CREW:
Compression with Reversible Embedded Wavelets

Ahmad Zandi James D. Allen Edward L. Schwartz Martin Boliek

RICOH California Research Center
2882 Sand Hill Road, Suite 115, Menlo Park, CA 94025
e-mail: zandi@crc.ricoh.com

ABSTRACT

Compression with Reversible Embedded Wavelets (CREW) is a unified lossless and lossy continuous-tone still image compression system. It is wavelet-based using a "reversible" approximation of one of the best wavelet filters. Reversible wavelets are linear filters with non-linear rounding which implement exact-reconstruction systems with minimal precision integer arithmetic. Wavelet coefficients are encoded in a bit-significance embedded order, allowing lossy compression by simply truncating the compressed data. For coding of coefficients, CREW uses a method similar to Shapiro's zerotree, and a completely novel method called Horizon. Horizon coding is a context based coding that takes advantage of the spatial and spectral information available in the wavelet domain. CREW provides state of the art lossless compression of medical images (greater than 8 bits deep), and lossy and lossless compression of 8-bit deep images with a single system. CREW has reasonable software and hardware implementations.

1 Introduction

Since CREW uses a "reversible" approximation of one of the best known wavelet filters, its performance is equal to or better than other existing methods in both lossy and lossless modes. It encodes the wavelet coefficients in a bit-significance embedded order similar to Shapiro [Sha93], and a completely novel method called *Horizon*. Horizon coding is a context-based coding that takes advantage of the spatial and spectral information available in the wavelet domain. While Zerotree is rightfully considered to be one of the best encoding methods of the wavelet coefficients, it is not efficient when it reaches the lesser significant bits of the coefficients. This usually is tolerable in a lossy system, but for lossless compression the encoding of the less significant bits are of prime importance. Horizon coding, which is particularly useful for the lesser significant bits is general and powerful enough to be used alone.

The original motivation for this system is the compression of medical images, although the same feature set could be very useful for other applications such as pre-press images, satellite images, document processing, world wide web, and other communication systems. At this time, for various reasons, medical image compression is considered to belong to the lossless realm. However, there are definitely future possibilities for the use of lossy compression. Perhaps the image is kept in a lossless compressed form prior to the diagnosis and archived for permanent records using lossy compression. For such a scheme, it is very desirable to have a single system

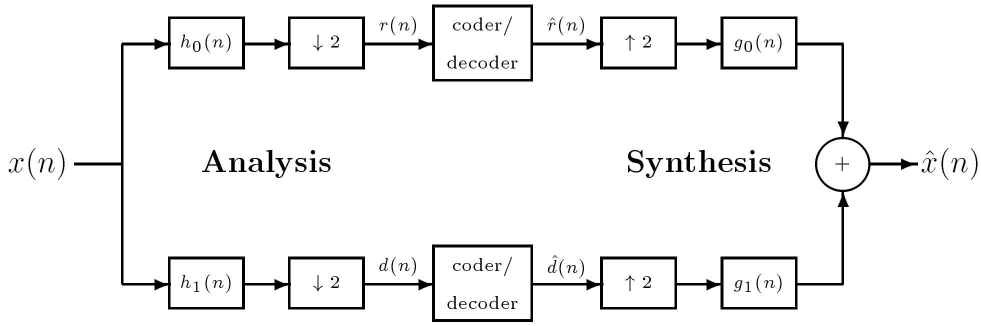

Figure 2.1: Block Diagram of a Wavelet Analysis/Synthesis System

that can perform both lossy and lossless compression. Moreover, with an *embedded* compression system, i.e., compressed data is in a visually important order, lossy compression can be performed by a simple truncation of the compressed bit stream.

In section 2 the basics of wavelet decomposition are explained, and reversible wavelets are defined. In section 3, two reversible wavelet transforms are described in detail. The first one, the S-transform is used to make the definitions easy to comprehend, while the second, the RTS-transform, is the suggested transform. Section 4 describes the embedded entropy coding, including a brief description of Shapiro's Zerotree, and Horizon context model. In section 5, the implementation of CREW in software and hardware is discussed. Section 6 contains some experimental results, comparing the performance of CREW with other system.

