

Cours traitement d'image

Saida Bouakaz

saida.bouakaz@univ-lyon1.fr

Introduction

- Notre environnement spatial est composé d'objets
 - Couleurs
 - Textures
 - Structure (3D)

- L'humain sait interpréter et décrire cet environnement



- Or l'information disponible sur la rétine n'est qu'un ensemble de points
- Pour un capteur on parle de pixels (picture elements)

L'histoire de ce domaine qu'est la vision artificielle est généralement découpée en quatre grandes parties :

Cible : observateur humain

- Transmission (avec le premier client : la presse).

Avant l'apparition des satellites : moyens limités (1920) 1 semaine pour 1 image entre Londres et N. York

Système Bartlane --> 3 heures : image codée 5 niveaux/point → Reconstruction après 1923 → on passe à 15 niveaux

Des premières images à leur traitement (1950-1970)

Début du *traitement* d'images : 1950 (avec les premières analyses d'images dans les chambres à bulles). Application : Rayons X, OCR, ...

Des images de mauvaise qualité et très volumineuses (déjà 700 x 500 pixels sur 8bits par image)

Trois domaines dominants :

Restauration (corriger les défauts liés à l'acquisition)

Amélioration (rendre l'image "belle" pour l'affichage)

Compression (réduire le volume de l'image)

Du traitement à « l'interprétation » d'images (1970 - 1980) :

- Une évolution naturelle vers l'extraction automatique d'informations
- Apparition de la notion de description structurelle (ensemble des informations contenues dans une image et que l'on peut extraire avec ou sans aucune connaissance a priori de l'application).
- Des nouveaux thèmes : seuillage, segmentation, extraction de contours, morphologie mathématique

Interprétation (au départ lié à l'IA)

- Un engouement explosif au moment de l'apparition des systèmes experts (à rapprocher de l'IA générative actuelle)
- Une déception proportionnelle à l'engouement

--> Les raisons de l'échec

- problème mal posé : il n'y a pas d'expert pour la vision (règles difficile à cerner)
 - le savoir est trop complexe pour être décrit formellement avec des règles et résumé par un seul modèle
- Retour de l'intérêt de l'IA pour la vision avec les réseaux : apprentissage (puissance des machine, apprentissage par la quantité de données et d'exemple)

La vision par ordinateur (vers 1980) :

- de l'image 2D aux modèles tri-dimensionnels
- Analyse du mouvement

--> Vision pour la robotique (mouvement, 3D, détection d'obstacle, trajectoire)

S'appuie sur le paradigme de Marr

!! Le bilan : pousse à revoir les buts (peu de problèmes généraux résolus mais a donné de l'élan à beaucoup de domaine
- 3D complet est-il possible ? est-ce même une solution, une quête ?

De la vision passive à la vision active !!!

Et maintenant...

- Plus réaliste, conscient des limites.

--- On sait que l'analyse dépend du but poursuivi ---

-->>Un nouvel élan :



- Robots/IA
- la réalité augmentée / Metaverses
- les bibliothèques numériques
- Représentation (quels traitement, quel codage ?)
-

il ne s'agit plus d'interpréter pour interpréter mais d'interpréter
pour recherche restituer une information ou faire une action
(reconstruction basée images, Indexation, robotique
« apprentissage profond », etc.)

Traitement, segmentation, reconnaissance

- **Revue (les plus anciennes)**
 - IEEE Transactions on Pattern Analysis and Machine Intelligence
 - Pattern Recognition
 - Computer Vision and Image Understanding
 - International Journal of Computer Vision
- **Conférences (les plus connues)**
 - International Conference on Pattern Recognition (ICPR)
 - International Conference on Image Processing (ICIP)
 - International Conference on Computer Vision (ICCV)
 - International Conference on Computer Vision and Pattern Recognition (CVPR)
- **Recherche bibliographique (Computer Vision Bibliography)**
 - <http://iris.usc.edu/Vision-Notes/bibliography/contents.html>

1- Outils Fondamentaux

Quelques définitions

L'image peut être plongée dans le plan discret : \mathbb{Z}^2

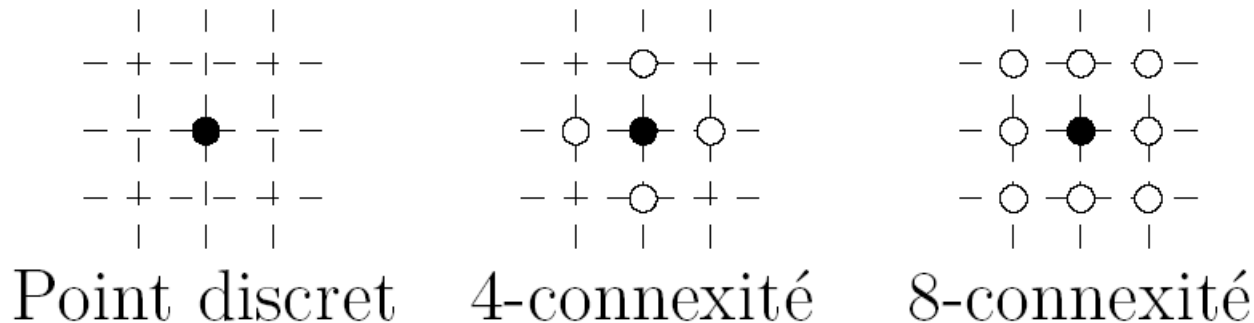
Point discret : point de \mathbb{Z}^2 , correspond au pixel centré en ce point

2 points discrets P et Q sont dits **4-adjacents** ssi

$$|x_P - x_Q| + |y_P - y_Q| = 1$$

2 points discrets P et Q sont dits **8-adjacents** ssi

$$\text{Max}(|x_P - x_Q|, |y_P - y_Q|) = 1$$



Distances Classiques - Rappel

L'image peut être plongée dans le plan discret : \mathbb{Z}^2

Point discret : point de \mathbb{Z}^2 , correspond au pixel centré en ce point

Distance euclidienne : P et Q

$$D_2(P, Q) = ((x_P - x_Q)^2 + (y_P - y_Q)^2)^{\frac{1}{2}}$$

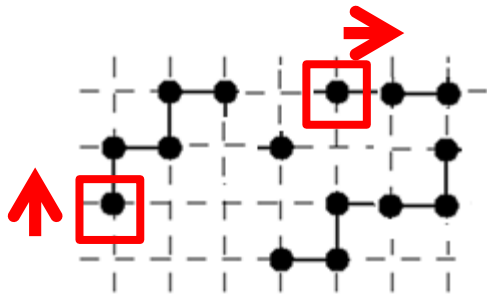
$$L_1(P, Q) = |x_P - x_Q| + |y_P - y_Q|$$

$$L_\infty(P, Q) = \text{Max}(|x_P - x_Q|, |y_P - y_Q|)$$

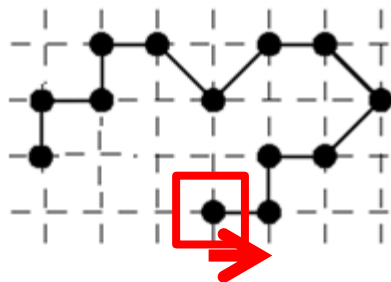
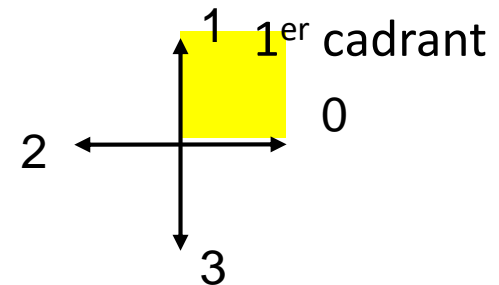
Chemin discret et connexité

Un **chemin discret** k -connexe est une suite de point discrets (P_0, P_1, \dots, P_n) tels que $\forall i \in \{1, \dots, n\}$, P_{i-1} et P_i sont **k -adjacents**.

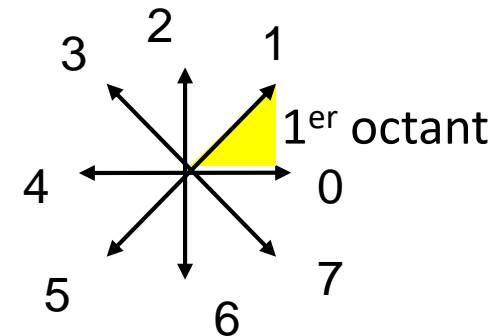
Code de Freeman : la suite (P_0, P_1, \dots, P_n) est représentée par $(P_0, d_0, \dots, d_{n-1})$. La valeur d_i code le déplacement relatif de P_i à P_{i+1} avec les configurations :



Codage du chemin en :
 $(P_0, 1010) ; (P_1)$
 $(P_2, 00332232)$



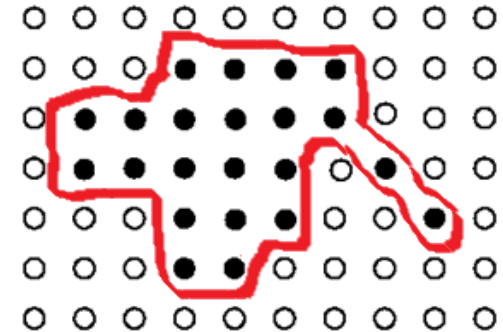
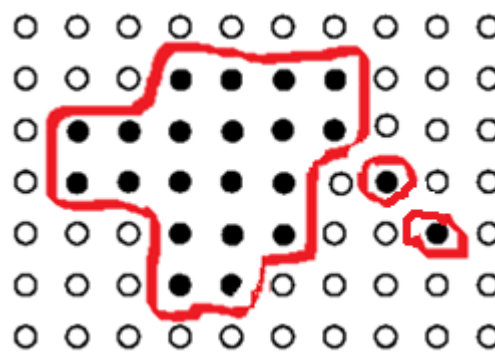
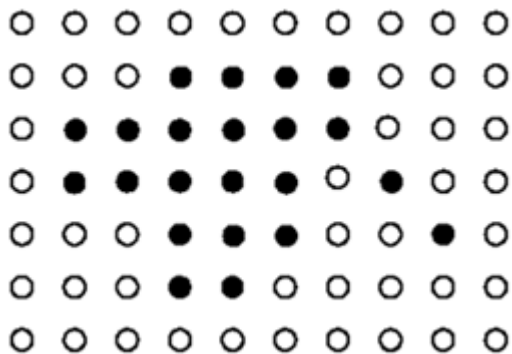
Codage du chemin en :
 $(P, 020134534646)$



Composante connexe / région 2D

Ensemble k-connexe E : ensemble de points discrets tels que $\forall P_1, P_2 \in E$, il existe un chemin discret k-connexe dans E d'extrémités P_1 et P_2 .

