

Segmentation : approches contours / régions

Master Informatique

Rachid JENNANE



POLYTECH[®]
ORLÉANS

École polytechnique de l'Université d'Orléans



Segmentation

- La segmentation des images consiste à regrouper les pixels de ces images qui partagent une même propriété pour former des régions connexes
- Deux approches :
 - Approche contour : les régions sont délimitées par les contours des objets qu'elles représentent (intuition, séparation)
 - Approche région : les régions sont déterminées en fonction de leurs propriétés intrinsèques (agrégation)



1 – Approche contours

- Principe (4 étapes)
 1. Mise en évidence des contours : différenciation de l'image
 2. Réduction de l'épaisseur des contours : l'épaisseur des contours doit être d'un pixel
 3. Binarisation des contours : au choix, réduction puis binarisation, ou inversement
 4. Description des contours : organisation des contours en structures simples telles que segments de droite, arcs de cercle, etc., fermeture des contours.



1.1 – Mise en évidence des contours

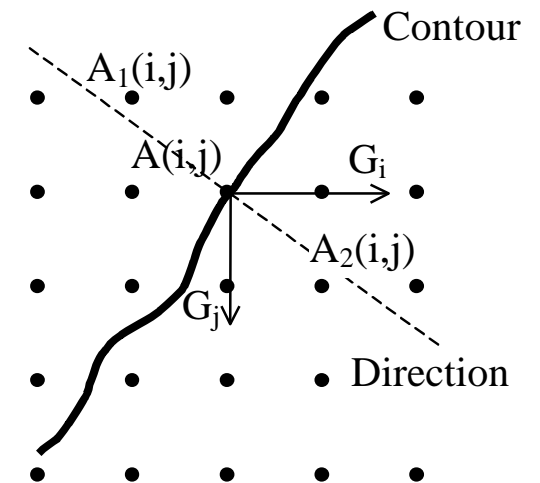
- Opérateurs de différenciation
 - Filtres passe-haut
 - Dérivées, Laplaciens
 - Morphologie mathématique, gradient morphologique
 - Réduction du bruit par filtrage passe-bas
- Contours \Rightarrow séparation (segmentation) des divers éléments de l'image en régions connexes supposées de même propriétés



1.2 – Réduction des contours

Méthode de suppression des points non maximaux (Canny) pour « amincir » les lignes de contour

1. Soient $A(i,j)$ et $D(i,j)$ les tableaux des amplitudes et directions des gradients (on dit aussi cartes des amplitudes et directions)
2. Pour chaque point $A(i,j)$, on détermine les deux points adjacents $A_1(i,j)$ et $A_2(i,j)$ qui sont dans la direction du gradient
3. Si $A(i,j)$ est supérieur à la fois à $A_1(i,j)$ et $A_2(i,j)$, alors $A(i,j)$ est conservé. Sinon, $A(i,j)$ est annulé

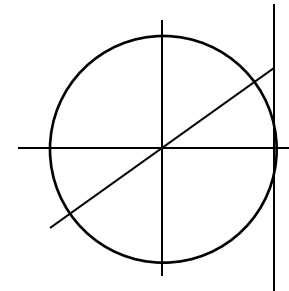


Pour les images binaires, squelettisation



1.3 – Amplitudes et directions du gradient

- Amplitude du gradient
 - $A(i,j) = [A_i(i,j)^2 + A_j(i,j)^2]^{1/2} \approx |A_i(i,j)| + |A_j(i,j)| \approx \text{Max}[A_i(i,j), A_j(i,j)]$
- Direction du gradient
 - $D(i,j) = \arctan(A_x(i,j)/A_y(i,j))$
 - En C/C++ :
 - $-\pi/2 \leq D[i,j] = \text{atan}(A_i[i,j]/A_j[i,j]) \leq +\pi/2$
 - $-\pi \leq D[i,j] = \text{atan2}(A_i[i,j], A_j[i,j]) \leq +\pi$
 - Formules de changement d'échelle
 - $-\pi \leq D[i,j] \leq +\pi \rightarrow 0 \leq (D[i,j] + \pi)/(2 \cdot \pi) \cdot 255 \leq 255$
 - $0 \leq D[i,j] \leq 255 \rightarrow -\pi \leq D[i,j]/255 \cdot (2 \cdot \pi) - \pi \leq +\pi$



