Segmentation d’images

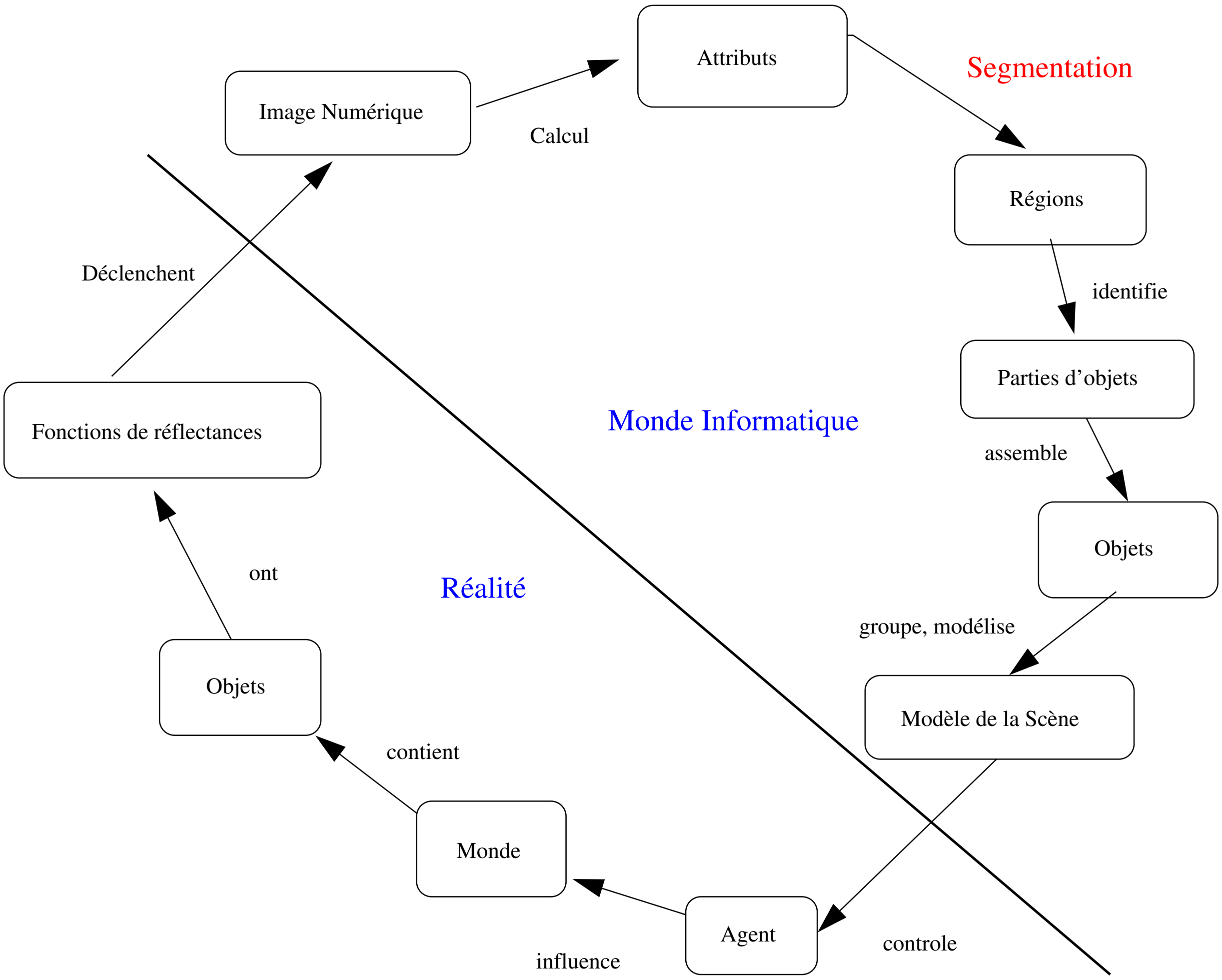
Luc Brun

6 janvier 2002

Références utilisées

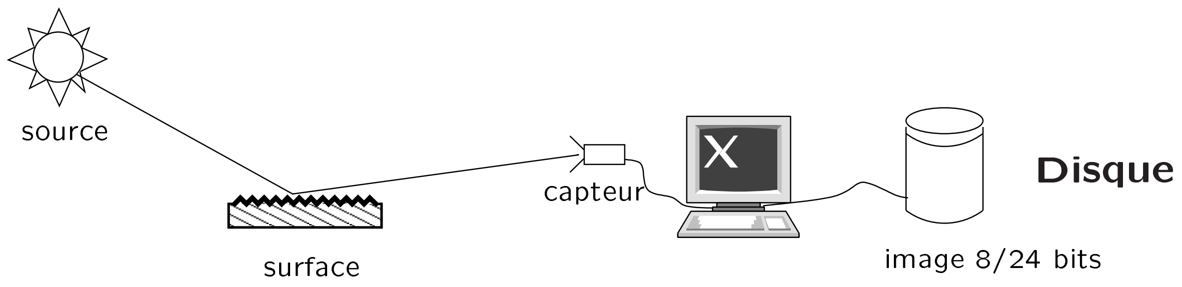
- Analyse d'images : filtrage et segmentation, J.-P. Cocquerez et S. Philipp Editions Masson
- Segmentation d'images à base topologiques, Thèse de doctorat, Luc Brun
  `http://www.univ-reims.fr/Labos/Leri/membre/luc`
  Voir les chapitres 4, 5 et 2

Place de la segmentation
dans un processus de vision



Le premier niveau d'abstraction d'un système de vision

Fonctions de Réflectance



Connaissant :

– Les paramètres et la position de la source
– Les paramètres de la surface
– Les paramètres et la position de la caméra

Calculer l'intensité (ou la couleur) de chaque pixel.

Fonctions de réflectance

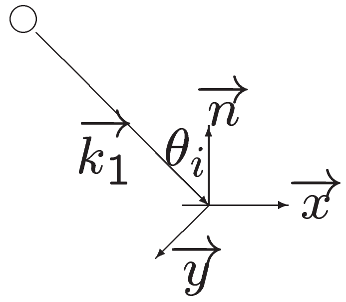
Le modèle lambertien

Modèle encore mal expliqué. Explication usuelle :

Le rayon penètre dans la surface où il est réfléchi aléatoirement. Une partie des rayons "ressortent" de la surface selon des directions aléatoires.

$$\boxed{I \approx K_{dl}.\cos(\theta_i)}$$

θ_i angle entre la normale et le rayon incident.
K_{dl} constante du lobe diffus (composante lambertienne)



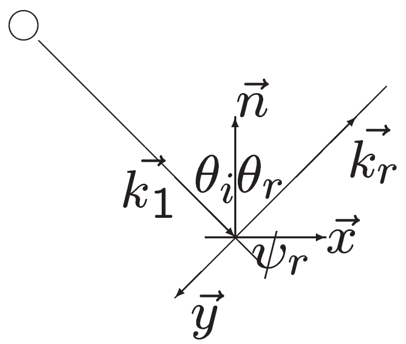
Fonctions de réflectance

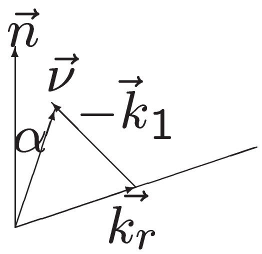
Le lobe spéculaire

Deux explications :

- Par l'electromagnétisme (Modèle de Beckman)
- Par l'optique géométrique (Modèle de Towrance-Sparow)

L'intensité est fonction de la source, de la normale et de l'observateur.





$\vec{k_r}$ rayon réfléchi $\qquad\qquad \vec{\nu} = \vec{k_r} - \vec{k_1}$

$$\boxed{I \approx K_{sl} e^{-\frac{\alpha^2}{2\sigma^2}}}$$

σ : ruguosité du matériaux. K_{sl} constante du lobe spéculaire.

Fonctions de réflectance

Le pic spéculaire

Direction spéculaire : $\alpha = 0$
$\vec{n}$ est la médiatrice de $\vec{k_r}$ et $-\vec{k}_1$.

Phénomène modélisé uniquement par les modèles physiques (Beckman) phenomène en :

$$\frac{sin(\nu_x)}{\nu_x} . \frac{sin(\nu_y)}{\nu_y} \approx \delta(\theta_i - \theta_r)\delta(\psi_r)$$

δ : fonction de dirac ($\delta(0) = 1, \delta(x) = 0 \forall x \neq 0$)

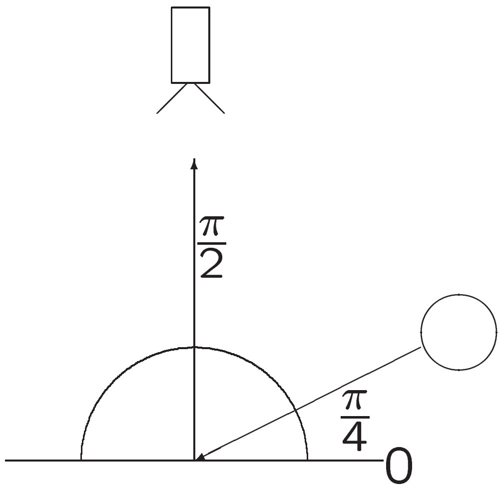
$$\boxed{I \approx K_{ss}\delta(\theta_i - \theta_r)\delta(\psi_r)}$$

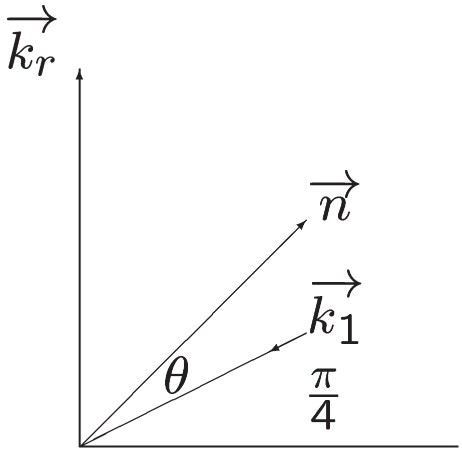
Fonctions de réflectance

Récapitulatif

$$I = K_{dl}\cos(\theta_i) + K_{sl}e^{-\frac{\alpha^2}{2\sigma^2}} + K_{ss}\delta(\theta_i - \theta_r)\delta(\psi_r)$$

Application : Éclairage d'une sphère (en fait un cercle) :





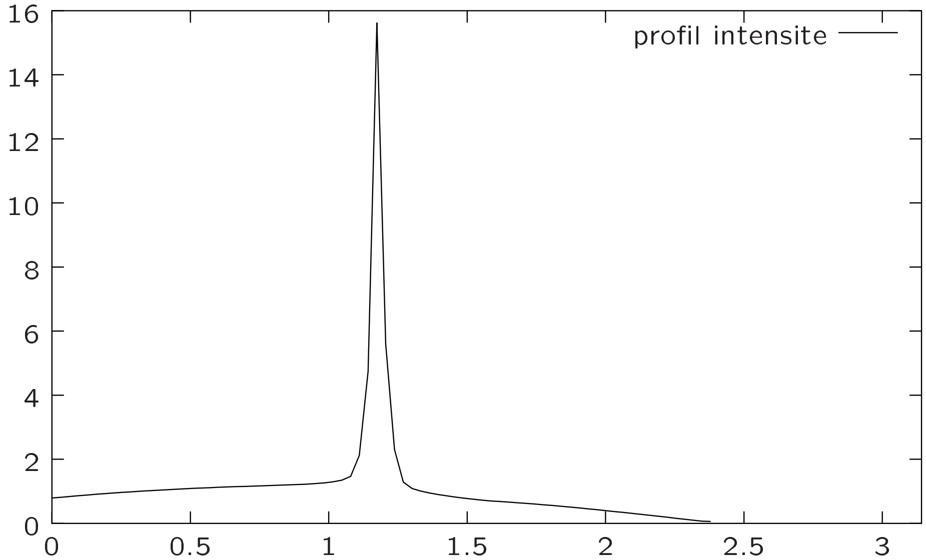
Calculer $I(\theta)$

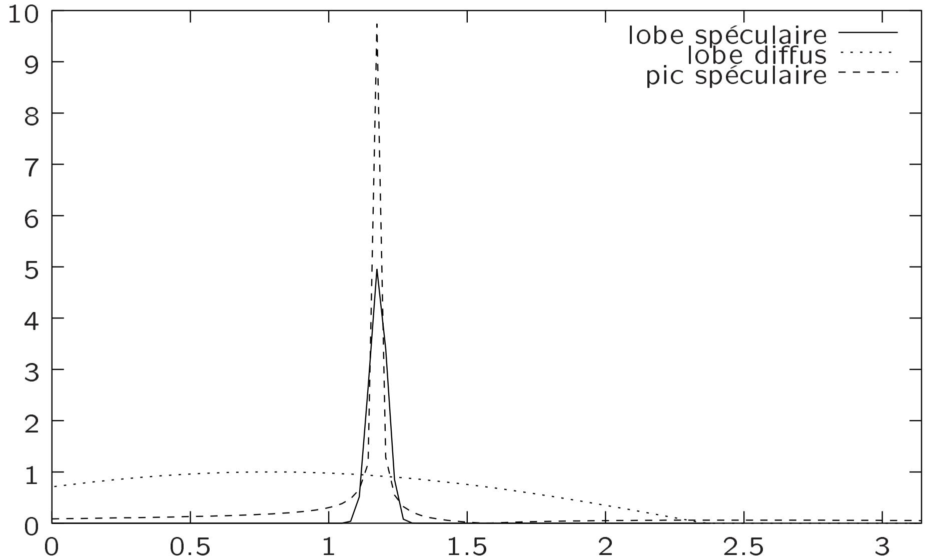
Fonctions de réflectance

Récapitulatif (Solution)

$$I(\theta) = K_{dl}cos\left(\theta - \frac{\pi}{4}\right) + K_{sl}e^{-\frac{(x-\frac{3\pi}{8})^2}{2\sigma^2}} + K_{ss}\delta\left(x - \frac{3\pi}{8}\right)$$

AN : $\sigma = 10^{-2},\ K_{dl} = 1,\ K_{sl} = 5, K_{ss} = 10$





Définition de la segmentation
Horowitz et Pavlidis 1976
Soient :
- X le domaine de l'image.
- I la fonction qui associe à chaque pixel une valeur I(x,y).
- P un prédicat d' homogénéité défini sur l'ensemble des parties de X.

