Visual Compression: Understanding How It Works

Scott Bondurant Chandler

Abstract

From web graphics to instructional media, compression plays a significant role in the visuals we use. By addressing current compression techniques, the benefits and the costs of algorithms is evaluated. Terms such as temporal compression, spatial compression, run-length, LZW, microblock, DCT and delta frames are defined.

Overview

In a digital world, bandwidth (the data pipe for digital data) is at a premium. We have all experienced the immense delays in loading a slow web page. With innovations like high-definition television on the horizon, making the most of limited spectrum is increasingly important.

There are three common types of compression: reductionist, spatial and temporal. Additionally, these strategies may be lossless or lossy. With lossless compression the uncompressed image is a perfect digital match for the original; in lossy compression greater compression is possible because acceptable quality loss occurs.

Reductionist Compression

The most obvious form of compression occurs when data is simply thrown away. The simplification of the signal to be compressed is certainly the easiest kind of compression.

Working At Multimedia Quality

Consider the difference between print quality and screen quality. Most professional scans are created at 300 pixels per inch (ppi) when adjusted for the print size. Multimedia and web designers work at a far lower resolution of 72 ppi. Throwing away these extra pixels result in a screen quality file that takes up only 5.76% of its original size (compression of 17.36:1).

Reducing The Size And Frame Rate Of Digital Video

When digitizing video for use on CD-ROMs, a similar trick is used. High quality video intended for CD-ROM is roughly a third the horizontal size and half the vertical size of broadcast television. This cuts the data rate by about 83%. For web video, it is common to cut the physical size even more dramatically. While this might seem dramatic, consumer VHS video cassette records less horizontal resolution (Taylor, 2000).

CD-ROM based video uses only 15 frames per second instead of the 30 found in NTSC broadcast video. This results in halving the bandwidth before other types of compressions are used.

Reducing The Color Signal Of Television

Even analog broadcast television is reduced in analog form. Because of the way that the human perceptual system works, luminance (tone) is more important than chrominance (color). Computers tend to store values using the RGB color model. The amount of red, green and blue is stored on an additive scale in which 0 is no light of a certain color and 255 is solid color. When all three values are present in equal amounts, the viewer perceives gray. When no light is present, black is seen; and when solid

Figure 1
An Example Of Temporal Compression
(video clip used with permission of Carrie Steffey, 2000)

red, green and blue light are present, white is visible. In this color model, all three primary colors of light are equally represented. Color television uses a different system known as YUV or more accurately YC_BC_R. The Y value represents luminosity, perhaps more easily thought of as grayscale. When black and white television was standard, only the Y signal was broadcast. Color television extended the standard to include color, represented in YC_BC_R by the C_B and C_R components. C_B represents the differences between Y and the blue color component while C_R represents the difference between between Y and red color component. The obvious question is "where is the difference between luminance and green?" The answer to this question is revealed in the formula for luminosity: Y = 0.59G + 0.30R + 0.11B (Fibush, 1999). Note that luminance is based much more heavily on green than on red or blue; as such, the Y value is used as an indication of green. Further, the frequency bandwidth allocated for television weights the luminance (Y) equal to the sum of the chrominance signals (U+V). Converted into a digital equivalent, luminance has 8 bits of data while chrominance have 4 bits each. This system of weighting luminance greater than chrominance, results in a savings of 33.3% over the RGB systems used in most computers. Interestingly, modern compression schemes for digital video have begun to adapt to giving tone more bandwidth and resolution than color.

Blur Video

When capturing digital video, it is common to have sampling errors. These visual problems can complicate the compression process. One common trick is to eliminate these "stray pixels." One technique is to shoot video with a short depth of field. Throwing the background out of focus not only directs the viewer's attention on the foreground but also makes the background easier to compress. Good compression software will either slightly blur an image before compressing or actually hunt down stray pixels and remove them. In effect, we're reducing the complexity of our footage before applying other forms of compression.

Color Palettes

Another way to reduce bandwidth is to simplify the colors palette. Tonally flat images compress better than images that use a wide palette of colors. In fact, some formats severely limit the number of colors available. Full color images can use up to 16.7 million distinct colors. Only a fraction of these colors may be used in a specific image.

