M2 STIM

Compte-rendu Traitement d'images

HAO Chang

Sommaire

Tp1. Amélioration d'une photo prise sans flash

I. Ajouter du bruit

En appliquent la fonction ***imnoise***, on ajoute deux modèles de bruits sur l'image :

- Le bruit gaussien : le paramètre est la variante du bruit gaussien. Plus le paramètre est élevé, plus l'image est bruitée.

Figure 1-1 : gauche : variance =0.1 ; droit : variance = 0.5

- Le bruit poivre et sel : le paramètre est la densité du bruit poivre et sel. Plus le paramètre est élevé, plus l'image est bruitée.

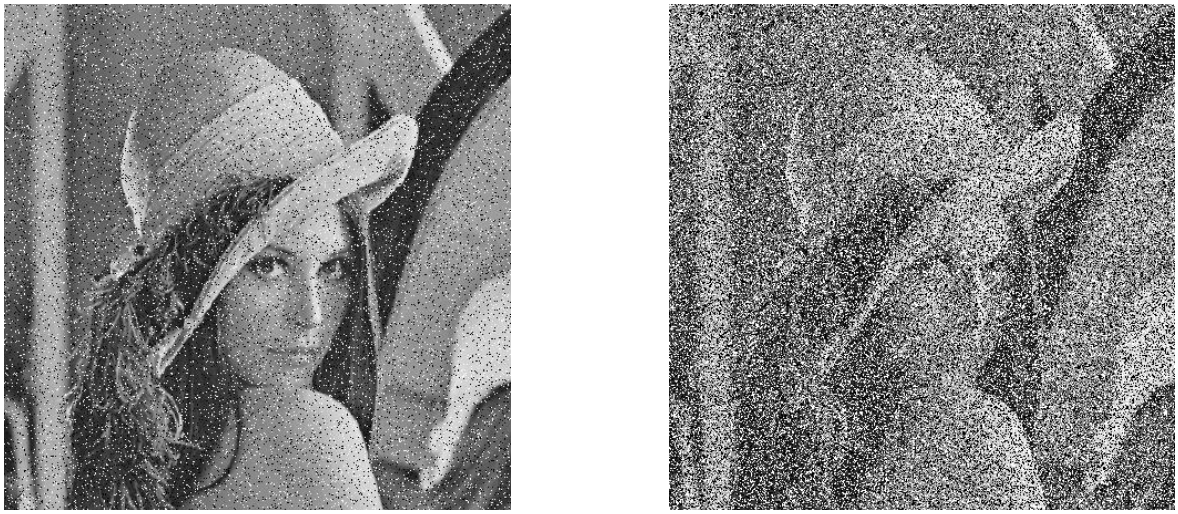


Figure 1-2 : ajouter le bruit poivre et sel. Gauche : densité =0.1, droite : densité =0.5

II. Appliquer des filtres:

En utilisant la fonction ***fspecial***, appliquer des filtres gaussiens aux images bruité (variante = 0,1 ; densité = 0,1) :
Le résultat varie en fonction du paramètre sigma de la fonction ***fspecial***.

Figure 1-3 : bruit gaussien(v=0.1) filtré par le filtre gaussien avec sigma =0,5 et 1

Figure 1-4 : bruit poivre et sel (d=0.1) filtré par le filtre gaussien avec sigma =0,5 et 1

En suit, on applique le filtre médian sur les images bruités :

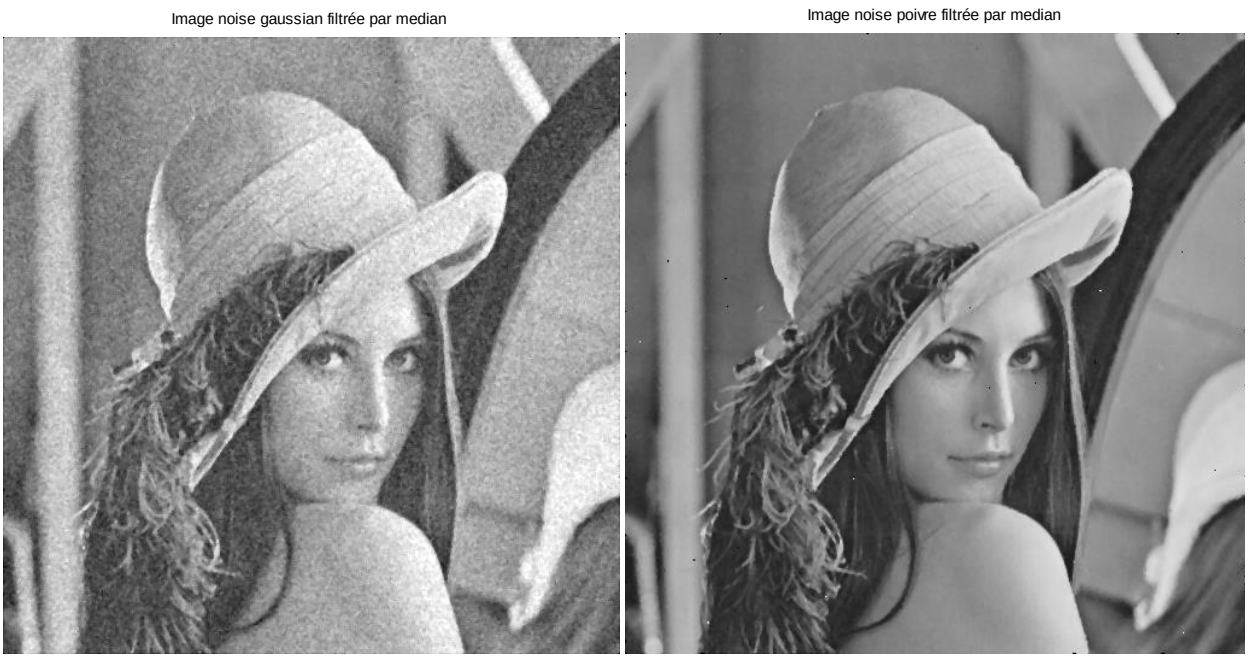


Figure 1-5: l'mage bruitée (gaussien et poivre et sel) filtré par le filtre médian

Le filtre médian filtre bien le bruit poivre et sel. Pour le bruit gaussien, il a presque le même résultat que filtre gaussien