1.3 – Binarisation des contours

- La binarisation des contours permet d'éliminer et/ou conserver certains contours
 - La segmentation par extraction de contours est efficace sur des images contrastées
 - Dans le cas contraire, il peut y avoir perte d'une partie du contour lors de la binarisation
- ⇒ Seuillage adaptatif : le choix du seuil est fonction de la direction du gradient pour pallier les distorsions d'amplitude



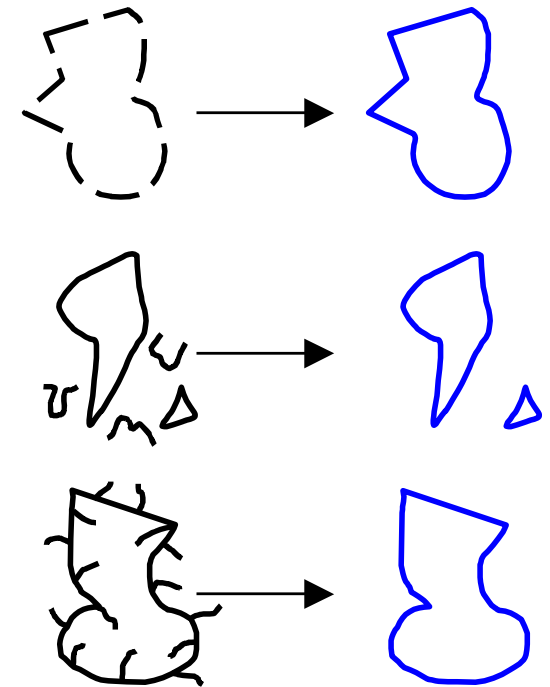
1.3 – Seuillage adaptatif

- Principe
 - Choix d'un seuil initial
 - Détermination de la réponse de l'opérateur de dérivation
 - Multiplication du seuil initial par un coefficient pondérateur calculé à partir de la réponse de l'opérateur de dérivation suivant la direction trouvée
 - Seuillage avec la valeur pondérée du seuil initial



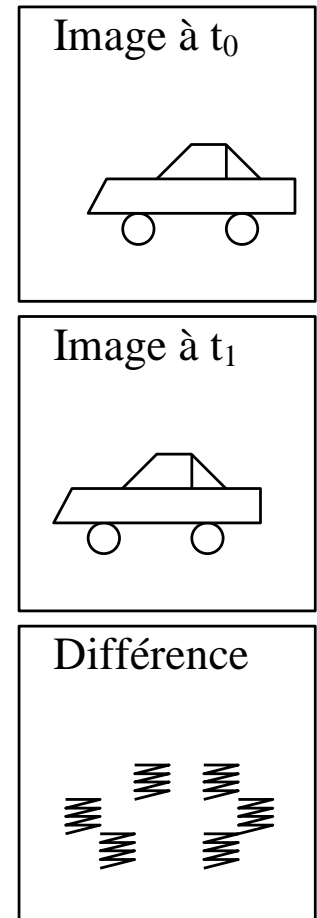
1.4 – Description des contours

- Post-traitements sur les contours
 - Fermeture des contours par extrapolation
 - Suppression des contours non fermés
 - Suppression des branches pendantes des contours fermés
- Codage
- Reconnaissance



1.5 – Objets en mouvement

- Il est possible de suivre le contour d'objets en mouvement dans une séquence d'images
- Principe
 - La différence entre des images acquises à des instants proches fait apparaître les contours perpendiculaires au déplacement (ainsi que direction, sens et vitesse du mouvement)
- Inconvénients
 - Sensibilité au type de mouvement
 - Sensibilité à la cadence d'acquisition et à la vitesse du déplacement
- Avantage
 - Méthodes très utilisées lorsque les paramètres précédents sont contrôlés



