

JPEG Compression

Dylan Lathrum

November 1, 2019

1 Introduction

[Some fancy opening about how data is integral to the information age] As with everything else, data costs money to store and transfer. Whether one is looking to store a file on their computer or send an image through the internet, it is often in everyone's best interest to use as little storage space and bandwidth as possible while still maintaining quality. This is achieved through compression, a method of reducing the footprint of data by reducing the number of bits needed to represent the data. Compression methods can be divided into two main categories: lossless and lossy. Lossless compression is a method where the filesize of the data is reduced without any degradation of the file's contents; a quality essential to compressing text documents and computer code. Lossy compression is a more efficient method to reduce file sizes at greater ratios at the cost of losing some detail in the data; a tradeoff that is more acceptable for images and sounds.

Each compression method has their own use cases; for example, it would be a terrible idea to apply a lossy compression algorithm to an essay as some of the data would be lost or changed in the process, while it is perfectly acceptable to run the same algorithm on an image file where the user can afford to lose some detail. In either case, compression is a trade-off. While some storage space or bandwidth may be saved, something else must be lost. Whether that be detail in an image, or the processing power required to decompress a file, different algorithms have their strengths and weaknesses.

2 What is JPEG?

Specifically, Joint Photographic Experts Group (JPEG) is a lossy compression algorithm that is widely used for images on the Internet. The entire process that JPEG uses to compress images will be covered in detail, but one of JPEG's most predominant features is a variable compression ratio. Most compression methods run at predefined or dynamically generated ratios, making it impossible for a user to choose how compressed they want their data to be. Because of this, and the algorithm's efficiency and speed, JPEG has become a standard for digital images.

3 Preprocessing

Before an image can be processed using JPEG, a couple steps are executed to simplify the process. For this paper, the image shown in Figure 1 will be used at a resolution of 256x138px. The first step is to split the color channels of the image into three distinct grayscale images. In this example a grayscale picture is used, but in the case of a color image, the image would be split into YCbCr color space, comprised of Luminance (Y), Chroma Blue (Cb) and Chroma Red (Cr), or YCbCr channels. That way, all calculations are done against three grayscale versions of the image that are recombined at the end of the process. Once split, image is partitioned into 8x8 chunks as shown in Figure 2. This is done to break up the work into smaller chunks to be processed individually. 8x8

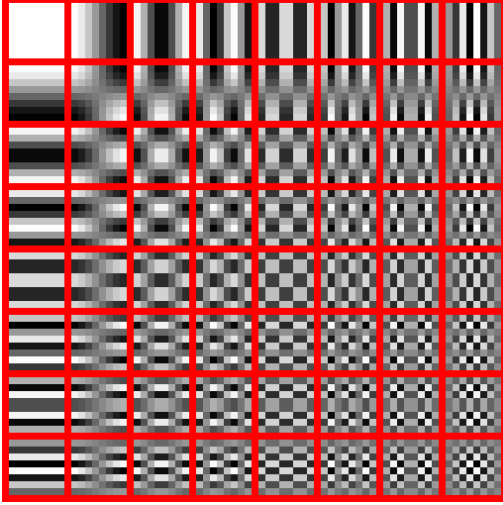


Figure 3: Discrete Cosine Transform

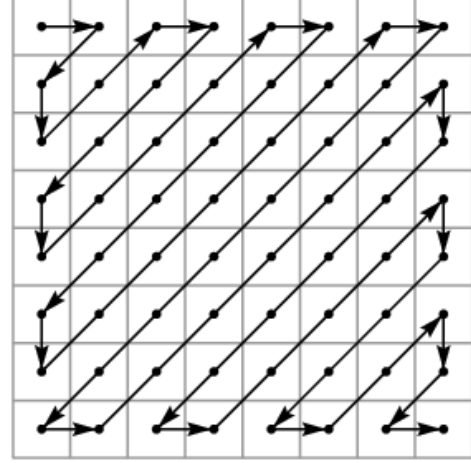


Figure 4: Processing order

because of this, half of the work is done. The challenging part is figuring out the coefficients that can be used to create the matrix. This is achieved using the following function:

$$X_{u,v} = \alpha(u)\alpha(v) \sum_{x=0}^7 \sum_{y=0}^7 g_{x,y} \cos \left[\frac{(2x+1)u\pi}{16} \right] \cos \left[\frac{(2y+1)v\pi}{16} \right] \quad (3)$$

Where,

u is the horizontal index from 0 to 7

v is the vertical index from 0 to 7

$\alpha(u) \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u = 0 \\ 1 & \text{otherwise} \end{cases}$ is a normalizing scale factor to make the result orthonormal

$g_{x,y}$ is the pixel intensity at the coordinates (x,y)

$X_{u,v}$ is the DCT coefficient at the coordinates (u,v)

5 Quantization

6 Compression

7 Decompression

8 Conclusion

9 Further Examples