

# BIMA - Examen repartitionné 2

12 janvier 2021 - 14h00

Le barème, sur 20, donné à titre indicatif, est susceptible d'être modifié.

**Documents non autorisés. Téléphone portable éteint et rangé. Calculatrice autorisée.**

Durée de l'examen : 1 heure

## Exercice 1 Questions de cours (5 points)

1. Expliquer pourquoi la méthode de passage par zéro du laplacien permet d'obtenir des contours fins, au contraire d'un seuillage de la norme d'un gradient d'image.
2. Donner deux méthodes basées sur des filtres détecteur de contours d'ordre 1 qui permettent de calculer une carte binaire de contours (1=contour) avec l'assurance que les contours soient d'épaisseur 1.
3. Dans la méthode d'Otsu, sur quel critère la valeur du seuil est-elle choisie ?
4. Le descripteur SIFT est-il invariant à une rotation du plan image ? Justifier.
5. On s'intéresse à un problème d'identification de visages. Est-ce une bonne idée de calculer des descripteurs invariants en rotation ? Justifier.
6. L'ACP est-elle pertinente comme descripteur d'image ? Si oui, à quelle condition ?
7. Expliquer le principe de fonctionnement du détecteur de Moravec et ce qui le différencie du détecteur de Harris.
8. Définir la dérivée normalisée ? À quoi sert-elle ?

### Answer of exercise 1

1. Avec le détecteur laplacien, le pixel visité sera qualifié de contour s'il existe un passage par zéro dans le voisinage 3x3 de pixel. Ce critère est donc une opération locale qui tient compte du voisinage. On peut imaginer des configurations "dense" de contours (avec des alternances de passage par zéros) mais ce type de configuration ne correspond pas à un contour délimitant deux régions homogènes. En pratique donc les contours seront d'épaisseur 1.  
Un critère basé sur le seuillage de la norme du gradient ne garantit pas des contours fins puisque le seuillage est une opération ponctuelle qui ne tient pas compte du voisinage.
2. Les détecteur d'ordre 1 permettent de calculer la norme et la direction du gradient. Un pixel sera marqué contours si sa norme du gradient est maximale dans la direction du gradient. Une première méthode consiste à discrétiser l'orientation du gradient en 4 directions ce qui permet de déterminer les pixels voisins dans la direction discrète du gradient. La seconde méthode conserve la direction du gradient et interpole les valeurs de la norme du gradient à partir de celles des 8 pixels voisins.
3. On choisit la valeur de seuil qui minimise la variance intra-classe des deux classes obtenue par seuillage. (on peut aussi dire maximiser la variance inter-classe).

4. Oui il est invariant car chaque histogramme des orientations a été recalé par rapport à l'orientation dominante.
5. Non, dans ce contexte on suppose qu'on a une acquisition contrôlée avec les visages alignés verticalement, l'invariance va conduire à une perte d'information inutile qui pourrait être exploitée lors de la phase de reconnaissance.
6. L'ACP peut être un descripteur acceptable (on en a un bon exemple avec les Eigenfaces) à condition que l'essentiel de l'information soit porté par un petit nombre de directions principales.
7. Le détecteur de Moravec calcule la valeur minimale des 8 auto-corrélation correspondant aux 8 directions. Un pixel est noté coin si cette valeur est un maximal local. Le détecteur de Harris est une approximation du critère d'auto-corrélation de Moravec mais qui lui permet d'être invariant par rotation. Harris applique aussi un lissage qui lui permet d'être plus robuste au bruit.
8. La dérivée normalisée consiste à multiplier par  $s$  la dérivée d'une image lissée par un filtre gaussien de variance  $s$ . Cela permet d'avoir une réponse de la dérivée qui ne dépende pas de l'échelle  $s$ .

### Exercice 2 Images intégrales (5 points)

On rappelle la définition de l'image intégrale,  $I_{\text{int}}$ , pour une image  $I$ , illustrée à la figure 1 :

$$I_{\text{int}}(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y') \quad (1)$$

1. Donner la formule par récurrence qui permet de calculer  $I_{\text{int}}$  en un seul parcours de l'image  $I$ . Rappel : il faut raisonner en terme de soustraction/addition d'images intégrales.
2. Écrire un code Python qui prend en entrée une image  $I$  et retourne l'image intégrale de  $I$ .

