***TRAVAIL 2***

***Commande Octave***

Imread() = Lit l’image et affiche sa matrice

Size() = dimension / taille

Imshow = ouvre les images dans des fenêtres

Double () = converti en double

Mean() = moyenne

Ones() = crée une matrice avec des 1

Eig () = diagonalisation, renvoie valeurs propres et vecteurs propres

Diag () = crée une matrice diagonale

Sort () = trie les données dans l’ordre croissant

Sum () = somme

Min () = minimum

Max() = maximum

Unint8 () = convertit en .unint8

***Variable***

I1 = image en couleur

P1, P2, P3 = couche de gris

N1, N2 = taille de la matrice P1 (512 et 512)

V1, V2, V3 = vecteur colonne des données des matrice P1, P2, P3

n = taille du vecteur V1 (1)

W1, W2, W3 = transposée de V1, V2, V3

X = matrice recomposée avec les vecteur colonnes W1, W2, W3

g = moyenne de la matrice X

Y = matrice des données centrée

Yt = transposée de la matrice Y

V = matrice de la variance covariance

vp = matrice diagonale de V

Intot = inertie totale

Taux, taux0 = taux d’inertie

C = matrice des données centrée sur la base des vecteurs propres

Ynew = première colonne de la matrice C

m = minimum de la matrice Ynew

M = maximum de la matrice Ynew

amplitude = différence entre le maximum et le minimum

a =

b =

Xnew =

newIm1 =

Xnew2 =

***Code :***

% Lit l'image et affiche sa matrice

I1=imread('lena\_color\_512.tif');

% récupère la première couche de l'image I1

P1=I1(:,:,1);

% récupère la deuxième couche de l'image I1

P2=I1(:,:,2);

% récupère la troisième couche de l'image I1

P3=I1(:,:,3);

% dimension de la matrice P1 stockée dans n1 et n2

[n1,n2]=size(P1);

close all

% Ouvre toute les couches de gris en même temps et l'image en couleur

imshow(P1);figure;imshow(P2);figure;imshow(P3);figure;imshow(I1);

% qCrée un vecteur colonne V1 à partir des données de la matrice P1 ( de même pour V2 et V3)

V1 = P1(:) ;

V2 = P2(:) ;

V3 = P3(:) ;

% Récupère le nombre de ligne de V1

n=size(V1,1);

% Recompose la matrice à partir des 3 vecteurs colonnes

X=[V1 V2 V3];

% g est la moyenne de la matrice X

g=mean(X);

% Y c'est la matrice centrée de X

Y=X-ones(n,1)\*g;

% Yt est la transposée de Y

Yt=Y';

% Convertir en double afin de les multiplier

Yt = double(Y');

Y = double (Y) ;

%Matrice des variances covariances

V=(1/n)\*Yt\*Y;

% La matrice U est la matrice des vecteurs propres et vp1 et la matrice des valeurs propres

[U,vp1]=eig(V);

%vp vecteur colonne des valeurs propres

vp=diag(vp1);

% vpo tri par ordre décroissant les valeurs propres et i et l'indice de chaque valeurs

[vpo,I]=sort(vp,'descend');

% met dans l'ordre décroissant des valeurs propres les vecteurs propres associés

P(:,:)=U(:,I(:));

% somme des valeurs propres (Inertie total)

Intot=sum(vpo);

% Proportion que prennent chaque valeurs propres (taux d'inertie)

taux=vp/Intot;

% De même

tauxO=vpo/Intot;

% Multiplication des données centrée par la matrice des vecteurs propres triés dans l'ordre décroissant

C=Y\*P;

%Ct=C';

% Premiere colonne de la matrice C

Ynew=C(:,1);

% Minimum de la matrice Ynew

m=min(Ynew);

% Maximum de la matrice Ynew

M=max(Ynew);

% Différence entre le maximum et le minimum

amplitude=M-m;

% Comme on veut récupérer une image NB (sous une seule couche de couleurs) la projection s'effectue sur un seul axe, on déduit que l'axe fonction est un droite affine d'équation y = ax+b

% Le coefficient a se situe entre 0 et 1. En effet, les valeurs sont comprises entre 0 et 255. Pour ne pas dépasser cette valeurs a = 255/amplitude

a=255/amplitude;

% D'après l'équation ax+b+ m =0 et ax + M + b = 255 on déduit que b = -a \* m

b=-a\*m;

% Affecte nos valeurs à l'équation

Xnew=a\*Ynew+b;

% Crée une matrice nulle au format unint8

newIm1=uint8(zeros(n1,n2));

% Affecte les valeurs afin de recrée l'image

for (i=1:n1)

newIm1(:,i)=uint8(Xnew((i-1)\*n2+1:i\*n2)');

end

% Autre méthode pour résumer une image

Xnew2=double(X)\*[0.333;0.333;0.333];

for (i=1:n1)

newIm2(:,i)=uint8(Xnew2((i-1)\*n2+1:i\*n2)');

end

close all

imshow(P1);figure;imshow(P2);figure;imshow(P3);figure;imshow(I1);

figure;imshow(newIm1);figure;imshow(newIm2);