Composante connexe d'un ensemble de points discrets : ensemble connexe maximal E tels que $\forall P_1, P_2 \in E$, il existe un chemin discret k-connexe dans E d'extrémités P_1 et P_2 .

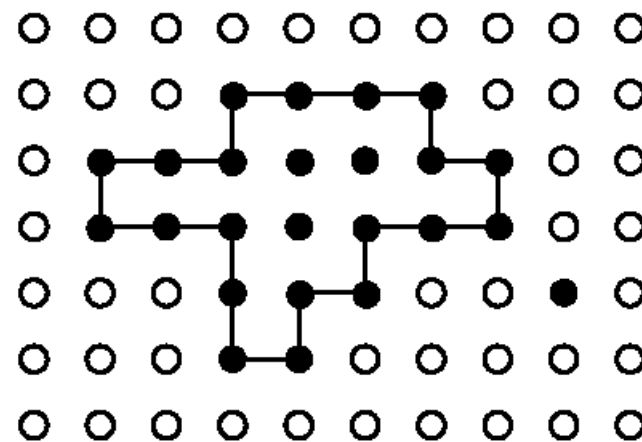
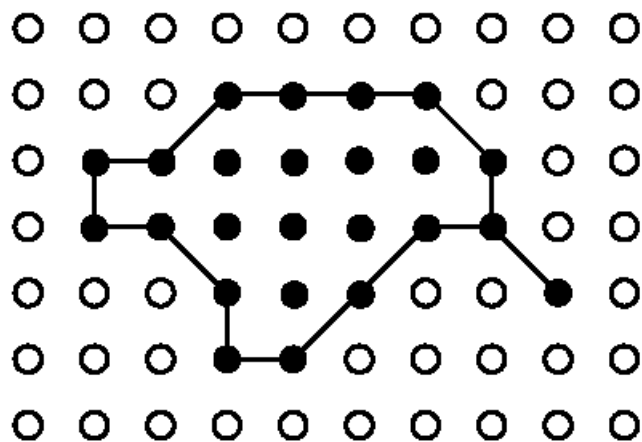


Frontière d'une région 2D

Définition classique

La frontière d'une région 8-connexes (respectivement 4-connexes) R est l'ensemble des points de R dont au moins un des 4-voisins (resp. 8-voisins) n'est pas élément de R .

⇒ La frontière est composée de chemins 8-connexes (resp 4-connexes).



Produit de convolution

Soit $h : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$;

Soit f une fonction donnée ; $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$

La convolution de f par h est définie par :

$$(f * h)(x, y) = \iint_{\mathbb{R}} f(u, v) h(x_u, y - v) du dv$$

$$(h * f)(x, y) = \iint_{\mathbb{R}} h(u, v) f(x_u, y - v) du dv$$

$$f * h(x, y) = h * f(x, y)$$

Passage au domaine discret de l'image

Soit I le support d'une image,
On définit la fonction h comme suit :

$$h : [m_1, m_2] \times [n_1, n_2] \rightarrow R$$

La convolution de I par h est définie par :

$$(fI * h)(x, y) = \sum_{u=-m_1}^{m_2} \sum_{v=-n_1}^{n_2} h(u, v) I(x - u, y - v)$$

h est appelée noyau de la convolution ou du filtre
On utilise également le terme de masque

Quelques propriétés de la convolution

Associativité

$$f * h * g = (f * h) * g$$

Commutativité

$$f * h = h * f$$

Distributivité +/*

$$(f * h)' = f * h'$$

Chapitre 1

Traitements à base d'histogramme

Opérations ponctuelles

Moyenne : la Moyenne (ou Luminance, Brillance) d'une image est définie comme étant la moyenne des pixels dans l'image I à N_l lignes et M_c colonnes

$$Moy = \frac{1}{N_l M_c} \sum_{i=0}^{N_l-1} \sum_j^{M_c-1} I(i, j)$$

Ecart type : des variations des niveaux de gris dans l'image

$$\sqrt{\frac{1}{N_l M_c} \sum_{i=0}^{N_l-1} \sum_j^{M_c-1} (I(i, i) - Moy)^2}$$

Histogramme

- L'Histogramme d'une image représente le nombre (ou la proportion) de pixels en fonction du niveau de gris dans l'image.

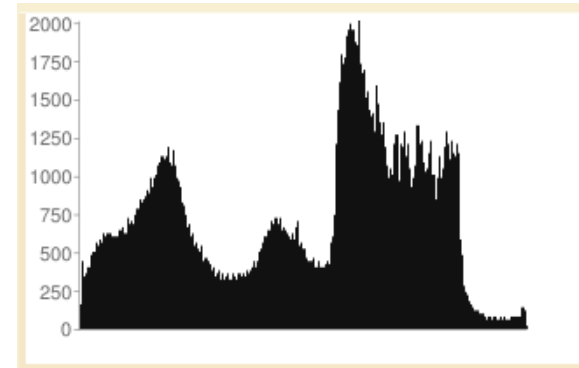
D'un point de vue mathématique, l'histogramme cumulé correspond à une fonction que l'on peut noter $H(l)$ telle que :

$$H(l) = \sum_{i,j} X_{(I(i,j)=l)}$$

$X_{(.)}$ est la fonction indicatrice : elle vaut 1 lorsque son argument est vrai

La manipulation de l'histogramme permet de changer l'apparence (augmenter le contraste, changer la dynamique, ...) de l'image, catégoriser les éléments, ...





Remarques

L'histogramme d'une **image surexposée** aura tendance à avoir ses pics poussés vers la droite, avec une accumulation de pixels vers les niveaux de gris ou les valeurs de couleur élevés.

Une image **sous-exposée** aura un histogramme déplacé vers la gauche. Les pixels seront concentrés vers les niveaux de gris basses

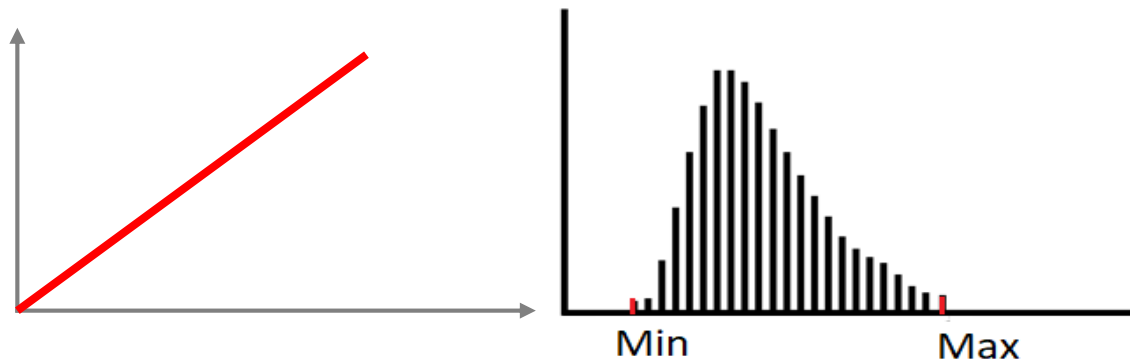
Un histogramme avec une **répartition répartie** sur toute la plage de niveaux de gris (ou de couleur) indique un bon contraste

Un histogramme concentré sur une **plage restreinte** peut indiquer un faible contraste.

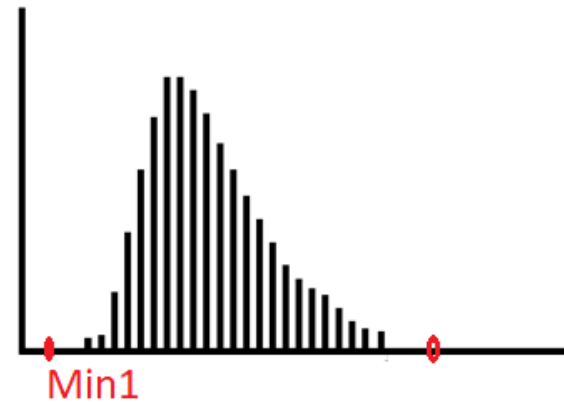
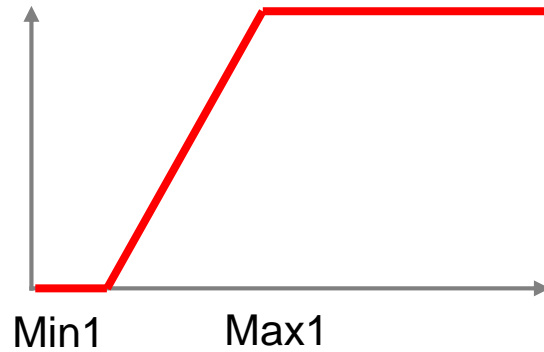
Opérations ponctuelles (action sur le contraste)

