

Compte Rendu TP2

**Transformation Géométriques d'Image Sous
MATLAB**

IGE43 – Groupe 1

Présenté par :

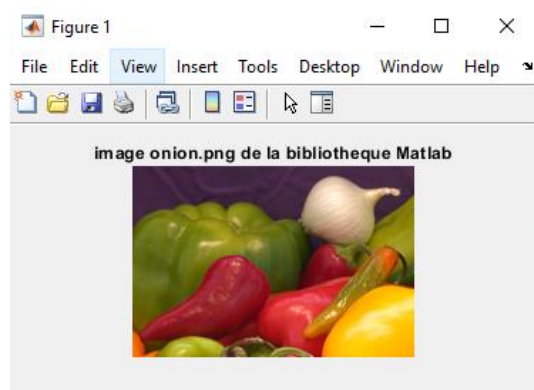
- ❖ HALLA Senia
- ❖ KAID Nassima

Mardi 25 Mai 2021

Exercice 1 :

1. Lire l'image :

L'image onion.png est dans la bibliothèque Matlab

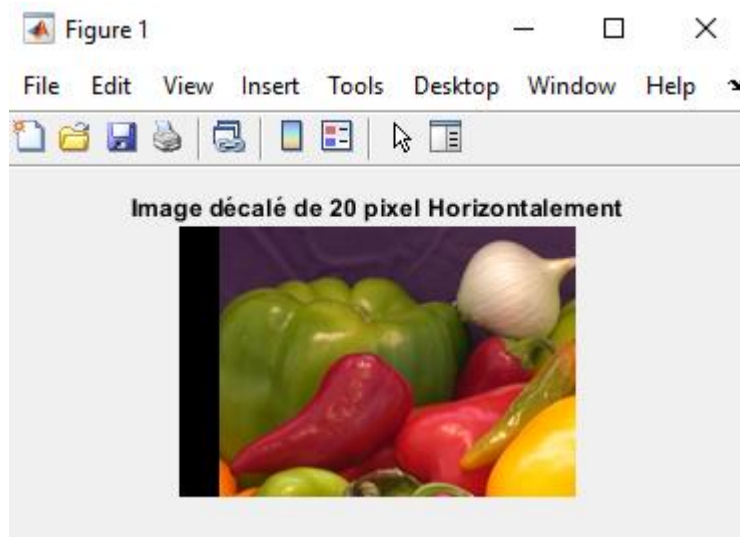


2. La résolution spatiale et la résolution spectrale de cette image :

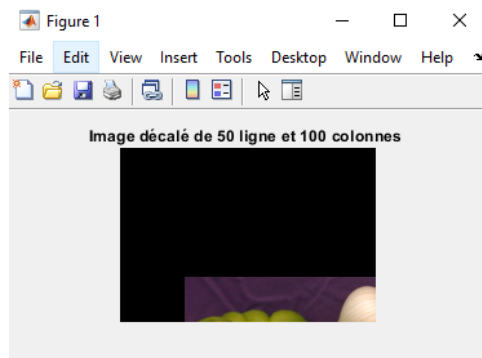
Res_spatial = 135 x 19

Res_spectral = 3

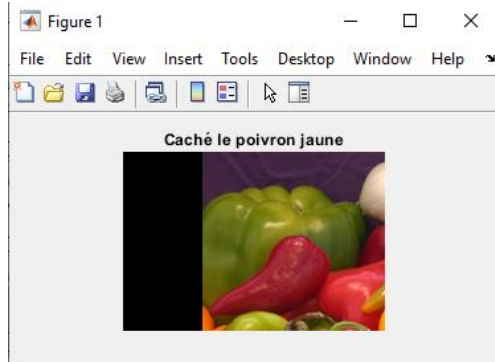
3. Translation horizontale de 20 pixels à droite :



