University of California, Berkeley
College of Engineering
Department of Electrical Engineering and Computer Sciences

Prof. A. Zakhor Spring 2010

EE225b – Digital Image Processing
Lab Assignment #1 – Image Compression

Overview:

In this assignment, you explore some image compression techniques, and evaluate their performances by fidelity criteria.

Assignment specifics:

1. Objective Fidelity Criteria

(a) Write a program to compute the root-mean-square error [see Eq. (8.1-10)] and mean-square signal-to-noise ratio [per Eq. (8.1-11)] of a compressed-decompressed image. This project is generic in the sense that it will be used in other projects that follow.

(b) Download the image of Figure 1.1 from the course website and write a program to generate the results using uniform quantization and IGS quantization. Use your fidelity criteria program to characterize any loss of visual information and comment on your results.

2. Image Entropy

(a) Write a program to compute the entropy of an image.

(b) Download the images of Figures 1.2(a) and 1.2(b) and use your program to estimate their entropies.

3. Transform Coding

(a) Write a program to compute the information loss associated with the following transform coding schemes:

	Case 1	Case 2
Transforms	Fourier	Cosine
Subimage Size	8×8	8×8
Bit Allocation	8-largest coding	8-largest coding

Use the routines developed in 1 to quantify the loss of information. Download the image Figure 1.3 and use the program to compare Cases 1 and 2.

(b) Gradually decrease the number of retained coefficients until the reconstruction error for Case 2 becomes objectionable. That is, try 7-largest, 6-largest, … coding as the bit allocation method.

Please submit a written lab writeup in class on the due date.

Here are some helpful Matlab commands:

`X = fft2(x)`	Computes the 2D-DFT of the matrix `x`
`x = ifft2(X)`	Computes the inverse 2D-DFT of the matrix `X`
`X = dct2(x)`	Computes the 2D-DCT of the matrix `x`
`x = idct2(X)`	Computes the inverse 2D-DCT of the matrix `X`
`I = uint8(x)`	`I` is a matrix of integers ranging from 0..255
`imshow(I)`	Displays `I` as a grayscale image in the current figure
`I = imread('small.bmp', 'bmp')`	Reads the image file small.bmp and stores it in matrix `I`
`imwrite(I, 'result.bmp', 'bmp')`	Writes the matrix I to the image file result.bmp
`[B I] = sort(A, mode)`	Sorts the array `A` in the ascending or descending order