Transformations Linéaires

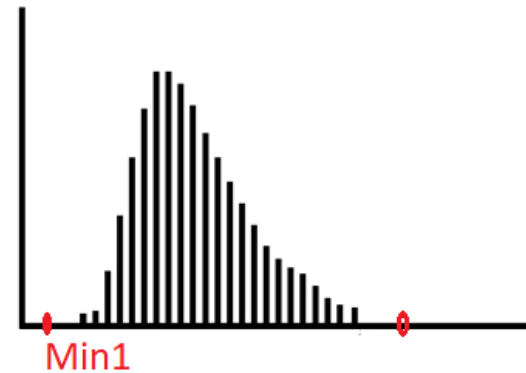
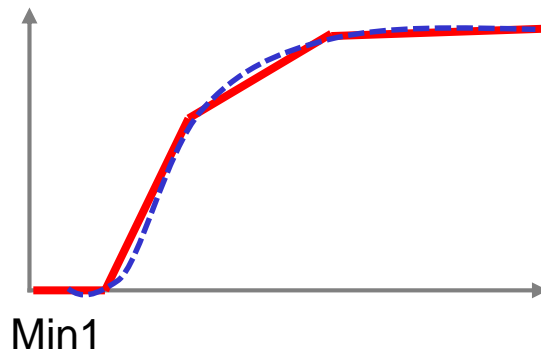
$$F(x, y) = 255 * \frac{f(x, y) - \text{Min}}{\text{Max} - \text{Min}} \quad \text{on a} \quad \frac{f(x, y) - \text{Min}}{\text{Max} - \text{Min}} \in [0, 1]$$



Transformation par morceaux

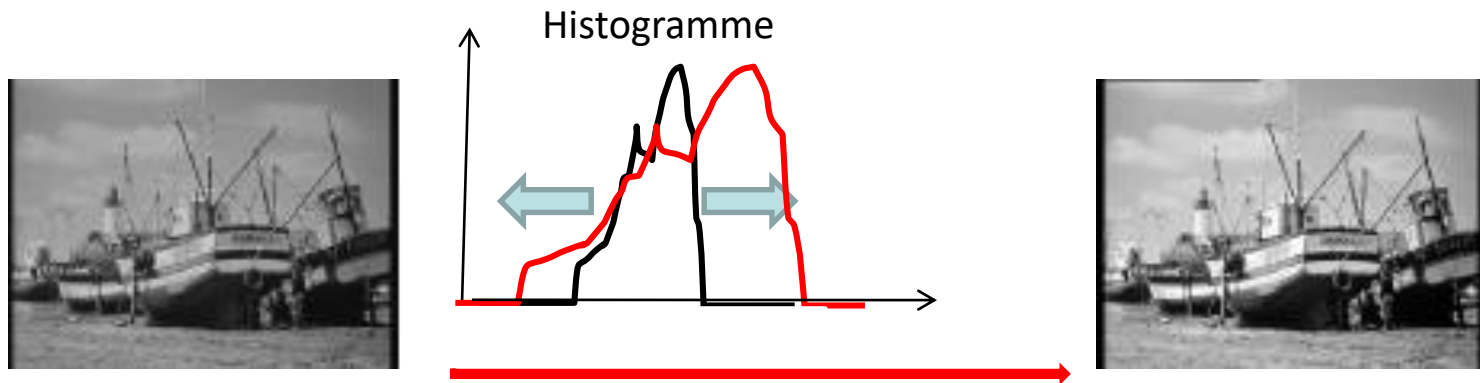


$$F(x, y) = 255 * \frac{f(x, y) - \text{Min1}}{\text{Max1} - \text{Min1}} (f(x, y) - \text{Min1})$$



Etirement d'histogramme (expansion)

consiste à répartir les valeurs de niveau de gris des pixels (majoritairement répartis sur un intervalle $[Min, Max] \subset [0, 255]$) sur toute la largeur de l'intervalle disponible.



Une telle application permet de corriger la surexposition

Par exemple, on rapporte la valeur d'intensité la plus faible à zéro et la plus haute à la valeur maximale.

Egalisation d'histogramme

L'Objectif est de cette opération est de réajuster le contraste dans une image.

L'**égalisation** d'histogramme est une transformation de l'intensité lumineuse dont le principe est **d'équilibrer/harmoniser** le mieux possible la distribution des pixels dans l'intervalle disponible.

La tendance est d'arriver à avoir que chaque niveau de l'histogramme contienne le même nombre de pixels. Concrètement, on essaye d'aplatir au maximum l'histogramme original.

Egalisation d'histogramme

L'Objectif est de cette opération est de réajuster le contraste dans une image.

Expression :

$$F(x, y) = 255 * \frac{Hc(l)}{Nb}$$

$$F(x, y) = 255 * \sum_{i=0}^{i=l} \frac{h(l)}{Nb}$$

Avec : Nb : nombre de pixel dans l'image,

$h(l)$: valeur de l'histogramme pour la valeur l

$Hc(l)$ histogramme cumulé pour la valeur l

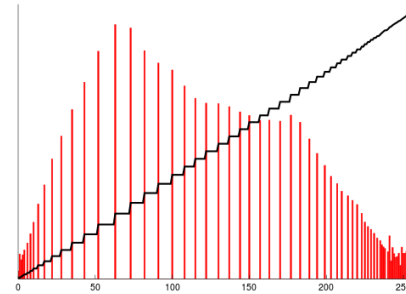
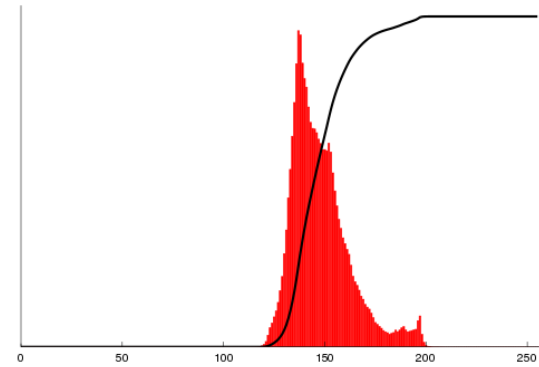
$l = f(x, y)$

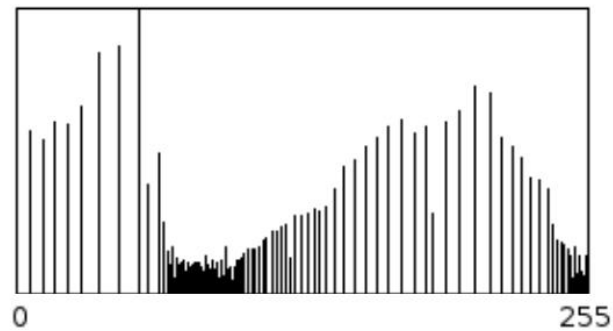
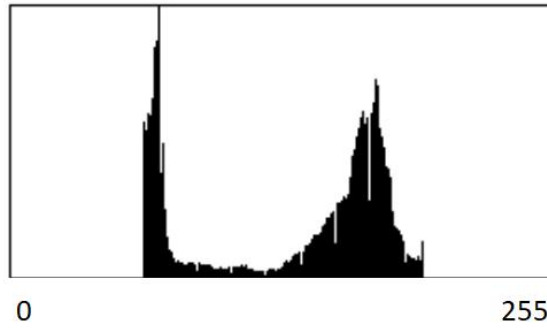
- Autre alternative : $F(x, y) = (2^D - 1) * \frac{Hc(l)}{Nb}$

Etapes

1. Calcul de l'histogramme $h(i)$ pour $i \in [0, 255]$
2. Normalisation de l'histogramme $h_n(i) = \frac{h(i)}{Nb}$ $i \in [0, 255]$
3. Calcul de la densité de probabilité cumulée $C(i) = \sum_{i=0}^l h_n(i)$; $i \in [0, 255]$
4. Transformation des valeurs de N. G de l'image
$$F(x, y) = C(f(x, y)) * 255$$

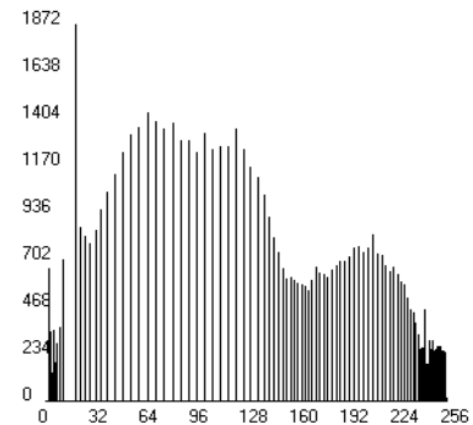
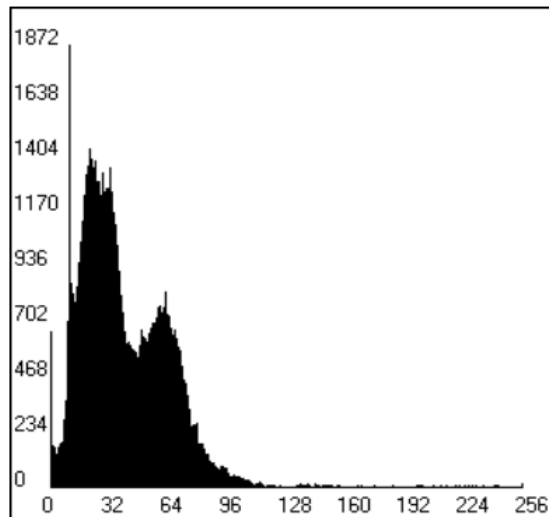
Partie traitement d'image





Egalisation d'histogramme

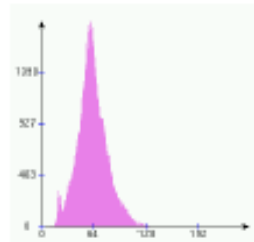
→ On augmente globalement le contraste



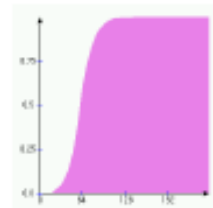
Autre exemple



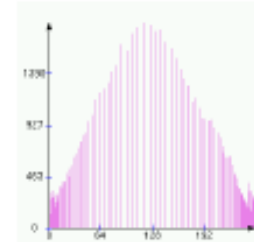
Original $f[x,y]$



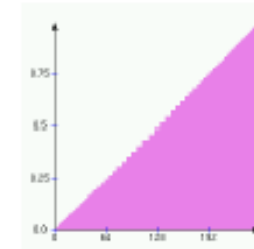
Histogramme de f



Histogramme cumulé de f



Histogramme de $f_{\text{ég}}[x,y]$



Histogramme cumulé de $f_{\text{ég}}[x,y]$



Après égalisation $f_{\text{ég}}[x,y]$

Exemple tiré de : Antoine Manzanera (compléter la référence !!)