2 – Approche régions

- Contrairement à l'extraction des contours qui s'intéresse aux bords des régions, la segmentation en régions homogènes vise à segmenter l'image en se basant sur des propriétés intrinsèques des régions



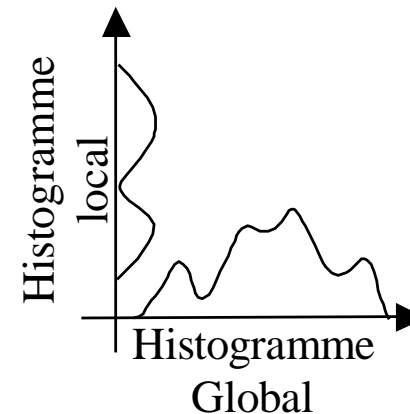
2.1 – Segmentation par seuillage

- Principe
 - Comparer la valeur de chaque pixel de l'image à un ou plusieurs seuils de manière à attribuer chaque pixel à une certaine classe
- Problème
 - Trouver les $L-1$ seuils S_l qui permettent de classer les pixels de l'image en L classes disjointes
- Remarque
 - La segmentation par seuil segmente les images indépendamment de la position des pixels (classement des pixels indépendamment de leur position)
- Trois approches : globale, locale, hybride



2.1 – Segmentation par seuillage

- Approche globale
 - Les seuils utilisés ne dépendent que d'une mesure globale calculée sur toute l'image \Rightarrow Histogramme 1D sur l'image
- Approche locale
 - Les seuils dépendent d'une mesure locale sur une portion de l'image \Rightarrow Histogramme 1D sur des portions de l'image
- Approche hybride
 - Les seuils dépendent d'une mesure à la fois globale et locale \Rightarrow Histogramme 2D



2.2 – Segmentation par division

- Segmentation par division [Split]
- Principe
 - Définition d'un critère d'homogénéité
 - Test de la validité du critère sur l'image
 - Si le critère est valide, l'image est segmentée [arrêt de la méthode]
 - Sinon, l'image est découpée en zones plus petites et la méthode est réappliquée sur chacune des zones

