

Segmentation : détection de contours et de régions

Plusieurs zones de niveaux de gris différents correspondant aux différents objets de la scène.

Il faut trouver une manière de repérer ces objets dans l'image

Deux grandes approches peuvent être envisagées pour extraire les zones pertinentes des images :

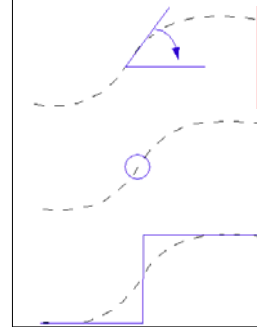
- ❖ on recherche des zones de niveaux de gris homogènes, c'est l'approche région
- ❖ on recherche les discontinuités dans la scène, c'est l'approche contour.

Nous allons nous intéresser à cette dernière méthode, la segmentation en région sera présentée ensuite.

Transparent 1

Segmentation : détection de contours

Contours = forte discontinuité. Regardons sur un signal à une dimension comment retrouver ces discontinuités.



❖Pente élevée de la transition
→ maxima locaux du gradient

❖Changement de concavité
→ passages par zéros de la dérivée seconde (du laplacien pour les images)

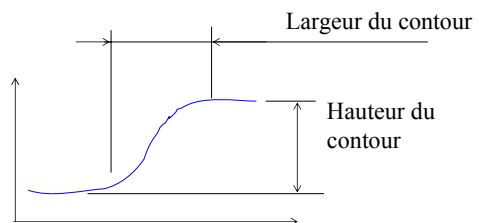
❖Forme particulière du signal
→ adaptation à un gabarit

Transparent 2

Contours

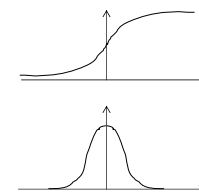
Zone de transition

Attributs : Largeur du contour, contraste
Contraintes de connexité et de géométrie



Transparent 3

Contours : approche gradient



→ Recherche des maxima locaux du gradient

Transparent 4

Contours : approche gradient

Dans le cas 2D des images, le vecteur gradient est défini au point de coordonnées (x,y) par :

$$\overrightarrow{Grad}(y,x) = \begin{pmatrix} \frac{\partial I(y,x)}{\partial x} \\ \frac{\partial I(y,x)}{\partial y} \end{pmatrix} = \begin{pmatrix} I_x(y,x) \\ I_y(y,x) \end{pmatrix}$$

Le module du gradient est défini par :

$$G(y,x) = \sqrt{I_x(y,x)^2 + I_y(y,x)^2}$$

tandis que son orientation est définie par :

$$\Phi(y,x) = \arctan\left(\frac{I_y(y,x)}{I_x(y,x)}\right)$$

Transparent 5

Contours : approche gradient



Image originale

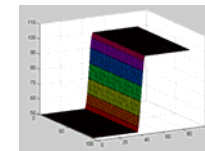
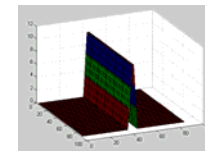
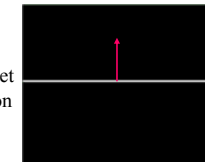


Image originale
sous forme surfacique



Gradient

Gradient et
orientation



Transparent 6

Approche gradient : calcul direct des dérivées

Calcul direct des dérivées

Il est réalisé de la façon la plus simple qu'il soit :

$$I_x(y,x) = I(y,x+1) - I(y,x) = (I * G_x)(y,x)$$

$$I_y(y,x) = I(y+1,x) - I(y,x) = (I * G_y)(y,x)$$

Ce qui correspond à une convolution avec les masques :

$$G_x = \begin{bmatrix} -1 & 1 \end{bmatrix} \quad \text{et} \quad G_y = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Malheureusement, ces dérivées sont très sensibles au bruit
→ filtrage préalable.

Transparent 7

Approche gradient : masque de Prewitt

Masque de Prewitt

Les masques dérivateur sont maintenant :

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

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

$$\begin{array}{l} \text{Lissage des lignes} \\ \text{ou des colonnes} \end{array} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

$$\begin{array}{l} \text{Dérivée des lignes} \\ \text{ou des colonnes} \end{array} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

On combine à la fois un filtrage et une dérivée

→ moins sensible au bruit que le calcul direct des dérivées.

