Compressing Images for the Internet

Giordano Beretta
Computer Peripherals Laboratory

HPL–97–163
December, 1997

World Wide Web, Internet, Image Encoding, Data Compression, JPEG, GIF, PNG, FlashPix

The World Wide Web has rapidly become the hot new mass communications medium. Content creators are using similar design and layout styles as in printed magazines, i.e., with many color images and graphics. The information is transmitted over plain telephone lines, where the speed/ price trade-off is much more severe than in the case of printed media. The standard design approach is to use palettized color and to limit as much as possible the number of colors used, so that the images can be encoded with a small number of bits per pixel using the Graphics Interchange Format (GIF) file format. The World Wide Web standards contemplate a second data encoding method (JPEG) that allows color fidelity but usually performs poorly on text, which is a critical element of information communicated on this medium. We analyze the spatial compression of color images and describe a methodology for using the JPEG method in a way that allows a compact representation while preserving full color fidelity.

To be published in and presented at *Electronic Imaging: Science and Technology—Color Imaging: Device Independent Color, Color Hardcopy, and Graphic Arts III*, San Jose, California, January 28–30, 1998. *SPIE Proceedings*, Volume 3300

1 Introduction

The World Wide Web is a hypertext system based on the Internet. It was conceived primarily to allow teams of physicists to relate sets of data and sequences of interpretations of this data, independently of the physicist's location. For improved readability a simple markup language called HyperText Markup Language (HTML) was created, which allowed the use of more readable fonts and the inclusion of graphs and images. However, for maximum platform independence, the main paradigm was that the author of a document defined the contents and the structure, while the appearance was defined by the reader. In essence, from its inception the World Wide Web has been a tool for distributed cognition in a community of practice.

In the meantime the World Wide Web has been embraced by the general public and savvy publishers have quickly recognized its potential as a new publications medium. The World Wide Web has become a hot and visually rich one-way delivery mass-medium for visually rich information. Commercial publication requires a strict control of space and appearance. Consequently a large proportion of document contents is transmitted as raster images rather than a combination of text and images. Color is used throughout to give a distinctive look to the pages for each publication. Gradients and textures are used profusely.

Until recently many on-line computer applications used the teletype (TTY) paradigm, where a screenful of transmitted information consists of 24×80 = 1,920 bytes. Current applications often use at least a bitmap of so-called VGA size, which contains 640×480 picture elements (pixels). A full color image requires 24 bits to represent the color of a pixel, thus at least 921,600 bytes are required for a screenful, almost 500 times as much.

Due to this 500-fold size increase, typical World Wide Web files require much more time to transmit than ASCII text files. Concomitantly, many readers access the Internet from slow connections in their homes, as opposed to the physicists of the early World Wide Web who have the some of the fastest direct Internet connections available. The standard design approach to reduce file size is to use palettized color and to limit as much as possible the number of colors used, so that the images can be encoded with a small number of bits per pixel using the Graphics Interchange Format (GIF) file format[1].

When the number of colors is restricted to those available in a palette, accurate color reproduction is no longer possible, because colors have either to be approximated or simulated from palette colors using dither patterns. Designers put a large effort to specify with great precision the color of each item, and color scientists invent algorithms that allow the perceptually accurate reproduction of these colors across media. Therefore, in professional applications such as product catalogs and brochures, color palettization is not acceptable. Dithering is unacceptable because it introduces a visible texture in the images, which are typically reproduced on a monitor at 72 dpi. The dither pattern can interfere with the texture on the object in the image, changing its appearance, or in the case of text, making it harder to read.

HTML allows the use of a second file format called informally JPEG for the Independent Joint Photographic Experts Group that first worked on its standardization[2]. In this encoding system full color information can be preserved for each pixel, but spatial information that cannot be perceived by the human visual system is omitted. Although JPEG is routinely used to encode color images, it is not very popular to encode colored graphics and text. The reason is that when the file is sufficiently compressed the quality of text and graphics is significantly degraded.

We show how the parameters for JPEG compression can be selected to achieve high compression rates while preserving the readability of text and smoothness of color areas. This allows graphic designers to exploit the full color gamut available on computers. The main idea is to design customized quantization and Huffman tables for each type of graphic element, such as image, gradient, texture, text, etc.

2 The GIF format

GIF was developed by CompuServe Incorporated and defines a protocol intended for the on-line transmission and interchange of raster graphic data in a way that is independent of the hardware used in their creation or display. Three channels are used to specify colors and these channels are red, green and blue. The color specification is in arbitrary device units and no colorimetric characterization is made, i.e., the spatial information is hardware independent, but the color information is not.

