

18.335 FINAL PROJECT REPORT

JUAN M ORTIZ

1. Introduction. Along with the numerous benefits that we have come to enjoy in our transition to the Digital Age have come a plethora of technical challenges. In particular, the rise in the prominence of the role that data plays in our services has forced us to invent efficient solutions to challenges related to storing, retrieving, processing, and transmitting various types of data. One of the most common types of data that we utilize in both computer and mobile environments is that of images. As such, there has been a lot of research in the field of image compression. Although the problem of compressing images falls within the general problem of compressing arbitrary raw binary data, we can greatly improve our performance by designing encoders that take advantage of the statistical properties that images possess. Through the use of our domain-specific knowledge about how images are composed as well as how we perceive images, modern compression algorithms are able to remove a great deal of redundancies so as to create much smaller representations that are able to be decoded into good reconstructions of the originals in accordance to human visual perception.

1.1. Compression Principles. Today, the majority of methods utilized in image compression are based on the principles of **entropy coding** and **transform coding**.

The first, entropy coding, refers to a type of lossless compression scheme which is not dependent on the medium to which it is applied. Entropy coding aims to reduce the size of data processed by defining a mapping between the original symbols/patterns of the data to one in which frequently occurring symbols/patterns are represented using few bits and those that occur rarely are represented with more bits. The two most common algorithms used for entropy coding are Huffman coding and Arithmetic coding. A more detail overview of Huffman coding will be given in **Section 2**. On the other hand, transform coding refers to a technique utilized when dealing with natural data sources such as images and audio. The main idea behind transform coding is to utilize a linear process to convert the original representation of the data to one in which the most relevant information is contained within a few coefficients. On its own, a naive transform coding approach is lossless and does not significantly reduce the information needed to encode the original data. However, the new representation along with domain knowledge about the data being encoded enables us to keep only the most relevant information to recreate something that is mostly faithful with the original but is much smaller. The most widely used transform coding algorithm when dealing with image compression is that of the discrete cosine transform (DCT) which will be further described in **Section 2**. Additionally, we will describe a method based on the singular value decomposition (SVD) and compare it against the commonly used DCT.

1.2. Modules of Image Coder. The typical configuration of an image encoder system is composed of the following modules: source encoder, quantizer, and entropy coder [5]. A brief description of each of these modules is provided below:

- (a) **Source Encoder.** The purpose of this module in the encoder system is to decorrelate the original input and transform it into a set of sparse data values through the use of a transform coding method. In doing so, the new data representation compacts the information of the original data into a small

- number of coefficients.
- (b) **Quantizer.** The goal of the quantizer module is to reduce the amount of information needed to store the representation created by the source encoder through decreasing the precision of the coefficients of the representation. During this step, we can take advantage of domain-knowledge to discard information that will not significantly affect the reconstruction of the input data.
- (c) **Entropy Coder.** The last step in a typical compression system is the entropy coder module. This step in the compression is lossless and aims to remove redundancies in the bit-patterns produced by the quantized output of the quantizer module.

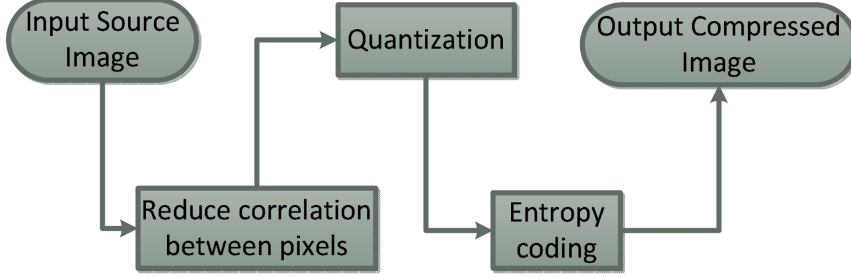


FIG. 1. *Typical Compression System Architecture from [5]*

1.3. Evaluation Metrics. In order to meaningfully compare different compression systems, we need to have metrics to objectively compare their results. In image compression, two important criteria that these metrics need to capture are the level of compression and the quality of the reconstruction. The most common metrics for both of these criteria are listed and defined below:

