

8

Représentation des images

8.1 Introduction

Après la segmentation d'une image en régions, les agrégations de pixels doivent être représentées dans une forme appropriées pour faciliter leur traitement avec l'ordinateur. La représentation d'une région implique deux choix:

- On peut représenter la région avec sa frontière.
- On peut représenter la région avec son contenu interne.

Par exemple, la frontière d'une région peut être représentée par sa longueur, son orientation, etc. L'intérieur d'une région peut être représenté par la distribution de la couleur, de la texture, etc. Les caractéristiques utilisées pour représenter les régions doivent être le moins sensible possible au changement de taille, la translation et la rotation des objets.

8.2 Représentation des frontières

8.2.1 Parcours de la frontière des régions

Plusieurs algorithmes supposent que les frontières des régions sont ordonnées dans le même sens, ou dans le sens contraire des aiguilles d'une montre. Dans la suite, on va considérer le sens positif de parcours comme le sens de mouvement des aiguilles d'une montre. On suppose ici qu'on travaille sur des images binaires dans lesquelles les objets et le fond sont étiquetés par 1 et 0 respectivement.

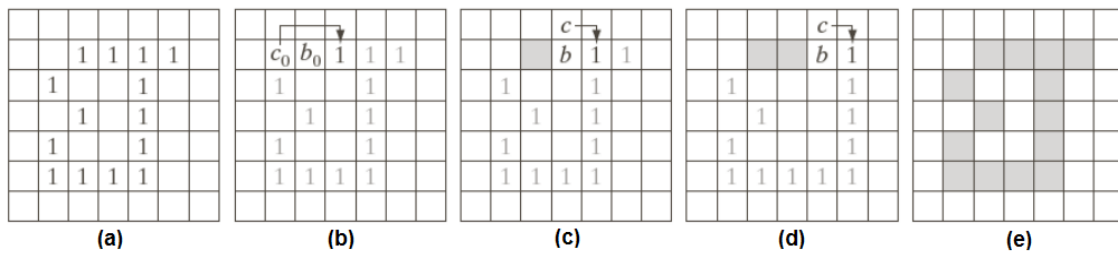


FIG. 8.1: Illustration des premières étapes de l'algorithme de parcours de frontières. Le point à traiter ensuite est marqué en noir, les points à traiter sont en gris, et les points trouvés par l'algorithme sont des carrés en gris.

8.2.2 Algorithme de parcours de frontières

En ayant une région R ou sa frontière (en image binaire), l'algorithme de parcours de frontières de région est comme suit:

- 1) Soit b_0 le point de départ de l'objet et c_0 le point à sa gauche (voir la Fig. 8.1(b)). Examiner les 8-voisins de b_0 , en commençant par c_0 dans le sens positif. Soit b_1 le premier point rencontré ayant la valeur 1. Soit c_1 le point du fond précédent à b_1 . Sauvegarder les emplacements de b_0 et b_1 pour les utiliser à l'étape 5.
- 2) Soit $b = b_1$ et $c = c_1$ (voir la Fig. 8.1(c)).
- 3) Soit les 8-voisins de b commençant à partir de c : n_1, n_2, \dots, n_8 . Trouver le premier point n_k ayant une étiquette 1.
- 4) Soit $b = n_k$ et $c = n_{k-1}$.
- 5) Répéter les étapes 3 et 4 jusqu'à ce que $b = b_0$ et le prochain point est b_1 .

8.2.3 Code chaîne

Le code chaîne permet de représenter la frontière par une séquence connectée de ligne de segment droit. Chaque segment est spécifié par sa longueur et son orientation. Le code est basé soit sur le voisinage 4-connexe ou 8-connexes. La direction de chaque segment est codée en utilisant le schéma de numérotation montré dans les Figs. 8.2(a) et 8.2(b).

Les images numériques sont souvent captées sous forme de grilles régulières dont les cellules sont les pixels. L'application du code chaîne sur une image telle quelle a les inconvénients suivants:

- 1) Le code sera trop long.
- 2) Les erreurs de segmentation et le bruit peuvent produire des perturbations dans le code.

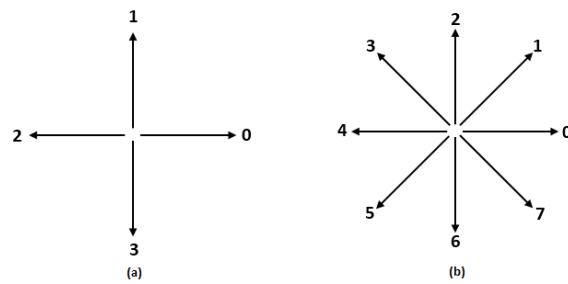


FIG. 8.2: Numéros de directions (a) Code chaîne 4-directions. (b) Code chaîne 8-directions.

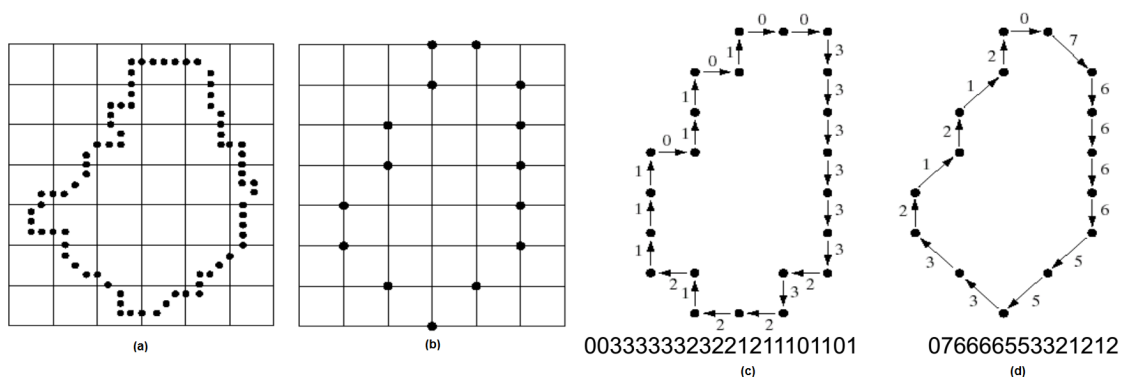


FIG. 8.3: (a) Frontière dans l'image originale. (b) Frontière après rééchantillonnage. (c) Code à 4 voisins. (d) Code à 8 voisins.