4. Translation de 50 lignes et 100 colonnes à droite :



5. Translation permettant de cacher le poivron jaune :



6. Commentaire :

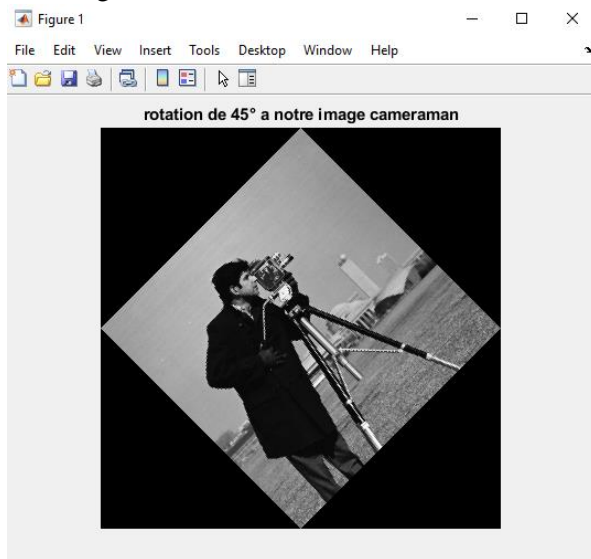
- Une opération de translation décale une image d'un nombre spécifique de pixels dans la direction x ou y, ou les deux. (Respectivement décalage par ligne ou colonnes ou les deux)
- La translation garde la taille originale de l'image
- On décale chaque pixel d'un saut (intervalle défini),
- Les cases vides issues des pixels décalés auront une valeur de 0 (noir) par défaut avec possibilité de changer la couleur

Exercice 2 :

1. Appliquer une rotation anti horaires de 45° :



2. Affichage du résultat :



3. Application Pratiques du translation et rotation :

- Cacher une partie indésirable d'une image en faisant une translation : par exemple avant de donner les images de radio a un algorithme de classification nous devons cacher le nom du patient placé en haut. De un pour garder son anonymat et de deux pour ne pas fausser le traitement de notre algorithme
- La rotation d'une image peut nous donner une meilleure vision de certain détail : par exemple en faisant une petite rotation d'une image prise par une caméra de sécurité nous pouvons mieux identifier le visage d'une personne ou bien la plaque d'immatriculation d'une voiture
- Mosaïque d'image comme la photo ci-dessous :

Mosaïque d'images



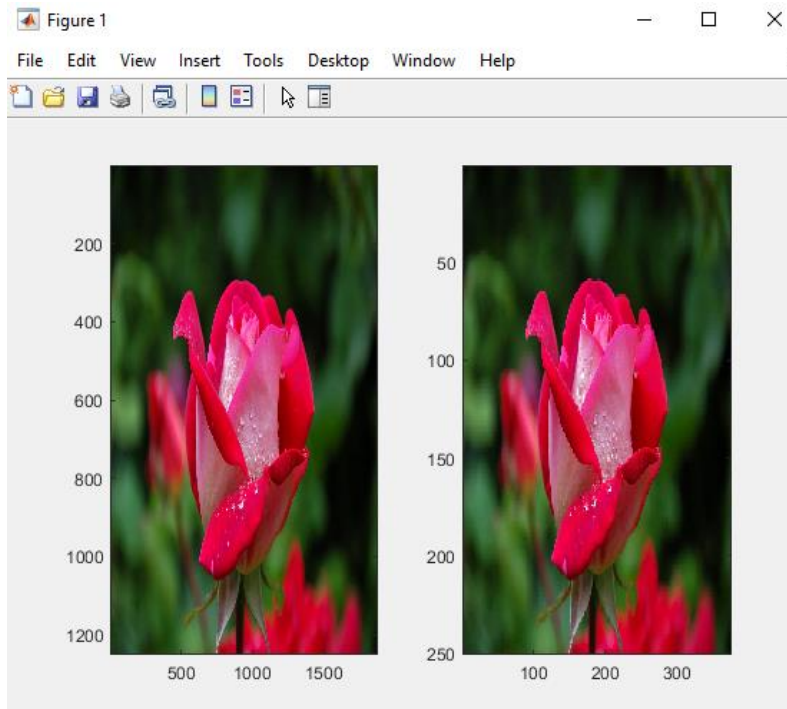
Figure 1: Left: First input image for mosaic. Right: Second input image for mosaic.