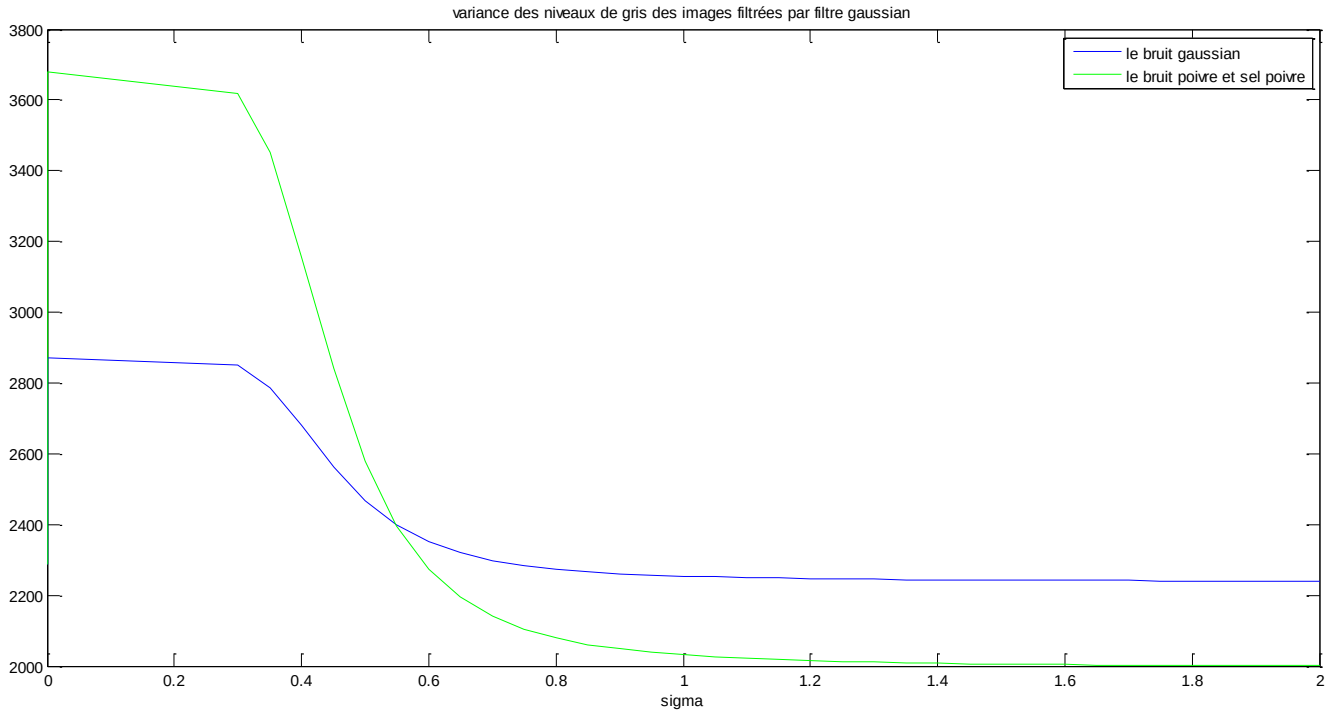


Figure 1-6 : variance des niveaux de gris de l'image bruitée filtrée par le filtre gaussien de variance croissante

Figure 1-7 : image sans flash filtrée par le filtre gaussien et médian

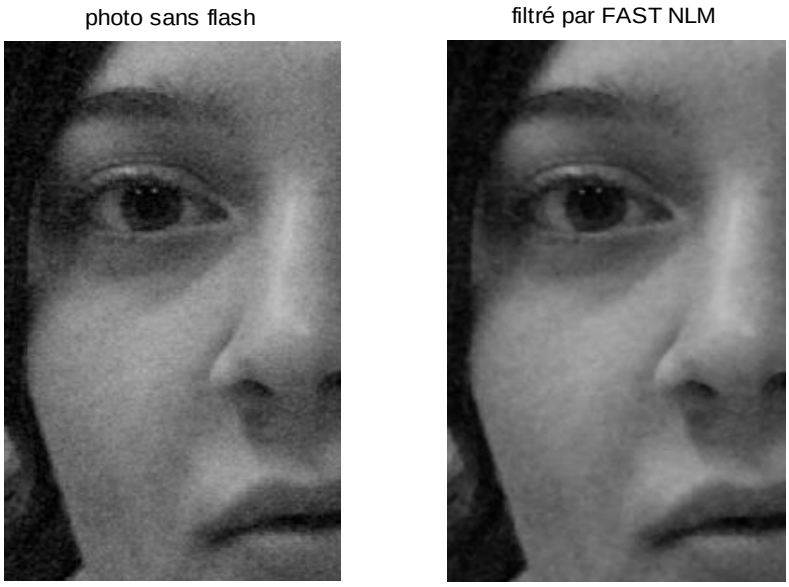


Figure 1-8 : image filtrée par FAST-NL means

Nous observons qu'il n'y a pas grande différence entre les filtre gaussien et médian sur l'image sans flash mais la différence est visible après le filtre fast-nl manse

Tp2. Restauration d'une ville photo

Travail préliminaire :