Une approche fréquemment utilisée pour contourner ces problèmes est de rééchantillonner les points de frontières des régions en utilisant des cellules plus grandes que les pixels, voir la Fig. 8.3(b). La frontière obtenue par le rééchantillonnage peut alors être représentée par une chaîne à 4 ou 8 voisins. Les Figs. 8.3(c) et (d) montrent les frontières représentées par un code 4-voisins et 8-voisins, respectivement.

Remarque 1 *Le code d'une frontière dépendra du point de départ pris pour la frontière.*

Solution

Le code peut être normalisé par rapport au point de départ. L'astuce utilisée est de considérer le code chaîne comme *une séquence circulaire de nombres*. On redéfinit alors le point de départ tel que la séquence résultante produit *la séquence de nombre dont l'amplitude est la plus petite* (voir la Fig. 8.4).

Remarque 2 *Le code d'une frontière est sensible à la rotation de l'objet. Autrement dit, si l'objet subit une rotation, son code va changer; ce qui n'est pas désirable.*

Solution

Le code peut être normalisé par rapport à la rotation. En utilisant la différence entre les nombres adjacents du code chaîne au lieu d'utiliser la chaîne elle-même (voir la Fig. 8.5).

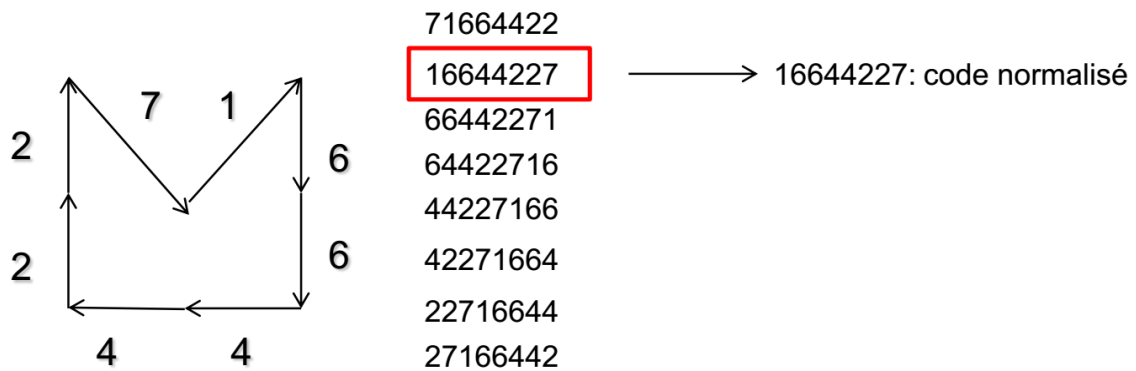


FIG. 8.4: Illustration de la normalisation d'un code chaîne par rapport au point de départ.

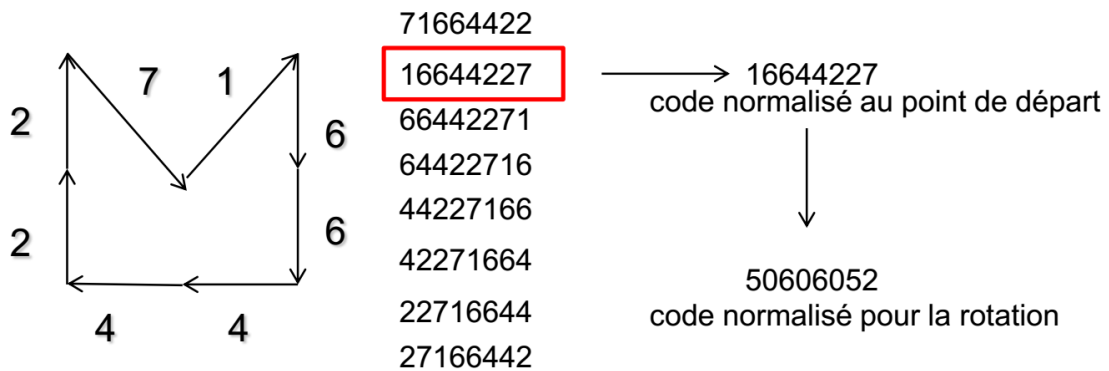


FIG. 8.5: Illustration de la normalisation d'un code chaîne par rapport à la rotation.

La Fig. 8.6 présente un exemple qui illustre un code chaîne avec sa rotation ainsi que la normalisation par rapport à la rotation qui permet d'obtenir le même code avant et après la rotation.

La Fig. 8.7(a) montre une image à niveaux de gris de taille 570×570 . L'objectif de cet exemple est d'obtenir le code chaîne normalisé par rapport à la rotation de la frontière de l'objet de la Fig. 8.7(a). La Fig. 8.7(b) montre l'image originale lissée avec un filtre moyenneur de taille 9×9 , et la Fig. 8.7(c) est le résultat du seuillage de cette image avec un seuil global obtenu en utilisant la méthode d'Otsu. La Fig. 8.7(d) est la frontière de la Fig. 8.7(c). L'obtention directe du code de chaîne entraînerait une longue séquence avec de petites variations qui ne sont pas représentatives de la forme de l'objet. Il est courant de rééchantillonner la frontière avant d'obtenir son code de chaîne afin de réduire la variabilité. La Fig. 8.7(e) est le résultat du rééchantillonnage de la frontière dans une grille avec des positions séparées de 50 pixels et la Fig. 8.7(f) présente le résultat après la liaison des sommets résultants par des lignes droites.

Le code original de la chaîne à 8-voisins de la frontière est:

0000606666666664444444242222202202

Le code normalisé au point de départ est:

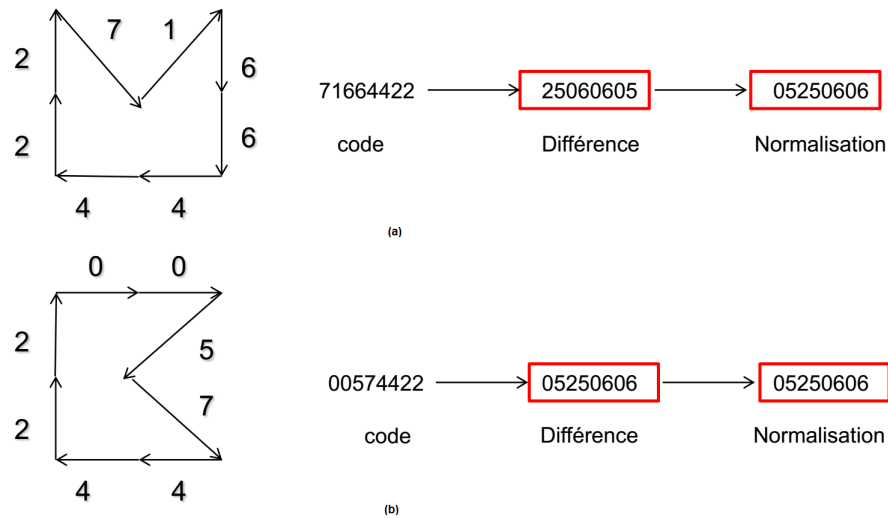


FIG. 8.6: Exemple de la normalisation d'un code chaîne par rapport à la rotation. (a) Le code chaîne et la normalisation avant la rotation. (b) Idem. que (a) après rotation.