La segmentation S de X est définie comme une partition de X en n sous-ensembles $\{R_1, \ldots, R_n\}$ tels que :

1. $X = \bigsqcup_{i=1}^{m} R_i$
   La segmentation S de X est une partition
2. $\forall i \in \{1, \ldots, n\}$ R_i est connexe.
   en composantes connexes
3. $\forall i \in \{1, \ldots, n\}$ $P(R_i) = vrai$
   dont toute région est au moins homogène
4. $\forall i, j \in \{1, \ldots, n\}^2, i \neq j$ / R_i est adjacent à R_j $\Rightarrow P(R_i \cup R_j) = faux$
   et qui ne peut être décomposé plus avant sans créer des régions dont la fusion resterait homogène.

Les différentes façons de considérer une image

- Un processus stochastique.
- Un vecteur aléatoire $(I[0,0], I[0,1], \ldots, I[n,m])$ si la taille de l'image I est $n \star m$.
- Une surface 3D (dans le cas d'un signal mono-dimensionnel).
- Un ensemble de données liées par des contraintes géométriques.
- La discrétisation d'un signal continu.
- ...

Les deux principales approches

1. L'approche région : Définir des mesures de similarité entre pixels
   $\rightarrow$ recherche de l'ensemble des pixels d'une région.
2. L'approche contour : Définir des mesures de dis-similarité entre pixels
   $\rightarrow$ recherche des frontières entre les régions.

L'approche contours

L'image est considérée comme la discrétisassion d'un signal continu :

$$I = Q \circ f$$

- I est l'image discrète,
- f le signal continu ($f \in C^2(\mathbb{R}^2, \mathbb{R})$: Images en niveaux de gris),
- Q un opérateur d'échantillonage.

Différentielle de f :

$$\begin{aligned} Df(p).\vec{n} &= \frac{\partial f}{\partial x}(p).n_x + \frac{\partial f}{\partial y}(p).n_y \\ &= \lim_{h \to 0} \frac{f(p) - f(p + (hn_x, hn_y))}{h} \end{aligned}$$

Seule la direction de n compte

$\rightarrow \|\vec{n}\| = 1$

Utilisation du gradient :

$$\triangledown f = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}$$

On a alors :

$$Df(p).\vec{n} = \triangledown f(p) \bullet n$$

Où $\bullet$ désigne le produit scalaire.
De plus :

$$\max_{\|\vec{n}\|=1} |Df(p).\vec{n}| = \max_{\|\vec{n}\|=1} |\triangledown f(p) \bullet \vec{n}| = \|\triangledown f(p)\|$$

En tout point la norme du gradient permet de connaître la variation maximum de la différentielle. Le maximum étant atteint pour $\vec{n}$ colinéaire à $\triangledown f(p)$ la direction du gradient donne la direction de plus grande variation de la fonction f.

Fort gradient $\Rightarrow$ présence de contour (peut être non maximum).
Maximum local de la différentielle première $\rightarrow$ zéro de la différentielle seconde.

$$\begin{aligned} D_2 f(p).\vec{n} = \; & \frac{\partial^2 f}{\partial x^2}(p).n_x.n_x + \frac{\partial^2 f}{\partial y^2}(p).n_y.n_y \\ & + 2\frac{\partial^2 f}{\partial x \partial y}(p).n_x n_y \end{aligned}$$

Extremum de $Df(p).\vec{n}$ en $\vec{n}$ $\Rightarrow$ zéros de $D_2 f(p).\vec{n}$ en $\vec{n}$.

Utilisation du Laplacien

$$\triangle f(p) = \frac{\partial^2 f}{\partial x^2}(p) + \frac{\partial^2 f}{\partial y^2}(p)$$

Le Laplacien étant invariant par rotation, le calcul des zéros du Laplacien ne fait pas intervenir la direction n de plus grande variation de $Df(p)$. L'utilisation du Laplacien évite donc de calculer le gradient en plus de la différentielle seconde.

Attention ! !

$$\triangle f(p) = 0 \not\Rightarrow \exists \vec{n} \ / \ D_2 f(p).\vec{n} = 0$$

Coïncidence uniquement si (Marr 1980) : les variations d'intensité sont linéaires sur la ligne de passage par zéro et sur les lignes parallèles dans un voisinage de p. Le laplacien peut donc être vu comme une bonne approximation de la différentielle seconde.

Les opérateurs de convolution

Diminution du bruit $\rightarrow$ convolution avec un filtre passe bas.

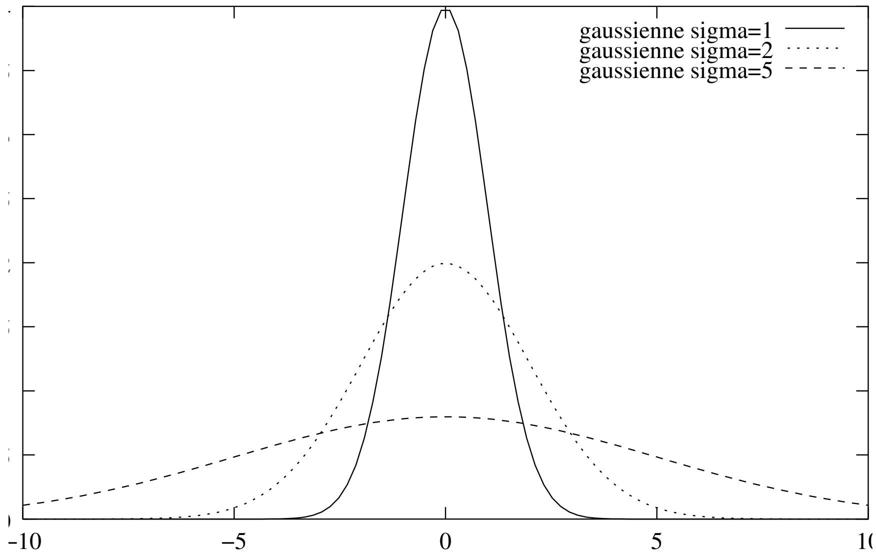
$$\begin{aligned} f * g(x,y) &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(u,v)g(x-u,y-v)dudv \\ &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x-u,x-v)g(u,v)dudv \end{aligned}$$

La Gaussienne :

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{x^2}{2\sigma^2}}$$

Le paramètre σ contrôle l'aplatissement de la Gaussienne et permet donc de contrôler la puissance du filtre. Une forte valeur de σ induit un fort lissage et vice-versa.

La Gaussienne



La convolution d'un signal avec la Gaussienne s'effectue en limitant celle-ci à un support fini $[-M_\epsilon, M_\epsilon]$ avec :

$$\forall x \in [-M_\epsilon, M_\epsilon] \quad G(x) > \epsilon$$

M_ϵ : fonction croissante de σ
$\rightarrow$ Plus on veut lisser plus il faut élargir le support

1. Approche Pyramidale.
2. Filtres récursifs

Séparabilité

$$G(x,y) = \frac{1}{\sigma^2(2\pi)} e^{-\frac{x^2+y^2}{2\sigma^2}} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{y^2}{2\sigma^2}}$$

$$G(x,y) = G(x)G(y)$$

La convolution de la fonction 2D f avec G donne donc :

$$\begin{aligned}
f * G(x,y) &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x-u, y-v) G(u,v) du dv \\
&= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x-u, y-v) G(u) G(v) du dv \\
&= \int_{-\infty}^{+\infty} G(u) \left(\int_{-\infty}^{+\infty} f(x-u, y-v) G(v) dv \right) du \\
&= \int_{-\infty}^{+\infty} G(u) \left(f *_y G \right) (x-u, y) du
\end{aligned}$$

$$f * G(x,y) = \left(G *_x (f *_y G) \right) (x,y)$$

où $*_x$ et $*_y$ représentent les convolutions par rapport aux variables x et y.

Séparabilité
(suite)

$$f * G(x,y) = (G *_x (f *_y G))\,(x,y)$$

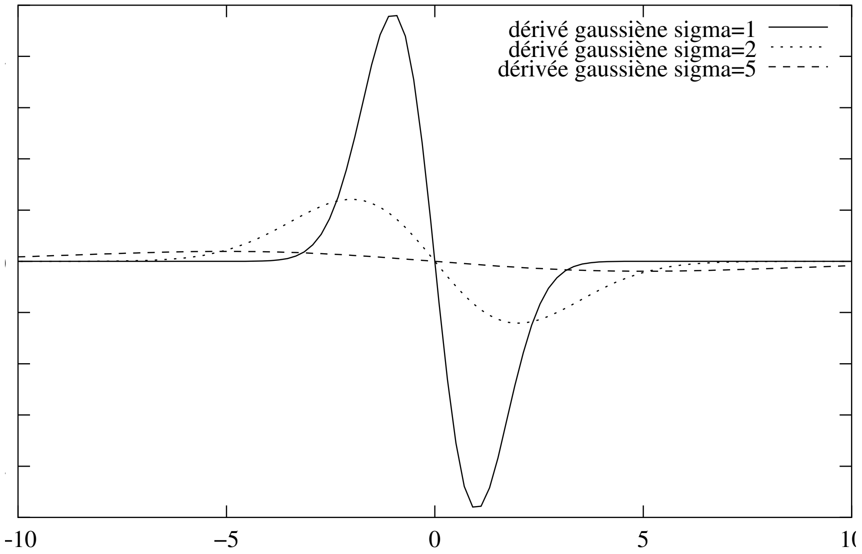
Utilisation de deux masques 1D de taille $[-M_\epsilon, M_\epsilon]$ plutot qu'un masque 2D de taille $[-M_\epsilon, M_\epsilon]^2$
$\rightarrow$ complexité $\mathcal{O}(2|I||M|)$ plutôt que $\mathcal{O}(|I||M|^2)$.
- $|I|$: taille de l'image
- $|M|$: taille du masque

Utilisation de la convolution

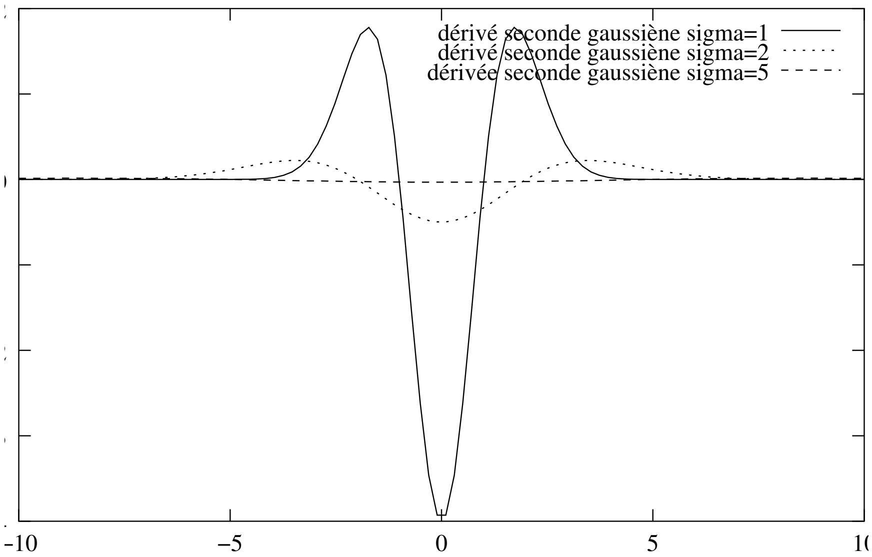
$$(f * g)^{(k)} = (f^{(k)}) * g = f * (g^{(k)})$$

Plutôt que de lisser f et calculer son gradient, on le convolu avec la dérivé d’une Gaussienne (pré-calculé). De même, le calcul du Laplacien s’effectue en convoluant f avec le Laplacien d’une Gaussienne.

Utilisation de la convolution



Dérivée de la gaussiène.



Dérivée seconde de la gaussiène.