Transparent 8

Approche gradient : masque de Sobel

Masque de Sobel

Même principal mais autre filtrage préalable

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

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

Transparent 9

Approche gradient : dérivée de gaussienne

Le filtrage précédent la dérivation peut être réalisé grâce à un masque gaussien G. Dans ce cas, le calcul de la dérivée en x se fait par :

$$Ix = \frac{\partial(I * G)}{\partial x} = I * \frac{\partial G}{\partial x}$$

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \Rightarrow \frac{\partial G(x, y)}{\partial x} = -\frac{x}{2\pi\sigma^4} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

Le paramètre σ règle le degré de lissage.

Même raisonnement pour la dérivée en y

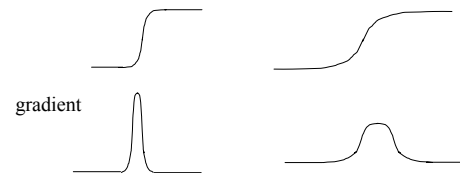
Transparent 10

Approche gradient : dérivée de gaussienne

Dualité filtrage / raideur de la transition :

Faible lissage

Fort lissage



Faible lissage = pic de haute amplitude et étroit

→ contours fins, beaucoup de bruit dans la dérivée.

Fort lissage = pic de plus faible amplitude et beaucoup plus large

→ contour épais, gradient moins perturbé par le bruit.

Transparent 11

Seuillage

Une fois la norme du gradient calculée en chaque point de l'image, il faut seuiller cette norme pour décider si un pixel fait partie ou non d'un contour.

Tous les pixels possédant une norme supérieure à un seuil S sont déclarés appartenir à un contour.

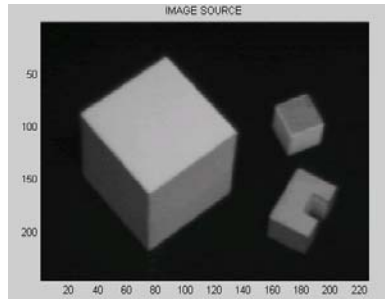
Tout le problème réside alors dans le choix du seuil :

- un seuillage trop bas nous amène à une **sur-détection** : on détecte beaucoup de bruit et des contours épais apparaissent

- Un seuillage trop haut amène à une **sous-détection** : contours non fermés

Transparent 12

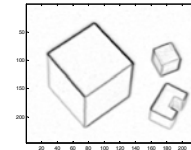
Exemple de seuillage



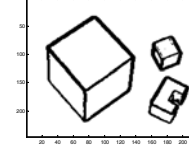
Transparent 13

Exemple de seuillage

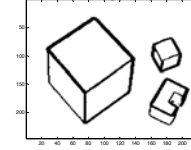
Gradient



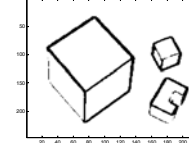
Seuillage à 10



Seuillage à 15



Seuillage à 20



Transparent 14

Seuillage avec hystérésis

-Il s'agit de diminuer l'importance du seuil lors du seuillage du gradient

-Avant, il y avait un seuil unique qui amenait soit à des fausses alarmes, soit à des lacunes. Il était très difficile, voir impossible, de trouver un seuil adapté à toute l'image.

- On introduit maintenant 2 seuil : un seuil haut et un seuil bas.

- ✓ Si norme > seuil haut → contour sur 1
- ✓ Si norme < seuil bas → pas de contour 0
- ✓ Si seuil bas < norme < seuil haut → contour de fermeture 2

Les contours hypothétiques de fermeture sont transformés en contours sûrs s'ils sont adjacents à un contour déjà codé à 1.

Transparent 15

Seuillage avec hystérésis

1	0	1	0	2	0	1	0	0	2
2	3	3	4	0	2	3	1	0	0
0	2	2	1	1	2	3	3	1	2
0	0	3	2	3	2	2	0	2	0
9	10	11	11	7	9	10	9	9	10
8	11	9	11	9	10	9	11	11	10
10	7	8	10	9	7	11	11	7	8
8	8	10	10	8	10	11	8	8	10

Calcul de la dérivée
 $|I(i-j)-I(i+1,j)|$
 $+ |(I(i-j)-I(i,j+1))|$

4	2	5	6	4	3	3	1
3	1	4	1	1	1	2	3
2	4	2	3	1	1	5	3
11	9	8	8	9	9	10	7
4	4	2	4	2	2	4	2
7	2	3	1	5	6	0	8
1	4	0	3	5	1	6	1

