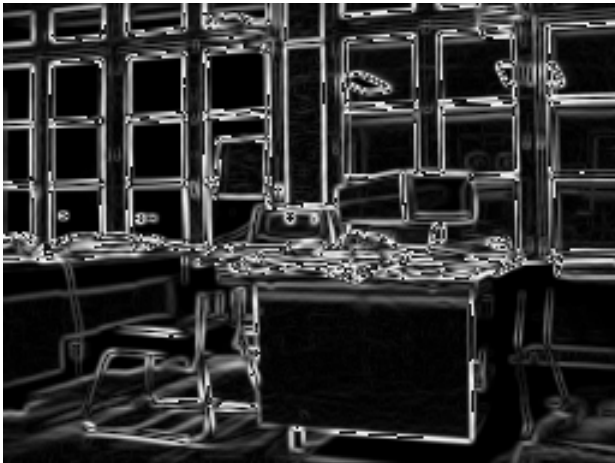
- * on effectue un calcul de gradient sur l'image à niveau de gris
- * tous les pixels en dessous de $T1$ sont mis à zéro
- * tous les pixels au dessus de $T2$ sont conservés
- * on ne garde les pixels dont le niveau est entre $T1$ et $T2$ que s'ils sont connexes à au moins un pixel de niveau supérieur à $T2$

Détection de contours

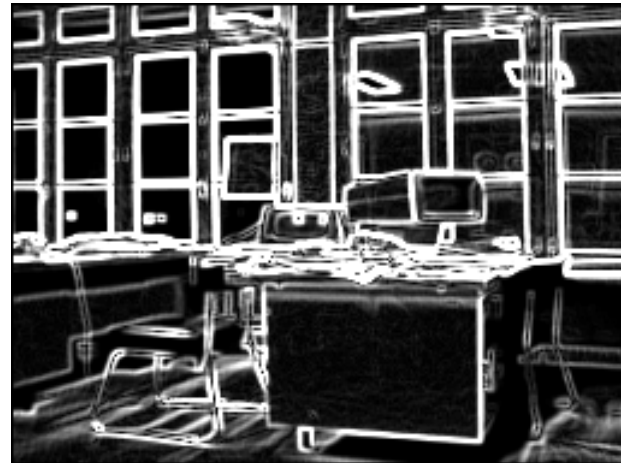
Exemples



Filtre de Roberts



Filtre de Prewitt



Filtre de Sobel

Détection de contours

Exemples



Filtre de Roberts



Filtre de Prewitt

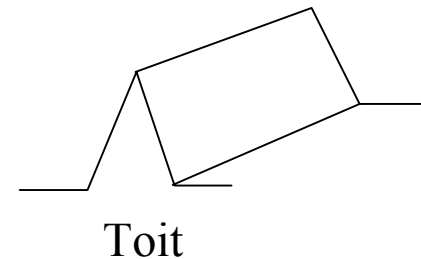
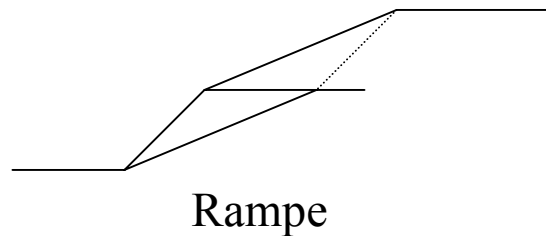
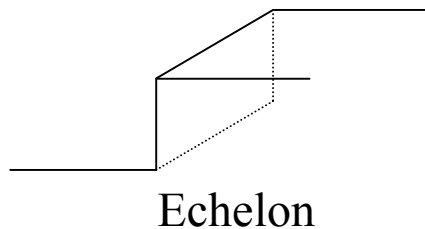


Filtre de Sobel

Filtrage 1D - mise en oeuvre

- Modèles de contours

1. Types de contours



2. Bruit

Des bruits divers sont présents dans les images : additifs, blanc, impulsionnels.

3. Modèle classiques de contours

Souvent on choisit un modèle type échelon perturbé par un bruit blanc à moyenne nulle.

Filtrage 1D - mise en oeuvre

- Soit $I(x) = A.u(x) + \eta(x)$

avec : $\sigma_0^2 = E(\eta^2(x))$ et $u(x)$: échelon unité, A : amplitude

4. Critères de performances

- *Détection* : l'opérateur doit donner une réponse au voisinage du contour,
- *Localisation* : le contour doit être localisé avec précision,
- *Réponse unique* : un contour ne doit provoquer qu'une seule réponse de l'opérateur d'extraction.