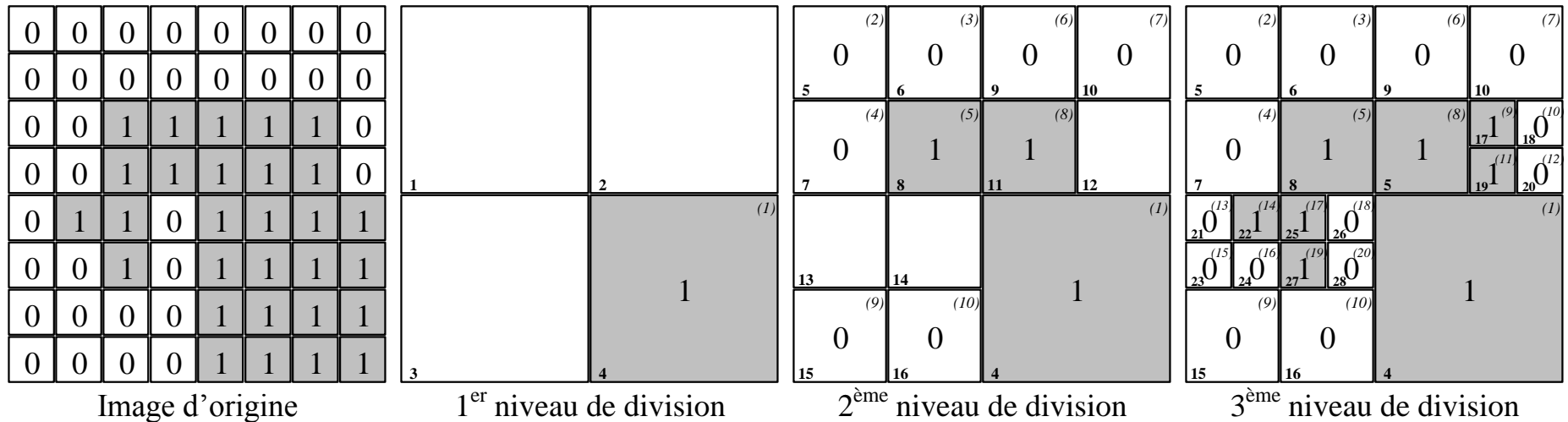


2.2 – Segmentation par division

- Paramètres de la segmentation par division
 - Critère d'homogénéité : forme *a priori* de l'histogramme, *extrema* de l'image (valeurs minimum et maximum), valeurs identiques, variance limitée, approximation par un polynôme de degré N, etc.
⇒ définition d'erreurs, de seuils ou d'intervalles de validité
 - Matière : niveaux de gris, couleurs, textures, etc.
 - Décomposition de l'image : division en 4, en 6, en polygones, etc.



2.2 – Segmentation par division



- Tant que tous les pixels ne sont pas identiques dans la région, celle-ci est divisée en 4
- Le numéro de division apparaît en bas à gauche
- Le numéro de région apparaît en haut à droite
- Quel est le critère de division ? Est-il unique ? Que penser de la division obtenue ?