- (a) **Compression.** The most common measure for the degree by which an image was compressed is perhaps the most obvious of just utilizing the compression ratio CR defined as follows[8]:

$$CR = \frac{\text{Size}_{\text{compressed}}}{\text{Size}_{\text{original}}}$$

- (b) **Quality of Reconstruction.** A commonly accepted method of measuring the quality of image reconstruction is that of peak signal-to-noise ratio (PSNR). To define the PSNR between the original image I and compressed representations C using the common error metric of the mean square error (MSE) as follows:

$$\text{PSNR}(I, C) = 10 \log_{10} \left(\frac{\max_{i,j} (I_{ij})^2}{\text{MSE}(I, C)} \right)$$

where MSE is defined as

$$\text{MSE}(I, C) = \frac{1}{m \cdot n} \sum_{i=1}^m \sum_{j=1}^n (I_{ij} - C_{ij})^2$$

2. Background. The following section will provide a brief exposition to the theory behind the methods utilized in the experiment section.

2.1. Color Specification. One of the most commonly utilized tricks to improve the performance of image compression systems is to store images using YUV color coordinates rather than RGB coordinates. Rather than storing the red, green, and blue components of the different pixels in an image, descriptions of pixels in YUV coordinates are made out of a luminance (Y) and chrominance (U, V) components. The reason this scheme is useful is because the luminance plays a much more crucial role than color in the way the human-eye perceives light. Separating the luminance and chrominance components therefore allows us to discriminate how much information we throw away from each of these components based on how much they affect the way we perceive them. [4]

We can relate YUV and RGB color coordinates according to the following formula:

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.334 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix}$$

2.2. Transform Coding. At a fundamental level, a transform coding of an m by n matrix A can be thought as a factorization of the form:

$$A = U_l C U_r^\top$$

where U_l and U_r are unitary matrices and C is an n by n matrix coefficient whose form depend on the transform. As such, a transform is reversible and the coefficient matrix can be computed as:

$$C = U_l^\top A U_r$$

Two prominent examples of transforms are the discrete cosine transform (DCT) and singular value decomposition (SVD) and are briefly described below. [2]

2.3. Singular Value Decomposition. One of the most famous factorizations that follow the above form is that of the singular value decomposition. In SVD, a matrix A with rank r is decomposed into two eigenvector matrices ($U_l = U$ and $U_r = V$) and a non-negative diagonal matrix $C = \Sigma$. A possible approach to compute these matrices is to calculate the eigenvalues λ_i of the matrix $A^\top A$ and then solving the system of equations defined by finding the null space of all matrices in the form

$$A^\top A - \lambda_i \text{ s.t. } 0 \leq i \leq r$$

In practice, however, iterative methods such as the Householder Method or the QR iteration are used.[9]

2.4. Discrete Cosine Transform. One of the most widely used transform coding methods by far is that of the discrete cosine transform (DCT). Most notably, it has been part of the compression algorithm of choice for the popular image format JPEG. Some of the properties that make DCT a great transform choice for image compression include: (1) It has a fixed set of basis functions, so there is no need to transmit the basis used. (2) It has the ability to reduce the compression artifact effect (i.e. visible block boundaries). (3) It is able to compress energy into low frequencies in image data. [6] [1]

More formally, we can use the framework of **2.2** to define DCT as a factorization of the form

$$A = U_L C U_R^\top$$

where the orthogonal matrices U_L and U_R are defined as

$$U_{Lij} = \begin{cases} \sqrt{\frac{1}{m}} & i = 1 \\ \sqrt{\frac{2}{m}} \cos(\frac{\pi(2j-1)(i-1)}{2m}) & 2 \leq i \leq m \end{cases}$$

$$U_{Rij} = \begin{cases} \sqrt{\frac{1}{n}} & i = 1 \\ \sqrt{\frac{2}{n}} \cos(\frac{\pi(2j-1)(i-1)}{2n}) & 2 \leq i \leq n \end{cases}$$

The fact that U_L and U_R are independent of the matrix to be encoded A and can thus be inferred without explicitly being included in the encoding of A give DCT a huge advantage in terms of the the compression ratio it can achieve to transforms like SVD where the normal basis that makes up U_L and U_R are dependent on A and need to be included for proper decoding.

2.5. Quantization. After the transform has been applied, we can utilize our knowledge about the mechanism of the transform as well as our perception to selectively discard information that we believe to be unimportant to reasonably reconstructing the original input.