2 Wavelet Decomposition of Digital Signals

A wavelet transform[1] is defined by a pair of FIR analysis filters $h_0(n), h_1(n)$, and synthesis filters $g_0(n), g_1(n)$. The filters h_0 and g_0 are the low-pass and h_1 and g_1 are the high-pass. For an input signal, $x(n)$, the filters h_0 and h_1 are applied and the results are decimated by 2 (critically subsampled) to generate the transform signals $r(n)$ and $d(n)$, so-called the *reference* and the *detail* signals (analysis part of Figure 2.1). In the synthesis part the transformed signals are upsampled by 2 (a zero is inserted after every term) and then passed through the synthesis filters. Coefficients of the reference signal $r(n)$ are processed through the low-pass synthesis g_0 and the coefficients of the detail signal $d(n)$ through the high-pass synthesis filter g_1 (synthesis part of Figure 2.1). In this paper we are only interested in *quadrature mirror filters*[2], i.e., the synthesis filters are defined in terms of the analysis filters as follows:

$$\begin{cases} g_0(n) &= (-1)^n h_1(n) \\ g_1(n) &= -(-1)^n h_0(n). \end{cases}$$

The coder/decoder blocks contain all the processing in the transformed domain, e.g., quantization, coding etc. The filters can be recursively applied to the reference and detail signal. Of special interest are the *pyramidal* systems, in which the filters are

[1]For basic wavelet transformation we adopt the terminology and notations of [VBL94].

[2]For details and extensive references on QMF systems cf. [SA90]

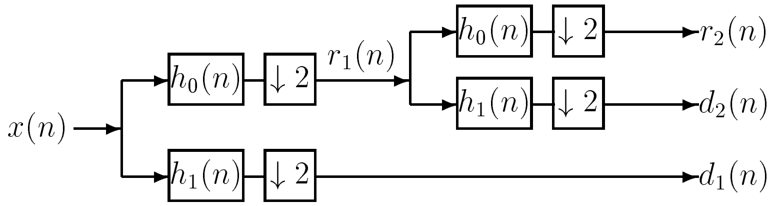

Figure 2.2: Block Diagram of a two-level Pyramidal Transform

recursively applied only to the reference signal. Figure 2.2 shows the block diagram of a two-level (the filters applied twice) pyramidal system.

Definition 1: Exact Reconstruction Systems: The system in Figure 2.1 is called exact reconstruction if the signals, $x(n)$ and $\hat{x}(n)$ are identical up to a multiplicative constant and a delay term [LGT88].

Definition 2: Efficient Reversible Systems: A reversible system is an implementation of an exact-reconstruction system, in integer arithmetic, so that a signal with integer coefficients can be losslessly recovered. An efficient reversible system is a reversible system with transform matrix[3] of determinant $\approx \pm 1$.

The construction of reversible transforms is by no means difficult. Given enough precision, any exact reconstruction transform can be made reversible. The challenge is to construct *"efficient"* reversible transforms. Intuitively, efficient means that a reasonable and practical entropy coder can efficiently encode the coefficients. While efficiency criteria might be specific to a particular encoder, in general, systems with determinant $= \pm 1$ are efficient. Like any transform which is used for coding, good energy compaction is also a primary factor. The goal is to design a single system which performs well in both lossy and lossless modes. The concepts will be made more clear in the next section with two examples.

3 Two Examples of Reversible Transforms

3.1 Exact Reconstruction

Example 1: Hadamard Transform: In normalized form it has the following filter coefficients:

$$\begin{cases} h_0 = \frac{1}{\sqrt{2}}(1,1) \\ h_1 = \frac{1}{\sqrt{2}}(1,-1), \end{cases} \tag{3.1}$$

which is clearly an exact-reconstruction transform.