Seuillage simple
 $S=5$

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

Seuillage par
Hystérésis
 $S_b=5, S_h=8$

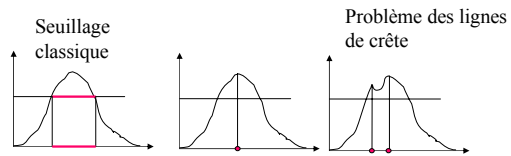
0	0	2	2	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	2	0
1	1	1	1	1	1	1	2
0	0	0	0	0	0	0	0
2	0	0	0	2	2	0	1
0	0	0	0	2	0	2	0

Transparent 16

Ligne de crête

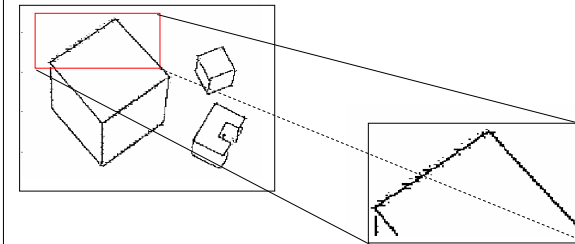
Le seuillage par hystérésis permet de limiter le nombre de fausses alarmes et de lacunes mais il amène aussi à des contours épais lorsqu'un fort filtrage a été réalisé.

Pour revenir à des contours plus fins, on procède à une recherche des lignes de crête. Un point est gardé comme contour s'il est supérieur au seuil et s'il est maximum local dans une direction



Transparent 17

Ligne de crête

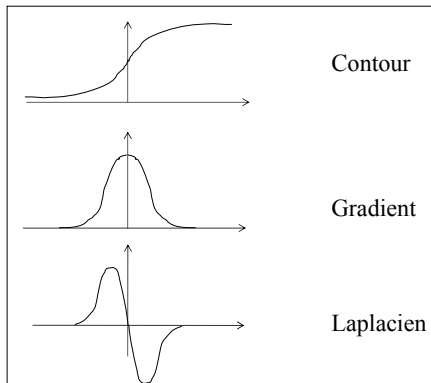


Les lignes de crête ne résolvent pas tous les problèmes.

- ❖ Elles peuvent amener à des contours parasites.
- ❖ Elles ne permettent pas de garder des contours fermés, même si le seuil choisi est tel que les contours l'étaient.

Transparent 18

Approche Laplacien



Transparent 19

Approche Laplacien – opérateur discret 2D

Opérateur Laplacien discret 2D

Contours = passages par zéro du laplacien.

Celui-ci peut-être estimé grâce à une convolution avec le masque :

0	1	0
1	-4	1
0	1	0

Comme la dérivée second est très sensible au bruit, on préfère utilisé le laplacien à partir de dérivées de gaussiennes :

$$\nabla^2 (IM \otimes G) = IM \otimes \nabla^2 G \text{ avec } G(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

$$\nabla^2 G(x,y) = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2} = -\frac{1}{\pi\sigma^4} \frac{1-x^2-y^2}{2\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

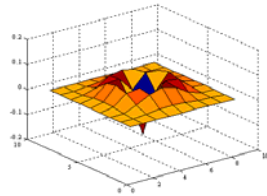
Transparent 20

Approche Laplacien – dérivée de Gaussienne

La variance σ^2 règle le degré de lissage

La dimension du masque dépend de la variance

Exemple : $\sigma = 1$, taille du masque : -4σ à 4σ



Transparent 21

Approche dérivative précédée d'un filtrage non linéaire

Le filtre de Nagao se prête très bien à une détection de contour : il supprime le bruit sans toucher à la raideur de la transition.

Une façon de détecter les contours est donc de réaliser un filtrage de Nagao, suivi d'un calcul de dérivée simple :

$$Gx = \begin{bmatrix} -1 & 1 \end{bmatrix} \quad \text{et} \quad Gy = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Cette méthode amène à de très bons résultats de détection.

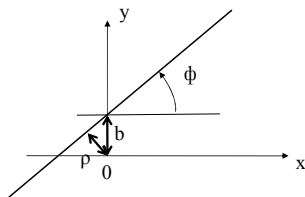
Transparent 22

Transformée de Hough

Cette méthode a pour but de détecter des droites lorsque l'on a les points de contour.

Considérons une droite dans le plan (x,y) . Elle a pour équation :

$$y = ax + b \quad \text{ou} \quad x \sin \phi + y \cos \phi = \rho \quad (\text{équation 1})$$



Transparent 23

Transformée de Hough

Si $\phi \in [0, \pi]$, les paramètres de la droite (ϕ, ρ) sont uniques.

La transformée de Hough est une transformation qui permet de passer du plan image (x,y) au plan des paramètres.