Passage au discret

Approximation des dérivées : Différences finies (Taylor)

$$\frac{\partial f}{\partial x}(i,j) \approx \frac{\Delta I}{\Delta i}(i,j) = I(i+1,j) - I(i,j)$$

$$\frac{\partial^2 f}{\partial x^2}(i,j) \approx 2I(i,j) - I(i+1,j) - I(i-1,j)$$

Convolution avec des masques discrets :

$$M * I(i,j) = \sum_{k=-p}^{p} \sum_{l=-q}^{q} M[k][l] I[i-k][j-l]$$

Exemples de masques

$$\frac{\partial}{\partial x} = \begin{bmatrix} -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix}$$

Opérateurs de Sobel(lissage) :

$$\frac{\partial}{\partial x} = \frac{1}{4}\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \frac{\partial}{\partial y} = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

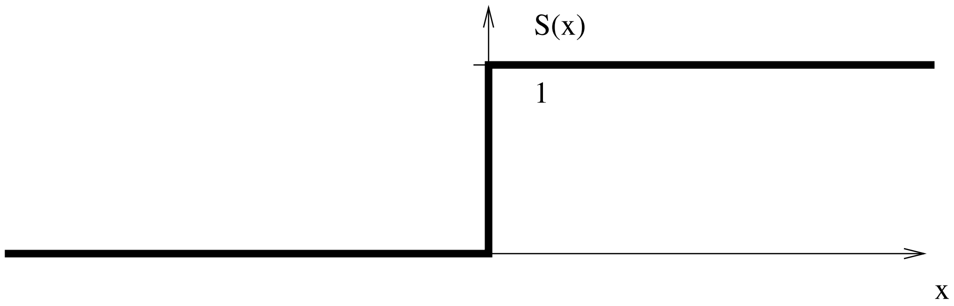
Opérateurs de Prewitt(approximation polynômial de surface) :

$$\frac{\partial}{\partial x} = \frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \frac{\partial}{\partial y} = \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Les opérateurs optimaux
Canny 86

Modélisation d'un contour

$$C(x) = S_0 S(x) + B(x)$$



L'opérateur optimal h convolé avec C doit avoir un maximum en zéro et permettre[Canny86] :

1. Une bonne détection
2. Une bonne localisation
3. Une faible multiplicité des maxima dûs au bruit.

$$\rightarrow 2h(x) - 2\lambda_1 h^{(2)}(x) + 2\lambda_2 h^{(3)}(x) + \lambda_3 = 0$$

Les opérateurs optimaux
Deriche 87

Canny : Contrainte support finie :

$$h(0) = 0 \ ; \ h(M) = 0 \ ; \ h'(0) = S \ ; \ h'(M) = 0$$

$\rightarrow$ Fonction complexe à implémenter.
Deriche 87 : Même équation, support infini :

$$h(0) = 0 \ ; \ h(+\infty) = 0 \ ; \ h'(0) = S \ ; \ h'(+\infty) = 0$$

$$\Rightarrow h(x) = ce^{-\alpha|x|} sin\omega x$$

Meilleurs résultats pour ($\omega \approx 0$)

$$h(x) = c\omega x e^{-\alpha|x|} = Cxe^{-\alpha|x|}$$

Lissage :

$$l(x) = \int_0^x h(x)dx = b(\alpha|x| + 1)e^{-\alpha|x|}$$

α contrôle le lissage (même rôle que σ)

Les opérateurs optimaux
Deriche 87(suite)

Utilisation de la transformée en $z \rightarrow$ implémentation récursive. Exemple : convolution de $x(i)$ avec $l(i)$ (résultat dans y)

$$
\begin{array}{lll}
Pour & m = & 1, \ldots, M \\
y^+(m) & = & a_0 x(m) + a_1 x(m-1) - b_1 y^+(m-1) - b_2 y^+(m-2) \\
Pour & m & = M, \ldots, 1 \\
y^-(m) & = & a_2 x(m+1) + a_3 x(m+2) - b_1 y^-(m+1) - b_2 y^-(m+2) \\
Pour & m & = 1, \ldots, M \\
y(m) & = & y^+(m) + y^-(m)
\end{array}
$$

M : taille du signal. Coefficients a_i et b_i déduis du paramètre α.
8 opérations quelque soit α.

Avantages des filtres de Deriche :

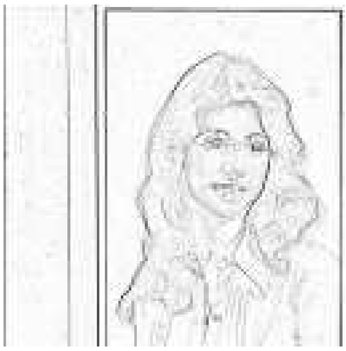
1. Formalisation de la notion de contour
2. Temps de calculs indépendants de α.
3. Pas de coupure du filtre

Les opérateurs optimaux
Deriche 87(Exemples)

Image Originale

$\alpha = 1$

$\alpha = 5$



$\alpha = 10$

Remarque : Résultats similaires avec les opérateurs de Shen-Castan 89

Gradient Couleur

$$f \left(\begin{array}{lcl} \mathbb{R}^2 & \to & \mathbb{R}^3 \\ (x,y) & \mapsto & (f_1(x,y), f_2(x,y), f_3(x,y)) \end{array} \right.$$

Différentielle de f en p :

$$Df(p).(n_x, n_y) = \begin{pmatrix} \frac{\partial f_1}{\partial x}(p).nx + \frac{\partial f_1}{\partial y}(p).n_y \\ \frac{\partial f_2}{\partial x}(p).nx + \frac{\partial f_2}{\partial y}(p).n_y \\ \frac{\partial f_3}{\partial x}(p).nx + \frac{\partial f_3}{\partial y}(p).n_y \end{pmatrix}$$

Norme de la différentielle au carré :

$$\begin{array}{rcl} S(p, \vec{n}) & = & \|Df(p).(n_x, n_y)\|^2 \\ & = & E{n_x}^2 + 2F n_x n_y + G{n_y}^2 \end{array}$$

avec :

$$E = \sum_{i=1}^{3} \left(\frac{\partial f_i}{\partial x} \right)^2 \quad F = \sum_{i=1}^{3} \frac{\partial f_i}{\partial x} \frac{\partial f_i}{\partial y} \quad G = \sum_{i=1}^{3} \left(\frac{\partial f_i}{\partial y} \right)^2$$

Gradient Couleur (suite)

$$S(p,\vec{n}) = E{n_x}^2 + 2Fn_xn_y + G{n_y}^2$$

Si $\vec{n} = (cos(\theta), \sin(\theta))$, maximum de $S(p,\vec{n})$ pour :

$$\theta_0 = \frac{1}{2}arctan\left(\frac{2F}{E-G}\right)$$

Maximum correspondant $\lambda(x,y) =_{not} S(p,(cos(\theta_0), sin(\theta_0)))$:

$$\lambda(x,y) = \frac{E + G + \sqrt{(E-G)^2 + 4F^2}}{2}$$

λ égal au carré de la norme du gradient dans le cas mono-dimensionnel.

Laplacien Couleur

Différentielle de $S(p, \vec{n})$:

$$\begin{aligned} D_S(p).\vec{n} = \; & E_x(p){n_x}^3 + (2F_x(p) + E_y(p))n_x^2 n_y \\ & +(G_x(p) + 2F_y(p))n_x n_y^2 + G_y(p) n_y^3 \end{aligned}$$

E_x, E_y,F_x, F_y et G_x, G_y : dérivées partielles de E, F et G par rapport à x et y.

$D_S(p).\vec{n} \approx$ Laplacien

Références :

Zenzo 86, Cumani 89, 91

Calcul des contours :

Utilisation de la norme du gradient :

1. Calculer le gradient en chaque point
2. Créer l'image de la norme du gradient
3. Extraire les maxima locaux dans la direction du gradient.
4. Effectuer un seuillage à effet hystérésis de l'image des maxima locaux.

Seuillage à effet hystérésis : Ne conserver que :

1. Les points dont la norme du gradient est supérieure à un seuil haut (sh)
2. Les points dont la norme du gradient est supérieure à un seuil bas (sb avec $sb < sh$) et appartenant à un bout de contour dont au moins un point possède une norme du gradient supérieure à sh.

Calcul des contours :

Utilisation de la différentielle seconde :

- Calculer $D_2 f(p).\vec{n}$ pour tout point p de l'image ($\vec{n} = \frac{\nabla f}{\|\nabla f\|}$)
- Rechercher les passages par zéro de $D_2 f(p).\vec{n}$ dans la direction $\vec{n}$.
- Créer l'image des passages par zéro et de la norme du gradient.
- Effectuer un seuillage à effet hystérésis de l'image des maxima locaux.

Calcul des contours :

Utilisation du Laplacien :
- Calculer le Laplacien.
- Rechercher les passages par zéro.
- Créer l’image des passages par zéro et de la norme du gradient.
- Effectuer un seuillage à effet hystérésis de l’image des maxima locaux.

Les contours actifs

Minimisation de l'énergie d'une courbe C : $v(s), s \in [0,1]$.

$$E(C) = E_{interne}(C) + E_{image}(C)$$

- $E_{interne}(C)$: Contraintes internes à la courbe. Permet de régulariser la solution.
- $E_{image}(C)$: Energie de l'image. Généralement liée au gradient.

$$E(C) = \int_0^1 F_{interne}(s) + F_{image}(s) ds$$

$F_{interne,image}$: Forces appiquées en chaque point de la courbe.

Les contours actifs
(L'Énergie interne)

Généralement 2 types de forces :

- Forces de tensions : $\frac{\partial v}{\partial s}$
- Forces de courbures : $\frac{\partial^2 v}{\partial s^2}$

$$\frac{\partial v}{\partial s} \quad \frac{\partial^2 v}{\partial s^2}$$

$$E_{interne}(C) = \int_0^1 \alpha(s)\|\frac{\partial v}{\partial s}\|^2 + \beta(s)\|\frac{\partial^2 v}{\partial s^2}\|^2 ds$$

$\alpha(s), \beta(s)$: Coefficients de pondération

Les contours actifs
(L'Énergie de l'image)

Isssue d'une force dans la direction du gradient $\bigtriangledown I$ et proportionelle à celui ci.

$$E_{image} = -\int_0^1 \|\bigtriangledown I(v(s))\|^2 ds$$

$\Rightarrow$ Énergie totale :

$$E(C) = \int_0^1 \alpha(s)\|\frac{\partial v}{\partial s}\|^2 + \beta(s)\|\frac{\partial^2 v}{\partial s^2}\|^2 - \|\bigtriangledown I(v(s))\|^2 ds$$

Les contours actifs (Résolution)

Trois type de méthodes :

– Algorithmes gloutons : Chaque point est bougé en fonction de la somme des forces.
– Programmation dynamique : Le déplacement d'un point n'influence que ces deux voisins.
– Résolution de système : On tient compte de toutes les influences

–

– Problèmes dans les zones sans informations de gradient.
– Les choix des coefficients de pondération influencent de manière importante l'évolution du contour : Difficile à fixer a priori.

Méthodes principalement utilisées pour le suivi d'objets.

Conclusion

- Méthodes extrêmement rapides
- Pré-traitrement pour (par ex.) des algorithmes de fermeture de contour.
- Difficulté du choix d'un seuil pertinent.
  - Trop de lissage $\rightarrow$ perte de détails
  - Pas assez $\rightarrow$ détection de bruit.
- Critère extrêmement local (pas de concept de régions)

L’approche Région

L'approche région

Définition d'Horowitz

1. $X = \bigsqcup_{i=1}^{m} R_i$

2. $\forall i \in \{1, \ldots, n\}$ R_i est connexe.

3. $\forall i \in \{1, \ldots, n\}$ $P(R_i) = vrai$

4. $\forall i, j \in \{1, \ldots, n\}^2, i \neq j$ / R_i est adjacent à R_j $\Rightarrow P(R_i \cup R_j) = faux$

Subdivision en trois grandes familles

1. Les méthodes descendantes(Utilisation de (3))
2. Les méthodes ascendantes (Utilisation de (4))
   - Méthodes d'agrégation
   - Méthodes de fusion
3. Les méthodes Mixtes (Utilisation de (3) et (4))

Notion de multi-ensemble

I une image avec n composantes par pixels
$\rightarrow$ Multi-ensemble (C, f)

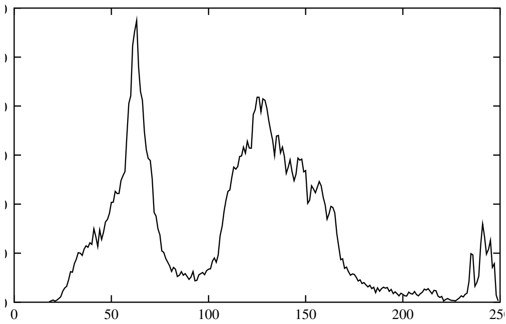
- $C \subset \mathbb{R}^n$ ensemble des valeurs prises par les pixels
- $f(c)$ nombre de pixels de l'image de valeur c.