5. Détection

Elle est réalisée par convolution avec un filtre antisymétrique : $f(x)$

$$I_F(x_0) = \int_{-\infty}^{+\infty} I(x) f(x_0 - x) dx \quad \text{avec } I(x) \text{ le signal à filtrer}$$

Filtrage 1D – critères de Canny

6. Formalisation des critères de performances – Critères de Canny

- *Détection* : maximisation du rapport signal sur bruit. (sortie du filtre au point de discontinuité/écart type du bruit)

$$RSB = \frac{A}{\sigma_0} \frac{\int_{-\infty}^0 f(x) dx}{\sqrt{\int_{-\infty}^{+\infty} f^2(x) dx}} \quad \text{si} \quad \eta = \frac{\int_{-\infty}^0 f(x) dx}{\sqrt{\int_{-\infty}^{+\infty} f^2(x) dx}} \quad \text{alors} \quad RSB = \eta \frac{A}{\sigma_0}$$

- *Localisation* :

$$L = \frac{A}{\sigma_0} \frac{|f(0)|}{\sqrt{\int_{-\infty}^{+\infty} f^2(x) dx}} \quad \text{si} \quad \lambda = \frac{|f(0)|}{\sqrt{\int_{-\infty}^{+\infty} f^2(x) dx}} \quad \text{alors} \quad L = \lambda \frac{A}{\sigma_0}$$

Filtrage 1D - critères de Canny

- *Réponse unique* : un contour ne doit provoquer qu'une seule réponse de l'opérateur d'extraction; la distance moyenne entre les pics de la réponse au bruit, notée x_{\max} , est posée comme une fraction k de la taille de l'opérateur W (taille du masque de convolution nécessaire à l'implantation). x_{\max} se calcule facilement car on connaît la distance moyenne des passages par zéro de la dérivée seconde de la réponse d'un filtre à un bruit blanc gaussien.

$$x_{\max} = kW = 2\pi \sqrt{\frac{\int_{-\infty}^{+\infty} f'^2(x) dx}{\int_{-\infty}^{+\infty} f''^2(x) dx}}$$

Filtre de Canny

7. Résolution

On maximise le produit $\eta\lambda$ sous la contrainte du troisième critère, on obtient ainsi une équation différentielle dont la solution est :

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

Les coefficients a_i et ω sont déterminés à partir de la taille du filtre, α est le paramètre d'échelle qui indique en dessous de quelle distance 2 contours parallèles seront confondus.

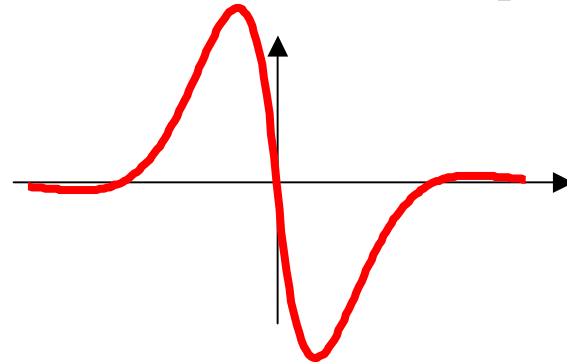
Si la taille du filtre tend vers l'infini alors $a_1 = a_2 = a_3 = a_4 = 0$ et on obtient:

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

Filtre de Canny

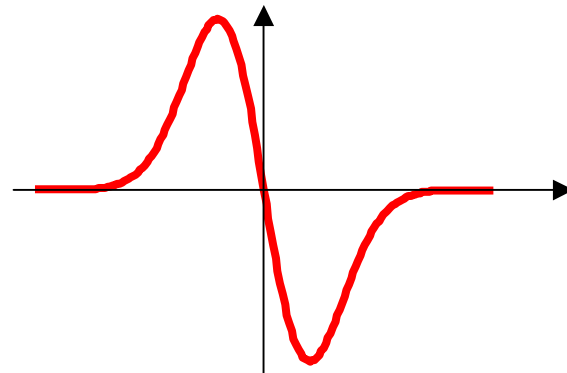
Le filtre étant impair on obtient pour le filtre de dérivation optimal :

$$f(x) = a_3 e^{-\alpha|x|} \sin(\omega x)$$



Canny montre que la dérivée d'une gaussienne est une bonne approximation de son filtre :

$$f(x) = -x \cdot \exp\left(-\frac{x^2}{2\sigma^2}\right)$$



Le critère $\eta\lambda$ vaut 1,12 pour le filtre optimal et 0,97 pour la dérivée de la gaussienne !

Filtre de Shen Castan

Le filtre de lissage de Shen s'écrit : $s_1(x) = c \cdot \exp(-\alpha|x|)$

c est choisi de manière à obtenir un filtre normalisé :

$$\sum_{-\infty}^{+\infty} s_1(k) = 1 \quad \longrightarrow \quad c = \frac{1 - e^{-\alpha}}{1 + e^{-\alpha}}$$

Le filtre de dérivation s'écrit : $f(x) = d \cdot \text{sgn}(x) \cdot \exp(-\alpha|x|)$

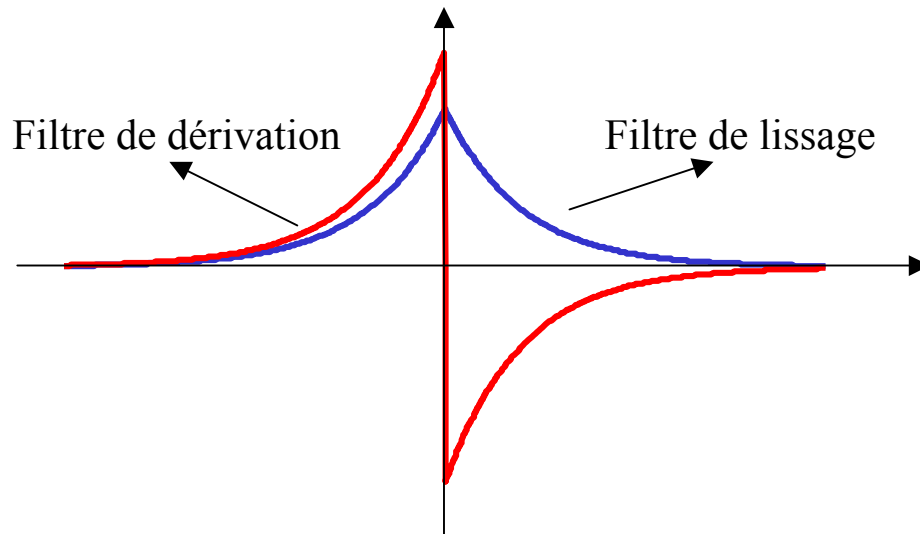
d est choisi de manière à obtenir un filtre normalisé :