Le plan image est parfaitement défini, le plan des paramètres devra être discrétisé en $N \times M$ cellules

Par chaque point de contour, on peut en théorie faire passer une infinité de droite. Ici, comme on a discrétisé les orientations, on pourra seulement faire passer M droites.

Transparent 24

Transformée de Hough

Considérons un point contour (x_i, y_i) .

Pour chaque orientation ϕ , on peut déterminer avec l'équation 1 le ρ . On incrémente alors de un la case (ϕ, ρ) correspondante dans l'espace des paramètres.

Chaque point de contour vote donc pour M points dans l'espace des paramètres (un pour chaque orientation).

Si une droite est présente dans l'image, chaque point de contour de cette droite va voter pour la même case de l'espace des paramètres provoquant ainsi une accumulation. En la détectant, on a les coordonnées de la droite.

Transparent 25

Transformée de Hough généralisée

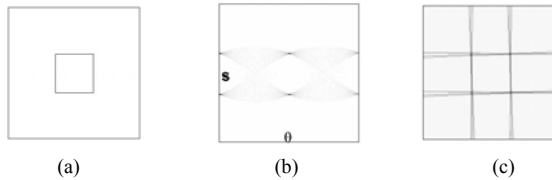
Une amélioration peut-être amenée si en plus de connaître les points de contour (image binaire), on connaît l'orientation de ces points (facile à avoir : c'est l'orientation du gradient).

Pour chaque point contour, on calcule ρ uniquement pour l'orientation perpendiculaire à celle du gradient. L'accumulation dans l'espace des paramètres est alors beaucoup plus nette.

Cette méthode peut être étendue à des courbes de forme quelconque.

Transparent 26

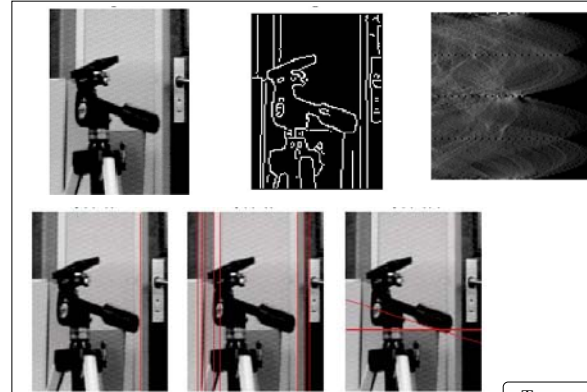
Transformée de Hough : un exemple



- (a) Image originale
- (b) Transformée de Hough : espace (ϕ, ρ)
- (c) Principales droites reconstruites

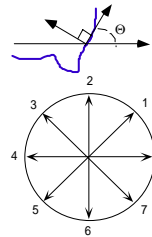
Transparent 27

Transformée de Hough : un autre exemple



Transparent 28

Codage des orientations - Freeman



L'orientation codée est celle de la tangente au contour, normale à l'orientation du gradient

Référence angulaire

La résolution angulaire avec 4 masques est de 45°
Codage de Freeman : 8 états

Un contour est décrit par la succession des codes de Freeman de ses pixels (chaînes de Freeman). Ce code est invariant :
 → strictement en translation
 → à une constante additive près, modulo 8, en rotation
 → à la longueur des sous-chaînes près en homothétie.

Transparent 29

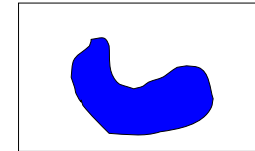
Régions

Régions :

Zone homogène au sens d'un critère donné :
Radiométrie, chrominance, texture, ...

Plusieurs attributs :

Aire, périmètre, centre de gravité, axes principaux, ...



Transparent 30

Segmentation par agrégation de pixels

- ❖ Initialisation : chaque pixel forme une région.
- ❖ On regroupe les régions avec un double critère de similarité des niveaux de gris et d'adjacence.
- ❖ Le critère de similarité peut par exemple être : la variance des niveaux de gris de la région R est inférieure à un seuil.
- ❖ Le principe de l'agrégation de pixel est le suivant :
 - on choisit un germe et on fait croître ce germe tant que des pixels de son voisinage vérifient le test d'homogénéité.
 - Lorsqu'il n'y a plus de pixel candidat dans le voisinage, on choisit un nouveau germe et on itère le processus.

