IIT Armour College of Engineering

ILLINOIS INSTITUTE OF TECHNOLOGY

ECE 744

EMBEDDED DIGITAL SYSTEMS FOR TIME-FREQUENCY DISTRIBUTION, SIGNAL MODELING AND ESTIMATION

PROFESSOR: JAFAR SANIIE

DUE DATE: 04/20/2015

IMPLEMENTATION

NAME: KRISHNA PRAMOD KANAKAPURA UMESH

CWID: A20337195

ACKNOWLEDGEMENT

I hereby accept that this work of IMPLEMENTATION done by myself without getting assistance from anyone else.

KRISHNA PRAMOD KANAKAPURA UMESH
(A20337195)

ABSTRACT:

The main objective of this report is to develop a software application and implement the three different Time/Frequency algorithms i.e. Gabor Transform, Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT) which were covered in the course lectures. The software application implemented is in menu driven approach and the algorithms is realized using matlab. The simulated signals and images are discussed in the further sections of the report.

IMPLEMENTED ALGORITHMS:

1. GABOR TRANSFORM:

In image processing, a Gabor filter, named after Dennis Gabor, is a linear filter used for edge detection. Frequency and orientation representations of Gabor filters are similar to those of the human visual system, and they have been found to be particularly appropriate for texture representation and discrimination. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave.

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right)$$

where

$$x' = x\cos\theta + y\sin\theta$$

and

$$y' = -x\sin\theta + y\cos\theta$$

2. DISCRETE WAVELET TRANSFORM (DWT):

In numerical analysis and functional analysis, a discrete wavelet transform (DWT) is any wavelet transform for which the wavelets are discretely sampled. As with other wavelet transforms, a key advantage it has over Fourier transforms is temporal resolution: it captures both frequency and location information (location in time).

- Convert a signal into a series of wavelets.
- Provide a way for analyzing waveforms, bounded in both frequency and duration.
- Allow signals to be stored more effectively than by Fourier Transform.
- Be able to better approximate real-world signals.
- Well-suited for approximating data with sharp discontinuities.

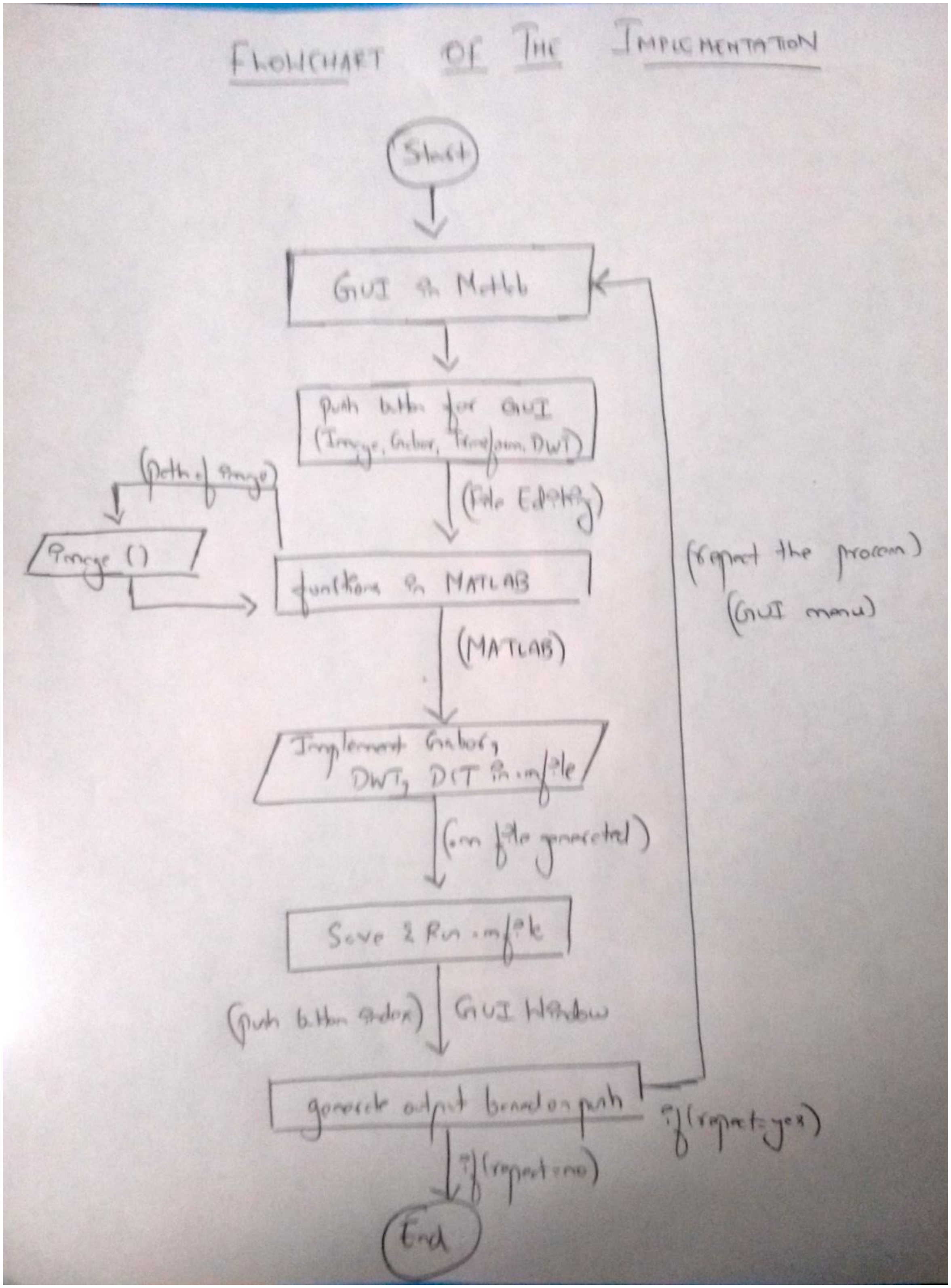
3. DISCRETE COSINE TRANSFORM (DCT):