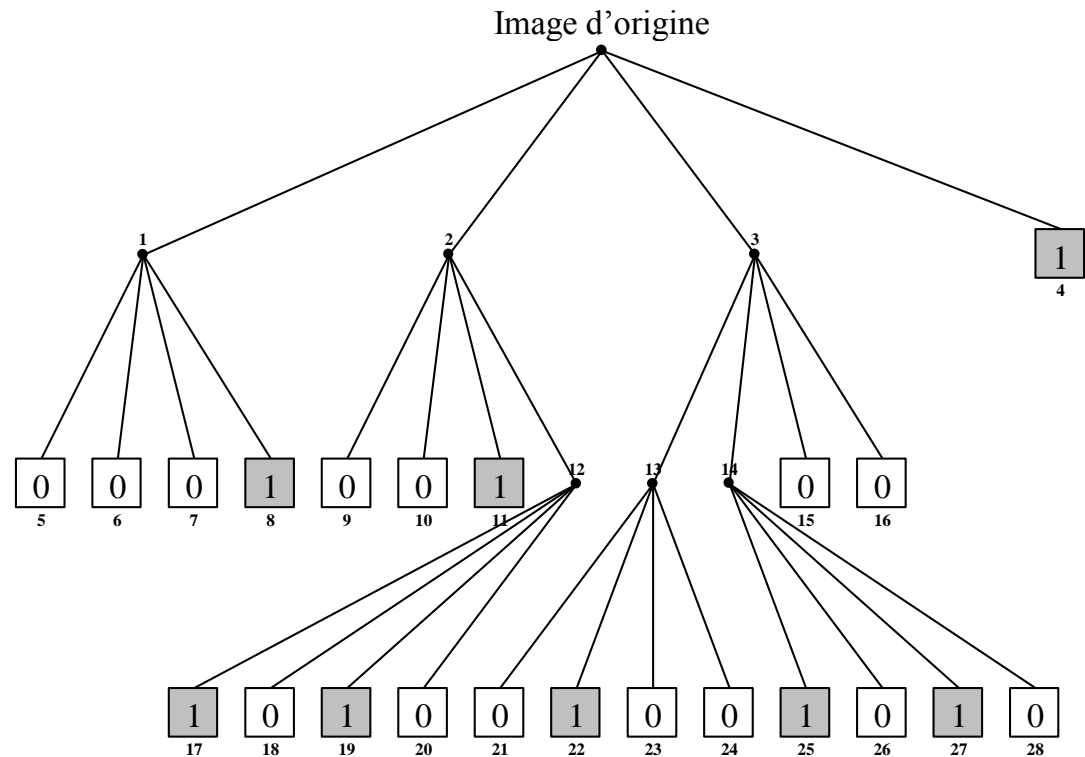
2.2 – Mise en évidence des régions

- La segmentation en régions ne doit pas être systématiquement associée à une représentation en image (informations sur la structure de l'image)
 - Néanmoins, il est courant de visualiser des images de régions. Il faut pour cela :
 - Affecter une couleur à chacune des régions
 - Attribuer des niveaux très différents à deux régions adjacentes (pour pouvoir visualiser cette différence)
 - Prendre en compte des nombres de régions supérieurs à 256
- ⇒ Assigner aux régions des niveaux aléatoires entre 0 et 255



2.2 – Représentation par arbre

- ⇒ Structure hiérarchisée des régions
- Une région qui satisfait au critère forme un nœud terminal, ou feuille de l'arbre
 - Une région qui ne satisfait pas au critère forme un nœud d'où partent 4 branches (division par 4) correspondant aux 4 zones sur lesquelles l'algorithme est ré-appliqué



2.2 – Exemple de critère d'homogénéité



Image d'origine
Division
réursive
d'une image
en quadrants

Image reconstruite à l'aide de poly-
nômes 2D de degré 0 (8317 régions)

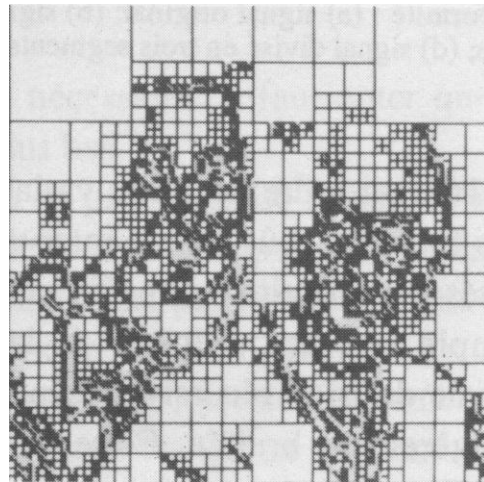
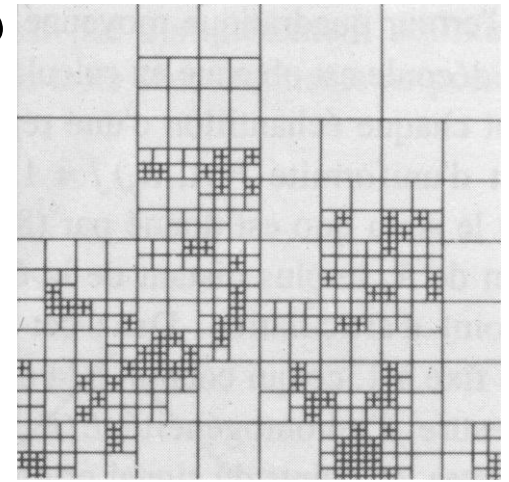


Image reconstruite à l'aide de poly-
nômes 2D de degré 3 (583 régions)



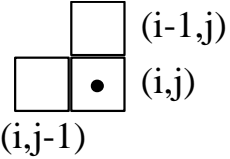
2.3 – Segmentation par rassemblement

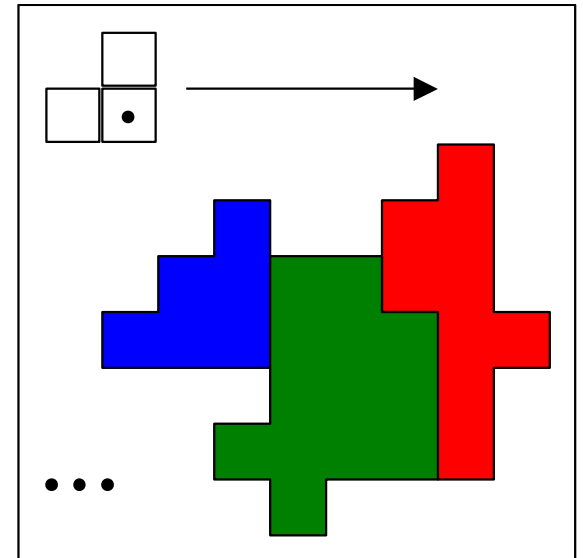
- Segmentation par rassemblement [Merge]
- Principe
 - On explore l'image à partir de petites régions
 - On fait croître celles-ci si elles satisfont à un critère d'homogénéité ou de regroupement
- Paramètres
 - Choix du critère d'homogénéité : différence de niveau de gris moyen, valeurs similaires, etc.
 - Critère d'arrêt