Spécification d'histogramme

Rendre la distribution d'intensité de l'image voisine d'une distribution spécifiée à l'avance.

- Conservation des positions et sens des transitions avec une transformation :
 $F \rightarrow G$ croissante.
- Distribution de référence = image ou région d'une image

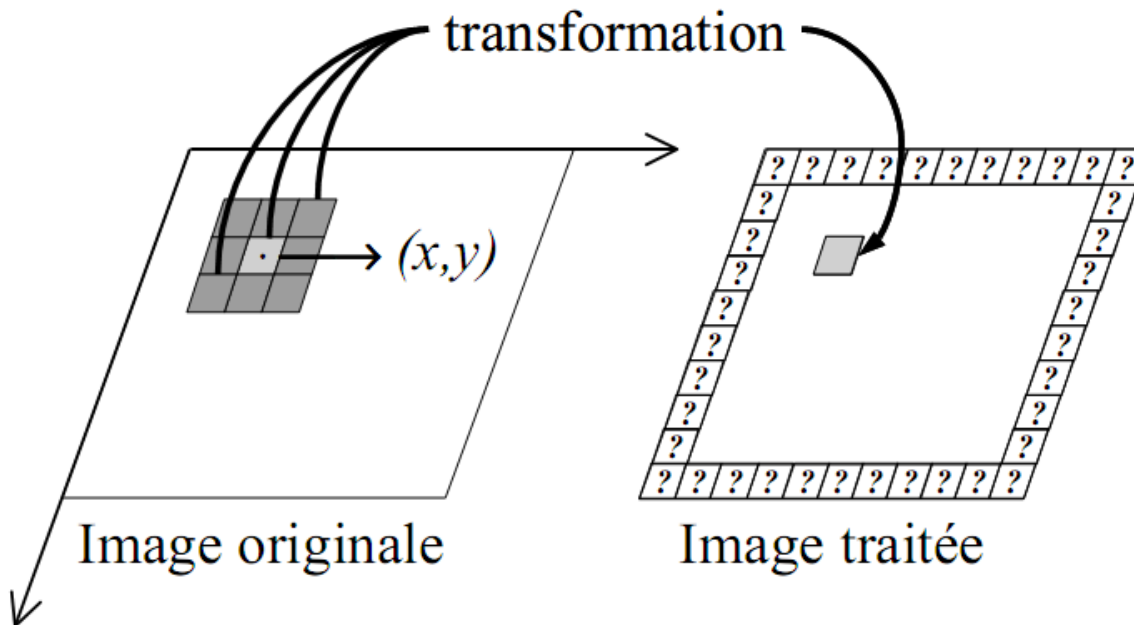
- Filtrage spatial
 - Lissage et réduction du bruit

Les filtres de lissage sont des opérateurs qui éliminent des éléments perturbateurs (ou bruit) / non significatifs dans les images numériques. L'objectif est d'améliorer la visualisation, ou simplifier les données dans le but d'un traitement postérieur :

- Filtrage par convolution
- Filtrage dans le domaine de Fourier

- Filtrage spatial
 - Lissage et réduction du bruit

Transformation basée sur le voisinage (spatial) d'un point (x,y) , sur la base d'opérateurs de convolutions.

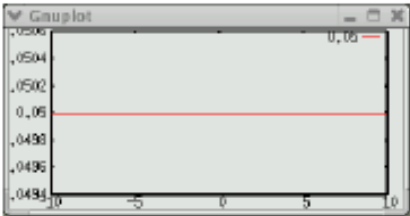
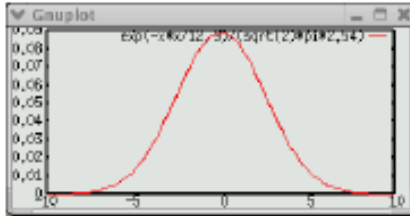
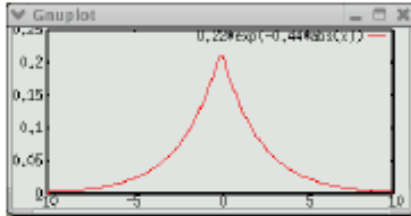


Filtrage par convolution

multiplication dans le domaine fréquentiel → convolution dans le domaine spatial

Filtre de lissage obtenu un noyau de convolution symétrique et normalisé
Somme des coeffs = 1 (pour conserver certaines propriétés)

Exemples de noyaux de filtres

Moyenne	Gaussienne	exponentiel
$h(x, y) = \frac{1}{\lambda^2} \text{ si } (x, y) \in \left[-\frac{\lambda}{2}, \frac{\lambda}{2}\right]^2$ $h(x, y) = 0 \text{ sinon}$	$h(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$	$h(x, y) = \frac{\gamma^2}{4} \exp(-\gamma(x + y))$
 $\frac{1}{25} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$ <p><i>Filtre moyenneur (5x5)</i></p>	 $\frac{1}{864} \begin{pmatrix} 11 & 23 & 29 & 23 & 11 \\ 23 & 48 & 62 & 48 & 23 \\ 29 & 62 & 80 & 62 & 29 \\ 23 & 48 & 62 & 48 & 23 \\ 11 & 23 & 29 & 23 & 11 \end{pmatrix}$ <p><i>Filtre gaussien ($\sigma = 1,41$)</i></p>	 $\frac{1}{80} \begin{pmatrix} 1 & 1 & 3 & 1 & 1 \\ 1 & 3 & 7 & 3 & 1 \\ 3 & 7 & 16 & 7 & 3 \\ 1 & 3 & 7 & 3 & 1 \\ 1 & 1 & 3 & 1 & 1 \end{pmatrix}$ <p><i>Filtre exponentiel ($\gamma = 0,8$)</i></p>

Filtrage par convolution

Masques (noyaux) à taille fini (réduite)

 $\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

 $\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

Domaine
spatial

Domaine
fréquentiel

$$f(x, y) * h(x, y) \Leftrightarrow F(u, v) H(u, v)$$

Convolution Multiplication

$$f(x, y) h(x, y) \Leftrightarrow F(u, v) * H(u, v)$$

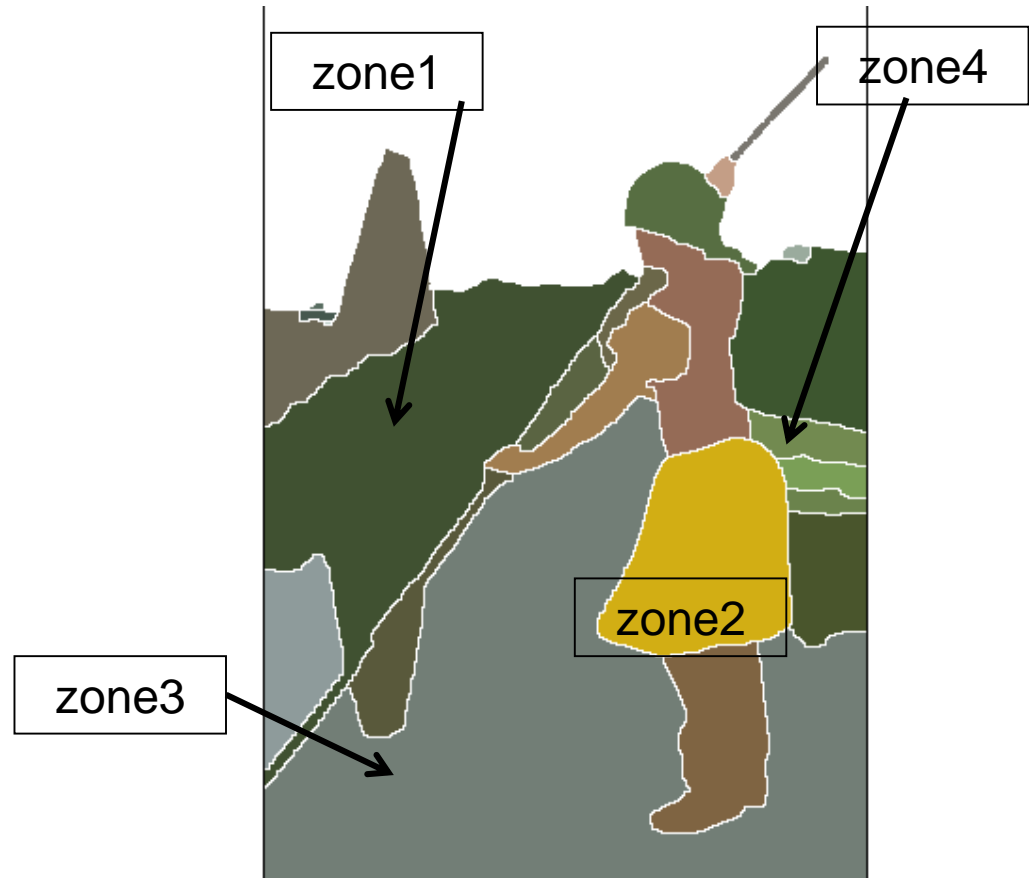
Multiplication Convolution

Chapitre 2

Segmentation : quelques méthodes

Segmenter une image?

*Partitionner l'image en zone homogènes selon un critère déterminé:
couleur, texture, niveau de gris, indice,...*



Un problème difficile.

Pourquoi les résultats des algorithmes sont largement moins performants que la vision humaine ?