The DCT is closely related to the Discrete Fourier Transform. The DCT, however, has better energy compaction properties, with just a few of the transform coefficients representing the majority of the energy in the sequence. The energy compaction properties of the DCT make it useful in applications such as data communications. A discrete cosine transform (DCT) expresses a finite sequence of data points in terms of sum of cosine functions oscillating at different frequencies. DCTs are important to numerous applications in science and engineering, from lossy compression of audio (e.g. MP3) and images (e.g. JPEG) (where small high-frequency components can be discarded), to spectral methods for the numerical solution of partial differential equations.

IMPLEMENTATION:

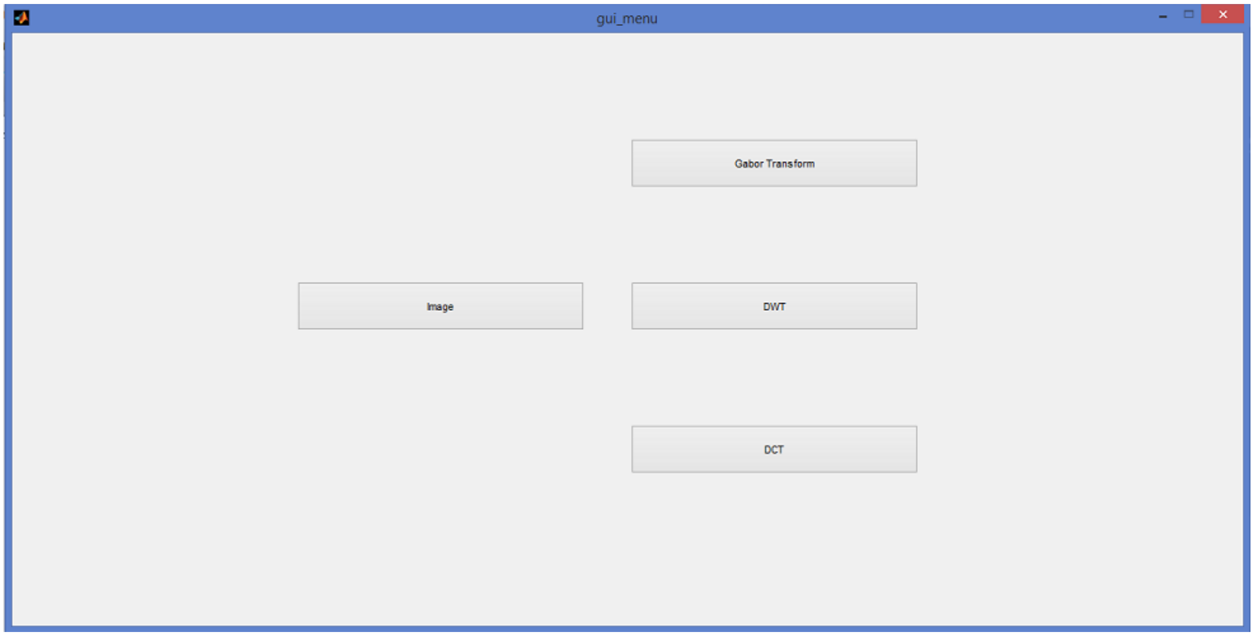
1. The first step is to perform GUI in matlab to implement the different T/F transforms in menu driven approach.
2. Now in the GUI, we have to create push buttons for Image, Gabor Transform, DWT and DCT.
3. After editing the GUI and saving, we get the matlab code for menu driven approach in .m file (gui_menu.m).
4. In the matlab code for menu driven approach, we can find the functions for Image, Gabor Transform, DWT and DCT.
5. In the Image function we have to write the path for the image file.
6. Then we have to write the matlab code to implement Gabor Transform in .m file (gab_full.m) and call the required function which performs Gabor Transform in the menu driven .m file.
7. Now we have to write the matlab code to implement DWT in .m file (DWT_full.m) and call this function in the menu driven .m file.
8. Now we have to write the matlab code to implement DCT in .m file (DCT_full.m) and call this function in the menu driven .m file.
9. Now we have to save the menu driven .m file and run the program to implement the application.
10. In the GUI window whichever push button is clicked its corresponding output will be generated and displayed on the screen.

