



Cours Magistral

Traitement Numérique de l'Image

Partie 1

Adrien LAROUSSE

2A - 2023

20 novembre 2023

Table des matières

1	Pré-traitement	3
1.1	Traitement au niveau du pixel	3
1.1.1	Quelques définitions	3
1.1.2	<i>Look Up Table (LUT) ou Table de Transfert</i>	4
1.1.3	Re-Quantification	6
1.1.4	Binarisation - Seuillage	6
1.1.5	Egalisation d'histogramme	7
1.1.6	Opérations	8
1.2	Traitement aux environs du pixel	8
1.2.1	Filtrage spatial	8
1.2.2	Filtre linéaire	8
1.2.3	Filtre basse-bas : exemples de bruits	9
1.2.4	Filtre moyenneur	9
1.2.5	Filtrage Passe-haut	10
1.2.6	Filtre non-linéaire	11
1.2.7	Filtre non-linéaire : médian	12
1.2.8	Filtre non-linéaire : Morphologie mathématique	12
1.2.9	Morphologie mathématique - Opérations géodesiques	18
1.2.10	Morphologie mathématique - Squelette	18
2	Segmentation et Extraction d'informations	20
2.1	Généralités	20
2.2	Approche contours	21
2.3	Approche Classification de pixels	22
2.3.1	Classification sur l'histogramme	23
2.3.2	Codage d'un contour	23
3	Interprétation, Décision et Reconnaissance de formes	26
3.1	Clustering vs Classification	26
3.2	Clustering	26
3.2.1	C'est quoi le <i>clustering</i> ?	26
3.2.2	Deux types de <i>clustering</i>	27
3.3	Classification	29
3.3.1	Les différentes phases	29
3.3.2	Evaluation	30
3.3.3	Arbres de décision	32
3.4	Conclusion	32
4	TD1 : Traitements au niveau du pixel (LUT)	34
5	TD3 : Prétraitements - Filtrage	39
6	TD5 : Prétraitements - Morph. Mathématiques	42
7	TD6 : Segmentation, Interprétation & Décision	44

CHAPITRE

1

PRÉ-TRAITEMENT

1.1 Traitement au niveau du pixel

1.1.1 Quelques définitions

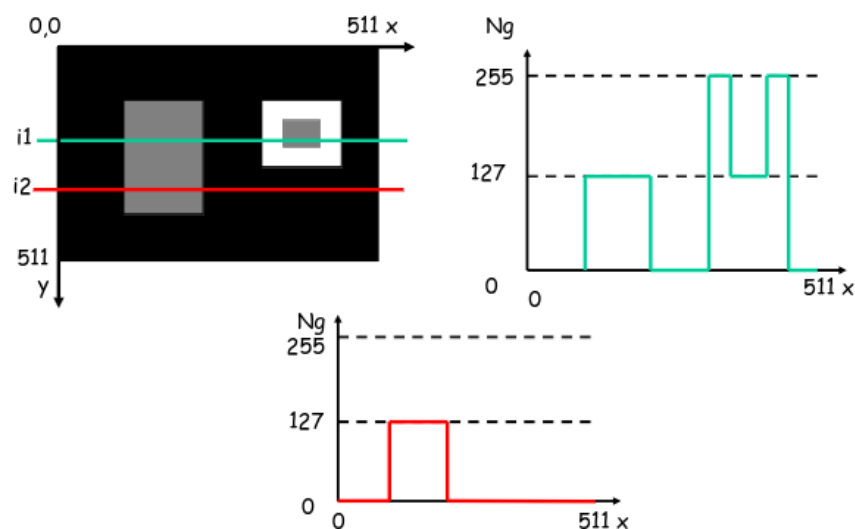


FIGURE 1.1 – Profil de ligne

Définition 1.1 : Contraste

$$C = \frac{\max[I(i, j)] - \min[I(i, j)]}{\max[I(i, j)] + \min[I(i, j)]}$$

On une autre définition possible est :

$$C = \Delta B \approx \frac{\Delta I}{I}$$

Définition 1.2 : Histogramme

Représente le nombre (ou la proportion) de pixels en fonction du niveaux de gris

- Distribution des valeurs de niveaux de gris dans une image.

Définition 1.3 : Histogramme cumulé

$HC(x)$ représente le taux de pixel dont le niveau de gris est inférieur à x

$$HC(x) = \frac{\sum_{j=0}^x H(j)}{Nb_{colonne} \times Nb_{lignes}}$$

1.1.2 Look Up Table (LUT) ou Table de Transfert**Définition****Définition 1.4 : Look Up Table**

Une LUT est essentiellement une table de correspondance où chaque valeur d'entrée (par exemple, l'intensité d'un pixel dans une image en niveaux de gris) est associée à une valeur de sortie spécifique.

Algorithm 1 Table de transfert (LUT)

```

1:                                     ▷ Initialisation de la LUT à rampe croissante
2: for  $i = 0$  à 255 do
3:    $LUT[i] = i$ ;
4: end for
5:                                     ▷ Application de la LUT
6: for  $i = 1$  à  $nb$  de lignes do
7:   for  $j = 1$  à  $nb$  de lignes do
8:      $I'(i, j) = LUT(I(i, j))$ 
9:   end for
10: end for

```

avec :

- I : Niveau de gris en entrée
- I' : Niveau de gris en sortie

Les différentes LUT

- L'expansion de dynamique

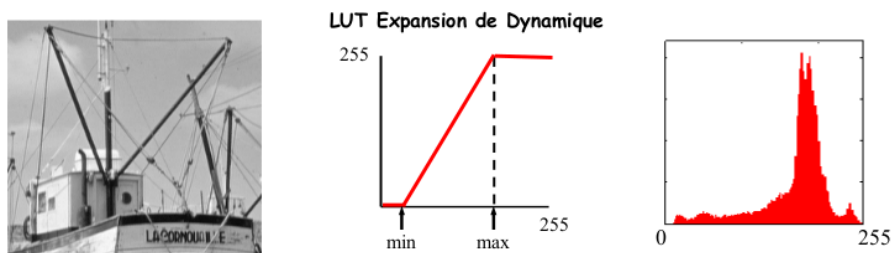


FIGURE 1.2 – Expansion de dynamique

- Transformation linéaire avec saturation

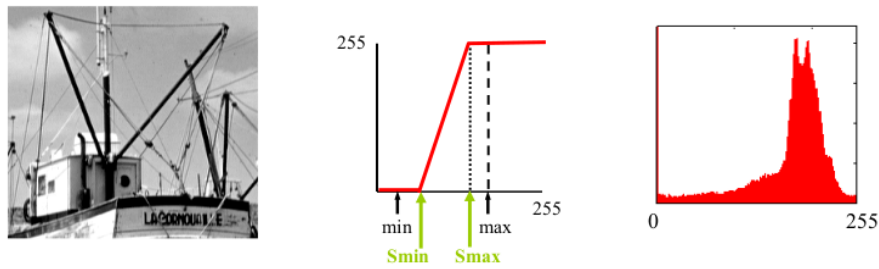


FIGURE 1.3 – Transformation linéaire avec saturation

- Transformation linéaire par morceaux

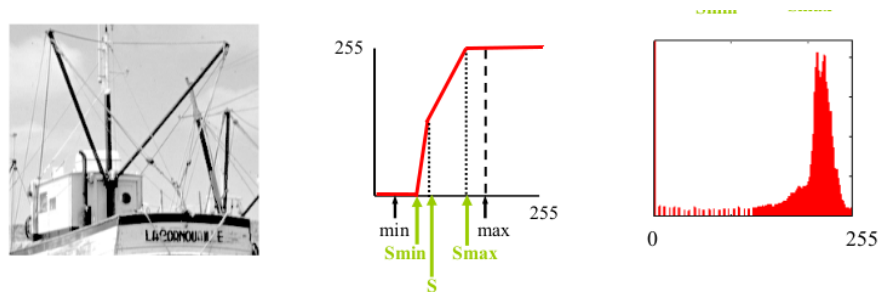


FIGURE 1.4 – Transformation linéaire par morceaux

- Il existe aussi des transformations non-linéaire

Algorithm 2 Expansion de dynamique

```

1:                                     ▷ Initialisation de la LUT à rampe croissante
2: for  $i = 0$  à 255 do
3:    $LUT[i] = 2^{NG} - 1 \times \frac{i - \min(I)}{\max(I) - \min(I)}$ ;
4: end for
5:                                     ▷ Application de la LUT
6: for  $i = 1$  à  $nb$  de lignes do
7:   for  $j = 1$  à  $nb$  de lignes do
8:      $I'(i, j) = LUT(I(i, j))$ 
9:   end for
10: end for

```

Proposition 1.1 : Niveau de gris en sortie

$$I'(i, j) = \frac{(2^{NG} - 1)}{\max - \min} \times (I(i, j) - \min)$$

avec $\frac{(I(i, j) - \min)}{\max - \min} \in [0, 1]$

1.1.3 Re-Quantification

Définition 1.5 : Re-Quantification

C'est la répartition des valeurs originales en N niveaux. Les valeurs de transition T_k et de reconstruction R_k sont données par :

$$T_k = (k-1) \times W \quad \text{avec } W = \frac{b-a}{N}$$

$$R_k = T_k \times \frac{N}{N-1} \quad \text{pour } k=1\dots N$$

L'intervalle $\frac{b-a}{N} = T_k - T_{k-1}$ est constant et est appelé l'intervalle de quantification.

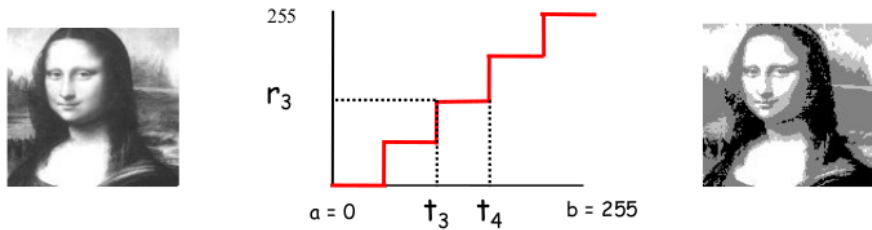


FIGURE 1.5 – Re-Quantification

Remarque 1.1

On peut faire des quantification non uniforme.

1.1.4 Binarisation - Seuillage

Définition 1.6 : Seuillage

$$I'(i, j) = 255 \text{ si } I(i, j) > \text{seuil}$$

$$I'(i, j) = 0 \text{ sinon}$$

On peut avoir un deuxième seuil pour avoir un encadrement.

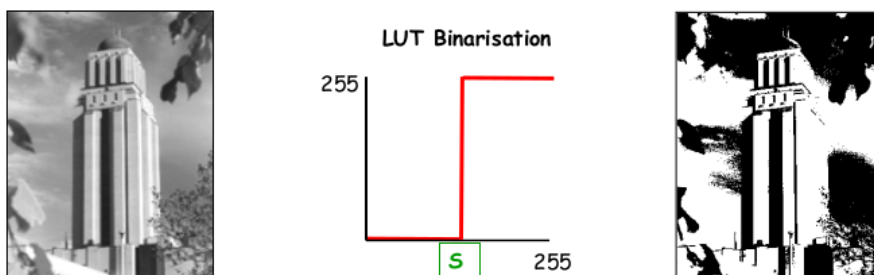
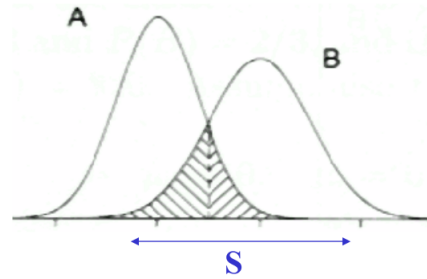


FIGURE 1.6 – Binarisation

- On balaie toutes les valeurs de seuil possible S
- Pour chaque seuil S :
 - ↪ On calcule les moyennes et les variances de chaque classe
 - ↪ On s'intéresse à la variance intra-classes



Moyennes : μ_1 et μ_2
 Variances : σ_1^2 et σ_2^2
 Variance Intra-classes :
 $\sigma_w^2 = P_1 \cdot \sigma_1^2 + P_2 \cdot \sigma_2^2$
 Le seuil optimal est celui qui donne σ_w minimum

- Basé sur le fait que les classes sont bien définies et regroupées

$$\sigma_1^2 = \frac{1}{S} \sum_{i=0}^{S-1} (H(i) - \mu_1)^2$$

$$\sigma_2^2 = \frac{1}{256-S} \sum_{i=S}^{255} (H(i) - \mu_2)^2$$

$$\mu_1 = \frac{1}{S} \sum_{i=0}^{S-1} h(i) \quad P_1 = \frac{1}{NbLig \cdot NbCol} \sum_{i=0}^{S-1} H(i)$$

$$\mu_2 = \frac{1}{256-S} \sum_{i=S}^{255} h(i) \quad P_2 = \frac{1}{NbLig \cdot NbCol} \sum_{i=S}^{255} H(i)$$

FIGURE 1.7 – Algorithme d'Otsu

Définition 1.7 : Exponential fit

On approxime l'histogramme par une exponentielle décroissante :

$$f(x) = \frac{1}{\mu} \times e^{-\frac{x}{\mu}}$$

μ est la moyenne de niveau de gris

Le seuil est calculé en fonction du pourcentage A de pixels attribué à l'objet :

$$\int_{x_s}^{255} f(x) dx = A \times \int_0^{255} f(x) dx$$

1.1.5 Egalisation d'histogramme**Définition 1.8 : Fonction de répartition**

Pour une variable aléatoire continue, et dans le cas courant Ω est un intervalle $[a, b] \subseteq \mathbb{R}$, on définit la notion de fonction de répartition F de X comme :

$$F(X) = P(X < x) = \int_a^x p_x(t) dt = HC(x)$$

En discret, la fonction de répartition est l'histogramme cumulé.

