

Prof. A. Zakhor

Spring 2014

EE225b – Digital Image Processing  
Assignment #6 – 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	$8 \times 8$	$8 \times 8$
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:

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