

The JPEG 2000 Image Coding Standard

Diego Santa Cruz, Touradj Ebrahimi, and Charilaos Christopoulos

(Reprinted From Dr. Dobb's Journal)

The JPEG 2000 is a new image coding standard that specifies only the decoding algorithm and the compressed data format—the encoder algorithm is

When it comes to image coding, JPEG is the first thing that comes to mind. JPEG is the most widely used compression Standard for true color continuous-tone images. It was created by the Joint Photographic Experts Group (<http://www.jpeg.org/>) around 10 years ago, and although it delivers good quality at low- and medium-compression ratios, it lacks flexibility of use and its quality at high compression leaves much to be desired.

Being aware of this, the JPEG members have been hard at work on JPEG 2000 — a new image-coding Standard that is flexible and provides better compression. Now that some parts of this new Standard are finalized, and many others nearly so, it seems like a good time to review what JPEG 2000 is, what it offers, how it works, and what improvements over JPEG can be expected. As you will see, there is more to it than just the millennium end-naming convention. For lossy compression, the only widely used standard is JPEG. For lossless compression, the popular formats are GIF and PNG, which are also well known. A more recent format is JPEG-LS, which has also been created by JPEG and targets low-complexity, highly efficient lossless coding systems, and therefore does not include many of the features present in JPEG 2000.

JPEG 2000 is based on the current state-of-the-art, delivering both lossy and lossless compression in one unified way, with additional features such as random access, scalability, region of interest coding, and so on, with great flexibility of use.

What Does JPEG 2000 Bring?

JPEG 2000 specifies only the decoding algorithm and the compressed data format. The encoder algorithm is deliberately left unspecified to stimulate competition and leave the door open to improvements — as long as any such improvement produces a compliant format compatible with the decoding algorithm. This standard is organized in several parts. Currently there are six parts, some of which are already frozen and some still in development. Part 1 defines the JPEG 2000 image-coding system; that is, the compressed data format and the decoding algorithm. All JPEG 2000-compliant software should implement Part 1 in its entirety so that data exchange works seamlessly. Additionally, Part 1 also specifies an optional file format called "JP2," which encapsulates the compressed data and provides additional information, such as accurate color definition and metadata.

Part 1 is intended to cover the majority of applications such as image archival, mobile Internet, the Web, and so on. In addition, Part 2 specifies extensions that are targeted to specific applications such as compression of hyperspectral data (for example, satellite images). Part 3 specifies Motion JPEG 2000 (a superset of Part 1 that provides the extensions necessary to the coding of video and animations), and Part 6 specifies a compound image-file format for prepress and fax-like applications. Other than decoding algorithms and compressed data formats, Part 4 specifies the rules for conformance to the JPEG 2000 standard, Part 5 provides reference software for Part 1, and Part 7 provides guidelines for minimum support (for example, for digital cameras). In the reference software, two implementations are provided in source-code form--one in C and the other in Java. The license is fairly liberal and the software can essentially be used freely in products claiming JPEG 2000 conformance. The Java source code already implements all the algorithms in Part 1 and is available to the public at <http://jj2000.epfl.ch>. The C implementation is currently provided only in executable form at <http://www.imagepower.com/>, but is slated to be released in source code soon.

While Part 1 is already finished and waiting official ISO/ITU-T approval, the other parts are being finalized and will be finished later this year. On the functionality side, JPEG 2000 provides new

The JPEG 2000 Image Coding Standard

Diego Santa Cruz, Touradj Ebrahimi, and Charilaos Christopoulos

(Reprinted From Dr. Dobb's Journal)

features as well as many that already exist in other formats. What is new is the degree at which all these features are integrated and the flexibility of use that is allowed.