The GIF format is limited to palettized color, i.e., only a fixed number of colors can be used in an image. This number is a power of two between 0 and 7, so that theoretically a palette can contain a maximum of 256 colors. The GIF format uses color tables to list the colors in a palette; a raster image then consists of a matrix of indices into a color table, one index per pixel. If a color can be represented with a 3 byte quantity (one byte per channel) and there are a maximum of 256 colors, the size of the image is reduced by a factor of three, since an index can be stored in a byte.

A GIF data stream can contain several raster-based graphics. The data stream can contain a global color table that applies to all graphics; if no global color table is specified, the last global table used is preserved, or the palette is undefined if no global table has been saved. In addition, each graphic in a stream can contain a local color table that supersedes the global table in that scope. A common application for sequences of graphics is to draw them at the same location to produce simple animations. The animation speed is controlled with the delay parameters that can be embedded in GIF streams.

So far we have seen how the storage space for an image can be reduced by reducing the number of colors to those in a small palette and representing each pixel by an index entry into a color table instead of a full color specification in the colorimetric sense. This raster data is a string that can be compressed using a data compression algorithm. The GIF format uses a variant of the Ziv and Lempel adaptive dictionary based technique[3]; this variant was published by Welsh and the algorithm is designated by the author's initials, i.e., LZW[4].

In the LZW algorithm's encoder a sliding window is moved across the data stream and a dictionary is built that maps variable length bit strings from the data stream into fixed length codes. The output is a sequence of codes. The decoder parses the code sequence, recursively builds the same dictionary, and reconstructs the original data stream.

The LZW algorithm in GIF has some small differences from the standard LZW algorithm. The code size is limited to 12 bits per code, which permits a maximum dictionary of 4095 entries. In addition there are some special codes, like a clear code to reset all compression parameters and tables, and an end of information code to mark the end of the image data stream.

Color illustrations that are simple line art, without gradients or smoothing by anti-aliasing pixels, can be produced with a dozen or so colors and are suitable for an indexed representation as a GIF data stream. However, most professionally produced pages for the World Wide Web do contain gradients, drop shadows, etc. and the artwork is often anti-aliased. Professional designers are forced to spend considerable time trying to come up with small color palettes that produce an acceptable number of artifacts in the images due to color approximation.

3 The JPEG format

In the GIF format color information can be lost while spatial information in the image is preserved. The JPEG algorithm takes the opposite approach: the color information can be preserved while some spatial information can be lost. For the World Wide Web application we are interested in, the so-called DCT-based JPEG coders — in which spatial information is lost — are most relevant, but there are also lossless JPEG coders. The art of using the JPEG algorithm is to choose the parameters so that the only spatial information lost is that which cannot be perceived, so that the result is perceptually lossless.

Unlike GIF, JPEG does not prescribe a color encoding. Each pixel in an image is described with up to 255 so-called color components [5] (p. 133]. This allows representation of color images using any colorimetric color space or even to use a discrete spectral representation. DCT-based coders allow each color component to be represented as an 8 or 12 bit quantity. Even with the more compact 8-bit representation, it is obvious how to use a colorimetric color space that keeps the color error below 1 just noticeable difference (jnd).

3.1 The JPEG pipeline

In the following we will consider only the baseline JPEG compression method, because it is virtually ubiquitous. As shown in Fig. 1, this method consists of four stages. First, the raster image is subdivided into blocks. Second, a sequential discrete cosine transform (DCT) is applied, which is an orthogonal and separable transform that allows near-optimum energy compaction and for which a number of fast algorithms with low computational complexity have been developed.

