

Portable Network Graphics PNG is designed as a GIF successor, and uses the LZ77-variant found in *gzip*: <http://www.libpng.org>. A short history of PNG may be found in [60].

Wavelets The Wavelet Digest at <http://www.wavelet.org> may be a good starting point. Colm Mulcahy's Mathematics Magazine article [52] contains an elementary introduction. A number of his papers, along with Matlab code and images, are available from <http://www.spelman.edu/~colm>.

The images in [Figures 10.18](#) and [10.19](#) were generated with Geoff Davis' Wavelet Image Compression Construction Kit, available through <http://www.cs.dartmouth.edu/~gdavis>.

Ross Williams opened "Dr Ross's Compression Crypt" as the first edition of this book was going to press. <http://www.ross.net/compression/> contains notes and sources for his work on various compression-related topics, including the LZRW family of algorithms.

C.2 Data compression and patents

The area of patents is a minefield for those interested in data compression. In testimony prepared by the LPF for the 1994 Patent Office Hearings, Gordon Irlam and Ross Williams write:

As a result of software patents, many areas of software development are simply becoming out of bounds. A good example is the field of text data compression. There are now so many patents in this field that it is virtually impossible to create a data compression algorithm that does not infringe at least one of the patents. It is possible that such a patent-free algorithm exists, but it would take a team of patent attorneys weeks to establish this fact, and in the end, any of the relevant patent holders would be able to launch a crippling unfair lawsuit anyway.¹

Companies such as Oracle, Adobe, and Autodesk presented testimony against software patents; on the other side were companies such as IBM, Intel, Microsoft, and SGI. There were middle-ground positions: Sun testified that "the [patent] system is indeed broken and needs addressing," but did not call for elimination.

Donald E. Knuth, in a letter to the patent office, writes

In the period 1945–1980, it was generally believed that patent law did not pertain to software. However, it now appears that some people have received patents for algorithms of practical importance—e.g., Lempel-Ziv compression and RSA public key encryption—and are now legally

¹From "Software Patents: An Industry at Risk" by Gordon Irlam and Ross Williams.

preventing other programmers from using these algorithms...If software patents had been commonplace in 1980, I would not have been able to create [T_EX], nor would I probably ever have thought of doing it, nor can I imagine anyone else doing so...The basic algorithmic ideas that people are now rushing to patent are so fundamental, the result threatens to be like what would happen if we allowed authors to have patents on individual words and concepts...There are far better ways to protect the intellectual property rights of software developers than to take away their right to use fundamental building blocks.²

Perhaps the best known patent problem (other than the infamous exclusive-or patent)³ concerns an LZ78-type scheme (Lempel-Ziv-Welch). The scheme is widely used, but the general internet user probably only heard of the patent problem when Unisys pressed for royalties in late 1994 in connection with the GIF graphics format:⁴

The LZW algorithm used in *compress* is patented by IBM and Unisys. It is also used in the V.42bis compression standard, in Postscript Level 2, in GIF and TIFF. Unisys sells the license to modem manufacturers for a onetime fee. CompuServe is licensing the usage of LZW in GIF products for 1.5% of the product price, of which 1% goes to Unisys; usage of LZW in non-GIF products must be licensed directly from Unisys.

And, as an example of the patent mess,

The IBM patent application was first filed three weeks before that of Unisys, but the US patent office failed to recognize that they covered the same algorithm. (The IBM patent is more general, but its claim 7 is exactly LZW.)⁵

To be precise, the patent office maintains that *algorithms* are not patentable, but an algorithm used to solve some particular problem is considered patentable. Irlam and Williams write: “Thus the ‘RSA algorithm’ is not patentable, but ‘use of the RSA algorithm to encrypt data’ is patentable...For all practical purposes, such patents can be considered patents on algorithms.”

The Stac–Microsoft lawsuit involved an LZ77-type scheme:

Waterworth patented⁶ the algorithm now known as LZRW1 (the “RW” is because Ross Williams reinvented it later and posted it on comp.compression on April 22, 1991). The *same* algorithm has later been patented by

²Reported in *Programming Freedom*, the Newsletter of the League for Programming Freedom, February 1995.

³4,197,590 Method for dynamically viewing image elements stored in a random access memory array, filed Jan 19, 1978, granted Apr 8, 1980. Cadtrack has collected large sums of money and successfully defended this patent which includes claims of “XOR feature permits part of the drawing to be moved or ‘dragged’ into place without erasing other parts of the drawing.”

⁴A short note on the Unisys action and an introduction to software patent issues can be found in the March 1995 issue of *Scientific American* [10]. A new graphics specification, Portable Network Graphics (PNG or “ping”), was developed partly in response to the Unisys action. PNG is a lossless scheme with more capabilities than GIF.

⁵The patents are 4,814,746 (IBM) and 4,558,302 (Unisys). Much of this patent information comes from the FAQ maintained by Jean-loup Gailly, and from the LPF.

⁶4,701,745 Data compression system, filed Mar 3, 1986, granted Oct 20, 1987.

Gibson & Graybill.⁷ The patent office failed to recognize that the same algorithm was patented twice, even though the wording used in the two patents is very similar.

The Waterworth patent is now owned by Stac Inc., which won a lawsuit against Microsoft, concerning the compression feature of MSDOS 6.0. Damages awarded were \$120 million. (Microsoft and Stac later settled out of court.)

The Gibson & Graybill patent is very general and could be interpreted as applying to any LZ algorithm using hashing (including all variants of LZ78). However, the text of the patent and the other claims make clear that the patent should cover the LZRW1 algorithm only. (In any case the Gibson & Graybill patent is likely to be invalid because of the prior art in the Waterworth patent.)

The LZRW1 scheme was presented by Williams in [83]. The original GNU zip (*gzip*) was to have used LZRW1. Patents on arithmetic coding affect the graphics compression scheme known as JPEG:

IBM holds many patents on arithmetic coding.⁸ It has patented in particular the Q-coder implementation of arithmetic coding. The arithmetic coding option of the JPEG standard requires use of the patented algorithm. No JPEG-compatible method is possible without infringing the patent, because what IBM actually claims rights to is the underlying probability model (the heart of an arithmetic coder).

From the the documents in the Independent JPEG Group's source distribution:

It appears that the arithmetic coding option of the JPEG spec is covered by patents owned by IBM, AT&T, and Mitsubishi...For this reason, support for arithmetic coding has been removed from the free JPEG software. (Since arithmetic coding provides only a marginal gain over the unpatented Huffman mode, it is unlikely that very many implementations will support it.)

More information and references (on both sides of the patent issue) can be found in the LPF materials.

⁷5,049,881 Apparatus and method for very high data rate-compression incorporating lossless data compression and expansion utilizing a hashing technique, filed Jun 18, 1990, granted Sep 17, 1991.

⁸Here's a few from the FAQ: 4,286,256 Method and means for arithmetic coding using a reduced number of operations, granted Aug 25, 1981.

4,463,342 A method and means for carry-over control in a high order to low order combining of digits of a decodable set of relatively shifted finite number strings, granted Jul 31, 1984.

4,467,317 High-speed arithmetic compression using concurrent value updating, granted Aug 21, 1984.

4,652,856 A multiplication-free multi-alphabet arithmetic code, granted Feb 4, 1986.

4,935,882 Probability adaptation for arithmetic coders, granted Jun 19, 1990.