2.3 – Segmentation par rassemblement

Exemple 1 : méthode locale itérative : coloration de taches [blob coloring]

- Soit la fenêtre suivante :

- Chaque couleur représente un index qui caractérise les régions, déterminées selon l'algorithme suivant
- Inconvénients :
 - Segmentation uniligne.
 - Parcours déterminé *a priori*



2.3 – Segmentation par rassemblement

- Algorithme de coloration de taches

Pour chaque pixel $I(i,j)$ *Faire*

Si Critère $(i,j) = \text{Critère}(i-1,j)$

Alors Couleur $(i,j) \leftarrow \text{Couleur}(i-1,j)$

Sinon *Si* Critère $(i,j) = \text{Critère}(i,j-1)$

Alors Couleur $(i,j) \leftarrow \text{Couleur}(i,j-1)$

Sinon Couleur $(i,j) \leftarrow \text{NouvelleCouleur}$

Si (Critère $(i,j) = \text{Critère}(i-1,j)$) ET (Critère $(i,j) = \text{Critère}(i,j-1)$)

Alors Fusionner les régions en donnant la même couleur aux pixels $I(i,j)$, $I(i-1,j)$ et $I(i,j-1)$

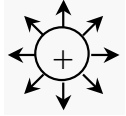
FinPour



2.3 – Segmentation par rassemblement

Exemple 2 : méthode locale récursive

- Principe : on fait croître une région avant de passer à la suivante, sans parcours particulier déterminé *a priori* (méthode par agrégation libre de pixels)

- Germe [seed] : 
- Croissance suivant un critère de similarité
- Critère d'arrêt : convexité maximum, etc.

- Inconvénients

- Méthode récursive \Rightarrow risques de débordements (pile)
- Influence de la position initiale du germe



2.3 – Segmentation par rassemblement

- Algorithme de la méthode locale récursive

Pour chaque pixel $I(i,j)$ *Faire*

Si $I(i,j)$ n'a pas déjà été traité

Alors Sauvegarder (i,j) , Croissance (i,j) , Incrémenter Région

FinPour

Croissance (i,j)

Pour tout Pixel (k,l) adjacent à $I(i,j)$ *// Pour* tous les 8-pixels

Si (Pixel (k,l) pas déjà traité) ET

 (Critère (Pixel (k,l)) = Critère $(I(i,j))$)

Alors Croissance (k,l)

FinPour



2.4 – Division et rassemblement

- Constat
 - La segmentation par division fournit une structure hiérarchisée qui permet d'établir des relations de proximité entre les régions, mais qui peut fractionner une même région en plusieurs ensembles distincts
 - La segmentation par fusion produit un nombre minimal de régions connexes, mais fournit celles-ci dans une structure horizontale qui n'indique pas de relation de proximité



2.4 – Division et rassemblement

- Proposition
 - **Rassembler**, à partir de la division grossière obtenue par **division**, les différents blocs adjacents de l'image
- ⇒ Algorithme de division et rassemblement, aussi appelé algorithme Split and Merge