Vision probabiliste :

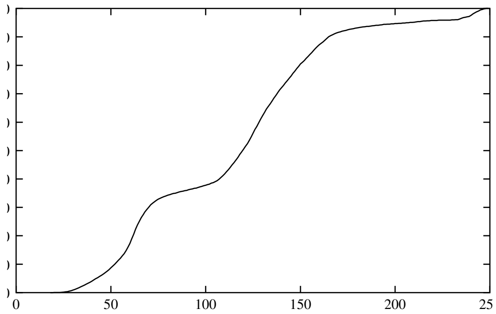
- $p(c) = \frac{1}{|I|} f(c)$: probabilité qu'un pixel ait la valeur c
- Si $C' \subset C$, $\frac{1}{|I|} \sum_{c \in C'} f(c)$ probabilité que la valeur d'un pixel appartienne à C'.

Notion de multi-ensemble
Cas du niveau de gris

Si I en niveau de gris $C \subset [0, 255]$.



Histogramme

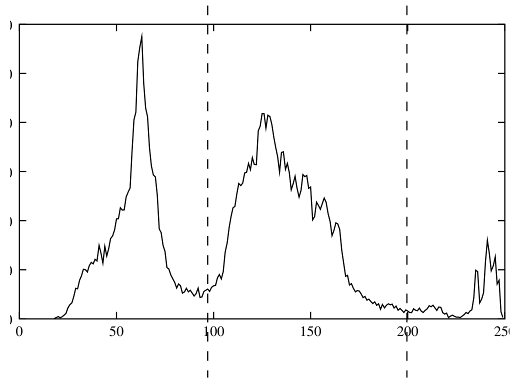


Histogramme cumulé.

L'approche région
Les méthodes descendantes

Les méthodes de division

Découpe de l'histogramme $[m, M]$ par un ensemble de seuils $T_0, \ldots, T_n$.
Exemple :



Fille Définition des seuils.

Deux seuils $T_0 \approx 100, T_1 \approx 200$:

Les méthodes de division
Classification des méthodes de seuillage :

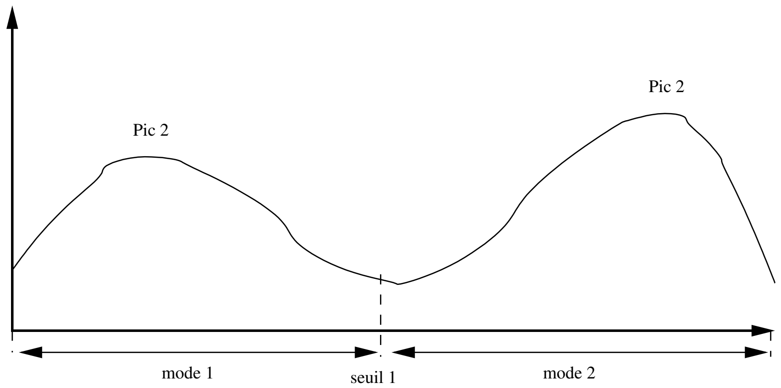
- Seuillages globaux basés uniquement sur l'histogramme.
- Seuillages locaux basés
  - sur l'histogramme,
  - sur un voisinage de chaque point de l'image.
- Seuillages dynamiques, tiennent simultanément compte
  - De l'histogramme
  - Du voisinage des points
  - De la position d'un point dans l'image.

Deux Familles

1. Décomposition en pics et vallées
2. Homogénéité des multi-ensembles.

Décomposition en pics et vallées : Somme d'histogrammes à distribution Gaussienne.

$$h(x) = \sum_{i=1}^{n} \frac{p_i}{\sigma_i} e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}} \; avec \, \mu_1 < \mu_2 \cdots < \mu_n$$



Décomposition en pics et vallées

Définitions d'un seuil :

$$\frac{\partial h}{\partial x}(s) = 0 \text{ et } \frac{\partial^2 h}{\partial x^2}(s) > 0$$

Bhattacharya-67 : limite supérieure de décroissance de $ln(h(n)) - ln(h(n-1))$ $n \in [m, M]$.

Bartel 79 : Approximation des modes par des fonctions Gaussiennes(amélioration itérative).

Papamarkos 94 :

1. Détermination des pics
2. Approximation des vallées par des fonctions rationnelles

Witkin 84 : Lissage avec un filtre passe-bas $\rightarrow$ seuils fonction du paramètre de lissage (cf [Lim 90] et [Celenk 90]).

Décomposition segments homogènes

Erreur quadratique :

$$\begin{aligned} EQ(C) &= \sum_{x \in C} \|x-\mu\|^2 = |C| \sum_{i=1}^{n} var_i \\ EQ([T_i, T_{i+1}]) &= \sum_{x \in [T_i, T_{i+1}]} (x-\mu)^2 \\ &= |T_{i+1} - T_i| var([T_i, T_{i+1}]) \end{aligned}$$

Erreur de Partition :

$$E(N) = \sum_{i=1}^{N} EQ(C_i)$$

Minimisation de $E(N) \rightarrow$ Régions de valeur constantes
Qualité de la partition :

$$\beta(N) = E(N)\Delta(N)\; avec\; \Delta(N) = \sum_{i=1}^{N} \|\mu_i - \mu_0\|^2$$

$\Delta(1) = 0$, $\boldsymbol{E}(M) = 0 \rightarrow$ existence d'au moins un maximum : $\exists N \;/\; E(N) = \Delta(N)$[Ca79].

Seuillages locaux
Classification bayésienne itérative
[Mardia 88]

Effectuer une partition initiale de l'histogramme en N classes.

1. Calculer μ_i et $|C_i|$ pour chaque classe.
2. Extraire les composantes connexes de chaque classe.
3. Déterminer le filtre γ_i minimisant $var(C_i)$ (possibilité de prendre la moyenne).
4. Calculer les seuils
$$s_{ij} = \frac{\mu_i + \mu_j}{2} + \frac{\sigma^2}{\mu_i - \mu_j} ln \frac{|C_j|}{|C_i|}$$
5. Lisser l'image avec γ et réaffecter les pixels à chaque classe.
6. Retour en 1 tant que non stabilité.

Seuillages locaux
Méthode de Relaxation gradient
[Bhanu 82,87]

Adaptée aux images uni-modale.

$$\forall p \begin{cases} I(p) \in C_1 \Rightarrow |I(V_8(p)) \cap C_1| > |I(V_8(p)) \cap C_2| \\ I(p) \in C_2 \Rightarrow |I(V_8(p)) \cap C_2| > |I(V_8(p)) \cap C_1| \end{cases}$$

Vecteur de probabilité d'appartenance :

$$P(s) = \begin{pmatrix} P_{C_1}(s) \\ P_{C_2}(s) \end{pmatrix} \text{ avec } P_{C_1}(s) + P_{C_2}(s) = 1$$

Vecteur de compatibilité :

$$Q(s) = \begin{pmatrix} Q_{C_1}(s) \\ Q_{C_2}(s) \end{pmatrix} \text{ avec } Q_{C_i}(s) = \frac{1}{8} \sum_{t \in V_8(s)} P_{C_i}(t)$$

Maximiser :

$$J = \sum_{i=1}^{M} {}^tP(s_i)Q(s_i)$$

Seuillages locaux/dynamiques
Méthode de Watanabe 74

Seuils $\rightarrow$ frontières entre les régions $\rightarrow$ fort gradient.

$$d(z) = \sum_{s \ / \ I(s)=z} \|\bigtriangledown I(s)\|$$

Seuil $\leftrightarrow$ $d(z)$ max.

Seuillages dynamiques
Méthode de Hertz 88

Calcul de deux images binaires :

- IG : Image de gradient seuillé
- IS :
  - Décomposition de l'image en blocs
  - Calcul de seuils sur chaque bloc
  - Calcul des composantes connexes associées.
  - Marquage à 1 des pixels frontières

Calcul de $ID = IG - IS$ et modification des seuils :

- Si $Nb(-1)$ grand et $Nb(+1)$ grand :bon nombre de seuils mais mal placés.
- Si $Nb(-1) << Nb(+1)$: faux seuil. Eliminer le seuil le moins significatif.
- Si $Nb(+1) >> Nb(-1)$ seuil absent. Introduire un nouveau seuil.

Seuillages dynamiques
Binarisation, Méthode de Nakagawa 79

1. Découper l'image en bloc
2. Faire l'hypothèse bi-modale dans chaque bloc $\rightarrow$ seuil s solution de :
   $(\frac{1}{\sigma_1^2}-\frac{1}{\sigma_2^2})s^2+2(\frac{\mu_2}{\sigma_1^2}-\frac{\mu_1}{\sigma_2^2})s-\frac{\mu_2}{\sigma_2^2}+\frac{\mu_1}{\sigma_1^2}+2ln(\frac{p_2\sigma_1}{p1\sigma_2}) = 0$
   Si non bimodal :
   seuil$=$ Moyenne pondérée des seuils des blocs voisins.
3. Construire l'image des seuils :
   $S[p]$: Moyenne pondérée des seuils des blocs.
4. Construire l'image seuillée IB :

$$IB[i,j] = \begin{cases} 0 \text{ si } I[p] > S[p] \\ 1 \text{ sinon} \end{cases}$$

Possibilité d’extension à trois Gaussiennes (Choux, Kaneko 72)

Seuillages globaux
Extension à la couleur

Notion de projection d'un multi-ensemble :

La **projection** de $(C, f) \in \mathcal{ME}_n$ sur l'axe vectoriel défini par le vecteur $\vec{n} \in \mathbb{R}^n$ est égale à $([m, M], F) \in \mathcal{ME}_1$ avec :

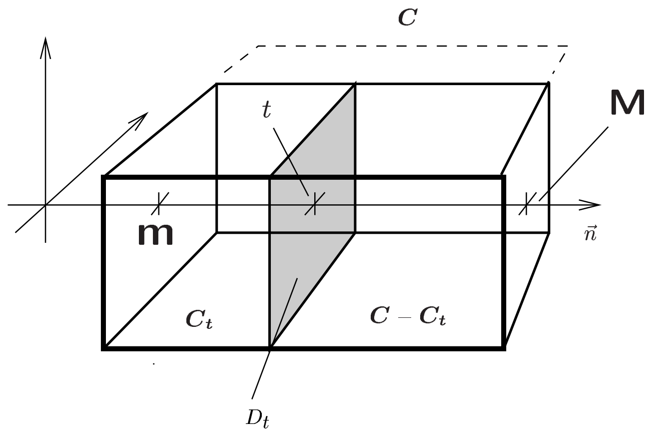
$$\begin{aligned} m &= \min_{v \in C} v.\vec{n} \ , \ M = \max_{v \in C} v.\vec{n} \\ D_t &= \{v \in C / v.\vec{n} = t\} \\ F(t) &= \sum_{v \in D_t} f(v) \end{aligned}$$

Si T_0 est un seuil de $([m, M], F)$:
$(C, f) = (C_t, f) \cup (C - C_t, f)$ avec :

$$C_t = \{v \in C \ / \ v.\vec{n} \leq t\}$$

Seuillages globaux
Extension à la couleur

Notion de projection d'un multi-ensemble : Illustration

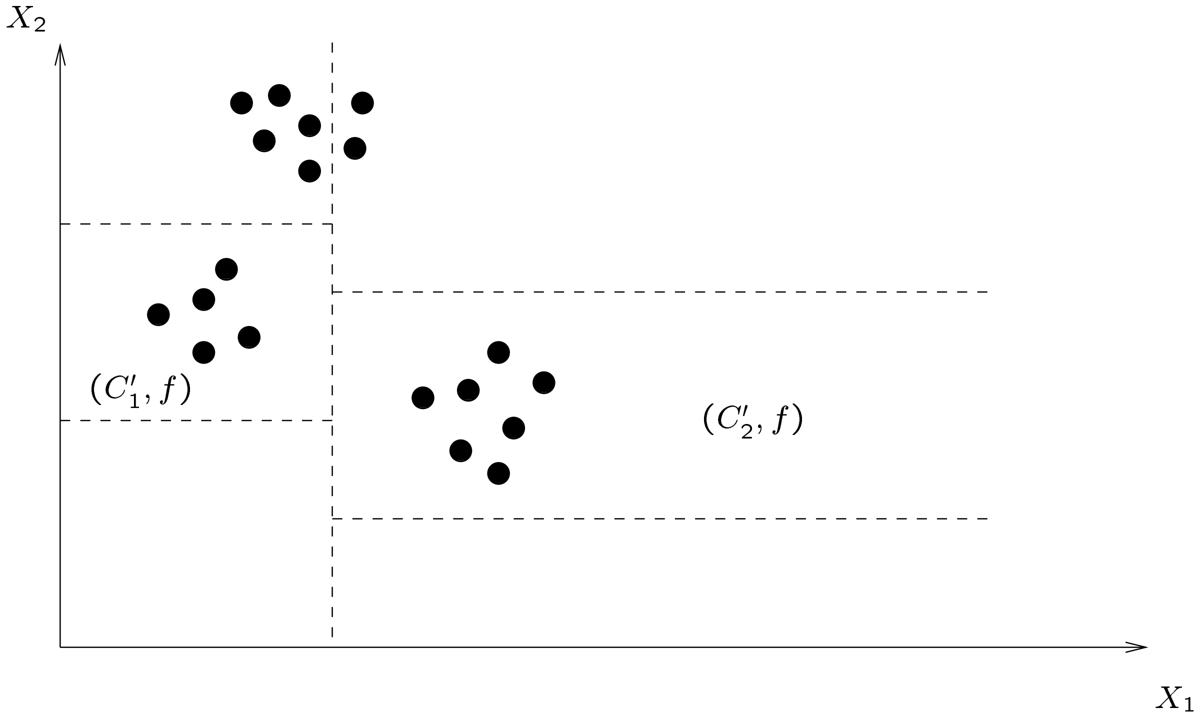


Seuillages globaux
Extension à la couleur

Méthodes utilisant les projections 1D :
- Projection sur l'axe principal(de plus forte variance).
  Exemple : Ohlander 75 (+approximation binomiale)
- Projection sur les axes de coordonnées.
  Ohlander 78 :
    Détermination de seuils sur chaque axe puis découpe suivant l'axe possédant les meilleurs seuils.

