

Exercice 1.

1) Implémentation de la transformation linéaire avec utilisation d'une LUT (Look Up Table):

```
/* Initialisation de la LUT */  
Pour i allant de 0 à 255  
    LUT(i) ← 255*(i-minI)/(maxI-minI);  
/* Initialisation de la LUT */  
Pour i allant de 1 à nblig  
    Pour j allant de 1 à nbcol  
        I'(i,j) ← LUT(I(i,j));
```

2) La taille d'une image RGB de taille 512×512 codée sur 24 bits :

$$\text{Taille de l'image} = 512 \times 512 \times 24 / 8 \text{ octets}$$

3) Un algorithme qui calcule l'histogramme d'une image :

```
pour i allant de 1 à hauteur faire  
    pour j allant de 1 à largeur faire  
        histo(imgage(i,j)) ← histo(imgage(i,j)) + 1  
    fin pour  
fin pour
```

- **histo** : tableau contenant le résultat final.
- **hauteur** : l'hauteur de l'image.
- **largeur** : la largeur de l'image.
- **image** : matrice qui contient les valeurs de pixels.

4) La soustraction de fond dans les vidéos en utilisant la distribution gaussienne :

- Le pixel est modélisé par une **gaussienne** centrée sur μ et un écart type σ . Pour estimer la moyenne et l'écart-type de la gaussienne associée au pixel, on a besoin d'une série de **N** frames **sans mouvement**, c-à-d. ne contenant que le fond:

$$\mu(x,y) = \frac{1}{N} \sum_{t=0}^{N-1} I_t(x,y)$$

$$\sigma(x,y) = \frac{1}{N} \sum_{t=0}^{N-1} (I_t(x,y) - \mu(x,y))^2$$

- La détection de mouvement se fait à l'aide de l'opération suivante:

$$c(x,y) = \begin{cases} 1 & \text{si } p(I_t(x,y)) > \delta \\ 0 & \text{sinon} \end{cases}$$

5) Algorithme de Horn et Schunck.

1. Initialiser v_x et v_y pour tous les pixels.

2. Mettre à jour itérativement les valeurs de v_x et v_y par les equations:

$$v_x^{(t+1)} = \bar{v}_x^{(t)} - I_x \frac{I_x \bar{v}_x^{(t)} + I_y \bar{v}_y^{(t)} + I_t}{\lambda^2 + I_x^2 + I_y^2},$$

$$v_y^{(t+1)} = \bar{v}_y^{(t)} - I_y \frac{I_x \bar{v}_x^{(t)} + I_y \bar{v}_y^{(t)} + I_t}{\lambda^2 + I_x^2 + I_y^2},$$

Ex. 1 : 01 point pour chaque réponse

Exercice 02. Répondre avec vrai ou faux

1. L'échantillonnage est limité par la quantité de tons. (**faux**)
2. La quantification est limitée par la quantité de niveaux de gris. (**vrai**)
3. Le nombre de bit pour représenter un image = $M.N.K$ où N, M : la dimension de l'image, L : nombre de niveau de gris, et $K = \log_2(L)$. (**vrai**)
4. PNG est un format de fichiers d'images sans compression. (**faux**)
5. PPM est un format de fichiers d'images qui permet de sauvegarder des images en niveaux de gris 8 bits. (**faux**)
6. La technique d'égalisation d'histogramme est utilisée pour réduire le bruit dans une image. (**faux**)
7. Le filtre médian est un filtre robuste dans la présence des valeurs aberrantes. (**vrai**)
8. L'opérateur Laplacien ne peut pas être utiliser pour accentuer les détails de l'image. (**faux**)
9. La transformée de Hough est une technique utilisée pour extraire des lignes. (**vrai**)
10. Les régions où le gradient de l'image s'annule ont généralement une valeur de mouvement très grandes. (**faux**)

Ex. 2 : 0.5 point pour chaque réponse

Exercice 03.

1) Ecrire un programme **Matlab itératif** qui permet de générer N nombres entiers aléatoires dans l'intervalle $[a, b]$, où a, b et N sont des nombres réels donnés.

2) Donner une **version vectorisée** de ce programme.

Exemple :

Entrée : $a = 5$; $b = 30$; $N = 5$;
Résultat : 9 16 28 25 29

Solution :

1) Version itérative :

```
a = 5;  
b = 10;  
N = 5;  
x = zeros(N,1);  
for i=1:n  
    x(i) = 2*round( a + (b/2-1-a)*rand(1,1) ) + 1;  
end
```

2) Version vectorisée :

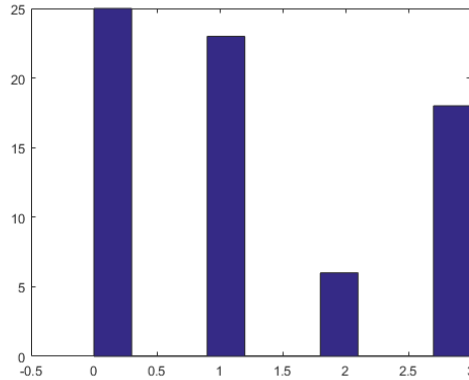
```
a = 5;  
b = 10;  
N = 5;  
x = 2*round( a + (b/2-1-a)*rand(1,N) ) + 1;
```

Ex. 3 : 2.5 points pour chaque réponse

Exercice 04.

1- L'histogramme de l'image :

```
Commande Matlab: >> hist(I(:))
Pour plus d'infos: >> doc hist
```



- 2- Cette image n'est pas contrastée. On peut remarquer l'existence de deux objets dans l'image :
- (i) L'arrière-plan représenté par les deux niveaux 0 et 1
 - (ii) L'objet qui est représenté par les deux niveaux de gris 2 et 3.

3- Le passage de l'image originale (I) vers l'image finale (F) peut être fait sur deux étapes :

(i) Un simple seuillage :

$$F(x, y) = \begin{cases} 1 & \text{si } I(x, y) > 2 \\ 0 & \text{sinon} \end{cases}$$

(ii) Suivi d'un filtre moyenne de 3 × 3.

4- Le contour de l'image binaire.

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

5- Application du masque sur l'image :

-3	5	-5	5	-5	5	-5	6
5	2	-1	-1	0	1	1	3
-7	-4	1	0	2	-16	5	0
1	11	4	3	5	5	10	-2
-10	8	2	1	1	3	10	1
2	2	4	-8	3	7	-12	-5
-4	-1	12	8	10	-3	4	-1
-1	-4	-8	-2	0	-6	-2	-1
0	0	-1	-2	-2	-1	0	0

Conclusion : ce masque est un masque passe bas qui pourra être utilisé pour la détection de contour ainsi que l'amélioration du contraste.

Ex. 4 : 01 points pour chaque réponse