

Université Echahid Hamma Lakhder El Oued.

Module : Image, son, vidéo codage et transmission

TP3

Opérations géométriques sur les images, quantification et ré-échantillonnage d'images

Objectifs du TP3 :

Nous commençons ce TP par la réalisation de quelques opérations géométriques sur les images. Par la suite, nous évoquons diverses variantes pour la quantification des images dans le but d'aboutir à la résolution en niveaux de gris déterminant le plus petit changement de niveau de gris discernable dans l'image. Nous achevons ce TP par un exemple de ré-échantillonnage d'une image donnant une nouvelle résolution spatiale qui est sensés déterminer le plus petit détail perceptible dans l'image.

1- Opérations géométriques sur les images

a) Rotation d'images

```
load cape;  
Y=imrotate(X,35);  
subplot(1,2,1); imshow(X,map);  
subplot(1,2,2); imshow(Y,map);
```

b) Modification de la taille d'images

```
load clown;  
Y=imresize(X,2); % Doubler la taille de l'image  
T = imresize(X,[100, 120]);  
subplot(1,2,1); imshow(Y, map);  
subplot(1,2,2); imshow(T, map);
```

Cette commande ne fonctionne pas en tant que telle pour les images couleurs. Afin de réaliser cette transformation sur une image couleur, il suffit que cette image dernière soit d'abord convertie de façon à avoir des valeurs entre 0 et 1, puis de définir avec la fonction **zeros** une image couleur de la taille souhaitée, et enfin pour chaque composante de la remplir avec le résultat de l'application de la fonction **imresize** sur chaque composante couleur.

```
img=imread('autumn.tif');  
imgbis=double(img)/255;  
imgz=zeros(103,172,3);% Les nouvelles dimensions  
imgz(:,:,1)=imresize(imgbis(:,:,1),[103 172]);  
imgz(:,:,2)=imresize(imgbis(:,:,2),[103 172]);  
imgz(:,:,3)=imresize(imgbis(:,:,3),[103 172]);  
subplot(1,2,1); imshow(img);  
subplot(1,2,2); imshow(imgz);
```

2- Quantification d'une image

```
Y=imread('autumn.tif');  
X=rgb2gray(Y);
```

Pour visualiser l'image avec n niveaux de gris, il existe de nombreuses variantes :

- Utiliser les fonctionnalités d'imshow :

```
subplot(1,2,1) ; imshow(X) ;  
n=50; % Choisir le nombre de niveaux de gris  
subplot(1,2,2) ; imshow(X,[0 n]);% Afficher l'image à n niveaux
```

- Quantification « manuelle » :

```
n=5; % Choisir le nombre de niveaux de gris  
d=256/n; % Déterminer le pas de quantification  
Xbis=double(X)/255 ;  
subplot(1,2,1) ; imshow(Xbis) ;  
Z=(floor(Xbis*255/d)/(n-1)); % Quantification  
subplot (1,2,2) ; imshow(Z) ; % Afficher l'image à n niveaux
```

- Quantification en utilisant la fonction gray2ind :

```
n=5; % Choisir le nombre de niveaux de gris  
[X,map]=gray2ind(Xbis,n); % Convertir en image indexée avec n niveaux  
imshow(X,map) % Afficher l'image à n niveaux
```

Que pouvez-vous conclure concernant le choix du nombre n des niveaux de gris ??

3- Ré-échantillonnage d'une image

Nous écrivons une fonction dans **subsample.m** :

```
function Y=subsample(X)  
% SUBSAMPLE sous-échantillonnage d'une image par 2  
% SUBSAMPLE sous-échantillonne une image par un facteur 2 dans chaque direction.  
% I2=SUBSAMPLE(I) retourne dans I2 la version sous-échantillonnée  
% de I. Les colonnes et les lignes sont sous-échantillonnées par 2.  
% Cette fonction utilise une double boucle 'for'  
s=size(X);  
height=s(1); % nombre de lignes de l'image  
width=s(2); % nombre de colonnes de l'image  
% Créer une matrice de zéros de taille moitié dans chaque direction  
Y=zeros(round(height/2),round(width/2));  
% Remplir Y avec les données de X en prenant un pixel sur deux  
for i=1:round(height/2)
```

```
for j=1:round(width/2)
Y(i,j)=X(2*i-1,2*j-1);
end
end
```

Nous ferons appel à la fonction **subsample** comme suit :

```
img=imread('cameraman.tif');
img=double(img)/255;
subimg= subsample(img) ;
imshow(subimg) ;
```

Que remarquez-vous sur l'image **cameraman.tif** ré-échantillonnée ??