Example 2: TS-transform (Two-Six-transform): For the origin and qualities of the TS-transform cf. Remark 1 below. TS-transform is defined by the following filter coefficients:

$$\begin{cases} h_0 = \frac{1}{\sqrt{2}}(1,1) \\ h_1 = \frac{1}{8\sqrt{2}}(-1,-1,8,-8,1,1). \end{cases} \tag{3.2}$$

This is also an exact-reconstruction transform.

[3]For the definition of the transform matrix cf. [SA90].

3.2 Basic Reversible Versions

The reversible transforms, in general, are non-linear. Hence they will be defined as expressions. However, the linear system approximation which is useful for evaluation will also be given. Recall that for the input signal $x(n)$, $r(n)$ and $d(n)$ are the reference and the detail signal respectively. The reversible S-transform will be used as a simple example before explaining the reversible TS-transform.

Example 3: S-transform: An efficient reversible version of the Hadamard transform, Example 3, known as the S-transform [SAAJ91, SP93] is defined as follows:

$$\left\{ \begin{array}{rcl} r(0) & = & \lfloor \frac{x(0)+x(1)}{2} \rfloor \\ d(0) & = & x(0) - x(1). \end{array} \right. \tag{3.3}$$

Notice that this is an approximation to the linear transform (with determinant $= -1$.) The floor function in the definition of $r(0)$ is the source of non-linearity.

$$\left\{ \begin{array}{rcl} r(0) & = & \frac{x(0)+x(1)}{2} \\ d(0) & = & x(0) - x(1). \end{array} \right. \tag{3.4}$$

A constructive proof for the reversibility of the S-transform is the inverse transform:

$$\left\{ \begin{array}{rcl} x(0) & = & r(0) + \lfloor \frac{d(0)+1}{2} \rfloor \\ x(1) & = & r(0) - \lfloor \frac{d(0)}{2} \rfloor. \end{array} \right. \tag{3.5}$$

The idea behind the reversibility of the S-transform is the observation of two facts. One, knowledge of the sum and the difference of two integers are sufficient to recover the numbers. Two, the sum and the difference have the same parity, i.e., they share the same least significant bit. Hence the integer division by 2 (or a shift right by 1) in Eq. 3.3, eliminates a redundant least significant bit. The S-transform, where redundant information can be detected and easily eliminated, is an example of efficient reversible transform.

Example 4: RTS-Transform (Reversible TS-transform): An efficient reversible version of the TS-transform, which we call RTS-transform is defined as follows:

$$\left\{ \begin{array}{rcl} r(0) & = & \lfloor \frac{x(0)+x(1)}{2} \rfloor \\ r(1) & = & \lfloor \frac{x(2)+x(3)}{2} \rfloor \\ r(2) & = & \lfloor \frac{x(4)+x(5)}{2} \rfloor \\ \vdots & & \\ d(0) & = & \lfloor \frac{-\lfloor \frac{x(0)+x(1)}{2} \rfloor + 4(x(2)-x(3)) + \lfloor \frac{x(4)+x(5)}{2} \rfloor}{4} \rfloor \\ \vdots & & \end{array} \right. \tag{3.6}$$

Notice that this is an approximation to the following linear version (with determinant $=$ -1) of the TS-transform:

$$
\begin{cases}
r(0) &= \frac{x(0)+x(1)}{2} \\
r(1) &= \frac{x(2)+x(3)}{2} \\
r(2) &= \frac{x(4)+x(5)}{2} \\
\vdots & \\
d(0) &= \frac{-x(0)-x(1)+8(x(2)-x(3))+x(4)+x(5)}{8} \\
\vdots &
\end{cases}
\tag{3.7}
$$

The proof that the RTS-transform is reversible is quite simple. We show how to recover $x(2)$ and $x(3)$ from $r(0), r(1), r(2)$ and $d(0)$, the recovery of other samples are similar. Notice the expression for $d(0)$ in Eq. 3.6, can be written as,

$$d(0) = \lfloor \frac{-r(0) + 4(x(2) - x(3)) + r(2)}{4} \rfloor.$$

From this it follows that:

$$x(2) - x(3) = d(0) - \lfloor (-r(0) + r(2))/4 \rfloor.$$

Hence $x(2) - x(3)$ is completely known. This, combined with $r(1) = \lfloor (x(2) + x(3))/2 \rfloor$ in Eq. 3.6, and the use of inverse S-transform Eq. 3.5, leads to the recovery of $x(2)$ and $x(3)$.