Définition 1.9 : Egalisation

L'égalisation revient à grouper par paquets de L éléments les intensités de l'image de départ. L'image par HC de deux niveaux de gris successifs est séparée par la même distance. L'image par HC de x_k , $k < \frac{N}{L}$ est dans l'intervalle $[0, \frac{1}{L}]$

Proposition 1.2 : Implémentation dans le cas discret

$$I'(i, j) = (2^{(NG-1)} - 1) \times \frac{HC(I(i, j))}{nb_l \times nb_c}$$

1.1.6 Opérations

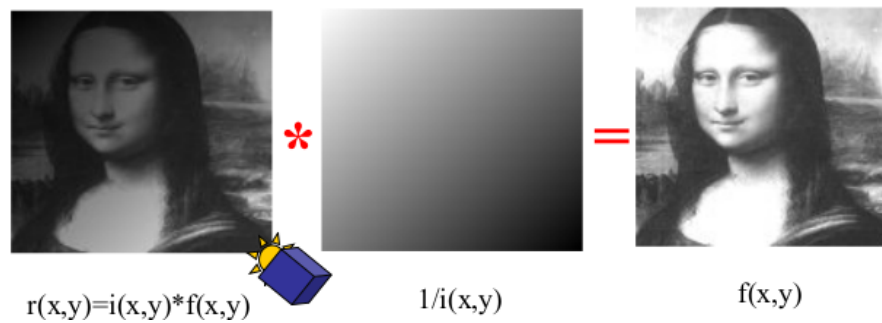
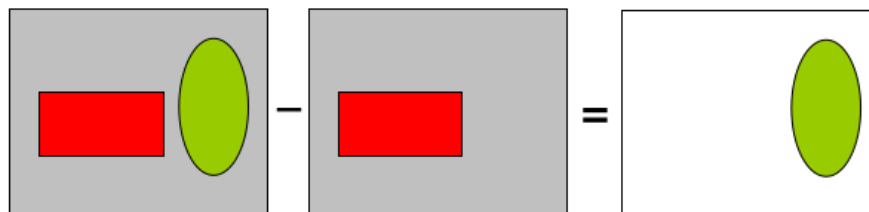


FIGURE 1.8 – Opérations logiques : + - * / ET OU XOR NON



$$| \text{ImageCourante} - \text{ImageRéférence} | > \text{Seuil}$$

FIGURE 1.9 – Soustraction d'images

1.2 Traitement aux environs du pixel

1.2.1 Filtrage spatial

Proposition 1.3 : Filtrage *linéaire* et Filtrage *non-linéaire*

- **Filtrage linéaire** : Convolution
 - Passe-bas : lissage, réduction du bruit
 - Passe-haut : accentuation des contours
- **Filtrage non-linéaire** : modèle ensembliste
 - médian
 - morphologie mathématique

1.2.2 Filtre linéaire

Définition 1.10 : Convolution discrète 2D

- Le domaine de l'image et le support de H sont bornés
- Pour un masque de taille $(2h_x + 1) \times (2h_y + 1)$ on a

$$(f \times h)(i, j) = \sum_{k=-h_x}^{+h_x} \sum_{l=-h_y}^{+h_y} f(i-k, j-l) h(k, l)$$

- La valeur transformée au pixel (i, j) dépend des valeurs de son environnement. Cette influence est décrite par le **masque** H

Remarque 1.2 : Masque ou Noyau de convolution

- Le noyau doit être symétrique pour que le filtre soit à phase nulle.
- Les masques sont une approximation discrète des réponses impulsionnelles (R.I.) des filtres continus

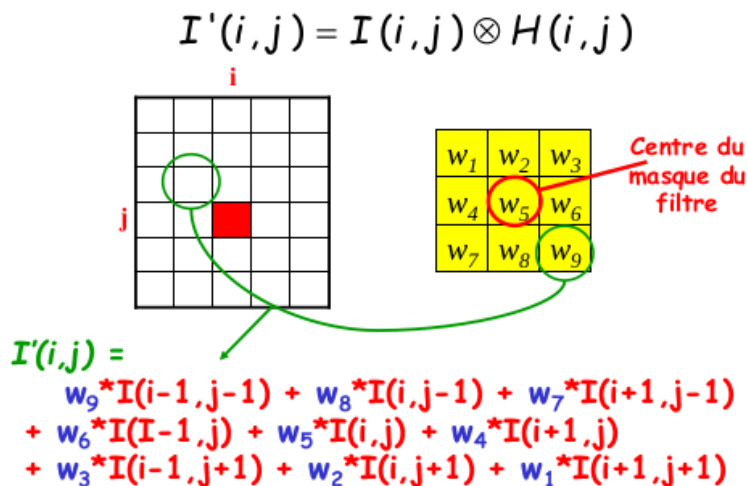


FIGURE 1.10 – Convolution discrète 2D

Remarque 1.3

Généralement, masque de dimension impaire et symétrique

1.2.3 Filtre basse-bas : exemples de bruits

Les filtres passe-bas permettent la réduction du bruit.

- Bruit **poivre et sel** d'ordre n : obtenu en ajoutant n pixels blancs et n pixels noirs aléatoirement dans une image.
- Bruit **Gaussien** : obtenu en ajoutant à chaque pixel une valeur aléatoire suivant une loi de probabilité Gaussienne.

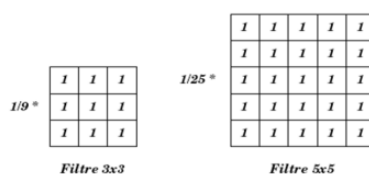
1.2.4 Filtre moyennneur

FIGURE 1.11 – Masques de convolution discrète d'un filtre moyennneur

Remarque 1.4

Plus le filtre grossit, plus le lissage devient important et plus le flou s'accroît

1.2.5 Filtrage Passe-haut

Objectif : Mettre en avant les détails de l'image, les variations brutales d'intensité.

- Filtres différentiels : détection de contours
 - Type de contours
 - Dérivée première : Gradient
 - Dérivée seconde : Laplacien
- Rehaussement de contraste

Notion de contours

Définition 1.11 : Contours

- Les contours se caractérisent par les discontinuités de la fonctions d'intensité de l'image
- Ils proviennent des discontinuités de la fonction de réflectance (ombre, texture) ou des discontinuités de profondeur (bord d'objets)

Proposition 1.4 : Paramètres caractérisant le contour

- son **orientation** θ par rapport à un axe de l'image.
- son **épaisseur** e mesurée perpendiculairement au contour.
- sa **longueur** L mesurée le long d'un contour.
- sa **hauteur** h exprimant la différence de luminance entre le niveau et le niveau bas.

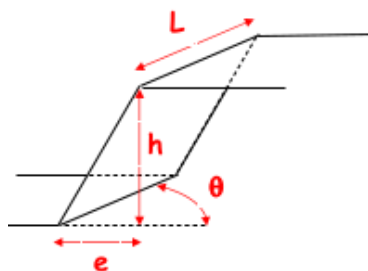


FIGURE 1.12 – Caractéristiques contour

Filtres différentiels

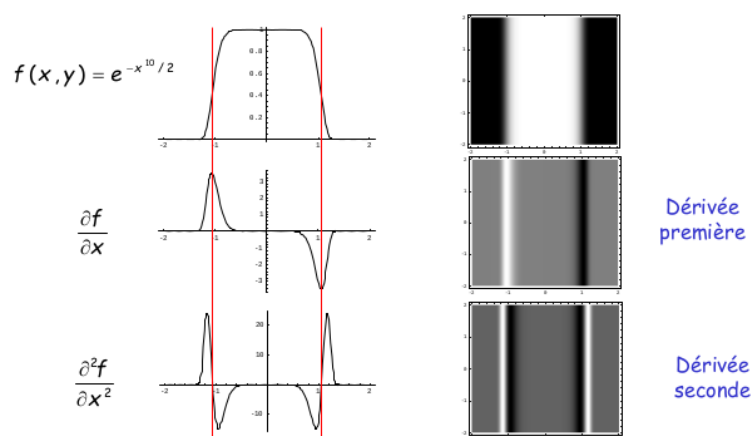


FIGURE 1.13 – Filtre différentiel

Masque classique

Définition 1.12 : Opérateur de Prewitt

Dérivée d'un opérateur de lissage de type moyenneur

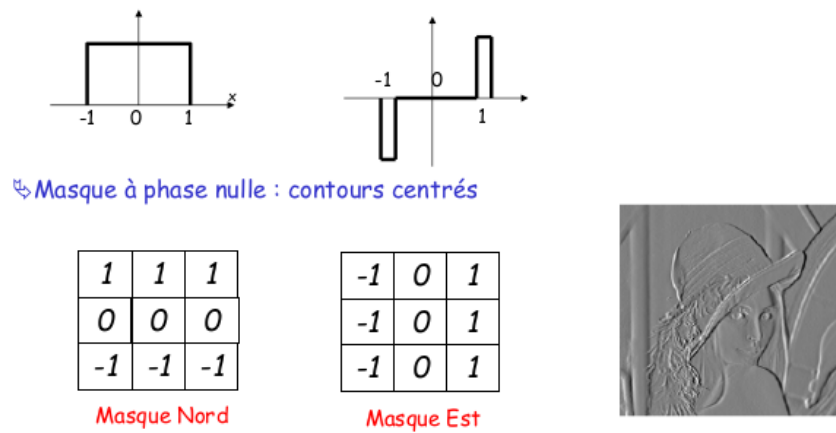


FIGURE 1.14 – Opérateur de Prewitt

Définition 1.13 : Opérateur de Sobel

Dérivée d'un opérateur de lissage de type Gaussien

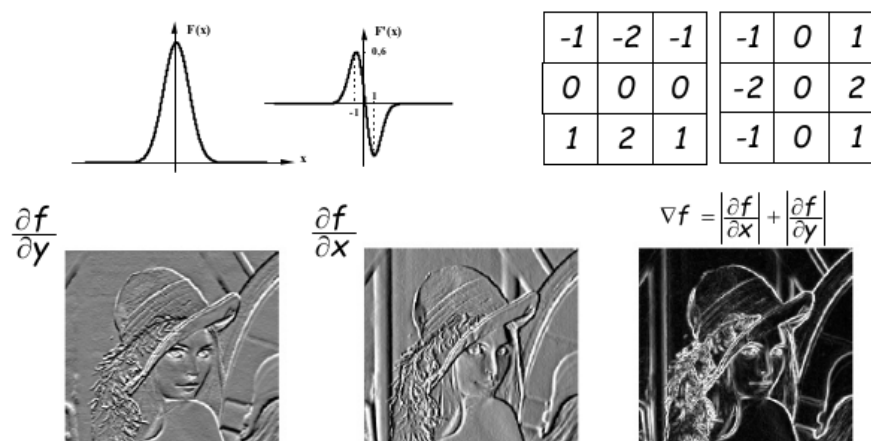


FIGURE 1.15 – Opérateur de Sobel

Filtres séparables

$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = D(y) * S(x) \text{ avec } D(y) = [1, 2, 1]^T \text{ et } S(x) = [-1, 0, 1]$$

→ Lissage en y et dérivée en x : moins sensible au bruit

FIGURE 1.16 – Exemple de séparation sur le noyau de Sobel

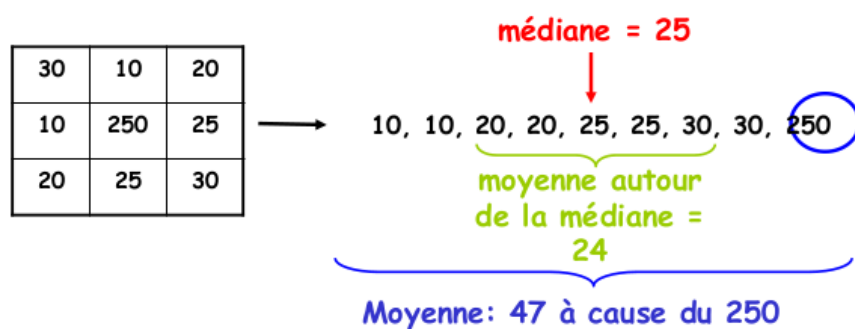
1.2.6 Filtre non-linéaire

- **Non-linéarité** : Les coefficients du masque ne sont pas indépendants de l'image

- **Filtres d'ordre** : ces filtres procèdent en remplaçant les valeurs de chaque pixel par la valeur qui occupe un certain rang lorsqu'on trie les valeurs observées dans un certain voisinage du pixel.
 - Médian
 - Erosion morphologique - Dilatation morphologique

1.2.7 Filtre non-linéaire : médian

➤ Principe de calcul :



☞ Particulièrement utile pour un bruit de type Poivre & Sel (0 et 255)

FIGURE 1.17 – Filtre médian

1.2.8 Filtre non-linéaire : Morphologie mathématique

Les transformations de la **morphologie mathématique** sont faites pour travailler sur des images binaires (0 : noir ; 1 : blanc) avec des objets blancs sur fond noir.