To start, JPEG 2000 is a multiresolution format. The image is stored at several resolutions, without redundancy, and can be transmitted or decoded at a resolution suitable for the display device in use. It can also provide access to different qualities from the same file at any resolution, ranging from low quality to visually lossless quality, or even truly lossless. Thus, an image needs to be compressed only once at a high enough resolution and high enough quality (eventually lossless), and then it can be decoded in many possible ways. After encoding an image, the compressed file can also be reorganized to suit some new transmission order needs, with minimal work.

Region of interest (ROI) is one of JPEG 2000's new features. It allows you to encode an arbitrarily shaped region of the image with a higher quality than the rest and/or make it arrive earlier during a progressive transmission. For transmission in noisy environments, JPEG 2000 incorporates many error resilience strategies. In the event of a transmission error, the image quality is only partially degraded and successful decoding is possible. JPEG 2000 also provides random access to different image regions without the need to decode the rest. This is especially attractive for large images to provide efficient pan and zoom capabilities. In addition, it is also possible to flip or do simple rotations without having to decode and encode again. Thus, no loss of quality is incurred by such common operations. Another advantage of JPEG 2000 is that it is possible to control the compressed file size quite precisely without an iterative approach, which is not possible in JPEG. Also of importance to some applications is that JPEG 2000 is capable of coding multicomponent images in addition to color and gray-level ones. Last, but by no means least, JPEG 2000 provides improved compression over JPEG. As you'll see, the difference is particularly striking at high compression.

All these features are integrated in one uniform algorithm and can be used together in a very flexible way. Previous standards already support some of this functionality but they do so in a nonintegrated or inefficient manner. A notable example is JPEG, which defines many modes that provide different capabilities. However, the modes cannot be freely mixed and only some of them are supported by most of the software. For example, lossless and lossy coding cannot be mixed in JPEG.

The optional JP2 file format supports paletted color as well as true color, accurate color definition, multiple color spaces, opacity information (alpha channels, for instance), and metadata.

How Does it Work?

We'll now turn to how all this functionality is provided. Although it isn't specified by the Standard, we will examine the typical encoding algorithm since it is much easier to follow and understand. The decoding algorithm is obtained by reversing the process. For simplicity's sake, we'll skip many details including analysis of some of the more advanced functionality, such as region of interest coding or error resilience. Figure 1 shows the block diagram of a typical encoder.

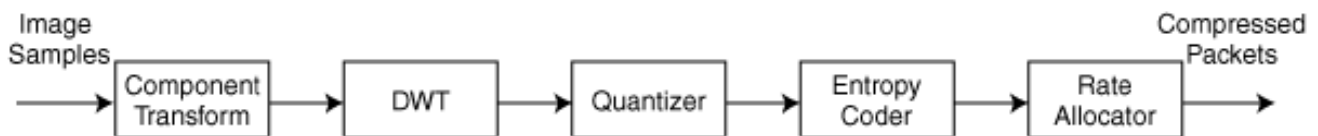


Figure 1
Block diagram of a typical JPEG 2000 encoder.

The JPEG 2000 Image Coding Standard

Diego Santa Cruz, Touradj Ebrahimi, and Charilaos Christopoulos

(Reprinted From Dr. Dobb's Journal)

JPEG 2000 is based on the discrete wavelet transform (DWT). The DWT transforms the spatial image data to a space-frequency representation, achieving energy compaction in the process. That is, most of the energy is concentrated in a few coefficients. The DWT in JPEG 2000 is, of course, two dimensional, but it is best explained by first introducing the one-dimensional DWT (1D DWT). Before examining what the DWT is all about, note that for color and multicomponent imagery, JPEG 2000 can optionally use a component transform prior to the DWT if the original image components are not suitable for compression. For example, an RGB image can be converted to YUV, which is more easily compressed. Part 1 of the Standard provides one irreversible (that is, subject to rounding errors) and one reversible component transform. Both of them are similar to RGB or YUV, but can be used on data that is not RGB. An external file format, such as JP2, can define an arbitrary color space for the input data, so JPEG 2000 Part 1 is by no means limited to RGB or YUV. Part 2 of the Standard defines other decorrelating transforms, which are suitable to multicomponent data, such as those generated by multiband sensors. After the component transform is applied, the rest of the encoding algorithm works on each component independently. The 1D DWT is performed by a filter pair that decomposes a signal into two: a low-frequency signal and a high-frequency one, which are referred to as "subbands." The former contains a blurry version of the original signal, and the latter the details.