- Chaque cas est particulier : difficile pour un traitement automatique
 - *Condition ne sont pas tjrs les mêmes*
 - *Difficile de tenir compte du local et du global (problème de temps, difficulté de formaliser la sémantique)*
 - Pas d'exploitation explicite de la continuité spatiale ou/et temporelle (cas du mouvement)
 - Difficile d'introduire une sémantique globale !!!!
-

Définition

Formal Definition

- ◆ Given region R and uniformity criterion U , define predicate $P(R) = \text{True}$, if $\exists a \ni |U(i,j) - a| < \epsilon$,
 $\forall (i,j) \in R$
- ◆ Partition image into subsets R_i , $i = 1, \dots, m$, such that
 - ◆ Complete: $\text{Image} = \cup R_i$, $i = 1, \dots, m$
 - ◆ Disjoint subsets: $R_i \cap R_j = \emptyset$, $\forall i \neq j$
 - ◆ Uniform regions: $P(R_i) = \text{True}$, $\forall i$
 - ◆ Maximal regions: $P(R_i \cup R_j) = \text{False}$, $\forall i \neq j$

Prédicat $P(R_i)$: la région R_i est homogène »

- Contraste : $P(R_i)$ vrai $\Leftrightarrow \text{Max}_{R_i}(f(x, y)) - \text{Min}_{R_i}(f(x, y)) < \alpha$
- Écart-type : $P(R_i)$ vrai $\Leftrightarrow \sqrt{\frac{1}{N} \sum (f(x, y) - m)^2} < \sigma$
- Distance interquartile
- Différence limitées : $P(R_i)$ vrai $\Leftrightarrow (\forall k, l) \text{voisin } |f(k) - f(j)| < \sigma$

Segmenter une image I c'est réaliser sa partition en n régions

$\{R_i\}_{i=1, \dots, n}$ tel que :

$$I = \bigcup_i R_i$$

$$\forall i \neq j \quad R_i \cap R_j = \emptyset$$

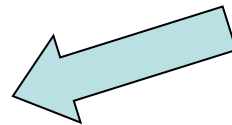
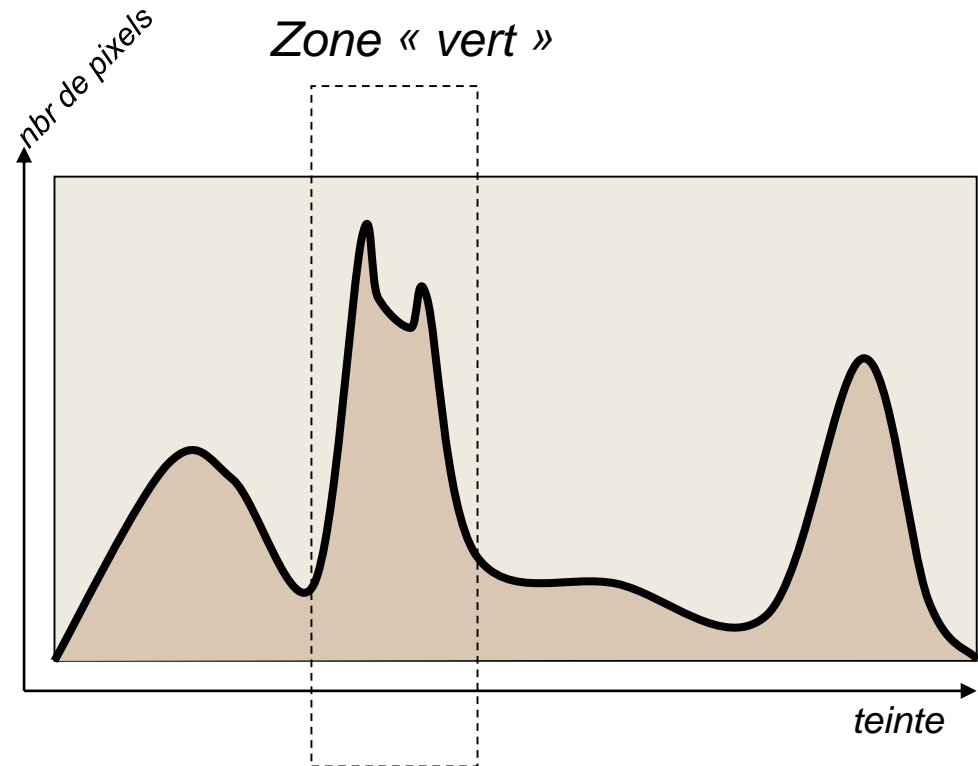
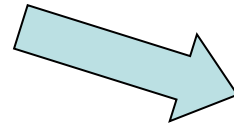
$P(R_i \cup R_j) \quad \forall (R_i \text{ et } R_j) \text{ deux régions adjacentes (toute région est connexe)}$

Plusieurs approches:

On ne sait pas choisir quelle est la "bonne" partition.
Choix empiriques :

- Approches **GLOBALES**: histogrammes
- Approches **LOCALES**: region growing,
split & merge
contours
- Approches presque **HYBRIDES** ?

Idée: Si les objets présents dans l'image ont des couleurs bien distinctes et uniformes, ils vont apparaître comme des pics dans l'histogramme.



=> Segmentation dans un espace dérivé de l'image

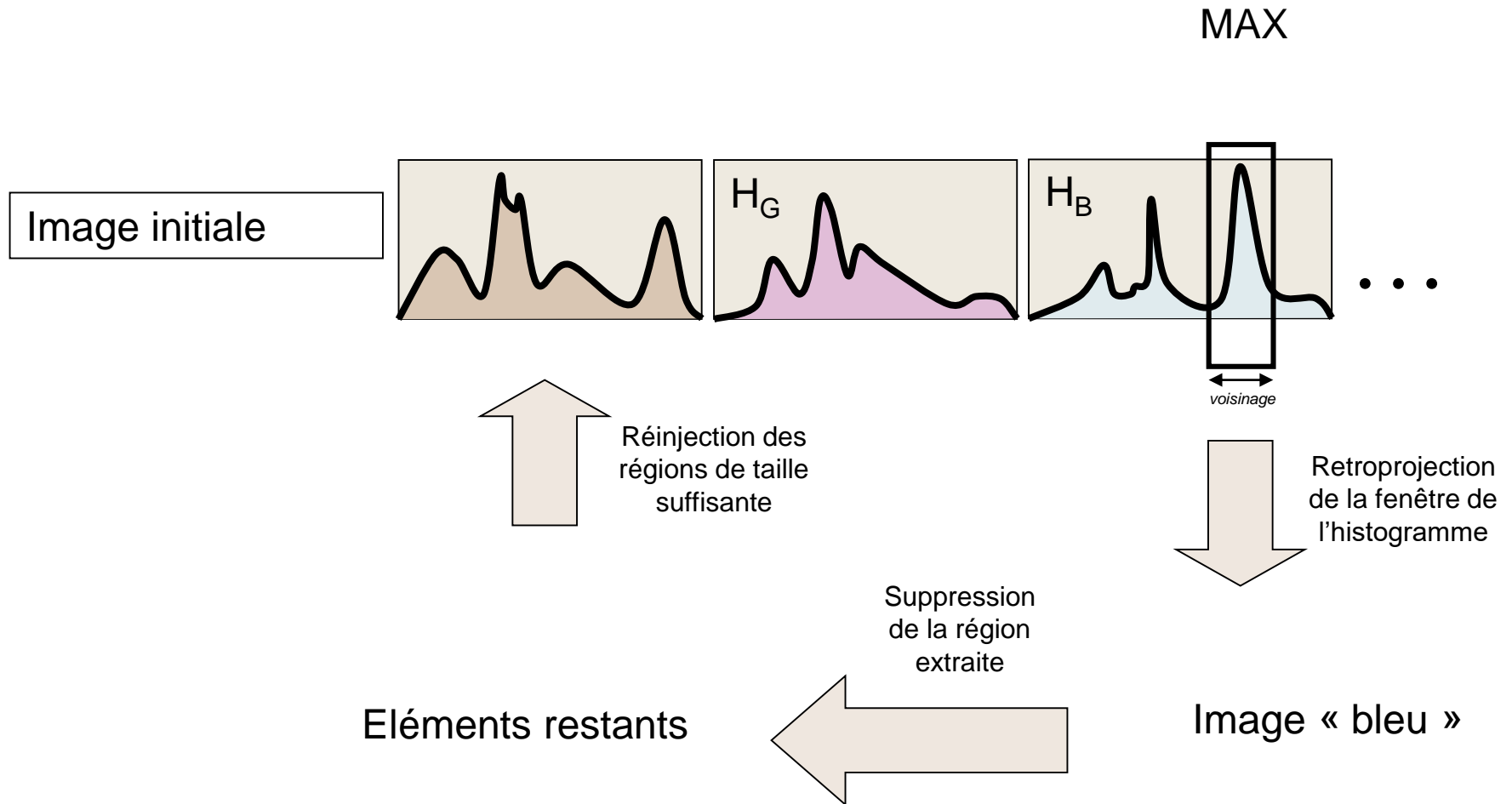
Exemple de technique connue



Tirée de : images.icube.unistra.fr/fr/img_auth.php/.../20161017143742%213-Segmentation.pdf

Chaque pixel est décrit selon certains canaux : R,G,B,H,S,V,...

=> *L'algorithme travaille sur plusieurs histogrammes, un par channel*



AVANTAGES

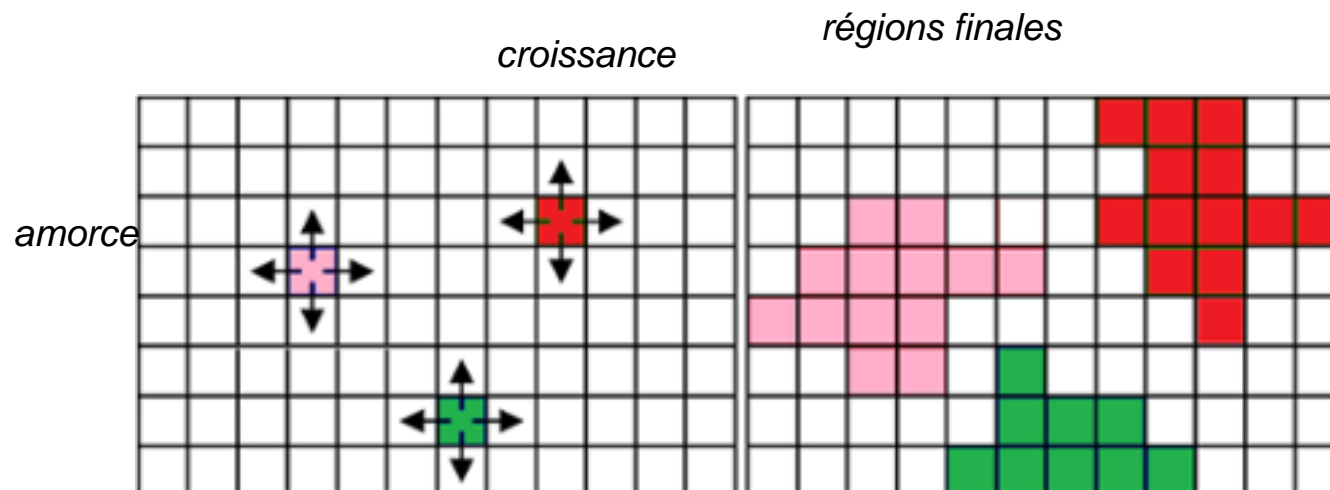
- Méthode très rapide

INCONVENIENTS

- Méthode globale : ignore les informations de proximité qui permettent d'utiliser des seuils variables locaux.
- Sensible au bruit
- Que se passe-t-il si deux objets ont la même couleur ?
→ nécessite en général d'affiner le résultat et de détacher les composantes connexes (→ méthode complémentaire).