0000606666666664444444242222202202

Le code normalisé à la rotation est:

0006260000000600000626000062062

8.2.4 Signatures d'objets

Une signature est une représentation unidimensionnelle de la frontière d'un objet. Elle peut être générée de différentes manières. Un exemple simple de signature peut être généré par le graphe de la distance de la frontière au centre de l'objet, comme fonction de l'angle, tel que illustré dans la Fig. 8.8. Cependant, indépendamment de la manière dont une signature est générée, l'idée de base consiste à réduire la représentation de la frontière à une fonction unidimensionnelle (1-D) qui est (peut-être) plus facile à manipuler qu'une fonction à deux dimension (2-D).

Les Figs. 8.9(a) et (b) montrent deux objets binaires et les Figs. 8.9(c) et (d) montrent leurs frontières. Les signatures correspondantes $r(\theta)$ sont montrées sur les Figs. 8.9(e) et (f), ces signatures se situent dans l'intervalle 0° à 360° avec une incrémentation de 1° . Noter le nombre important des pics dans les signatures, ce qui permet de discriminer les formes des deux objets.

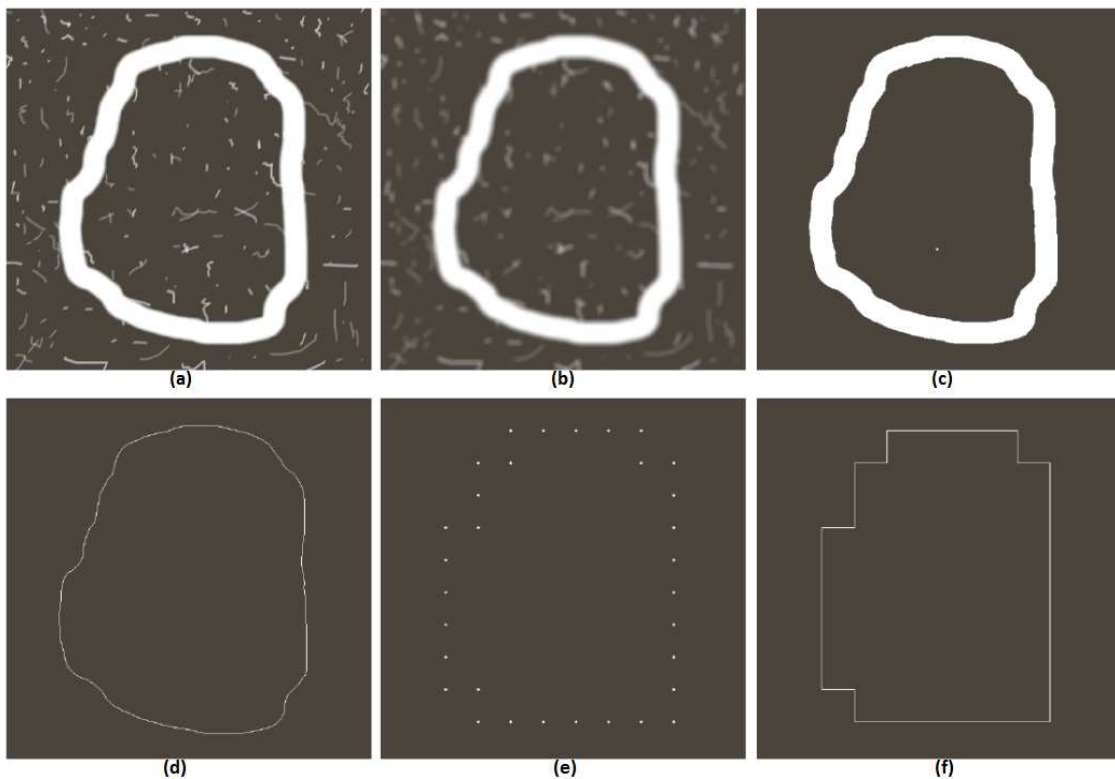


FIG. 8.7: (a) Image bruitée. (b) Image lissée avec un masque moyenneur 9×9 . (c) Image lissée, segmentée par la méthode d'Otsu. (d) La frontière extérieure de (c). (e) La frontière échantillonnée. (f) Points connexes de (e).

Remarque 3 Cette représentation est invariante à la translation, mais elle est sensible à la rotation et au changement d'échelle de l'objet.

Solutions proposées

- Pour rendre la représentation invariante à la rotation, **on peut choisir le point de départ de la signature comme le point le plus loin** (le plus prêt) du centre (on supposant que le point est unique).
- Pour rendre la représentation invariante au changement de taille des objets, **on peut faire en sorte que les distances calculées soient normalisées et prennent leurs valeurs dans l'intervalle $[0, 1]$.**

8.2.5 Squelettes d'objets

Le squelette d'objets est une approche pour représenter la *forme structurelle* d'un objet sous forme de graphe. La réduction peut être obtenue en construisant le squelette de l'objet via un *algorithme de squelettisation*.

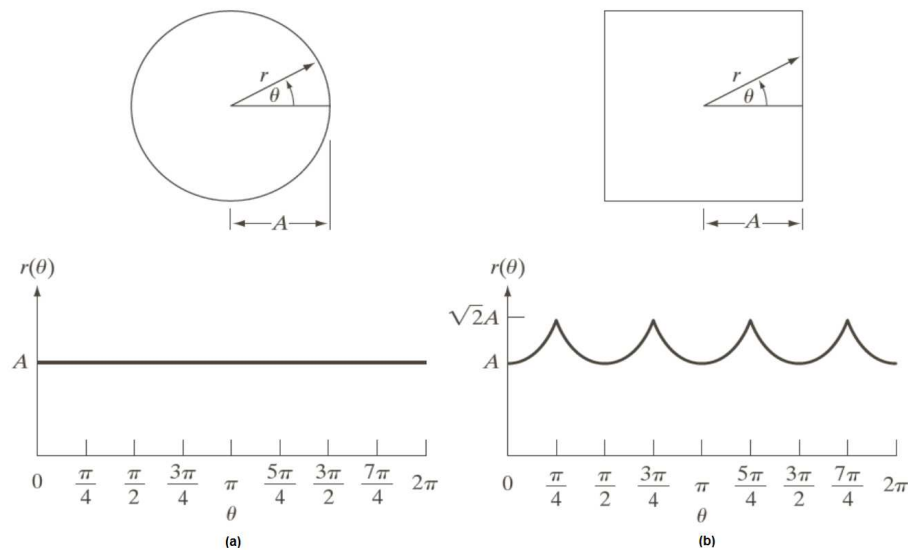


FIG. 8.8: Distance vs Signatures. (a) $r(\theta)$ est constant. (b) la signature consiste en des répétitions du motif.