Each of these signals has a sample rate that is half of the original, thus the total amount of samples remains unchanged. The original signal can be perfectly reconstructed from the two subbands. Typically, the DWT is performed using floating- or fixed-point arithmetic. This does not allow for a lossless transform due to rounding errors. However, it is also possible to perform it, with slight modifications, in integer-only arithmetic so that lossless reconstruction is possible. These inherent kinds of transform are referred to as "irreversible" and "reversible," respectively.

The 2D DWT in JPEG 2000 is performed by applying the 1D DWT first on the columns of the image, which results in two subbands, and then on the rows of these, which results in four subbands. Consequently, the horizontal and vertical low frequencies are contained in the LL subband, the horizontal high and vertical low in the HL, the horizontal low and vertical high in the LH and the horizontal and vertical high in the HH. This decomposition into four subbands constitutes one level of the 2D DWT. The process can be repeated several times on any of the four subbands, resulting in a tree structure.

Typical images contain most of their energy in the low frequencies, and thus the decomposition is generally repeated on the LL subband only, which is referred to as a "dyadic decomposition." Figure 2 illustrates a 2D DWT with two levels. The LL₂ subband will look like a downscaled version of the original image. At the decoder the process is reverted, and by reconstructing only some of the decomposition levels, a lower resolution version of the image can be obtained. Each of these resolutions is called a "resolution level." Typically, five decomposition levels are used, which results in six resolution levels, all related by a factor of two.

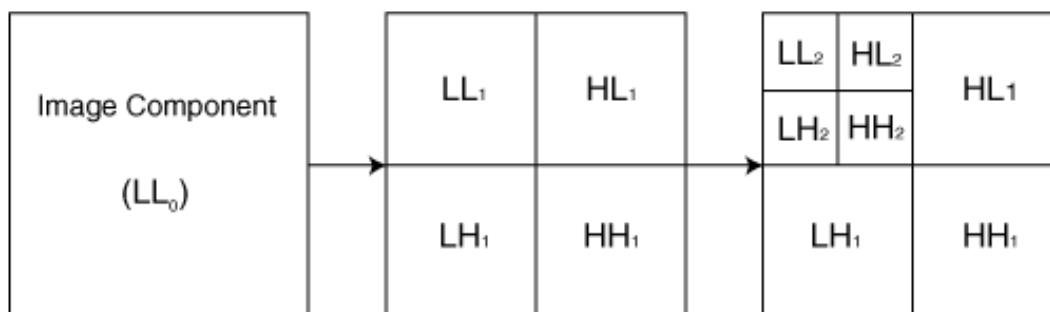


Figure 2

Example of a two-level discrete wavelet transform (DWT) using a dyadic decomposition.

The JPEG 2000 Image Coding Standard

Diego Santa Cruz, Touradj Ebrahimi, and Charilaos Christopoulos

(Reprinted From Dr. Dobb's Journal)

For simplicity, JPEG 2000 Part 1 supports only dyadic decompositions and provides two filter pairs. This fulfills the needs of most applications. Part 2 of the Standard allows for arbitrary decompositions and filter pairs. The ones provided by Part 1 are the 9/7 and the 5/3, the numbers referring to the filter lengths along the low- and high-pass channels. The 9/7 filter is irreversible and provides very good compression, but is not capable of delivering lossless reconstruction. The 5/3 filter is reversible with slightly inferior compression efficiency, but is capable of lossless coding, as well as lossy. Unlike the DCT in JPEG, the DWT in JPEG 2000 is not performed on small image blocks but on the entire image; thus, the well-known blocking artifact of JPEG is avoided. After the DWT coefficients have been calculated, they are quantized. This reduces the precision and sets many of the low-valued coefficients, which contribute little to image quality, to zero. The quantization used in JPEG 2000 is scalar and embedded. Figure 3 is the quantization function, where Δ is the quantization step, w the value to quantize, and $(Q)w$ is the index of the quantized value. The value Δ can be set independently for each subband. Because the quantizer is embedded, dropping the last bit of the quantized index $(Q)w$ (that is, integer division by two) is equivalent to doubling the quantization step size. This nice property will be used later. It is important to note that if lossless coding is desired, there is no quantization, or equivalently the quantization step size is set to 1.