The Compuserve GIF format only supports 256 colors. When images are converted to GIF87 or GIF89a, the numbers of colors must be reduced to 256 colors or less.

There are several different techniques that can be used to reduce the number of colors. One way is to convert to a distributed palette of colors. In the middle 1980s Apple developed a matrix of 256 colors developed to closely fit any of the millions of available colors. To maintain cross-platform compatibility, most web developers in the middle 1990s used a similar palette of 216 colors. These 216 index colors are often called the 6x6x6 clut (color look up table). This three dimensional cube of color is developed by permuting six distinct, distributed values of the three additive colors. These values are 0, 51, 102, 153, 204, and 255. For example, Pantone 202's closest RGB value is 145, 44, 69. This means that the red component is 145 out of 255, the green component is 44 out of 255 and the blue component is 69 out of 255. Because this color is not within the 6x6x6 palette, the color is scaled to the closest possible color 145, 44, 69. The color is not an exact match but is certainly a maroon. It is possible to develop a common palette of 256 color for each image. For compatibility with older computers, it is best to build a super palette. A super palette optimizes all of the images in a given web site to the same 256 colors.

In many cases, graphics don't need to use all 256 colors. For instance, most type-only buttons on the web are really one color graphics with a transparent background. These files compress much more accurately and create smaller files. In this case, the color reduction offers tremendous storage savings.

If the colors within a web graphic can be named—a smaller, better looking file will be created from reducing the color palette than from using other compression techniques. In general, each named color exists with an anti-alias to a background color. This soft edge actually contains a series of separate colors which act as tints against the background. Allow eight small steps of color for each color you can name in a web graphic. By using only 32 colors in a graphic, instead of the default 256, you can dramatically improve the overall compression.

Reducing the number of colors in a specific graphic or web site is a very tedious and time consuming process. Most of the graphics on the web no longer use the GIF file format with its color reduction techniques. GIF is now used primarily for specialty graphics such as "spot color" buttons, text buttons, images that must have transparent backgrounds and because it offers animation.

Temporal Compression

Video can be thought of as a sequence of still images presented at 29.97 frames per second. Many of these individual frames share the identical image data.

Consider a perfume logo freezeframe at the end of an advertisement on broadcast television. Even though no movement can be seen, the same 349,920 pixels are being

sent 29.97 times per second. In effect, the same image is being sent over and over. In the world of analog television, spectrum is reserved and there is nothing to be gained by not sending the same picture over and over. In other formats, however, we could better utilize that bandwidth for other network users, higher quality video or more instructional content.

Figure 1 shows a concrete example of temporal compression. The top row of the figure shows four sample images taken from a "talking heads" lecture captured at a rate of five images per second. Because the digital video camera is setting on a tripod, only the professor is moving. The benefit of temporal compression is illustrated in the bottom row of images. The white rectangle indicates that the entire image of the professor and background is sent once. However, the second image shows that only a small percentage of the actual image changes in the first 0.20 seconds. Although the parts of the women in motion continue to change with each successive frame, most of the image need not be resent. In this example, temporal compression reduces file size to less than 10% of the original movie.

Frankly, there is no advantage to not temporally compressing delivered content and significant reason to compress. Image makers are encouraged to aggressively use temporal compression.

Temporal Frame Types

Temporally compressed footage works by defining several different types of frames. A combination of temporal frame types allows the compression to occur.

The simplest type of frame is a reference frame, often called a keyframe. Because keyframe has a completely different meaning in animation than in digital video, this article will use the more technical term "i" frame. These "i" frames are used to define a specific moment in time. The "i" frame acts as a starting point for other types of frames. Reference frames tend to be large in terms of file size because they attempt to capture the entire content in one moment of time.

Following an "i" frame, there are usually a series of delta frames also known as "p" frames. Delta frames look at the change between the current frame and previous frames. A "p" frame has a smaller footprint than the "i" frame because it only specifies data that has changed in the last fraction of a second.

Revisiting the example of the freezeframe above, the delta frame would only need to contain the code to retain the previous screen. This code generally takes 8 bytes of data while resending the complete image would require 1,049,760 bytes per frame.