$$\sum_0^{+\infty} f(k) = 1 \quad \longrightarrow \quad d = 1 - e^{-\alpha}$$

Le paramètre α définit la largeur du filtre, plus α est petit plus le lissage est important

Filtre de Shen Castan

Graphique du filtre de Shen-Castan



Le filtre obtenu par dérivation est une solution optimale pour les deux premiers critères de Canny. Il correspond à une valeur optimale pour le produit $\eta\lambda$ (meilleur compromis détection-localisation). La discontinuité en zéro évite une délocalisation importante même pour les faibles valeurs de a , mais cette discontinuité peut entraîner des contours multiples

Filtre de Deriche

Le filtre de Deriche est à réponse impulsionnelle infinie (RII). Il est synthétisé de façon récursive et répond aux critères de Canny avec un $\eta\lambda = 2$. Il est donc supérieur au filtre de Canny.

Le filtre de Deriche a une expression générale de la forme :

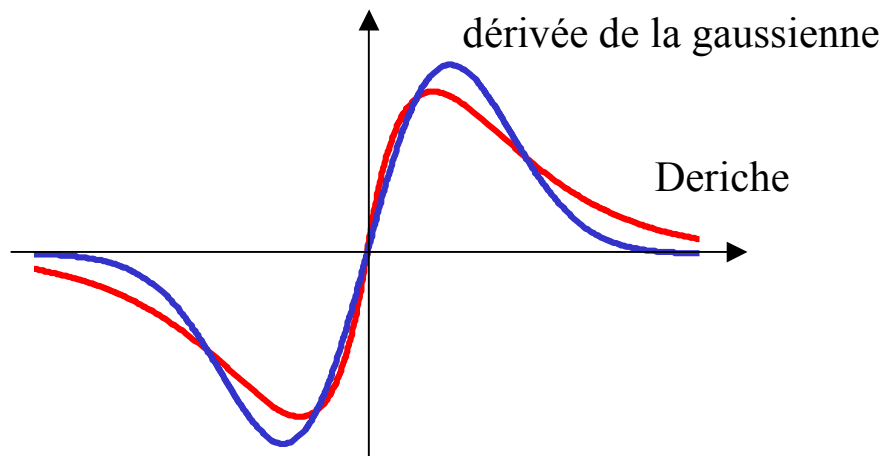
$$f(x) = -cx \cdot \exp(-\alpha|x|)$$

c est choisi de manière à obtenir un filtre normalisé :

$$\sum_{0}^{+\infty} f(k) = 1 \quad \longrightarrow \quad c = \frac{(1 - \exp(-\alpha))^2}{\exp(-\alpha)}$$

Filtre de Deriche

Comparaison du filtre de Deriche à la dérivée de la gaussienne



Le paramètre α de Deriche représente l'inverse de l'écart type σ de la gaussienne du filtrage de Canny ($\alpha = \sqrt{\pi}/\sigma$)

Filtre de Deriche

Implantation du filtre de Deriche

C'est un filtre récursif qui s'implémente en trois étapes :