Exercice 3 :

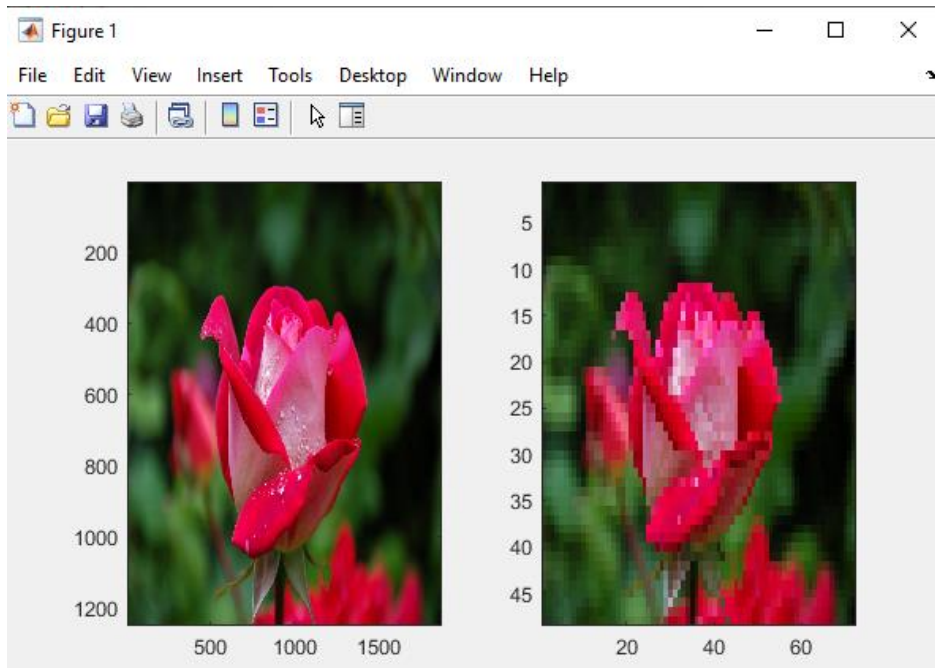
1. (Explication des lignes au niveau du script) :

Scale =5



2. Changement de la valeur du scale à 26, puis à 17 :

Scale = 26



Scale = 17: Error

Index in position 2 exceeds array bounds (must not exceed 1880).