In SVD, this can easily be done by selecting the rank $r' \leq \text{rank}(A)$ by which to approximate the original matrix. After choosing r' we can throw away any columns of $U_L = U$ and $U_R = V$ that have an index greater than r' and prune $C = \Sigma$ to the square matrix $\Sigma[1 : r', 1 : r']$. With respect to the Frobenius norm, this resultant matrix is the closest matrix with the restriction $\text{rank} \leq r'$ to the original matrix A . [9]

For DCT, quantization aims to reducing the coefficients of the least important (high) frequencies to zero. This approximation takes advantage of the fact that the high frequencies components of images are often inconsequential to the way that we perceive them. To perform these quantization, we have defined fixed standard matrices Q that determine how relevant each frequency coefficient is. The quantized coefficients, K , are then calculated by a scaling followed by rounding as follows:

$$K_{ij} = \text{round}(\frac{C_{in}}{Q_{ij}})$$

The JPEG standard provides the following two tables for the quantization of luminance and chrominance respectively: [7]

$$\begin{aligned}
146 \quad Q_Y &= \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix} \\
147 \quad Q_C &= \begin{pmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{pmatrix} \\
148
\end{aligned}$$

149 Once the data has been quantized, it is necessary to define an ordering to the
150 data so as to define how it is to be stored. In the case of transforms like DCT where
151 only the coefficients need to be packed, it is customary to encode the coefficients using
152 a zig-zag pattern like the one shown in figure 2.

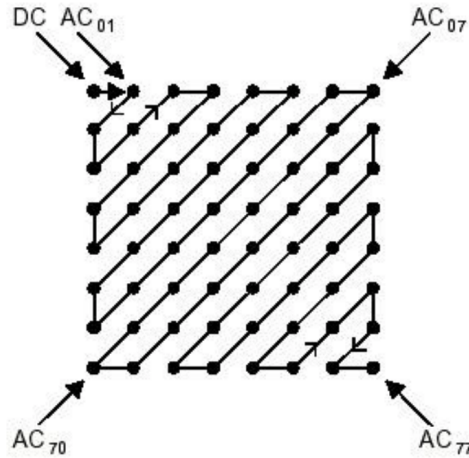


FIG. 2. Diagram of Zig-zag sequencing from [5]

153 This ordering takes advantage that low-frequencies normally have non-zero coeffi-
154 cients and high-frequencies tend to have zero-coefficients. Packing the zero-coefficients
155 together allows for better compression using an entropy-coding algorithm such as Huff-
156 man Coding.

157 **2.6. Block Algorithms.** A common technique when designing image and video
158 compression systems is that of using a block-based approach to coding. Rather than
159 applying a transform directly on the entire image, the image is first divided into
160 smaller blocks or sub-matrices to which the transform is then applied. This approach

facilitates taking advantage of spacial redundancies often encountered images [1]. For DCT, it is customary to utilize square blocks of size 8.

2.7. Huffman Coding. As depicted in figure 1, after the quantization step, further lossless compression is obtained using a form of entropy coding. One of the widely used algorithms used for this purpose was suggested by David Huffman in the paper “A Method for the Construction of Minimum-Redundancy Codes” [3].

The algorithm can efficiently create encodings for the symbols utilized in the original data given a list of frequencies for each symbol. If the frequencies of the symbols used are ordered, it is possible to design an appropriate encoding in linear time as follows:

Algorithm 2.1 Huffman Encoding

- 1: For each symbol $s_i \in S$, create a leaf vertex v_i
 - 2: Create two priority queues Q_1, Q_2
 - 3: Enqueue all v_i in the first queue according to their symbol frequency $f(v_i)$ in increasing order
 - 4: **while** $|Q_1| + |Q_2| > 1$ **do**
 - 5: Dequeue the lowest two vertices of the queues by examining the fronts of both queues
 - 6: Create a new inner vertex that has the two dequeued as child nodes and arbitrarily label one as the ‘0’-child and the other as the ‘1’-child. The frequency of this new node will be defined as the sum of the frequencies of its two children.
 - 7: Enqueue the new node into the rear of Q_2
 - 8: **end while**
 - 9: **return** The single remaining root-vertex.
-

The encoding for a given symbol is then given by the path taken from the root node to the leaf vertex that contains the symbol in question. This scheme produces a code with an optimal weighted average code length as defined by the frequencies of the symbols. [3].

3. Experiments. The experiments carried out consisted of compressing the photo shown in figure 3 using a block size of 8 with both SVD and DCT. For each trial, the compression ratio to the original image was recorded and the quality of reconstruction was measured using PSNR.