1. Filtrage de gauche à droite (filtrage causal)

$$z^+(x) = c.a + 2a.z^+(x-1) - a^2.z^+(x-2)$$

avec : $a = \exp(-\alpha)$

2. Filtrage de droite à gauche (filtrage anticausal)

$$z^-(x) = -c.a + 2a.z^-(x+1) - a^2.z^-(x+2)$$

3. Sommation du filtrage causal et anticausal

$$z(x) = z^+(x) + z^-(x)$$

Filtre de Shen - exemples



Gradient en x



Gradient en y



Norme du gradient

$$\alpha = 0,5$$



Recherche des maxima locaux
et seuillage par hystérésis

Filtre de Deriche - exemples



Gradient en x



Gradient en y

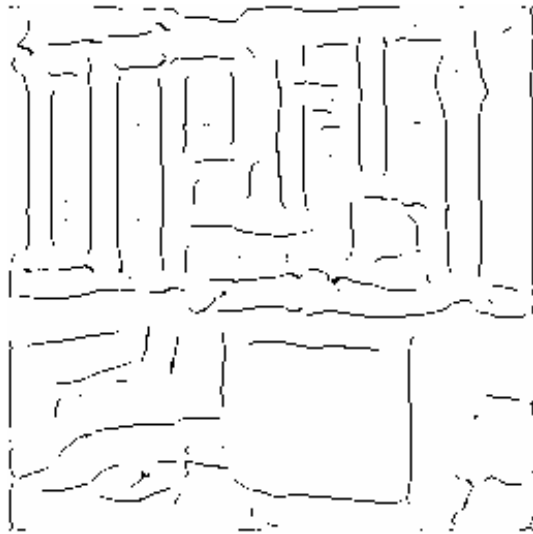


Norme du gradient

$$\alpha = 1$$

Il n'y a pas de différence visible avec le filtre de Shen

Comparaison des filtres



Deriche $\alpha = 0,2$



Deriche $\alpha = 1$



Shen $\alpha = 0,1$

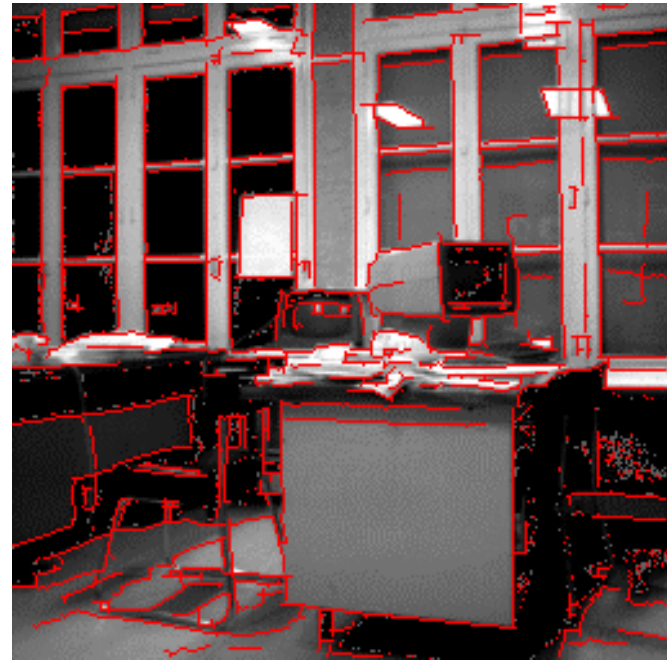


Shen $\alpha = 0,5$

Comparaison des filtres



Deriche $\alpha = 0,9$



Shen $\alpha = 0,51$

La localisation est excellente pour les deux filtres

Comparaison des filtres



Deriche $\alpha = 1$



Shen $\alpha = 0,5$

Détection de contours - conclusions

Extension en 2D

Un détecteur de contours se compose de deux estimateurs de dérivées, l'un selon les lignes l'autre selon les colonnes. Que ce soit pour les lignes ou les colonnes, cet estimateur est composé de du produit de deux fonctions :

- Une fonction de lissage (symétrique, passe-bas) dont l'étendue est fonction de l'importance du bruit et de la distance que l'on souhaite respecter entre deux contours (par exemple dans la direction des colonnes, si on souhaite calculer le gradient selon les lignes).
- Une fonction de dérivation (antisymétrique, passe-haut), très souvent obtenue à partir de la dérivation de la fonction de lissage.
- Les couples fonctions lissage/dérivation sont issus de :
 - La gaussienne : Canny
 - De l'exponentielle décroissante : Shen-Castan
 - Du produit de l'exponentielle par x : Deriche

Les contours actifs – « Snakes »

Hypothèse : on connaît approximativement le contour à détecter

→ On place au voisinage du contour à détecter une ligne initiale qui va se déformer progressivement jusqu'à atteindre un équilibre sur les contours réels.

Les déformations résultent de l'action de plusieurs forces :

- une énergie propre (tension et torsion mécanique du contour),
- une énergie potentielle imposée par l'image pour plaquer la courbe sur les contours,
- une énergie externe pour traduire les contraintes spécifiques du problème.

Le contour va se déformer progressivement pour atteindre l'énergie minimum, compromis entre les contraintes du problème

Les contours actifs – « Snakes »

La formalisation du problème nécessite de paramétrer le contour en fonction de l'abscisse curviligne.

$$v(t) = [x(t), y(t)]^T \quad t \in [0, 1]$$

$$E_{totale} = \int_0^1 [E_{interne}(v(t)) + E_{image}(v(t)) + E_{ext}(v(t))] dt$$

$$E_{interne} = \alpha(t) \left(\frac{dv}{dt} \right)^2 + \beta(t) \left(\frac{d^2v}{dt^2} \right)^2$$

- La première dérivée exprime les variations de longueur de la courbe (c'est un terme de tension - élasticité du contour),
- La seconde dérivée traduit les variations de courbure (c'est un terme de flexion – contrôlé par la raideur du contour),

Les contours actifs – « Snakes »

La combinaison de ces deux termes permettra l'obtention d'un contour régulier.

La seconde énergie : E_{image} caractérise les lignes de fort gradient. Elle s'écrit :

$$E_{\text{image}} = -|\nabla I|^2$$

La dernière énergie : E_{ext} est choisie par l'utilisateur. Elle peut prendre diverses formes (suivi d'un gabarit par exemple)

En l'absence de cette dernière énergie, la relation énergétique s'écrit :

$$E_{\text{totale}} = \int_0^1 \left[-|\nabla I|^2 + \alpha(t) \left(\frac{dv}{dt} \right)^2 + \beta(t) \left(\frac{d^2v}{dt^2} \right)^2 \right] dt$$

Les contours actifs – « Snakes »

La résolution de cette équation est généralement effectuée sous forme variationnelle. La discrétisation de la courbe de contour peut se concevoir de deux façons:

- selon les différences finies : les éléments de la courbe sont réduits en des points auxquels sont attachés les éléments mécaniques (masse, raideur) de la courbe concentrée en ces points,
- selon les éléments finis, on remplace chaque portion de courbe par des segments élémentaires et les éléments mécaniques sont calculés sur ces segments.

