CompE565, Spring 2014

Project 2

Prepared by
Alex Egg
809236396
E-mail: eggie5@gmail.com
Electrical and Computer Engineering
San Diego State University

Table of Contents

List of figures

Introduction

There is a great need for image/video coding to transmit such in a way that can be done in a timely manner. Also, it is necessary to code them to keep a reasonable space constraint, or minimize the space they do take up.

In this lab we take the image fundamentals we learned in project 1 and apply it to image compression. In this lab we implemented JPEG image compression in matlab. JPEG image compression consists of subsampling, DCT transform and quantization and encoding. We will also implement the reverse process: decoding, in order to reconstruct the original image.

Procedural Section

JPEG compression hinges on the characteristics of the DCT transform. The DCT transform maps pixel domain to frequency domain. The DCT transform has a property where it will organize high frequency components on the top-left of the transform and low frequency components on the bottom right of the transform. Coefficient magnitude represents the energy and its location represents the horizontal and vertical frequencies. The human eye responds more to high changes in frequency and because of this property the DCT suits us well for the next step of compression: quantization.

Quantization is a simple mathematical operation performed on each DCT block. When the quantization operation is performed, precision is lost and thus achieving compression. Or, in other words: When the number of discrete symbols in a given stream is reduced, the stream becomes more compressible. For example, reducing the number of colors required to represent a digital image makes it possible to reduce its file size.

After quantization, many coefficients turn to 0, which will provide a large degree of compression w/ the final Arithmetic coding step. In order to proceed to the Arithmetic coding step we have to process the data first; zig-zag scanning will help us with this step.. Many AC coefficients become zero after quantization. Many AC coefficients become zero after quantization. Zig Zag scanning arranges the coefficients in the increasing order of frequency.

With this understanding we preceding our experiment thusly:

We started this project by loading the .jpg image into Matlab and displaying it, this was done with the help of imread() and imshow(). Next we displayed the DCT coefficient matrix as well as image of the DCT transformed image blocks of the first two blocks in the 4th row, also finding the luminance component. We then quantize the DCT image by using the JPEG luminance and chrominance quantization matrix from the lecture notes. This reconstructs the image by computing the inverse DCT coefficients and sub sampling through interpolation. We compute the inverse quantized images from the previous steps, bringing the image back into RGB format so we can compare the newly uncompressed image to the original. These images cannot be deciphered by looking at them from an untrained eye. The PSNR is finally calculated and results are interpreted.

Results

As an end result we were left with five different figures representing different aspects of the project. The first figure simply showed we could read and output an image through the Matlab environment, which was fairly easy. Figure two is the result of taking just the DCT blocks from row 4 block one and two. Next we convert back into RGB format, figure three shows that the reconstructed images and the original nearly identical to the human eye. The PSNR between the original and reconstructed images are then taken and the findings show that for the luminance component of the decoded image is just over 40.