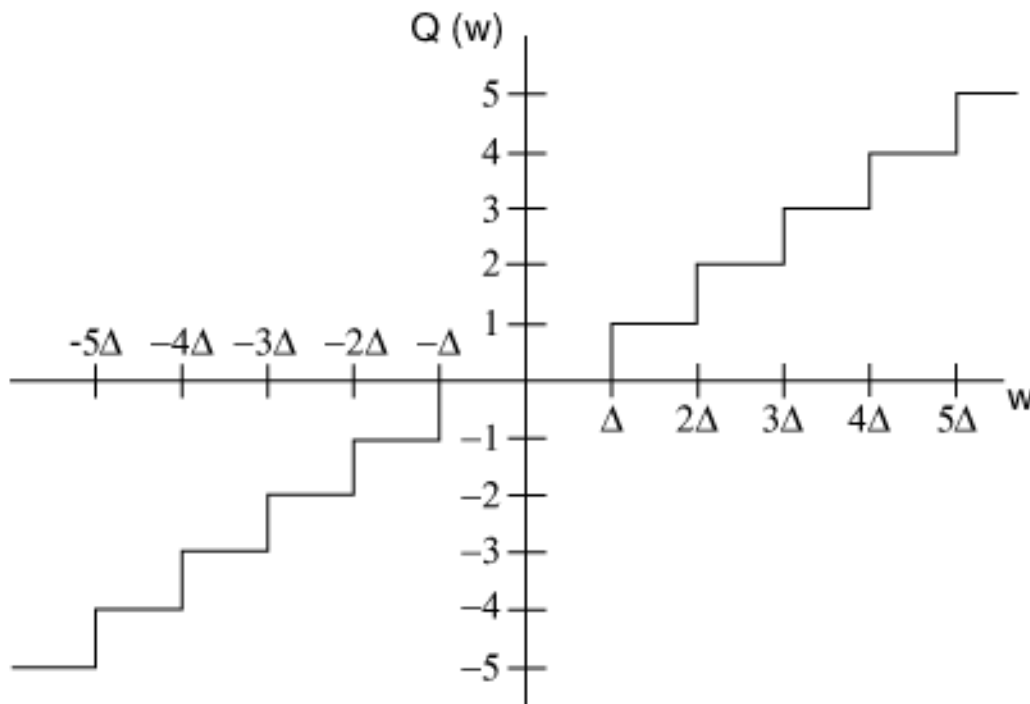


Figure 3
The quantizer function as used in JPEG 2000.

Once quantized, each subband is divided into rectangular regions called "code-blocks," which are typically of sizes 64x64 or 32x32. Each code-block is coded independently in a lossless manner using entropy coding. The entropy coder is a context-based bitplane arithmetic coder, where the bitplanes are encoded from the most to the least significant. Thus, truncating a code-block's compressed data discards some of the least significant bitplanes, which is equivalent to having used a larger quantization step size, since the quantizer is embedded. The arithmetic coder used is the MQ-coder, which is an adaptive binary arithmetic coder that does not require multiplications or divisions.