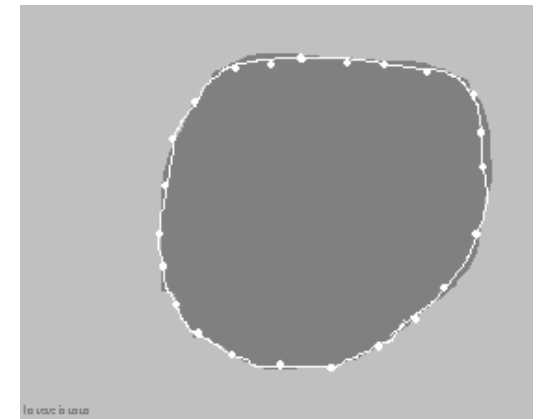
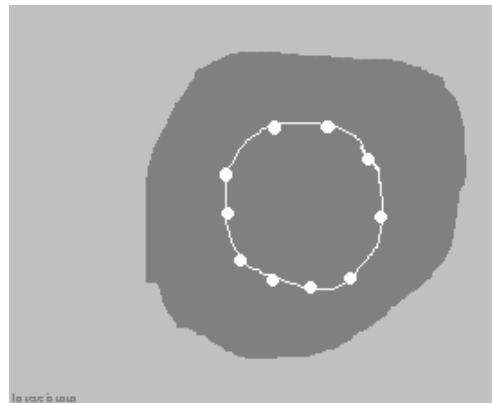
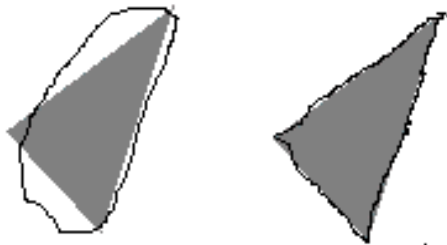
La convergence vers un contour convenable va dépendre du choix des paramètres initiaux et de l'estimation de ∇I . L'utilisation d'un filtre de Deriche est souvent préconisée.

Les contours actifs – « Snakes »

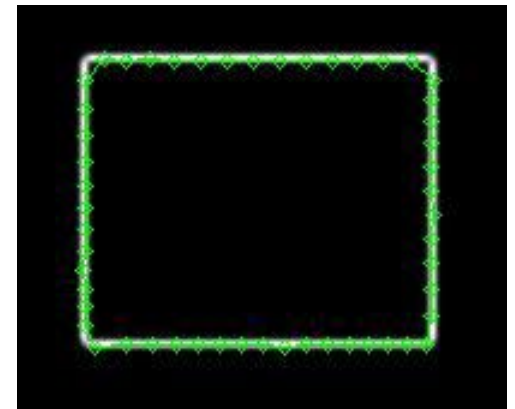
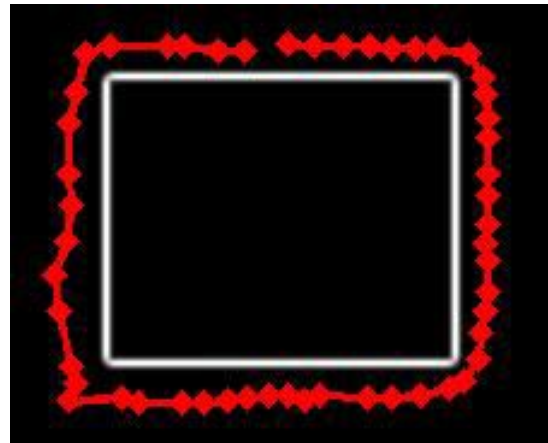
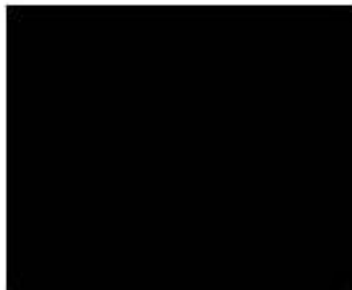
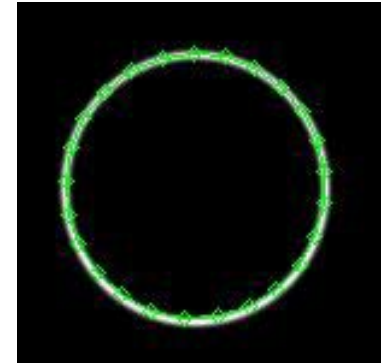
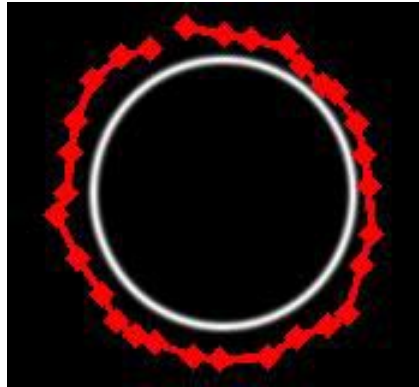
3 types de contours actifs sont possibles :

- les contours fermés,
- Les contours à extrémités libres,
- les contours à extrémités fixées en positions et dérivées.