Transparent 31

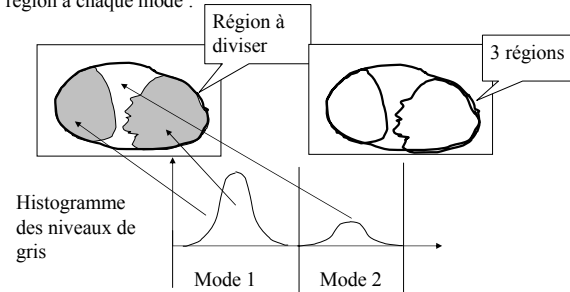
Segmentation par division

- ❖ Initialisation : une seule région.
- ❖ Si la région n'est pas homogène (critère d'homogénéité), elle est divisée en plusieurs régions, sinon, le processus se termine.
- ❖ Chaque région nouvellement créée est potentiellement redivisée en plusieurs régions si elle n'est pas homogène.
- ❖ Lors de la division, on peut rajouter un critère d'arrêt sur la taille des régions.

Transparent 32

Segmentation par division

- ❖ Une technique consiste à déterminer les différents modes de l'histogramme des niveaux de gris et à affecter une ou plusieurs régions à chaque mode :



Transparent 33

Segmentation par division-fusion

- ❖ Cette méthode consiste à reprendre le résultat de la division et à essayer de fusionner des régions qui ont été malencontreusement séparées.
- ❖ Pour ce faire, on utilise un outil très précieux : le graphe d'adjacence des régions.

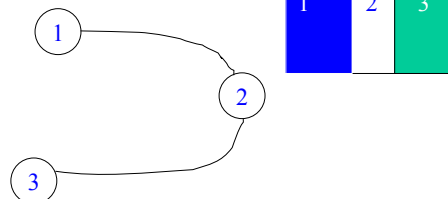
Transparent 34

Graphe d'adjacence

Graphe d'adjacence :

nœud = une régions

lien = adjacence entre deux régions



Transparent 35

Graphe d'adjacence

Sous matlab, la fonction *Imgcxreg()* renvoie un tableau :

1	2	4	5	6	8	
2	1	3	4	7	8	
3	2	4	5	7		
4	1	2	3	5		
5	1	3	4	6	7	
6	1	5	8			
7	2	3	5	8		
8	1	2	6	7		

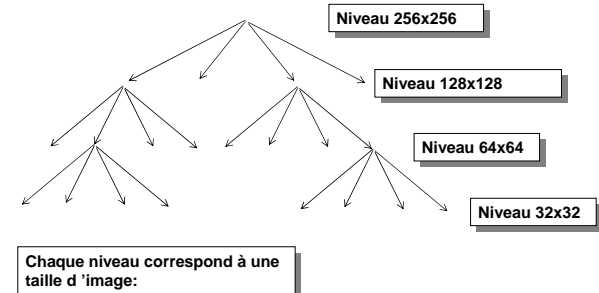
Transparent 36

Décomposition en Quadtree

- Méthode récursive de découpage d'une image
- Un carré est découpé en 4 quarts s'il n'est pas jugé homogène (seuil)
- Les 4 nouveaux carrés sont ses « fils » dans un arbre (quadtree)
- Cette méthode peut servir à compresser une image ou à la segmenter

Transparent 37

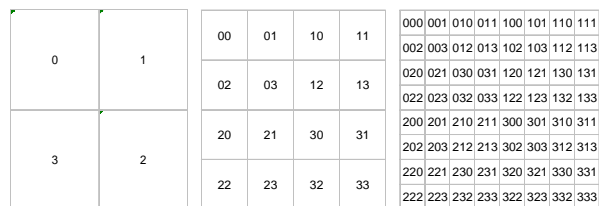
Schéma d'un quadtree



Transparent 38

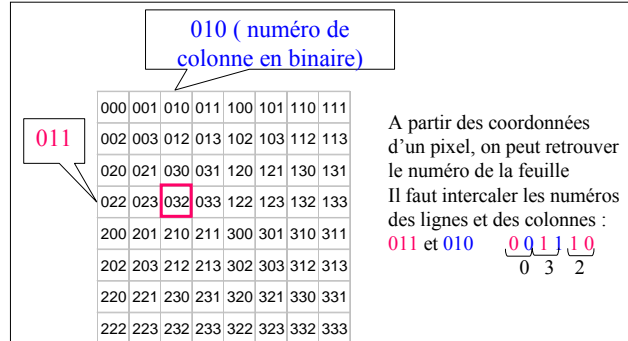
Le quadtree linéaire

Le but est de coder chaque feuille du quadtree de manière à pouvoir retrouver sa position. Principe de codage retenu :