Calculer l'histogramme des différents types d'images

Figure 2-1： image Claire et son histogramme

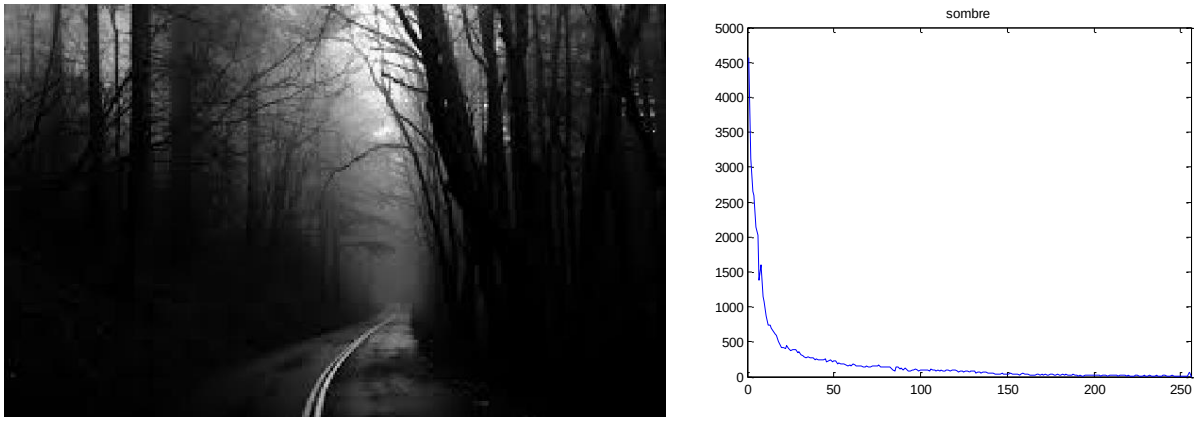


Figure 2-2： image sombre et son histogramme

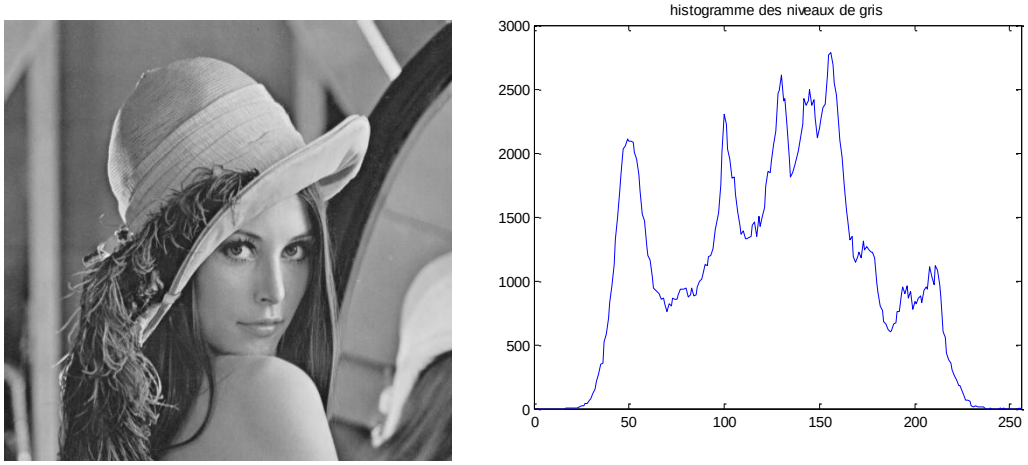


Figure 2-3： histogramme de l'image lenna

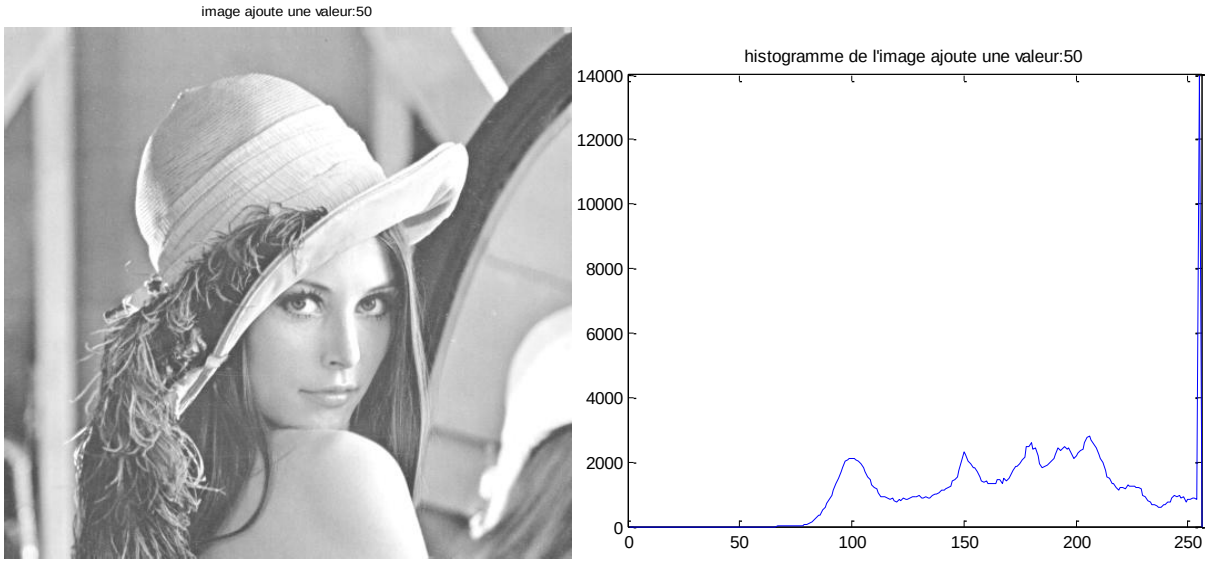


Figure 2-4 : image + 50 et son histogramme

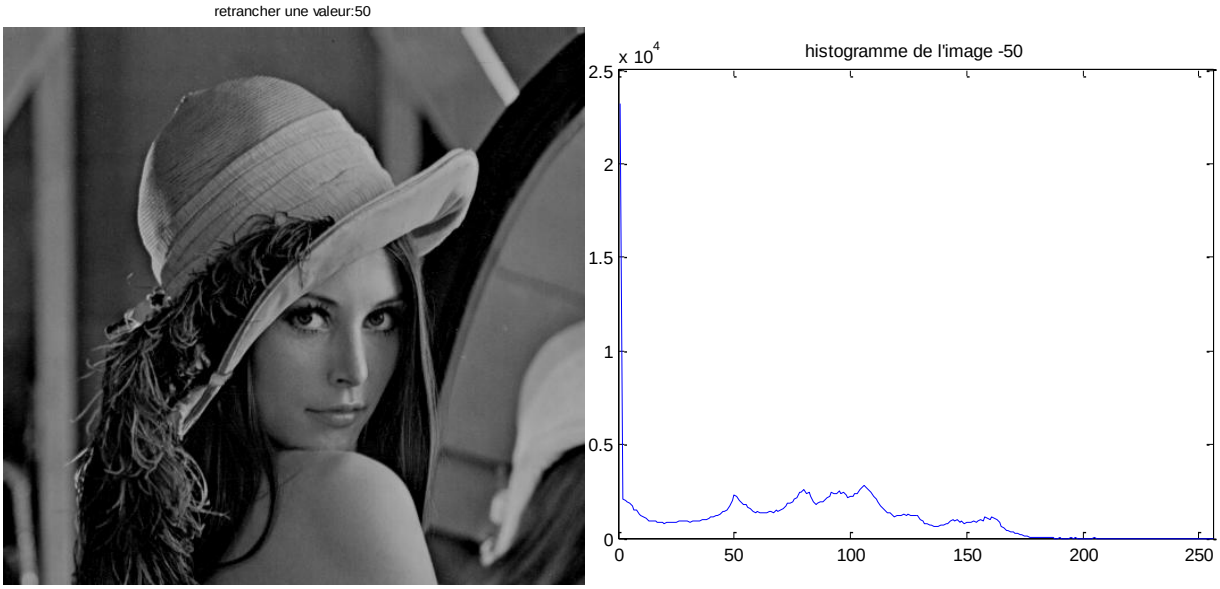
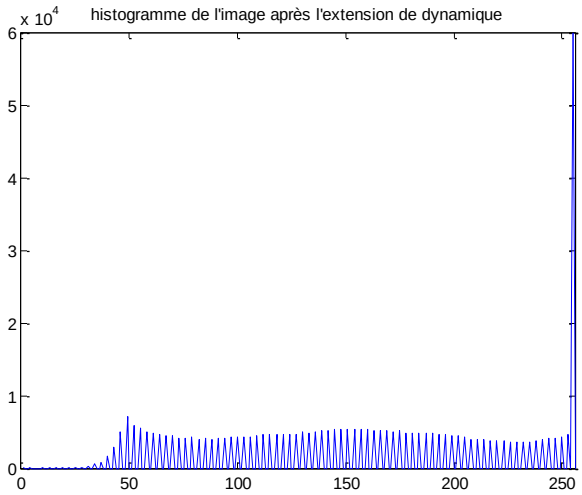