Celenk 90 :

1. Découper (C, f) sur X_1
   $\rightarrow (C_1, f)$ et (C_2, f)
2. Calculer les projections de (C_1, f) sur X_2 et X_3
3. Découper récursivement (C_1, f) sur X_2 et X_3 en ne retenant que les "meilleurs" modes$\rightarrow (C_1', f)$.
4. Faire de même avec (C_2, f).
5. $(C, f) = (C, f) - (C_1', f) \cup (C_2', f)$. Continuer tant que modes significatifs



Seuillages globaux
Extension à la couleur

Découpe 3D :

Nuées dynamiques

1. Choisir au hasard K éléments $(v_1, \dots, v_K)$
2. Affecter chaque élément de (C, f) au v_i le plus proche $\rightarrow ((C_1, f), \dots, (C_K, f))$.
3. Calculer la moyenne $\mu(C_i)$ de chaque (C_i, f).
4. Définir $(w_1, \dots, w_K)$ tel que w_i est l'élément de (C, f) le plus proche de $\mu(C_i)$.
5. Si $\exists v_i \;/\; v_i \neq w_i$, itérer avec $(w_1, \dots, w_K)$.

Utilisation de seuil :

$$\max_{i \in \{1,\dots,K\}} \|v_i - w_i\| < \epsilon$$

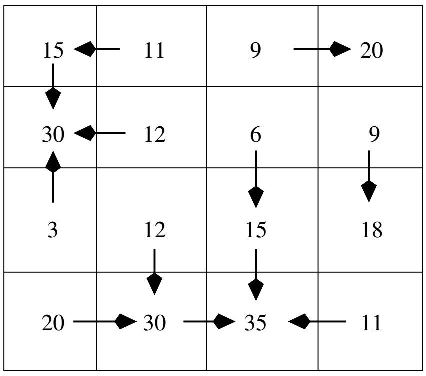
Convergence vers le minimum local le plus proche de la solution initiale

Quantification 3D :

- Méthodes Top-down [Heckbert, Wu. . .] Découpe récursive du multi-ensemble initial. Possibilité de définir des critères d'arrêt tels que β.
- Méthodes Bottom-up : Xiang. . .
  1. Initialisation de K multi-ensembles "vides"
  2. Affectation de chaque couleur à son multi-ensemble le plus proche.

Sommets - Bassins :

- Mathas 90 : Trouver les points de fréquence maximale.



- Bonnet-Herbin : Estimation de la fonction de densité de probabilité suivie d'une estimation des modes par bassins versants inversés.

Les méthodes de division

Récapitulatif :

– Prise en compte de la globalité de l'image $\neq$ méthodes de contours.
– Image$\rightarrow$ multi-ensemble : perte d'information
  $\Rightarrow$ Très difficile de distinguer les petites régions ou de segmenter les régions comportant de grandes variations.

Recherche des détails $\rightarrow$ Méthodes de division récursive

Les méthodes de division récursive

1. Calculer le multi-ensemble de l'image.
2. Partitionner le multi-ensemble.
3. Calculer la partition de l'image induite par la partition du multi-ensemble.
4. Pour chaque région ainsi définie calculer son multi-ensemble.
5. Si un multi-ensemble généré en 4 est partitionnable, revenir en 2.

Principe de la focalisation d'attention.
Problème : Irréversibilité des découpes.
Références : Ohlander 78, Coleman 79, lee 90, Celenk 90

Les méthodes agrégation

Deux étapes :

1. Sélection d'un ensemble de germes(utilisation du gradient, de la variance...)
2. Croissance des germes : Affecter itérativement chaque pixel de l'image à un germe.

Utilisation d'une fonction Oracle(p,g).

A l'étape k définition de l'oracle à partir des :
- attributs de p,
- attributs de g,
- attributs de g à l'étape $k-1$.

Exemple des Watersheeds
(Ligne de partage des eaux)

1. Obtenir une image des gradients.
2. Germes : pixels de plus bas gradient. Marquer ces pixels.
3. A l'étape i :
   - Sélectionner les pixels non marqués de plus bas gradient.
   - Marquer les pixels sélectionnés.
   - Agréger à un germe tout pixel sélectionné et adjacent à celui-ci.
   - Initialiser de nouveaux germes à partir des pixels sélectionnés restant.
4. Répéter le point 3 tant qu'il reste des pixels non marqués.

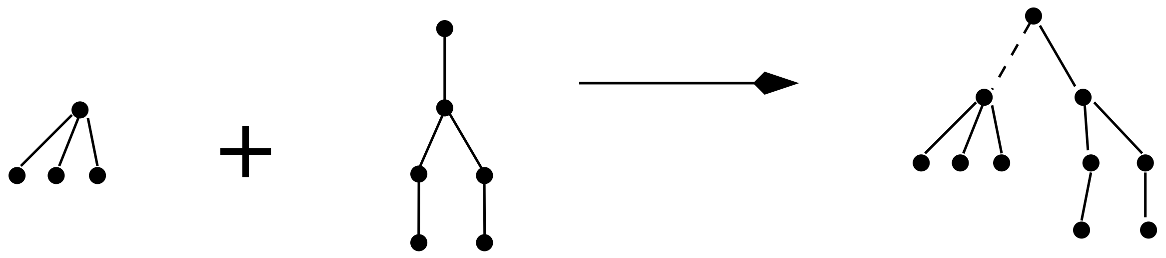
Les méthodes d'Agrégation/Fusion

Méthodes d'agrégation : Régions initiales d'un pixel.

Méthodes de fusion : Taille des régions initiales supérieure à un pixel.

Exemple des méthodes d'Unir-Trouver :

Codage des régions par des arbres $\rightarrow$ Choix d'un élément canonique.
Union de régions : Union d'arbres

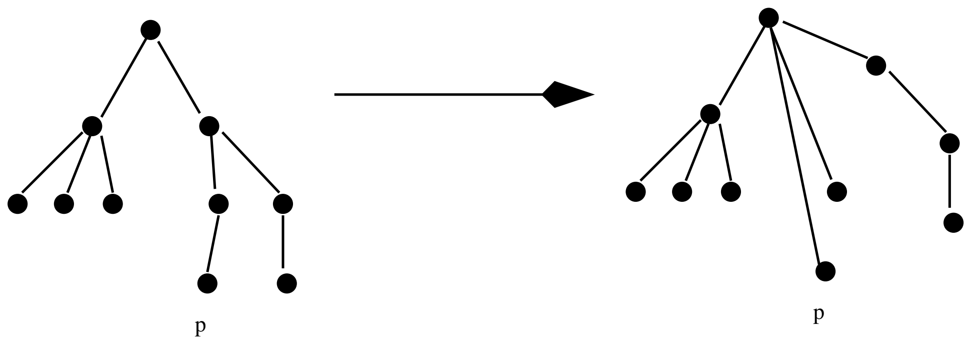


Les méthodes Unir-Trouver

Retrouver le pixel canonique d'une région :

```
TrouveCompresse(p)
{
    si ( racine(p) )
        return p ;
    sinon
    {
        p.parent = TrouveCompress(p.parent)
        return p.parent
    }
}
```

Algorithme 1: *Compression de chemins*



Les méthodes Unir-Trouver

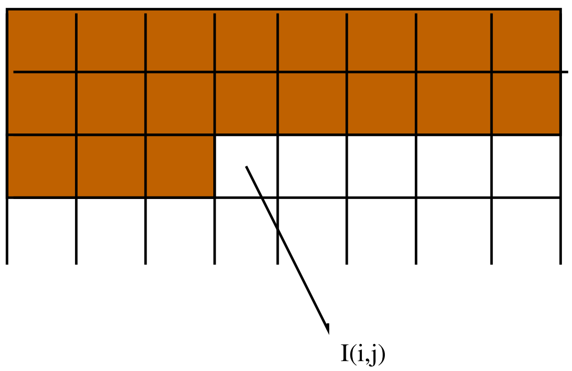
```
Segmente(Image I)
{
    Initialiser chaque pixel comme la racine
    d’un arbre (un germe)
    Adapter le traitement pour la premiere ligne.
    Pour i = 2 a hauteur(I)
    {
        traitement special pour la premiere colonne
        Pour j=2 a largeur(I)
        {
            gauche = TrouveCompresse(I[i][j-1])
            haut = TrouveCompresse(I[i-1][j])
            courant =TrouveCompresse(I[i][j])
            si (Oracle(gauche,courant))
                Uni(gauche,courant)
            si (Oracle(haut,courant))
                Uni(haut,courant)
        }
        Aplatit (ligne i)
    }
}
```

Algorithme 2: *Création de la partition*

Les méthodes Unir-Trouver

Deux type d'opérations :

1. Union pixel-region
2. Union région-région



Les méthodes de Fusion

1. Créer une partition initiale
2. Fusionner tout couple de régions adjacentes dont l'union reste homogène (prédicat 4 d'Horowitz) :

$$\forall i, j \in \{1, \ldots, n\}^2, i \neq j \ / \ R_i \text{ est adjacent à } R_j \\ \Rightarrow P(R_i \cup R_j) = faux$$

$\rightarrow$ Prédicat Fusionne(R_i,R_j).

L'algorithme Phagocyte de Brice et Fennema (1970)

$|S|$: longueur du segment S,

$|S|_w$: nombre de points de S dont le potentiel est inférieur à σ.

δR : Frontière de R

```
phagocyte()
{
    Pour toutes regions R_i, R_j
    {
        S_ij = δR_i ∩ δR_j
        Si ( |S_ij|_w / min(|δR_i|,|δR_j|) < θ)
        {
            fusionner R_i et R_j en supprimant S_ij
        }
    }
}
```

Algorithme 3: *Algorithme phagocyte de Brice et Fennema.*

Le prédicat de Beveridge

Utilisation de nombreux attributs :

- Attributs d'une région R_i :
  μ_i : moyenne du multi-ensemble,
  σ_i : écart type,
  $|R_i|$: nombre de pixels,
  δR_i : frontière,
  $|\delta R_i|$: périmètre.
- Attribut entre deux régions R_i et R_j :
  $|\delta R_i \cap \delta R_i|$: longueur de la frontière commune

Trois critères de fusion :

S_{sim} : similitude photométrique,

S_{size} : taille,

S_{conn} : connectivité.

$S_x = 0$: forte propension à la fusion,
$S_x = 1$: critère neutre vis à vis de la fusion,
$S_x > 1$: fusion déconseillée.

Le prédicat de Beveridge (suite)

1. Similitude photométrique :

$$S_{sim}(R_i, R_j) = \frac{|\mu_i - \mu_j|}{\max(1, \sigma_i + \sigma_j)}$$

2. Critère de taille : T_{opt} taille optimale

$$S_{size}(R_i, R_j) = \min\left(2.0, \frac{\min(|R_i|, |R_j|)}{T_{opt}}\right)$$

3. Connectivité :

$$S_{conn}(R_i, R_j) = \begin{cases} c(R_i, R_j) & \text{si } \frac{1}{2} \leq c(R_i, R_j) \leq 2 \\ \frac{1}{2} & \text{si } c(R_i, R_j) < \frac{1}{2} \\ 2 & \text{sinon} \end{cases}$$

avec :

$$c(R_i, R_j) = \frac{\min(|\delta R_i|, |\delta R_j|)}{4|\delta R_i \cap \delta R_j|}$$

Le prédicat de Beveridge (suite et fin)

Unification des critères :

$$S(R_i, R_j) = S_{sim}(R_i, R_j)\sqrt{S_{size}(R_i, R_j)}S_{conn}(R_i, R_j)$$

Fusion à chaque étape des deux régions dont la fusion est la plus conseillée :

$$\text{Fusionne}(R_i, R_j) = \text{vrai}$$
$$\Longleftrightarrow S(R_i, R_j) = \min_{(k,l)\in\{1,\ldots,N\}^2} S(R_k, R_l)$$

N :Nombre de régions.

Les méthodes d'Agrégation/Fusion
Récapitulatif :

Distinction Agrégation/Fusion : Taille des régions initiales (méthodologies très différentes).

