Institut Nationale des télécommunications et des technologies de l'information et de la communication

Module : Traitement D'Image

TP N° 01

Initiation au Traitement d'image sous MATLAB

Janvier 2014

TP01 : Initiation au traitement d'image sous MATLAB

Les comptes rendus doivent être donnés au format **.doc** ou **.pdf**. Ils doivent contenir les réponses aux questions posées, et les scripts ou fonction sous code Matlab ainsi que les images résultats. Le tout sera envoyé dans un fichier nommé « Nom1_Nom2_TP_01.zip » à l'adresse e-mail de votre enseignant.

L'objectif de ce TP est de comprendre le fonctionnement de Matlab et de ces outils de base qui serviront dans les TP suivants. Nous allons donc voir comment charger, afficher et sauvegarder une image et faire quelques opérations de base sur celles ci.

Lecture et affichage d'une image

1. Avant de commencer la programmation, il faut lire l'annexe.
2. Utiliser les commandes clear all, close all pour initialiser l'environnement Matlab.
3. Créer un répertoire qui porte le nom suivant « Nom1-P1_Nom2-P2_TP_01 », et copier les fichiers et/dossiers fournis dans ce dernier. (P1 : 1er lettre de votre prénom)

Note : tout résultat doit être sauvegardé dans votre répertoire personnel.

4. Créer une variable avec le chemin jusqu'à l'image "greens.jpg". Exp : "C:\Program Files\MATLAB\R2010a\toolbox\images\imdemos"
5. Créer une variable contenant le nom de l'image (avec extension).
6. Utiliser la fonction "help imread" pour obtenir les propriétés de la fonction.
7. En utilisant variable "ima", Lire l'image "greens.jpg" en concaténant le chemin et le nom (**cat**(2, n,p)).
8. Observer le format de l'image lue dans le gestionnaire de variables **whos**.
9. Afficher l'image avec **imshow, imtool, image** puis dans une autre figure avec **imagesc**. Observer les différences.
10. Créer une deuxième variable "imad" contenant la même image en format double utilisez la commande **double**().
11. Diviser l'image "imad" par la valeur 255. Refaire l'affichage.
12. Créer une image im2d content le résultat de la commande **im2double()**.
13. Comparer les deux images (les valeurs des pixels)
14. Convertissez l'image « ima » précédente en une image en niveau de gris. En utilisant la méthode suivante : ima_gray = ima(:,:,1)*coef1 + ima(:,:,2)*coef2 + ima(:,:,3)*coef3. (coef1 + coef2 + coef3 = Comparez les différents résultats obtenus pour différentes valeurs des coefficients (un coef très grand par rapport aux autres « i.e. 0.8 0.1 0.1). Aide : -le calcul sur les images se fait en double -help **rgb2gray**
15. Sauvegarder l'image résultant de la comparaison, en utilisant la commande « **imwrite** ».

Type d'Images dans Matlab

Une image en niveau de gris ou Grayscale image est une matrice d'intensité lumineuse dont toutes les valeurs indiquent une intensité comprise entre **intmin**(class(I)) et **intmax**(class(I)) où la classe désigne uint8, uint16 ou int16 en simple ou double.

1. Charger l'image ' cameraman.tif' de Matlab.
2. Décrire ses caractéristiques. utilisez la fonction **imageinfo.**

Une image en couleur vraie ou Truecolor Image ou RGB-image est une image dans laquelle chaque pixel est indiqué par 3 valeurs ou couleurs ; une pour le Rouge, une pour le Vert et la troisième pour Bleu. Matlab enregistre une image truecolor par 3 tableaux superposés ou un bloc m*n*3 ; les plans correspondent à R, G ou B. Les valeurs des plans sont en uint8, uint16 ou double. Pour d´déterminer la couleur d'un pixel (a, b), il faudra donner (a, b, 1 : 3).

1. Ecrire la commande Matlab **RGB=reshape(ones(64,1)*reshape(jet(64),1,192),[64,64,3]);** qui crée une image simple truecolor contenant les trois plans.
2. Affichez les trois plans séparément. Et commenter les commandes utilisées ainsi que les résultats obtenues, **help jet, help reshape.**

Une image indexée se présente sous la forme d'un tableau de valeurs noté X, accompagné d'une matrice de colormap nommée map représentée par m-lignes et 3-colonnes (RGB) de valeurs comprises dans [0, 1]. La valeur (en uint8, uint16 simple ou double) de chaque pixel de X désigne la position de la ligne dans le colormap. Le map se charge automatiquement avec l'image avec la fonction imread.

1. Charger l'image trees de Matlab [X, map]=imread('~\trees.tif') ;
2. Afficher cette image, puis ses caractéristiques.
3. Calculez et comparez la taille d'une image RGB avec la même image convertie en image indexé ?