Proposition 1.5 : Principes

Extraire des informations de l'images à partir de réponses à des tests : **Transformations**
Objet de référence = élément structurant

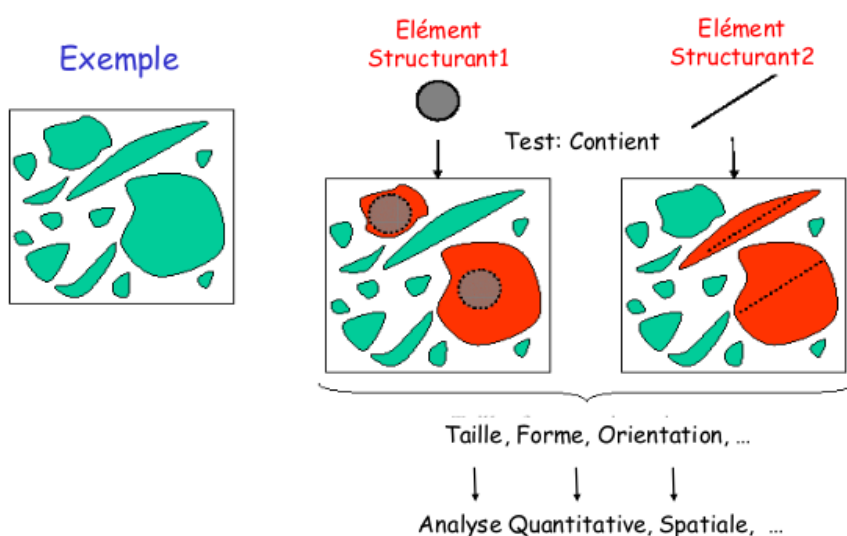


FIGURE 1.18 – Principe de la morphologie mathématique

Théorème 1.1 : Propriétés des opérateurs morphologiques

- **Croissance** : Une transformation Φ est croissante si

$$\forall (X, Y) \in E^2 \quad X \leq Y \Rightarrow \Phi(X) \leq \Phi(Y)$$

- **Extensivité** : Une transformation sur Φ est extensive si :

$$\forall X \in E \Rightarrow X \leq \Phi(X)$$

- **Anti-Extensivité** : Une transformation sur Φ est anti-extensive si :

$$\forall X \in E \Rightarrow X \geq \Phi(X)$$

- **Idempotence** : Une transformation Φ est idempotente si :

$$\forall X \in E \Rightarrow \Phi(\Phi(X)) = \Phi(X)$$

- **Dualité** : Deux transformation ϕ et Ψ sont duales si :

$$\forall X \in E \Rightarrow \Psi(X^C) = [\Phi(X)]^C$$

avec X^C qui désigne le complémentaire de X dans E : $X^C = E \setminus X$

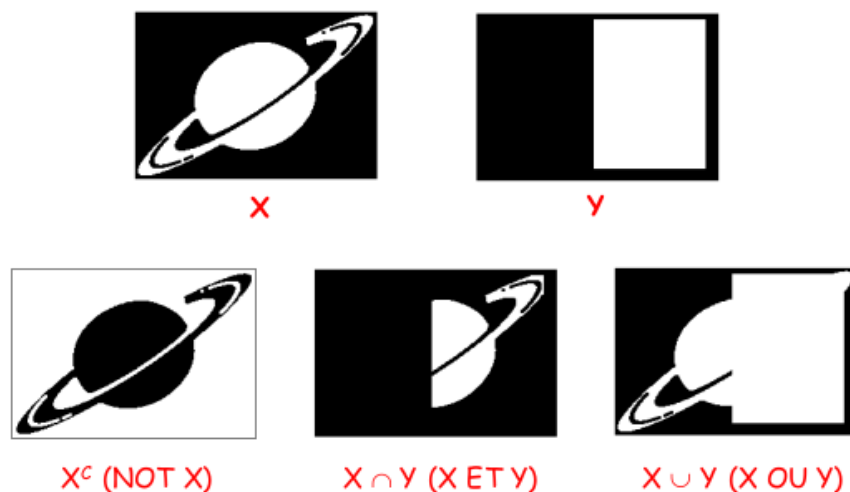


FIGURE 1.19 – Opérations de base

Définition 1.14 : Différence symétrique

$$Y \setminus X = (Y \cup X) - (Y \cap X)$$

Théorème 1.2 : Propriétés différence symétrique

- associative
- possède un élément neutre \emptyset
- commutative $X \setminus Y = Y \setminus X$

C'est quoi la **morphologie mathématique**? C'est la transformation d'une image binaire par un ensemble fixé, appelé **élément structurant**

Pour une transformation donnée, l'image finale est obtenue en translatant l'élément structurant sur l'image origine contenant un objet et en examinant si la relation entre l'objet et l'élément structurant est vérifiée. **Mise en évidence de propriétés locales**

L'élément structurant élémentaire dépend de la maille utilisée. Sur une maille carrée, on trouve deux éléments structurant élémentaire en fonction du voisinage choisi :



FIGURE 1.20 – Élément structurant élémentaire

L'élément structurant a une taille λ et une origine o . L'origine de l'élément structurant est appelé **centre**.

Effectuer une transformation de **taille** n correspond à utiliser un élément structurant de taille n ou à appliquer n fois la même transformation de taille 1.

Dilatation

Définition 1.15 : Dilatation

La dilatation morphologique d'un ensemble X par B est définie comme l'addition de Minkowski de X avec le transposé de B par rapport à son centre :

$$\delta_B(X) = X \oplus B^T = \{x + y / x \in X, y \in B^T\}$$

Exemple 1.1 : Element structurant symétrique

Si l'élément structurant est symétrique, la dilatation revient à l'addition de Minkowski

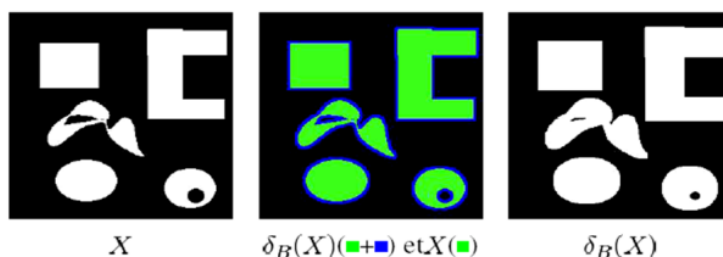


FIGURE 1.21 – Dilatation illustrée

Remarque 1.5 : Dilatation avec des éléments de taille croissante

- la taille des objets augmente,
- les trous et les concavités peuvent être bouchés,
- les objets voisins peuvent se connecter,
- des petits détails disparaissent.

Exemple 1.2 : Application

↳ Dilaté :
 Pour chaque position de B, est-ce que l'intersection entre B et l'objet A est non vide ? Si oui, x l'origine de B appartient à l'image dilatée

↳ Élément structurant de centre et de taille $\lambda = 1$

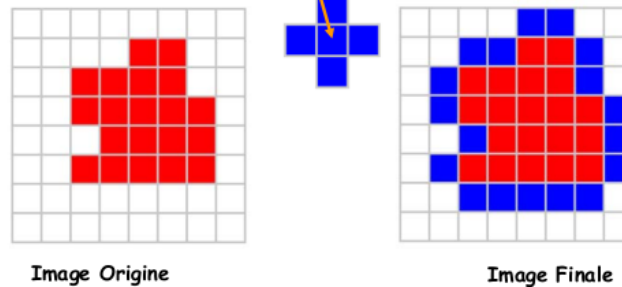


FIGURE 1.22 – Application dilatation

Erosion

Définition 1.16 : Erosion

L'érosion morphologique d'un ensemble X par B est définie comme la soustraction de Minkowski de X avec le transposé de B par rapport à son centre.

$$\epsilon_B(X) = X \ominus B^T = \{x / \forall y \in B, x + y \in X\}$$

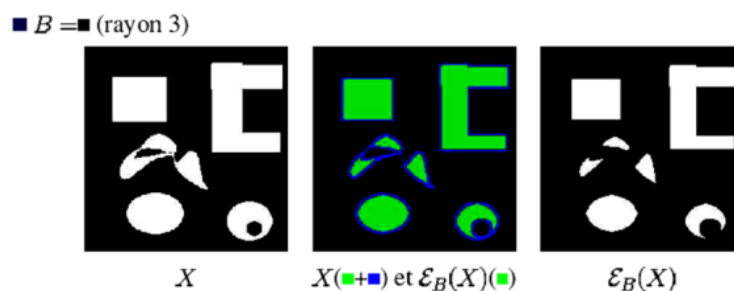


FIGURE 1.23 – Erosion

Remarque 1.6 : Erosion avec des éléments de taille croissante

- La taille des objets décroît,
- Un objet avec des concavités ou des trous peut être divisés en plusieurs,
- les petits objets et les détails disparaissent

Exemple 1.3 : Application érosion

- Soit B un élément structurant
 $Bx \rightarrow$ élément centré en un pixel x
- Erodé = On positionne l'origine de B en chaque pixel x de l'objet A : Si tous les pixels de B font partie de l'objet A , alors l'origine de B appartient à l'érodé

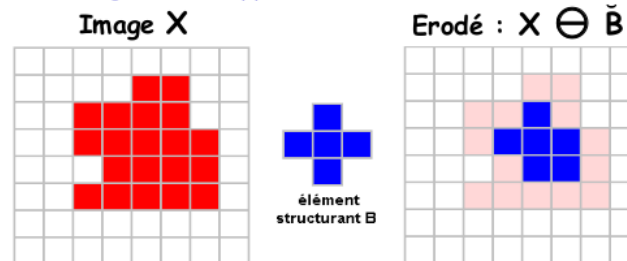


FIGURE 1.24 – Application érosion

Ouverture

Définition 1.17 : Ouverture

L'ouverture morphologique d'un ensemble X par B est définie comme une composition d'une érosion et d'une dilatation

$$\gamma_B(X) = (X \ominus B^T) \oplus B^T$$

Théorème 1.3 : Propriétés

- anti-extensive
- croissante
- idempotente

Fermeture

Définition 1.18 : Fermeture

La fermeture morphologique d'un ensemble X par B est définie comme composition d'une dilatation et d'une érosion.

$$\phi_B(X) = (X \oplus B^T) \ominus B^T$$

Théorème 1.4 : Propriétés

- extensive
- croissante
- idempotente

Fermeture vs. Ouverture

- L'ouverture élimine les petites composantes et ouvre les petits isthmes
- La fermeture bouche les petits trous et ferme les petits détroits

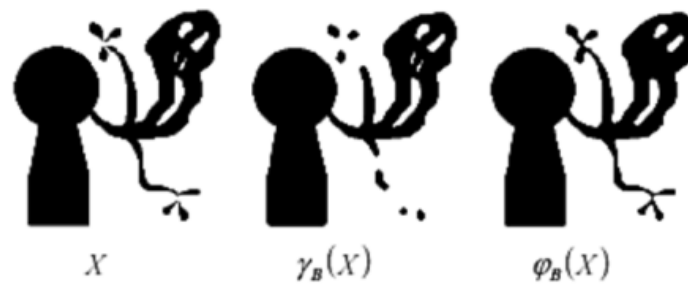


FIGURE 1.25 – Fermeture vs Ouverture

Gradient morphologique

Définition 1.19 : Gradient Morphologique

C'est la différence entre :

- gradient interne : $\nabla_B^- = X - \epsilon_B(X)$
- gradient externe : $\nabla_B^+ = \delta_B(X) - X$

Gradient morphologique : $\nabla_B^+ + \nabla_B^- = \delta_B(X) - \epsilon_B(X)$

➤ Gradient interne : contour intérieur: $\nabla_B^- = X - \epsilon_B(X)$



➤ Gradient externe : contour extérieur: $\nabla_B^+ = \delta_B(X) - X$



FIGURE 1.26 – Gradient interne et externe



FIGURE 1.27 – Gradient morphologique

1.2.9 Morphologie mathématique - Opérations géodesiques

Définition 1.20 : Opérations géodesiques

Les opérations géodesiques sont des opérations conditionnées par un élément de référence. Elles sont définies à partir des opérations géodésiques de base

- **Dilatation géodesique (dilatation conditionnelle) :**

- Dilatation de X dans Y un élément de taille 1

$$\delta_{B_1}^Y(X) = \delta_{B_1}(X) \cap Y$$

- Dilatation de X dans Y par un élément de taille λ

$$\delta_{B_\lambda}^Y(X) = [\delta_{B_1}(X) \cap Y]^\lambda$$

- **Erosion géodesique (érosion conditionnelle)**

- Erosion de X dans Y par un élément de taille λ

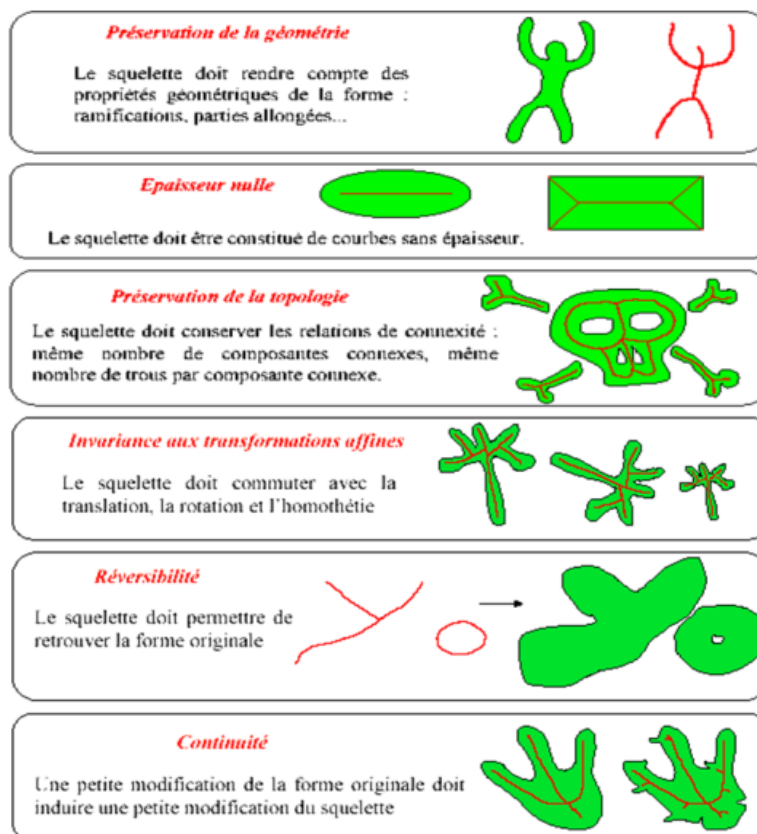
$$\epsilon_{B_\lambda} = X \setminus \delta_{B_\lambda}^Y(X)$$