Le squelette d'un objet peut être construit par la méthode de *transformation en axe médian TAM* (medial axes transformation *MAT*). La TAM pour un objet R qui a une frontière B et définie comme suit:

Pour chaque point p à l'intérieur de l'objet, on cherche son point le plus proche sur la frontière B . S'il y a plus d'un point proche, alors p va appartenir à l'axe médian.

La TAM d'un objet a une définition intuitive basée sur ce qu'on appelle *concept de feu de prairie*. On considère que l'objet est une prairie qui contient de l'herbe sèche distribuée uniformément. On suppose qu'on allume un feu le long de toute la frontière de l'objet. Les fronts du feu vont avancer à la même vitesse.

La TAM est alors définie comme étant tous les points de la prairie qui seront touchés par plus d'un front du feu en même temps.

Remarque 4 *L'implémentation directe de la squelettisation est très coûteuse en calcul car elle implique le calcul de la distance de chaque point intérieur de l'objet vers la frontière. On procède en général par des algorithmes plus efficaces.*

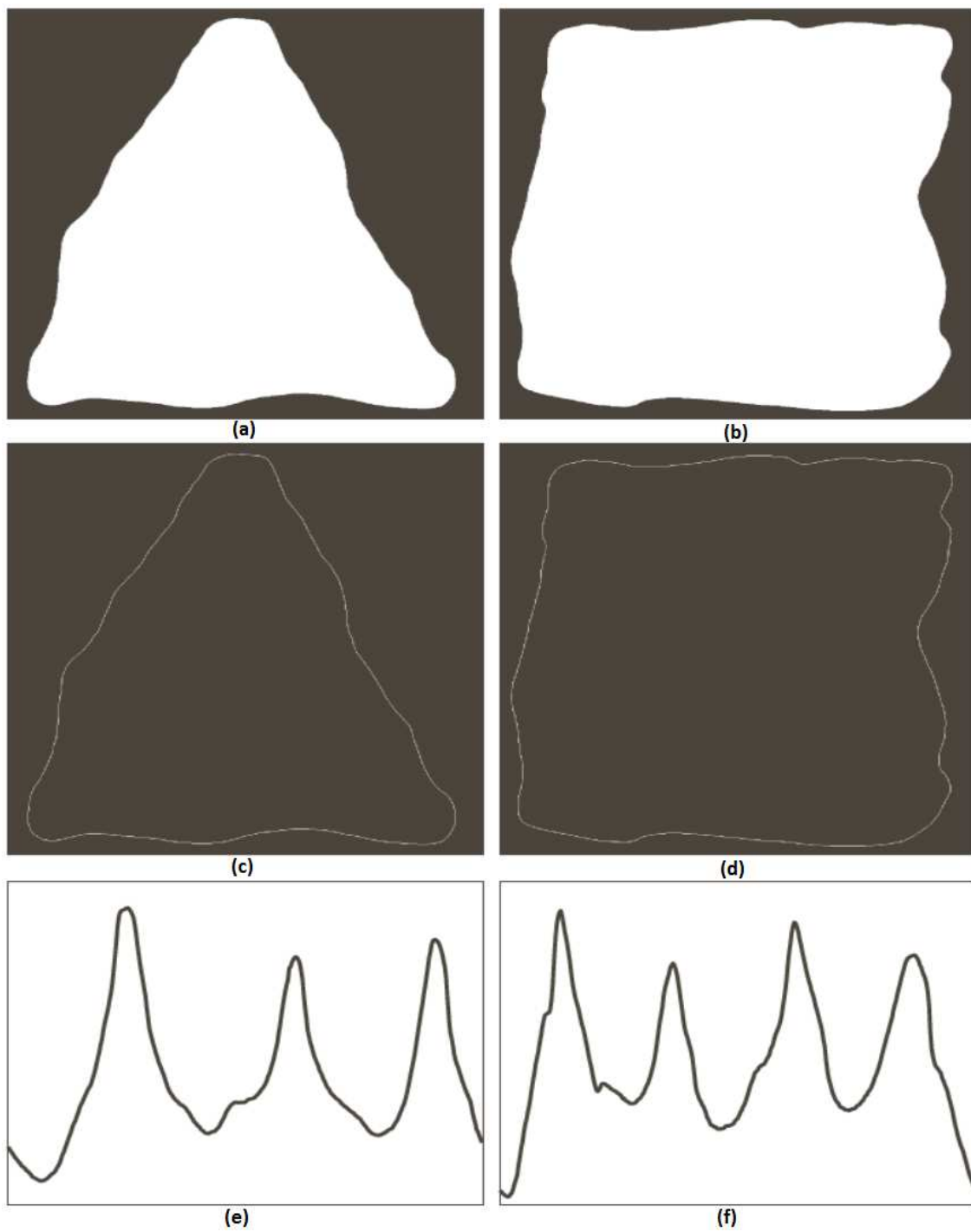


FIG. 8.9: Signatures de deux objets binaires simples. (a),(b) Image binaire d'un objet sur un fond. (c),(d) Frontière de l'objet. (e),(f) Signature de l'objet.

8.2.5.1 Algorithme de squelettisation

- On suppose que les pixels de l'objet sont des 1 et le fond des 0.

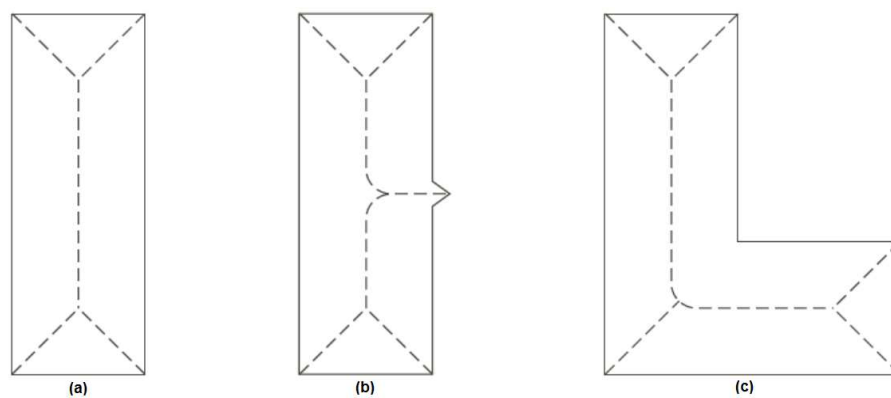


FIG. 8.10: Axes médians (pointillés) de trois régions simples.