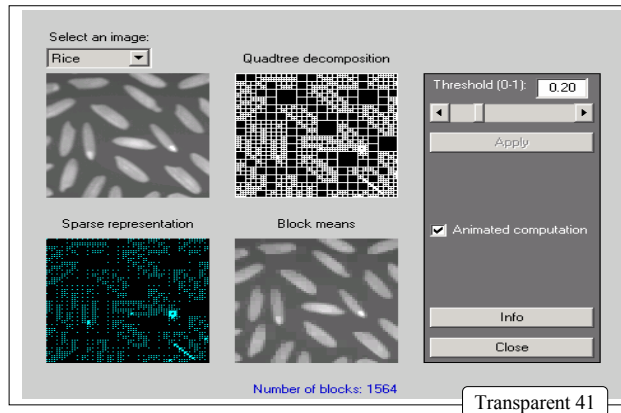
Transparent 39

Le quadtree linéaire



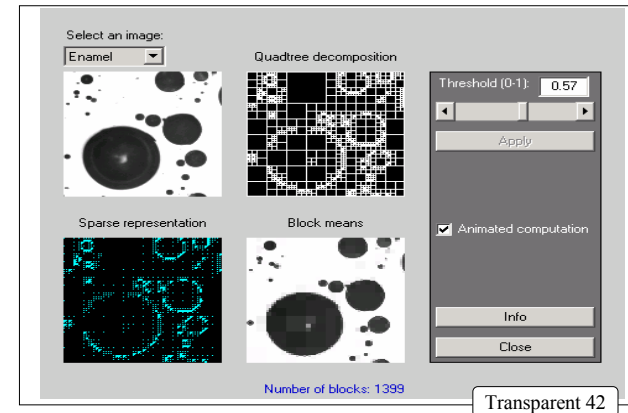
Transparent 40

Exemple de Matlab



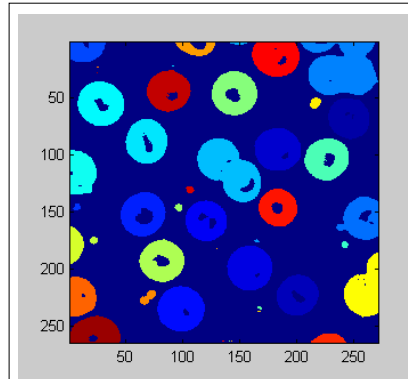
Transparent 41

Exemple de Matlab



Transparent 42

Etiquetage

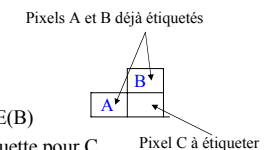


Etiquetage de chaque région par un numéro (traduit ici par une couleur)
On appelle composante connexe une telle région
Il y a 46 régions dans cette image

Transparent 43

Etiquetage par balayage

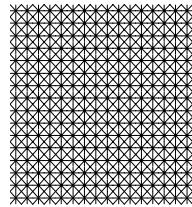
1. Binarisation de l'image
2. On déplace un masque en L
 1. Si C et A proches, $E[C]=E(A)$
 2. Sinon, si C et B proches, $E[C]=E(B)$
 3. Sinon, on crée une nouvelle étiquette pour C
 4. Si C proche de B et A, on remplace les étiquettes B et A par $\min(E(B), E(A))$



L'instruction (4) correspond à la présence de concavités vers le haut ou à des escaliers; elle est pénalisante et peut être remplacée par d'autres approches

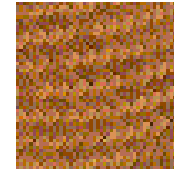
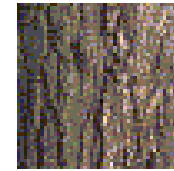
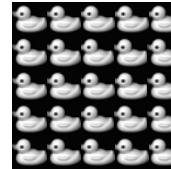
Transparent 44

Qu'est ce que la texture ?



Transparent 45

Qu'est ce que la texture ?



Transparent 46

Deux grandes approches

- ✓ L'approche structurale :
 - Un ou plusieurs motifs se répètent
 - ➔ Détection de l'élément de base et de son arrangement
- ✓ L'approche statistique :
 - Pas de primitives élémentaires
 - ➔ Utilisation des statistiques

Transparent 47

L'approche structurale

- ✓ Elle fait appel à des éléments de reconnaissance des formes
- ✓ Problème non trivial :
 - Détection de l'élément de base
 - Détection des éléments d'arrangement spatial
- Bien souvent, l'élément de base évolue au cours de l'image, évolution due par exemple à un effet perspectif. La règle d'arrangement spatiale évolue elle aussi, rendant le problème très complexe

Transparent 48

L'approche statistique

✓ On ne recherche plus de primitives élémentaires, tout est basé sur des notions de statistique déterminées sur un ensemble de pixel.