FLOWCHART OF THE IMPLEMENTATION:

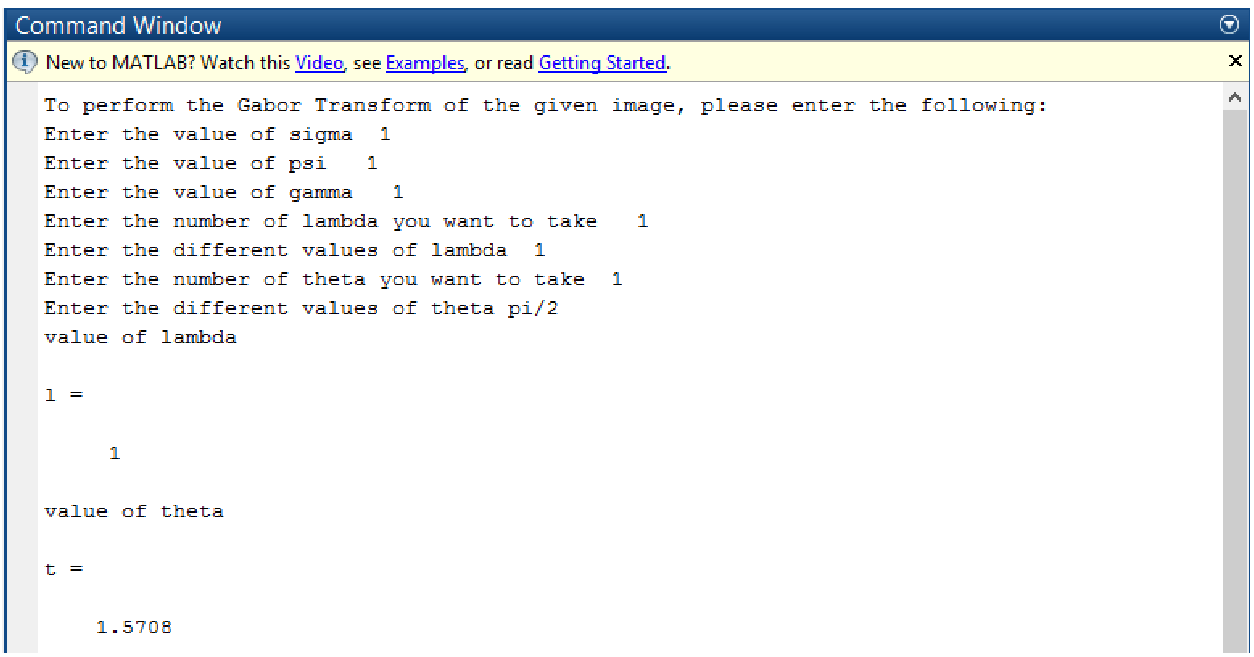


OBTAINED RESULTS:

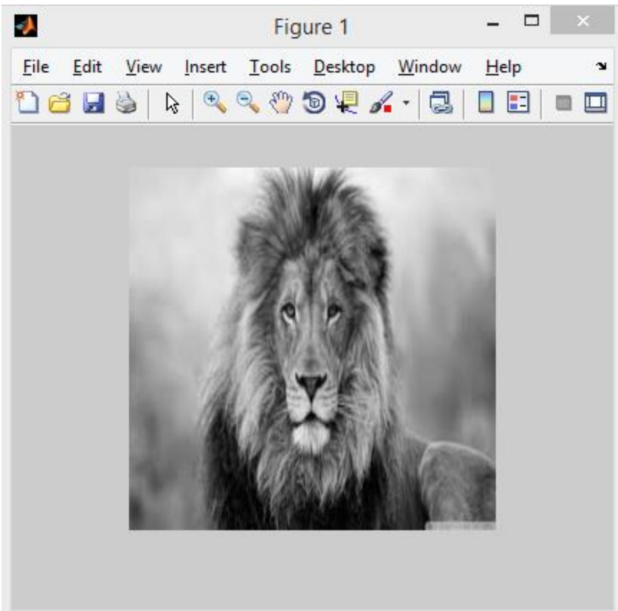
The screenshot of the menu driven approach implemented.