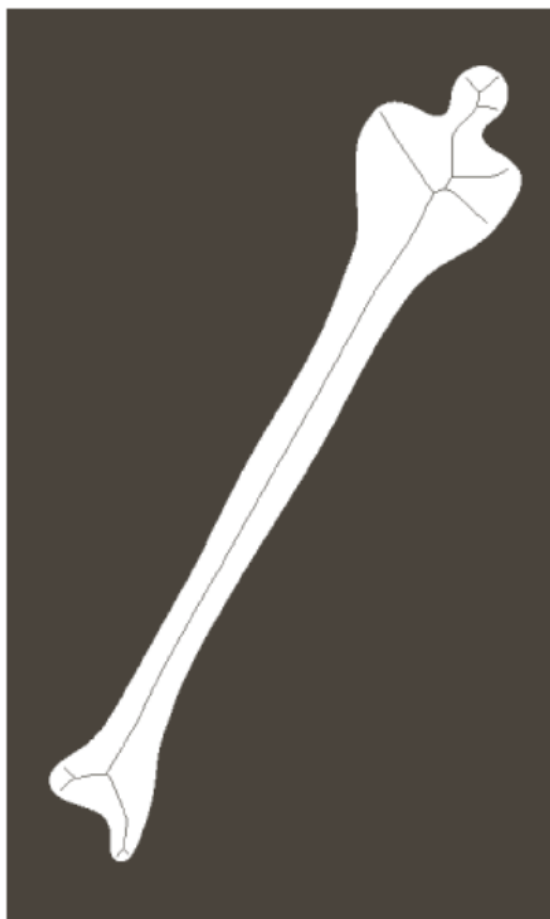


FIG. 8.11: Jambes humaines et squelette de la région superposés.

- Un point de frontière doit avoir une valeur 1 et au moins un voisin 0.
- On suppose le système des 8-voisins est représenté par la figure suivante :

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

- L'algorithme se base sur deux étapes itératives principales :

- 1) Etiqueter un point p_1 de la frontière pour la suppression si les conditions suivantes sont remplies:

$$\begin{aligned}
 (a) \quad & 2 \leq N(p_1) \leq 6 \\
 (b) \quad & T(p_1) = 1 \\
 (c) \quad & p_1 \cdot p_4 \cdot p_6 = 0 \\
 (d) \quad & p_4 \cdot p_6 \cdot p_8 = 0
 \end{aligned} \tag{8.1}$$

où

$N(p_1)$: est le nombre de points $\neq 0$ du voisinage de p_1 :

$$N(p_1) = p_2 + \dots + p_8 + p_9 \tag{8.2}$$

$T(p_1)$: est le nombre de transitions $0 \rightarrow 1$ dans la séquence: $p_2 p_3 \dots p_9$

Une fois qu'on passe par tous les points de la frontière, on supprime les points étiquetés pour la suppression.

- 2) Etiqueter un point p_1 de la frontière pour la suppression si les conditions suivantes sont remplies:

$$\begin{aligned}
 (a) \quad & \text{Idem que l'étape 1.} \\
 (b) \quad & \text{Idem que l'étape 1.} \\
 (c') \quad & p_2 \cdot p_4 \cdot p_8 = 0 \\
 (d') \quad & p_2 \cdot p_6 \cdot p_8 = 0
 \end{aligned} \tag{8.3}$$

Une fois qu'on passe par tous les points de la frontière, on supprime réellement les points étiquetés pour la suppression.

Répéter les étapes 1 et 2 jusqu'à ce qu'il n'y ait aucun point étiqueté.

- 1) La condition (a) est violée quand on possède 1 ou 7 voisins de l'objet. p_1 est alors une arête du squelette qu'on ne doit pas supprimer.

- 2) La condition (b) est violée quand on est sur une arête. Autrement dit, cette condition empêche la cassure des arêtes du squelette.
- 3) Les conditions (c) et (d) sont satisfaites par la condition minimale : ($p_4 = 0$ ou $p_6 = 0$) ou ($p_2 = 0$ et $p_8 = 0$). Si un point satisfait cette condition, ainsi que (a) et (b), alors le point est soit sur la frontière droite ou inférieure.
- 4) Les conditions (c') et (d') sont satisfaites par la condition minimale : ($p_2 = 0$ ou $p_8 = 0$) ou ($p_4 = 0$ et $p_6 = 0$). Si un point satisfait cette condition, ainsi que (a) et (b), alors le point est soit sur la frontière gauche ou supérieure.

8.3 Représentation des régions

Dans cette section, nous allons présenter quelques approches pour la description des régions dans une image. Il est courant d'utiliser une combinaison de descripteurs de région et de frontières.

8.3.1 Descripteurs simples d'objets

L'*aire* d'un objet est le nombre de pixels appartenant à l'objet. Le *périmètre* d'un objet est la longueur de sa frontière. Bien que l'aire et le périmètre soient parfois utilisés comme descripteurs, ils s'appliquent principalement aux situations dans lesquelles la taille des *régions d'intérêt* (ROI) est invariante. Une utilisation plus fréquente de ces deux descripteurs consiste à mesurer la compacité d'une région $Compactness = (perimetre)^2 / aire$.

Par exemple, un descripteur simple d'objets peut être utilisé pour extraire des informations à partir des images. La Fig. 8.12 montre une image satellitaire des Amériques. Ce type d'images peut fournir un inventaire des agglomération humaines. Le capteur utilisé pour acquérir ces images peut détecter des spectres visibles et infrarouges, par exemple, la lumière, les feux, etc. Le tableau de la Fig. 8.12 montre le rapport de l'aire occupée par la lumière à la surface totale dans les quatre régions. Une telle mesure peut donner, par exemple, une estimation par région de l'énergie électrique consommée.

8.3.2 Descripteurs topologiques

Les descripteurs topologiques servent à donner une description globale des propriétés géométriques d'un objet (ex. connexité, nombre de trous, etc.).

