How File Compression Works

Compression may be the savior of the Internet. Already sarcastically dubbed as the World Wide Wait, the World Wide Web makes use of numerous compressed images that speed up download times. Large programs, such as Netscape 6, often come as compressed archives that simplify and greatly reduce the amount of time that is needed to download the files.

Different compression techniques are best suited for various applications. Lossy compression is most often used on image, sound, and video files. Some information can be easily removed from these files without noticeably changing the quality of the file. Other files, such as program files, are not tolerant of any type of data loss. Lossless compression scans a file for redundant strings of data and uses a pointer to refer back to the first instance of the string. The process can be reversed to reconstruct the data perfectly, without any loss of data.

Data Degradation

Most lossy-compression algorithms let you adjust the level of compression. The higher the level of compression, the more information is removed and the smaller the file size. Higher compression levels, however, begin to noticeably affect the quality of the image. The image to the right is an original bit-mapped image (uncompressed). It is 6,287,486 bytes in size. As a JPEG (Joint Photographic Experts Group) image compressed at 25%, the file's size has shrunk to 268,635 bytes. At 100% compression, the image is only 39,514 bytes, but it has suffered significant loss of data and image degradation.

Original Bit Map
Size: 1,198,081 bytes

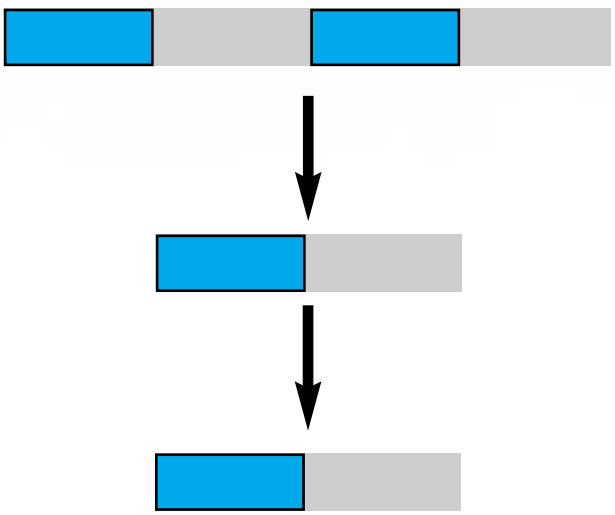
25% Compression
Size: 268,635 bytes

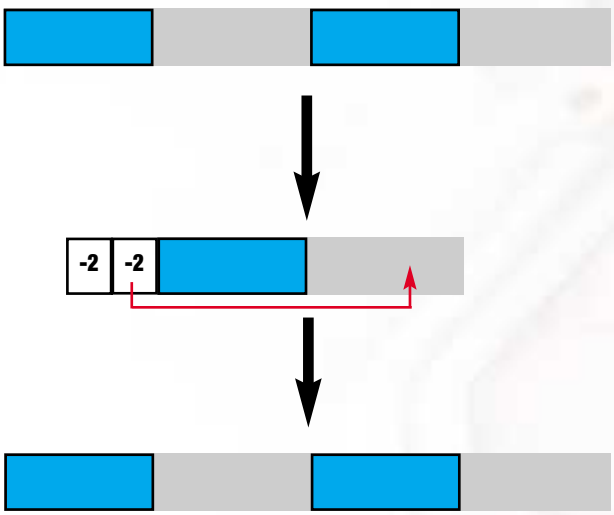
Lossy Compression

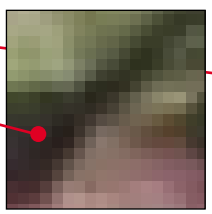
Lossy compression uses an image-compression algorithm that focuses on areas of high contrast and eliminates a few pixels to save space. Although not equivalent, the compressed image will look almost exactly like the original. There will, however, be some loss of information.

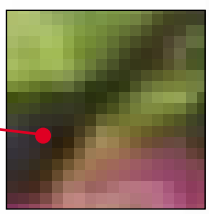
Lossless Compression

When compressing an image, a lossless algorithm may search for redundant colors and place a pointer to reference the first instance of that color. Because the pointer is smaller than the information needed to define the color, the file size is smaller after compression. No data is lost with this type of compression.









50% Compression
Size: 191,749 bytes

75% Compression
Size: 115,121 bytes

100% Compression
Size: 39,514 bytes

Compiled by Chad Denton
Graphics & Design by Lori Garris