An example of Gabor Transform input and output file:



```
output of gabor filter will be

g1 =

    0.0000    0.0001    0.0006    0.0010    0.0006    0.0001    0.0000
    0.0001    0.0016    0.0071    0.0116    0.0071    0.0016    0.0001
    0.0006    0.0071    0.0316    0.0522    0.0316    0.0071    0.0006
    0.0010    0.0116    0.0522    0.0860    0.0522    0.0116    0.0010
    0.0006    0.0071    0.0316    0.0522    0.0316    0.0071    0.0006
    0.0001    0.0016    0.0071    0.0116    0.0071    0.0016    0.0001
    0.0000    0.0001    0.0006    0.0010    0.0006    0.0001    0.0000

>>
```



An example for DWT input and output files:

gb_full.m × gui_menu.m × DCT_full.m × DWT_full.m ×

```
function dwt_full()

clc;
clear all;
close all;

map=gray(256);

%n=1:256;
%x=sin((pi/64)*n)+0.01*(n-100);

% get db filter length 6
[Lo_D,Hi_D,Lo_R,Hi_R] = wfilters('bior5.5');

% 2D dwt
img=imread('lion.jpg');
figure();
imshow(img);

% first dwt
[P, Q]=size(img);

% dwt in row
dwt_row_img=zeros(P, Q);
tmp=zeros(1, Q);
for j=1:P
    tmp(1, 1:Q)=img(j, 1:Q);
    tmp(1, 1:Q)=dwt(tmp, Lo_R);
    dwt_row_img(j, 1:Q)=tmp(1, 1:Q);
end
```

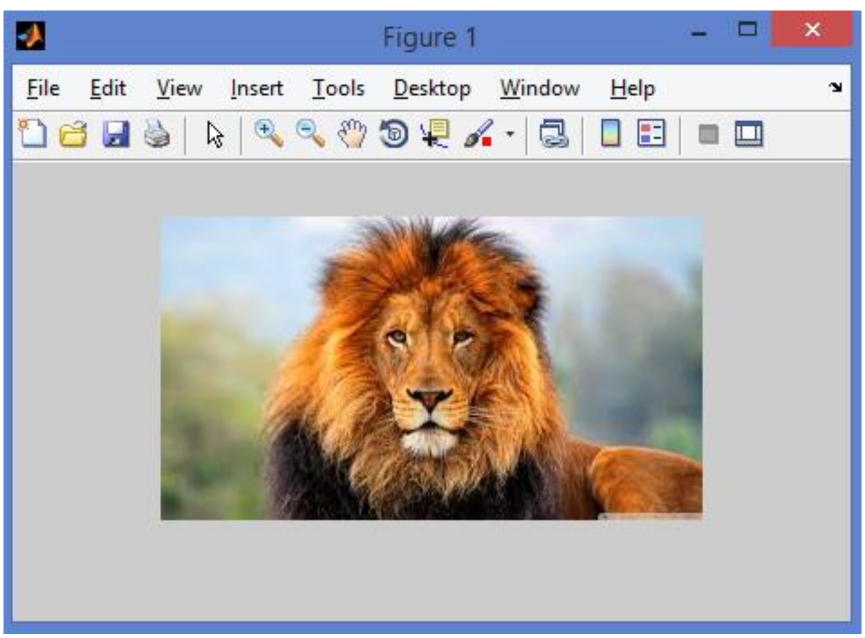
gb_full.m × gui_menu.m × DCT_full.m × DWT_full.m ×

```

figure();
imshow(dwt_row_img, map);


% dwt in column
tmp=zeros(1, P);
dwt1_img=zeros(P, Q);
for j=1:Q
    tmp(1, 1:P)=dwt_row_img(1:P, j)';
    tmp(1, 1:P)=dwt(tmp, Lo_R);
    dwt1_img(1:P, j)=tmp(1, 1:P)';
end

figure();
imshow(dwt1_img, map);

%second dwt

img=zeros(P, Q);
img(1:P, 1:Q)=dwt1_img(1:P, 1:Q);
[P, Q]=size(img);
% dwt in row
dwt_row_img=zeros(P, Q);
tmp=zeros(1, Q);
for j=1:P
    tmp(1, 1:Q)=dwt1_img(j, 1:Q);
    tmp(1, 1:Q)=dwt(tmp, Lo_R);
    dwt_row_img(j, 1:Q)=tmp(1, 1:Q);
end
```

gb_full.m | gui_menu.m | DCT_full.m | DWT_full.m

```matlab

figure();
imshow(dwt_row_img,map);

% dwt in column
tmp=zeros(1, P);
dwt2_img=dwt1_img;
for j=1:Q
    tmp(1, 1:P)=dwt_row_img(1:P, j)';
    tmp(1, 1:P)=dwt(tmp, Lo_R);
    dwt2_img(1:P, j)=tmp(1, 1:P)';
end

figure();
imshow(dwt2_img,map);

end


function g = dwt(f,h)


N = length(h);
c = f;
h0  = fliplr(h);                                % Scaling filter
h1 = h;  h1(1:2:N) = -h1(1:2:N);                % Wavelet filter

L = length(c);
c = [c(mod((-(N-1):-1),L)+1) c];                % Make periodic

```

```matlab

d = conv(c,h1);   d = d(N:2:(N+L-2));          % Convolve & d-sample
c = conv(c,h0);   c = c(N:2:(N+L-2));          % Convolve & d-sample

g = [c,d];                                      % The DWT
end