2.5 – Mesures d'inhomogénéité

- Inhomogénéité régionale

$$E(R_1) = \frac{1}{N_1} \sum_{n=0}^{N_1-1} [I(n) - \mu(R_1)]^2$$

- N_1 : nombre de pixels dans la région R_1 ($N_1 = \text{Card}(R_1)$)
- $I(n)$: valeur des pixels n ($0 \leq n < N_1$) de la région R_1
- $\mu(R_1)$: valeur moyenne des pixels de la région R_1

$$\mu(R_1) = \frac{1}{N_1} \sum_{\substack{n=0 \\ n \in R_1}}^{N_1-1} I(n)$$



2.5 – Prédicats d'uniformité

- Prédicat d'uniformité

$$P[A, R_1] = \begin{cases} \text{Vrai} & \text{si } E(R_1) < T_1 \\ \text{Faux} & \text{sinon} \end{cases}$$

- $P[A, R_1]$: prédicat d'uniformité de la région R_1 [la région R_1 est considérée uniforme si le prédicat d'uniformité est vrai]
- $E(R_1)$: valeur d'inhomogénéité de la région R_1
- T_1 : seuil de tolérance sur l'inhomogénéité de la région R_1



2.5 – Mesures d'inhomogénéité

- Inhomogénéité interrégionale

$$E(R_1, R_m) = \frac{1}{N_1 + N_m} \sum_{n=0}^{N_1-1+N_m-1} [I(n) - \mu(R_1, R_m)]^2$$

- N_1, N_m : nombre de pixels dans les régions R_1 et R_m
- $I(n)$: valeur des pixels n ($0 \leq n \leq N_1-1+N_m-1$) des régions R_1 et R_m
- $\mu(R_1, R_m)$: moyenne des pixels de la région $R_1 \cup R_m$

$$\mu(R_1, R_m) = \frac{1}{N_1 + N_m} \sum_{\substack{n=0 \\ n \in R_1 \cup R_m}}^{N_1-1+N_m-1} I(n)$$



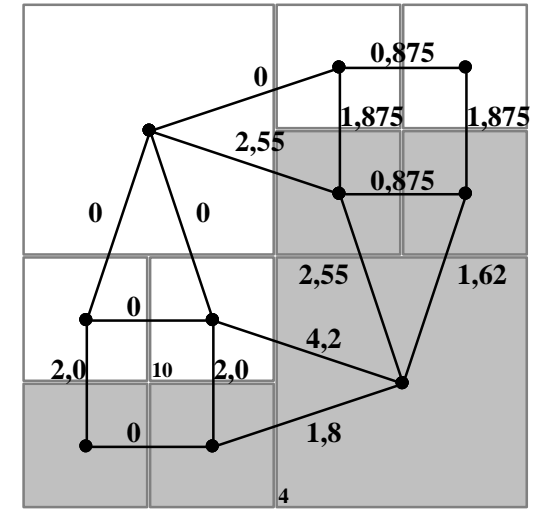
2.5 – Graphes de contiguïtés

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

Image d'origine

$\mu=0,0$		$\mu=0,0$	$\mu=0,25$
		$\mu=0,75$	$\mu=1,0$
$\mu=0,0$	$\mu=0,0$	$\mu=0,875$	
	10		
$\mu=1,0$	$\mu=1,0$		
		4	

Moyenne des régions



Graphe des contiguïtés

- L'image d'origine est segmentée en 10 régions
- Sur le graphe de contiguïtés des régions, la valeur $E(R_i, R_j)$ est indiquée pour chaque lien
- $\mu(R_i)$ représente la moyenne des pixels dans chaque région
- Exemple de calcul : $E(R_4, R_{10})$
 $\mu(R_4, R_{10}) = 14/20$
 $E(R_4, R_{10}) = 6(0-\mu)^2 + 14(1-\mu)^2 = 4,2$



2.5 – Critères de fin de rassemblement

- Critère de fin de rassemblement des régions

$$SEQ(L) = \sum_{i=0}^{L-1} \sum_{\substack{n=0 \\ n \in R_i}}^{N_i-1} [I(n) - \mu(R_i)]^2$$