1.2.10 Morphologie mathématique - Squelette

Définition 1.21 : Squelette

Le squelette est une représentation compacte des objets, homotopique à l'objet de départ

Propriétés recherchées :



Définition 1.22 : Squelette morphologique

Le squelette morphologique est par définition l'union (pour tout les $\lambda > 0$) de l'intersection (pour tous les $\lambda > 0$) de la différence entre l'érosion de X par l'élément structurant B de la taille λ et de l'ouverture par B_μ de l'érosion de X par B_λ

↳ Formule de Lantuéjoul:

$$S(X) = \bigcup_{\lambda} \bigcap_{\mu} [\varepsilon_{B_\lambda}(X) \setminus \gamma_{B_\mu}(\varepsilon_{B_\lambda}(X))]$$

$$= \bigcup_n [\varepsilon_{B_n}(X) \setminus \gamma_{B_1}(\varepsilon_{B_n}(X))]$$

➤ Propriétés:

- ↳ Sans épaisseur,
- ↳ anti-extensif,
- ↳ idempotent,
- ↳ non-homotopique,

- + ■ + ■ : X
- + ■ : $\varepsilon_{\lambda B}(X)$
- : $\gamma_{\mu B}(\varepsilon_{\lambda B}(X))$
- : $\varepsilon_{\lambda B}(X) - \gamma_{\mu B}(\varepsilon_{\lambda B}(X))$

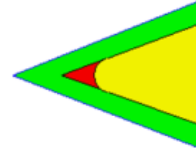


FIGURE 1.28 – Squelette morphologique

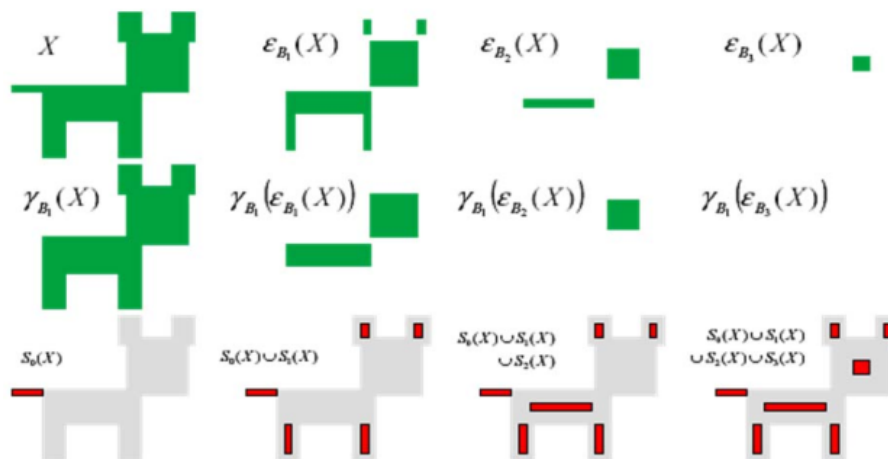


FIGURE 1.29 – Application squelette morphologique

CHAPITRE

2

SEGMENTATION ET EXTRACTION D'INFORMATIONS

2.1 Généralités

Définition 2.1 : C'est quoi la *ségmentation* ?

La **ségmentation** est une étape complexe qui regroupe plusieurs opérations.

- Sélection de la région ou des objets recherchés (on parle de R.O.I. pour *Region of Interest*)
- Représentation de la Segmentation
- Calcul des caractéristiques de la région sélectionné

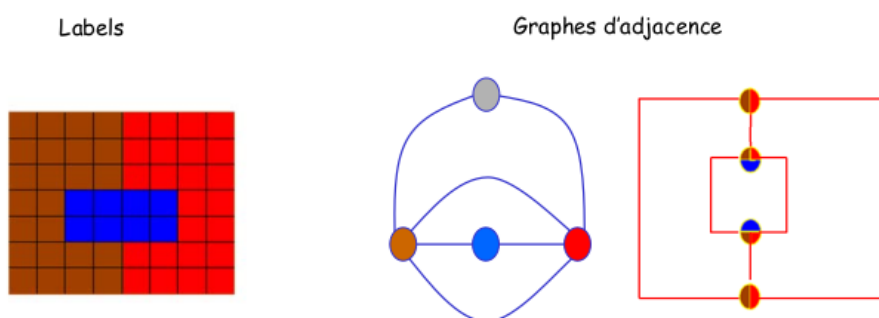


FIGURE 2.1 – Représentation de la segmentation

Méthode 2.1 : Etape clé de la segmentation

La segmentation vise à sous-diviser l'image en constituants distincts - objets. On appellera **segmentation** la tâche qui consiste à cataloguer chaque pixel comme appartenant à un certain groupe.

- Création d'une partition de l'image I en sous-ensemble R_i de Régions de non vides
- L'union des régions forme l'image I
- Un pixel appartient à une et une seule région ($R_i \cap R_j = \emptyset$)

Remarque 2.1 : Objectifs de la segmentation

- Extraire (séparer les entités d'une image)
 - Pour y appliquer un traitement spécifique
 - Pour interpréter le contenu de l'image
- Dans la pratique
 - Construire une image de masques
 - Chaque masque est une composante connexe
- On cherche des régions :
 - Connexes,
 - Homogènes en intensité
 - Délimités par des contours nets
 - Les plus grandes possibles

Remarque 2.2 : Difficultés de la segmentation

Dans les problèmes de la segmentation, il s'agit de diviser l'image selon une certaine sémantique, d'où la difficulté de donner une définition précise.

La segmentation est liée à la reconnaissance

2.2 Approche contours

Rappel : un contour est la frontière qui sépare 2 objets dans une image, c'est une **discontinuité de l'image**.

Définition 2.2 : Segmentation en contours

- Detections des morceaux de lignes dans l'image
- Une segmentation sera complète si les contours sont fermés et d'épaisseur minimale



FIGURE 2.2 – Exemples de détection des discontinuités

Après l'application d'un opérateur de détection de contours (discontinuités), on doit pouvoir obtenir une segmentation complète de l'image :

- Choisir les pixels appartenant aux contours
- Fermer les contours sélectionnés

La sélection des points de contours dépend fortement de :

- l'opérateur de détection utilisé
- du contenu de l'image et de ses conditions d'acquisition
- de l'objectif visé.

La plupart de ces opérateurs (*sauf Roberts*) lisse l'image afin d'obtenir un meilleur résultat.

En pratique, on obtient des contours incomplets

- il y a des pixels superflus
- il y a des manques
- il y a des erreurs dans la position et l'orientation des pixels contours

Un opérateur de détection de contour n'est qu'une première étape dans la chaîne de segmentation.

Il faudra utiliser d'autres techniques pour sélectionner les contours obtenus avec ces opérateurs et pour les compléter (approximation polyfonal).

On a deux méthodes de sélection :

- **Seuillage du gradient** : Simple, sensible au bruit et aux différences de contraste
- **Recherche du max de gradient par seuillage** : Plus long en temps de calcul, plus précis.

Définition 2.3 : Les grandeurs du gradient

- La **norme** : intensité du gradient en chaque pixel

$$|G| = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y|$$

- La **direction** : la direction de gradient le plus fort pour chaque pixel

$$\theta = \arctan\left(\frac{G_y}{G_x}\right)$$

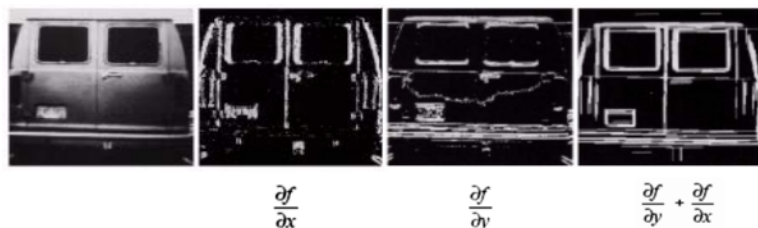


FIGURE 2.3 – Les grandeurs du gradients

On peut tracer les deux composantes sur l'image : on forme un vecteur (norme + direction)

2.3 Approche Classification de pixels

Définition 2.4 : Segmentation par classification

La segmentation par classification (**clustering**) consiste à regrouper les pixels par classes de même critère, mais contrairement à l'approche région, on ne fait pas intervenir la notion de voisinage.

2.3.1 Classification sur l'histogramme

- Seuillage – Binarisation
 - Méthode simple et très populaire pour la segmentation d'objets dans les images numériques
- Le seuillage peut être de nature
 - Globale : un seuil pour toute l'image
 - Locale : un seuil pour une portion de l'image
 - Adaptative : un seuil qui s'ajuste selon les images/parties de l'image
- Inconvénients
 - Connaître le nombre de classes
 - Apparition de faux éléments (aucune prise en compte de la composante spatiale)
 - Choix du ou des seuils : seuillage automatique (Otsu)

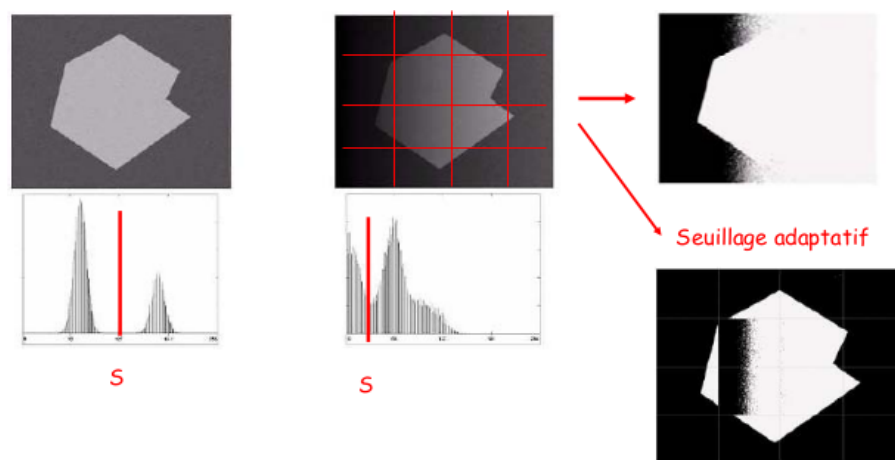


FIGURE 2.4 – Applications et limites de la classification sur histogramme

2.3.2 Codage d'un contour

Méthode 2.2 : Principe du codage de Freeman

On part d'un pixel du contour et on code le contour en le parcourant dans le sens des aiguilles d'une montre

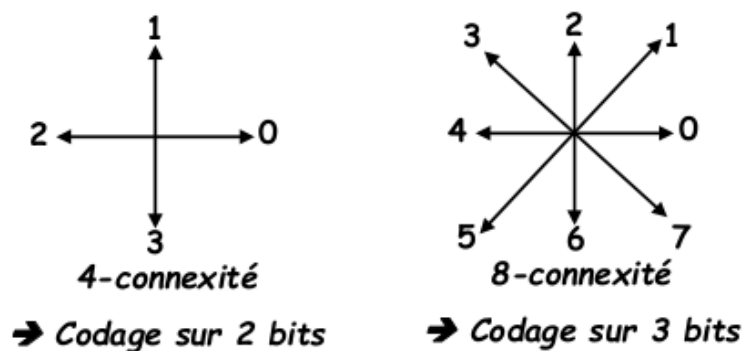
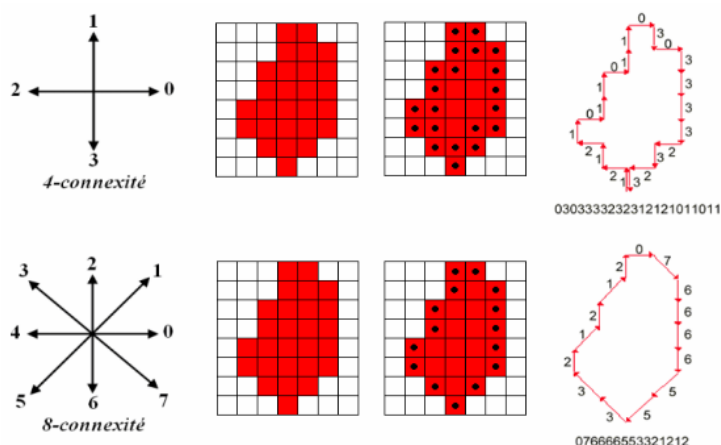