✓ Il faut considérer des voisinages de taille importante autour du pixel d'intérêt pour que la notion de statistique ait un sens (au moins 8x8 pixels)

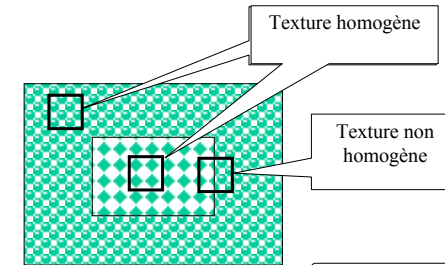
✓ Ceci introduit bien sur des problèmes d'imprécision le long des frontières. Il est très difficile de faire de la segmentation sur des critères texturaux de manière précise.

Transparent 49

L'approche statistique

→ Notion de **secteur**

→ Problème aux frontières des régions



Transparent 50

1 - Histogramme des différences de niveaux de gris

$$F(x,y) = |f(x,y) - f(x+\Delta x, y+\Delta y)|$$

Soit H, l'histogramme d'un secteur de F. On peut extraire :

- ✓ La valeur maximale,
- ✓ La moyenne,
- ✓ La variance,
- ✓ Les moments, ...

Inconvénient :

- ✓ Calcul de plusieurs histogrammes.

Transparent 51

1 - Histogramme des différences de niveaux de gris

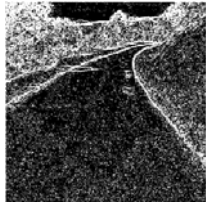
- ✓ Exemple de résultat obtenu avec $\Delta x=1$ et $\Delta y=0$
- ✓ Histogramme calculé avec 64 niveaux de gris
- ✓ Caractéristique de texture : valeur max de l'histogramme
- ✓ Taille secteur : 16x16 pixels



Transparent 52

2 - Méthode fondée sur le gradient de la fonction luminance

Les contours d'une image semblent caractériser la texture de celle-ci.



→ On peut caractériser la texture par le nombre de petits contours dans une fenêtre, ...

Inconvénients :

- ✓ Nécessite un seuil pour la détection de contours.
- ✓ Méthode coûteuse en temps de calcul.

Transparent 53

3 - Filtres de Gabor

Filtre de Gabor à deux dimensions :

$$h(x,y) = \exp\left(-\frac{1}{2}\left[\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right]\right) \cos(2\pi u_0 x + \phi)$$

En faisant varier les u_0 , ϕ , σ_x et σ_y , plusieurs filtres h_j sont obtenus.

Une image résultat f_j est déterminée pour chaque filtre :

$$f_j(x,y) = f(x,y) * h_j(x,y)$$

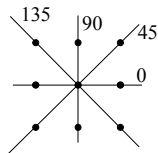
Plusieurs paramètres peuvent être extraits comme par exemple, la moyenne absolue de la déviation :

$$m_j = \frac{1}{n^2} \sum_{x,y=0}^{n-1} |f_j(x,y)|$$

Transparent 54

4 - Les matrices de cooccurrences

Elles nécessitent la définition d'un voisinage.



Dimension :
(Nb de niveaux de gris)²

Chaque élément de la matrice vaut :

$$p_{\delta,\theta}(i,j) = \sum_x \sum_y 1[f(x,y)=i] 1[f(x,y')=j]$$

$$P_{\delta} = P_{\delta,0} + P_{\delta,45} + P_{\delta,90} + P_{\delta,135}$$

Transparent 55

Exemple de calcul de matrice

secteur

0	6	0	6
6	0	6	0
0	6	0	6
6	0	6	0

secteur

0	0	5	5
0	0	5	5
0	0	5	5
0	0	5	5

secteur

0	2	5	2
4	2	3	6
3	0	1	4
6	4	2	1

Matrice

0	0	0	0	0	0	12
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
12	0	0	0	0	0	0

Matrice

8	0	0	0	0	4	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
4	0	0	0	0	8	0
0	0	0	0	0	0	0

Matrice

0	1	1	1	0	0	0
1	0	1	0	1	0	0
1	1	0	1	2	2	0
1	0	1	0	0	0	1
0	1	2	0	0	0	1
0	0	2	0	0	0	0
0	0	0	1	1	0	0

Transparent 56

Exemples de paramètres extraits

- ✓ Moment second ou énergie

$$\sum_i \sum_j [P(i, j)]^2$$

- ✓ Moment des différences inverses ou homogénéité

$$\sum_i \sum_j \frac{1}{1+(i-j)^2} P(i, j)$$

- ✓ Avantages :