Figure 2-5： image -50 et son histogramme

Ajouter une valeur fixe à tous les pixels permet d'augmenter la luminance.
Retrancher une valeur fixe à tous les pixels permet de diminuer la luminance.
La conséquence sur l'histogramme est la translation de l'histogramme

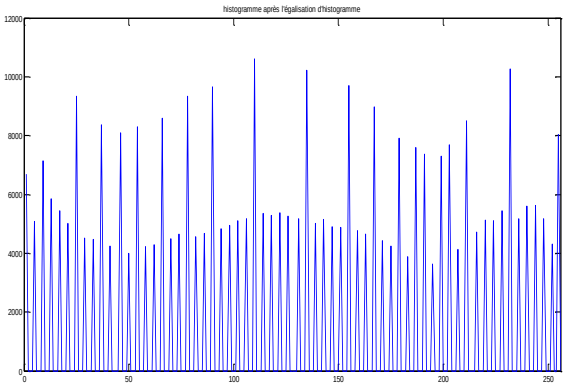
Extension dynamique : en appliquant la formule suivant

```
Igray_modi(i,j) = 255/(Vmax-Vmin)*(Igray(i,j)-Vmin);
```



Par rapport à la figure 2-1, on peut constater que l'extension e dynamique affine du niveau de gris des pixels et que l'image utilise toute la dynamique de représentation.

Égalisation d'histogramme par la fonction ***histeq*** :



Cette transformation équilibre le mieux possible la distribution des pixels dans la dynamique. L'image obtient un mieux contraste après la transformation.

Travail à faire : Améliorer la photo par l'égalisation d'histogramme

On propose d'utiliser la fonction ***histeq*** pour améliorer la photo ancienne.

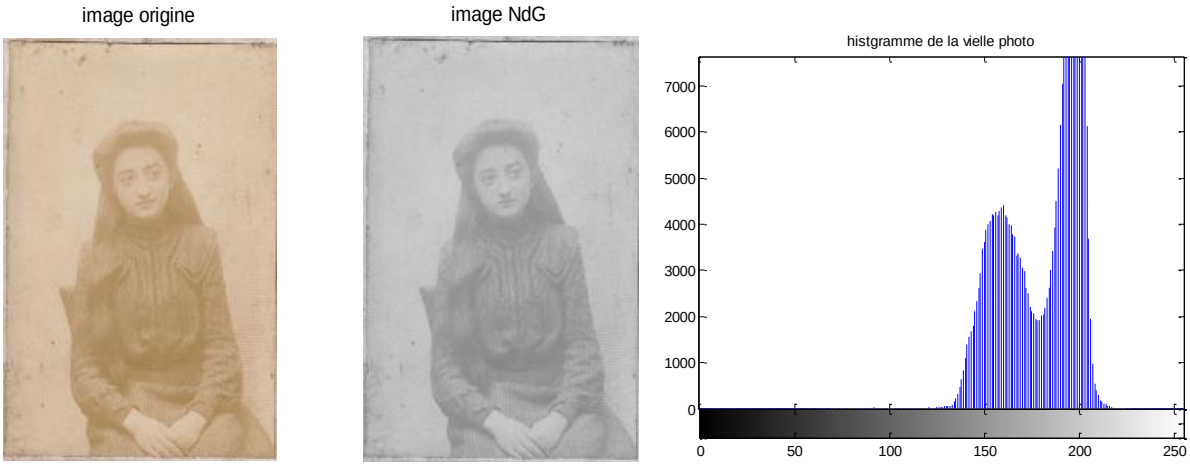


Figure 2-6 :image ancienne et son histogramme

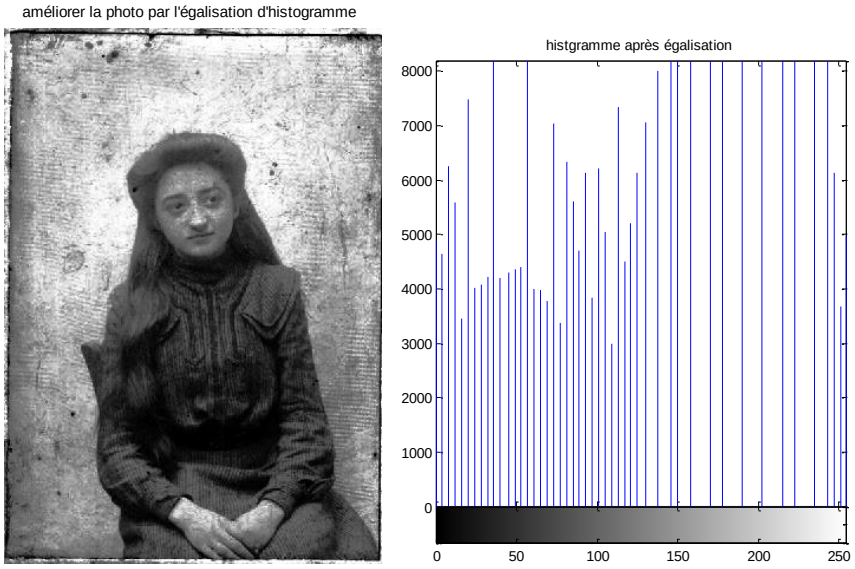
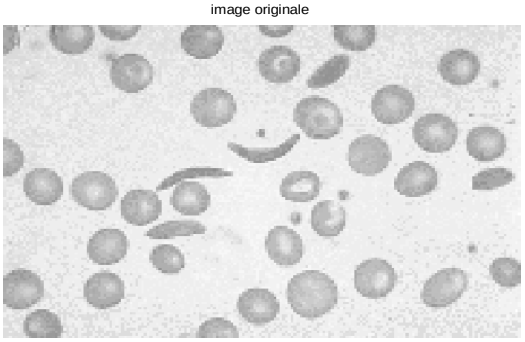


Figure 2-7 : image améliorée et son histogramme

Ce TP nous permet d'observer les différents types de la correction d'histogramme. L'égalisation d'histogramme est une méthode d'ajustement du contraste d'une image qui utilise l'histogramme. Cette méthode est rapide, facile d'implémentation.

