ABSTRACT

Our project is about image compression and reconstruction, in this project we are going to use Fast Fourier transforms, we are going to convert time domain into frequency domain which computer can analyze easily. By using FFT we obtain an image which has lower frequency and gradually goes to higher frequency, with the help of this frequency patterns a computer can construct the original image. This is done with the help of inverse Fast Fourier transforms.

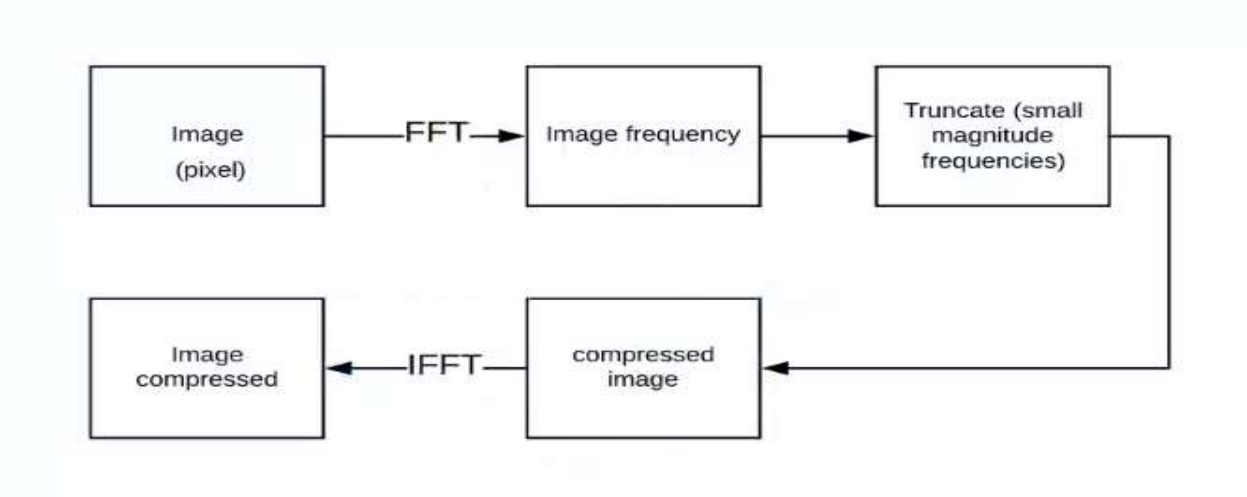
By using the above concept, we are going to code it with the help of MATLAB SOFTWARE.

INTRODUCTION

Image coding is a kind of method by which the data compression can achieve with the ensure quality of images and with reduce code rate. Thus, by using image coding we can get the goal for saving bandwidth or space and it may also be provided for multimedia computer processing. The Fourier transform is an important image processing tool which is used to decompose in image into its sine and cosine components. The output of the transformation represents the image in the Fourier or frequency domain while the input image is the spatial domain equivalent.

METHODOLOGY

FLOW CHART



FOURIER SERIES

A Fourier series may be defined as an expansion of a function in a series of sines and cosines such as

$$f(x) = \tfrac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n \cos(nx) + \sum_{n=1}^{\infty} b_n \sin(nx),$$

Where,

$$a_0 = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\,dx$$

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos(nx)\,dx$$

$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin(nx)\,dx$$

FOURIER TRANSFORMS

The Fourier transform of F(u) of a single variable, continuous function, f(x)

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx$$

The Fourier transform of F(u,v) of a double variable, continuous function, f(x,y)

$$F(u,v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) e^{-j2\pi(ux+vy)} dxdy$$

DISCRETE FOURIER TRANSFORM

The Discrete Fourier Transform (DFT) is the equivalent of the continuous Fourier Transform for signals known only at N instants separated by sample times T (i.e., a finite sequence of data).

Let f(t) be the continuous signal which is the source of the data. Let N samples be denoted f[0],f[1],f[2],….f[k],….f[N-1].