Dans les questions suivantes, on considère l'image  $I$  dont l'image intégrale  $I_{\text{int}}$  est la suivante :

$$I_{\text{int}} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 7 & 9 & 11 \\ 3 & 7 & 10 & 14 & 17 \\ 4 & 9 & 14 & 19 & 23 \\ 5 & 11 & 17 & 23 & 28 \end{pmatrix}$$

3. On considère les 4 régions suivantes de  $I$  :

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

telles que  $A$  possède 2 lignes et 3 colonnes. Calculer les images intégrales de  $A$ ,  $B$ ,  $C$ , et  $D$ . Pour les régions  $B$ ,  $C$  et  $D$  on procédera par soustraction d'images intégrales en détaillant les calculs.

4. Donner la formule qui permet de retrouver  $I$  à partir de  $I_{\text{int}}$
5. À l'aide de la question précédente, calculer les valeurs de  $I$ .

### Answer of exercise 2

1.

$$I_{\text{int}}(x, -1) = 0 \quad x \geq -1$$

$$I_{\text{int}}(-1, y) = 0 \quad y \geq 0$$

$$I_{\text{int}}(x, y) = I_{\text{int}}(x, y-1) + I_{\text{int}}(x-1, y) - I_{\text{int}}(x-1, y-1) + I(x, y) \quad x \geq 0, y \geq 0$$

2. **def** I\_int(I):

n,m = I.shape

I\_int = np.zeros((n,m), dtype=int)

*# premiere ligne*

I\_int[0,0] = I[0,0]

**for** x **in** range(1,m):

I\_int[0,x] = I\_int[0,x-1] + I[0,x]

**for** y **in** range(1,n):        *# premiere colonne*

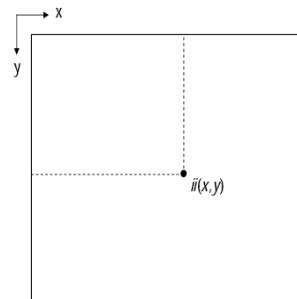
I\_int[y,0] = I\_int[y-1,0] + I[y,0]

**for** x **in** range(1,m):

I\_int[y,x] = I\_int[y-1,x] + I\_int[y,x-1] - I\_int[y-1,x-1] + I[y,x]

**return** I\_int3.  $A_{\text{int}} = I_{\text{int}}(2, 1) = 7$ .  $B_{\text{int}} = 11 - 7 = 4$ ,  $C_{\text{int}} = 17 - 7 = 10$ ,  $D_{\text{int}} = 28 - B_{\text{int}} - C_{\text{int}} - A_{\text{int}} = 28 - 4 - 10 - 7 = 7$ 4.  $I(x, y) = I_{\text{int}}(x, y) - I_{\text{int}}(x, y-1) - I_{\text{int}}(x-1, y) + I_{\text{int}}(x-1, y-1)$ 

$$5. \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 & 1 \\ 1 & 2 & 0 & 2 & 1 \\ 1 & 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

FIGURE 1 – Image intégrale :  $I_{\text{int}}(x, y)$  est la somme des valeurs de  $I$  dans le carré en pointillé en haut à gauche.**Exercice 3      Filtrage fréquentiel et détection de contours (5 points)**

On souhaite réaliser un détecteur de contours à l'aide d'un filtrage dans le domaine fréquentiel.