Tp3. Comptage de cellules dans une image de microscope



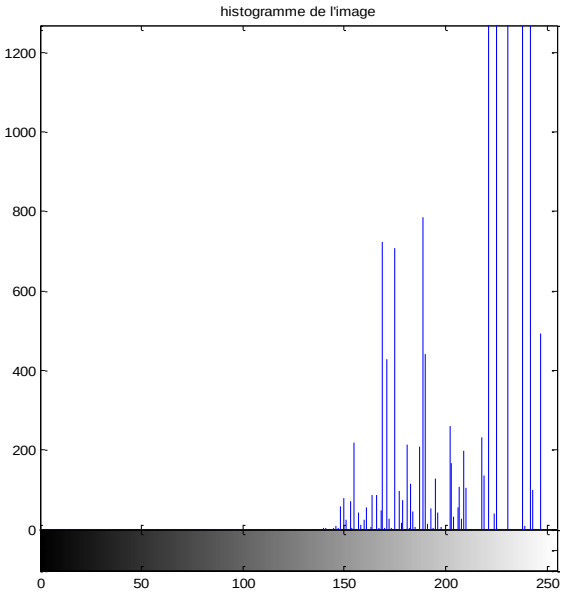


Figure 3-1 :image originale et son histogramme

Algorithme d'Otsu :

Le but de cet algorithme est la bipolarisation d'images à niveaux de gris. Dans la méthode d'Otsu, le seuil qui minimise la variance intra-classe est recherché à partir de tous les seuillages possibles :

$$\sigma_w^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t)$$

Otsu montre que minimiser la variance intra-classe revient à maximiser la variance inter-classe2:

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = \omega_1(t)\omega_2(t)\left[\mu_1(t) - \mu_2(t)\right]^2$$

Ici, on maximise la variante interclasse et obtient le seuil idéal.

Algorithme:

1. Calculer l'histogramme et les probabilités de chaque niveau d'intensité
2. Définir les $\omega_i(0)$ et $\mu_i(0)$ initiaux
3. Parcourir tous les seuils possibles $t = 1 \ldots$ intensité max
    1. Mettre à jour ω_i et μ_i
    2. Calculer $\sigma_b^2(t)$
4. Le seuil désiré correspond au $\sigma_b^2(t)$ maximum.

image binarise par la methode Otsu

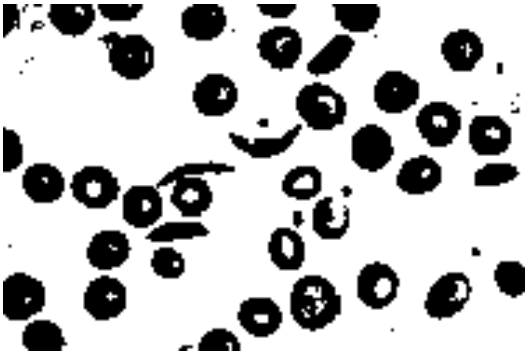


image binarise par graythresh

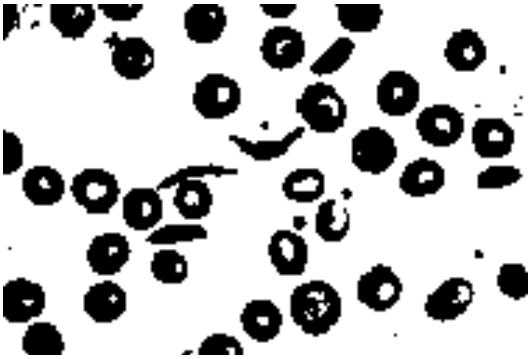


Figure 3-2 : image binarisée par la méthode Otsu (gauche) et la fonction Matlab (droit)

La méthode Otsu segmente l'image en 2 classes et on retrouve la même segmentation par la fonction Matlab ***graythresh.***

Compter et caractériser les formes :

On nettoie l'image et supprime les petits morceaux qui sont inférieurs aux P pixels avec la fonction ***bwareaopen*** . En suit on étiquette les composants et compte le nombre total. Le résultat obtenu est ci-dessous :