The Fourier Transform of the original signal, f(t), would be

$$F(j\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt$$

We could regard each sample f[k] as an impulse having area f[k]. Then, since the integrated exists integrated exists only at the sample points:

$$X_p = \sum_{n=0}^{N-1} x_n \, e^{-j\frac{2\pi}{N}np}, \quad p \in \{0, 1, ..., N-1\}$$

This Equation can be rewritten in matrix form:

$$X_p = \sum_{n=0}^{N-1} x_n \times W^{np}, \quad p \in \{0, 1, ..., N-1\}$$

as follows:

$$\begin{bmatrix} X_0 \\ X_1 \\ \cdots \\ X_{N-1} \end{bmatrix} = \underbrace{\begin{bmatrix} W^0 & W^0 & W^0 \cdots & W^0 \\ W^0 & W^1 & W^2 \cdots & W^{N-1} \\ \vdots & \vdots & \vdots \cdots & \vdots \\ W^0 & W^{N-1} & W^{2(N-1)} \cdots & W^{(N-1)^2} \end{bmatrix}}_{N \times N \text{ matrix}} \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_{N-1} \end{bmatrix}$$

complex numbers can be computed off-line and stored

IMPLEMENTATION

- The given input image is transformed to a frequency domain using FFT2 function in MATLAB.
- In the Fourier transformed image there is lot of data which is not useful for constructing an image , by truncating small value frequencies we are going to remove the extra data.
- The extra data is truncated by removing small value frequencies(i.e., when compared to large value frequencies) i.e., many values will be removed so that image will be compressed
- The compressed data will be reconstructed using IFFT2 so the Size occupied by the image will be reduced by minimum 90%
- if the image is large, then the data that should be truncated will be more
- If the image is small, then the data that should be truncated will be less

MATLAB CODE

```
clear all, close all,clc
A = imread('1.jpg');
figure(3)
imshow(A)

Abw2=rgb2gray(A);
```

➢ Here the Size of matrix that is no of rows are stored in nx and no of columns are stored in ny

```
[nx,ny] = size(Abw2);
```

- Original(color) image is converted to black and white

```
figure(1),subplot(2,2,1), imshow(Abw2)
title('Original Image','FontSize',16)
```

- The black&white image is fourier transformed such that we get the image frequency domain of the original image

```
transformed = fft2(Abw2);
```

- To obtain the difference between the small values and large values log is used

```
F = log(abs(fftshift(transformed))+1);
```

- This command converts matrix to intensity image of values in range 0 to 1

```
F = mat2gray(F);
figure(4)
imshow(F,[]);

count_pic = 2;
```

- This for loop removes the data that is smaller than the max value in the matrix
- From this for loop we obtain three types of images based on removing values.
- Here the values that are truncated are
    1)10,000 times smaller than the maximum value
    2)50,000 times smaller than the maximum value

3)1,000 times smaller than the maximum value

```
for thresh = .1*[0.001 0.005 0.01]*max(abs(transformed(:)))
```

- The values that are smaller are stored in ind variable and all others are kept as zeroes

```
ind=abs(transformed)>thresh;
```

- The no of values are counted to calculate the percentage

```
count =sum(ind(:));
```

- The matrix having high values is stored into AhatFilt variable

```
    AhatFilt = transformed.*ind;
    percent=count/(nx*ny)*100;
    Afilt = uint8(ifft2(AhatFilt));
    figure(1), subplot(2,2,count_pic)
    imshow(Afilt);
    count_pic = count_pic + 1;
    drawnow
    title([num2str(percent) '% of FFTbasis'],'FontSize',14)
end


figure(5),subplot(2,2,1)
Anew = imresize(Abw2,.1);
surf(double(Anew));


title('original image');
figure(5),subplot(2,2,2)
Anew = imresize(Afilt,.1);
surf(double((Anew)));
title('Filtred image');
```

