

# Programming Assignment - 3

Niraj J. Dedhia

March 14, 2016

## Abstract

In this programming assignment we were supposed to understand the concepts of FFT, Inverse FFT, DCT, Inverse DCT, Hybrid Image, Cooley–Tukey algorithm and visualization.

## 1 Summary

This assignment was divided in four parts. 1) Implementation of DFT, IDFT using Cooley Tukey algorithm , 2) Visualization part. 3) Implementation to create Hybrid Image and 4) Implementation of DCT which was for bonus marks.

## 2 Methodology

### 2.1 DFT and IDFT

Fast Fourier Transform(FFT) algorithm computes the Discrete Fourier transform (DFT) of a sequence. FFT converts the signal from one domain to the frequency domain and IFFT converts the signal from frequency domain to the space or time domain. I have implemented Cooley Tukey algorithm for computation of FFT and IFFT for 1D and 2D. Following are the formulas for FFT1D and IFFT1D and steps for Implementation of FFT2D-IFFT2D. *Originalimage = IFFT2D(FFT2D(originalimage))*

$$X_k = \sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{2\pi i}{N}(2m)k} + \sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N}(2m+1)k} \quad \Bigg| \quad X_k = \frac{1}{N} \sum_{m=0}^{N/2-1} x_{2m} e^{\frac{2\pi i}{N}(2m)k} + \frac{1}{N} \sum_{m=0}^{N/2-1} x_{2m+1} e^{\frac{2\pi i}{N}(2m+1)k}$$

FFT IFFT

Figure1 : FFTandIFFTFormula

- Read an image converted it in to gray-scale and double.
- Perform FFT for each row in Image.
- Take transpose of Image.
- Again perform FFT over each row in an Image (i.e. column transformation).
- Take transpose of Image and this is the final result.
- For inverse FFT perform the same steps the only difference is instead of FFT perform IFFT.

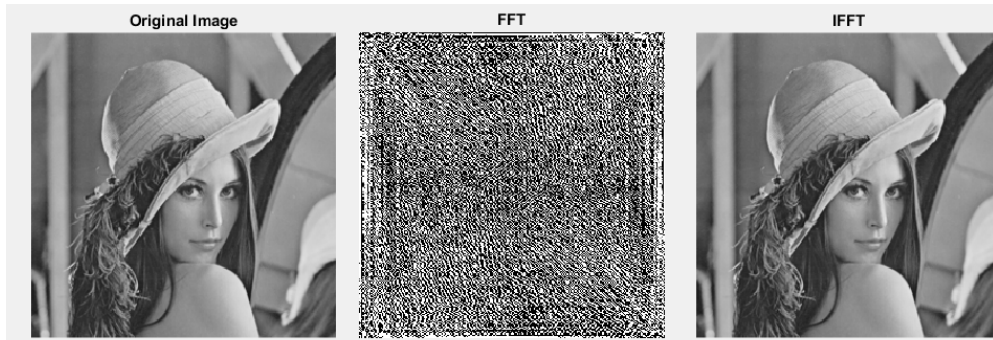


Figure2 :  $OriginalImage \rightarrow FFT \rightarrow IFFT$

## 2.2 Hybrid Image

Two images are combined in a way that resultant image after combining them looks different based on the viewing distance. Here I have taken two images 1) Dog and 2) Cat and created hybrid image (figure 2) by applying following algorithm.

- Read two images converted it in to gray-scale and double.
- Padded images with the nearest value of power of 2.
- Defined Gaussian low pass and high pass filters of size of padded images.
- Took DFT of Images and filters.
- Performed element wise multiplication of first image with high pass filter and second image with low pass filter and performed(A) .
- Subtracted the high frequency image with its original image (B).
- Add both the images (A + B).
- Removed the padding and displayed it.



Figure3 :  $HybridImage$

## 2.3 Visualization

We were supposed to fill the stub with our implementation for showing the flow how this FFT will turned in to IFFT, pixel by pixel visualization of conversion of FFT to IFFT. There are total six components a) Original Image, b) Log magnitude of DFFT of image, c) Used DFFT from 'b' component, d) Current component inserted in a component 'c' e) Scaled sine of component 'd', f) Inverse DFFT of Used DFFT component. Following are the steps for visualization.

- Read an image convert it in to gray-scale and double displayed that in first component.
- Computed DFFT of input image and calculated the log10 of its magnitude. Then normalized it and displayed in component 2.
- Perform the following steps iteratively for continuous visualization.
- Searched maximum value from 'b' component(Brightest pixel value) and inserted it in component 'c' (This is also log magnitude value).
- Took inverse DFT of de-normalized Used DFFT component. Displayed it in 'f' component.
- For every iterations initialized the current component and scaled sine component.
- Took the current used pixels and display it in 'd' component.
- Computed the inverse DFFT of current component and displayed it in 'e' component.

