**Matlab code :**

function varargout = imageprocessing(varargin)

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @imageprocessing\_OpeningFcn, ...

'gui\_OutputFcn', @imageprocessing\_OutputFcn, ...

'gui\_LayoutFcn', [] , ...

'gui\_Callback', []);

if nargin && ischar(varargin{1})

gui\_State.gui\_Callback = str2func(varargin{1});

end

if nargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:});

else

gui\_mainfcn(gui\_State, varargin{:});

end

function imageprocessing\_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

guidata(hObject, handles);

function varargout = imageprocessing\_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

function pushbutton1\_Callback(hObject, eventdata, handles)

global X;

global xd;

global rc;

siz = size(X);

%to convert RGB image into gray scale

if(siz(3) == 3)

X = rgb2gray(X);

end;

% inputting the decomposition level and name of the wavelet

n=4;

wname = 'haar';

x = double(X);

NbColors = 255;

map = gray(NbColors);

x = uint8(x);

% A wavelet decomposition of the image

[c,s] = wavedec2(x,n,wname);

% wdcbm2 for selecting level dependent thresholds

alpha = 1.5; m = 2.7\*prod(s(1,:));

[thr,nkeep] = wdcbm2(c,s,alpha,m)

% Compression

[xd,cxd,sxd,perf0,perfl2] = wdencmp('lvd',c,s,wname,n,thr,'h');

disp('Compression Ratio');

disp(perf0);

% Decompression

R = waverec2(c,s,wname);

rc = uint8(R);

% Plot original and compressed images.

subplot(221), image(x);

colormap(map);

title('Original image')

subplot(222), image(xd);

colormap(map);

title('Compressed image')

% Displaying the results

xlab1 = ['2-norm rec.: ',num2str(perfl2)];

xlab2 = [' % -- zero cfs: ',num2str(perf0), ' %'];

xlabel([xlab1 xlab2]);

subplot(223), image(rc);

colormap(map);

title('Reconstructed image');

%Computing the image size

disp('Original Image');

imwrite(x,'original.tif');

imfinfo('original.tif')

disp('Compressed Image');

imwrite(xd,'compressed.tif');

imfinfo('compressed.tif')

disp('Decompressed Image');

imwrite(rc,'decompressed.tif');

imfinfo('decompressed.tif')

% --- Executes on button press in pushbutton2.

function pushbutton2\_Callback(hObject, eventdata, handles)

global rc;

global X;

global xd;

subplot(221);

imhist(rc);

subplot(222);

imhist(xd);

subplot(223);

imhist(X);

% --------------------------------------------------------------------

function File\_Callback(hObject, eventdata, handles)

% --------------------------------------------------------------------

function About\_Callback(hObject, eventdata, handles)

about;

% --------------------------------------------------------------------

function Help\_Callback(hObject, eventdata, handles)

% --------------------------------------------------------------------

function New\_Callback(hObject, eventdata, handles)

global X;

[filename, pathname] = uigetfile('\*.m', 'Pick a MATLAB code file');

if isequal(filename,0) || isequal(pathname,0)

disp('User pressed cancel')

else

X=imread(filename);

imshow(X);

end

% --------------------------------------------------------------------

function Exit\_Callback(hObject, eventdata, handles)

delete(get(0,'Children'))

function axes1\_CreateFcn(hObject, eventdata, handles)

im4=imread('C:\Program Files\MATLAB\R2012a\bin\project\image2.jpg');

imshow(im4);

im3=imread('C:\Program Files\MATLAB\R2012a\bin\project\im3.png');

imshow(im3);

function axes3\_CreateFcn(hObject, eventdata, handles)

im1=imread('C:\Program Files\MATLAB\R2012a\bin\project\im1.jpeg');

imshow(im1);

%%%%new gui for about

function varargout = about(varargin)

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @about\_OpeningFcn, ...

'gui\_OutputFcn', @about\_OutputFcn, ...

'gui\_LayoutFcn', [] , ...

'gui\_Callback', []);

if nargin && ischar(varargin{1})

gui\_State.gui\_Callback = str2func(varargin{1});

end

if nargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:});

else

gui\_mainfcn(gui\_State, varargin{:});

end

% End initialization code - DO NOT EDIT

% --- Executes just before about is made visible.

function about\_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

guidata(hObject, handles);

function varargout = about\_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

im5=imread('C:\Program Files\MATLAB\R2012a\bin\project\shi.jpg');

imshow(im5);