- Prise en compte des informations spatiales et colorimétriques
- Grande souplesse dans la définition de l'oracle.
- Très sensible à la stratégie de Fusion/Agrégation

Union find (HG-BD)

Union-find (BD-HG)

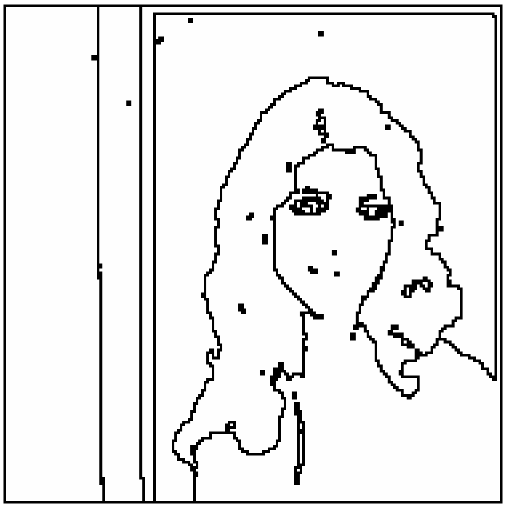
HG : en haut à gauche (0,0)

BD : en bas à droite

Les méthodes de découpe-fusion

Inconvénients des approches précédentes :
Pour les méthodes de découpe récursives :

1. Impossibilité de revenir sur une découpe



2. Balayage répété de parties de régions

Pour les méthodes de fusion : Instabilité vis-à-vis de la stratégie. Spécialement lorsque les régions sont non significatives.

Découpe-Fusion : Supprime (ou diminue) ses inconvénients.

Principe de la méthode

Introduit par Horowitz. Utilisation des points 3 et 4 de la définition. Partant d'une partition initiale :

1. Découper toute région non homogène

$$\forall i,\ P(R_i) = faux \Rightarrow decouper(R_i)$$

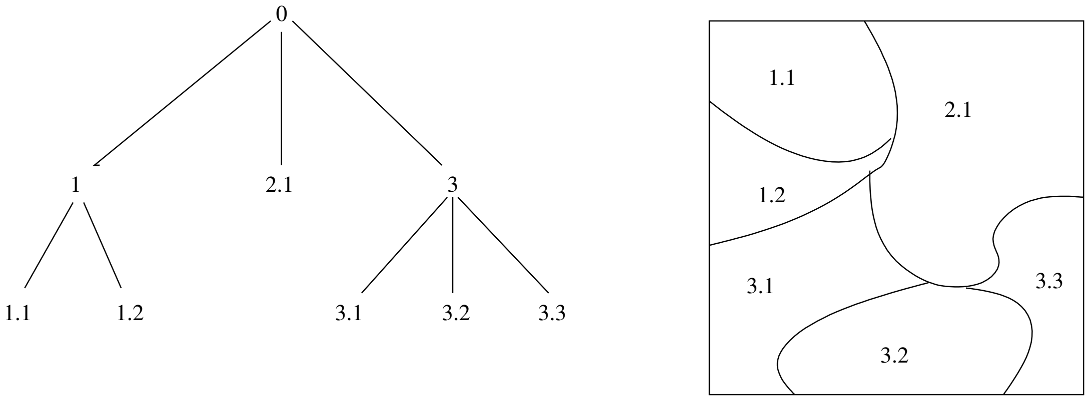
2. Fusionner tout couple de régions dont l'union est homogène.

$$\begin{array}{l}\forall (i,j)\ /\ R_i \text{ est adjacent à } R_j,\\ P(R_i \cup R_j) = vrai \Rightarrow fusionner(R_i, R_j)\end{array}$$

Utilisation d'une partition initiale quelconque.

Mise en oeuvre :
Notion d'arbre de partition

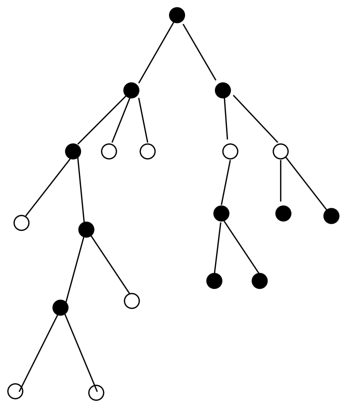
- Racine de l'arbre : L'image I.
- Successeurs d'un noeud correspondant à une région R : régions partitionnant R.
- Feuilles de l'arbre : plus petites régions pouvant être considérées (dans le pire des cas, les pixels).



Notion d'arbre de partition (suite)

Partition : Coupure de l'arbre

Tout chemin d'une feuille à la racine comporte un et un seul noeud sélectionné.



○ Noeuds appartenant à la coupure

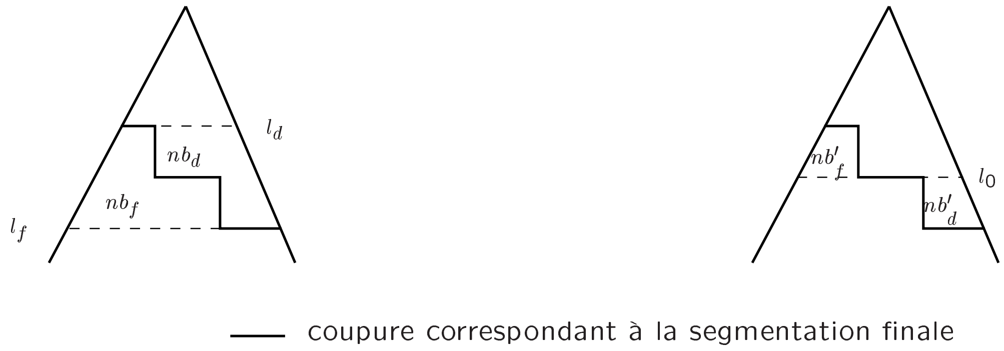
● Noeuds n'appartenant pas à la coupure

Méta-algorithme

1. Partitionner l'image en un ensemble de régions de taille fixe, créer l'arbre de partition associé.
2. $\forall i \ / \ P(R_i) = faux$
   – Découper(R_i) $\rightarrow \{R_{i_1}, \ldots, R_{i_p}\}$ fils de R_i
   – $R_i \leftarrow noeud$
3. Si $\{R_{i_1}, \ldots, R_{i_p}\}$ fils de R_i, $\forall j \ Feuille(R_{i_j}) = vrai$ et $\cup_{j=1}^{p} R_{i_j}$ homogène.
   – $Fusionner(\{R_{i_1}, \ldots, R_{i_p}\})$
   – $R_i \leftarrow feuille$.
4. Revenir au point 2 jusqu'à ce qu'aucune région ne puisse être découpée ou fusionnée.
5. Fusionner toutes les régions adjacentes R_i, R_j telles que $P(R_i \cup R_j) = vrai$ (fusions non restreintes).

Partition initiale immédiate.

Notion d'optimisation des découpes/Fusions



l_d : Niveau de départ minimum pour un algorithme de découpe

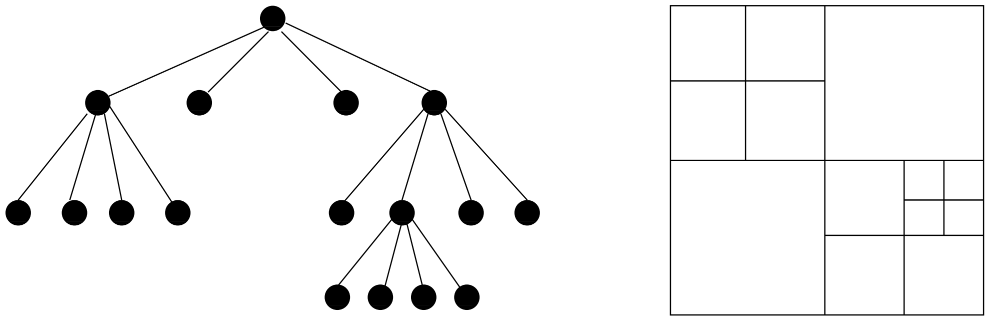
nb_d : Nombre de découpes minimum nécessaires à un algorithme de découpe

l_f : Niveau de départ maximum pour un algorithme de fusion

nb_f : Nombre de fusion minimum nécessaires à un algorithme de fusion

Une implémentation :
Le quadtree

Découpage par blocs :



1. Découper l'image en N^2 blocs carrés.
2. Découper tout bloc non homogène en quatre sous-blocs.
3. Si quatre blocs B_1,B_2,B_3,B_4 possédent le même père dans le quadtree et vérifient $P(B_1 \cup B_2 \cup B_3 \cup B_4) = vrai$, fusionner les quatre blocs en un seul.

4. Revenir au point 2 tant que l'on peut appliquer les points 2 ou 3.
5. Fusionner tous les couples de blocs adjacents (B_1, B_2) vérifiant $P(B_1 \cup B_2) = vrai$.

Conclusion :

- Plus grande souplesse que les méthodes de découpe pures ou fusion pures.
- Limité par la structure de données : Partitionnement régulier.

Une segmentation basée sur une partition par blocs.

Structures de données utilisées en segmentation

Énumération des besoins

Opérations effectuées par les algorithmes de segmentation :

1. Création d'une partition
2. Modification par Découpes/Fusions

Deux impératifs :

1. Accès rapide aux informations sur la partition
2. Modifications rapides de la partition

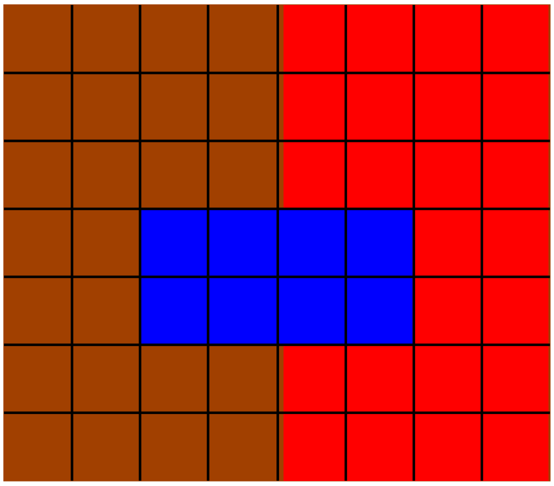
Classification des informations

- Informations géométriques :
  Ensemble des informations relatives à une région indépendamment des autres régions.
  - Ensemble des pixels d'une région
  - Région englobant un pixel
  - Frontières d'une région
  - ...
- Informations topologiques :
  Ensemble des informations relatives à la partition
  - Ensemble des voisins d'une région
  - Frontière entre deux régions
  - Composantes connexes incluses
  - Région englobante
  - ...

Tableaux de labels

Numéroter les pixels :

$$L[P_1] = L[P_2] \Leftrightarrow \exists i \in \{1, \ldots, n\} \; / \; \left\{ \begin{array}{ccc} P_1 & \in & R_i \\ & \text{et} & \\ P_2 & \in & R_i \end{array} \right.$$



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

Tableaux de labels
(suite)

Avantages :

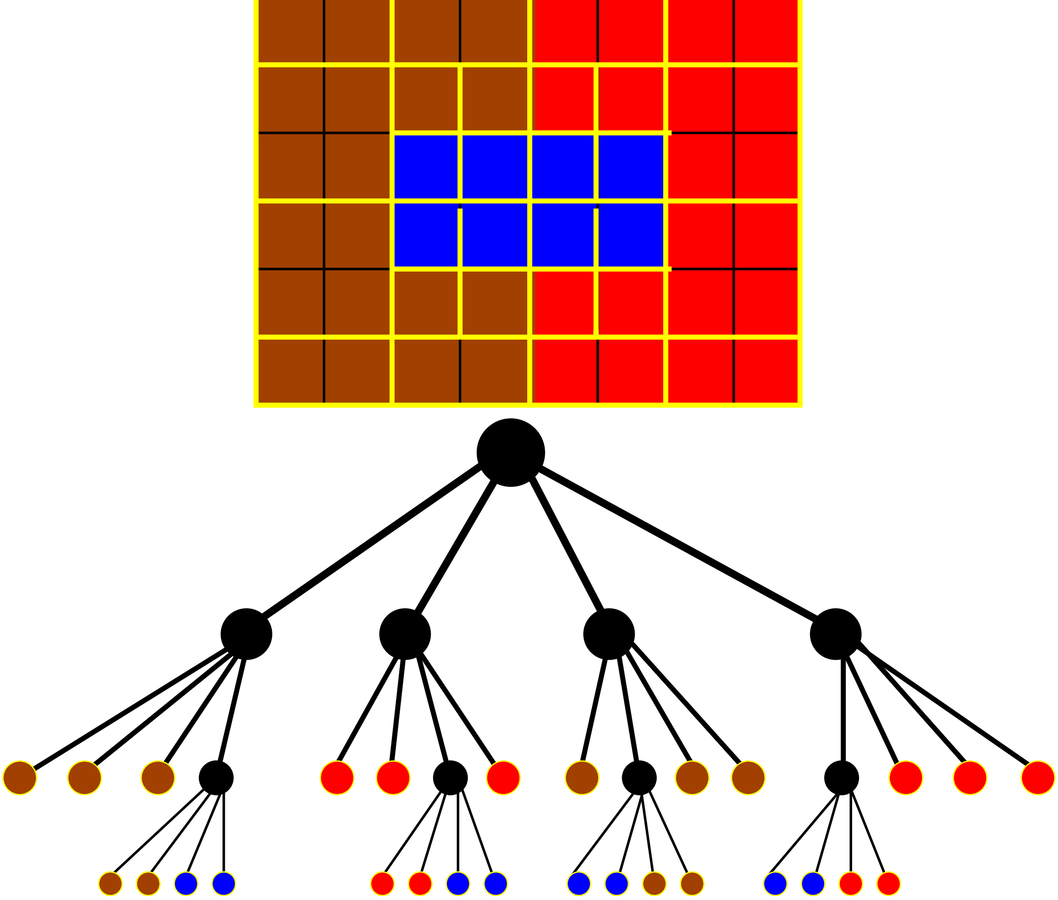
- Extrêmement simple à implémenter.
- Accès à de nombreuses informations géométriques (relation point-région directe)