La palette d'une image permet de définir la couleur de chaque pixel en fonction de sa valeur. Affichez l'image précédente avec une palette de niveau de gris, en utilisant la fonction colormap. Essayez de visualiser l'image I avec différentes palettes. Définissez votre palette personnelle. Conclure sur le fonctionnement et l'utilité de la palette. Aide : -help **colormap** -help **colormapeditor** -**colormap**('gray')

La profondeur

La profondeur d'une image peut être définie comme le nombre de couleurs/niveaux de gris possibles pour l'image. Créez 3 palettes différentes de 20, 100 et 10000 valeurs. Quelle est l'importance de la profondeur pour l'affichage, le traitement d'une image ?

Une image binaire est un tableau de pixels prenant la valeur 0 ou 1.

1. Utiliser les commandes i**m2bw** pour convertir une image en binaire puis affichez le résultat.
2. À partir de **help graythresh,** utilisez cette commande pour binariser l'image.
3. Afficher cette image et comparer avec le résultat précédent.
4. Donner les caractéristiques de cette image.

Propriétés d'une image

Pour l'image en niveaux de gris ima_gray :

1. Afficher les des pixels de la ligne 11 jusqu'à la ligne 14 et de la colonne 44 jusqu'à 50.
2. Afficher la dimension de l'image. Stocker le nombre de ligne dans "nl", nombre de colonne "nc".
3. Mettre la valeur max de chaque ligne dans un vecteur maxlig
4. Faire de même avec le maximum, le minimum et la moyenne
5. Faire un tracé de ces trois vecteurs sur le même graphique.
6. Faire un tracé du maximum en fonction du minimum
7. Calculer la moyenne, le max et le min et la somme de toute l'image, puis de la partie centrale de celle-ci (entre 100 et 200 en x et en y)
8. Tracer l'histogramme de l'image, avec « imhist », puis avec « plot ».

Annexe

Note importante : Matlab occupe une grande partie de la mémoire vive, et pour chaque image rgb de résolution 512x512, et si on considère que chaque pixel occupe 8 bits x 3 (RGB), la taille de cette image sera donc : 512 x 512 x 3 x8 = 6 291 456, Donc 6 Méga pour une petite image !!!

Donc il faut faire attention à ne pas saturer la mémoire vive « RAM » se qui dégrade considérablement les performances de l'ordinateur (il devient trop lourd !!).

Quelques rappels sur les fonctions de manipulation des images :

La commande **imread** charge l'image située à un endroit précis du disque dur.

Exemple : I **= imread**('mondelimage.tif');

imshow affiche l'image, **imtool** affiche l'image avec les options à modifier sur celle-ci.

Syntaxe : **imshow**(I)

La commande **whos** vous donne des informations sur l'image et toutes les variables courantes.

imhist est la commande pour afficher l'histogramme de la distribution des intensités lumineuses de l'image.

Syntaxe : **figure, imhist(I)**,

figure pour que matlab se souvienne de la figure précédente pour ne pas l'écraser.

I2=**histeq**(I),I2 est une nouvelle image avec une distribution moyennée de l'intensité lumineuse. figure, **imhist**(I2) On peut aussi utiliser les commandes **imadjust** et **adapthisteq** pour ajuster son l'intensité lumineuse.

Pour sauvegarder une image sur le disque, on utilise les commandes: **imwrite** (on peut spécifier le format de sauvegarde); la syntaxe est **: imwrite**(I2,'pout2.png') On utilise la commande **imfinfo** pour avoir des renseignements sur ce que matlab enregistre pour la nouvelle image. Syntaxe : **imfinfo** ('pout2.png').

N'hésitez pas de demander de l'aide à Matlab sur les fonctions : faire help + nom de fonction puis valider

imread	**help**	**double**	**imshow**
imagesc	**axis**	**colorbar**	**colormap**
figure	**rgb2gray**	**size**	**zeros**
ones	**for**	**max**	**min**
mean	**sum**	**plot**	**hold**
hist	**true**	**false**	**fspecial**
imfilter	**medfilt2**	**imdilate,**	**imopen**
repmat	**linspace**	**[n p]=uigetfile() ; a=imread(cat(2,p,n), 'jpeg');**	

Pour transformer l'arrière-plan de cette image il faut il faut utiliser la commande: background = **imopen**(I, **strel**('**disk**',15)), pour séparer une forme circulaire « disk » de rayon 15 pixels.

Pour créer la même image avec un arrière-plan uniforme. I2=**imsubtract**(I, background).

[X,map] = rgb2ind(aa, n) convertir une image RGB en une image indexée.

a = ind2rgb(X, map) convertir une image indexée en une image RGB.