FIGURE 2.5 – Codage de Freeman

Exemple 2.1 : Application du codage de Freeman

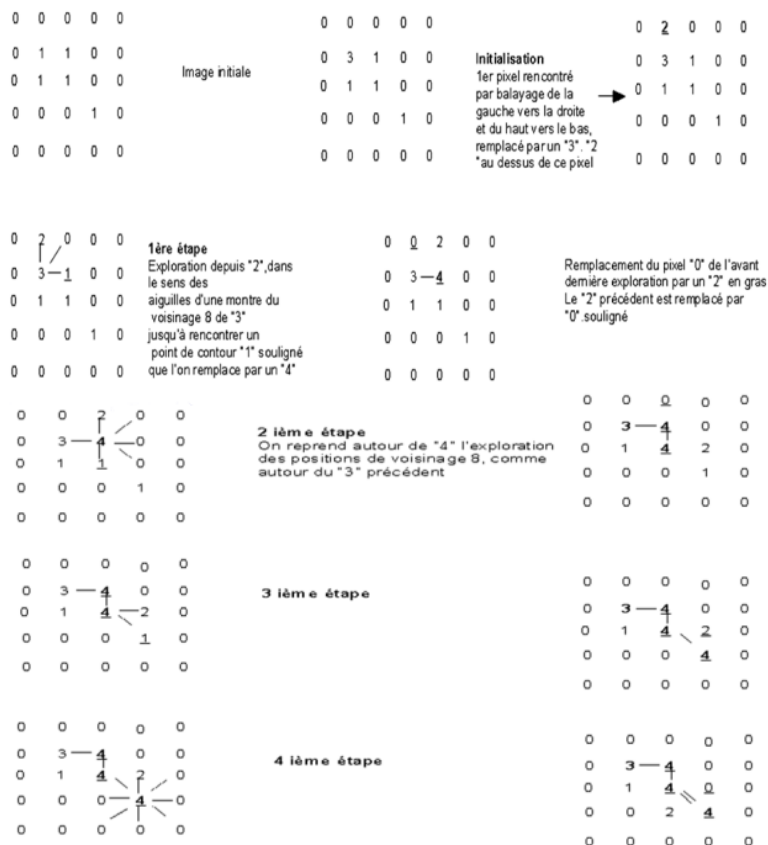


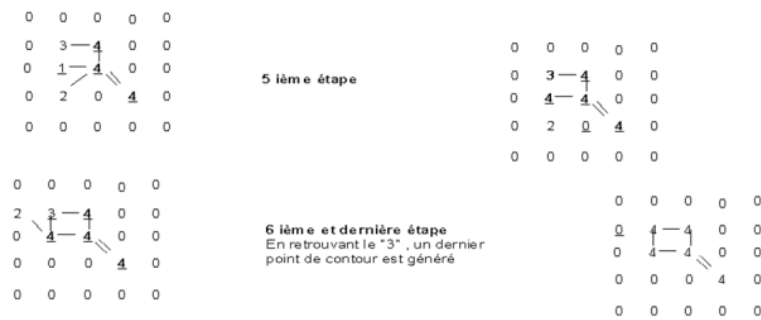
Principe de codage

L'image binaire de départ sera remplacée par un tableau de nombres qui utilisera les nombres :

- 0 : pour les pixels de fond,
- 1 : pour les pixels de l'objet,
- 2 : pour repérer la position de voisinage du pixel courant,
- 3 : pour la xase du premier pixel (1) rencontré lors d'un balayage de gauche à droite et de haut en bas de l'image,
- 4 : pour le remplacement d'un pixel par cette valeur dès que l'on sait que la case où se trouve le pixel appartient au contour.

A l'issue de l'algorithme, les pixels du contour seront remplacés par la valeur 4 : alors l'intérieur de l'objet conservera la valeur 1 des pixels, l'extérieur conservant des 0.





Calcul de périmètre

A partir du code Freeman : il y a 3 critères possible pour calculer $P_{\text{périmètre}}$:

- Compter le nombre de paires adjacentes $(p, q) : p \in S \text{ et } q \in \bar{S}$
- Compter le nombre d'étapes pour suivre le bord par l'algorithme de suivi de contour
- Même critère que le précédemment, mais les étapes obliques sont comptées pour $\sqrt{2}$

Ex pour critère 3 : $\text{péri}(X) = 9.414$

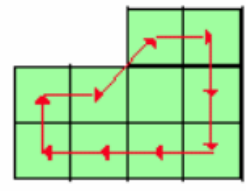


FIGURE 2.6 – Calcul du périmètre sur un exemple

Calcul de surface

On peut calculer la surface d'un objet à partir du code de freeman :

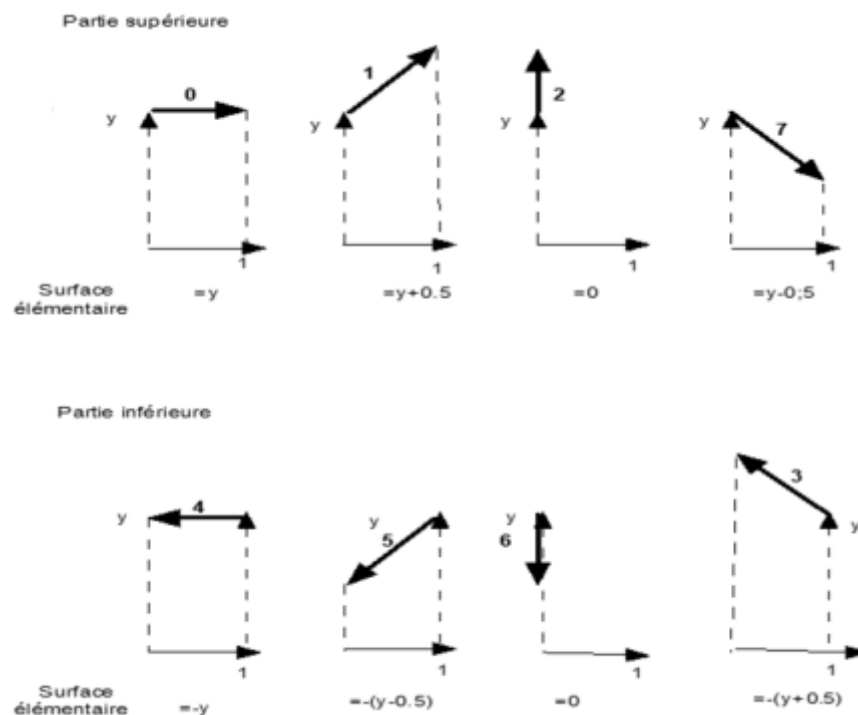


FIGURE 2.7 – Calcul surface à parti d'un code de freeman

CHAPITRE

3

INTERPRÉTATION, DÉCISION ET
RECONNAISSANCE DE FORMES

3.1 Clustering vs Classification

Définition 3.1 : Regroupement (Clustering)

Construire une collection d'objets

- Similaires au sein d'un même groupe
- Dissimilaires quand ils appartiennent à des groupes différents

Le Clustering est **de la classification non supervisée** pas de classes prédéfinies (Data Mining)

Définition 3.2 : Classification

Rechercher des règles ou formules (patterns) pour ranger les données dans des **classes prédéfinies** représentant un groupe d'individus homogènes permettant de classer les nouveaux arrivants : **Prédiction**.

C'est un processus en deux étapes :

- Construction d'un modèle sur les données dont la classe est connue → **apprentissage supervisé** (training data set)
- utilisation pour classification des nouveaux arrivants

3.2 Clustering

3.2.1 C'est quoi le *clustering* ?

Le clustering est l'affectation d'objets dans différents groupes ou plus précisément le **partitionnement** de données en **sous-ensembles** (clusters) de telle sorte que les données dans chaque sous ensemble (idéalement) soient ressemblantes au sens d'une **mesure de distance** prédéfinie

Le clustering est aussi appelé **apprentissage non-supervisé**, ou classification par les statisticiens, tri par les psychologues, segmentation dans le domaine du marketing

Affecter une étiquette à chaque observation ou donnée dans un ensemble : intuitivement, on affectera le même label à des données qui sont proches les unes des autres.

Quelquefois on dit pour le clustering, que le choix de *la mesure de distance est plus importante que l'algorithme lui même*

3.2.2 Deux types de *clustering*

Tout d'abord, qu'elles sont les propriétés recherchées pour un algorithme de clustering ?

- Passage à l'échelle (temps et mémoire)
- Capacité à traiter différents types de données
- Connaissance minimale sur le domaine pour paramétrer la méthode
- Capacité à résister au bruit et aux points aberrants
- Insensibilité à l'ordre de traitement des échantillons
- Possibilité d'incorporer des contraintes liées au domaine
- Interprétabilité et utilité

Définition 3.3 : Algorithmes de partitionnement

Construction de partitions variées et évaluation de ces partitions par un critère donné (k-means / Fuzzy C-means)

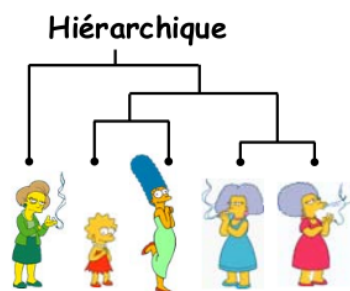


FIGURE 3.1 – Représentation de l'algorithme hiérarchique

Définition 3.4 : Algorithmes hiérarchiques

Création d'une décomposition hiérarchique de l'ensemble des objets en fonction d'un critère de décision (dendrogramme)

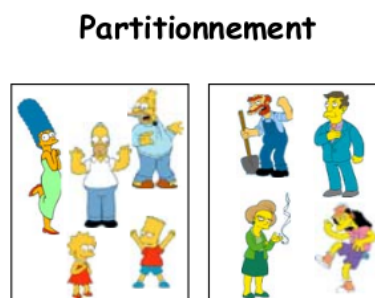


FIGURE 3.2 – Représentation de l'algorithme de partitionnement

Remarque 3.1 : But des algorithmes de cette section

Construire une partition d'une base de données de n objets en K clusters.

- Etant donné K , trouver une partition de K clusters qui optimise une fonction objectif (un critère d'évaluation du clustering)
 - Pour trouver l'optimum global, il n'y a pas de méthode générale efficace : il faut souvent examiner toutes les partitions possibles
 - k-moyennes et k-méloïdes sont des méthodes permettant de trouver des optimums locaux.

K-Moyenne : Chaque cluster est représenté par son centre

K-Méloïdes : Chaque cluster est représenté par un objet du cluster

Impact du choix des centroïdes initiaux

S'il y a autant de centroïdes qu'il y a de vrais clusters et qu'ils se trouvent chacun dans un cluster alors l'algorithme converge vers les vrais clusters. Mais on ne connaît pas au départ où se trouvent les clusters et on ignore parfois le nombre de clusters.

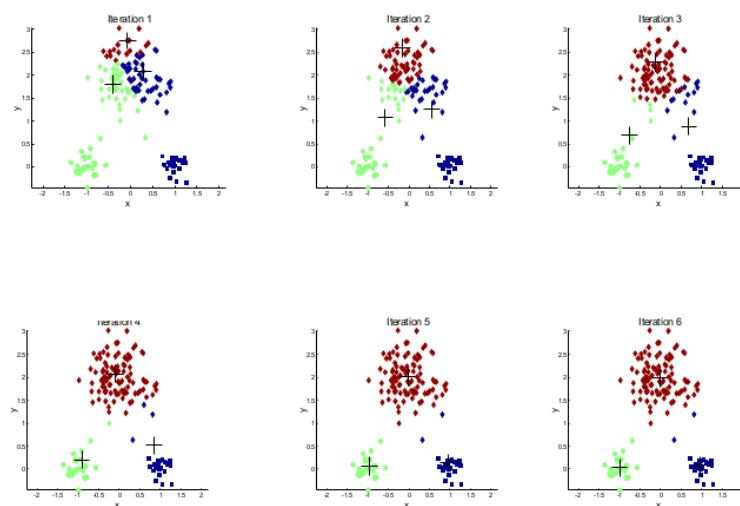


FIGURE 3.3 – Impact du choix des centroïdes initiaux

Solution au problème des centroïdes initiaux

- Multiples exécutions : les probabilités ne sont pas de notre côté
- A partir d'un échantillon, déterminer les centroïdes initiaux par une méthode de clustering hiérarchiques
- Choisir un premier centroïde aléatoirement ou comme étant le centre des données et ensuite choisir successivement $k-1$ centroïdes comme étant les points les plus éloignés des centroïdes déjà choisis
- Traiter le problème une fois le clustering terminé
- Choisir une variante moins susceptible au problème du choix des centroïdes initiaux : Mise à jour incrémentale des centroïdes (exemple : K-moyenne par bisection)

Force et Faiblesse**Forces :**

- Très efficace : $O(tkn)$, où
 - n = nbr objets
 - k = nbr clusters
 - t = nbr d'itérations

— avec $k, t < n$

- Aboutit souvent à un optimum local. L'optimum global peut-être trouvé en utilisant des méthodes telles que les algorithmes génétiques ou le recuit déterministe

Faiblesses :

- Applicable seulement quand la "moyenne" est définie. Que faire des données nominales ?
- Le nombre de clusters K doit être spécifié à l'avance
- Inadapté aux données bruitées et aux données aberrantes (outliers)
- Incapable de trouver des clusters de formes non-convexes

3.3 Classification

Remarque 3.2 : Les différentes formes de reconnaissance de formes

- Classification : modèle de classes / frontières de classes
- Détection de défauts : Modélisation de la "non-normalité"
- Identification : Modélisation d'individus parmi des inconnus
- Vérification : Identification d'un individu d'un ensemble connu sans étiquetage
- Reconnaissance de formes : étiquetage d'un individu parmi un ensemble connu

3.3.1 Les différentes phases

Définition 3.5 : Caractéristique (ou descripteur)

On appelle **caractéristique (ou descripteur)** une information qui peut-être mesurée sur la forme à reconnaître.