Idée: Partant d'un point amorce (germe) et l'on l'étend en ajoutant les points adjacents de la frontière la région réduite qui satisfont le critère d'homogénéité

Ainsi, on associe aux germes les pixels adjacents qui sont en accord avec le prédicat.





- Le point amorce peut être choisi soit par un humain, soit de manière automatique en évitant les zones de fort contraste (gradient important).
- Si le critère d'homogénéité est local (comparaison de la valeur du pixel candidat et du pixel de la frontière).

Méthode par semence grossissement

- Initialement, on pose des germes \rightarrow ***R contient l'amorce.***
- Initialement, on part de ***S qui contient le voisinage du germe.***

Test local (pixels voisins)
ou statistique global
(calcul sur la couleur
moyenne de R)

\rightarrow On peut utiliser une structure S qui contient les points frontière, à partir desquels on propage

- On procède à la propagation du germe (agglomération)

AVANTAGES

- Méthode rapide
- Conceptuellement très simple

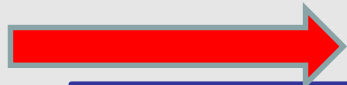
• Méthode locale: aucune vision globale du problème. En pratique, il y a presque toujours un chemin continu de points connexes de couleur proche qui relie deux points d'une image...

Problème de passage:

- Tenir compte de l'homogénéité globale donne un algorithme sensible à l'ordre de parcours des points (méthode par amorce)
- Algorithme très sensible au choix du critère de propagation.



---> Agglomérations multiples : plusieurs germes



2 étapes

- ❑ Etape locale: plusieurs agglomérat
 - ❑ Tenir compte de l'homogénéité locale
- ❑ Etape globale : revenir sur les différents agglomérat et fusionner

Peut être + efficace !!!!!

→→→ Evoquer les différentes stratégies

Idée: Diviser pour régner ---> « tri fusion »

•>>>>>>> 2 PHASES

Phase 1 : Subdiviser l'image en zones homogènes élémentaires (C1) = SPLIT

Phase 2 : fusionner les zones adjacentes (C2) = MERGE

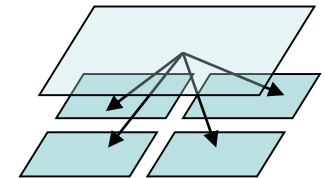
SPLIT:

Pour que la subdivision soit automatique --->> Procédé (structure de données + Critère

De façon naïve :

L'image est stockée dans un arbre.

Initialement, *arbre racine = image complète*



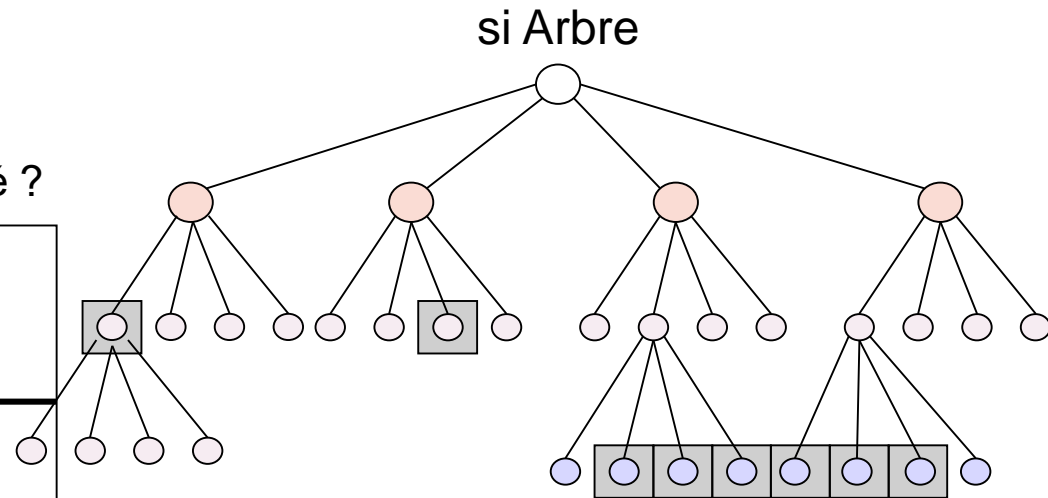
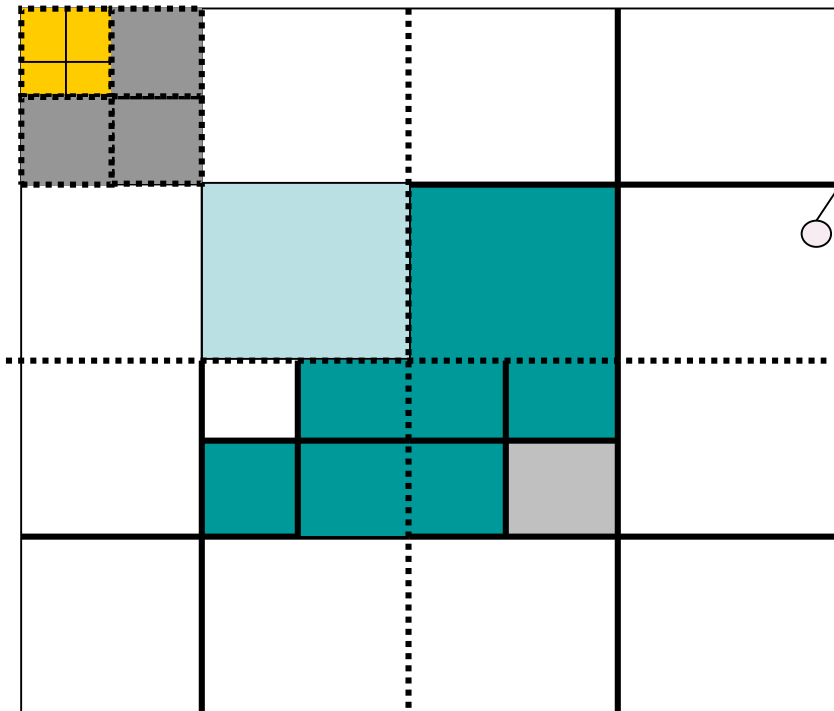
Récurivement, chaque feuille F est subdivisée en quatre si elle n'est pas « C1-homogène », et les quatre sous images sont ajoutée en tant que feuilles de F.

Fusion : Merge

coder la propriété d'adjacence

Construction du RAG : graphe d'adjacence ? Comment le coder*Region Adjacency Graph*

- Connecte les régions adjacentes
- Arrêtes = val du critère d'homogénéité ?

**Faire le RAG**

La distance en terme d'homogénéité de régions est portée par l'arrête valuée qui les relie dans le RAG

MERGE:

- Chaque nœud du graphe d'adjacence des régions (RAG) est examiné selon l'algorithme :
 - Tant que un des voisins de ce nœud est à une distance (au sens du critère) inférieure à un seuil de regroupement, les deux nœuds fusionnent dans le RAG.

La distance en terme d'homogénéité de régions est portée par l'arrête évaluée qui les relie dans le RAG

Quelques remarques

Difficultés : structuration et codage

Gestion des adjacences

Critères : division fusion

Finesse de la subdivision (compromis : finesse \leftrightarrow temps de calcul

----> Inconvenients : structure par bloc régions obtenues ne respectent pas les arrondis

Original



Split & Merge



AVANTAGES

- Méthode presque hybride locale/globale
- simple à comprendre
- Réalise une segmentation exhaustive

INCONVENIENTS

- Méthode assez complexe (temps)
- Découpage un peu « carré », dû à la topologie des quadrees rigide
- demande un effort pour la conception des structures de données

- la segmentation par ligne de partage des eaux: considère une image comme un relief topographique, dont on simule l'inondation.

Comment ? → détection des bassins d'écoulement sur l'image de la norme du gradient (en général)

<http://cmm.ensmp.fr/~beucher/wtshed.html>

Existe-t-il une meilleure technique de segmentation?

- Non, tout dépend du contexte, de l'application
- Dans sa catégorie (segmentation région selon couleur à base de merge/split), CSC est souvent le meilleur

Existe-t-il une « super technique », peut être encore inconnue, qui obtienne de très bon résultats en segmentation?

- Non, en l'absence de contexte, la quantité d'information disponible est limitée. Les algorithmes du type CSC ne sont probablement pas loin de l'optimum.

Mais le cerveau obtient de bien meilleurs résultats... Quelle est le secret?