1. Soit  $P_a$  la fonction de transfert de ce filtre fréquentiel. Quelle est sa nature (passe-haut/bas) ? Justifier.

2. Proposer un filtre idéal. On donnera les valeurs de  $P_a$ .
3. Comment choisir fréquence de coupure du filtre ? Justifier.
4. Écrire une fonction Python qui prend en entrée une image, retourne cette image filtrée par  $P_a$  dans le domaine fréquentiel.
5. On considère le filtre dont la fonction de transfert est  $P_b = 1 - P_a$ . Quelle est la nature de ce filtre ? Si on filtre une image avec  $P_b$  quel effet constatera-t-on sur l'image filtrée ?
6. Soient les images  $I_a$  et  $I_b$  respectivement filtrées par  $P_a$  et  $P_b$  à partir de  $I$ . Démontrer que  $I = I_a + I_b$
7. Soit le filtre de horizontal de Prewitt :  $P = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}$  Pour une image  $I$ , exprimer  $I \star P(x, y)$  en fonction de  $I(x \pm 1, y)$  et  $I(x \pm 1, y \pm 1)$ . On prendra comme convention pour  $x$  et  $y$  celle de la figure 1.
8. Montrer que  $I \star P$  peut s'exprimer comme combinaison linéaire de décalage de l'image  $I$  d'une colonne vers la gauche ou vers la droite, d'une ligne vers le haut ou vers le bas. On ignorera les conditions aux bords.

### Answer of exercise 3

1. Les contours sont des structures images qui sont caractérisées par un changement brusque des valeurs de l'image dans une direction.  $P_a$  doit donc être un passe-haut dans cette direction.
2. En s'inspirant de ce qui a été vu dans l'UE, on peut proposer l'un des deux filtres suivants :

$$\begin{aligned}
 P_a(f, g) &= \begin{cases} 1 & \text{si } \max(f, g) \geq d \\ 0 & \text{sinon} \end{cases} \\
 &= \begin{cases} 1 & \text{si } \sqrt{f^2 + g^2} \geq d \\ 0 & \text{sinon} \end{cases}
 \end{aligned}$$

$d$  est la fréquence de coupure.

3. La fréquence de coupure doit être choisie suffisamment haute car les contours sont des structure à petites échelles. La valeur dépend de l'échelle des contours : plus la transition est rapide, plus la fréquence de coupure doit être élevée.
4. 

```
def DectCont(I):
    from numpy.fft import fft2, ifft2, fftshift
    n,m = I.shape
    r = max(n,m)-40 # frequence haute

    Fa = np.zeros((n,m))
    Fa[(n-r)//2:(n-r)//2+r, (m-r)//2:(m-r)//2+r] = np.ones((r,r))

    Fa = fftshift(Fa) # penaliser si oubli

    return ifft2(fft2(lena)*Fa)
```
5.  $P_b$  un filtre passe-bas. Une image filtrée avec  $P_b$  sera lissée/floutée.

6.

$$\begin{aligned}
I_a + I_b &= \text{FT}^{-1}(\text{FT}(I) \times P_a) + \text{FT}^{-1}(\text{FT}(I) \times P_b) \\
&= \text{FT}^{-1}(\text{FT}(I) \times P_a + \text{FT}(I) \times P_b) \\
&= \text{FT}^{-1}(\text{FT}(I) \times (P_a + P_b)) \\
&= \text{FT}^{-1}(\text{FT}(I)) = I
\end{aligned}$$

car  $P_a + P_b = 1$  et la transformée de Fourier inverse est linéaire.

7.  $I \star P(x, y) = I(x+1, y-1) - I(x-1, y-1) + I(x+1, y) - I(x-1, y) + I(x+1, y+1) - I(x-1, y+1)$
8. On remarque que  $J(x, y) = I(x+1, y)$  correspond à un décalage de l'image  $I$  d'une colonne vers la gauche : le pixel en  $x, y$  de l'image  $J$  a la valeur du pixel qui était en colonne  $x+1$ . En faisant le même raisonnement pour les autres cas on trouve que :  $I(x-1, y)$  correspond à un décalage d'une colonne vers la droite,  $I(x, y+1)$  correspond à un décalage d'une ligne vers le haut et  $I(x, y-1)$  un décalage d'une ligne vers le bas. On note ces décalages par les opérateurs respectifs :  $S_l, S_r, S_u$  et  $S_d$ . On a alors :  $I \star P = S_d(J) + J + S_u(J)$  avec  $J = S_l(I) - S_r(I)$ .

### Exercice 4 Reconnaissance de formes (5 points)

On dispose d'un ensemble  $E$  de  $N$  individus. Un individu  $n$  est décrit par un vecteur  $X_n \in \mathbb{R}^d$  et est associé à une classe décrite par un scalaire  $Y_n \in \{1, \dots, K\}$ .

L'objectif est de déterminer la classe  $y$  d'un individu  $x \in \mathbb{R}^d$  qui n'appartient pas à l'ensemble  $E$ .