Le vecteur caractéristique est composé des plusieurs caractéristiques choisies pour représenter la forme (espace de représentation)

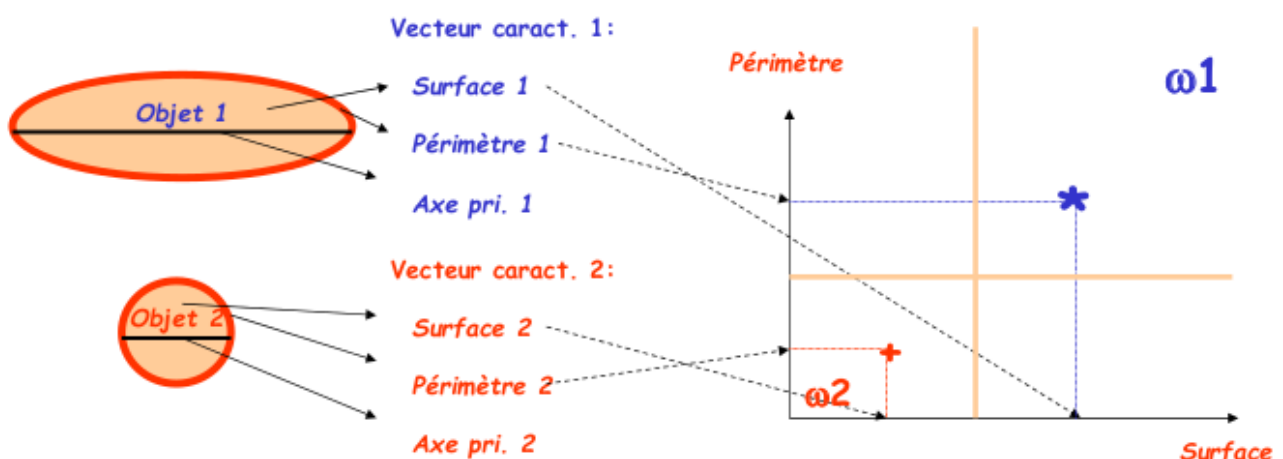


FIGURE 3.4 – Caractéristique en reconnaissance de forme

➤ **En extension (caractérisation):**

- ↳ Un ou plusieurs exemplaires : Prototypes de la classe (K-ppv).
- ↳ Ensemble d'exemplaires "résumés" : Modèle de la classe (gaussienne).

➤ **En intension (discrimination):**

- ↳ Un sous espace de R^n .
- ↳ Une frontière entre les formes

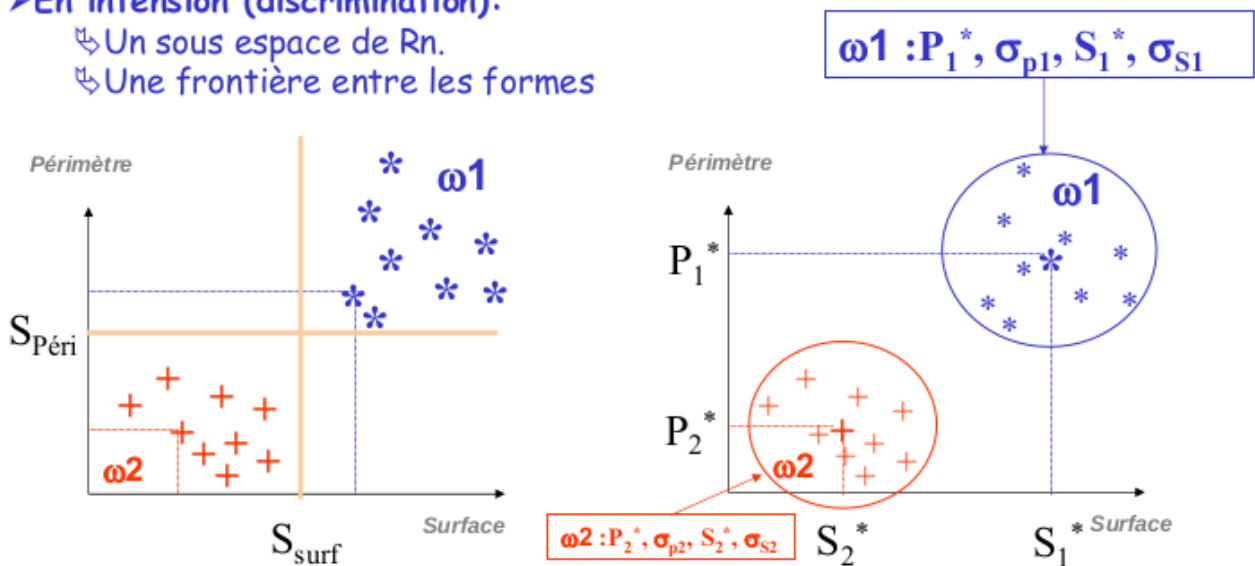


FIGURE 3.5 – Caractérisation et discrimination

Définition 3.6 : Primitive

On appelle **primitive** une composante élémentaire d'une forme, les primitives ne sont pas décomposables.

Par exemple : un segment de droite, une boucle ...

3.3.2 Evaluation

L'important est la capacité de prédiction du modèle : la vitesse de classification, d'induction du modèle et sa maintenance sont des éléments secondaires

➤ **La matrice de confusion**

		PREDICTED CLASS	
		Class=Yes	Class=No
TRUE CLASS	Class=Yes	a	b
	Class=No	c	d

- a: TP (true positive)
- b: FN (false negative)
- c: FP (false positive)
- d: TN (true negative)

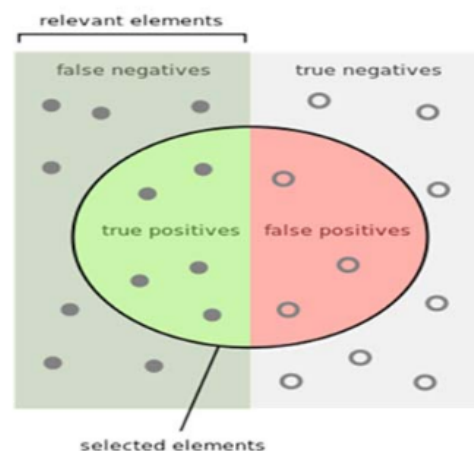


FIGURE 3.6 – Matrice de confusion

Justesse :

$$\text{Accuracy (Justesse)} = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}$$

	Classe prédite	
	Class=Yes	Class=No
Classe réelle	a (TP)	b (FN)
	Class=No	
	c (FP)	d (TN)

Taux d'Erreur ou Taux de Reconnaissance (Accuracy) : évaluation duale

Mémorisation : taux calculé sur le lot d'apprentissage

Généralisation : taux calculé sur un lot test « inconnu »

Justesse totale : mémorisation généralisation

↳ La justesse ne tient pas compte de la répartition des données dans les classes

	Classe prédite	
	Class=Yes	Class=No
Classe réelle	9990	0
	Class=No	
	10	0

↳ Si le modèle classe toutes les données dans la classe yes,

• sa justesse est

$$(9990+0) / (9990 + 0 + 10 + 0) = 99.9 \%$$

• mais il est incapable de reconnaître une donnée de la classe No

↳ Est-ce un classificateur utile ?

$$\text{Precision} = \frac{a}{a + c} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{a}{a + b} = \frac{TP}{TP + TN}$$

$$F\text{-measure} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} = \frac{2TP}{2TP + FP + FN}$$

#	Classe prédite	
	Class=Y	Class=N
Classe réelle	a	b
	Class=N	
	c	d

$C(\text{prédit}/\text{réel})$

- Précision est sensible à $C(\text{Yes}|\text{Yes})$ & $C(\text{Yes}|\text{No})$
- Rappel est sensible à $C(\text{Yes}|\text{Yes})$ & $C(\text{No}|\text{Yes})$
- F-mesure (moyenne harmonique) n'est pas sensible à $C(\text{No}|\text{No})$
 - plus proche de la valeur la plus faible (entre p et r)
 - donc une valeur forte signifie que la justesse et le rappel est bon

3.3.3 Arbres de décision

Définition 3.7 : Arbres de décision

- Règles de classification basant leur décision sur des tests associés aux attributs organisés de manière arborescente.
- Les noeuds internes (noeuds de décision) sont étiquetés par des tests applicables à toute description d'un individu. En général, chaque test examine la valeur d'un unique attribut. Les réponses possibles correspondent aux arcs issus de ce noeud
- Les feuilles sont étiquetées par une classe par défaut. Chaque noeud internes ou feuille est repéré par sa position : la liste des numéros des arcs qui permettent d'y accéder depuis la racine.
- Tout arbre de décision définit un classifieur
- Ce classifieur a une traduction immédiate en terme de règles de décision.

Exemple 3.1 :

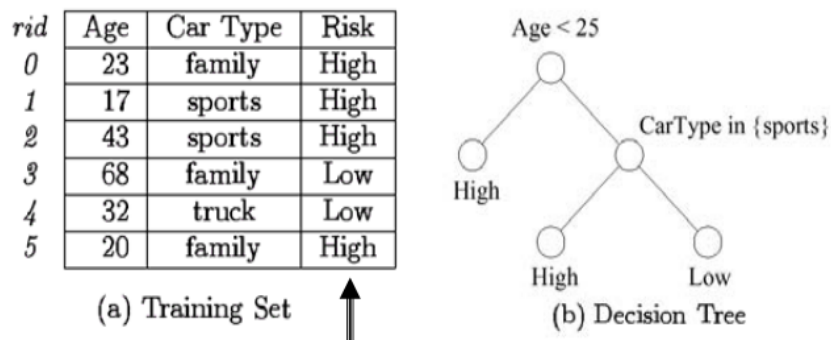


FIGURE 3.7 – Exemple d'arbre de décision

Méthode 3.1 : Procédure de construction

- Recherche à chaque niveau de l'attribut le plus discriminant
- Partition (noeud P)
 - si (tous les éléments de P sont dans la même classe) alors retour ;
 - pour chaque attribut A faire
 - évaluer la qualité du partitionnement sur A
 - utiliser le meilleur partitionnement pour diviser P en P_1, P_2, \dots, P_n

3.4 Conclusion

Théorème 3.1 : No Free Lunch Theorem

Le "No Free Lunch Theorem" dit qu'en l'absence d'information a priori sur le problème à traiter, il n'y a pas d'algorithme d'apprentissage supérieur à un autre

- Il y a autant de problèmes pour lesquels un algorithme est supérieur à un autre qu'il y a de problèmes où c'est l'inverse
- En moyenne, sur tous les problèmes de classification possibles, les algorithmes de classifications ont les mêmes performances

Théorème 3.2 : Ugly Ducking Theorem

Le “Ugly Ducking Theorem” dit qu’il n’y a pas d’ensemble de caractéristiques meilleur qu’un autre pour l’ensemble des problèmes (ou en l’absence d’a priori sur la nature du problème).

- La qualité d’un ensemble de caractéristiques dépend donc du problème.
- Ensemble ces deux théorèmes nous disent qu’il n’y a tout simplement pas d’algorithme ou ensemble de caractéristiques “universellement” meilleur
- La théorie et les algos. ne suffisent donc pas ! Il faut connaître le problème.

CHAPITRE

4

TD1 : TRAITEMENTS AU NIVEAU DU PIXEL (LUT)

Exercice 1

On considère l'image suivante qui représente une scène contenant deux formes :

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

1. Donner les dimensions de l'image. Sur combien de niveau de gris est-elle quantifiée ? Donner sa dynamique

Remarque 4.1 : Dimensions

La dimension d'une image se donne *lignes* × *colonnes*

On a donc pour la dimension de l'image :

$$7 \text{ lignes} \times 14 \text{ colonnes} = 98 \text{ pixels}$$

Remarque 4.2 : Niveau de gris

Si on prend $N = \max(NG)$ dans l'image, alors l'image est quantifiée sur m niveaux de gris avec $N \leq m = 2^k$. De préférence, on prend la puissance de 2 la plus proche de N

Ici, 16 est la puissance de 2 la plus proche supérieur à 10. Donc au minimum, l'image est quantifiée sur 16 niveaux de gris

Remarque 4.3 : Dynamique d'une image

$$D = \max(I) - \min(I)$$

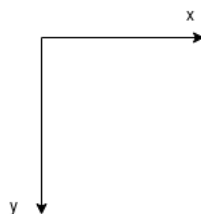
D'après cette définition, la dynamique de cette image est de $10 - 0 = 10$

2. La forme encadrée est un carrée de 2 mm de coté. Pourquoi apparait-il rectangulaire ? Quelle est la résolution de l'image ?

L'image apparait rectangulaire à cause du capteur.

Remarque 4.4 : Résolution en x et y

Cette valeur s'exprime en pix/mm



— Résolution en $x = \frac{4}{2} = 2pix/mm$

— Résolution en $y = \frac{3}{2}pix/mm$

3. Quelle est le numéro de la ligne i ? Tracer son profil.

Le numéro de la ligne i est 3.

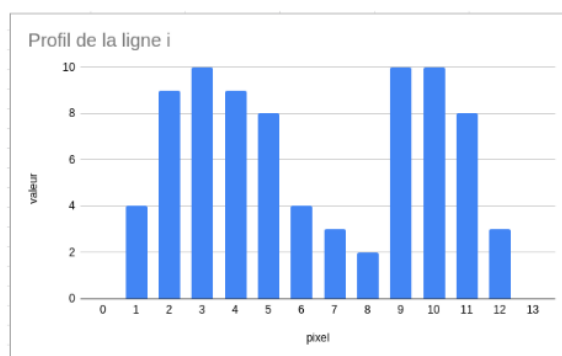


FIGURE 4.1 – Profil de la ligne i

4. Donner l'histogramme de cette image.

NG	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Effectif NG	48	2	7	6	6	6	2	2	8	5	6	0	0	0	0	0

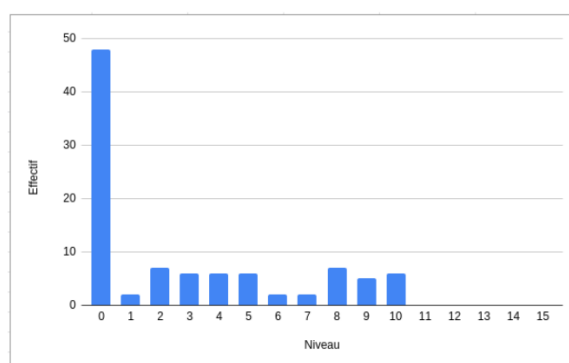


FIGURE 4.2 – Histogramme de l'image

5. Quelle transformation pourrait-on appliquer à l'image pour faire ressortir les deux formes ? Préciser les paramètres du traitement.

