JPEG

Hany Farid

The JPEG image format is the standard compression scheme for digital cameras. Compression schemes allow for the trade-off between image file size and image quality. A highly compressed image requires relatively little memory for storage and transmission, but may have noticeable distortion. A less compressed image will have greater fidelity to the original, but requires more memory and bandwidth. The ubiquitous JPEG standard was established in 1992, based on a compression scheme proposed in 1972, based on basic mathematics dating back to 1882. We will explore the history and impact of JPEG images on the digital and internet revolution.

The Sinusoid

Despite its prominence in the modern, digital world, the JPEG image standard has its roots in the 19^{th} century. In his seminal 1882 work on heat flow, French mathematician and physicist Jean-Baptiste Joseph Fourier (1768–1830) claimed any function can be expressed as a sum of multiple sinusoids [3]. Although this claim requires some additional assumptions, Fourier's insight was a breakthrough whose impact has rippled through astronomy, biology, computer science, mathematics, music, physics, and more.

The shape of the elegant sinusoid – $\sin(\omega)$ – can be described by spinning a line around a circle and measuring the vertical distance between the circle's center and the line's tip. The speed with which the line spins

around the circle defines the sinusoid's *frequency* – the number of oscillations per second; the length of the line defines the *amplitude* – the height of the oscillations; and the starting position of the line defines the sinusoid's *phase* – the relative position of the oscillations.

A high frequency sound (a squeak), for example, has many rapid oscillations in air pressure per second, while a low frequency sound (a rumble) has fewer, slower oscillations per second. Turn the radio volume up or down, and the sound's amplitude increases or decreases.

There is a visual analogue to a sound's sinusoidal representation. A high frequency visual pattern has many abrupt changes in appearance across space (a grass texture) while a low frequency visual pattern has only gradual changes across space (a cloud pattern). Analogous to volume, amplitude corresponds to the brightness of the visual pattern.

Digital Images

At the point of recording, a digital image is made up of an array of picture elements – pixels. Each pixel is itself composed of three numbers corresponding to the primary red, green, and blue colors (RGB). An uncompressed, modest-sized, 1000×1000 RGB image[1] requires approximately three megabytes of memory to store on a camera or computer[2]. These days, high-end digital cameras record images in the many-tens of megapixels, and even mobile devices record 12 megapixel images, which, left uncompressed would require 36 megabytes of memory per image.

In the early days of the digital and internet revolution, it was impossible to store and transmit a large number of even sub-megapixel digital images. Fourier's insights into the power of the sinusoid to represent signals and patterns laid the groundwork for an efficient way to

[1] A 1000×1000 pixel image consists of a total of $1,000,000$ pixels, referred to as a one megapixel image. The megapixel moniker refers to how many millions of pixels are in an image. A 2000×3000 image with six million pixels, for example, is a six megapixel image.

[2] (1000×1000 pixels) $\times$ (8 bits per color) $\times$ (3 colors per pixel) = ($24,000,000$ bits) = (2.8 megabytes)

digitally represent, store, and transmit audio, image, and video, in turn revolutionizing the power and reach of the internet.

JPEG Compression

The simplest way to compress an image is to throw away pixels. Starting with a 1000×1000 pixel image, for example, throwing away every other pixel results in a 500×500 pixel image with only $250,000$ pixels as compared to the original $1,000,000$ pixels, for a savings of $4\times$. This, however, is highly undesirable. Why should we build high-resolution cameras, capable of recording high-fidelity images, only to reduce the image resolution immediately after recording because we can't store or transmit the images?

We seek, therefore, to compress an image to reduce memory and transmission costs, while retaining resolution and visual quality.

The digital-camera revolution was kick-started in 1969 – the same year as the Apollo moon landing – when Willard Boyle and George Smith invented the charge-coupled device (CCD) for electronically recording and storing light. Starting in the late 1960s and early 1970s, researchers were already considering how to best compress digital images. The Karhunen-Loeve transform (KLT) emerged as the best way to compress digital data. This transform, however, was computationally costly, leading Nasir Ahmed in 1972 to develop the Discrete Cosine Transform (DCT) [1], itself inspired by Fourier's insights into the power of sinusoidal representations [3]. The DCT quickly emerged as an effective and efficient way to compress digital images and eventually was adopted by the Joint Photographic Experts Group who, in 1992, established the JPEG compression standard.

The JPEG compression standard was designed to take advantage of the human visual system's differential sensitivity to various forms of visual information. This compression scheme attempts to preserve

the image information to which we are most sensitive while discarding information we are unlikely to notice. For example, we are more sensitive to luminance contrast – a change from light to dark – than to color contrast – a change from red to green. Consequently, JPEG compression preserves more information about luminance than about color. JPEG compression also treats spatial frequencies differently. Humans are more sensitive to low spatial frequencies (grass texture) than to high spatial frequencies (cloud pattern), and, accordingly, JPEG compression preserves more information about low spatial frequencies than about high spatial frequencies.