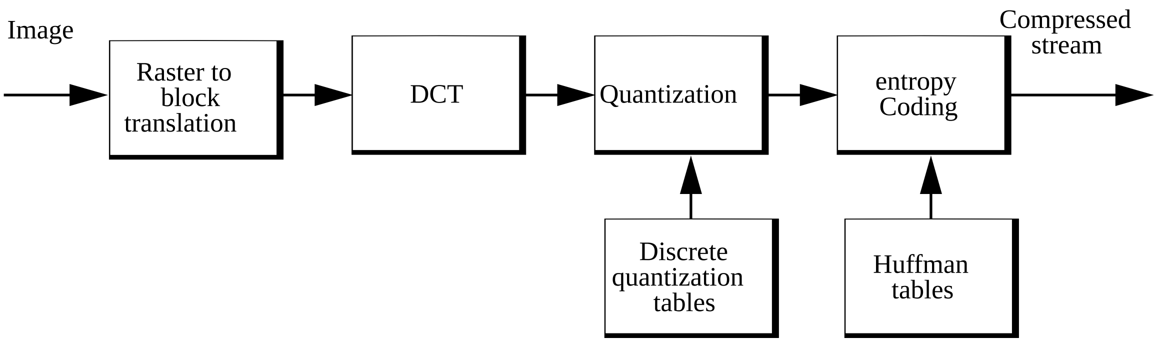

FIGURE 1. Simplified diagram of a discrete cosine transform (DCT) encoder. The processing steps are specified in the JPEG standard, while the tables are specified in each image.

In a third stage the data is quantized based on the discrete quantization table (DQT); each element of the DQT matrix can be any integer value between 1 and 255 and defines the quantization step for the corresponding DCT coefficients. This stage is lossy and implementors seek to design the DQT so that no visible artifacts are introduced in the image. The compression ratio of a file can be increased by setting a so-called *q-factor* or *scaling factor*, which is essentially a uniform multiplicative parameter that is applied to the quantization tables.

The final step in the JPEG method is a lossless Huffman encoder, which "eliminates the entropy" in the image file (see reference[5] on page 16 for a definition of entropy coding). The Huffman table (HT) controls the effectiveness of the lossless compression. At the receiving end, the inverse transforms are applied in reverse order, creating a rendition of the original image.

We revisit the JPEG algorithm and optimize each step for publishing on the World Wide Web. These steps consist of designing new quantization tables for the quantizer of the DCT-encoded image and new Huffman coding tables for the entropy coder. Both table types are part of the JPEG data structure, so it is possible to change the tables at will.

3.2 Designing JPEG DQT matrices

Let us first clarify the terms of lossless and lossy. *Lossless* compression means that no data is lost when an image is compressed and then decompressed, *i.e.*, the original and the processed image are physically identical. *Perceptually lossless* compression means that no visible data is lost when an image is compressed and the decompressed, *i.e.*, the original and the processed image are perceptually identical. This encoding is called *lossy* compression because information is lost.

In *perceptually lossy* compression the loss of information is visible. The perceptual alterations to the image are called *compression artifacts* or, in the case of JPEG, *quantization errors*, because they are caused by large quantization steps. In Lohscheller's psychophysical method based on the human visual system (HVS), the elements in the DQT are chosen so that the quantization errors are just below the visibility threshold, *i.e.*, the compression is lossy but perceptually lossless[2].

The usual method for increasing the compression ratio of the JPEG method is to scale the DQTs by increasing the q-factor until the quantization errors can no longer be accepted by the user. Even when the tables are carefully designed to be perceptually lossless, a large q-factor will introduce artifacts, such as blockiness in areas of constant color or ringing on text characters.

Thus, the q-factor is only a very crude tool to control the compressed file size. It is useful in such applications as video conferencing over the Internet, where the images are sequences of talking heads, and where it is hard to predict the available bandwidth yet image quality is not essential. This method is not useful when static images are used to control appearance in World Wide Web media applications. While in conventional image processing applications images are mostly pictorial, images typical for World Wide Web publishing contain a fair amount of text and computer graphics. Blockiness artifacts in text and graphics deteriorate the æsthetic appeal of the artwork.

As noted by Hoshino et al. [7] the q-factor method is not applicable for compound documents such as are typical in color facsimile, where the images are static, observed for a relatively long time, and the preservation of detail in fine structures such as characters is important. Hoshino et al. note that text must be compressed more conservatively than pictorial images to preserve the sharp edges important for reading efficiency. Their proposed solution is to sharpen the image after decompression, a step that requires additional computation and the addition of a Java applet for each image in order to perform the sharpening. Our proposed solution is to modify the DQT on an element by element basis to preserve the features important for World Wide Web publishing.

In a sequence of increasing sophistication, we design the DQT matrices in two steps, that for reasons that will become clear later, we call the physical world and the perceptual world. The two design iterations are carried out in a certain order; as a completely different framework is used in each step, we call them "worlds."

3.2.1 Physical world

We examine the statistical properties of typical images used in World Wide Web pages. We use a well-known technique called *bit-rate control*, where the key idea is to allocate more bits to the coefficients in which the images contain more energy. A common correlate for the energy is the statistical variance. For our experiments we followed the example proposed by Bhaskaran and Konstantinides[5] (p. 155). We established experimentally that with a bit rate of 6 bits per pixel there are no visible artifacts like ringing in text or blockiness in smooth areas.