Il faut binariser l'image, on va donc la seuiller. Le seuil est au-dessus du niveau de gris le plus élevé du fond (4) et en dessous du niveau le plus bas (8).

On peut donc prendre $seuil = 6$

6. Proposer une méthode pour calculer automatiquement la valeur de seuil.

- Méthodes locales (sur la ligne i)
 - Moyenne : Pour la ligne i elle est de 5,7 (Ce seuil ne fonctionne pas pour l'image entière)
 - Médiane : 6
 - Dynamique : $\frac{10}{2} = 5$
- Méthodes globales
 - Moyenne : 2,88 donc 3
 - Médiane : 1
 - Dynamique : 5

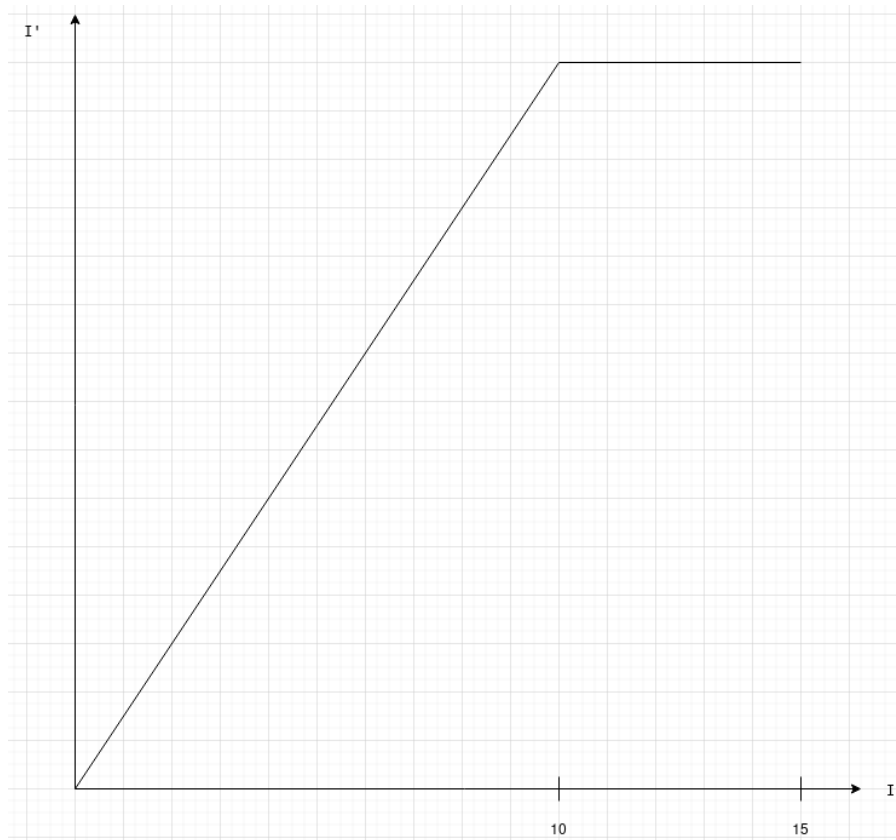
Remarque 4.5 : Otsu

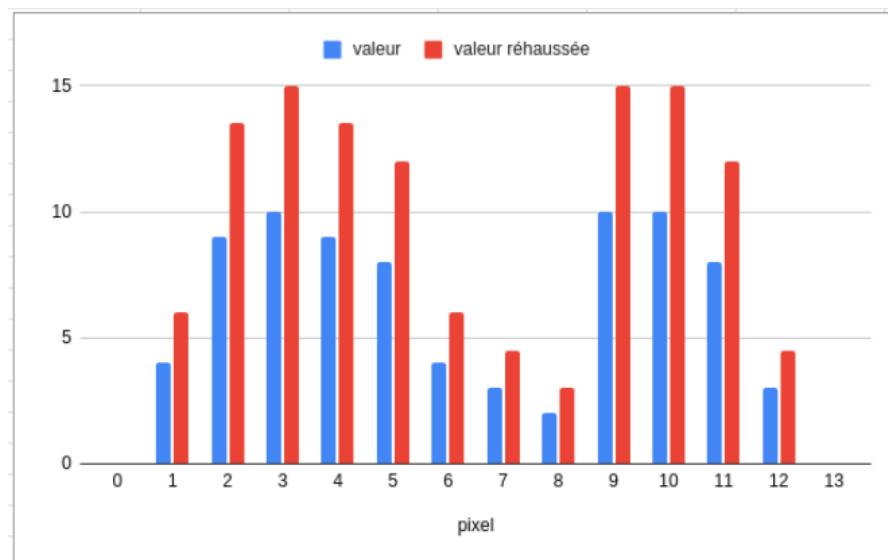
Pour la méthode globale, on pourrait appliquer l'algorithme d'Otsu, mais la modélisation de l'histogramme par deux gaussiennes est pas facile dans ce cas là.

7. Donner la LUT qui permet d'améliorer la dynamique de l'image. Donner le profil de la ligne rehaussée.

Comme notre image est quantifié sur 16 niveaux de gris, on veut que $\max(I') = 15$. Or $\max(I) = 10$. Donc la LUT correspondante pour réhausser I est :

$$I'(i, j) = 1,5 \times I(i, j)$$



FIGURE 4.3 – Profil de la ligne i rehaussée (rouge)

8. Donner la transformation qui permet de re-quantifier cette image sur 4 niveaux de gris. Détailler la graphiquement ou numériquement.

Le pas de quantification est $w = \frac{b-a}{n} = \frac{15-0}{4} \approx 4$ avec $n = 4$ le nombre de niveau de gris et $b - a$ la dynamique.

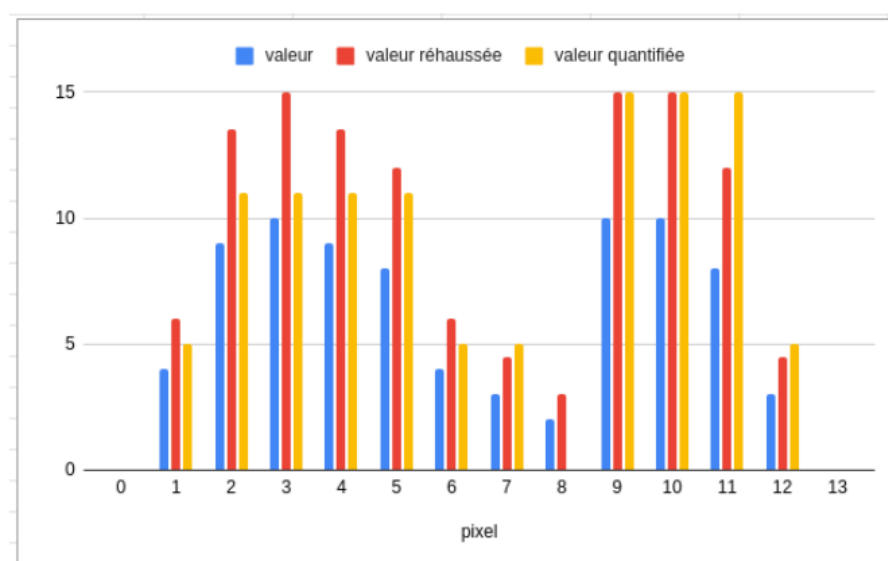
On a :

$$T_k = (k - 1) \times w$$

et

$$R_k = T_k \times \frac{N}{N-1} \quad \text{pour} \quad k = 1, \dots, N$$

$T_1 = 0$	$T_2 = 4$	$T_3 = 8$	$T_4 = 12$
$R_1 = 0$	$R_2 \approx 5$	$R_3 \approx 11$	$R_4 \approx 15$

FIGURE 4.4 – Profil de la ligne i rehaussée quantifiée (jaune)

9. Donner la LUT correspondant à l'égalisation de l'histogramme de l'image d'origine. Donner le profil de la ligne i après égalisation.

$$I'(i, j) = (2^4 - 1) \times \frac{HC(I(i, j))}{Nb_{lignes} \times Nb_{colonnes}}$$

Exercice 2

CHAPITRE

5

TD3 : PRÉTRAITEMENTS - FILTRAGE

Exercice 1

1. Calculer la convolution de l'image suivante avec un masque 3×3 ne contenant que des $1/9$. Quel est le résultat ? Quel est le filtre appliqué ? Comment traiter les bords ?

100	100	100	100	100	100	100	100	100	100	100	100
100	100	100	10	100	100	100	200	100	100	100	100
100	100	100	100	100	100	100	100	100	100	100	100

Le masque est le suivant :

$$\frac{1}{9} \times \begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}$$

Remarque 5.1 : Gestion des bords

- On peut faire comme s'il y avait que des 0.
→ Création de faux contour
- On peut faire comme un prolongement comme si on avait un élargissement.

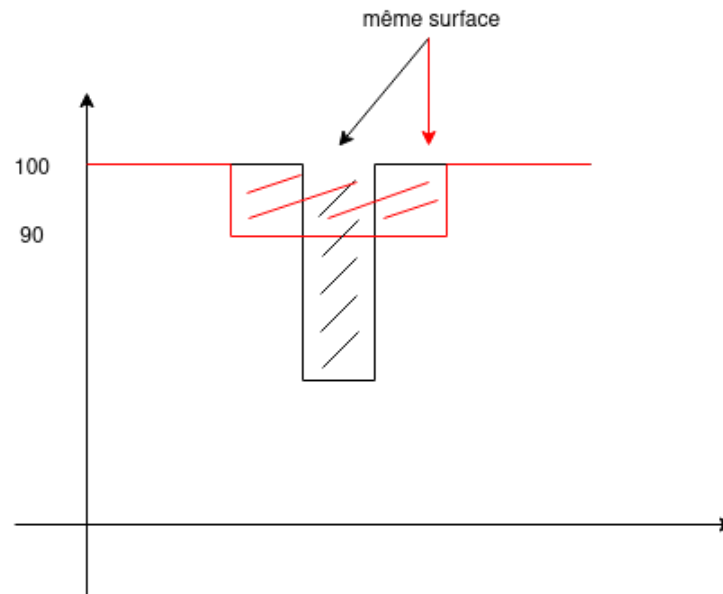
100	100	100	100	100	100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100	100	100	100	100	100
100	100	100	100	10	100	100	100	200	100	100	100	100
100	100	100	100	100	100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100	100	100	100	100	100

Ci-dessus, on a représenté l'image avec un élargissement des contours.

L'application du filtre donne :

100	100	90	90	90	100	111	111	111	100	100
100	100	90	90	90	100	111	111	111	100	100
100	100	90	90	90	100	111	111	111	100	100

2. Calculer la surface du pic d'intensité dans les deux images.



Exercice 2

Soit l'image suivante, où les nombres entiers désignent des niveaux de gris :

```

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

```

la direction Nord des masques :

```

1 1 1 1 2 1
0 0 0 0 0 0
-1 -1 -1 -1 -2 -1
Prewitt Sobel

```

1. A quel type de transformation correspondent ces masques ? Qu'obtiendrait-on si on appliquait un de ces masques sur toute l'image.

Ces masques sont des masques passe-haut qui permettent la **détection des contours**. Ces masques permettent la détection de contours **horizontaux**.

2. Remplir le tableau ci-dessous

Masque	Composante Nord	Composante Est	Module	Angle Est par rapport à Nord
Prewitt				
Sobel				

Remarque 5.2 : Masque de Prewitt

Filtre Prewitt Est : $P_E = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$, Filtre Prewitt Nord : $P_N = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$

Remarque 5.3 : Masque de Sobel

Filtre Sobel Est : $S_E = \begin{pmatrix} -1 & 0 & 1 \\ -c & 0 & c \\ -1 & 0 & 1 \end{pmatrix}$, Filtre Sobel Nord : $S_N = \begin{pmatrix} 1 & c & 1 \\ 0 & 0 & 0 \\ -1 & -c & -1 \end{pmatrix}$ avec $c > 1$

Composante Nord et Est

Remarque 5.4 : Composante Nord et Est

Composante Nord = $\frac{dI}{dy}$, et Composante Est = $\frac{dI}{dx}$

Pour appliquer un masque, il faut faire son symétrique : $P_N^R = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$, et on l'applique au pixel

entouré : $(-3) + (-6) + (-9) + 1 + 2 + 4 = 9$

On calcule de cette manière la composante nord et est pour les filtres de Prewitt et Sobel

Masque	Composante Nord	Composante Est	Module	Angle Est par rapport à Nord
Prewitt	-9	-12		
Sobel	-13	-17		

Calcul du module

Remarque 5.5 : Module

$$Module = \sqrt{\left(\frac{dI}{dy}\right)^2 + \left(\frac{dI}{dx}\right)^2}$$

Masque	Composante Nord	Composante Est	Module	Angle Est par rapport à Nord
Prewitt	-9	-12	15	
Sobel	-13	-17	$\approx 21,4$	

Angle Est par rapport à Nord

Remarque 5.6 : Angle

$$\theta = \arctan\left(\frac{\frac{dI}{dy}}{\frac{dI}{dx}}\right)$$

Masque	Composante Nord	Composante Est	Module	Angle Est par rapport à Nord
Prewitt	-9	-12	15	37°
Sobel	-13	-17	$\approx 21,4$	$39,5^\circ$

3. Que peut-on dire du pixel encadré, appartient-il à une zone particulière de l'image ?

Le pixel est dans une zone de transition.