- Beaucoup de paramètres peuvent être extraits de la même matrice : 14 ont été définis par Haralick en 1973
- Paramètres facilement extractibles et discriminants

Transparent 57

Exemples de résultat

Paramètre : homogénéité

Taille du secteur : 16x16 pixels

Nombre de niveaux de gris : 64



Transparent 58

8 – Transformée de Fourier

La transformée de Fourier bi-directionnelle peut être utilisée pour caractériser la texture :

$$F(u, v) = \frac{1}{n} \sum_{l, m} f(l, m) \exp(-2i\pi(ul + vm))$$

Les textures vont produire des concentrations d'énergie dans le spectre de Fourier.

Transparent 59

Transformée de Fourier : caractérisation de texture

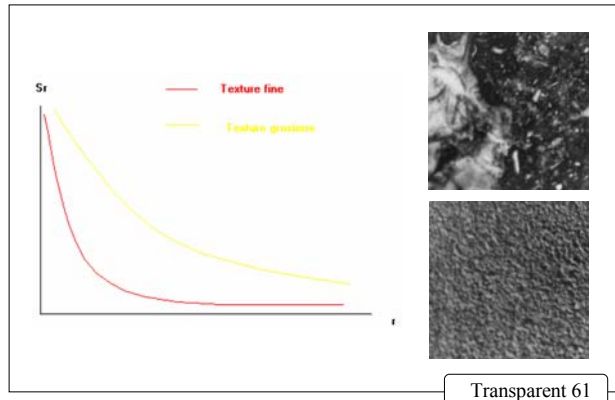
F Transformation de Fourier

$$S_r(r) = \int_0^\pi |F(r, \theta)| d\theta$$

$$S_\theta(\theta) = \int_0^R |F(r, \theta)| dr$$

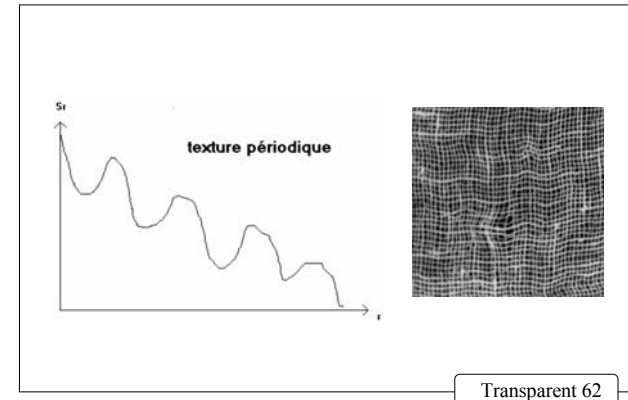
Transparent 60

Transformée de Fourier : caractérisation de texture



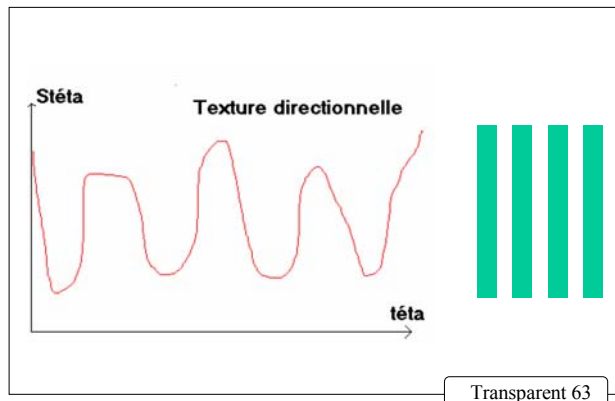
Transparent 61

Transformée de Fourier : caractérisation de texture



Transparent 62

Transformée de Fourier : caractérisation de texture



Transparent 63

10 - Masque de Laws

Ensemble de masque 3x3 ou 5x5 déterminés de manière expérimentale.
Les caractéristiques de texture : somme en valeur absolue des pixels des images filtrées.

Exemple de filtre utilisé :

$$\begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix}$$

Avantages :

- ✓ Très simple à mettre en œuvre.
- ✓ Donne de bons résultats.

Inconvénients :

- ✓ C'est une méthode expérimentale qui ne repose sur aucune théorie.



Transparent 64

Conclusion

- ✓ Pas de définition universelle de la texture.
- ✓ Plusieurs méthodes permettent d'obtenir des paramètres de texture.
- ✓ Aucune d'entre elles ne peut être a priori considérée comme meilleure.
- ✓ Bien souvent, le choix se fera en fonction de l'application

Transparent 65