```



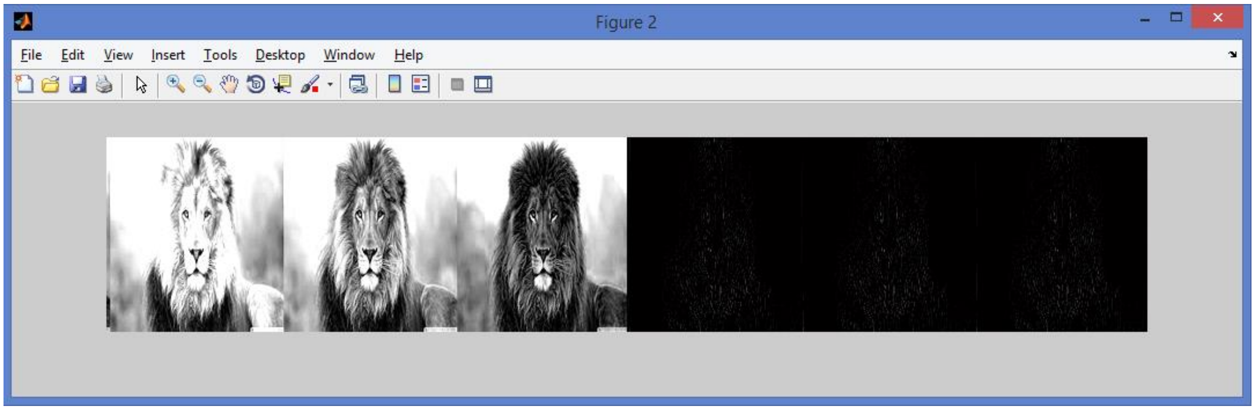


Figure 2
File Edit View Insert Tools Desktop Window Help

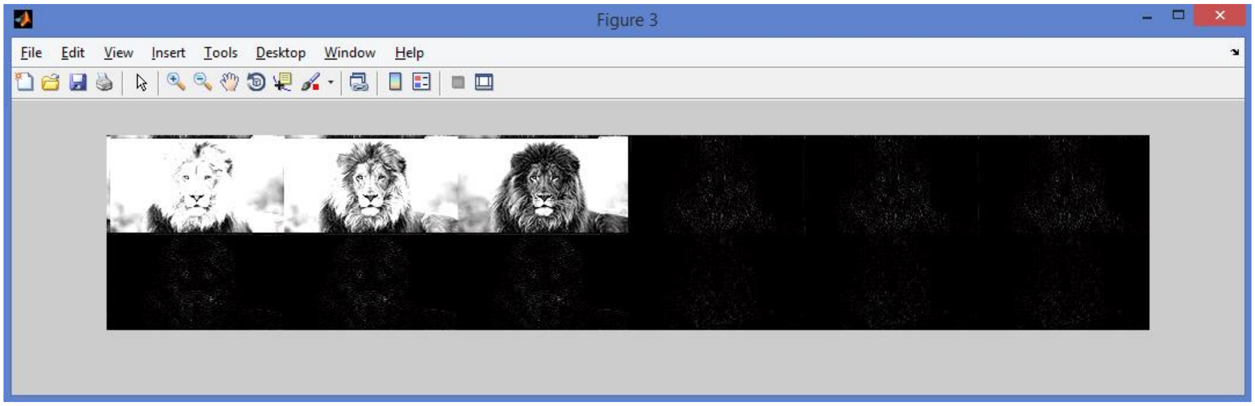


Figure 3
File Edit View Insert Tools Desktop Window Help

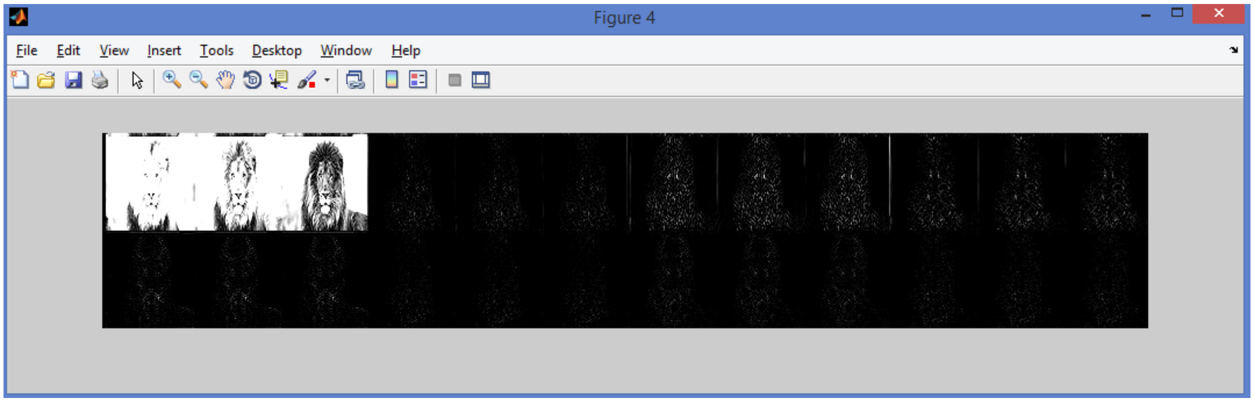


Figure 4
File Edit View Insert Tools Desktop Window Help

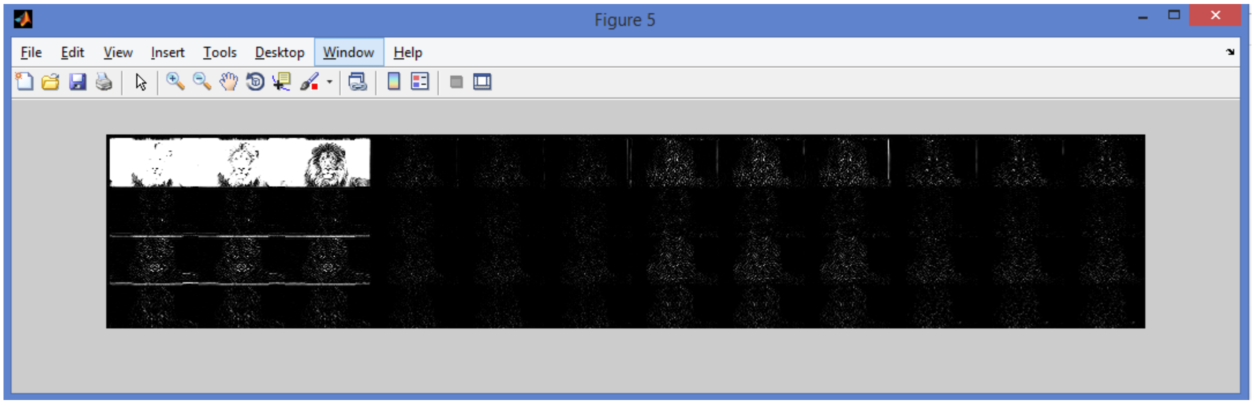


Figure 5
File Edit View Insert Tools Desktop Window Help

An example for DCT input and output files:

gb_full.m × gui_menu.m × DCT_full.m × DWT_full.m ×

```
function DCT_full()


img = imread('lion.jpg');
[p,q] = size(img(:,:,1));
p = floor(p/8)*8;
q = floor(q/8)*8;

A = rgb2gray(img);

figure();
imagesc(A);
title('original');
colormap(gray);

% quantization matrix
Q = [16  11  10  16  24  40  51  61; ...
     12  12  14  19  26  58  60  55; ...
     14  13  16  24  40  57  69  56; ...
     14  17  22  29  51  87  80  62; ...
     18  22  37  56  68 109 103  77; ...
     24  35  55  64  81 104 113  92; ...
     49  64  78  87 103 121 120 101; ...
     72  92  95  98 112 100 103  99];

Y = zeros(p,q);

% loop over 8x8 blocks
for blockx = 1:p/8
for blocky = 1:q/8
```

gb_full.m × gui_menu.m × DCT_full.m × DWT_full.m ×

```
    block = double(A(blockx*8-7:blockx*8, blocky*8-7:blocky*8))-128; % get block, and shift it
    block = dct2(block);                                    % discrete cosine transform
    block = round(block ./ Q);                              % quantization
    Y(blockx*8-7:blockx*8, blocky*8-7:blocky*8) = block;
end
end
figure();
imagesc(Y);
title('DCT of image');
colormap(gray);


A = uint8(zeros(p,q));

% reconstruct Y channel
for blockx = 1:p/8
for blocky = 1:q/8
    block = Y(blockx*8-7:blockx*8, blocky*8-7:blocky*8); % get block
    block = block.*Q;                                   % dequantization
    block = idct2(block);                               % inverse discrete cosine transform
    block = uint8(block + 128);                         % shift
    A(blockx*8-7:blockx*8, blocky*8-7:blocky*8) = block;
end
end

figure();
imagesc(A);
title('reconstructed image');
colormap(gray);
end
```