Let $N_{k,l}$ be the number of bits allocated for the (k, l)th DCT element. The average bits-per-pixel (bpp) rate* r is given by

$$r = \frac{1}{64} \cdot \sum_{k=0}^{7} \sum_{l=0}^{7} N_{k,l} \qquad \text{(EQ 1)}$$

Let $Y_i[k, l]$ be the (k, l)-th (for $k, l \in [0, 7]$) output DCT element of the i-th 8×8 block in an image. Let B be the number of blocks in the image. The variance $V_y[k, l]$ of the (k, l) frequency component in an image is then computed as

$$V_y[k, l] = \mathrm{var}\,(Y[k, l]) = \frac{1}{B} \sum_{i=1}^{B} (Y_i[k, l] - M_y[k,l])^2 \qquad \text{(EQ 2)}$$

where $M_y[k, l]$ denotes the mean and is defined for each DCT output element as

$$M_y[k,l] = \frac{1}{B} \cdot \sum_{i=1}^{B} Y_i[k, l] \qquad \text{(EQ 3)}$$

For logarithm in base 2, according to rate-distortion theory the number $N_{k,l}$ of bits allocated for the (k, l)th DCT element is given by

$$N_{k,l} = r + \frac{1}{2} \cdot \log \frac{V_y[k, l]}{\sqrt[64]{\prod_{k,l} V_y[k, l]}} \qquad \text{(EQ 4)}$$

Finally, the elements $Q[k, l]$ of the DQT are defined as

*This is the bit rate *before* the Huffman encoder.

$$Q[k, l] = \frac{2046}{2^{N_{k,l}}} \quad \text{(EQ 5)}$$

where 2046 represents the maximal range that the AC output of the discrete cosine transform may have in images with eight bits per pixel, namely [-1023, 1023].

The bits-per-pixel rate does not represent a real bit-rate. In practice one has to experiment with different values and pick a value for which the visual quality is acceptable. The above technique yields discrete quantization tables that result in good compression ratios and acceptable visual quality. The compression can be further increased by taking into consideration the human visual system.

3.2.2 Perceptual world (contrast sensitivity)

In this step we note that image quality depends on the contrast visible at a given spatial resolution. We weight the DQT elements by the contrast sensitivity function (CSF) of the HVS[8]. The result is much better than just increasing the a global scaling factor like the *q*-factor, which is equivalent to a constant CSF. De Queiroz and Rao [9] have described in detail how to weight the DQT by the CSF and we refer to their paper for further details.

3.3 JPEG Huffman tables

The example Huffman tables (HT) in the JPEG standard have been determined for the perceptually lossless compression of pictorial YC_bC_r images. We are interested in those World Wide Web images that consist mainly of line art and text. Experiments by Konstantinides[10] have shown that we can design good constrained-length Huffman codes using the simple ad hoc technique of setting to 2^{-16} all symbol probabilities less than 2^{-16} and then proceeding as if there were no constraints on the codeword lengths. In the experiment described in this paper, for each of our test images, first we compute the probability distribution of all possible symbols for which Huffman code words are needed, and then we design image-dependent custom tables.

We have found that using custom Huffman tables we can improve the compression ratio by between 8% and 14% compared to using the example tables from the JPEG standard. We have also found that if we use image-independent tables obtained by averaging the statistics of all test images, we only lose about 0.5% compression ratio. Hence we are presently using image-independent custom Huffman tables, achieving an average improvement of 11% in the compression ratio.

4 Experimental results

With our custom quantization tables we have been able to achieve typical compression ratios of 1:13, which compares favorably with the GIF method. Unlike GIF, the JPEG method allows the accurate colorimetric communication of color information and features such as gradients and textures.

For each image contained in the Hewlett-Packard home page of May 1997 we have compared the encoded file size using the various formats as follows:

- raw: raw bitmap, stored at 8 bits per color per pixel
- GIF: image posterized from 24 bits per pixel to an adaptive 8-bit index table, then saved in the GIF format
- PNG: a new image that will replace GIF; in this format the images are kept in 24 bits per color and an eight bit alpha channel is used to provide transparency; the compression method is LZW
- TIFF: a popular color image format where a 24 bit per pixel image is compressed with the LZW method
- JPEG: for each image we have computed custom DQTs and Huffman tables for the luminance and chrominance channels

Table 1 summarizes the results. As a measure of the practical importance of these results, note that the customary upper limit for the total size of a page published on the World Wide Web is 70K bytes. This size allows for acceptable download times with a fast modem. With this rule of thumb, we note that this HP home page is only viable when JPEG encoding is used.