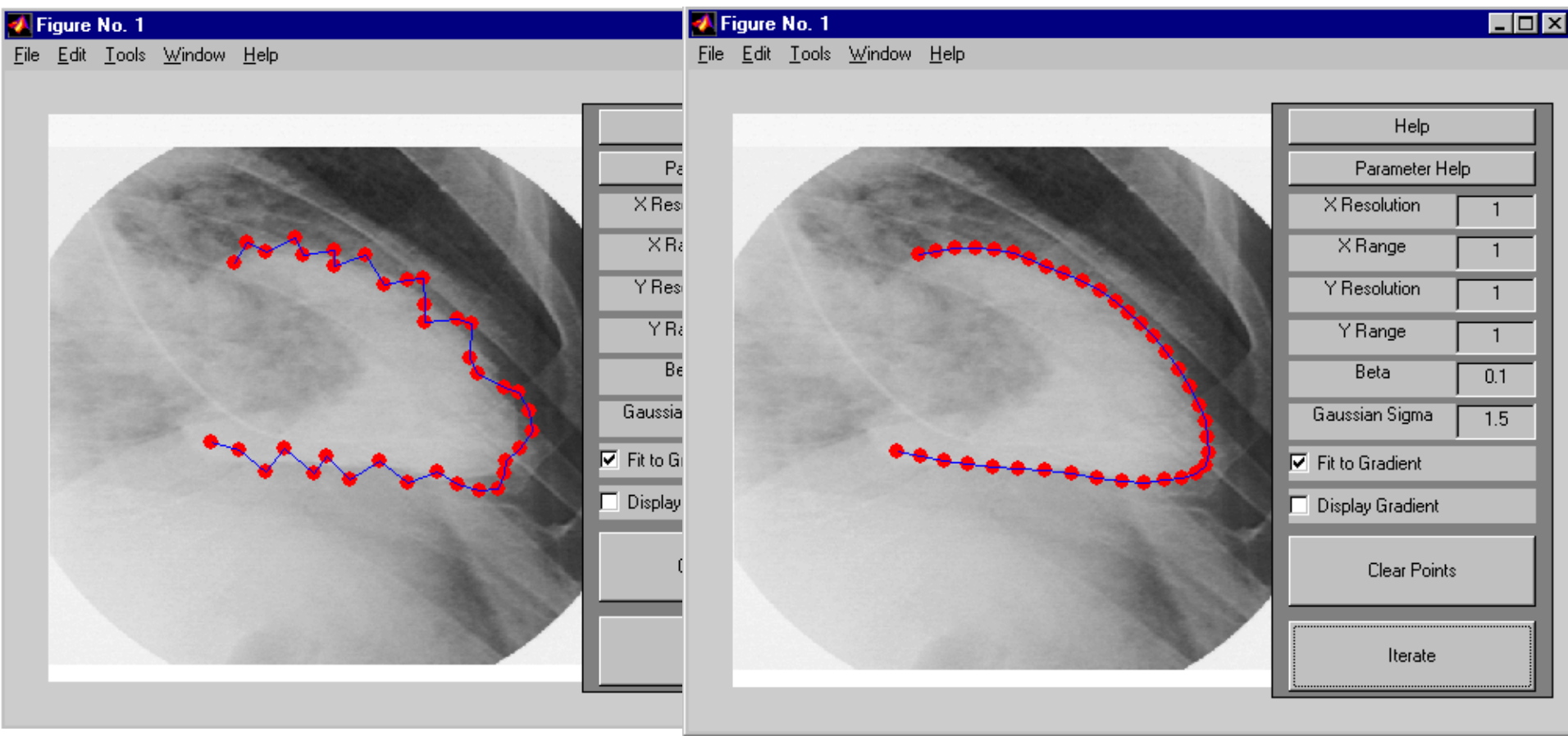
Exemples :