The JPEG 2000 Image Coding Standard

Diego Santa Cruz, Touradj Ebrahimi, and Charilaos Christopoulos

(Reprinted From Dr. Dobb's Journal)

After entropy coding, the compressed data of all the code-blocks is organized into layers by the rate allocator, where successive layers contain additional compressed data from some or all code-blocks. These layers can be thought of as successively increasing quality levels, although the definition of quality can be anything that is suitable for the application at hand. In most cases, it will be directly related to a distortion measure that is uniform across the image, but it can be a region dependent measure, or dependent on a particular user query, and so on. Decoding only some of the layers leads to truncated versions of each code-block's compressed data, resulting in a higher compression ratio with lower image quality. This data is then encoded in packets, where each packet contains the compressed data for one layer, one resolution level, and one component. Each packet contains a small header, which makes it possible to parse or skip its contents, thus providing the random access and parse features. Finally, the packets are output to a code stream along with a header containing the encoding parameters. They can be output from the lowest to the highest resolution, from the first layer (low quality) to the last one (high quality), from the top to the bottom of the image, from the first component to the last, or any mix of these, thus providing almost any kind of progression. Since the packets can be parsed, you can easily produce a new code stream from an already existing one, but with a different progression order, a subset of the layers, or a reduced resolution.

Comparative Performance

Comparing the performance of image coding systems is a tricky business. There are many variables that come into play such as compression ratio, image quality, encoding time, or processing memory. Changing some parameters might adversely affect one aspect but improve another. For this reason, we restrict ourselves to some basic comparisons against JPEG. The first thing that comes to mind is image quality. Figure 4 shows a scanned photograph encoded at high and very high compression ratios. From this example, it is clear that JPEG 2000 has a great advantage, particularly at very high compression ratios. At lower compression ratios, the difference is smaller and would not show in the printed version. These results might vary depending on the image, but in general, without special fine tuning, JPEG 2000 provides better quality than JPEG, from high to low compression ratios, while providing greater flexibility of use.

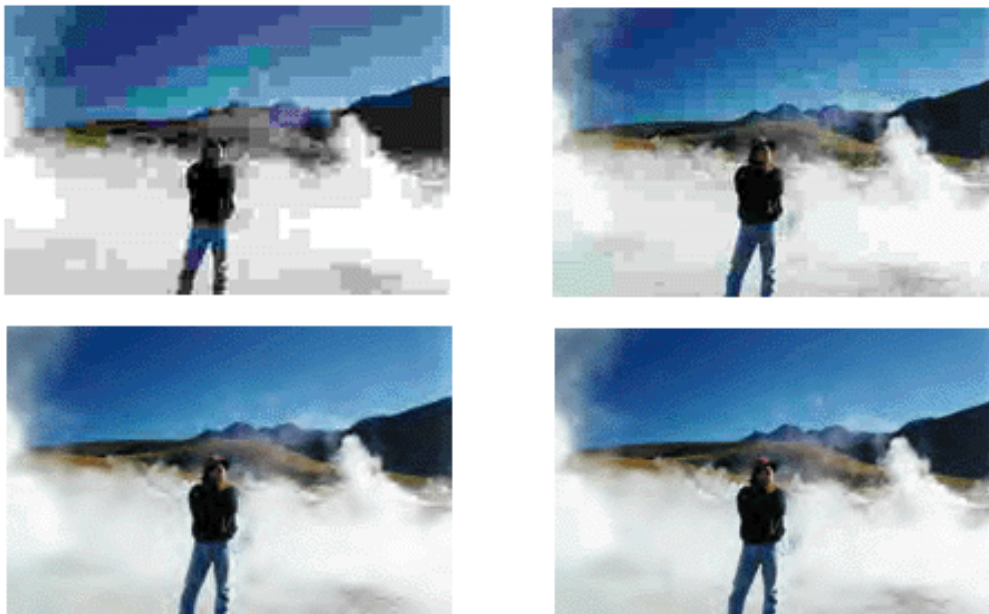


Figure 4

Comparison of JPEG and JPEG 2000 encoded images. The left and right columns have a compression ratio of 142 and 96 to 1, respectively. The top row shows JPEG, while the bottom shows JPEG 2000 encoded images. The image is 350 columns by 228 rows.