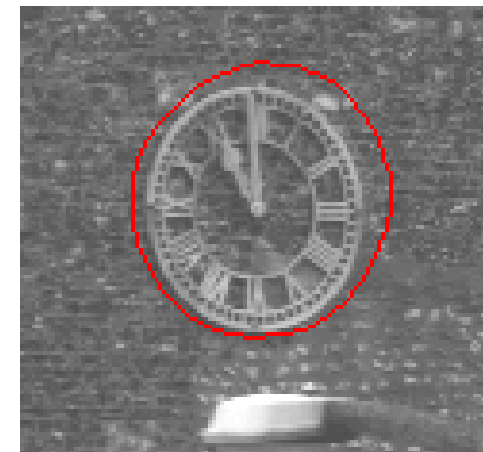
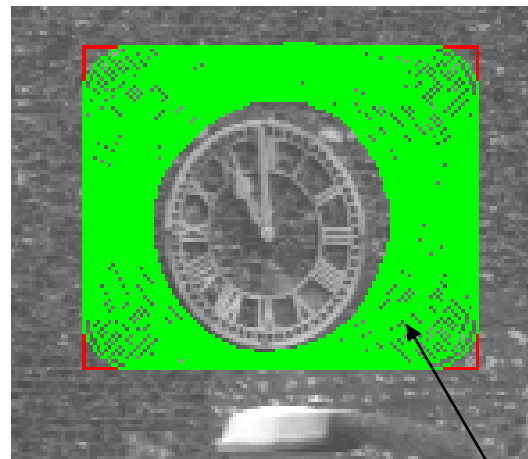
- Problème très difficile (50 ans de recherche et pas de solution)
- *Pistes crédibles*: - utilisation du contexte, vision liée aux objectifs, apprentissage, énorme bibliothèque d'algorithmes spécifiques

Outils**Librairies : OpenCV, ImageJ, matlab...**

Idée: Utiliser des courbes déformables qui sont « attirées » par les formes recherchées dans l'image.

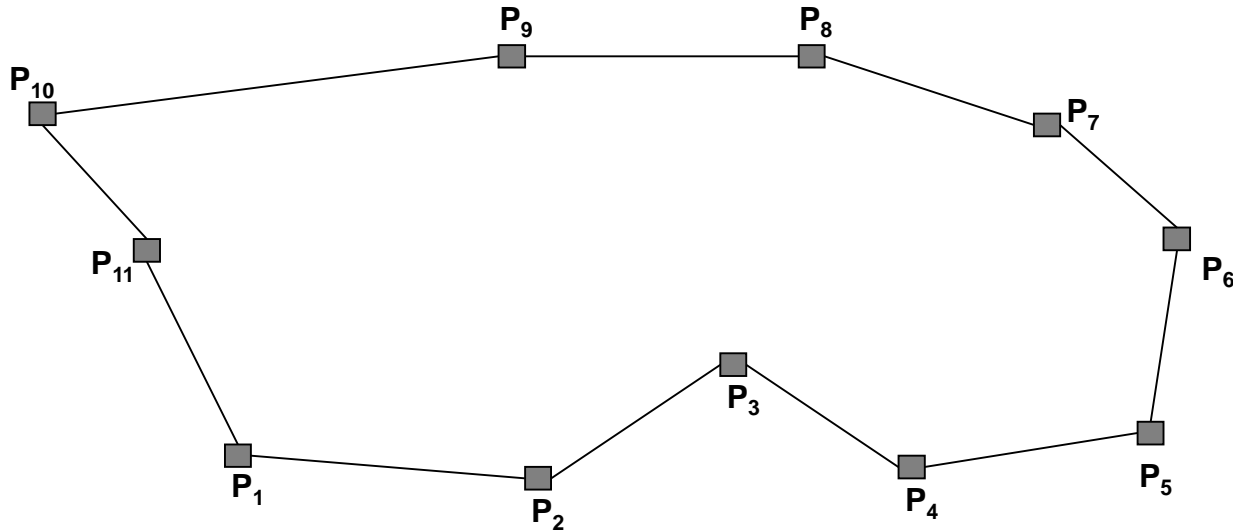
Analogie avec la physique: *fonction d'énergie à minimiser.*

Un exemple classique



Le « snake » se contracte et s'adapte à la forme de l'horloge

Qu'est ce qu'un snake?

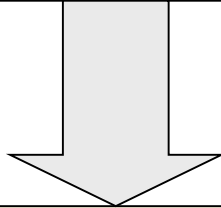


$$E_{\text{snake}} = E_{\text{interne}} + E_{\text{externe}}$$

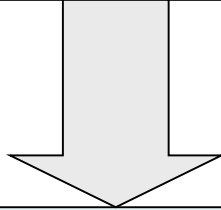
- Propriétés intrinsèques
- Longueur, courbure...

- Propriétés locales de l'image autour du snake

Comportement dynamique du snake?



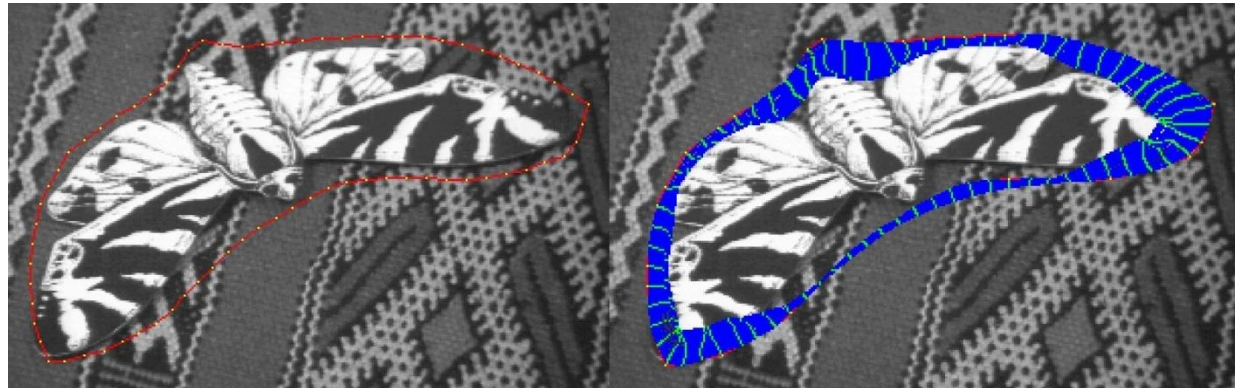
Minimiser l'énergie totale (interne + externe)



$$F = \nabla E$$

Calculer les forces à appliquer à chaque point de contrôle de telle sorte que l'énergie soit minimisée

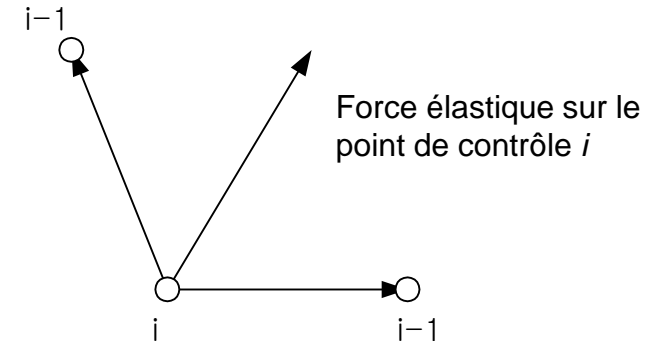
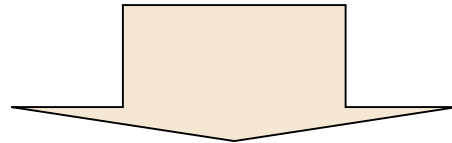
$$\begin{cases} F_x = \frac{\partial E(x, y)}{\partial x} \\ F_y = \frac{\partial E(x, y)}{\partial y} \end{cases}$$



Exemple d'énergie INTERNE:

Energie de contraction **élastique**

$$E_{elastic} = K_1 \times \sum_{i=1}^n |P_i - P_{i-1}|^2$$



$$F_{elasticX,i} = 2 \times K_1 \times ((x_{i-1} - x_i) + (x_{i+1} - x_i))$$

$$F_{elasticY,i} = 2 \times K_1 \times ((y_{i-1} - y_i) + (y_{i+1} - y_i))$$

Comportement: tend à rétrécir le snake, tout en maintenant une répartition homogène des espaces entre les points de contrôle. Forme limite: le point.

Exemple d'énergie INTERNE:

Energie **de courbure**

$$E_{\text{courbure}} = K_2 \times \sum_{i=1}^{n-1} |P_{i-1} - 2P_i + P_{i+1}|^2$$

Tend à redresser le snake et à limiter la courbure. Forme limite: le cercle

Exemple d'énergie EXTERNE (Image):

Zones brillantes ou sombres:

$$E_{\text{ext}} = \lambda I$$

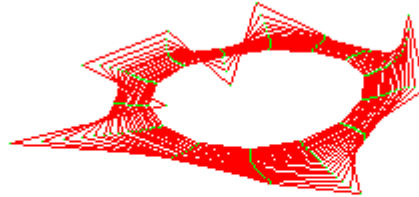
Contours en tant que maxima de la norme du gradient:

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

Répulsion d'une zone:

$$E_{\text{ext}}(\vec{x}) = \frac{\lambda}{1 + |\vec{x} - \vec{\Omega}|^2}$$

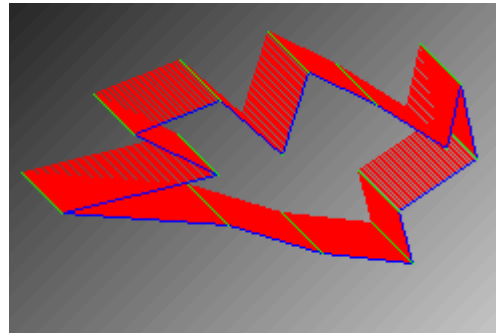
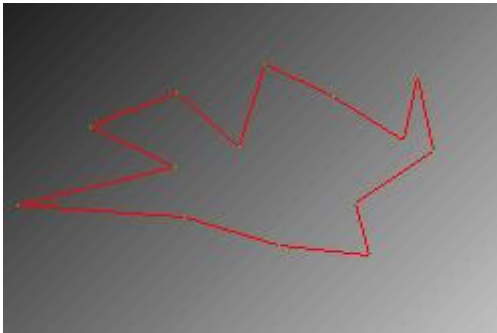
Illustrations