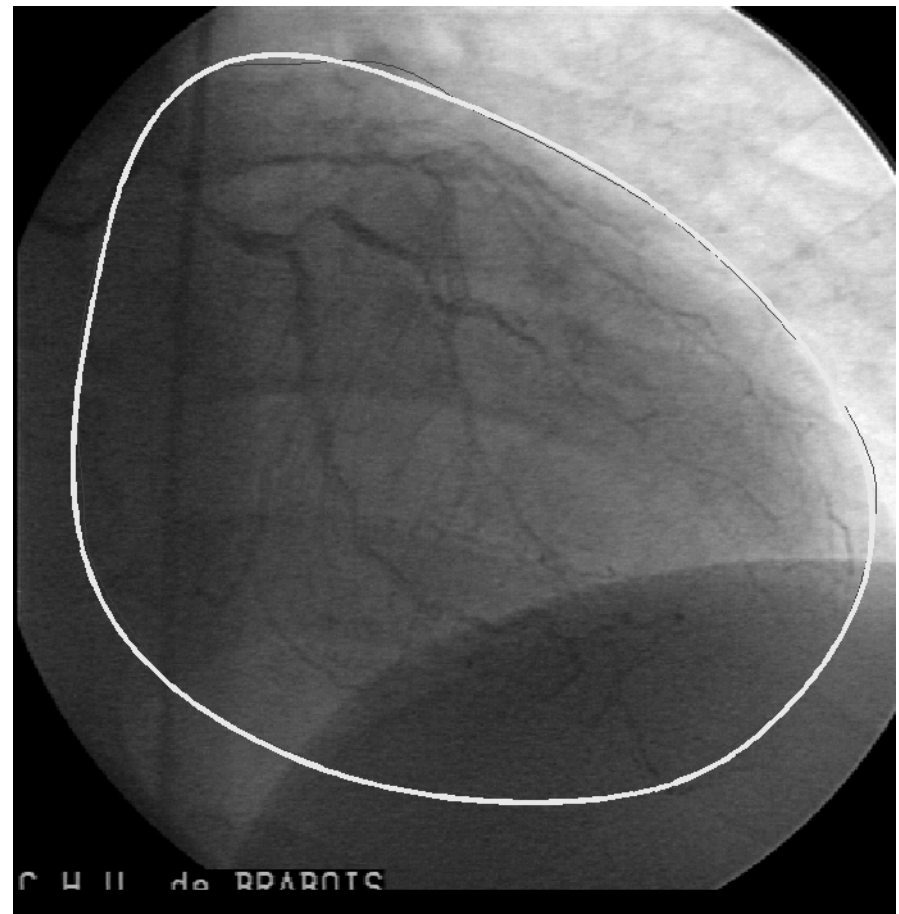
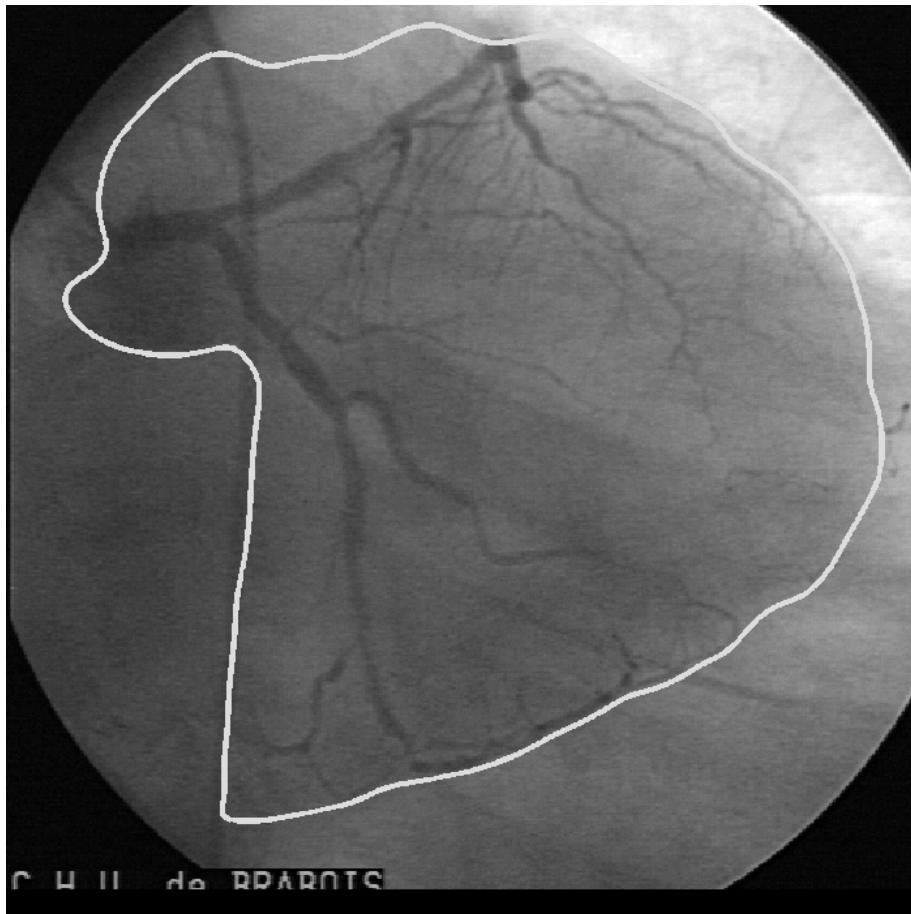
Les contours actifs – exemples



Les contours actifs – exemples



Les contours actifs – exemples



Segmentation par régions

Recherche de zones possédant des propriétés communes

- luminosité
- texture

Méthode sur l'histogramme

1. Pour des images qui présentent des classes évidentes (pièces industrielles, documents écrits) plus généralement pour les images très contrastées,
2. Pour les images définies sur plusieurs canaux – imagerie multispectrales.

Principe

- isoler les pics de l'histogramme,
- calculs des seuils par minimisation des erreurs (1ère et 2ème espèce).

Inconvénient

- les dépendances entre deux pixels spatialement voisins ne sont pas prises en compte.

Segmentation par régions

Segmentation par division

- définition d'un critère d'homogénéité,

→ Test du critère sur la totalité de l'image.

