

EE565: Video Communications

Prof. Kondi

Spring 2007

Homework #1

Total Points: 24

Due: Thursday, February 15, 2007

Problem 1

In this problem, you are asked to develop a fast implementation of the Discrete Cosine Transform (DCT) using Matlab.

$F(u)$, the one-dimensional DCT of a sequence $f(x)$, $x = 0, \dots, N - 1$ is given by

$$F(u) = w(u) \sum_{x=0}^{N-1} f(x) \cos\left[\frac{(2x+1)\pi u}{2N}\right], \quad (1)$$

where

$$w(u) = \begin{cases} \frac{1}{\sqrt{N}} & \text{if } u = 0 \\ \sqrt{\frac{2}{N}} & \text{otherwise,} \end{cases} \quad (2)$$

and $u = 0, \dots, N - 1$.

It is possible to relate the $F(u)$, the DCT of $f(x)$ to $G(u)$, the Discrete Fourier Transform (DFT) of the sequence $g(x)$ where

$$g(x) = \begin{cases} f(x), & 0 \leq x \leq N - 1 \\ f(2N - 1 - x), & N \leq x \leq 2N - 1 \end{cases} \quad (3)$$

The DFT of $g(x)$ is

$$G(u) = \sum_{x=0}^{2N-1} g(x) W_{2N}^{xu}, \quad (4)$$

where $W_N = e^{-j(2\pi/N)}$ and $u = 0, \dots, 2N - 1$.

$f(x)$ and $F(u)$ have length N while $g(x)$ and $G(u)$ have length $2N$. However, there exists the Fast Fourier Transform (FFT) which is a fast implementation of the DFT. Thus, by calculating $G(u)$ and then relating it to $F(u)$, we can greatly reduce the computational complexity of the implementation, when compared with implementing the definition of the DCT.

You are asked to do the following.

1. Analytically find the relationship between $G(u)$ and $F(u)$. Submit all details of the derivation.
2. Write a Matlab function `mydct.m` which, for an input sequence $f(x)$, creates $g(x)$, finds $G(u)$ using the `fft` function of Matlab, derives $F(u)$, the DCT of $f(x)$, from $G(u)$ and returns it.

3. Write a Matlab function `mydct2.m` which calculates the two-dimensional DCT of a two-dimensional matrix (image block) $f(x, y)$. Use your function `mydct.m` in conjunction with row-column decomposition, as discussed in class.
4. Write a Matlab function `slowdct2.m` which directly implements the definition of two-dimensional DCT. Create an arbitrary 8×8 matrix and calculate its DCT using `mydct2.m` and `slowdct2.m`. Show that the result is the same using either function.

Problem 2

The goal of this problem is to demonstrate the energy compaction properties of the DCT. You are asked to do the following:

1. Load the *Cameraman* image that is provided by Matlab's Image Processing toolbox using the command: `f=imread('cameraman.tif');`. You can then display the image using a variety of functions including `imagesc(f)`; . You will also need to issue command `colormap(gray)`; .
2. Take the DCT $F(u, v)$ of the image using Matlab function `dct2.m`.
3. Set all coefficients $|F(u, v)| < 5$ to zero and reconstruct the image using `idct2.m`. Display and print out the reconstructed image. Calculate and print out the Peak Signal-to-Noise Ratio (PSNR). How many coefficients were set to zero?
4. Repeat the above step for $|F(u, v)| < 10$ and $|F(u, v)| < 20$.