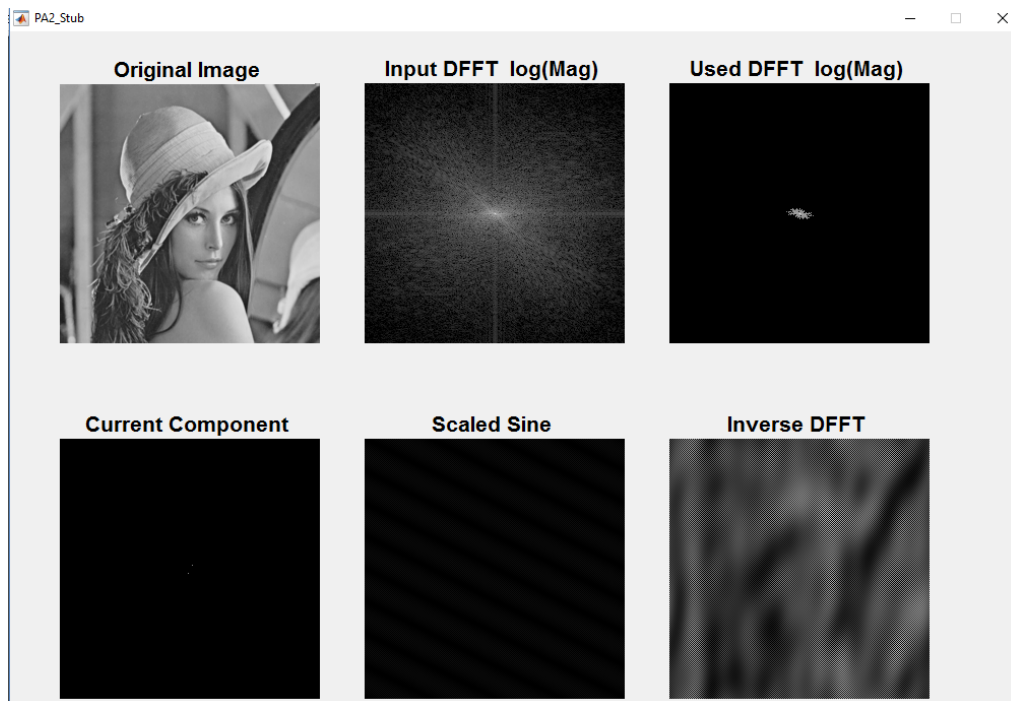


Figure4 : Visualization(After Few Iterations)

## 2.4 DCT and IDCT

Discrete cosine transform(DCT) expresses a finite sequence of data points in terms of a sum of cosine functions keep changing at different frequencies. Its easy as it deals with real part only. Following are the formulas for DCT1D and IDCT1D and steps for Implementation of DCT2D-IDCT2D.  $Originalimage = IDCT2D(DCT2D(originalimage))$

$$\begin{array}{l}
 Y[k] = \sqrt{\frac{2}{N}} E_k \sum_{n=0}^{N-1} X[n] \cos \left[ \frac{(2n+1)\pi k}{2N} \right] \\
 E_k = \begin{cases} \sqrt{\frac{1}{2}} & k=0 \\ 1 & k \neq 0 \end{cases} \quad k, n = \{0, 1, 2, \dots, N-1\} \quad \boxed{\text{DCT}}
 \end{array}
 \quad \Bigg| \quad
 \begin{array}{l}
 X[n] = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} E_k Y[k] \cos \left[ \frac{(2n+1)k\pi}{2N} \right] \\
 E_k = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } k=0 \\ 1 & \text{for } k \neq 0. \end{cases} \quad \boxed{\text{IDCT}}
 \end{array}$$

Figure5 : DCT and IDCT Formula

- Read an image converted it in to gray-scale and double.
- Perform DCT for each row in Image.
- Take transpose of Image.
- Again perform DCT over each row in an Image (i.e. column transformation).
- Take transpose of Image and store it as the final result.
- For inverse DCT perform the same steps the only difference is instead of DCT perform IDCT.

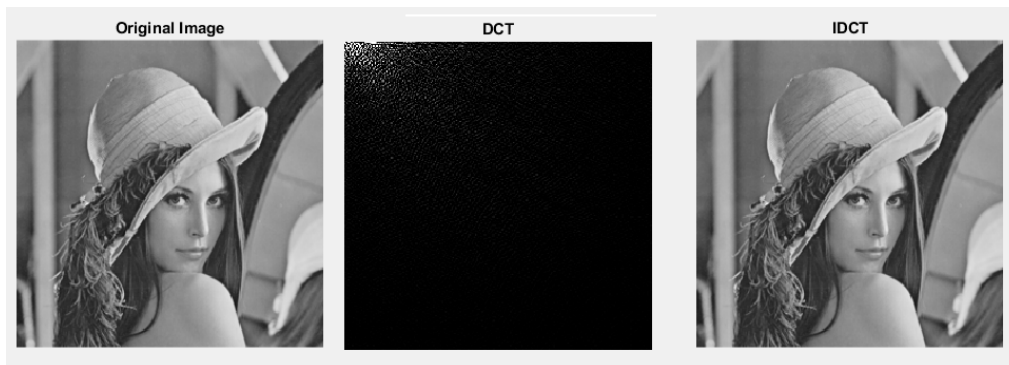


Figure6 : OriginalImage -- > DCT -- > IDCT

### 3 References:

- [1] [https://en.wikipedia.org/wiki/Discrete\\_Fourier\\_transform](https://en.wikipedia.org/wiki/Discrete_Fourier_transform).
- [2] [https://en.wikipedia.org/wiki/Hybrid\\_image](https://en.wikipedia.org/wiki/Hybrid_image).
- [3] [https://en.wikipedia.org/wiki/Discrete\\_cosine\\_transform](https://en.wikipedia.org/wiki/Discrete_cosine_transform).
- [4] <http://ieeexplore.ieee.org.ezproxy.rit.edu/stamp/stamp.jsp>.
- [5] <http://www.mathworks.com/help/>.
- [6] <https://www.overleaf.com>.