

CORRIGE TYPE DE L'EXAMEN FINAL EN TRAITEMENT D'IMAGE

I - QUESTIONS THEORIQUES

Exercice 1 (Opérations morphologiques : 6 points)

Soit **I** l'image binaire suivante :

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

Image binaire **I**

et soit **e₁** et **e₂** les deux types d'élément de structure suivants :

0	1	0
1	1	1
0	1	0

e₁

0	0	0
0	1	1
0	1	1

e₂

- 1) Calculer le résultat de l'érosion et de la dilatation de l'image binaire **I** en utilisant les deux types d'élément de structure **e₁** et **e₂**.
- 2) Calculer le résultat l'ouverture et de la fermeture de l'image **I** en utilisant les éléments de structure **e₁** et **e₂**.
- 3) Montrer comment on peut réaliser une opération de remplissage de trous d'un objet dans une image binaire en utilisant les opérations morphologiques.

Solution :

- 1) Calcul de l'érosion et de la dilatation de l'image binaire **I** par les éléments de structure **e₁** et **e₂** :

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

Érosion de **I** par **e₁**

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

Érosion de **I** par **e₂**

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

Dilatation de I par e1

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

Dilatation de I par e2

2) Calcul de l'ouverture et la fermeture de l'image I en utilisant les éléments de structure e1 et e2 :

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

Ouverture de I par e1

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

Ouverture de I par e2

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

Fermeture de I par e1

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

Fermeture de I par e2

3) Remplissage de trous d'un objet dans une image :

On commence par construire une matrice X_0 , composée de 0 (de la même taille que la région contenant A), sauf sur des endroits sur X_0 correspondant à des points du trou qui seront mis à 1.

$$X_k = (X_{k-1} \oplus B) \cap A^c, k = 1, 2, 3, \dots$$

où B est l'élément de structure défini par :

$$\begin{matrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{matrix}$$

L'algorithme termine à l'itération k, telle que: $X_k = X_{k-1}$

Exercice 2 (Seuillage d'une image : 6 points)

L'image **I** suivante est une image à niveaux de gris de taille 6×6 pixels dont les valeurs des niveaux de gris sont codées sur 4 bits. Cette image représente un cercle sur un fond sombre.

	1	2	3	4	5	6
1	2	1	14	14	1	2
2	1	14	12	12	14	1
3	14	12	8	8	12	14
4	14	12	8	8	12	14
5	1	14	12	12	14	1
6	2	1	14	14	1	2

- 1) Calculer le seuil obtenu par l'application de l'algorithme 1 de seuillage.
- 2) Considérant l'algorithme 2 de seuillage (algorithme d'Otsu) :
Donner les expressions mathématiques et calculer les éléments suivants :
 - l'histogramme normalisé de l'image $p_i, i = 0, \dots, L - 1$.
 - les sommes cumulatives $P_k(k), k = 0, \dots, L - 1$.
 - les moyennes cumulatives $M_c(k), k = 0, \dots, L - 1$.
 - la moyenne globale m_G .
 - la variance inter-classes $\sigma_B^2(k)$.
- 3) Quel est le meilleur seuil obtenu par application de l'algorithme d'Otsu.
- 4) Comparer les deux méthodes de seuillage.

Solution :

- 1) Les expressions mathématiques:

- Histogramme normalisé de l'image $p_k, k = 0, \dots, L - 1$:

$$p_k = n_k / (6 \times 6)$$

- Sommes cumulatives $P_1(k), k = 0, \dots, L - 1$:

$$P_1(k) = \sum_{i=0}^k p_i$$

- Moyennes cumulatives $M_c(k), k = 0, \dots, L - 1$:

$$M_c(k) = \sum_{i=0}^k i \times p_i$$

- Moyenne globale m_G :

$$m_G = \sum_{i=0}^{L-1} i \times p_i$$

- Variance interclasses $\sigma_B^2(k), k = 0, \dots, L - 1$:

$$\sigma_B^2(k) = \frac{(m_G \times P_1(k) - M_c(k))^2}{P_1(k) \times (1 - P_1(k))}$$