Si le critère est validé → arrêt de la segmentation

Sinon → découpage de l'image en zones plus petite puis
ré-application de la méthode

Segmentation par régions

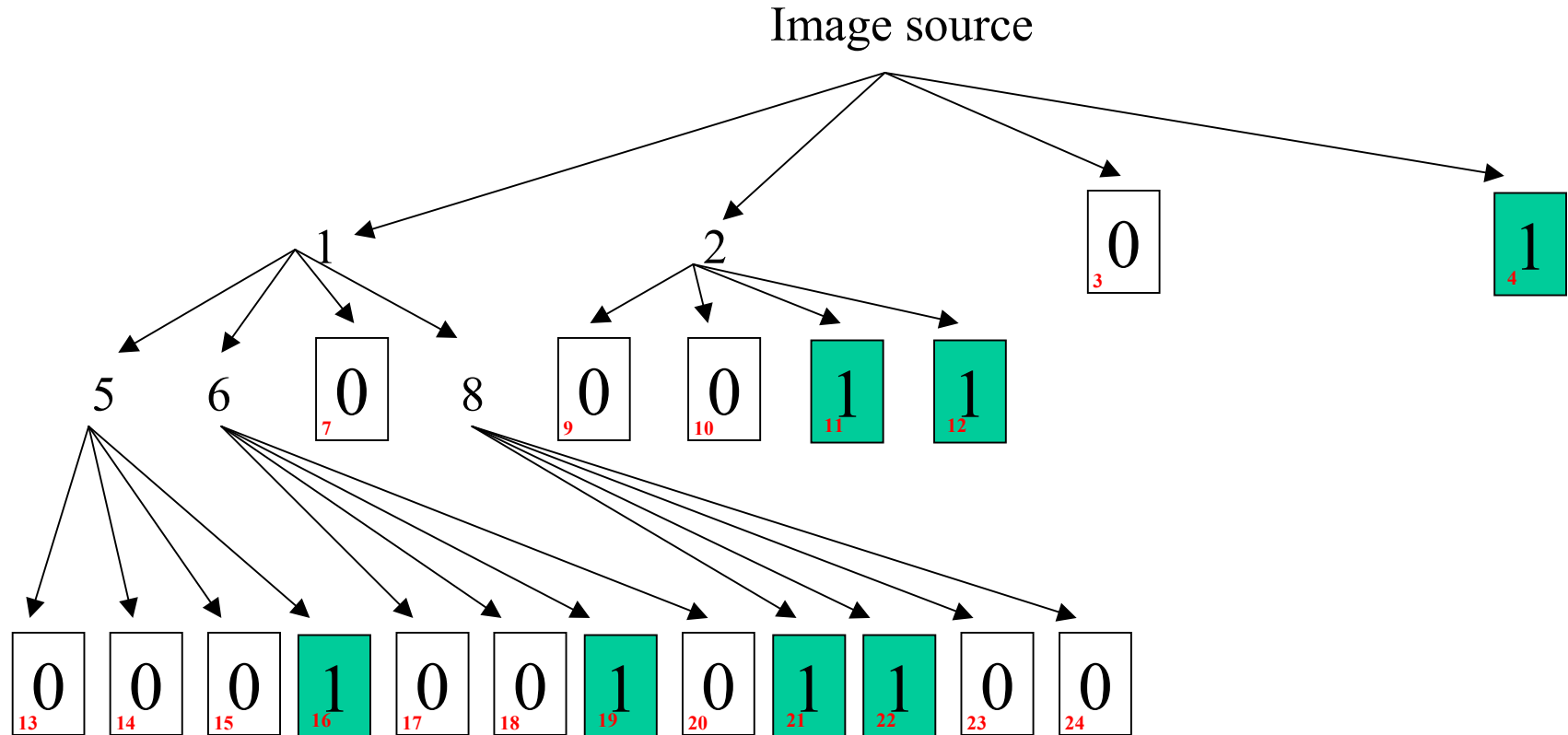
0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0
0	0	1	1	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1

1	2
0	1
3	4

		0	0
5	6	9	10
0		1	1
7	8	11	12
0		1	1

0	0	0	0	0	0
13	14	17	18		
	1	1			
15	16	19	20		
		1	1	1	1
0		21	22		
		23	24		
0				1	1

Représentation arborescente



Croissance de régions

Principe

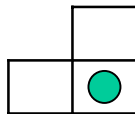
- exploration de l'image à partir de petites régions puis croissance de ces régions si elles satisfont à un critère d'homogénéité.

Paramètres

- le critère d'homogénéité
- le critère d'arrêt

Exemple

- soit la fenêtre suivante :



- chaque couleur est caractérisée par un label qui représente une région différente.

Croissance de régions

Algorithme

```
Pour chaque pixel  $I(i,j)$  faire
| si Critère de  $I(i,j)$ =critère de  $I(i-1,j)$ 
|   alors Couleur de  $I(i,j)$ = Couleur de  $I(i-1,j)$ 
|   sinon
|       si Critère de  $I(i,j)$ =critère de  $I(i,j -1)$ 
|       |   alors alors Couleur de  $I(i,j)$ = Couleur de  $I(i,j -1)$ 
|       |   sinon Couleur de  $I(i,j)$ = NouvelleCouleur
|   si Critère de  $I(i,j)$ =critère de  $I(i-1,j)$  et (Critère de  $I(i,j)$ =critère de  $I(i-1,j)$ )
|   |   alors Fusionner les régions en donnant la même couleur aux points
|   |        $I(i,j), I(i-1,J), I(i,j-1)$ 
finPour
```

Segmentation par rassemblement

Principe

- croissance de la région avant de passer à la suivante, sans parcours déterminé priori

Paramètres

- placement de germes initiaux
- croissance suivant un critère de similarité
- critère d'arrêt : convexité maximum, etc.

Inconvénients

- influence du placement des germes initiaux