"P" frames are dependent on the frames before them in time. Consider a digital movie that contains only one "i" frame and then 216,000 "p" frames. Because each "p" frame is based on the one before it, it wouldn't be feasible to rewind the media only a few seconds during the credits: backing up one frame would require the decompression engine to go all the way back to the beginning of the footage (the most recent "i" frame) and work forward until the desired frame. In fact, some early temporal compression schemes had this flaw because they used very few "i" frames. If the user is able to move backwards or rapidly forward through the content, "i" frames should occur often. If the user will only watch the media forward at actual speed, multiple "i" frames may not be needed.

In reality, there are other reasons why "i" frames are needed. Many compression algorithms are hungry for reference frames. Cinepak, for instance, decays in quality if an "i" frame isn't encountered every second or so. MPEG-2, the standard for mini-dish satellite and DVD-Video, tends to have an "i" frame about every 10 frames. Other animation routines use fewer keyframes, the animation CODEC at maximum quality needs only one reference frame if the user can't scrub through the media. Sorenson is an especially flexible CODEC: Sorenson uses a high number of keyframes for live video, about one reference frame for every five delta frames; but also does well with very few frames for static instructional graphics. With hundreds of delta frames for each reference frames, Sorenson will simply improve the quality of a small piece of the total graphic with each delta frame.

Another unique type of temporal frame is the bidirectional interpolated frame, known as a "b" frame. The "b" frame gets information from the "i" frame before it in time and the "i" frame after it in time. While this may at first seem complex, it isn't. Some transitions between motion footage involve wipes from one clip of footage to another. While these transitions are difficult to compress using conventional techniques, the "b" frames can simply specify which image to reference for each pixel at each stage of the transition (Lee, 2000).

There are other types of temporal frames. Almost all motion CODECs use "i" frames and some implementation of "p" frames. "B" frames and other types of temporal frames are more rare.

Video that doesn't use temporal compression can be thought of as containing only "i" frames. For example, Motion JPEG and DV (digital video CODEC) do not use temporal compression. Performing a cut at an "i" frame is very easy while cutting at a "p" frame requires that section of footage to be recompressed. In effect, the cut point must be converted to a reference frame and the delta frames following the cut point must be recompressed to be based off of the new reference frame. However, both of these compression systems are used only where there is adequate bandwidth to not need temporal compression.

In general, temporal compression is a delivery format.

While footage is being created, edited and composited temporal compression is not used. Temporal compression is performed as a last step before content is mastered.

Temporal compression can dramatically reduce file size or more appropriately use limited bandwidth. When temporal compression is done correctly, image quality is not harmed even though compression can be significant.

Spatial Compression

Spatial Compression Types

Spatial compression reorganizes data to take up less space and therefore make it easier to send. Spatial compression occurs in two different forms lossless or lossy. Different uses dictate the use of different compression schemes.

Lossless compression looks for patterns in each image that can be repeated exactly or conveyed as instructions rather than images. Despite being reorganized, an image that has been compressed with lossless compression will match the original image exactly. The "animation" compression scheme often used for the development of special effects is an example of lossless spatial compression.

Lossy data compression works in a different way. Because lossy compression doesn't attempt to recreate the image exactly, it can create much smaller files. It would be unfair to suggest that lossy compression is always unattractive. Lossy data compression can be of exceptionally high quality.

Run Length Compression (RLE)

Run length is a lossless compression routine which can compress simple files with dramatic results. Run length checks to see if a single color is contiguously repeated multiple times on the same line. For instance, Adobe Photoshop native files use RLE compression.

Icons are small raster images used to represent programs, data files and other structures. Historically, most icons are 32 pixel wide by 32 pixel tall and use a limited color palette. The smiley icon shown in Figure 2 is comprised of only black, white and yellow. The normal, uncompressed way to represent this icon would be to simply indicate the color of each of the 1,024 pixels in order. Figure 3 shows a simple representation of this technique using the letters K, W and Y to indicate black, white and yellow, respectively (K for blacK because B is commonly used for Blue). As with many stylized illustrations, there is a high degree of redundancy in the image. A simple form of run length is shown. Each letter has been preceded by the number of times it appear consecutively in each row (Figure 4). Because the icons dimensions are known and spaces need not be stored, the characters can be written as a single continuous stream (Figure 5). The run length compressed version of the entire icon is only 401 characters instead of 1,024.