The JPEG 2000 Image Coding Standard

Diego Santa Cruz, Touradj Ebrahimi, and Charilaos Christopoulos

(Reprinted From Dr. Dobb's Journal)

Another important aspect for most users is the time needed to encode and decode an image. Table 1 shows the encoding and decoding times obtained on a 700-MHz Pentium III PC, using the Independent JPEG Group (IJG) JPEG software and the JPEG 2000 verification model (VM), which is JPEG 2000's development software. Both are written in C. You can see that JPEG 2000 takes, roughly speaking, between three and eight times more CPU cycles than JPEG. However, JPEG 2000 provides much more flexibility of use and an improved image quality. It is very likely that this time ratio will decrease somewhat when JPEG 2000 software, as optimized as the IJG code, becomes available. It is also important to note that when JPEG was introduced 10 years ago, computers were 100 times less powerful, according to Moore's Law. Although the comparison presented here is brief, you can see that we can expect an improvement in compression ratio for the same quality, but that it comes at the expense of additional computer resources. I think that this increase is acceptable for most general-purpose, computer-based applications. This situation is probably different for current embedded applications, but as computing power increases, it might become interesting.

	Encoding time (ms)	Decoding time (ms)
JPEG	55	20
JPEG 2000 reversible (9/7 filter)	158	133
JPEG 2000 irreversible (5/3 filter)	—	78

Table 1

JPEG and JPEG 2000 encoding and decoding times for the image in Figure 4 at a 96 to 1 compression ratio. For JPEG 2000, results with both DWT filters, reversible and irreversible, are reported. Meaningful encoding times for the reversible filter cannot be calculated with the JPEG 2000 VM software.

The File Format

As mentioned earlier, Part 1 of JPEG 2000 specifies an optional file format called JP2. This file format encapsulates the JPEG 2000 code stream and supplies additional important information about the image. The file format is based on the concept of boxes, where each box is a contiguous stream of data containing type and length information. Some boxes — the superboxes — can contain other boxes, leading to a hierarchical structure. The basic boxes provide file type identification, transmission error detection (7-bit e-mail, ASCII ftp transfers, and the like), image size, number of components, component bit-depth, as well as capture and default display resolutions. JP2 also provides two methods to accurately identify the image colorspace. The enumerated (by name) method requires that all applications know how to handle each one of the possible colorspace, which is impractical if there is a large number of them. This is why this method has been limited to sRGB and nonlinear gray level. For other colorspace, JP2 supports a restricted form of ICC profiles, which allows for a nonlinearity curve and a 3x3 transformation matrix.

In addition to accurate colorspace definition, JP2 allows the specification of a color palette, which can hold up to 1024 entries, and opacity information (that is, alpha channel). This file format also provides support for embedding vendor-specific information by the use of XML or UUID boxes, which can be interpreted by specific applications and safely ignored by others. Finally, there is also a box that is devoted to carrying intellectual property rights information.

Potential Applications

JPEG 2000 brings new features to the field of image storage and transmission, but what will be the applications that will use this new format in the near future? While this is difficult to predict and only time will tell, we briefly outline some applications that could benefit from the new features. The first thing that everybody thinks of is the Web. JPEG 2000 provides better compression than JPEG, but

The JPEG 2000 Image Coding Standard

Diego Santa Cruz, Touradj Ebrahimi, and Charilaos Christopoulos

(Reprinted From Dr. Dobb's Journal)

the difference is not extremely significant at the qualities that most people use. However, the image viewing experience and site management could greatly improve with the multiresolution, random access, and progressive features. For example, a web browser could first access a low-resolution version of an image for viewing on your monitor. If you decide to print it, then the browser automatically retrieves more information from the same file to have a higher resolution and produce a higher quality printout. And if you desire a top-quality (that is, lossless) image for editing, you could continue downloading the rest of the file. To exploit these features, plain HTML and HTTP 1.1 suffice.