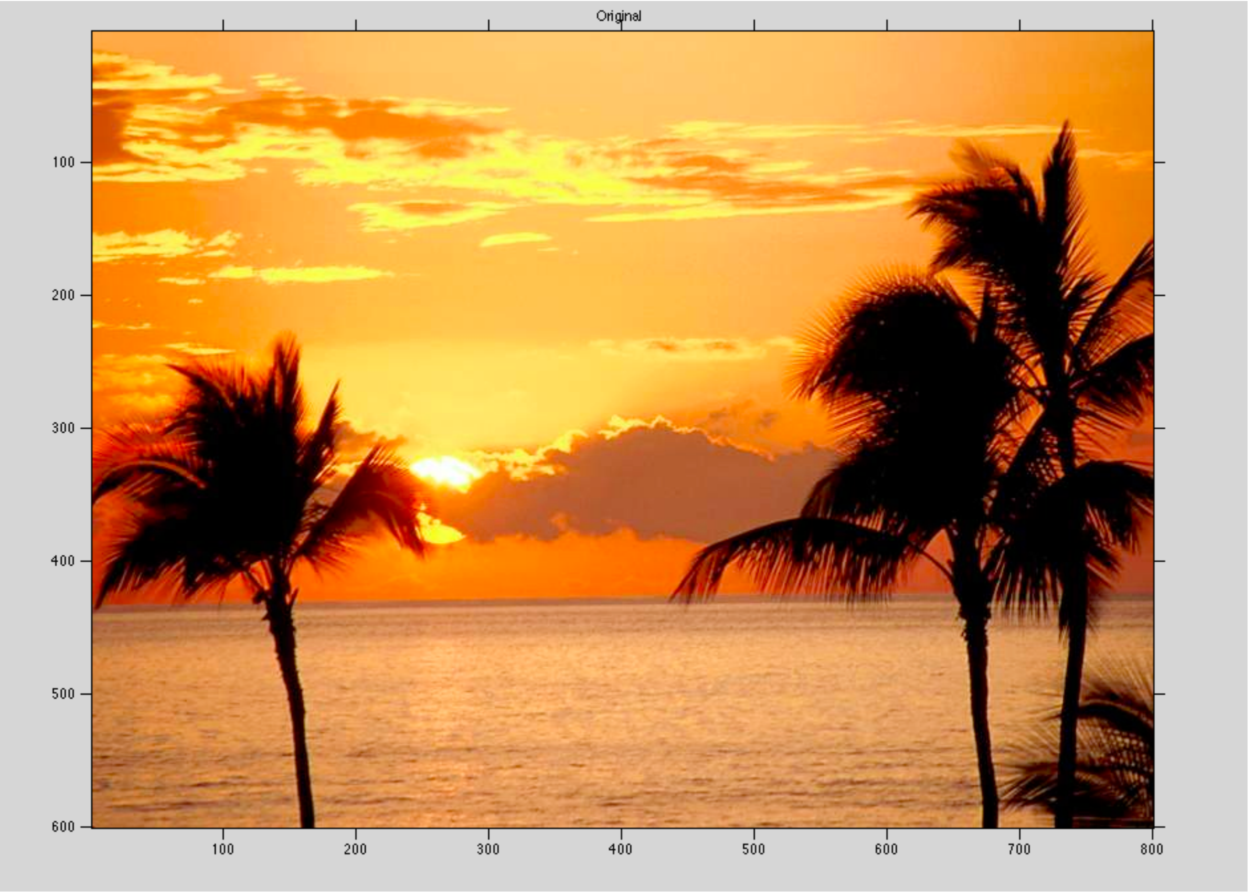

Figure 1: Original Image

Compute the 8x8 block DCT transform coefficients of the luminance and chrominance components of the image.