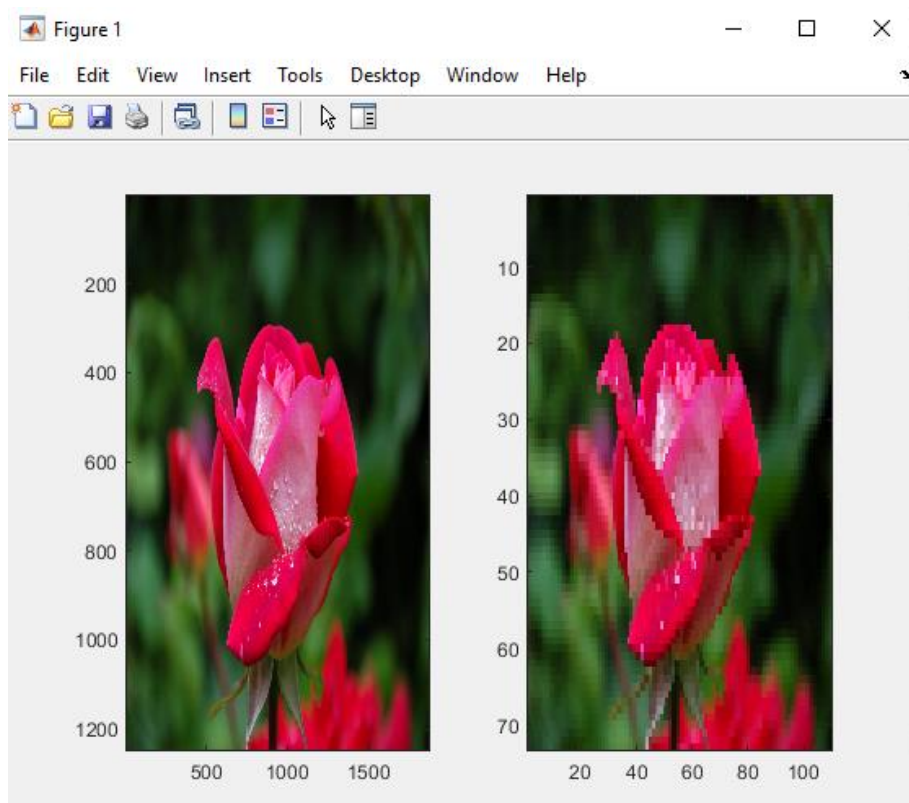
–Explication au niveau du script -

3. Explication du fonctionnement de l'algorithme

- Il s'agit d'un algorithme de sous-échantillonnage (Downsampling), ou on prend la taille de l'image (nombre de pixel ligne/colonnes) et on le divise sur un scale (qu'on définit)
- Après on reconstruit l'image avec le nouveau nombre de pixel
- Plus le 'scale' est grand ; en reconstruisant notre image ; on perd de l'information ce qui explique la mauvaise qualité de l'image résultante (réduction d'échelle)

4. Proposition de Modification pour qu'il fonctionne pour n'importe quelle valeur de scale :

On change round par floor et ça marchera pour toute valeur



Exercice 4 :

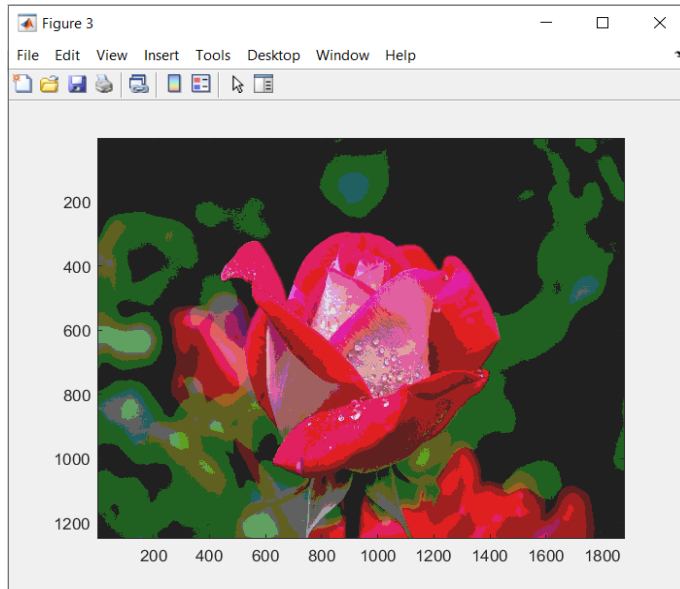
1. Échantillonnage avec la méthode du plus proche voisin : -Explication au niveau du script-



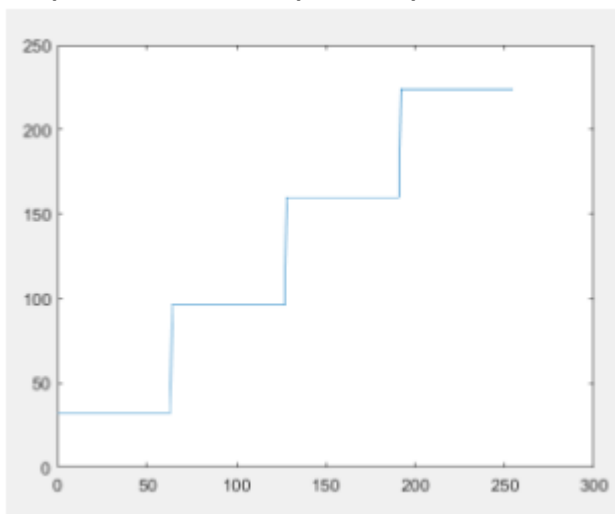
Exercice 5 :

1. Exécution du script donné :
2. Correction des erreurs de compilation :
3. On obtient : 2 Figures

L'image après le processus de quantification :