The width and height attributes of the img HTML tag would tell the web client what resolution to download first, then HTTP 1.1 range requests would be used to get extra data needed to augment the resolution. It would need only one JPEG 2000 image file on the web server — instead of many files at different resolutions and qualities or dynamic generation — thus reducing web site management and the total amount of transmitted data. In a similar manner, different resolutions or qualities could be downloaded to different devices with different display characteristics (PDAs, cell phones, PCs, TVs, and the like).

In the dawn of the mobile Internet, JPEG 2000 enables a dynamic use of compressed images. For example, a WAP portal could transcode a JPEG 2000 image from an HTML page into a size and quality suitable for the device on which it is to be displayed (PDA, cell phone, and so on). One could also envisage associating different quality of services to different layers of a JPEG 2000 compressed file. In the case of network congestion, the layers would start to be dropped, thus reducing the image quality but not affecting decodability. In this way, image quality can be adjusted according to the available bandwidth at several locations in the network, not only at the original server. Remote viewing of large images with pan and zoom can also be greatly improved by making use of random access and multiresolution. We are sure that there are many more applications that could take advantage of JPEG 2000.

Conclusion

JPEG 2000 has the potential to take imaging a step further, in terms of flexibility of use and user experience. However, widespread adoption of JPEG 2000 is subject to two main issues — patents and software support.

One of the factors in the success of JPEG is that its widely used form is free from any patents. The JPEG committee is very much aware of this and strives to make JPEG 2000 Part 1 free of any patent royalties. To this effect, all contributors to JPEG 2000 have signed agreements by which they provide free use of their patented technology for JPEG 2000 Part 1 applications. During the standardization process, some technologies were even removed from Part 1 because of unclear implications in this regard. Although it is never possible to guarantee that no other company has some patent on some technology, even in the case of JPEG, unencumbered implementations of JPEG 2000 should be possible.

Software support is also key to the success of any image format. To this effect, two implementations are provided as reference software with liberal enough licenses. At this time, we are not aware of any commitment of major web browsers to provide JPEG 2000 support, but several companies have announced plans for delivering JPEG 2000 software. Let's hope that JPEG 2000 support will appear soon.

Acknowledgments

JPEG 2000 has been developed through collaborative effort by hundreds of people. Although it is not possible to thank everybody that participated, we would like to mention Daniel Lee, the JPEG convener; David Taubman, the author of the basis algorithm and initial experimental software; Michael

The JPEG 2000 Image Coding Standard

Diego Santa Cruz, Touradj Ebrahimi, and Charilaos Christopoulos

(Reprinted From Dr. Dobb's Journal)

Marcellin for important contributions to the algorithm; Tom Flohr, maintainer of the experimental software; Martin Boliek and Eric Majani, editors of the Standard; Majid Rabbani, technical lead; Bernie Brower, for leading the compliance definition; Faouzi Kossentini, for leading the reference software; Raphaël Grosbois, Joel Askelöf, and David Bouchard, authors of the Java reference software; and Michael Adams, author of the C reference software.

Resources

JPEG 2000 drafts: <http://www.jpeg.org/CDs15444.htm>.

Official JPEG 2000 web page: <http://www.jpeg.org/JPEG2000.htm>.

Jawerth, B. and W. Sweldens. "An Overview of Wavelet based multiresolution analyses," SIAM Rev., vol. 36, no. 3. September 1994. <http://cm.bell-labs.com/who/wim/papers/overview.pdf>.

Santa-Cruz, D., T. Ebrahimi, J. Askelöf, M. Larsson, C. Christopoulos. "JPEG 2000 Still Image Coding versus Other Standards." Proceedings of SPIE, vol. 4115. October 2000.

Taubman, D. "High Performance Scalable Image Compression with EBCOT." IEEE Transactions on Image Processing, vol. 9, no.7. July 2000.