Energie *élastique*



Energie *de courbure*

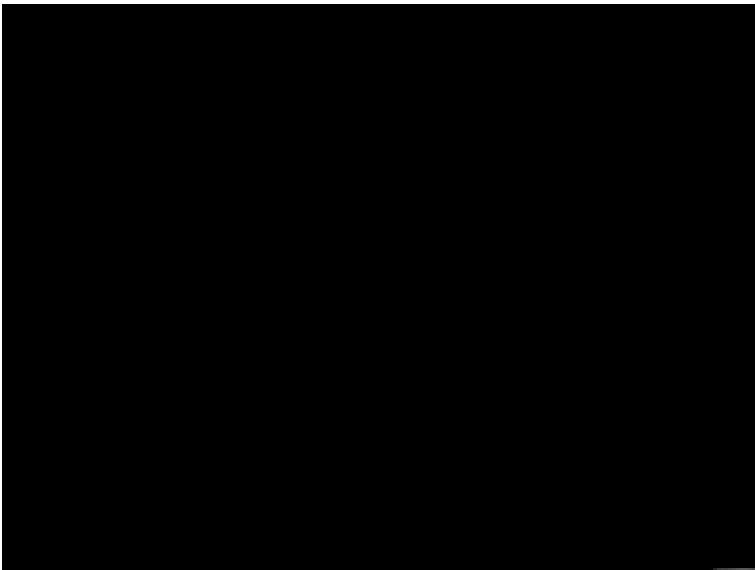


Zones brillantes

Condensation = **Conditional Density Propagation**
(*algorithme de type “contours actifs”*)

Algorithme probabiliste intégrant:

- des informations contextuelles (modèle observationnel adapté au problème)
- des connaissances *a priori*
- une intégration temporelle (tracking)



\mathbf{X} = espace de description des contours (\Rightarrow connaissances *a priori* sur la forme à segmenter)

\mathbf{Z} = espace de description des observations

Approche probabiliste bayésienne pour les contours actifs:

$$P(\mathbf{X} / \mathbf{Z}) = k \cdot P(\mathbf{X}) \cdot P(\mathbf{Z} / \mathbf{X})$$

Prior: modélisation
a priori des contours
possibles

Likelihood of X ou **modèle
observationnel:** modélisation
des observations attendues
pour un contour donné.

Choix de \mathbf{X} :

$$\hat{\mathbf{X}} = \arg \max_{\mathbf{X}} p(\mathbf{X} / \mathbf{Z})$$

Factored Sampling algorithm (approche du type « Filtrage Particulaire »):

Principe général: on construit itérativement un ensemble de points dans l'espace de recherche, tels que leur répartition corresponde à la densité de probabilité recherchée.

Un des avantages: modélisation utilisable pour des densités quelconques et pas uniquement gaussiennes.

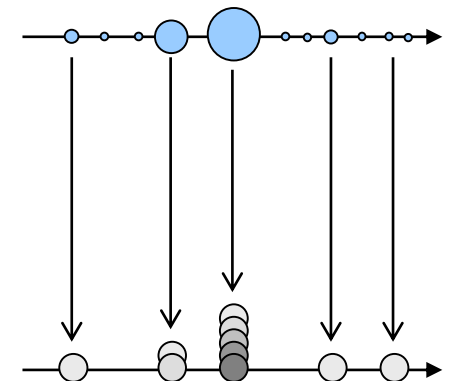
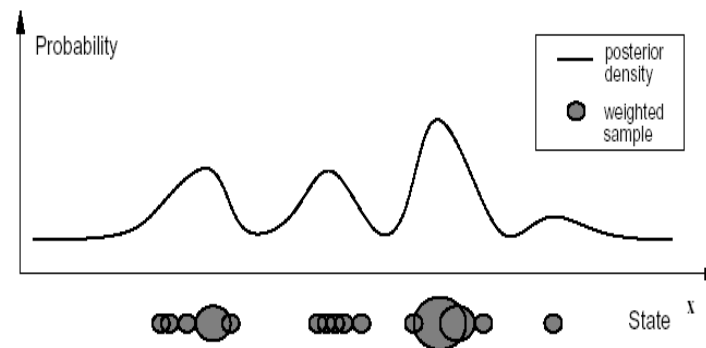
Exemple dans le cas de la segmentation par contours actifs:

1. A partir d'un prior $p'(X)$, générer une population de points $\{s^{(1)}, \dots, s^{(N)}\}$
2. Associer à chaque $s^{(i)}$ un poids donné par $\pi^{(i)}$ et échantillonner selon $(s^{(k)}, \pi^{(k)})$

$$\pi^{(i)} = \frac{p_z(s^{(i)})}{\sum_{j=1}^N p_z(s^{(j)})}$$

avec

$$p_z(s) = p(Z / X = s)$$



Condensation = **Conditional Density Propagation**

Entrée: l'échantillon au temps t_{k-1} soit : $\{ s_{k-1}^{(n)}, \pi_{k-1}^{(n)}, c_{k-1}^{(n)}, n=1, \dots, N \}$

Sortie: un nouvel échantillon au temps t_k $\{ s_k^{(n)}, \pi_k^{(n)}, c_k^{(n)}, n=1, \dots, N \}$

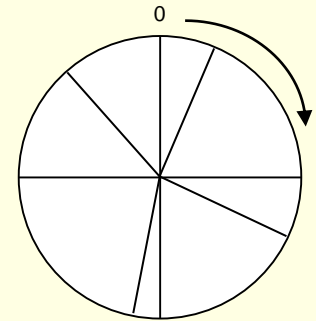


Itérer

Construction de la sortie:

1. **Sélectionner** un échantillon $s_k^{(n)}$ de la façon suivante:

- Générer un nombre aléatoire r dans $[0,1]$
- Trouver le plus petit j tel que $c_{k-1}^{(j)} \geq r$
- Choisir $s_k^{(n)} = s_{k-1}^{(j)}$



2. **Prédire** en échantillonnant sur $p(\mathbb{S}_k / \mathbb{S}_{k-1} = s_k^{(n)})$ pour choisir les $s_k^{(n)}$.

Par exemple, pour une dynamique Auto Regressive linéaire, utiliser:

$$s_k^{(n)} = A s_{k-1}^{(n)} + B w_k^{(n)} \quad \text{avec } w_k^{(n)} \text{ variable de bruit normal}$$

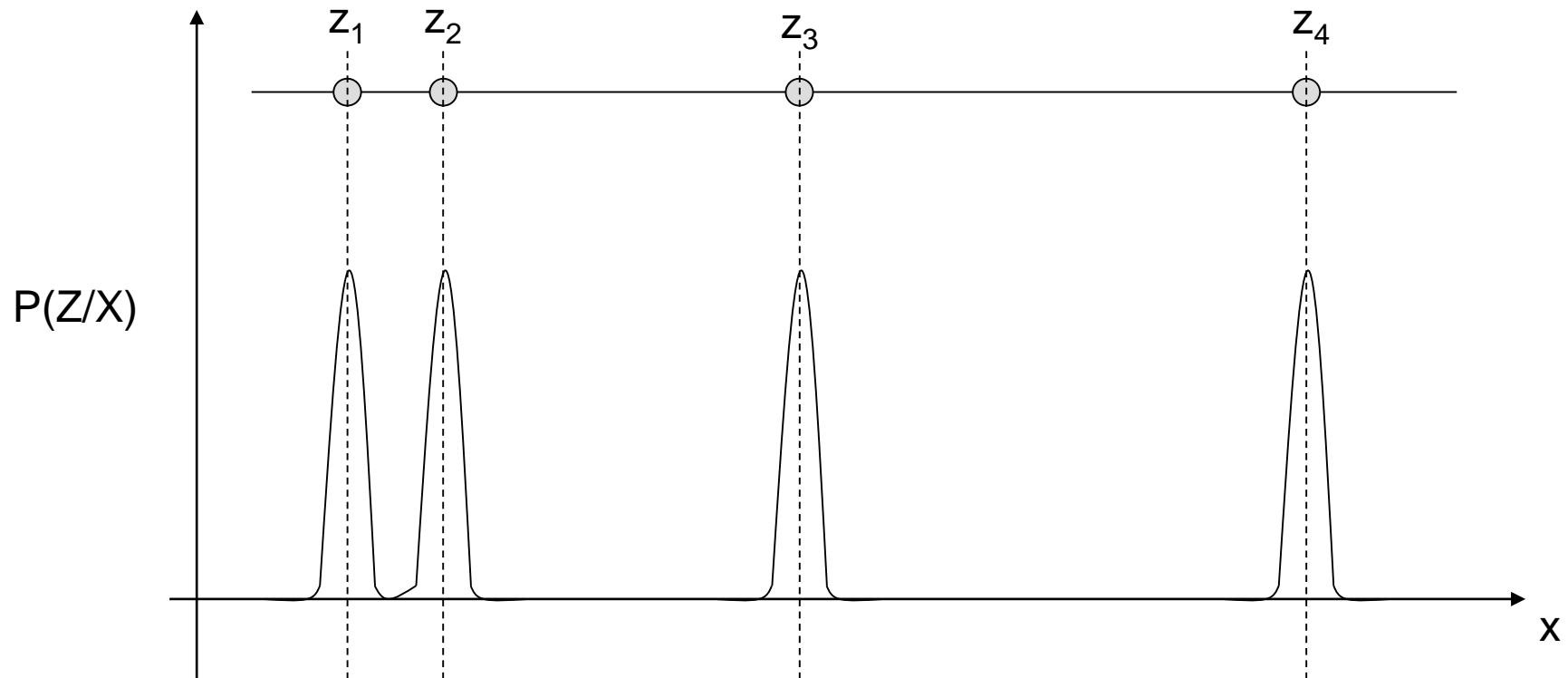
3. **Mesurer** et pondérer les nouveaux échantillons selon: $\pi_k^{(n)} = p(Z_k / \mathbb{S}_k = s_k^{(n)})$

Puis, normaliser de telle sorte que $\sum_n \pi_k^{(n)} = 1$ et stocker sous la forme $(s_k^{(n)}, \pi_k^{(n)}, c_k^{(n)})$

Avec

$$\begin{cases} c_k^{(0)} = 0 \\ c_k^{(n)} = c_k^{(n-1)} + \pi_k^{(n)}, n=1, \dots, N \end{cases}$$

Exemple simple de modèle d'observation, dans un cas monodimensionnel:



Condensation : **Schéma de fonctionnement**