Some images have very few or no adjacent matching pixels. This particular form of run length would actually increase the file size by putting the value "1" in front of

Figure 2
Smiley Icon, 32 pixels x 32 pixels, 256 colors

Figure 3
Smiley Icon, Top Half, Uncompressed As Color Initials
w=white, k=black, Y=yellow

```
Line 01 WWWWWWKKKKKWWWWWWWWWWWWWWWWWW
Line 02 WWWKKKKKKKKKKWWWWWWWWWWWWWWWWW
Line 03 WWKKKKYYYYKKKKWWWWWWWWWWWWWWWW
Line 04 WKKKYYYYYYYYKKKWWWWWWWWWWWWKKK
Line 05 KKKYYYYYYYYYYKKKKWWWWWWWWKKKKWW
Line 06 KKYYYYYYYYYYYYKKKWWWWWKKKKKKKWW
Line 07 KKYYYYYYYYYYYYYYKKKKKKKKKKKYKKWW
Line 08 KKYYYYYYYYYYYYYYYKKKKKKYYYYKKWW
Line 10 KKYYYYYYYYYYYYYYYYYYYYYYYYYKKWW
Line 11 KKYYYKKKKKKKKKKKKKKKKKKKKYYKKWW
Line 12 KKYYKKKKKKKKKKKKKKKKKKKKKYYKKWW
Line 13 KKKYYKKKKKKKKKKKKKKKKKKKKYYKKWW
Line 14 WKKYYYKKKKKKKKYYYKKKKKKKYYKKWWW
Line 15 WKKKYYYKKKKKKYYYYYKKKKKKYYKKWWWW
Line 16 WWKKYYYYKKKKYYYYYYYKKKKYYKKKWWWW
```

Figure 4
Smiley Icon, Top Half, Finding Repeating Color Pixels

```
Line 01  6W  5K 21W
Line 02  3W 10K 19W
Line 03  2W  4K  4Y  4K 18W
Line 04  1W  3K  8Y  3K 14W  3K
Line 05  3K 10Y  4K  9W  4K  2W
Line 06  2K 13Y  3K  5W  7K  2W
Line 07  2K 14Y 11K  1Y  2K  2W
Line 08  2K 16Y  6K  4Y  2K  2W
Line 10  2K 26Y  2K  2W
Line 11  2K  3Y 21K  2Y  2K  2W
Line 12  2K  2Y 22K  2Y  2K  2W
Line 13  3K  2Y 21K  2Y  2K  2W
Line 14  1W  2K  3Y  8K  3Y  8K  2Y  2K  3W
Line 15  1W  3K  3Y  6K  5Y  6K  2Y  2K  4W
Line 16  2W  2K  4Y  4K  7Y  4K  2Y  3K  4W
```

Figure 5
Smiley Icon, Stream Of Run Length Compressed Color Initials

```
6W5K21W3W10K19W2W4K4Y4K18W1W3K8Y3K14W3K3K10
Y4K9W4K2W2K13Y3K5W7K2W2K14Y11K1Y2K2W2K16Y6K4
Y2K2W2K26Y2K2W2K3Y21K2Y2K2W2K2Y22K2Y2K2W3K2
Y21K2Y2K2W1W2K3Y8K3Y8K2Y2K3W1W3K3Y6K5Y6K2Y2K
4W2W2K4Y4K7Y4K2Y3K4W
```

each color. For this reason, most run length compression schemes only specify the number of pixels that repeat when three or more identical pixels appear in a row.

Run length is commonly used because it is an extremely fast and easy way to compress and to decompress. As with most types of compression, the larger the size of image, the more dramatic the expected compression ratio.

Lossless Pattern Matching

Almost all data contain patterns. Some of these patterns are relatively simple and others are more subtle. In fact, many of the general purpose compression schemes began by attempting to compress type. Consider this document for a moment. Which words or phrases are most common? The word "compression" obviously appears many times. The word "the" is not only common but usually appears with a space on either side.

A simple form of compression might use an character like the tilde ("~") to replace the name of a company or organization. If you could replace every occurrence of "International Visual Literacy Association" with a single character like "›" 41:1 compression has been obtained for that phrase.

As the files is compressed, a hash table is built. The table has a list of all the abbreviations used in the file. Although the file is smaller during storage and transmission, the file can be precisely reconstructed with minimal effort. By building a separate hash table for each file, the unique patterns and word choices of each file can be exploited.