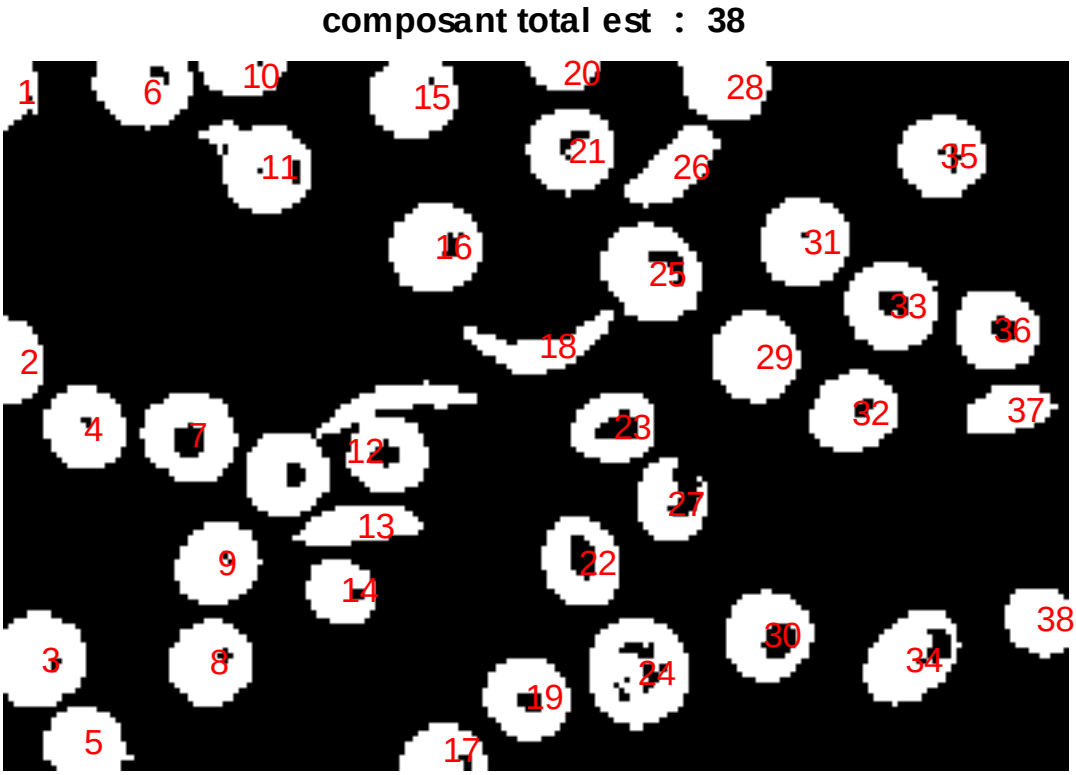


Figure 3-3 : étiquette les composants

D'après la figure 3-3, on observe qu'il a des composantes collent ensemble et sont considéré comme un seul élément. Par exemple numéro 12 se compose de trois cellules. Nous obtenons 38 composants au lieu de 40.

L'idée est de faire le prétraitement avant de labéliser et étiquetage. J'ai essayé la dilatation, l'érosion, l'ouverture et la fermeture sur l'image binaire originale mais les résultats ne sont pas suffisants. Il faut trouver une combinaison de multiple opération pour séparer les composants.

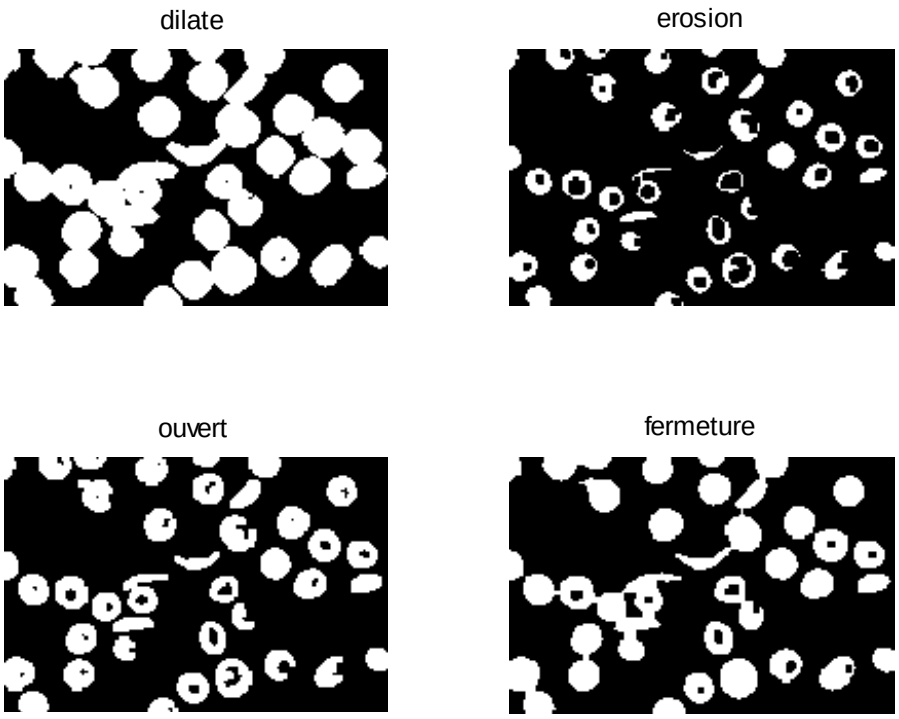


Figure 3-4 : les résultats des différentes opérations

Tp4. Détection de contours et de point

Appliquer un filtre gradient et un filtre laplacien sur Lena

Deux approches :

- Approche gradient (dérivation première ordre) : détermination des extréma locaux dans la direction du gradient.
- Approche laplacien (dérivation second ordre) : détermination des passages par zéro du laplacien.

Pour l'approche gradient, les dérivées sont calculées par convolution de l'image avec un masque de différences. Il existe plusieurs opérateur gradient : sobel, prewitt, robert et MavoVssy. On calcule le gradient en horizontal et en vertical. En utilisant les différents opérateurs on obtient les résultats ci-dessous :

