E9 222: Signal Processing in Practice
Assignment 2 Report

Bhaskar Karol
S.R. NO.: 04-02-02-10-51-24-1-24076
Department: Electrical Communication Engineering

January 22, 2025

Contents

1 Discrete Cosine Transform (DCT)

1.1 Introduction

The Discrete Cosine Transform (DCT) expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. It is widely used in signal processing and data compression due to its strong energy compaction properties.

1.2 1D Type-II DCT

The 1D Type-II DCT, commonly referred to as "the DCT," is defined for a sequence $x[n]$ of length N as:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right], \quad k = 0, 1, \ldots, N-1.$$

This transform converts the spatial domain data into frequency domain coefficients, facilitating operations like compression and filtering.

1.3 2D Type-II DCT

The 2D Type-II DCT is an extension of the 1D DCT to two dimensions, making it suitable for image processing applications. For an $M \times N$ image $x[m,n]$, the 2D DCT is defined as:

$$X[k_1,k_2] = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} x[m,n] \cos\left[\frac{\pi}{M}\left(m+\frac{1}{2}\right)k_1\right] \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k_2\right],$$

where $k_1 = 0, 1, \ldots, M-1$ and $k_2 = 0, 1, \ldots, N-1$.

1.4 Inverse 2D Type-II DCT

The inverse of the 2D Type-II DCT reconstructs the original image from its DCT coefficients. It is given by:

$$x[m,n] = \frac{1}{MN}\sum_{k_1=0}^{M-1}\sum_{k_2=0}^{N-1} X[k_1,k_2] \cos\left[\frac{\pi}{M}\left(m+\frac{1}{2}\right)k_1\right] \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k_2\right].$$

This inverse transform is essential for reconstructing images after processing in the frequency domain.

2 Problem 1: 1D DCT Implementation

2.1 Objective

Implement the 1D Type-II DCT for a given sequence and compare the results with a built-in DCT function.

2.2 Implementation

A sequence $x[n] = \cos\left(\frac{2\pi k_0 n}{N}\right)$ with $k_0 = 10$ and $N = 64$ was used. The DCT was computed using both a custom implementation and a built-in function.

2.3 Results

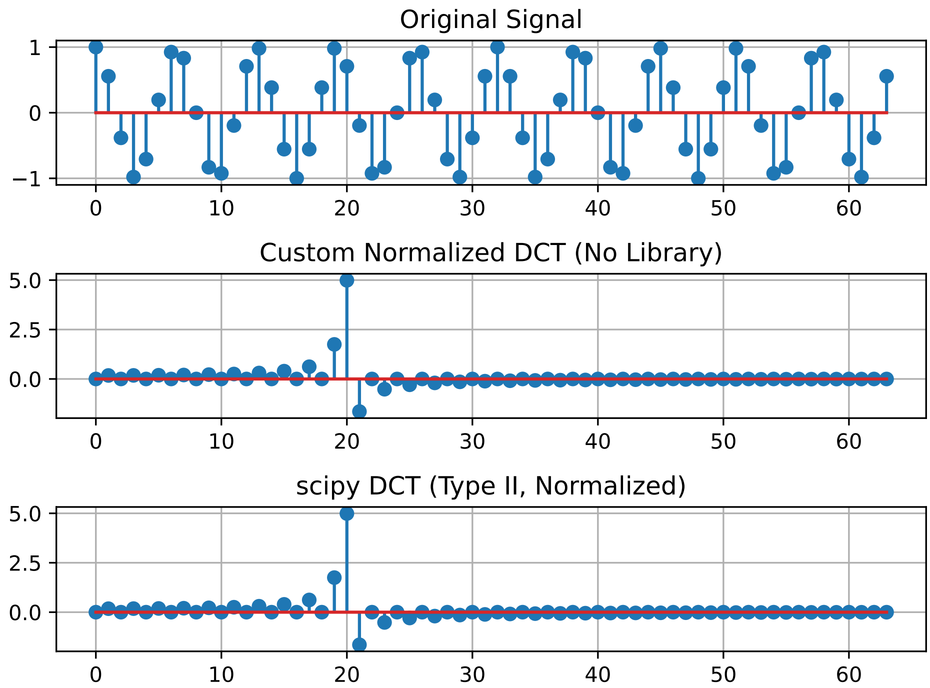

Figure 1: Sequence $x[n]$

3 Problem 2: 2D DCT Implementation

3.1 Objective

Implement the 2D Type-II DCT for a synthetic image and a real image, and compare the results with a built-in 2D DCT function.