CHAPITRE

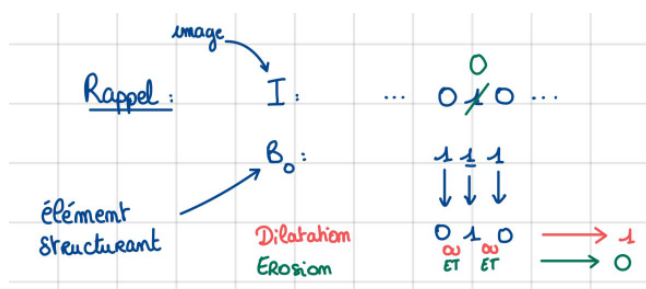
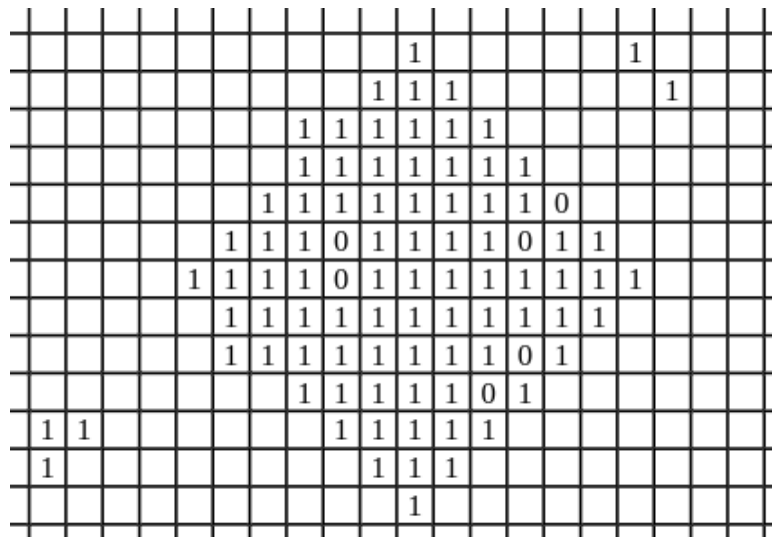
6

TD5 : PRÉTRAITEMENTS - MORPH. MATHÉMATIQUES

Exercice 1 : Morphologie mathématique

On considère l'image de l'annexe 1, elle contient un objet X noyé dans du bruit. La connaissance qu'on a de cet objet est qu'il a la forme d'un losange dont la particularité est d'être carré. Dans cette image binaire, ce carré devrait être uniformément composé de pixels blancs.

On a seulement noté, pour la clarté sur le dessin, les 0 et les 1 qui posaient éventuellement des problèmes dans ce contexte.



Remarque 6.1 : Dilatation

On fait le OU des éléments structurants et de l'image

Remarque 6.2 : Erosion

On fait le ET des éléments structurant et de l'image

Définition 6.1 : Ouverture

Erosion + Dilatation

Définition 6.2 : Fermeture

Dilatation + Erosion

Remarque 6.3 : Élément structurant non symétrique

Il faut le symétriser avant de l'appliquer :

$$B_0 = \begin{matrix} 1 \\ 1 \end{matrix} \longrightarrow \begin{matrix} 1 \\ 1 \end{matrix}$$

Soit le filtre $B_0 = \begin{matrix} 1 & 1 & 1 \\ 1 & & \end{matrix}$

Si on l'applique a la forme

	1	1	
	1		

On obtient par dilatation de l'élément structurant B_0 :

1	1	1	1
1	1	1	

Si en suivant on applique l'érosion (ce qui revient à faire une fermeture *dilatation + erosion*), on obtient :

0	1	1	0
0	1	0	

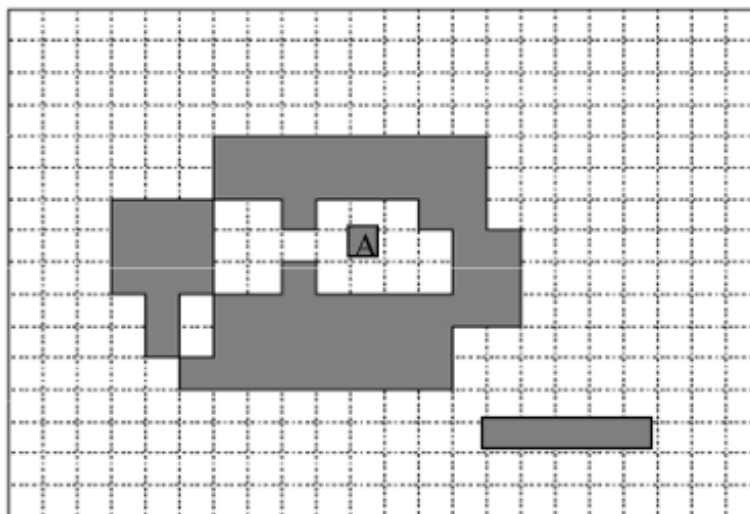
CHAPITRE

7

TD6 : SEGMENTATION, INTERPRÉTATION & DÉCISION

Exercice 1 : Topologie

Sur l'image suivante on considère que les niveaux de gris des pixels ont été inversés pour des raisons d'impression (le fond est noir).



1. Tracer la frontière des objets externes en utilisant une méthode de suivi de contours en 4-voisinage et en 8-voisinage.

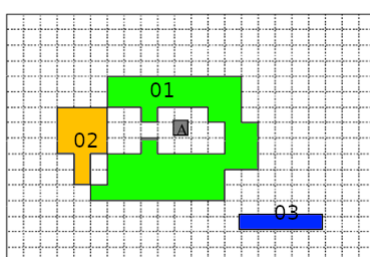


FIGURE 7.1 – 4 voisinages

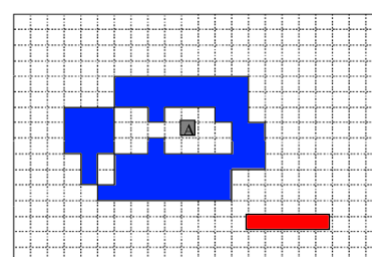


FIGURE 7.2 – 8 voisinages

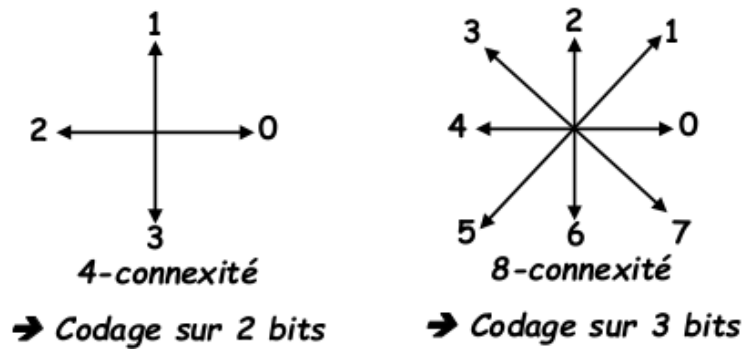


FIGURE 7.3 – Déplacement possibles en 4 et 8 voisinages

2. Donner le nombre d'objets et la relation d'imbrication de cette image en 4 et 8-voisinage en précisant le nombre de fils des objets externes (qui n'ont pas de parent) en 8-voisinage.

En 4 voisinages :

- 4 objets externes : O_1, O_2, O_3, A
- 0 évidements

En 8 voisinages :

- 2 objets externes : O_1, O_2 et A
- 2 évidements : E_1, E_2

3. Donner la chaîne de codes de Freeman dans le cas du 4-voisinage et du 8-voisinage.

En 4 voisinages :

- O_1 : 000000033303322332222201100130000011121222231221
- O_2 : 003323311211
- O_3 : 00002222
- A : \emptyset

En 8 voisinages :

- O_1 : 000000066766456444444323220012
- O_2 : 00004444
- A : \emptyset
- E_1 : 06002006064442446422
- E_2 : 62

4. A partir de la chaîne de codes, calculer dans les deux cas le périmètre et la surface des objets en pixels en supposant que l'objet A est sur la ligne i. Sachant que A est un défaut rectangulaire de 2mm de large et 3mm de haut, donner son périmètre mm et sa surface en mm²

- En 4-voisinage : $P_{O_1} = 32dx + 18dy = 118mm$, $S_{O_1} = 22 \text{ pixels}^2 = 118mm^2$, $P_{O_2} = 4dx + 8dy = 32mm$, $S_{O_2} = (i+1) + (i+1) + 0 + 0 - (i-1) + 0 + 0 + 0 - (i-1) + 0 + 0 = 4 \text{ pixels}^2 = 24mm^2$, $P_{O_3} = 8dx = 16mm$, $S_{O_3} = 0$
- En 8-voisinage : $P_{O_1} = 17dx + 9dy + 5$, $S_{O_1} = 68,5 \text{ pixels}^2 = 411mm^2$, $P_{O_2} = 8dx$, $S_{O_2} = 0$

Exercice 2 : Reconnaissance de formes

On souhaite effectuer la classification d'Iris en fonction de deux paramètres. Les données fournies dans le tableau 1 de l'annexe RDF correspondent à un ensemble de caractéristiques calculées sur 3 types d'Iris après la phase de segmentation.

1. On choisit dans un premier temps de faire une représentation des classes en intension en bornant chaque caractéristique indépendamment l'une de l'autre.

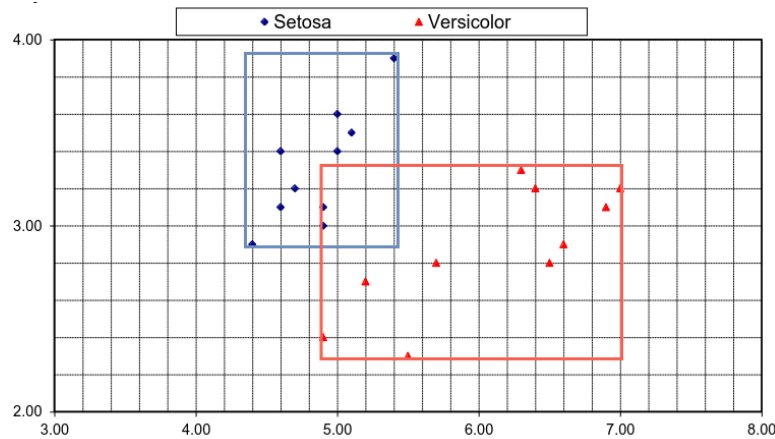
Donner les valeurs des bornes inf. et sup. pour les classes ω_1 et ω_2

Remarque 7.1 : En intension

Trouver des frontières et trouver des classes.

	Longueur	largeur	Longueur	largeur
min	4,9	2,3	4,4	2,9
max	7	3,3	5,4	3,9

Tracer sur le graphique 1 les frontières des classes ainsi obtenues



Quel est le taux de reconnaissance obtenu ?

On peut écrire la matrice de confusion :

	$\hat{\omega}_1$	$\hat{\omega}_2$
ω_1	10	0
ω_2	2	8

Remarque 7.2 : Justesse (Accuracy)

$$= \frac{\text{classe correctement}}{\text{total}}$$

On a donc $\text{Accuracy} = \frac{18}{20} = 90\%$

Précision :

— $\omega_1 : \frac{10}{12} = 83\%$

— $\omega_2 : \frac{8}{8} = 100\%$

Rappel :

— $\omega_1 : \frac{10}{10} = 100\%$

— $\omega_1 : \frac{8}{10} = 80\%$

Comment pourrait-on modifier les frontières de classes pour améliorer le résultat ?

A l'inverse des rectangles, on peut mettre des droites qui passe au milieu et tester si les points sont à gauche ou à droite.

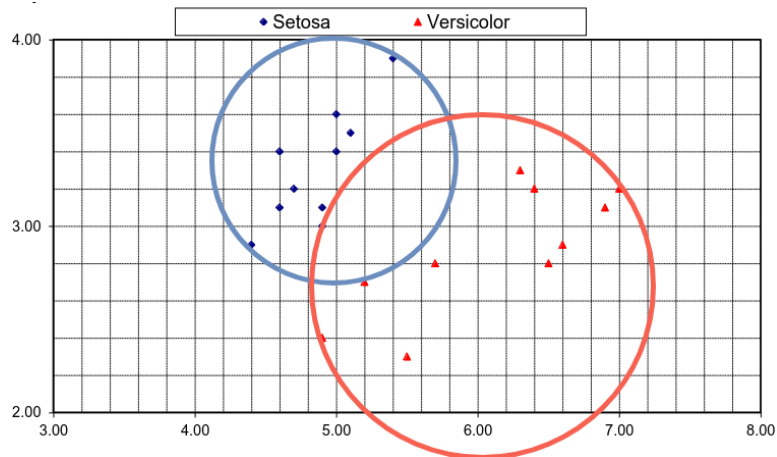
Sinon, on peut juste dire qu'on ne sait pas les classer car on est dans une zone de confusion

2. On choisit maintenant une représentation des classes en extension.

Donner le vecteur représentatif de chacune des classes.

	L	I
ω_1	6,1	2,87
ω_2	4,68	3,31

Tracer, sur le graphique 1, les formes de classes obtenues ainsi que la frontière inter-classe, dans le cas où on choisit la distance euclidienne pour la classification.



	Iris-versicolor : ω_1		Iris-setosa : ω_2		Iris-Virginica : ω_3	
	Param 1	Param 2	Param 1	Param 2	Param 1	Param 2
	7,00	3,20	5,10	3,50	6,30	3,30
	6,40	3,20	4,90	3,00	5,80	2,70
	6,90	3,10	4,70	3,20	7,10	3,00
	5,50	2,30	4,60	3,10	6,30	2,90
	6,50	2,80	5,00	3,60	6,50	3,00
	5,70	2,80	5,40	3,90	7,60	3,00
	6,30	3,30	4,60	3,40	4,90	2,50
	4,90	2,40	5,00	3,40	7,30	2,90
	6,60	2,90	4,40	2,90	6,70	2,50
	5,20	2,70	4,90	3,10	7,20	3,60
Moyenne	6,10	2,87	4,86	3,31	6,57	2,94
Ecart-Type	0,73	0,34	0,29	0,31	0,80	0,34

FIGURE 7.4 – Annexe exercice 2