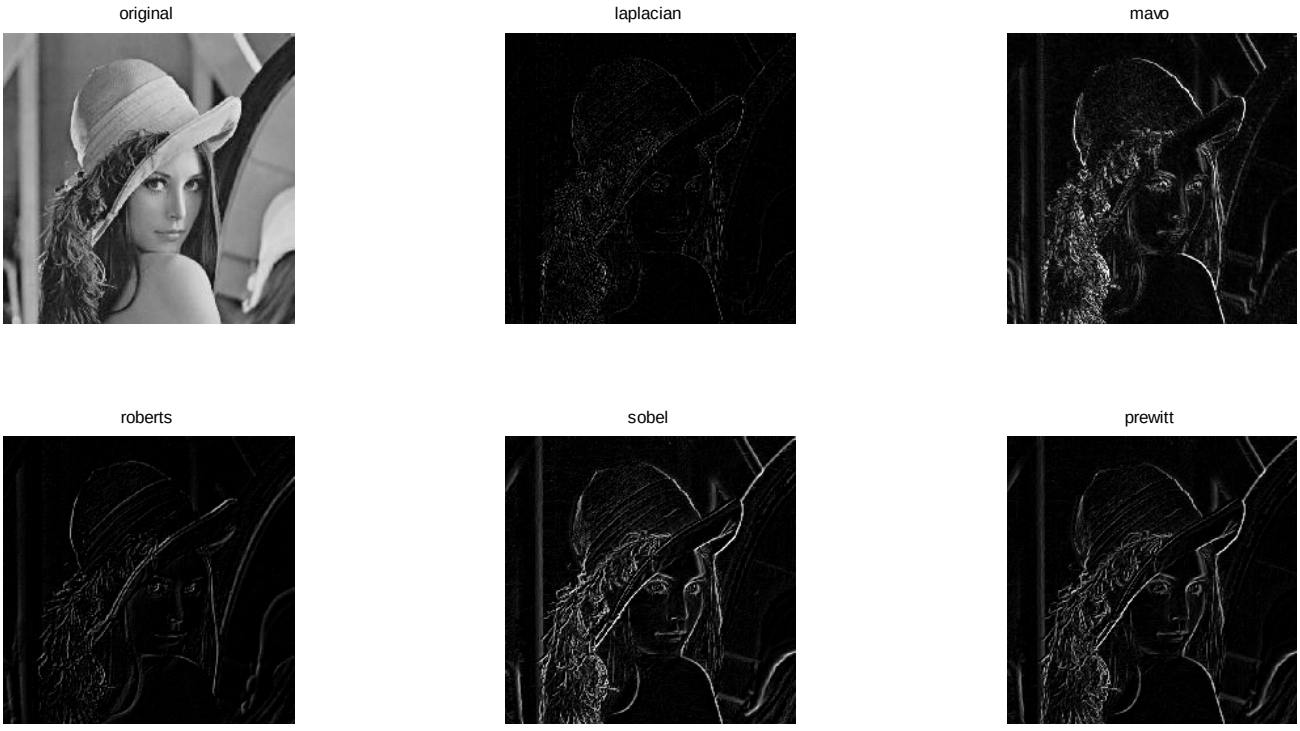


Figure 4-1 : les résultats des différents opérateurs

Figure 4-2 : Zoom in les images

En comparant les résultats obtenus, nous pouvons constater que l'opérateur de Robert n'est pas très précis. En revanche, les résultats ci-dessus montrent que Sobel performe un peu mieux que Prewitt a ce niveau sur l'image étudiée. Cela peut s'expliquer par la différence de filtrage passe bas effectue par ces deux opérateurs. Sobel utilise un filtre passe bas de masque [1 2 1] qui amplifie plus les basses fréquences que le filtre moyenneur [1 1 1] utilise par Prewitt.
Par rapport aux filtre gradient, le filtre Laplacien est plus sensible aux bruits mais en même temps il est plus précis.

Le détecteur de Moravec :

L'idée du détecteur de Moravec est de considérer le voisinage d'un pixel (une fenêtre) et de déterminer les changements moyens de l'intensité dans le voisinage considéré lorsque la fenêtre se déplace dans diverses directions. La fonction est :

$$E(x,y) = \sum_{u,v} w(u,v) \ |I(x+u, y+u) - I(u,v)|^2,$$

Où : E (x, y) représente la moyenne du changement d'intensité lorsque la fenêtre déplacée de (x, y)

En prenant les poids w = 1, le résusltat obtenu est :

Figure 4-3 : le résultat obtenu avec le détecteur de Moravec avec u=5, v=7

D'après le résultat, nous pouvons observer que les pixels qui situent au coin ont une valeur grande. Cette méthode permet de detecter les coins d'une image. Mais il ne peut pas détecter le bord de l'image.
Pour améliorer la performance, nous pouvons appliquer les poids sur les pixels, plus proche de la centre, plus grand le poid vaut. Donc par exemple une fenêtre gaussien.

TP5 code barre

Dans cette partie, nous voulons compléter les codes-barres pour qu'ils soient le plus lisibles. L'idée est de segmenter d'abord le numéro et les barres, en suit appliquer les opérations de la méthode morphologie pour compléter les barres.

D'abord nous utilisons le code de barre sans rotation ci-dessous :

Figure 5-1 : le code de barre sans rotation

Pour segmenter l'image, nous cherchons la ligne de séparation. Entre les barres et la ligne de numéro, il y a une ligne presque vide qu'on peut l'utiliser pour séparer l'image. Pour trouver cette ligne, nous calculons la projection dans la direction horizontale et chercher l'indice de la ligne séparée. (Voir fig. 5-2) L'image séparée est obtenue (figure 5-3)

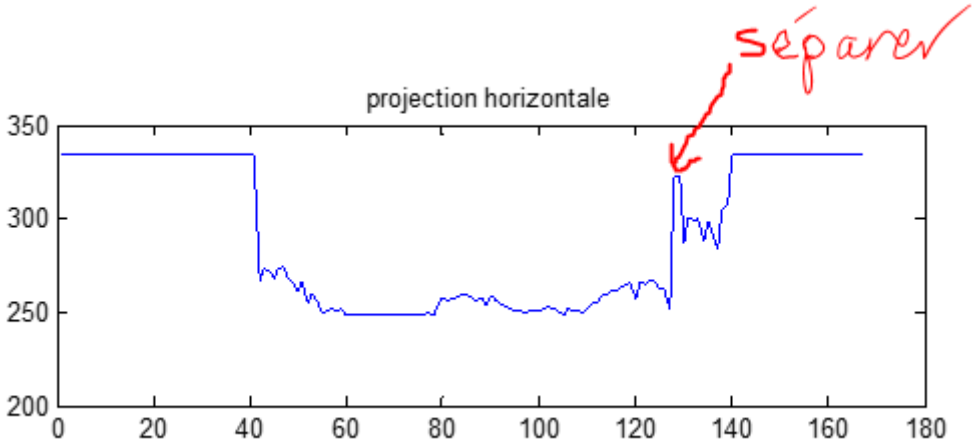


Figure 5-2: la projection horizontale

Figure 5-3: l'image séparant en deux parties

Nous appliquons la méthode de dilatation en utilisant le structure ligne et dans le sens vertical pour compléter les lignes. Comme le résultat montré dans à gauche, les lignes ne sont pas complétées, donc il faut compléter les barres de la première ligne à la dernière ligne.

Figure 5-4: le résultat de la dilatation (gauche) et complété(droite)

Et en combinant les deux parties, nous avons obtenu le résultat :

Figure 5-5 : le résultat finale

Dans le cas où l'image est male posée, il est recommandé d'effectuer un prétraitement pour aligner et redresser l'image.
L'idée est de d'abord segmenter le corps du code barre et le fond de l'image, en suit appliquer une opération de rotation.
Ici on utilise la fonction Matlab ***imrotate*** avec l'angle de rotation égale à 27. Pour éliminer les bruits, on applique le filtre médian sur l'image binaire.