Inconvénients :

- Pas compacte
- Pas d'information de frontière

Structures de données
Structures Géométriques

Quadtree



```
agregation(germes g_1,...,g_n)
{
    Tableau Regions[n] ;
    labels L ;
    Pile P ;

    Initialiser L a 0 ;
    Pour i de 1 a n
    {
        L[g_i]=i ;
        Initialiser R[i](g_i) ;
        Empiler(P,V_x(g_i)) /* x ∈ {4,8} */
    }

    Tant que P non vide
    {
        p = depiler(P) ;
        label_p=non-defini ;
        Pour tout p' ∈ V_x(p)
        {
            Si (L[p']==0)
                empiler(p') ;
            mise_a_jour(label_p,L[p']) ;
        }
        agreger(p,R[label_p]) ;
        L[p]=label_p ;
    }
}
```

Algorithme 4: *Un méta-algorithme d'agrégation*

Quadtrees (suite)

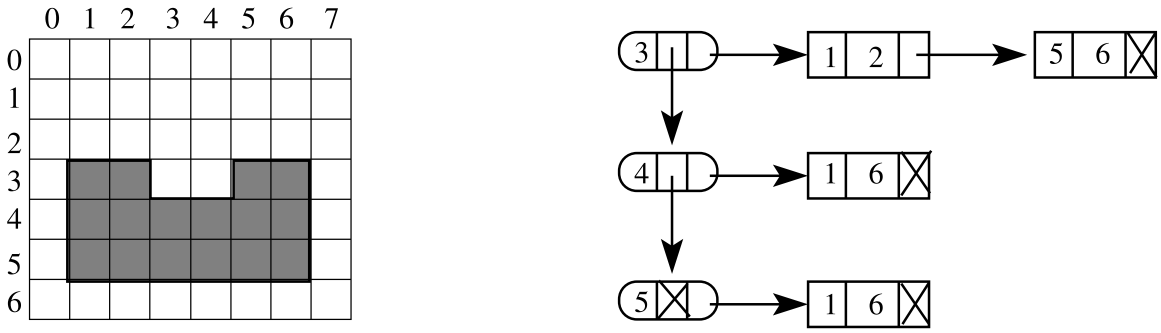
Avantages :

– Accès efficace aux informations géométriques
– Possibilité de multi-résolution

Inconvénients :

– Restreint les découpes et les fusions
– Pas d'information de frontière

Codage par plages



Avantages :

- Compression d'information
- Parcourt aisé de l'ensemble des pixels

Inconvénients :

- Pas d'information de frontière.

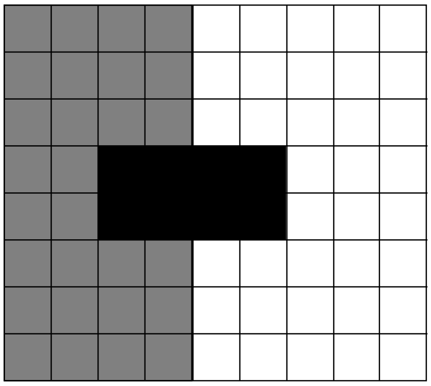
Assez peu utilisé en segmentation.

Transformation par axe médian
(median axis transform)

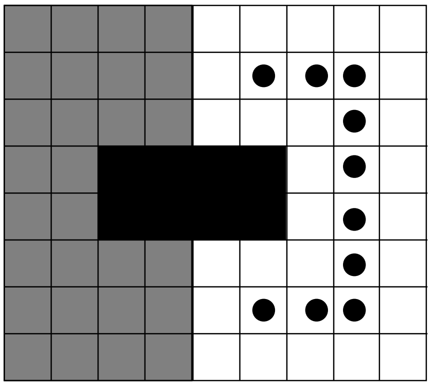
BB MAT : Block based MAT. Plus grand carré englobant

DB Math : Disk based MAT. Plus grand disque englobant.

- Calculer le plus grand bloc (resp. disque) centré sur chaque pixel et inclus dans la région.
- Rejeter les pixels dont le bloc(resp. le disque) est inclus dans un bloc (resp. disque) pré-existant.



Transformation par axe médian (suite)



Avantages :

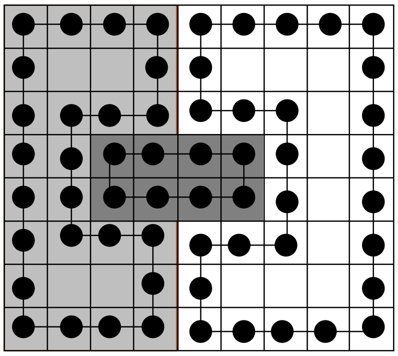
- Représentation compacte
- Axe Médian : Information sur la forme des régions $\rightarrow$ Reconnaissance de formes.

Inconvénients :

- Découpe ou fusion de région : $\rightarrow$ Reconstruction du squelette : Pas adapté à la segmentation.

Représentations par frontières

Ne coder que les points frontières :
P point Frontière $\Leftrightarrow \forall i \in \{1, \ldots, n\} V(P) \not\subset R_i$



Avantages : Accès aux informations de frontière.

Inconvénients :

- Ambiguïtés sur la localisation de la frontière (redondance d'information)
- Les segments peuvent ne pas vérifier le théorème de Jordan (séparation intérieur-extérieur)

Représentations par frontières demi-entières.
Brice et Fenema 1970

ensemble des pixels définit dans P_0 :

$$P_0 = \{(i, j) \in \mathbb{Z} \times \mathbb{Z}\}$$

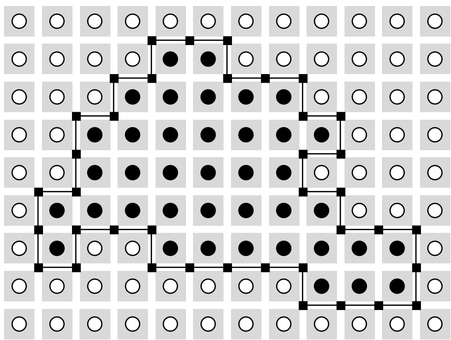
$\rightarrow$ ensemble des frontières définies dans :

$$P_{\frac{1}{2}} = \left\{(i + \tfrac{1}{2}, j + \tfrac{1}{2}),\ \text{avec } (i, j) \in \mathbb{Z}^2\right\}$$

$p \in P_{\frac{1}{2}}$ demis voisins de $P' \in P_0$ ssi :

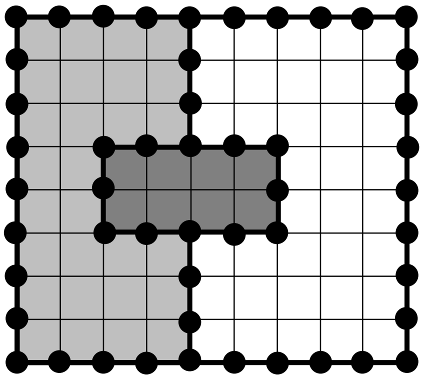
$$|x_p - x'_p| = |y_p - y'_p| = \frac{1}{2}$$

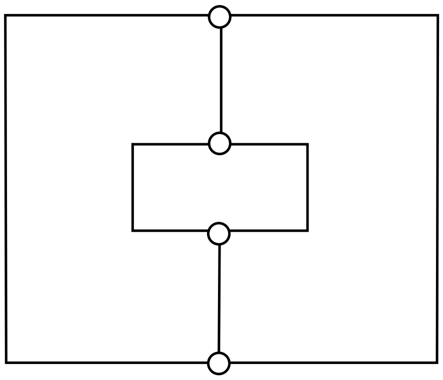
p point frontière si deux de ces demis voisins appartiennent à des régions différentes.



Représentations par frontières demi-entières.
(suite)

- Points inter-pixels : Coins des pixels
- Liens entre les points : Cotés des pixels





Structuration en segments maximaux et noeuds

segment maximal : Chemin maximal entre deux régions.

Noeud : Point frontière d'arité au moins 3.

Représentations par frontières demi-entières. (suite et fin)

Avantages :

- Informations de frontières
- Codage compact
- Pas d'ambiguité dans la localisation des frontières.

Inconvénients : Coût prohibitif des informations sur la partition

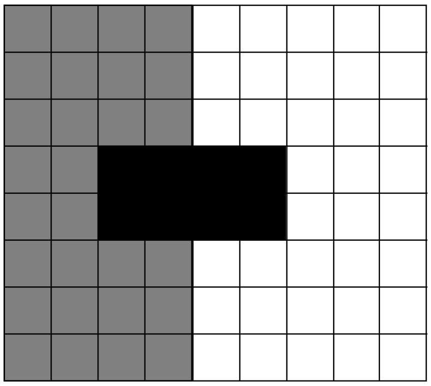
- Informations d'inclusion $\rightarrow$ Parcourir l'ensemble des pixels de la région.
- Information de voisinage $\rightarrow$ Parcourir tous les points frontières d'une région

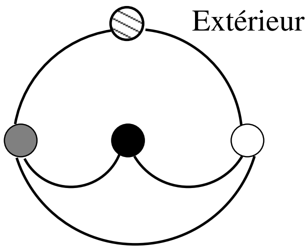
Structures Topologiques : Le graphe d'adjacence de régions

Graphe $G = (V, E)$:

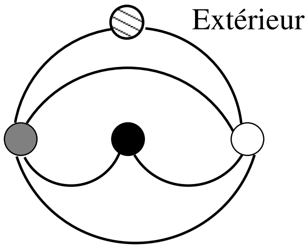
V : Un noeud par région

E : Une arête par relation d'adjacence entre régions.





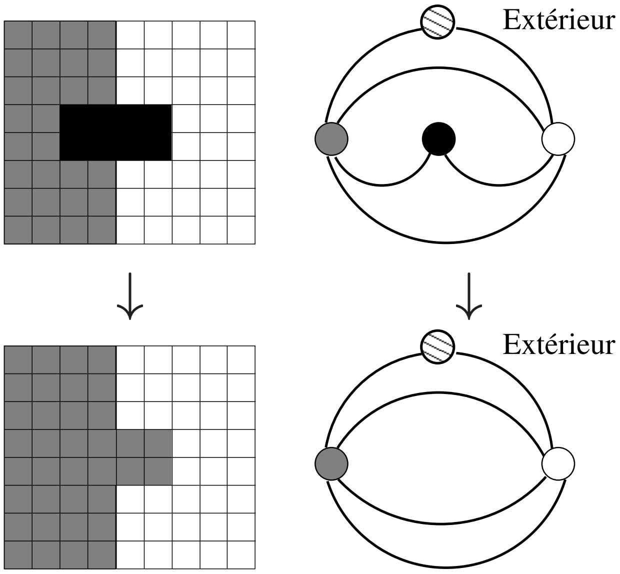
Amélioration :



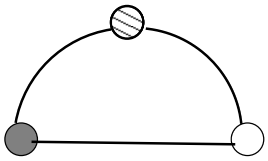
Le graphe d'adjacence de régions (suite)

Fusion de régions : Contraction d'arête

1. Identification des deux noeuds
2. Suppression de l'arête contractée



Simplification :

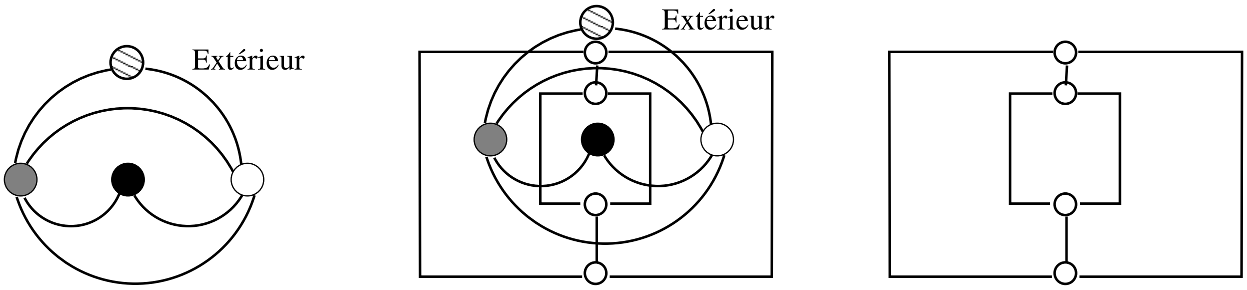


Les graphes duaux

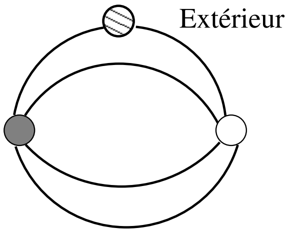
$G = (V, E) \rightarrow \overline{G} = (F, \overline{E})$

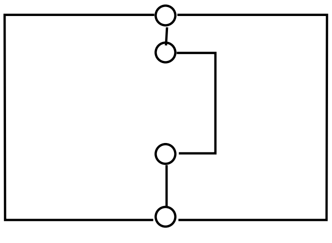
F : Ensemble des faces du graphe initial

$\overline{E}$ **:** Adjacence entre les faces (bijection entre E et $\overline{E}$.