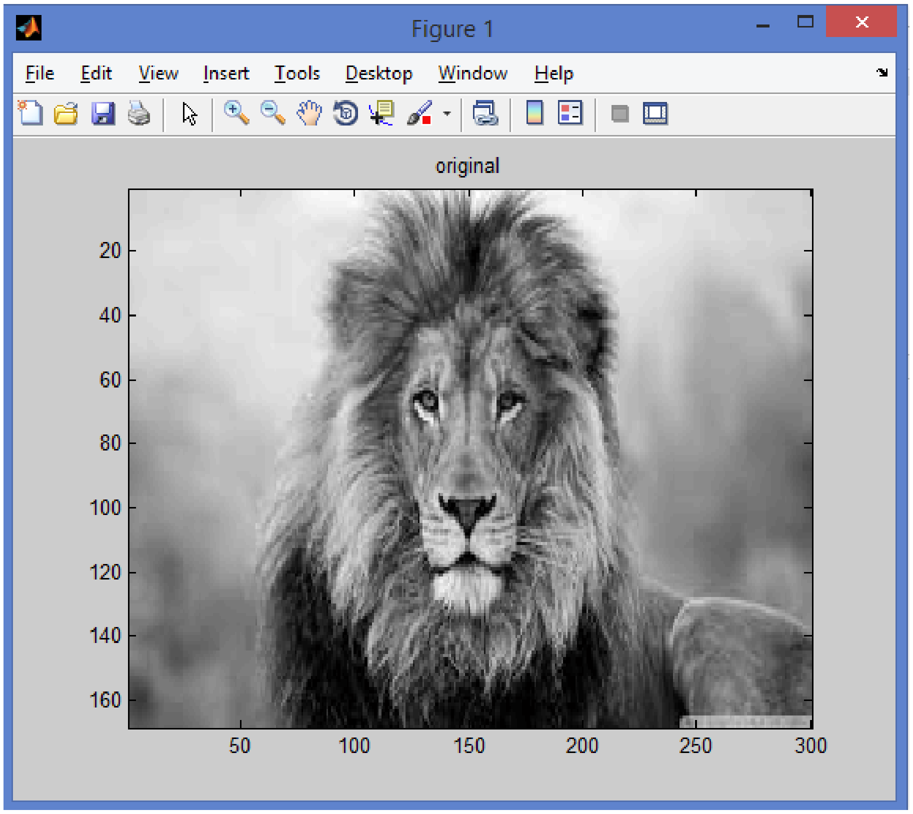


Figure 1
File Edit View Insert Tools Desktop Window Help
original
20
40
60
80
100
120
140
160
50
100
150
200
250
300

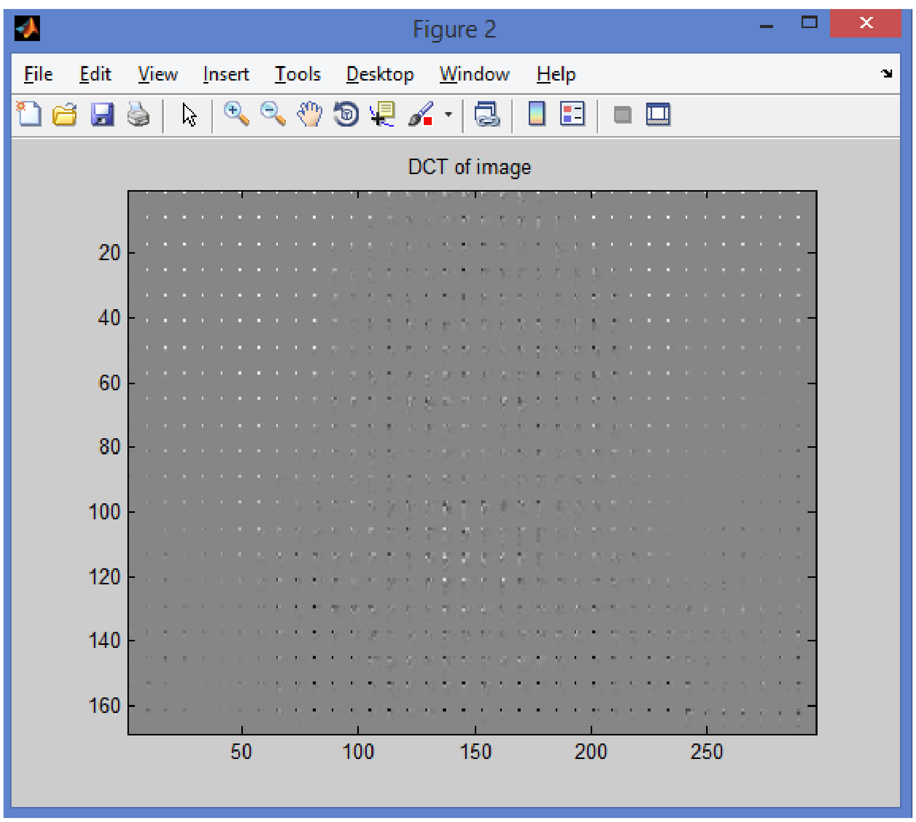
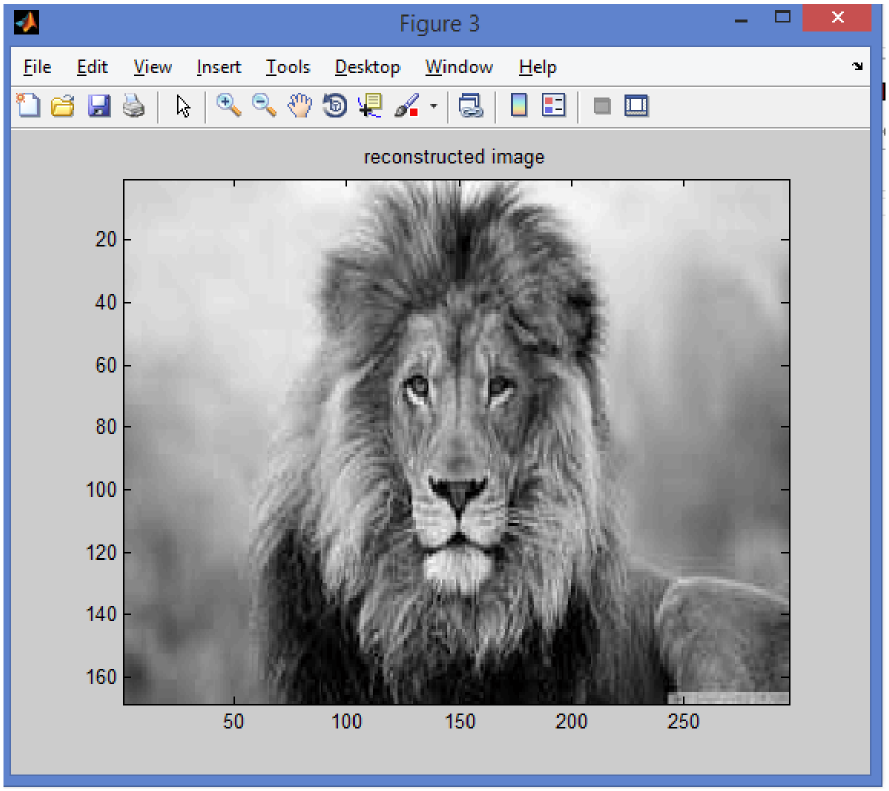