The most widely used pattern matching routines are Huffmann, Lempel-Ziv and LZW. These routines carefully build a encoding and decoding table but happen relatively quickly by modern standards. Huffmann builds a decoding table that is sent before the data component while Lempel-Ziv and the optimized LZW (Lempel, Ziv and Welch) build the table as the data is encoded. While building the table "on the fly" slightly degrades the effectiveness of compression, the one-pass technique results in a faster compression and decompression (Welch, 1984).

The most common "word" length is 12 bits. (Welch, 1984) Although this could make individual letters longer than the ASCII standard of 8 bits per character, the character set is extended to 4,096 entries. As such, many more words and patterns will be compressed than with an 8 bit word length. In general the gains in compression more than offset the length added to a single character.

The LZW routine is used for compressing GIF files and in most implementations of TIFF. Although better pattern matching techniques exist, the mathematical simplicity and relatively fast nature of LZW make it the compression method of choice for simple applications.

Figure 6
Examples of microblocks from JPEG, DCT hash table

Modern Lossy Compression

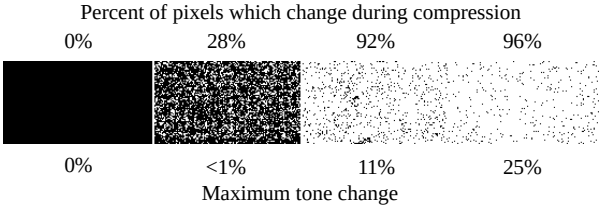
Discrete cosine transformation (DCT) became the best phenomenon in compression. Known best as the compression system behind JPEG images (Joint Photographer's Expert Group), DCT offers a wide range of highly lossy to almost-lostless compression. DCT also builds a table of the most common data patterns. However, an exact fit isn't needed. Under JFIF (JPEG File Image Format), the most common variant of JPEG, each small two dimensional section of the image is compared to the entries in the quantization tables. In most cases a "best fit" is determined by means of an absolute accuracy. Figure 6 shows 25 sample microblocks from a heavily compressed image. Each of these blocks is an 8x8 grid shown at high magnification.

JFIF offers the user 100 different quality levels. The quality level is used to determine the desired compression ratio. Given the approximate compression level, a certain number of small, two dimensional patterns are isolated from the image. The algorithm then compares each small sample in the image file to the available patterns in the compression table. Although the patterns may not be exact, a suitable replacement block is usually found.

Because JPEG has such a wide range of compression ratios and qualities, it is useful to test the amount of actual tonal shift in the different compression amounts. For this example, three different quality JPEGs were created using the Photoshop JFIF format from a scanned 8x10 inch tonal image. The files was saved at maximum, medium and very low qualities. Using the lossless TIFF as a reference, differing levels of compression can be seen (see Figure 7) although the halftone process used in printing hides much of the quality loss. Perhaps most remarkable is the relatively high quality seen at medium levels of compression. Although this image is quite visually complex, a version acceptable for use on the web is available at 3.48:1 compression. This offers a net result equal to 2.30 bits per pixel instead of the 8.00 bits per

Figure 7
Photograph Compressed At Four Qualities

Figure 8
Pixel Change During Compression, Bottom Of Same Image



pixel for an uncompressed image.

If we use the same data files to continue analyzing tonality, we can see an even more remarkable trend. One indicator of image quality is the "fit" of each pixel in each microblock. By analyzing the tonal shift in aggregated pixels we can determine how much the image has changed. As you might expect, a large percentage of the pixels do change as image compression ratio increases. This is perhaps best illustrated in figure 8 where nonwhite pixels indicate any change. However, the amount of that change is minimal at modest levels of compression. In the high-quality JFIF, the median pixel did not change value at all. Even in the medium quality JFIF, the median pixel changed only 1.5% Although the majority of the image is changing when losslessly compressed, the tonal shift is often very subtle. Only when image quality is low did we see large changes in tone. As mentioned previously the halftone process used for commercial printing can reduced visible change even further. Most professional printing, however, correctly uses only modest image compression.