1. Rappeler quel critère est optimisé dans le cadre de l'ACP, comment on le calcule en pratique, et ce qu'il signifie en terme de représentation des données.
2. Dans le cadre du problème de reconnaissance de forme décrit en préambule de l'exercice, décrire la mise en œuvre d'une méthode basée sur l'ACP. On donnera un premier algorithme pour décrire la phase d'apprentissage et un second pour la phase de reconnaissance. La phase d'apprentissage est-elle supervisée ?
3. Rappeler quel critère est optimisé dans le cadre de l'ALD, comment on le calcule en pratique, et ce qu'il signifie en terme de représentation des données
4. Proposer une méthode qui met en œuvre l'ALD dans le problème de reconnaissance de forme décrit dans l'exercice. Donner l'algorithme d'apprentissage, l'algorithme de reconnaissance. La phase d'apprentissage est-elle supervisée ?

#### Answer of exercise 4

1. L'ACP consiste à trouver une base orthogonale telle que les données projetées sur chaque élément de la base sont de variance maximale.  
Cela revient à calculer les vecteurs propres de la matrice de variance-covariance des données centrées.  
Dans ce nouvel espace de représentation les données projetées sont mieux dispersées (de variance maximale) et davantage décorréliées entre chaque paire de directions.
2. **Apprentissage :**
  - (a) Calcul de la matrice de variance-covariances des données de  $E$  :  $\Sigma = \frac{1}{N-1} X X^T$  avec  $X = (X_1 - g, X_2 - g, \dots, X_N - g)$  et  $g = \frac{1}{N} \sum_n X_n$ .

- (b) Calcul des  $M < N$  vecteurs propres (directions principales)  $V_n, n = 1 \dots K$  de  $\Sigma$  ordonnées selon leur valeurs propres associées décroissantes ( $\lambda_1 \geq \lambda_2 \geq \dots$ ).  $M$  est un hyper-paramètre.

- (c) Projection des données sur les  $M \ll d$  directions :  $\pi(X_n) = (X_n^T V_1, \dots, X_n^T V_M)$ , l'individu  $n$  est décrit par  $M$  scalaires.

La phase d'apprentissage est non supervisée car elle n'utilise pas les classes  $Y_i$  associées aux données.

**Reconnaissance :**

- (a) L'individu  $x$  est projeté sur les  $M$  directions principales :  $\pi(x) = (x^T V_1, \dots, x^T V_M)$ .
- (b) On cherche l'individu projetée  $\pi(X_i)$  qui est le plus proche dans l'ensemble  $E$  (métrique euclidienne) :  $i = \operatorname{argmin}_n \|\pi(X_n) - \pi(x)\|^2$ .
- (c) On attribue à  $x$  la classe  $Y_i$ .

3. L'ALD consiste à trouver une base orthogonale telle que les données projetées sur les éléments de la base ont une variance intra-classe **minimale** et une variance inter-classe **maximale**.

Cela revient à calculer les vecteurs propres de la matrice  $\Sigma^{-1}B$  avec  $\Sigma$  telle que défini dans la question 2 et  $B$  est la matrice des variances inter-classe.

Dans cet espace de représentation les données de même classe sont mieux agglomérées (elles forment des *clusters*) et les clusters sont mieux séparés.

4. **Apprentissage :**

- (a) Calcul de  $\Sigma$  et  $B = \frac{1}{N} \sum_{k=1}^K N_k (g_k - g)(g_k - g)^T$  avec  $g_k = \frac{1}{N_k} \sum_{n|Y_n=k} X_n$  et  $N_k = \sum_{n|Y_n=k} 1$ .
- (b) Calcul des vecteurs propres ( $V_n$ ) de  $\Sigma^{-1}B$  ordonnés selon par valeurs propres associées décroissantes.

La phase d'apprentissage est supervisée car elle utilise les classes  $Y_i$  associées aux données.

5. **Reconnaissance :**

- (a) Projection de l'individu  $x$  sur la base d'ALD.
- (b) On attribue à  $x$  la classe  $k$  la plus proche dans la base de l'ALD, c'est-à-dire qui minimise  $\|\pi(x) - \pi(g_n)\|$ .