La Fig. 8.13 illustre ce concept à travers deux exemples. Dans le premier exemple, l'objet possède 1 composante connexe et 2 trous (voir la Fig. 8.13(a)). Dans le deuxième exemple, l'objet possède 3 composantes connexes et 0 trou (voir la Fig. 8.13(b)).

Les nombres de *composantes connexes* C et le nombre de *trous* H sont utilisés pour le calcul du *nombre d'Euler* E , qui est une propriété topologique, définie par la formule suivante:

$$E = C - H \tag{8.4}$$



FIG. 8.12: Calcul de la proportion des régions habitées en utilisant des images infrarouges captées la nuit.

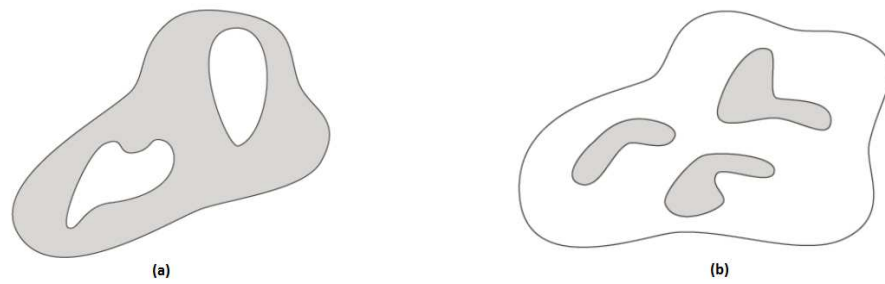


FIG. 8.13: (a) Une région à deux trous. (b) Une région avec trois composantes connexes.

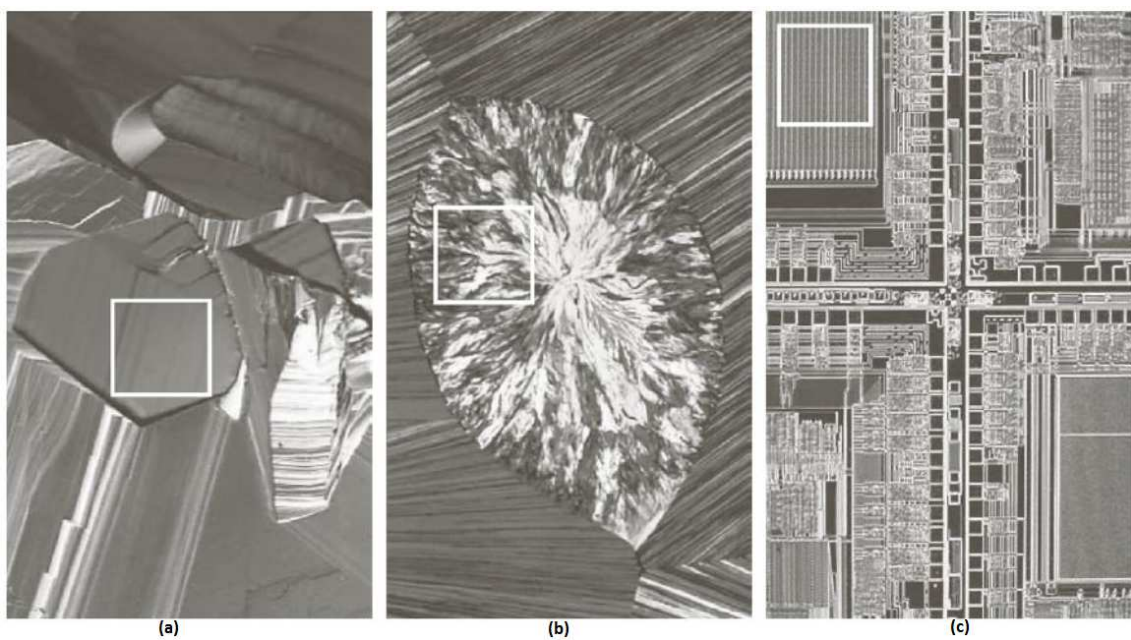


FIG. 8.14: (a) Texture lisse. (b) Texture grossière. (c) Texture régulière.

8.4 Représentation des textures

Une approche fondamentale de la description des objets est basée sur la description de texture. Intuitivement, la texture fournit des mesures de propriétés telles que la *douceur*, la *grossièreté*, et la *régularité* (la Fig. 8.14 montre quelques exemples). Il existent trois principales approches pour décrire la texture: *statistiques*, *structurelles* et *spectrales*. Les approches statistiques permettent de décrire les textures comme *lisses*, *grossières*, *granuleuses*, etc. Les techniques structurelles considèrent l'arrangement des primitives d'image, telles que la description de la texture basée sur des lignes parallèles régulières. Les techniques spectrales sont basées sur les propriétés du spectre de Fourier et sont principalement utilisées pour détecter la périodicité globale dans une image en identifiant les pics dans le spectre.

8.4.1 Descripteurs statistiques

L'une des approches les plus simples pour la description des textures est d'utiliser les *moments statistiques* de l'histogramme d'une image ou d'une région. Soit z la variable aléatoire qui représente l'intensité de l'image, et soit $p(z_i)$ l'histogramme de l'image, $z_i = 0, \dots, L-1$. On définit le moment d'ordre n des niveaux de gris comme suit :

$$\mu_n(z) = \sum_{i=0}^{L-1} (z_i - m)^n p(z_i) \quad (8.5)$$

où m est la moyenne des niveaux de gris, qui est donnée par :

$$m = \sum_{i=0}^{L-1} z_i p(z_i) \quad (8.6)$$

À noter que $\mu_0 = 1$, $\mu_1 = 0$ et $\mu_2 = \sigma^2$.

Le descripteur suivant est une mesure de contraste de la texture :

$$R = 1 - \frac{1}{1 + \sigma^2(z)} \quad (8.7)$$

- Si $R \rightarrow 0$, la région a une intensité constante.
- Si $R \rightarrow 1$, la région a une grande variance d'intensité.

Le moment d'ordre 3:

$$\mu_3(z) = \sum_{i=0}^{L-1} (z_i - m)^3 p(z_i) \quad (8.8)$$

mesure la *skewness* (asymétrie) de l'histogramme.

Finalement, on peut avoir des descripteurs comme :

$$U = \sum_{i=0}^{L-1} (p(z_i))^2 \quad (8.9)$$

qui mesure l'*uniformité* de la texture, et :

$$E = - \sum_{i=0}^{L-1} p(z_i) \log(p(z_i)) \quad (8.10)$$

qui mesure l'*entropie* de la texture.

