CMPT 365 Assignment 2 Report

Our JPEG encoder and decoder has been implemented in MatLab. Matlab facilitates easier matrix manipulation, which fits our needs of using many matrix operations during the entire JPEG encoding and decoding process.

Note: We have not complied an executable for our program. Please run the program through the MatLab interface.

Encoder:

After loading in the original image via the MatLab imread() function, we can extract the RGB color information simply by extracting the information stored in the matrices where the image information was loaded into. R, G and B components are used to find corresponding Y, U and V channels. The separate Y, U and V components are then concatenated to form a YUV image. Following figure shows the original RGB image and converted YUV image.

Original RGB Image

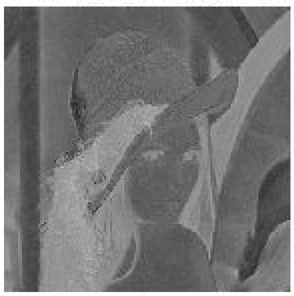
YUV Image

Color channels U and V are downsampled using 4:2:0 chroma subsampling. 4:2:0 subsampling reduces the number of pixels in each chrominance channel to ¼ of the original amount via removing every other pixel in both the X and Y dimensions. The following figure shows subsampled U and V channel image along with original U and V channel images for comparison.

Original U component

Original V component

U component after 4:2:0 Subsampling



V component after 4:2:0 subsampling

After chroma subsampling, all channels have their dimensions trimmed to a multiple of 8, then divided into collections of 8x8 blocks. Trimming is done by deleting rows and columns so that they are divisible by 8. We then apply the Discrete Cosine Transform on each block, via the matrix multiplying implementation. DCT converts spatial signals into numeric value so the different channels can be stored quantitatively.

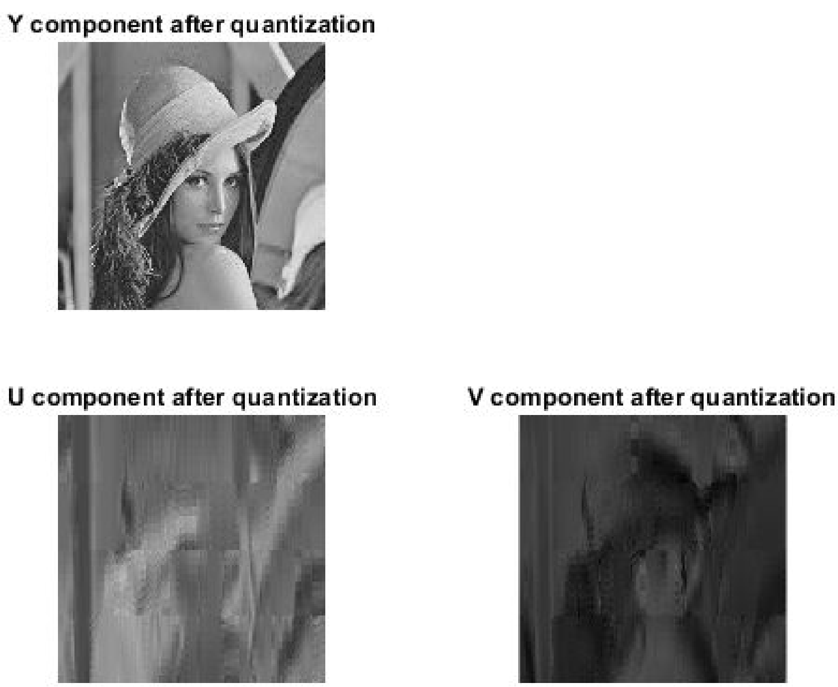
Quantization is the next step. Quantization is applied to the DCT matrices. It is a lossy compression techniques which reduces the DCT values to quantum values. Due to this reduction, data becomes highly compressed which makes it a lossy compression. The Y channel coefficients are quantized using a Luminance quantization table, while the U and V channel coefficients use the Chrominance quantization table. Quantization table entries tend to get progressively larger towards the bottom right of the table, which, when quantized coefficients are rounded, will eliminate higher order AC signals that tend to have very little significance in the appearance of the image.

After quantization completes, we are left with the quantized DCT coefficients. The JPEG encoding procedure is usually followed up by entropy coding, but we have opted to ignore this part due to the scope of the assignment specifications.

Decoder:

The decoder reverses the entire encoding process to give the original image back. Quantization is reversed by multiplying the Quantized DCT coefficients with the quantization matrix data, giving back the approximate values of the DCT coefficients. Due to the fact that many higher-order AC signal values were rounded down and eliminated, their details were lost upon reversing the quantization.

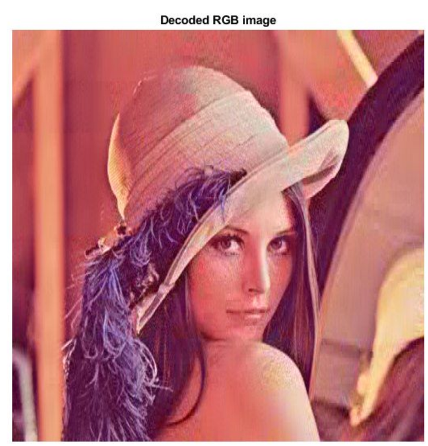
The DCT coefficients then has the Inverse Discrete Cosine Transformation applied to return the DC and AC signals back to YUV form. At this point, rendering separate channels as images provides an insight to the degree of details lost to quantization.



As observed in the image above, although the U and V channels preserve their original value trends, many details were lost to the quantization process. In addition, clear divisions are visible in the channels that were not present before, as a result of separating values into 8x8 blocks to use in DCT. The Y channel appears less affected, due to being quantized with a different table, as well as its very different set of values from U and V before DCT was applied.

After IDCT transforms the signals back into YUV channels, we need to reverse the chroma subsampling to return the U and V channels to the original dimensions. To do this, we opted to use a so-called “average of points” approach. Since each dimension had half their values removed, it stands to reason that each remaining value affects two consecutive pixels in that dimension. Therefore, our approach is one such that for each pixel, take the average of up to 4 values taken from its potential neighboring pixels. For example, if chroma subsampling removed pixel (2,2)’s chrominance information, we can calculate the average color of pixel (1,1), (1,3), (3,1), and (3,3) to get an approximate interpolation of its chrominance.

Once chroma subsampling is reversed, we can finally convert these YUV values back into RGB channel values. With some trivial matrix multiplication and addition/subtractions, we arrive at the decoded image (shown bottom-right).



Observe that although some observable loss remains, the overall quality of the image is not as compromised as when viewing the luminance and chrominance channels separately. This serves to demonstrate the observations regarding human perception that JPEG encoding relies upon.