When compression quality is low, it is possible to see the flaws or artifacts in a image compressed with DCT. When artifacting becomes extreme, it is possible to see numerous distracting artifacts known as a carnival. Carnivals are often observed around the edges or anti-aliased type and in areas of high contrast.

Non-linear Video And Spatial Compression

Almost all of the expensive, proprietary nonlinear editing systems use spatial compression. In fact, the DV format is compressed 5:1, while DVD-video is compressed approximately 14:1. Some high end edit systems allow users to choose their compression ratios from as low as 2:1 to as high as 120:1, depending on their situational needs to balance hard drive space and playback bandwidth with image quality considerations.

Future Compression

Over the last ten years, two new types of compression have become the focus of intense study. Although generally grouped together, Fractal compression and Wavelet technology take different approaches.

Fractal compression technology is contextually based. Like JPEG, segments of the image are sampled and used to define data which can be reconstituted into a lossy approximation (Fisher, 1995). The key to the fractal approach is that these reference blocks can be modified if needed. While JPEGs blocks are used exactly as they appear in the original imge, the fractal blocks can be manipulated in four distinct ways. Each original matrix block may be used again as a scale, stretch (non-proportional scale), skew and rotate (Fisher, 1995). Simply put, a gentle curve definition may be used repeatedly at different sizes. Although exceedingly processor intense, fractals can become more accurate the longer they are processed. Because of the iterative nature of fractal compression and a need for incredibly long processing times using existing technology, the true value of fractals has yet to be realized. With advances in technology, fractal compression may become common. For the immediate future, however, there seems to be no visible advantage to fractal compression technology.

Wavelet compression technology also uses an iterative approach but using fundamental shapes that are not contextually defined. Instead, a prefined series of waves are used to approximate part of the image. A series of these waves are then combined to create more complex forms. By iterating on image data by combining large

numbers of increasing smaller waves, the image can be compressed (Mallet & Falzon, 1999). Wavelet mathematics is similar to fractal compressions in many ways. In fact, one area of wavelet research covers refineable functions which, like fractals, are “short linear combinations of dilated and translated versions of themselves” (Perrier & Wickerhauser, 1999, p. 78.).

The new JPEG2000 standard uses wavelet compression. As of this writing, JPEG2000 is a specialty technology available only as a commercial product. However, many experts feel that the alpha transparency, region of interest support and progressive encoding will make JPEG2000 the de facto standard in the future. Region of interest compression, allowing encoding at different quality levels, is already common in video applications while the variety of resolutions supported by progressive encoding is popular in medical imaging.

Compression Ratios

Most forms of lossless compression achieve compression near 2:1 for grayscale or color photography. Some forms such as fax compression and orthochromatic LZW achieve higher ratios up to 12:1. The lossy algorithms routinely offer better than 2:1 compression and are commonly used at up to 50:1 compression. As processor speeds improve and embedded technology become more common, more “effort” can be put into compression and decompression technology.

Summary

All three varieties of compression offer unique advantages and challenges. Clearly, no one type or algorithm can meet all compression needs. By carefully weighing the options, however, one can make valid and reasonable compression choices that result in the best possible image in a given context.

References

Fibush, David, K. (1997). *A guide to television systems and measurements.* Tektronix: Beaverton, Oregon.

Fisher, Y. (1995), Mathematical Background. In *Fractal Image Compression: Theory and Application*. Ed. Yuval Fisher. Springer-Verlag. New York.

Lee, W., (2000), *MPEG compression algorithm*, Online: http://icsl.ee.washington.edu/~woobin/papers/General/node2.html, April 27, 2000.

Mallet, Stéphane & Flazon, Frédéric (1999) Understanding Wavelet Image Coding. In A*dvances in Wavelets*, Ed. Ka-Sing Lau.Spring-Verlag. Singapore.

Perrier, Valerie & Wicherhauser, Mladen Victor (1999) Multiplication of Short Wavelet Series: Using Connection Coefficients. In *Advances in Wavelets*, Ed. Ka-Sing Lau.Spring-Verlag. Singapore.

Taylor, J., (2000), *DVD-FAQ*, Online: http://www.dvddemystified.com/dvdfaq.html, April 27, 2000.

Welch T. A., (1984), *IEEE Computer*, A Technique for High-Performance Data Compression. Volume 17, Number 6, June 1984.