Remarque 1

L'*uniformité* est le contraire de l'*entropie*.

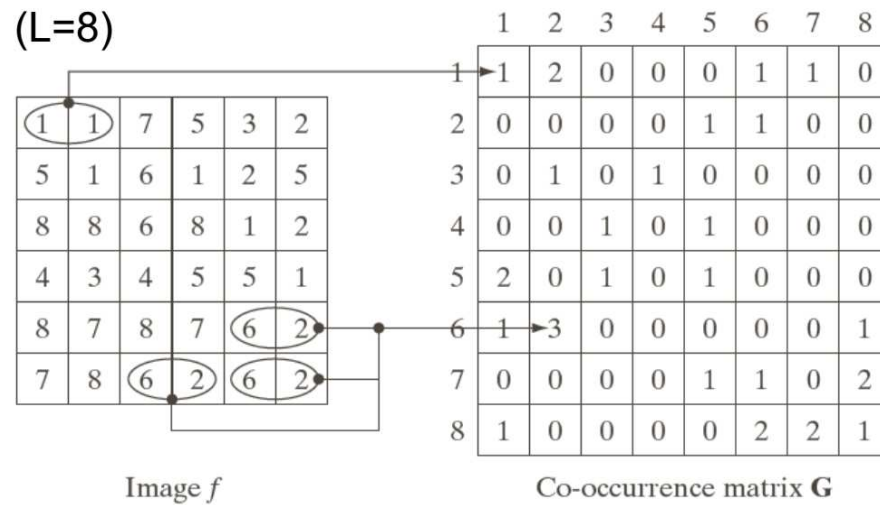


FIG. 8.15: Comment générer une matrice de co-occurrence.

8.4.2 Matrice de cooccurrence

Soit Q un opérateur qui définit la position de deux pixels de l'image l'un par rapport à l'autre. Soit L le nombre de niveaux de gris.

Soit G une matrice dont chaque élément représente le nombre de fois que les paires de pixels avec des intensités z_i et z_j apparaissent dans l'image suivant la position spécifiée par l'opérateur Q .

G est appelée une matrice de *co-occurrence*.

Soit n la somme des éléments de la matrice G ($n = 30$ dans l'exemple de la Fig. 8.15). On définit alors la quantité :

$$p_{ij} = \frac{g_{ij}}{n} \quad (8.11)$$

qui est la probabilité que la paire de points satisfaisant l'opérateur Q ait la valeur (z_i, z_j) . Il est alors clair que $p_{ij} \in [0, 1]$ et que:

$$\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} p_{ij} = 1 \quad (8.12)$$

On peut calculer plusieurs descripteurs à partir de la matrice de *co-occurrence*.

1) **Contraste:**

$$\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} (i - j)^2 p_{ij} \in [0, (L - 1)^2] \quad (8.13)$$

Mesure le contraste d'intensité entre chaque pixel et son voisin à travers toute l'image.

2) **Homogénéité:**

$$\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \frac{p_{ij}}{1 + (i - j)^2} \in [0, 1] \quad (8.14)$$

Mesure l'homogénéité de la texture (contraire du contraste). Autrement dit, elle mesure la proximité spatiale de la distribution des éléments de G sur sa diagonale.

3) **Uniformité:**

$$\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} p_{ij}^2 \in [0, 1] \quad (8.15)$$

Mesure l'uniformité de la texture. Elle est égale à 1 pour une image constante.

4) **Entropie:**

$$-\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} p_{ij} \log_2(p_{ij}) \in [0, 2 \log_2(L)] \quad (8.16)$$

Mesure le degrés auquel les éléments de G sont aléatoires (contraire de la texture régulière).

La Tab. 8.1 représente les valeurs des différents descripteurs calculés en utilisant les Eq. (8.13) à (8.16) et en se basant sur la matrice de co-occurrence de la Fig. 8.16.

Texture	Contraste	Uniformité	Homogénéité	Entropie
Texture 1	10838	0.00002	0.002	15.75
Texture 2	570	0.01230	0.079	6.43
Texture 3	1356	0.00480	0.017	13.58

TABLE 8.1: Les descripteurs calculés en utilisant les matrices de co-occurrence de la Fig. 8.16.

8.5 Conclusion

La représentation et la description des objets ou régions segmentés à partir d'une image est une étape préliminaire pour la plupart des processus automatisés de traitement et d'analyse d'image. Par exemple, les descripteurs étudiés dans ce chapitre constituent l'entrée aux méthodes de détection et reconnaissance d'objets. L'objectif est de choisir un descripteur qui permet une bonne discrimination entre les différents objets ou classes d'objets, avec le moins de sensibilité possible au changement de taille, de translation et de rotation des objets.