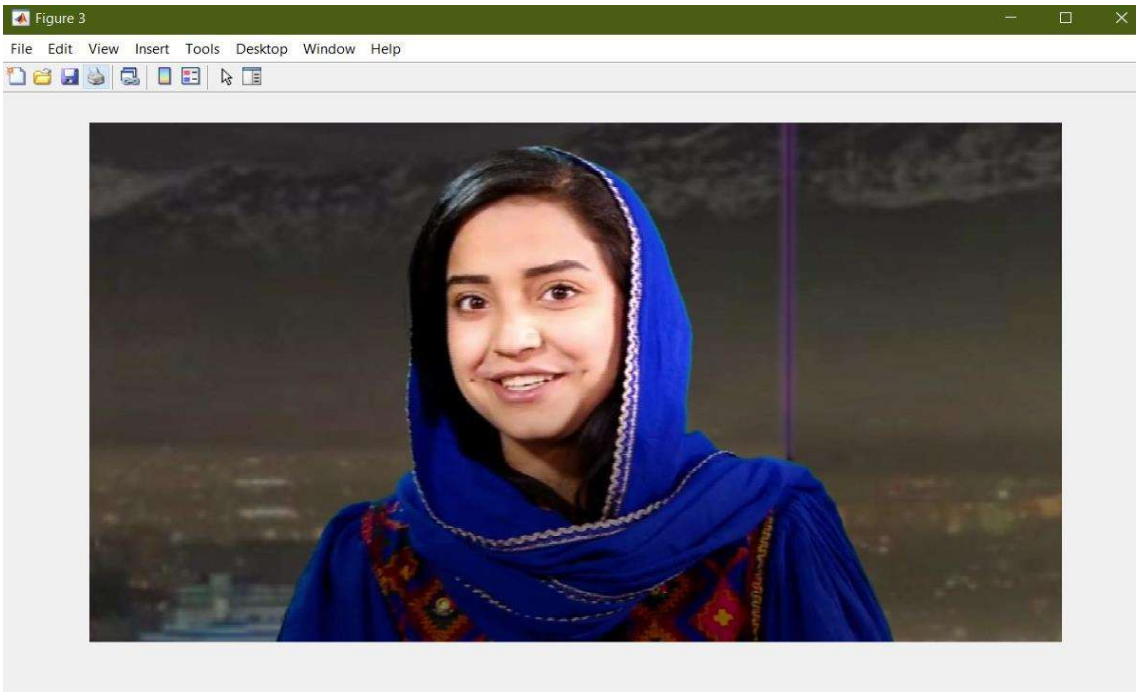
```
figure(5),subplot(2,2,3)
Anew = imresize(F,.1);
surf(double((Anew)));

title('Fourier Transformed');

figure(5),subplot(2,2,4)
Anew = imresize(abs(AhatFilt),.1);
surf(double((Anew)));
title('Filtered Fourier');
```

RESULT

INPUT:



OUTPUT

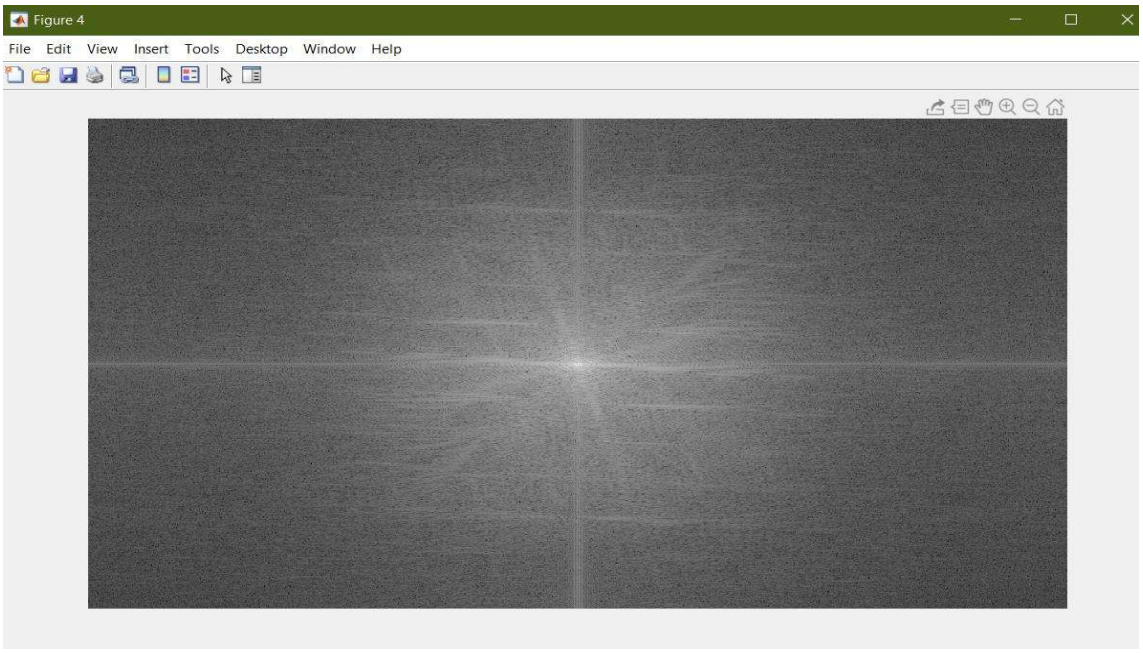


Figure 4
File Edit View Insert Tools Desktop Window Help


Original Image
9.6932% of FFT basis
2.3744% of FFT basis
0.97252% of FFT basis