

TP02 : Transformations géométriques d'image (IGE43)

Enseignante: Dr. E. Zigh épouse Slimane

Remarque 1 : Merci de noter que les comptes rendus des TP seront remis par binômes, **un seul étudiant** parmi le binôme se chargera de déposer le TP sur la plate forme Moodle. Le nom du fichier ou dossier (ou dossier compressé) déposé doit comporter les noms des binômes.

Remarque 2 : Merci aussi de noter que vous avez le choix entre utiliser **Matlab ou Python** pour effectuer les TP de traitement d'image.

Remarque 3 : Les réponses aux questions doivent être reportées sur un fichier compte rendu qui sera déposé avec le TP (TP = les scripts en matlab ou python) sur la plate forme.

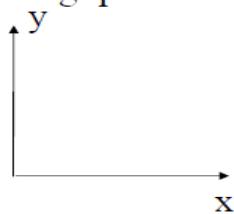
Remarque 4 : Vous devez écrire un script pour chaque exercice.

Rappel :

1/ Système de convention

Le système de convention des axes est le suivant :

En analogique :



En discret :



✓ la ligne 0 d'une image est celle située en haut.

2/ Il existe deux types de transformations sur les pixels de l'image :

- 2.1 les transformations géométriques qui modifient les positions des pixels
- 2.2 les transformations radiométriques qui modifient les intensités des pixels

3/ Opérations géométriques sur une image

Les opérations géométriques classiques sont permises avec la boîte à outils de traitement d'images : rotation, changement de taille, découpage.

3.1 Exercice1 :

Ecrire un script qui permet de :

lire l'image onion.png.

- Trouver la résolution spatiale et spectrale de cette image.

- Appliquer sur cette image, une translation horizontale de 20 pixels à droite.

-Appliquer sur cette image, une translation de 50 lignes et 100 colonnes à droite.

-Appliquer sur cette image une translation qui permet de cacher totalement le poivron jaune.

-Afficher les résultats et commenter.

3.2 Exercice 2 :

Ecrire un script qui permet, à partir de l'image d'origine donnée, lui appliquer une rotation anti horaire de 45° , selon un repère placé au centre de l'image.

-Afficher les résultats et donner des applications pratiques où la translation et /ou la rotation sont utiles.

Rappel :

Le sous échantillonnage appelé réduction d'échelle. Ceux sont les petits détails qui sont intéressants. Ils risquent de disparaître ou au contraire prendre la place d'une information importante.

Le sous échantillonnage est l'opération inverse du sur-échantillonnage (zoom) qui se fait par différentes méthodes telles que :

- La méthode du plus proche voisin.
 - La méthode de l'approximation bilinéaire.
 - La méthode de l'approximation bicubique.

https://www.youtube.com/watch?v=LPOh_Cymi1Q&ab_channel=AhladKumar

3.3 Exercice 3 :

Ecrire le script suivant :

```

clc
close all
clear all

I = imread('rose.jpeg');

[j, k,c] = size(I);

scale = 5;
x_new = round(j/scale);
y_new = round(k/scale);

M = zeros(x_new,y_new,c);

for ch = 1:c
    for count1 = 1:x_new
        for count2 = 1:y_new
            M(count1,count2,ch) =
I(count1*scale,count2*scale,ch);
        end
    end
end

```

```
end
```

```
subplot(1,2,1); imagesc(uint8(I)); axis tight;
subplot(1,2,2); imagesc(uint8(M)); axis tight;
```

- 1/ Expliquer chaque ligne de ce programme (sous forme de commentaire).
- 2/ Changer la valeur scale à 26, que remarquez vous ?, puis changer là à 17, commentez et expliquez ?
- 3/ Après avoir bien expliqué le fonctionnement de l'algorithme relatif à ce script, proposez une modification de ce programme pour qu'il fonctionne pour n'importe quelle valeur de scale tout en expliquant.

3.4 Exercice 4 : Ecrire un script qui permet de faire un sur échantillonnage sur l'image ‘rose’ en utilisant la méthode du plus proche voisin.

3.5 Exercice 5

-Veuillez saisir le programme Matlab suivant :

```
x = imread('rose.jpeg');
[r,c,w] = size(x);

B = 256;

q = 4;
l = B / q ;
E = zeros(256, 1);
for i = 0:255,
    E(i+1, 1) = floor(i / l) *l + 1/2 ;
End

figure; plot(0:255,E);

y = zeros(size(x));
for ch = 1:w1
    for i = 1:r1,
        for j = 1:c1,
            y(i, j,ch) = E(x(i,j,ch) + 1);
        end
    end
end

figure; imagesc(y)
```

- 1/ Exécuter le.

2/Corriger si jamais il y'a des erreurs de compilation. Mentionner sous forme de commentaires, à côté de chaque ligne corrigée une explication relative à cette correction.

3/Une fois que le programme s'exécute sans erreurs. Combien de figures obtenez-vous en sortie ?.

4/Que représente chaque figure ?. Donnez un titre à chaque figure.

5/Exécutez le programme une deuxième fois pour q=8, puis pour q= 14.

6/ Modifier le programme pour donner à l'utilisateur l'accès pour insérer la valeur de q selon son choix.

7/ Que fait ce programme. (Répondez sous forme de commentaire inséré au début du programme)

8/ Que représente chacune des variables q, l et B.

9/ Que fait la partie suivante du programme (expliquez) :

```
E = zeros(256, 1);
for i = 0:255,
    E(i+1, 1) = floor(i / l) *l + l/2 ;
End
```

11/ Que fait la partie suivante du programme (expliquez) :

```
y = zeros(size(x));
for ch = 1:w1
    for i = 1:r1,
        for j = 1:c1,
            y(i, j, ch) = E(x(i, j, ch) + 1);
        end
    end
end
```

10/ Quelle est la différence entre l'opération effectuée sur l'image dans cet exercice et l'opération échantillonnage qu'on a vu auparavant.