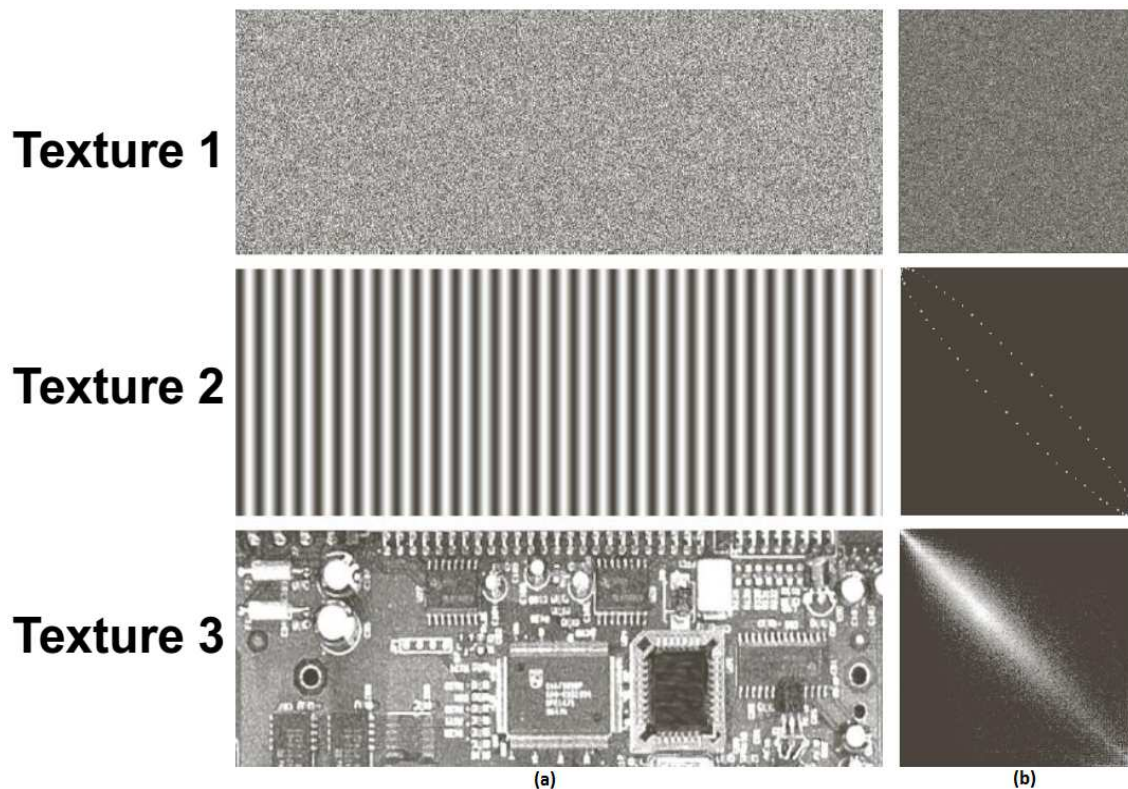


FIG. 8.16: (a) Images représentant trois types de textures, texture 1: aléatoire, texture 2: périodique, et texture 3: mixte. (b) La matrice de co-occurrence correspondante à chaque texture.

8.6 Travaux pratiques N° 8

8.6.1 Objectifs

- Introduire les notions de représentation des régions/objet dans l'image.

8.6.2 Enoncé

L'objectif de ce projet consiste à introduire les techniques permettant la représentation d'une région ou objet dans une image. Par exemple, une région peut être représentée par son contour et le contour peut être décrit par des caractéristiques telles que sa longueur, ses pixels, et le nombre de concavités qu'il contient.

1) Code chaine

Implémenter la fonction qui permet d'obtenir le code chaine et le contour d'objet d'une image numérique binaire f (Reproduire les images de la Fig. 8.7. Utiliser le fichier d'image *circular_stroke.tif*).

2) Squelette d'un chromosome

Soit l'image f d'un chromosome humain. L'objectif de cet exercice est de générer une image qui contient le squelette du chromosome contenu dans l'image f tel que montré sur la Fig. 8.17. Utiliser le fichier d'image *chromosome.tif*.

3) Matrice de co-occurrence

Ecrire le code Matlab qui permet de reproduire les images de la Fig. 8.16 en utilisant la fonction prédéfinie de MATLAB *graycomatrix*. Utiliser les fichiers des images: *texture_random.tif*, *texture_periodic.tif*, et *texture_mixed.tif*.

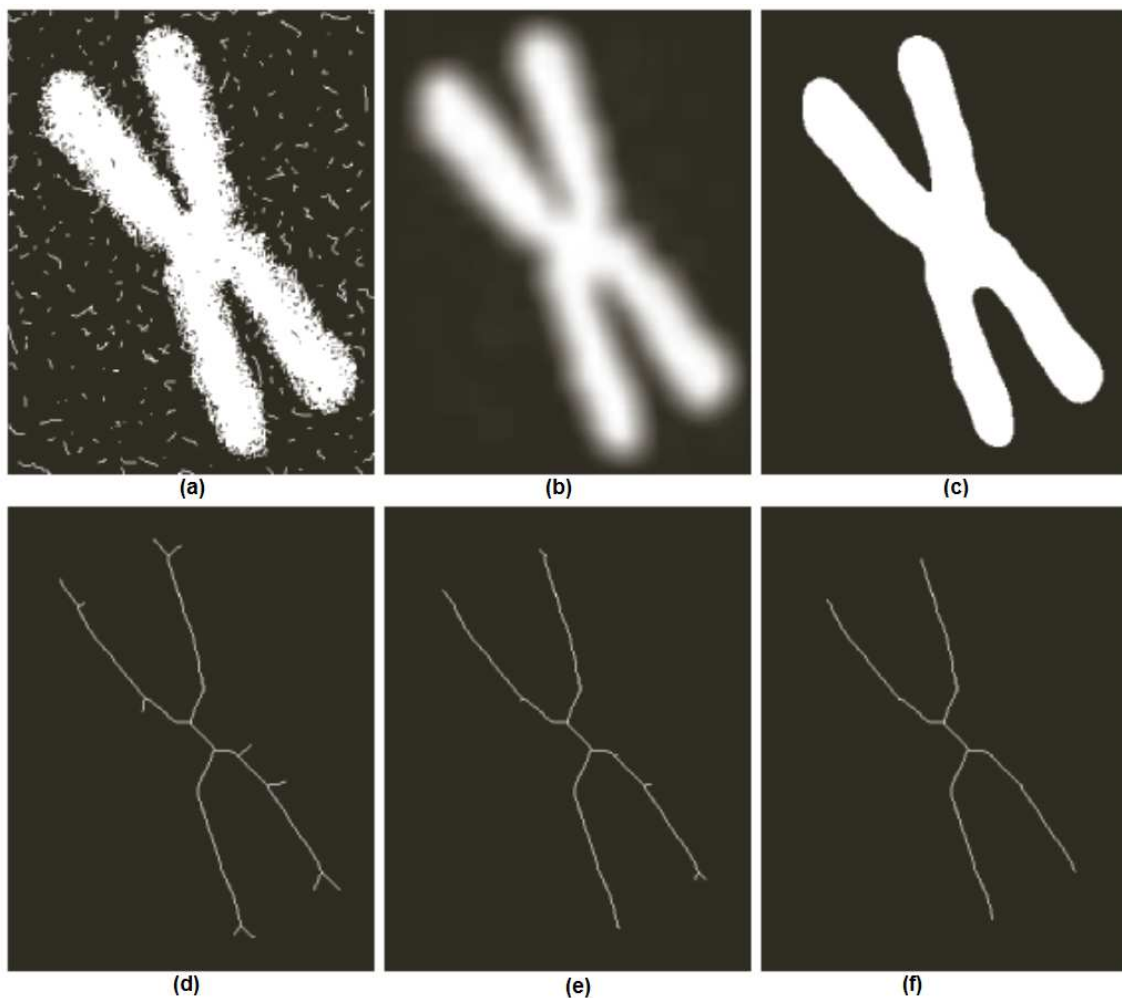


FIG. 8.17: (a) Image bruitée d'un chromosome humain. (b) Image lissée avec un filtre moyennneur 25×15 . (c) Image binarisée. (d) Squelette. (e) Squelette obtenue après l'application de 35 suppressions 'spur'. (f) Squelette obtenue après l'application de 30 autres suppressions 'spur'.