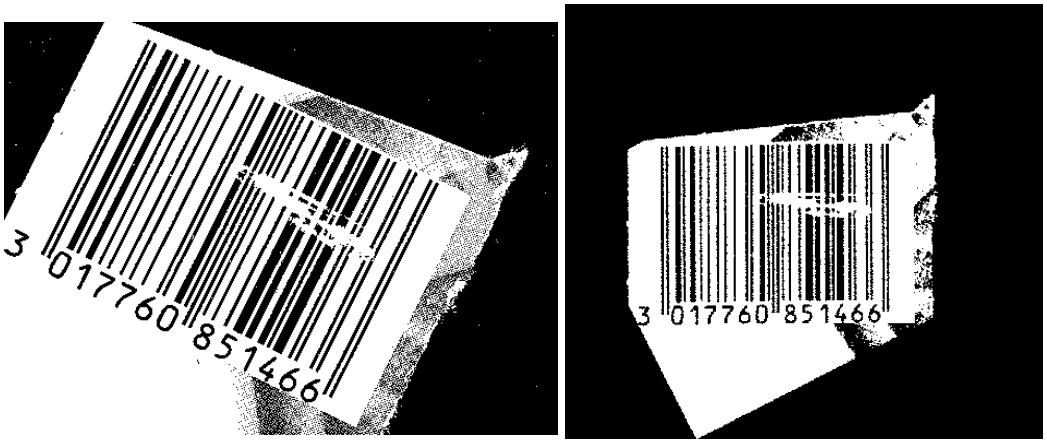
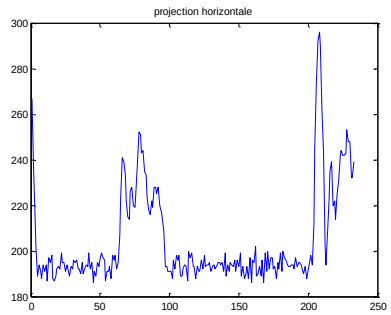


Figure 5-6: image originale binarisée et transformée

Pour simplifier, on choisit la région de code barre ROI (region of interest) à main et répéter les processus précédents pour séparer les barres et le numéro.

Choisir la ROI à main



Projection de la ROI horizontale

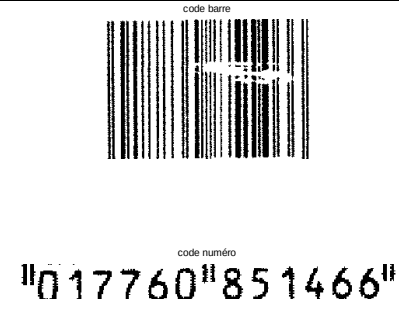


Figure 5-7: ROI code barre et la séparation

Dans ce cas, nous devons effectuer plusieurs fois les opérations dilatation et érosion jusqu'à les lignes ne changent plus.

Figure 5-8 : opération de la dilatation, l'erosion et le résultat final

Le résultat obtenu est lisible mais la première colonne est plus gros (une colonne de pixel) que l'image originale.
Pour améliorer cette application, nous pouvons appliquer les méthodes de détection de contour et lignes pour extraire les lignes et redresser l'image automatiquement.

TP 6 Matrice de cooccurrence

Dans ce TP, nous souhaitons caractériser les images de texture par leurs matrices de cooccurrence.

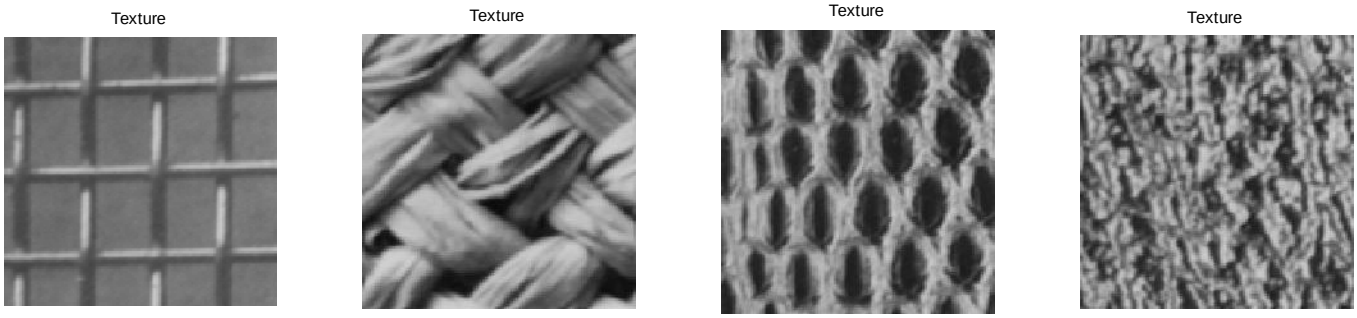


Figure 6-1: image de texture

On utilise la fonction ***graycomatrix*** pour calculer la matrice de cooccurrence et calculer les paramètres des images en utilisant la matrice de cooccurrence. On calcule la matrice de cooccurrence dans quatre directions (`offset = [0 2 ; -2 2 ; -2 0 ; -2 -2]`) et les paramètres obtenus des textures traités dessus sont montrés dans la figure 6-2.

Contrast	Correlation	Energy	Homogeneity
0.3308	0.7055	0.3656	0.8932
0.6701	0.8743	0.0711	0.7683
1.5940	0.6796	0.0739	0.6706
2.1818	0.4015	0.0458	0.5817

(a) 0°

Contrast	Correlation	Energy	Homogeneity
0.6686	0.4076	0.3133	0.8198
0.7574	0.8577	0.0724	0.7731
1.6874	0.6602	0.0694	0.6523
2.5894	0.2893	0.0436	0.5549

(b) 45°

Contrast	Correlation	Energy	Homogeneity
0.4434	0.6055	0.3624	0.8795
0.9963	0.8126	0.0598	0.7194
0.6326	0.8728	0.1020	0.7781
1.1199	0.6914	0.0600	0.6777

(c) 90°

Contrast	Correlation	Energy	Homogeneity
0.7204	0.3617	0.3070	0.8131
1.7264	0.6757	0.0489	0.6662
1.7495	0.6478	0.0714	0.6570
2.2889	0.3716	0.0452	0.5736

(d) 135°

Figure 6-2: les paramètres dans 4 direction

D'après les résultats, nous pouvons retrouver les relations entre les paramètres et les textures.
La texture_1 n'est pas bien contrasté donc son contraste est le plus petit.
La corrélation démontre la régularité de la texture, plus la valeur est grande, plus elle est corrélée. Donc l'image 4 est le moins corrélée parmi les quatre textures.
L'énergie exprime la distribution homogène de l'image gris et la finesse de la texture. Ici, à part la texture_1, la texture_2 est plus gros donc son énergie est plus grande.

Conclusion

A cause de manque de temps, je n'ai pas réussi de finir tous les TPs.
Ces TPs me permet de mettre en pratique les connaissances de ce que nous avons acquis pendant les cours et mettre en œuvre les techniques. Ce sont les connaissances basic et très utile pour le traitement d'image ainsi que les autres applications comme analyse des documents.