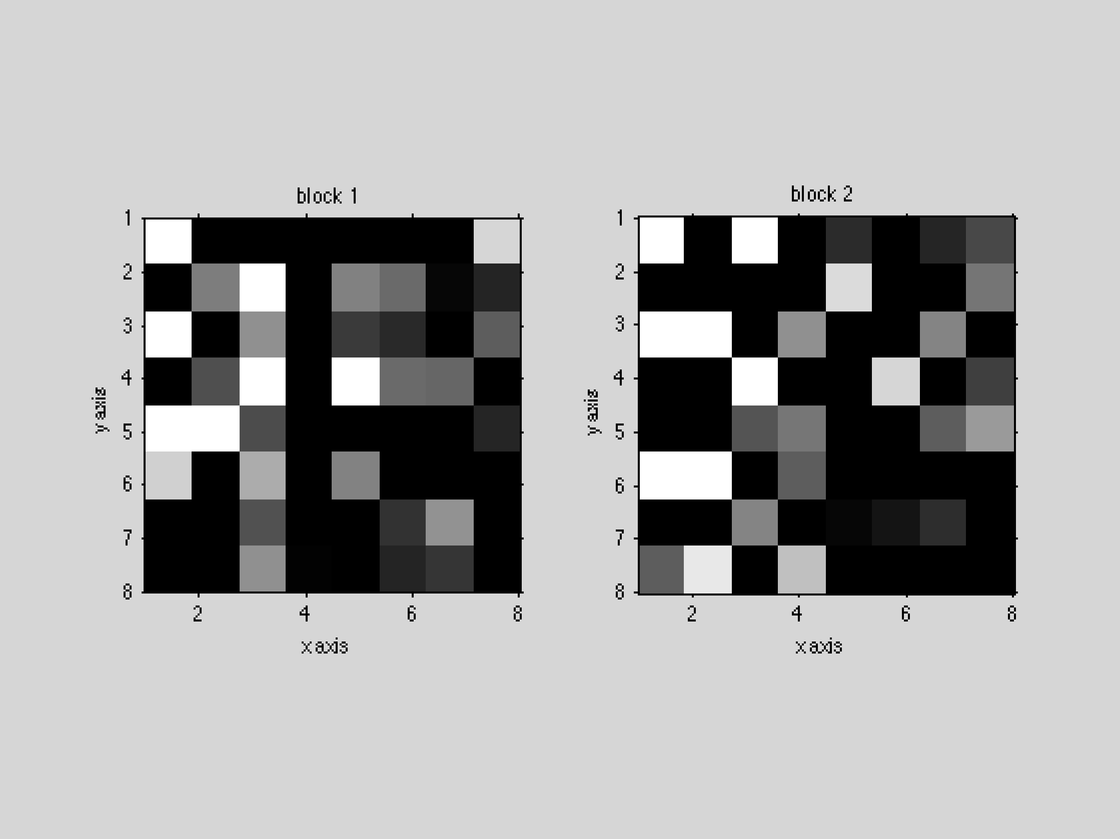

Figure 2: DCT transform block of blocks (1,4), (2,4)

```
block1 =

   1.0e+03 *

    1.0162   -0.0689   -0.0185   -0.0100   -0.0005   -0.0006   -0.0002    0.0008
   -0.1097    0.0004    0.0082   -0.0002    0.0004    0.0003    0.0000    0.0001
    0.1023   -0.0068    0.0005   -0.0141    0.0002    0.0001   -0.0002    0.0003
   -0.0761    0.0002    0.0127   -0.0006    0.0010    0.0003    0.0003   -0.0000
    0.0105    0.0130    0.0002   -0.0002   -0.0003   -0.0005   -0.0001    0.0001
    0.0008   -0.0003    0.0006   -0.0000    0.0004   -0.0002   -0.0010   -0.0001
   -0.0007   -0.0003    0.0003   -0.0003   -0.0009    0.0002    0.0005   -0.0002
   -0.0001   -0.0002    0.0005    0.0000   -0.0005    0.0001    0.0002   -0.0006


block2 =

   1.0e+03 *

    1.2119   -0.0547    0.0118   -0.0096    0.0001   -0.0006    0.0001    0.0002
   -0.0706   -0.0154   -0.0083   -0.0005    0.0008   -0.0001   -0.0000    0.0004
    0.0821    0.0154   -0.0004    0.0005   -0.0004   -0.0001    0.0004   -0.0003
   -0.0735   -0.0100    0.0121   -0.0002   -0.0001    0.0008   -0.0000    0.0002
   -0.0019   -0.0007    0.0003    0.0004   -0.0001   -0.0002    0.0003    0.0005
    0.0022    0.0013   -0.0013    0.0003   -0.0007   -0.0003   -0.0000   -0.0005
   -0.0014   -0.0004    0.0004   -0.0002    0.0000    0.0001    0.0001   -0.0006
    0.0003    0.0009   -0.0007    0.0007   -0.0000   -0.0000   -0.0002   -0.0007
```

Quantize the DCT image by using the JPEG luminance and chrominance quantizer matrix from the lecture notes.

block 1 DC DCT Coeff: 1016.250000

block 1 zig zag:

```
qb1zig =

  Columns 1 through 16
    64    -6    -9     7     0    -2    -1     1    -1    -5     1     0     0     0     0     0
  Columns 17 through 32
     0    -1     1     1     0     0     0     0     0     0     0     0     0     0     0     0
  Columns 33 through 48
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
  Columns 49 through 64
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
```

block 2 DC DCT Coeff: 1211.875000

block 2 zig zag:

```
qb2zig =

  Columns 1 through 16
    76    -5    -6     6    -1     1    -1    -1     1    -5     0    -1     0     0     0     0
  Columns 17 through 32
     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0
  Columns 33 through 48
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
  Columns 49 through 64
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
```

