MODULE : Analyse d'information multimédia

TP 3 : Traitement d'images sous Matlab

1 Lire et écrire des images sous Matlab :

Matlab est capable de lire et de décoder les fichiers images JPEG, TIFF, BMP, PNG, HDF, PCX ou XWD. Une image sous Matlab peut être représentée sous plusieurs formes, mais toujours **sous forme d'une matrice**. Avant de traiter une image dans Matlab, il faut la lire et décoder son format afin de la transformer en une matrice de valeurs. L'exemple ci-dessous permet de lire une image au format TIFF, de la décoder dans la variable img et de l'afficher à l'écran dans une figure.

```
>> img=imread('saturn.tif');
>>figure;imshow(img);
>> axis on     % affiche les graduations des axes
>> colorbar   % affiche la barre des couleurs de l'image
```

L'accès à un élément particulier d'une image est indexé par le nom et la position de cet élément. Par exemple, si on conserve l'image img ci-dessus, on peut récupérer les valeurs ou les modifier aisément.

```
>>img(3,2)
>>img(1:10,30:40)
>>img(1:10,30:40) = 0;
>>figure;imshow(img);
```

La commande subplot permet d'afficher plusieurs images dans la même figure. Elle s'utilise avec comme arguments le nombre de ligne, le nombre de colonnes et le numéro de l'image dans la figure. Dans l'exemple ci-dessous on souhaite afficher deux images sur la même ligne dans une seule figure.

```
>> img=imread('lena.jpg');
>> img2=imread('saturn.tif');
>> figure;subplot(1,2,1);imshow(img);
>> subplot(1,2,2);imshow(img2);
```

Close permet de fermer la dernière figure ouverte .

Close avec en paramètre le numéro de figure permet de fermer la figure donnée en paramètres.

Close all pour fermer toutes les figures.

Matlab autorise l'exportation d'images sous divers formats: BMP, TIFF, EPS, PS... La commande qui permet da sauvegarder une figure est ***print –dFORMAT fichier***. Un exemple est donné ci-dessous. Dans cet exemple, on affiche une image dans une figure et grâce à la commande ***print***, on exporte le résultat dans le format JPEG avec pour nom de fichier result.jpg.

```
>> img=imread('saturn.tif');
>> figure;imshow(img);
>> print -djpeg saturn.jpg
```

Les valeurs des images lues sous Matlab sont entières, mais dans certaines circonstances, on a besoin de travailler sur des valeurs réelles. La transformation pour passer d'entier à réel utilise la fonction ***im2double***.

```
>> img=imread('saturn.tif');
>> figure;imshow(img);
>> imgdbl=im2double(img);
>> figure;imshow(imgdbl);
>> imwrite(imgdbl,'test.jpg','jpeg'); % l'enregistrer sous format jpeg
>> imgint=im2uint8(imgdbl);
>> figure;imshow(imgint);
```

2 Histogramme - seuillage

2.1 Histogramme

L'histogramme d'une image permet d'obtenir le nombre de pixel de la même couleur dans une image, il donne la répartition de ses niveaux de gris. Ainsi pour une image qui possède 256 niveaux de gris, l'histogramme représente le niveau de gris en fonction du nombre de pixels à ce niveau de gris dans l'image.

Si on fait varier la luminosité on voit que les pic se déplacent soit vers la gauche quand la luminosité diminue et vers la droite quand la luminosité augmente.

Création de l'histogramme manuellement

```
 img = imread('pout.tif');
histo=zeros(1,256);
[l,c]=size(img)
 for i=1:l
 for j=1:c
 histo(img(i,j)+1)=histo(img(i,j)+1)+1;
 end
 end
figure;plot(histo);
```

Histogramme via matlab

```
img = imread('pout.tif');
histo = imhist(img);
figure;plot(histo);
```

2.2 Seuillage

On sait que les niveaux de gris à zéro correspondent au noir et que les niveaux de gris à 1 indiquent le blanc. L'histogramme donne donc une excellente idée de la séparation entre quelque chose qui est clair et quelque chose qui est foncé dans l'image. Typiquement, une utilisation de ce fait est le seuillage d'une image, ce terme désigne la définition d'un seuil au-dessus ou en-dessous duquel on va garder certaines valeurs de niveaux de gris.

Seuillage

Exemple1 :

```
img=imread('lena.jpg');
img=rgb2gray(img) ;
for i=1:512
        for j=1:512
                if(img(i,j)>=100) img(i,j)=255;
                else img (i,j)=0;
                end
        end
end
>> figure ;imshow(img) ;
```

Exercices :

Histogramme :

L'histogramme permet d'obtenir le nombre de pixel de la même couleur dans une image.

1. Charger l'image mandrill.bmp
2. Calculer la taille théorique de cette image. Vérifier que la taille obtenue est la même fournie par Windows.
3. Transformer la en image monochrome
4. Visualiser son histogramme avec la fonction imhist

Binarisation :

Le but d'une binarisation est de passer une image en noir et blanc, ce qui consiste à avoir que des pixels noir et blanc.

On définit un seuil qui permettra de savoir si un pixel deviendra noir ou blanc. Si le pixel est inférieur au seuil alors le pixel deviendra noir si non le pixel devient blanc.

Pour déterminer le seuil on peut utiliser l'histogramme et on donnera la valeur du seuil manuellement ou on peut utiliser un seuillage automatique via Matlab (graythresh permet de determiner le seuil automatiquement).

1. Ecrire un code qui permet de binariser l'image
2. Refaire le travail en utilisant la commande find
3. Binariser l'image en utilisant la commande im2bw
4. Commenter les résultats obtenus

Augmentation du contraste par étirement d'histogramme

Cette première transformation sur l'histogramme a pour objet l'augmentation du contraste d'une image. Pour cela, il convient d'augmenter sur l'histogramme l'intervalle [a,b] de répartition des niveaux de gris de l'image d'entrée « Ie >>. On parle alors d'étirement d'histogramme.

Du point de vue de la transformation, un étirement maximal est réalisé dès lors que la répartition des niveaux de gris de l'image de sortie « Is >> occupe l'intervalle maximal possible [0, Max]. Typiquement pour une image dont les niveaux sont codés sur 8 bits, l'intervalle [a,b] de Ie sera étiré jusqu'à l'intervalle [0, 255] pour Is.

1. Ecrire la fonction : function I = RecDyn(Im,a,b)
2. Recadrer votre image avec a=0 et b=100.
3. Afficher le résultat.
4. Voit on l'intérêt de faire un recadrage ? Justifier !
5. Essayer la commande imadjust. Y'a t'il une différence avec l'image précédente.

Égalisation d'histogramme :

La deuxième transformation que nous abordons maintenant a pour objet également l'augmentation du contraste d'une image. Il comprend l'étirement d'histogramme présenté précédemment avec en plus une répartition uniforme des niveaux de gris. Après transformation, l'histogramme devient constant : chaque niveau de gris est représenté dans l'image par un nombre constant de pixels. On parle aussi d'histogramme « plat ». Cette transformation n'est en théorie possible que dans la mesure où l'on dispose de données continues. Or le domaine spatial et, surtout, l'échelle des niveaux de gris sont des données discrètes. Dans la pratique donc, l'histogramme obtenu ne sera qu'approximativement constant.

1. Utiliser la commande histeq pour faire l'égalisation de l'histogramme
2. Afficher les deux histogrammes et les deux images correspondantes
3. Commenter