Caractérisations des simplifications : Locales dans le graphe dual.





Graphes/Graphes améliorés/Graphes duaux

Graphes usuels : Il existe “une” relation entre deux régions (adjacence ?, inclusions ?)...

Graphes améliorés : Une arête par relation d’adjacence (bijection entre les arêtes du graphe et les frontières de la partition)

Graphes duaux : Caractérisation des propriétés des faces.

Structures de Graphes :

Avantages : De nombreuses propriétés topologiques

Inconvénients : Pas adaptés aux découpes.

- **Graphes duaux :** Maintenance de deux structures de données

Cartes Combinatoires
(Rappels mathématiques)

Soit un ensemble D.
Permutation : Application bijective de D dans D. Soit π une permutation.

- π-orbite de $b \in D$:

$$\pi^*(b) = \{b, \pi(b), \pi^2(b), \ldots,\}$$

- Décomposition en cycles :

$$\pi = \pi^*(b_1)\pi^*(b_2)\ldots\pi^*(b_n)\ldots$$

  Cycle : Restriction de π à une orbite.

Exemple de permutation

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 3 & 10 & 6 & 4 & 2 & 1 & 7 & 5 & 8 & 9 \end{pmatrix}$$

Orbites :

$$\begin{array}{rcl} \pi^*(1) & = & \{1,3,6\} \\ \pi^*(2) & = & \{2,10,9,8,5\} \\ \pi^*(4) & = & \{4\} \\ \pi^*(7) & = & \{7\} \end{array}$$

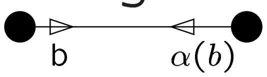
$$\pi = (1,3,6)(2,10,9,8,5)(4)(7)$$

Définition

$G = (V, E) \to G = (\boldsymbol{\mathcal{B}}, \sigma, \alpha)$

Décomposer chaque arête en 2 brins(demi-arêtes) :

– Élément de base : brin.
  $\boldsymbol{\mathcal{B}}$: Ensemble des brins
– Codage des arêtes :

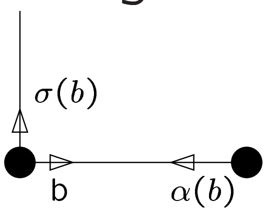


$\alpha(b)$: 2^{eme} brin de l'arête contenant le brin b.

  Peut être codée implicitement en dimension 2 :

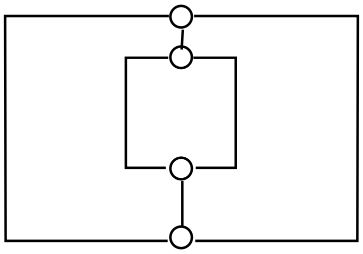
$$\forall b \in \boldsymbol{\mathcal{B}}\ \alpha(b) = -b$$

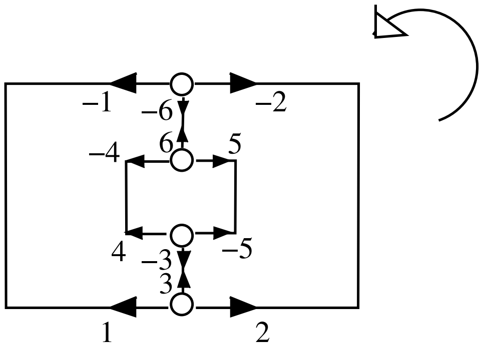
– Codage des sommets :



$\sigma(b)$: prochain brin rencontré en tournant dans le sens positif autour du sommet qui contient b.

Exemple





$$G = (V, E) \quad \rightarrow \quad G = (\mathcal{B}, \sigma, \alpha)$$

- $\mathcal{B} = \{-6, \ldots, -1, 1, \ldots, 6\}$
- Permutation α
  $\forall b \in \{-6, \ldots, -1, 1, \ldots, 6\}\ \alpha(b) = -b$
  $\Rightarrow \alpha = (1, -1)(2, -2)(3, -3)(4, -4)(5, -5)(6, -6)$
- Permutation σ

  $\sigma = (1, 2, 3)(-3, -5, 4)(5, 6, -4)(-6, -2, -1)$

Récapitulatif

Élement de base : Le brin ou demi-arête

Codage des arêtes : Permutation α

Codage des sommets : Permutation σ

Chaque brin appartient à une seule arête et un seul sommet.

Passage au dual : Permutation $\varphi = \sigma \circ \alpha$

Passage au dual

$$\varphi = (1, -6, -4, -3)(-2, 3, -5, 6)(5, 4)(2, -1)$$

Propriétés :

1. Chaque brin appartient à une seule face.
2. Tous les brins “voient” leur face associée à leur droite (bonhomme d’Ampère).
3. Toutes les faces sauf une sont parcourues dans le sens horaire.
4. La face parcourue dans le sens positif est appelée la face infinie (f^∞) de la carte $\rightarrow$ composante connexe.

Fonctions d'étiquetage

But : Fournir un identifiant de face.

λ étiquette de face $\Leftrightarrow$ constante sur chaque orbite de φ.

$$\begin{array}{r|cccc} \varphi : & (1,-6,-4,-3) & (-2,3,-5,6) & (5,4) & (2,-1) \\ \lambda : & f_1 = \lambda(1) & f_2 = \lambda(-2) & f_3 = \lambda(5) & f^\infty = \lambda(2) \end{array}$$

λ permet d'associer une seule valeur à tous les brins d'une même orbite :

$$\lambda(1) = \lambda(-6) = \lambda(-4) = \lambda(-3) = f_1$$

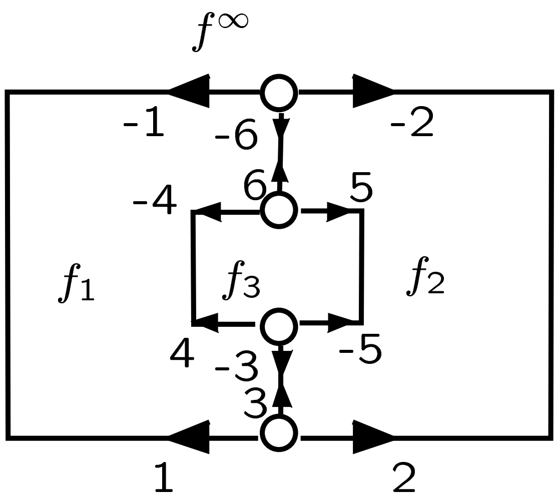
Fonction d'équitage inverse λ^{-1} ssi :

$$\forall b \in \boldsymbol{\mathcal{B}}\ \lambda^{-1}(\lambda(b)) \in \varphi^*(b)$$

exemple :

$$\begin{array}{rcl|rcl} \lambda^{-1}(f_1) & = & 1 & \lambda^{-1}(f_3) & = & 5 \\ \lambda^{-1}(f_2) & = & 3 & \lambda^{-1}(f^\infty) & = & -1 \end{array}$$

Fonctions d'étiquetage
Illustration



$$G = (\mathcal{B}, \sigma, \alpha),\ \mathcal{B} = \{-6, \ldots, -1, 1, \ldots, 6\}$$

$$\begin{array}{r|cccc}
\varphi : & (1, -6, -4, -3) & (-2, 3, -5, 6) & (5, 4) & (2, -1) \\
\lambda : & f_1 = \lambda(1) & f_2 = \lambda(-2) & f_3 = \lambda(5) & f^\infty = \lambda(2)
\end{array}$$

Implémentation

$$
\begin{aligned}
\sigma &= (1,2,3)(-3,-5,4)(5,6,-4)(-6,-2,-1)\\
&= \begin{pmatrix} -6 & -5 & -4 & -3 & -2 & -1 & 1 & 2 & 3 & 4 & 5 & 6\\ -2 & 4 & 5 & -5 & -1 & -6 & 2 & 3 & 1 & -3 & 6 & -4 \end{pmatrix}
\end{aligned}
$$

$$
\begin{aligned}
\varphi &= (1,-6,-4,-3)(-2,3,-5,6)(5,4)(2,-1)\\
&= \begin{pmatrix} -6 & -5 & -4 & -3 & -2 & -1 & 1 & 2 & 3 & 4 & 5 & 6\\ -4 & 6 & -3 & 1 & 3 & 2 & -6 & -1 & -5 & 5 & 4 & -2 \end{pmatrix}
\end{aligned}
$$

Implémentation par des tableaux d'entiers :

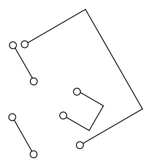
$$
\sigma = \begin{array}{r|r|}
\cline{2-2} -6 & -2 \\ \cline{2-2} -5 & 4 \\ \cline{2-2} -4 & 5 \\ \cline{2-2} -3 & -5 \\ \cline{2-2} -2 & -1 \\ \cline{2-2} -1 & -6 \\ \cline{2-2} 1 & 2 \\ \cline{2-2} 2 & 3 \\ \cline{2-2} 3 & 1 \\ \cline{2-2} 4 & -3 \\ \cline{2-2} 5 & 6 \\ \cline{2-2} 6 & -4 \\ \cline{2-2}
\end{array}
\quad
\varphi = \begin{array}{r|r|}
\cline{2-2} -6 & -4 \\ \cline{2-2} -5 & 6 \\ \cline{2-2} -4 & -3 \\ \cline{2-2} -3 & 1 \\ \cline{2-2} -2 & 3 \\ \cline{2-2} -1 & 2 \\ \cline{2-2} 1 & -6 \\ \cline{2-2} 2 & -1 \\ \cline{2-2} 3 & -5 \\ \cline{2-2} 4 & 5 \\ \cline{2-2} 5 & 4 \\ \cline{2-2} 6 & -2 \\ \cline{2-2}
\end{array}
\quad
\lambda = \begin{array}{r|l|}
\cline{2-2} -6 & 1 \\ \cline{2-2} -5 & 2 \\ \cline{2-2} -4 & 1 \\ \cline{2-2} -3 & 1 \\ \cline{2-2} -2 & 2 \\ \cline{2-2} -1 & 4 \\ \cline{2-2} 1 & 1 \\ \cline{2-2} 2 & 4 \\ \cline{2-2} 3 & 2 \\ \cline{2-2} 4 & 3 \\ \cline{2-2} 5 & 3 \\ \cline{2-2} 6 & 2 \\ \cline{2-2}
\end{array}
\quad
\lambda^{-1} = \begin{array}{r|r|}
\cline{2-2} 1 & 1 \\ \cline{2-2} 2 & 3 \\ \cline{2-2} 3 & 5 \\ \cline{2-2} 4 & -1 \\ \cline{2-2}
\end{array}
$$

Récapitulatif

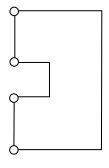
- Codage du graphe par des permutations
- Codage implicite du graphe dual (permutation φ)
- Codage explicite de l'orientation (permutation σ)

Graphes duaux/Cartes combinatoires :

Graphes duaux $\rightarrow$ Liste des frontières d'une région

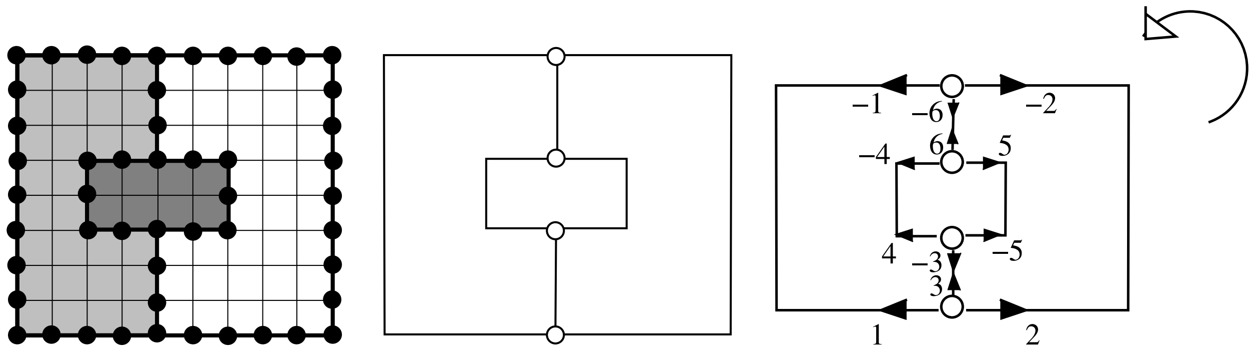


Cartes combinatoires $\rightarrow$ Liste **ordonnée** des frontières.



Extension

Association géométrie - topologie.



Associer :
– Les noeuds et les sommets
– Les segments et les arêtes
– $\Rightarrow$ Les régions et les faces
Possibilité de mise à jour efficace de la structure après une découpe.