Reconstructed Image

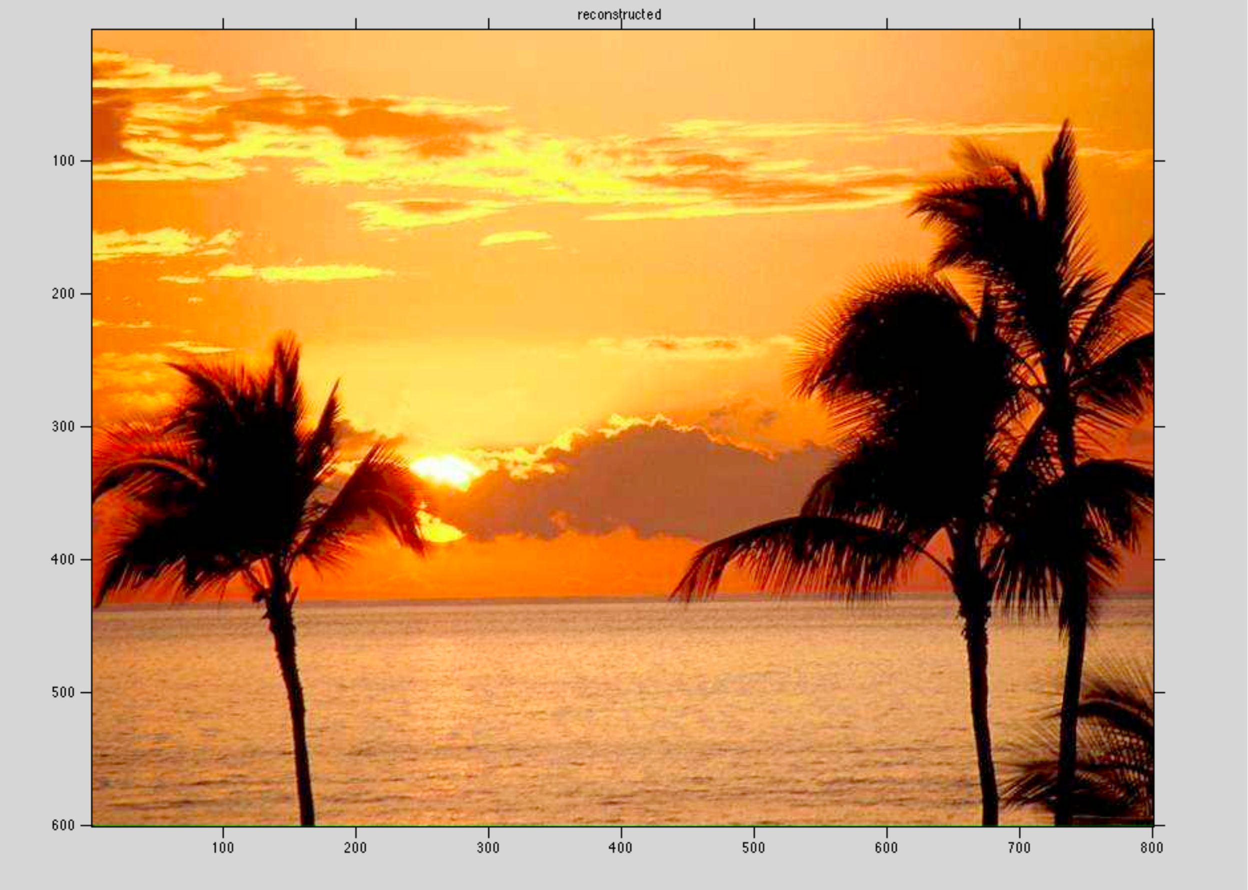

Figure 3: Reconstructed Image

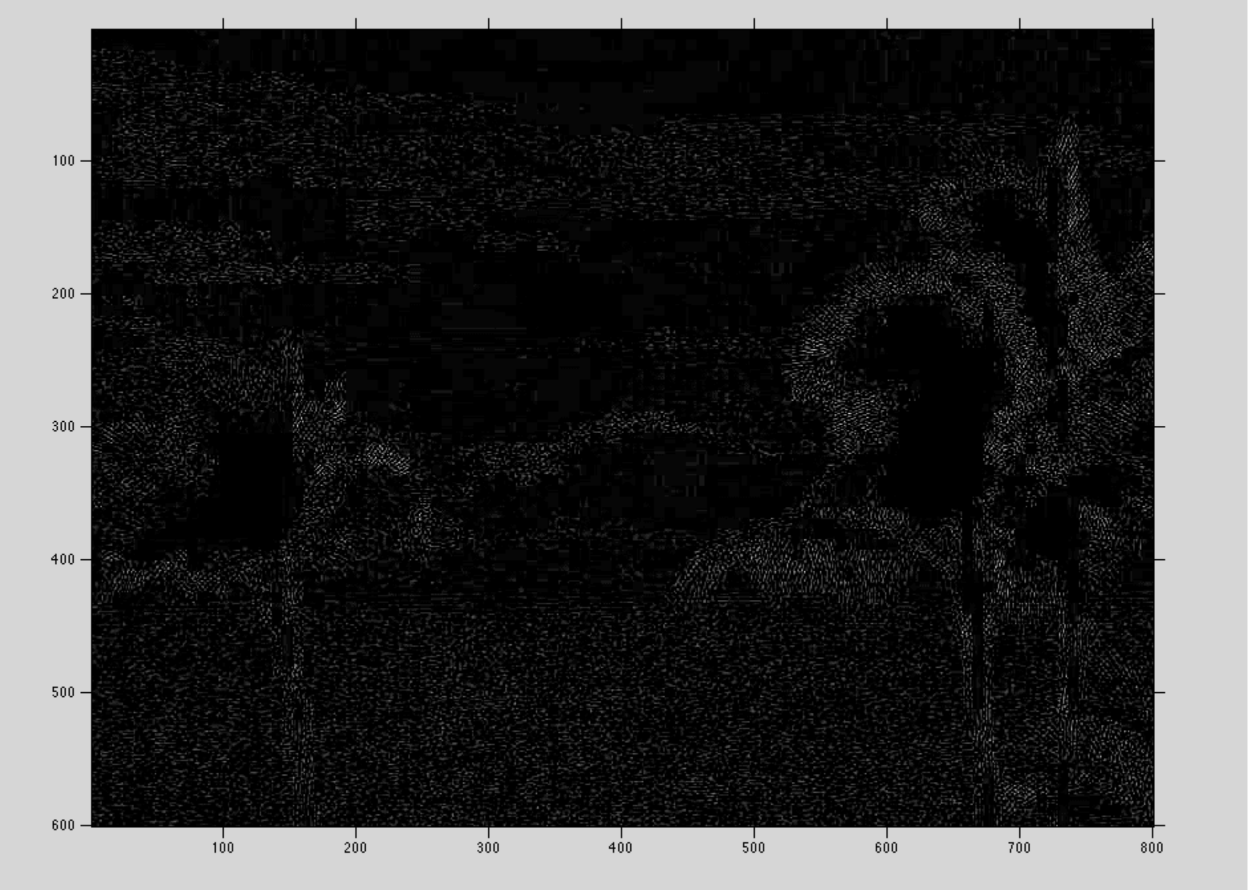

Figure 4: Error

PSNR:

```
PSNR_Y =

   42.7428
```

Conclusion

Ultimately, throughout this project we have succeeded in learning to read and display Digital Images in Matlab, find the DCT/ zigzag DCT coefficient matrices, and encode/ decode an image through quantization. I believe the main goal of this project was to teach us the basics of image processing, while gearing us toward more compression programming. This project reflects the core values of encoding and decoding images and I believe we will soon build from this foundation. With this foundation we should be able to start the study of video encoding.