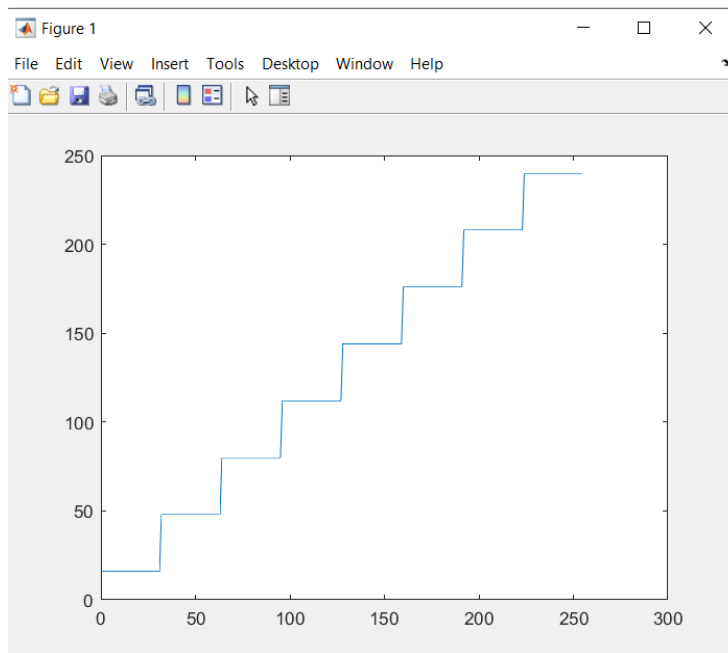
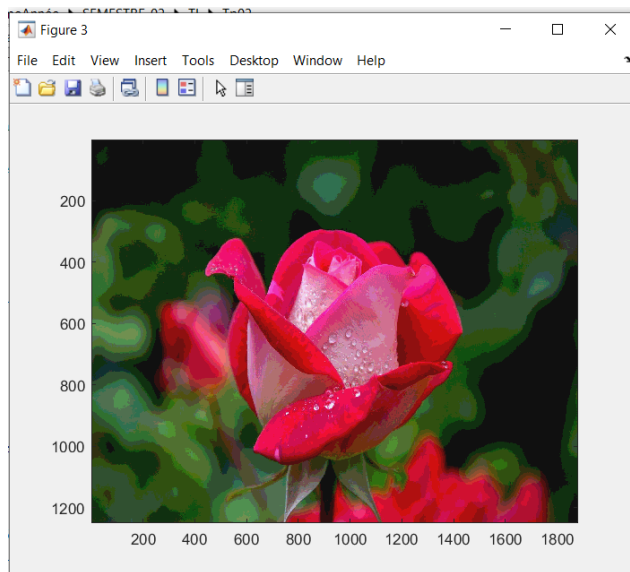


La fonction de quantification qui sépare entre les valeurs continues et les valeurs discrètes chaque 64 valeurs est représenté par un niveau :