3.2 Implementation

A synthetic image $x[m,n] = \cos\left(\frac{2\pi k_1 m}{M}\right)\sin\left(\frac{2\pi k_2 n}{N}\right)$ with $k_1 = 5$, $k_2 = 12$, $M = 48$, and $N = 32$ was used. Additionally, the 'Cameraman' image was processed. The 2D DCT was computed using both custom and built-in functions.

3.3 Results

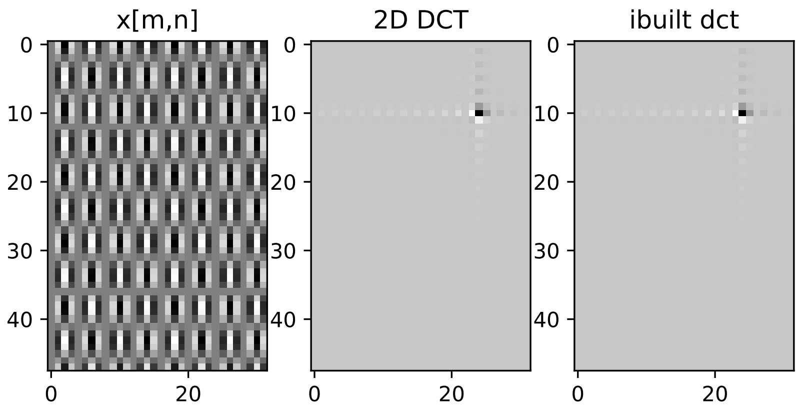

Figure 2: x[n] and 2D DCT and inbuilt DCT

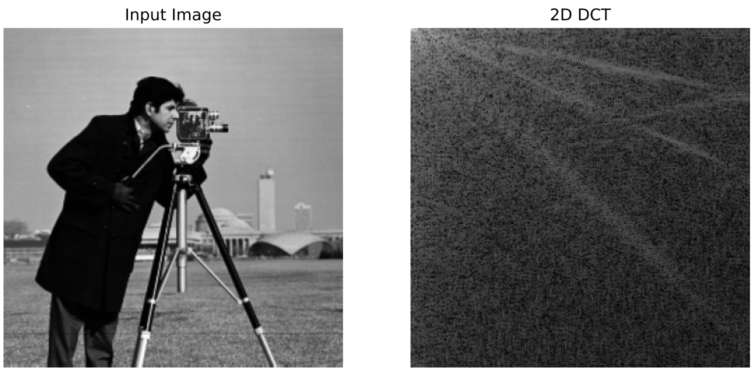

Figure 3: original image and 2D dct of original image in log scale

4 Problem 3: Inverse 2D DCT Implementation

4.1 Objective

Implement the inverse 2D Type-II DCT and use it for image reconstruction from DCT coefficients.

4.2 Implementation

The inverse 2D DCT was used to reconstruct:

- The synthetic image from Problem 2.
- The Cameraman image.

Partial reconstructions were performed by retaining the top m DCT coefficients, where $0 \leq m \leq MN$.

4.3 Results

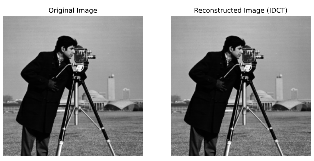

Figure 4: Original image and reconstructed image (IDCT)

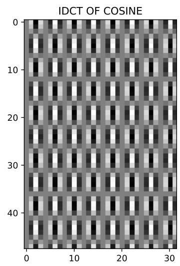

Figure 5: IDCT cosine

Figure 6: Partial reconstructed image $m = 100$

Figure 7: Partial reconstructed image $m = 6553$

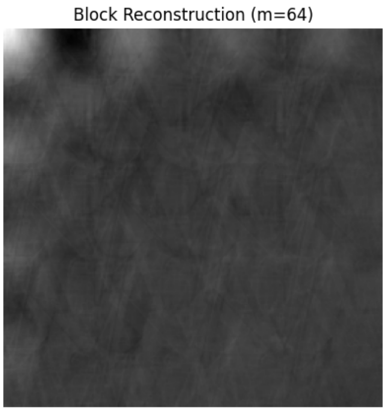

Figure 8: Block 2D DCT $m = 64$

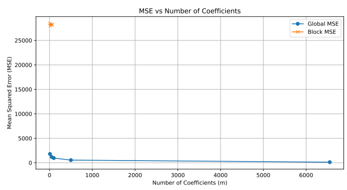

Figure 9: MSE

5 Problem 4: 2D Convolution in Spatial Domain

5.1 Objective

Sharpen a blurred image using 2D convolution with a specified kernel.

5.2 Implementation

The sharpening kernel:

$$K = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

was applied to the image `blurred.png` using a custom 2D convolution implementation.

5.3 Results

Figure 10: Original blurred image and sharpened image after 2D convolution