TP5: Segmentation d'images

Exercice 1: Seuillage

Les défauts d'éclairage provoquent toujours un mal seuillage. Pour cela, nous vous proposons dans cet exercice d'utiliser l'opération de la division pour corriger l'image avant le seuillage.

1. Chargez les deux images «defaut.png» et «fond.png» du dossier TP5
2. Ensuite, calculez la division des deux images
3. Appliquez le seuillage global sur l'image résultante
4. Interprétez le résultat

N.B:
Les fonctions de Matlab à utiliser: **graythresh() et im2bw()**

Exercice 2: Comptage d'objets

Parmi les applications fondamentales de la segmentation d'images est le comptage d'objets.

1. Chargez l'image «rice.png» du toolbox Matlab
2. Appliquez un seuillage global par la méthode «Otsu»
3. Convertissez l'image «rice.png» en une image binaire en utilisant le seuil global généré par la méthode «Otsu»
4. Puis calculez la matrice des labels et le nombre d'objets comptés
5. Affichez les objets segmentés en RGB

N.B:
Les fonctions de Matlab à utiliser**: graythresh(),im2bw(I,threshold), bwlabel() et label2rgb()**

Exercice 3: Otsu/K-means

Il s'agit donc de segmenter les images couleurs suivantes par les méthodes de seuillage et de K-means et de comparer les résultats obtenus

1. Chargez l'image «hand.bmp»
2. Convertissez cette image en niveaux de gris
3. Affichez son histogramme. Peut-on identifier des modes ?
4. Utilisez la fonction multithresh (par «*Otsu »*) qui retourne les valeurs des seuils en fonction du nombre de classes souhaité.
5. Appliquez le seuillage multiple sur l'image en niveaux de gris. On utilise la fonction **imquantize()**. Pour afficher le résultat, on utilise une palette de couleurs artificielles,par exemple colormap('jet').
6. Utilisez la méthode kmeans pour segmenter l'image en niveaux de gris et comparer les résultats des deux méthodes. La fonction **kmeans()** de Matlab prend en paramètre d'entrée un vecteur (et non une matrice). Utiliser la fonction reshape pour la transformation.
7. On vous propose à présent d'utiliser l'information couleur pour la méthode de kmeans. La valeur d'un pixel est maintenant une variable de dimension 3. Adaptez le code précédent du niveau de gris pour la couleur. Commentez le résultat.

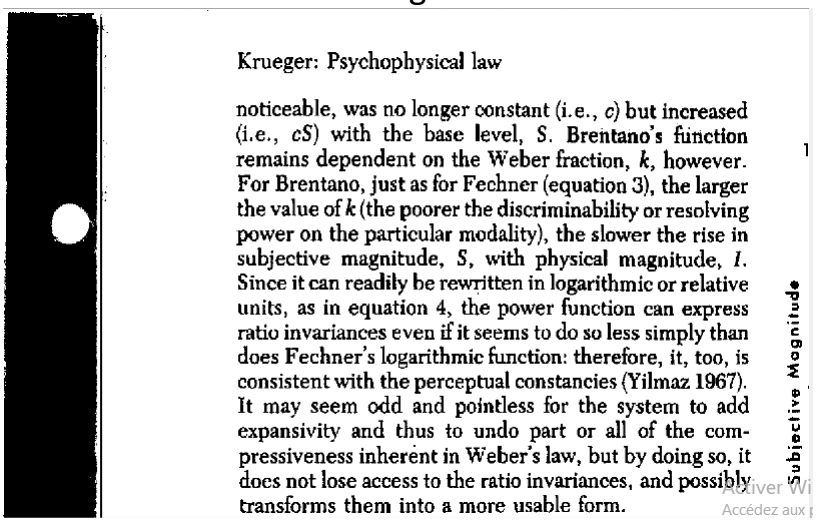
Exercice 4: Corrélation

Le but de cet exercice est de rechercher les positions de toutes les occurrences d'un motif dans une image I. Le motif est donné dans une autre image M.
On suppose que le motif se trouve dans l'image M à la même échelle, même orientation mais en une position inconnue. Le principe est basé sur la corrélation de l'image I avec le motif. Cette corrélation donne une valeur maximale partout où les motifs présentent une correspondance forte avec les pixels de l'image. La fonction normxcorr2 permet de calculer la correlation entre l'image I et le template M.

Il s'agit alors de déterminer toutes les occurrences de la lettre 'a' dans le document représenté par l'image I. Les étapes à suivre sont :

- Lecture des deux images (L'image I et le motif M)
- Utilisation de la fonction ***normxcorr2()*** pour calculer la correlation entre les deux images.
- Affichage du résultat par la fonction surf avec l'option shading flat
- Binarisation de l'image pour obtenir les occurences de la lettre 'a' en choisissant le bon seuil.
- Calcul du nombre d'occurrences de la lettre 'a'
- Affichage d'une étoile rouge sur chaque lettre trouvée

Image I



Motif M