- SEQ(L) : somme des erreurs quadratiques
- L : nombre de nœuds du graphe \equiv nombre de régions de l'image à l'itération considérée
- I(n) : valeur du pixel au point n
- $\mu(R_i)$: moyenne des pixels dans la région R_i
- Critère d'arrêt
 - Si $SEQ(L) > T_2$, seuil fixé *a priori*



2.5 – Rapport de vraisemblance

- Soient deux régions R1 et R2 de N1 et N2 pixels
- Hypothèse 1 : les deux régions appartiennent au même objet \Rightarrow les intensités des pixels suivent une même distribution Gaussienne (μ_0, σ_0)
- Hypothèse 2 : les deux régions appartiennent à deux objets différents \Rightarrow chaque région possède une distribution Gaussienne différente (μ_1, σ_1) et (μ_2, σ_2)

- Avec $p[I(n)] = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{[I(n)-\mu]^2}{2\sigma^2}}$ on a $\hat{\mu} = \frac{1}{N} \sum_{n=0}^{N-1} I(n)$ $\hat{\sigma} = \frac{1}{N} \sum_{n=0}^{N-1} [I(n) - \hat{\mu}]^2$



2.5 – Rapport de vraisemblance

Avec l'hypothèse H_1 , la densité de probabilité conjointe est

$$p[I(0), I(1), \dots, I(N_1 - 1 + N_2 - 1) | H_1] = \prod_{n=0}^{N_1-1+N_2-1} p[I(n) | H_1]$$
$$= \prod_{n=0}^{N_1-1+N_2-1} \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{[I(n)-\mu_0]^2}{2\sigma_0^2}} = \frac{1}{(\sqrt{2\pi}\sigma_0)^{N_1+N_2}} e^{-\frac{\sum_{i=0}^{N_1+N_2-2} [I(n)-\mu_0]^2}{2\sigma_0^2}}$$

$$= \frac{1}{(\sqrt{2\pi}\sigma_0)^{N_1+N_2}} e^{-\frac{N_1+N_2}{2}}$$



2.5 – Rapport de vraisemblance

Avec l'hypothèse H_2 , la densité de probabilité conjointe est

$$p[I(0), I(1), \dots, I(N_1 - 1 + N_2 - 1) | H_2] = \prod_{n=0}^{N_1-1+N_2-1} p[I(n) | H_2]$$

$$= \prod_{n=0}^{N_1-1} \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{[I(n)-\mu_1]^2}{2\sigma_1^2}} \cdot \prod_{n=0}^{N_2-1} \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{[I(n)-\mu_2]^2}{2\sigma_2^2}}$$

$$= \frac{1}{(\sqrt{2\pi}\sigma_1)^{N_1}} e^{-\frac{N_1}{2}} \cdot \frac{1}{(\sqrt{2\pi}\sigma_2)^{N_2}} e^{-\frac{N_2}{2}}$$



2.5 – Rapport de vraisemblance

Le rapport de vraisemblance L est défini par :

$$L = \frac{p[I(0), I(1), \dots, I(N_1 - 1 + N_2 - 1) | H_1]}{p[I(0), I(1), \dots, I(N_1 - 1 + N_2 - 1) | H_2]} = \frac{\sigma_0^{N_1 + N_2}}{\sigma_1^{N_1} \cdot \sigma_2^{N_2}}$$

$\sigma_0, \sigma_1, \sigma_2$ sont estimés par la relation $\hat{\sigma} = \frac{1}{N} \sum_{n=0}^{N-1} [I(n) - \hat{\mu}]^2$

- Si $L > \text{seuil}$, les deux régions sont distinctes
- A l'inverse, si $L \leq \text{seuil}$, les deux régions doivent être regroupées



2.6 – Segmentation multirésolution

- Un grand nombre de propriétés d'images peut être calculé à l'aide d'opérateurs appliqués localement
- Ex : statistique de niveau de gris, propriétés texturales, discontinuités locales
- Propriétés texturales:
 - Résolution \Rightarrow sélection des textures
 - Micro-textures \Rightarrow segments isolés
- Parade : multi-résolution pyramidale