TABLE 1. Typical image files sizes in bytes for various formats

file name	raw	GIF	PNG	TIFF	JPEG	ratio
drivers	24,975	4,703	11,901	9,698	1,753	1:14
main 1	120,540	27,732	91,671	78,372	11,146	1:11
main 2	88,770	18,307	55,558	46,614	6,724	1:13
main 3	523,392	116,265	395,117	349,082	39,129	1:13
main 4	130,872	16,890	50,824	46,944	8,088	1:16

5 Outlook

Sometimes the artwork is available as a hard copy original which is scanned into the computer and published as an image. Due to technical limitations of desktop scanners, the text in such color images is often fuzzy. We have presented elsewhere[6] a text sharpening algorithm that operates on the compressed color image and does not require extra computation time. For each cosine basis function, we compare the energy in the original image with the energy in the scanned image. We then scale the basis elements of the discrete cosine transform so that the spatial information attenuated in the scanner is boosted.

In professional World Wide Web pages "splash" image sizes are typically 384×256 pixels, and images are rarely larger than 600×400 pixels (the width of the popular VGA monitors is 640 pixels and typically 40 pixels are used by the browser's scroll bar). This compares to professional photographs scanned at 3072×2048 pixels. Computer display monitors usually operate at 72 dpi, while typical computer printers have resolutions of 600 dpi or more. One would like to have a higher resolution of an image available for printing. In addition, because so much detail is lost when the image is resampled for the low-resolution monitor, users would like to be able to zoom and pan in images.

These features can be provided if the image is stored as a pyramid where at each level the image is stored at a lower resolution and each level is tiled. The World Wide Web browser can then request the specific group of tiles that fills the image window at a given resolution. A new image format called FlashPix implements these features[11]. The tiles have a size of 64×64 pixels and are individually JPEG encoded. The JPEG table design methodology presented above allows new applications such as publishing maps and large drawings on the World Wide Web.

6 Acknowledgments

The research in this paper was joint work with Vasudev Bhaskaran, Konstantinos Konstantinides, and Balas K. Natarajan.

7 References

1. CompuServe, Incorporated, *Graphics Interchange Format (sm)*, July 1990. Electronic version (ASCII text) is available via anonymous ftp from ftp.ncsa.uiuc.edu file misc/file.formats/graphics.formats/gif89a.doc
2. ISO/IEC JTC1 CD 10918, *Digital compression and coding of continuous-tone still images — part 1: requirements and guidelines,* ISO, 1993. The text is identical to ITU-T Recommendation T.81.
3. J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory,* **IT-23,** 3, 337–343, May 1977.
4. T. Welch, "A technique for high-performance data compression," *IEEE Computer,* 8–19, June 1984.
5. V. Bhaskaran and K. Konstantinides, *Image and video compression standards — Algorithms and Architectures*, Second Edition, Kluwer Academic Publishers, Boston, 1997.

6. V. Bhaskaran, K. Konstantinides, and G. Beretta, "Text and image sharpening of scanned images in the JPEG domain," *IEEE International Conference on Image Processing*, **2**, 326–329, Santa Barbara, October 26–29, 1997.

7. Y. Hoshino, D. Ando, S. Higashino, and M. Tsukamoto, "Applicability of a standardized discrete cosine transform coding method to character images," *J. Electronic Imaging*, **1**, 3, 322–327, 1992.

8. W.-H. Chen and C.H. Smith, "Adaptive Coding of Monochrome and Color Images," *IEEE Transactions on Communications*, **COM-25**, 1285-1292, 1977.

9. R.L. de Queiroz and K.R. Rao, "Human visual sensitivity-weighted progressive image transmission using the lapped orthogonal transform," *J. Electronic Imaging*, 1, 3, 328–338, 1992.

10. K. Konstantinides, "Effects of Custom Huffman Tables on the Compression of CIELAB Images Using JPEG," *SID Digest of Technical Papers*, **XXVII**, 951–954, San Diego, 1996.

11. For more information see http://www.kodak.com/go/flashpix or http://image.hp.com/htdocs_sys/learnmore.html or http://www.livepicture.com/products/flashpix_viewer.html