Application numérique :

k	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
p_i	0,000	0,080	0,040	0,000	0,000	0,000	0,000	0,000	0,040	0,000	0,000	0,000	0,080	0,000	0,120	0,000
$P_1(k)$	0,000	0,080	0,120	0,120	0,120	0,120	0,120	0,120	0,160	0,160	0,160	0,160	0,240	0,240	0,360	0,360
$M_c(k)$	0,000	0,080	0,160	0,160	0,160	0,160	0,160	0,160	0,480	0,480	0,480	0,480	1,440	1,440	3,120	3,120
m_G	3,120	3,120	3,120	3,120	3,120	3,120	3,120	3,120	3,120	3,120	3,120	3,120	3,120	3,120	3,120	3,120
$\sigma_B^2(k)$	non définie	0,391	0,435	0,435	0,435	0,435	0,435	0,435	0,003	0,003	0,003	0,003	2,619	2,619	17,306	17,306

2) Le meilleur seuil obtenu par l'application de la technique d'Otsu :

$$k^* = \operatorname{argmax}_k(\sigma_B^2(k)), k = 0, \dots, L - 1$$

Application numérique :

La valeur de k qui maximise $\sigma_B^2(k)$ est $k^* = 14.50$ avec $\sigma_B^2(k^*) = 17.306$.

II - IMPLEMENTATION MATLAB

Exercice 3 (Algorithme d'Otsu : 4 points)

Considérons une image en niveaux de gris I et soit L le nombre de niveaux de gris.

- Donner le programme Matlab qui permet de calculer le meilleur seuil obtenu par l'application de la technique d'Otsu.

Solution :

```
% Exercice 3 (Algorithme d'Otsu)
```

```
% Initialiser l'image
```

```
I = [2  1  14  14  1  2
      1  14  12  12  14  1
      14 12  8   8   12  14
      14 12  8   8   12  14
      1  14  12  12  14  1
      2  1  14  14  1  2];
```

```
% Calculer et afficher l'histogramme
```

```
I = uint8(I);
[N,X] = hist(I(:), [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]);
h = bar(X,N);
ylim([0 21]);
grid on;
```

```

% Algorithme d'Otsu
L = 16;
% histogramme normalisé de l'image p_i:
p_i = N/100;
% les sommes cumulatives P_k:
P_k = cumsum(p_i);
% les moyennes cumulatives M_c:
M_c = cumsum(p_i.*X);
% la moyenne globale m_G:
m_G = sum(X.*p_i);
% la variance inter-classes sigma_B^2:
sigma2B = (m_G.*P_k - M_c).^2./(P_k.*(1-P_k));
% plot sigmaB
plot(sigma2B)
% le meilleur seuil
[maxsig, k] = max(sigma2B)

```

Exercice 4 (Algorithme K-means : 4 points)

Ecrire un programme en langage Matlab qui permet de :

- 1) Lire une image en niveaux de gris.
- 2) Donner une procédure "k_means" qui implémente l'algorithme K-means.
- 3) Afficher les résultats incluant l'image originale ainsi que les résultats de segmentation obtenus par l'application de la méthode K-means en supposant que le paramètre K prend les valeurs 2, 3 et 4.

Solution :

```

% 1) lire une image en niveaux de gris.
I = imread('cameraman.tif');
X = double(I(:));

% 2) Appel de la procédure k_means
label2 = k_means(X',2);
label3 = k_means(X',3);
label4 = k_means(X',4);
Y2 = uint8(reshape(label2, size(I)));
Y3 = uint8(reshape(label3, size(I)));
Y4 = uint8(reshape(label4, size(I)));

% 3) Afficher l'image originale ainsi que les résultats de segmentation
obtenus par l'application de la méthode K-means avec le nombre de
cluster k = 2, 3, 4.

figure
subplot(2,2,1);
imshow(I)
subplot(2,2,2);
imshow(mat2gray(Y2))
subplot(2,2,3);
imshow(mat2gray(Y3))
subplot(2,2,4);
imshow(mat2gray(Y4))

```

Procédure K_means :

```
function label = k_means(X, k)
n = size(X,2);
last = 0;
% initialisation aléatoire
label = ceil(k*rand(1,n));
while any(label ~= last)
    % transformer les étiquettes en matrice d'indices
    E = sparse(1:n,label,1,n,k,n);
    % Calculer la moyenne m de chaque cluster
    m = X*(E*spdiags(1./sum(E,1)',0,k,k));
    last = label;
    % affecter les données aux plus proche centres
    [~,label] = max(bsxfun(@minus,m'*X,dot(m,m,1)'/2),[],1);
end
```