3.1. Image Quality. To control the compression ratio obtained with DCT, we varied the multiplication coefficient by which the quantization table given in **section 2.5** was multiplied prior to quantization. The multiplication coefficient was varied from 0.01 to 1.0 with a resolution of 0.01.

In the other hand, for SVD, controlling the compression was done via placing a restriction on the fraction of the total value of the singular components that it used in its factorization. For example, if the limit was set to f , then it must be true that

$$f \leq \frac{\sum_{i=1}^{r'} \sigma_i}{\sum_{i=1}^r \sigma_i}$$

where r is the original rank and r' is the rank of the factorization used.

For the experiments involving SVD, the utilized fraction was varied from 0.75 to 1.0 with a resolution of 0.01.



FIG. 3. *Benchmark Image*

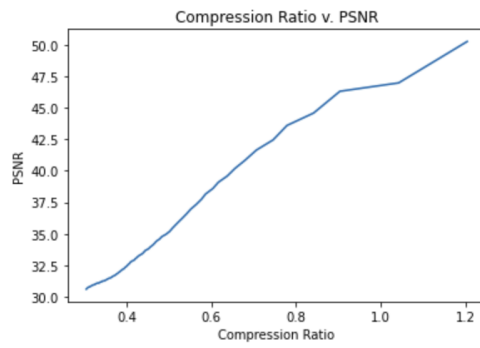


FIG. 4. *Quality of DCT Reconstruction v. Compression Ratio*

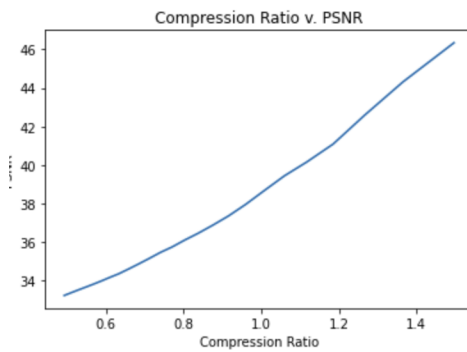


FIG. 5. *Quality of SVD Reconstruction v. Compression Ratio*

As shown by figures 4 and 5, it seems to be the case that, for comparable levels of compression, the reconstruction quality of DCT is superior to that of using SVD.

188 This is consistent with the discussion of **section 2.4** because SVD needs to encode
189 the basis for its transform while the DCT does not.

190 **4. Conclusion.** With the amount of information that we constantly encounter in
191 today's Digital Age image compression has become a fundamental part of the systems
192 we use. In this paper, we went over the fundamental components that make up a lossy
193 image compression system such as the one used in the JPEG standard. Additionally,
194 we numerically compared the compression achieved by utilizing similar compression
195 systems; one based on SVD and the other on DCT and empirically observed that
196 DCT is able to achieve better quality reconstructions for a given compression ratio.

- [1] N. AHMED, T. NATARAJAN, AND K. R. RAO, *Discrete cosine transform*, IEEE transactions on Computers, 100 (1974), pp. 90–93.
- [2] A. DAPENA AND S. AHALT, *A hybrid dct-svd image-coding algorithm*, IEEE transactions on Circuits and Systems for video technology, 12 (2002), pp. 114–121.
- [3] D. A. HUFFMAN, *A method for the construction of minimum-redundancy codes*, Proceedings of the IRE, 40 (1952), pp. 1098–1101.
- [4] M. PODPORA, G. P. KORBAS, AND A. KAWALA-JANIK, *Yuv vs rgb-choosing a color space for human-machine interaction.*, FedCSIS Position Papers, 18 (2014), pp. 29–34.
- [5] A. RAID, W. KHEDR, M. A. EL-DOSUKY, AND W. AHMED, *Jpeg image compression using discrete cosine transform-a survey*, arXiv preprint arXiv:1405.6147, (2014).
- [6] R. REININGER AND J. GIBSON, *Distributions of the two-dimensional dct coefficients for images*, IEEE Transactions on Communications, 31 (1983), pp. 835–839.
- [7] S. J. SANGWINE AND R. E. HORNE, *The colour image processing handbook*, Springer Science & Business Media, 2012.
- [8] K. SAYOOD, *Introduction to data compression*, Morgan Kaufmann, 2017.
- [9] L. N. TREFETHEN AND D. BAU III, *Numerical linear algebra*, vol. 50, Siam, 1997.