

**Image Compression and Packet Video**  
**September 13, 1999**  
**Laboratory Assignment #5**  
**Dr. Nicholas Beser**  
**e-mail: Nicholas.Beser@jhuapl.edu**

Purpose:

Transform based compression has been standardized in the ISO JPEG document. There is very little leeway available to improve the algorithm, with the exception of the design of the quantization tables. Using the Matrix form of the DCT, and the blkproc routine from matlab, we will study the effect of changes in quantization tables.

Problem:

An image file will be read into Matlab using the imread function. Using the matrix form of the DCT (used by DCTDEMO) perform the following processing:

1. Generate the basis matrix for the DCT (using the function dctmtx2(8) (in the c:\matlab\toolbox\images directory))
2. Read in the Image
3. Calculate the DCT coefficients using the blkproc function (detailed notes are provided, but you can also look at demodct.m in the image toolbox directory (Matlab\toolbox\images\imdemos). Use the following equation:

$$F = BIB^T$$

Where **B** is the Basis image created by the dctmtx2 routine and **B<sup>T</sup>** is its transpose (shown as **B'** in Matlab.) The line of code will look like:

```
imageDCT = blkproc(A,[8 8], 'P1*x*P2',dctm,dctm.');
```

Note how the second Basis image is transposed in the P2 argument.

4. Perform Quantization by using the blkproc function to do the following equation:

$$\hat{F} = \text{round}\left(\frac{F}{Q}\right)$$

The line of code will look like:

```
Fprime=blkproc(imageDCT,[8 8], 'round(x./P1)',Q);
```

- 5 At this point we now have quantized transform coefficients. You should calculate the entropy of the quantized data (compare it to the entropy of the original pixel data). Note that ratio of compression will be computed roughly by comparing the 8 bits to the Entropy of the quantized data. It is rough because JPEG using several coding approaches that we are not using to improve the lossless compression (such as reordering coefficients, and specially designed Huffman tables).
- 6 Use the blkproc to renormalize the coefficients:

```
Fpnew=blkproc(Fprime,[8 8], 'x.*P1',Q);
```

- 7 The renormalized coefficients will be converted back into pixel space by an inverse DCT using the basis image:

$$\hat{I} = B^T F B$$

- 8 Use the blkproc again to compute the new image:  
`newImage=blkproc(Fpnew,[8 8],'P1*x*P2',dctm.',dctm);`  
You should also find the max and min of the newImage data (since we have quantized the coefficients, you should not expect the reconstructed pixels to be scaled from 0 to 255. I would suggest that you record the max and min, and then clip the result (if greater than 255, clip to 255, and if less than 0, clip to 0).

- 9 Use the following Array definition for the Quantizer:

```
Q=[8 36 36 36 39 45 52 65;...  
36 36 36 37 41 47 56 68;...  
36 36 38 42 47 54 64 78;...  
36 37 42 50 59 69 81 98;...  
39 41 47 54 73 89 108 130;...  
45 47 54 69 89 115 144 178;...  
53 56 64 81 108 144 190 243;...  
65 68 78 98 130 178 243 255];
```

The quantizer table was taken from the NITF JPEG spec (quantizer Q2).

- 10 Save the scaled image in a TIFF file.
- 11 Use LAB1 to measure the MSE of the reconstructed image.

Extra Credit:

Noting that the Quantizer table is designed to scale coefficients that roughly map to frequency response, change the Q table to give better high frequency response (by scaling the higher column and row Q values with a lower number. What has happened to the ratio of compression and the MSE.