While there are many details in the complete JPEG compression scheme, the heart of this compression relies on representing visual patterns using sinusoids (or more precisely, a phase-shifted version of the sinusoid, the cosine, $\cos(\omega)$) and removing content to which the human visual system is less sensitive. The heart of the DCT is this variation of the Fourier transform:

$$F(x,y) \quad = \quad \alpha_{x,y} \sum_{u=0}^{7} \sum_{v=0}^{7} f(u,v) \cos\left(\frac{(2u+1)x\pi}{16}\right) \cos\left(\frac{(2v+1)y\pi}{16}\right),$$

transforming the pixel-based image representation (f) to a frequency-based representation (F), allowing for differential compression to different frequencies.

Compression is achieved through quantization: dividing each DCT value by an integer value q and rounding the result to the nearest integer. For example, an initial DCT value of 4.2 and a divisor $q = 2$ will yield a quantized DCT value of $4.2/2 = 2.1 \rightarrow 2$. Increasingly more compression is achieved by dividing by increasingly larger values of q, because this drives more DCT values to 0, which themselves can be more efficiently represented in the JPEG file. Quantizing the DCT value 4.2 by $q = 10$, for example, yields $4.2/10 = 0.42 \rightarrow 0$. By quantizing the higher frequencies more than the lower frequencies, we take advantage of the

```
# input:  B: 8x8 image-pixel block
#         Q: 8x8 quantization (integer-valued)
# output: D: 8x8 block-DCT
def dct2(B,Q):
    D = np.zeros((8,8)) # initialize
    y,x = np.meshgrid(np.arange(1,9,1), np.arange(1,9,1))
    for i in range(1,9):
        for j in range(1,9):
            ai = np.sqrt(1/8) if i == 1 else np.sqrt(2/8)
            aj = np.sqrt(1/8) if j == 1 else np.sqrt(2/8)
            D[i-1,j-1] = ai * aj * np.sum( np.sum(B * np.cos(np.pi*(2*x-1)*(i-1)/16) *
                                                  np.cos(np.pi*(2*y-1)*(j-1)/16)) )
    return( np.array( D/Q, dtype=int) ) # quantize
```

Figure 1: A modern JPEG encoder is highly optimized to quickly compress and decompress images. This Python code snippet implements a non-optimized version of the basics of a JPEG encoder, consisting of the DCT transform and DCT quantization.

differential sensitivity of the human visual system and remove information to which we are less sensitive, minimizing perceptual distortions, and reducing the final image file size. Our differential sensitivity to luminance and color is achieved by decomposing the RGB image into one luminance and two color channels, and compressing the color channels more than the luminance channel.

Because the DCT representation of visual patterns is best for relatively small image patches, the compression is applied to each non-overlapping 8×8 pixel block. This is why, in a low-quality JPEG image, a visible grid-pattern emerges along these block boundaries.

Although the JPEG compression allows for fine control over the compression of each spatial frequency across each luminance/color channel, all standard photo-editing and coding libraries, synthesize these compression parameters into a single setting ranging from high-compression/low-quality to low-compression/high-quality. For example, I compressed an 8-megapixel image across an entire compression range yielding, at one end of the compression/quality spectrum, a 0.2MB file size to, at the other end, 5.3MB – the uncompressed image came in at a whopping 25.8MB.

JPEG Forensics

Almost as soon as we moved from film photography to digital photography, the issue of digital image manipulation was upon us. Beyond playing its part in the digital and internet revolution, JPEG compression has played a critical and unintentional role in the forensic analysis of digital images.

I still distinctly remember the call in 2008 from detectives in Scotland. Under a strong Scottish accent, there was a strong sense of urgency and seriousness from the detective spearheading the investigation. A dozen men stood accused of abusing young children and distributing images of the abuse. At the center of the complex, multiyear case was a series of images of unknown origin. The detective asked if I could link the photographs to one of a handful of cameras that had been seized in the investigation. I could.

This forensic analysis required a two-step process, the first of which leveraged distinct JPEG compression settings that vary across devices and software. Most notably, the luminance/color and spatial frequency specific quantization values vary as a result of different compression and visual distortion tolerances for low-, medium-, and high-end cameras. Because these tolerances are constantly being refined, even successive releases of the same camera may use different compression settings [2]. These variations allowed me to identify the make/model used to record the images in questions. A secondary ballistic analysis allowed me to uniquely identify the camera based on subtle imperfections in the underlying camera sensor [2].

Closing Thoughts

Audio (MP3) and video (MPEG) compression operate on the same concept of transforming the original data to a frequency-based representation and differentially compressing based on human sensitivity.

In the early days of the internet, digital media was highly compressed with relatively low quality. But, at least we could share audio, images, and video. As bandwidth, and computer and device memory increased, it became easier to store and transmit increasingly higher quality content. Without data compression, however, it would have been impossible to record, store, and share uncompressed content at the scale we do today: nearly 2 billion images per day, and, on YouTube alone, 500 hours of video every minute. The untold hero in this digital landscape is Fourier's brilliant mathematical insights, and the brilliant application of these insights into the JPEG compression scheme.

References

[1] Nasir Ahmed. How i came up with the discrete cosine transform. *Digital Signal Processing*, 1(1):4–5, 1991.

[2] Hany Farid. *Fake Photos*. MIT Press, 2019.

[3] Joseph Fourier. *The analytical theory of heat*. The University Press, 1878.