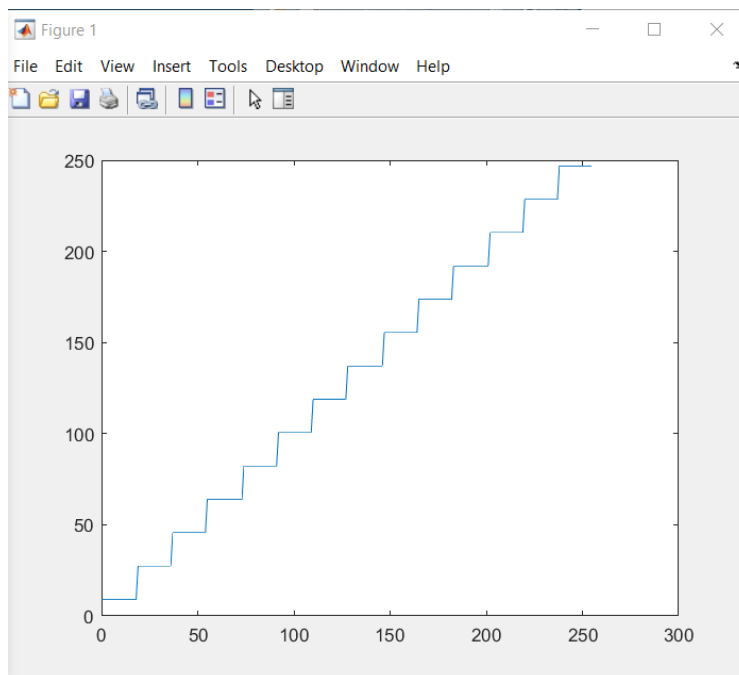
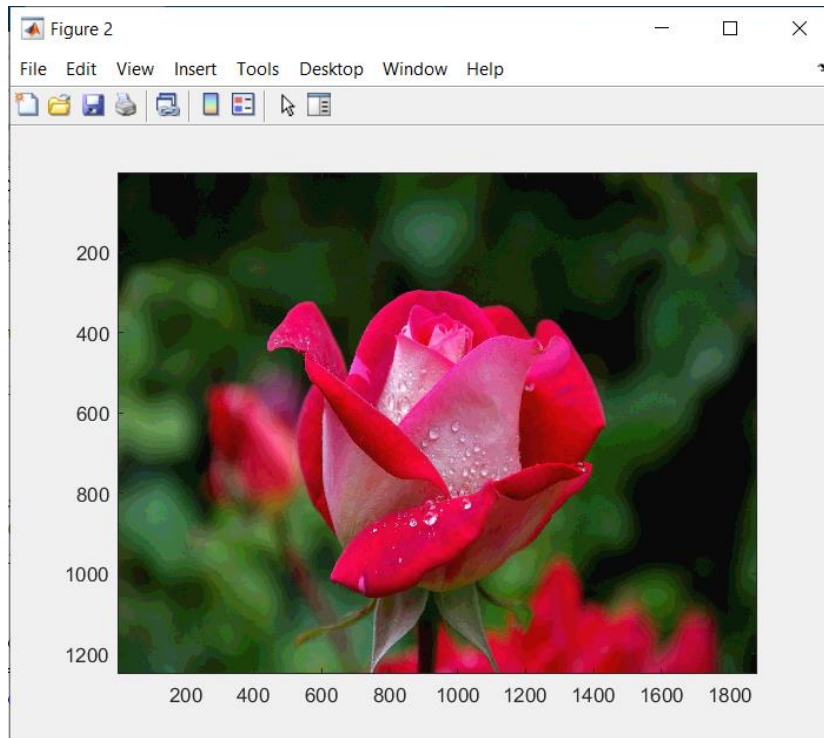


4. Explication de chaque figure : -Au niveau du script-
5. Exécution du programme pour $q=8$ puis pour $q = 14$:

- Pour $q = 8$:



- Pour $q = 14$:



6. Modification du programme en donnant accès à l'utilisateur pour insérer la valeur de q :
`q=input("Entrez q: ")`
7. (Explication du programme au niveau du script) :
8. Signification des valeur q, l et B

B =256 valeurs des niveaux de couleur de l'image

q est le nombre de niveau de valeurs qu'on veut de discrétiser

l : facteur de quantification

9. (Explication de la partie donnée du programme au niveau du script) :
10. La différence entre l'opération effectuée sur l'image dans cet exercice et l'opération d'échantillonnage vu auparavant :

L'échantillonnage joue sur la résolution spatiale qui est le nombre de pixels (taille de l'image) de l'autre côté on retrouve la quantification qui joue sur la résolution spectrale qui représente l'intensité des pixels (la profondeur de l'image)

Conclusion

Les opérations géométriques ont pour but de modifier la position des informations contenues dans l'image sans modifier la valeur /intensité du pixel :

Les translations et rotations qui présentent un intérêt dans de nombreuses applications : on peut citer

- correction des distorsions géométriques liées au matériel optique. La distorsion la plus fréquente est celle introduite par l'objectif lui-même (effet de "tonneau" ou de "barillet").

- correction de la perspective d'une scène 3D projetée sur un capteur 2D. Pour envisager des mesures précises des objets présents dans la scène, il peut être utile de restituer une image corrigée par une opération géométrique adaptée.

L'échantillonnage :

- **Le sur échantillonnage "down sampling"** appelé réduction d'échelle ou on réduit le nombre des pixels (selon un facteur 'scale'). En reconstruisant l'image certains détails risquent de disparaître ou bien des informations importantes si notre scale est grand
- **Le sous échantillonnage "upsampling"**: appelé zoom , c'est l'opération inverse du sous-échantillonnage ou on agrandit notre image par un facteur

La quantification :

Si l'échantillonnage modifie la taille de l'image (nombre de pixel), la quantification est le processus qui modifie sa profondeur (l'intensité des pixels)