Figure 2
File Edit View Insert Tools Desktop Window Help
DCT of image
20
40
60
80
100
120
140
160
50
100
150
200
250



SOURCE CODE OF IMPLEMENTATION:

```
function varargout = gui_menu(varargin)
% GUI_MENU MATLAB code for gui_menu.fig
%      GUI_MENU, by itself, creates a new GUI_MENU or raises the existing
%      singleton*.
%
%      H = GUI_MENU returns the handle to a new GUI_MENU or the handle to
%      the existing singleton*.
%
%      GUI_MENU('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in GUI_MENU.M with the given input arguments.
%
%      GUI_MENU('Property','Value',...) creates a new GUI_MENU or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before gui_menu_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to gui_menu_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help gui_menu
```

```
% Last Modified by GUIDE v2.5 20-Apr-2015 18:42:25

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @gui_menu_OpeningFcn, ...
                   'gui_OutputFcn',  @gui_menu_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
end
% --- Executes just before gui_menu is made visible.
function gui_menu_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to gui_menu (see VARARGIN)
% Choose default command line output for gui_menu
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
% UIWAIT makes gui_menu wait for user response (see UIRESUME)
% uiwait(handles.figure1);
end
% --- Outputs from this function are returned to the command line.
function varargout = gui_menu_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;
end
% --- Executes on button press in gb_tr.
function gb_tr_Callback(hObject, eventdata, handles)
% hObject    handle to gb_tr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
gb_full();
end
% --- Executes on button press in st_tr.
function st_tr_Callback(hObject, eventdata, handles)
% hObject    handle to st_tr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
DCT_full();
end
```

```
% --- Executes on button press in hb_tr.
function hb_tr_Callback(hObject, eventdata, handles)
% hObject    handle to hb_tr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
DWT_full();
end
% --- Executes on button press in Load_image.
function Load_image_Callback(hObject, eventdata, handles)
% hObject    handle to Load_image (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
img= ('C:\Users\KriahnaPramod\Desktop\Menu Interface\lion.jpg');
figure(1);
imshow(img);
end
```

PROCEDURE TO EXECUTE THE CODES:

- First we have to unzip the file folder “implementation.zip”.
- Then extract all the files to the matlab folder where it is installed on the drive.
- Change the path of the Editor window according to where the all the matlab file exists.
- If needed to load a different image then change the path of the variable ‘img’ in the menu driven matlab code in the Load_image_Callback() function.
- Now we can save and run the ‘gui_menu.m’ matlab file to generate the implemented software.

RECOMMENDATION OF FUTURE WORK:

We can further edit the codes of the Gabor Transform, DWT and DCT to obtain a more enhanced and efficient output. The menu driven approach can be designed in such a way so that we can load the image at that instant and get the three different T/F transforms for the loaded image. These Transforms are useful in image-processing and signal processing applications.

REFERENCES:

[1] Course Lectures Slides.

[2] http://en.wikipedia.org/wiki/Discrete_wavelet_transform

[3] http://en.wikipedia.org/wiki/Gabor_transform

[4] http://en.wikipedia.org/wiki/Discrete_cosine_transform

[5] https://www.youtube.com/watch?v=pfA9VGcDMDs

[6] http://www.mathworks.com/help/pdf_doc/matlab/buildgui.pdf

[7] http://en.wikipedia.org/wiki/Gabor_filter