Remark 1: Le Gall and Tabatabai in [LGT88] use a design procedure based on the factorization of a product filter into two linear phase low-pass components. These correspond to the low-pass analysis and synthesis filters. By using the quadrature mirror properties the high-pass filters are derived. In their most important example, which is by now classical, the following product filter is factored:

$$P(Z) = 1/16(1 + Z^{-1})^3(-1 + 3Z^{-1} + 3Z^{-2} - Z^{-3}).$$

Two factorizations are given in [LGT88],

$$
\begin{cases}
P(Z) &= [1/4(1 + Z^{-1})^3] \times [1/4(-1 + 3Z^{-1} + 3Z^{-2} - Z^{-3})] \\
P(Z) &= [1/2(1 + Z^{-1})^2] \times [1/8(1 + Z^{-1})(-1 + 3Z^{-1} + 3Z^{-2} - Z^{-3})].
\end{cases}
\tag{3.8}
$$

A third factorization,

$$P(Z) = [1/2(1 + Z^{-1})] \times [1/8(1 + Z^{-1})^2(-1 + 3Z^{-1} + 3Z^{-2} - Z^{-3})],$$

which was not mentioned in that paper, is, in fact, the rational version of the TS-transform (Eq. 3.7.) Speck in [Spe93] considers and analyses this third factorization. In [VBL94] the normalized version Eq. 3.2, is evaluated together with several thousand other wavelet transforms, and is rated as one of the overall best.

Remark 2: In both the S-transform and the RTS-transform, the reference signal $r(n)$ has the same range of values as the input signal $x(n)$, e.g., if the range of $x(n)$ is from 0 to 255 the same is true about $r(n)$. This property is especially important in a pyramidal system where the reference signal is successively decomposed.

Remark 3: Both normalized Hadamard transform and the normalized TS-transform have especially simple implementations in two dimensions, the domain of digital images. Recall that in the separable two dimensional wavelet transform the image is decomposed into four blocks, the so-called LL, LH, HL, and HH [Sha93]. Each L corresponds to an application of the low-pass filter h_0, and each H corresponds to an application of the high-pass filter h_1. If we denote h_0, and h_1 to be the low-pass and the high-pass filters of the normalized TS-transform, Eq. 3.2, and similarly h_0^r, and h_1^r for the rational version, Eq. 3.7, then

$$\begin{cases} h_0^r &= \frac{\sqrt{2}}{2} h_0 \\ h_1^r &= \sqrt{2} h_1. \end{cases}$$

Therefore the LL, and HH components of the rational TS-transform, Eq. 3.7 are 1/2 and 2 times the corresponding components of the normalized TS-transform. Moreover the components LH, HL are identical. This leads to an efficient implementation of the TS-transform through the rational TS-transform. More to the point for this article is the fact that if the rational TS-transform, Eq. 3.7, is replaced by the the reversible TS-transform (RTS-transform), Eq. 3.6, a very good approximation of the normalized TS-transform is realized, which in addition is reversible. Notice that the scale factors of 1/2 and 2 are especially easy to implement.

4 Embedded Entropy Coding

In most transform-based compression systems the coefficients are entropy coded. Of special importance for us are the so-called "embedded" coders. Briefly, an embedded coding is a system in which the coded bit stream is ordered by visual significance or, more accurately, ordered with respect to some error metric cf. also [Sha93]. The embedded order used in this paper is bit-significance in the transform domain, the same as used in [Sha93].

The Zerotree [Sha93] is an efficient embedded coding method of the wavelet coefficients, which takes advantage of the inherent similarity of different bands in the transform domain.

Horizon embedded coding, introduced below, is a spatial-spectral context model which uses the same embedding order, i.e., bit-significance, as the Zerotree. Non-embedded context models have been proposed to encode signed integers which take advantage of spatial correlations of coefficients [Lan91]. In Horizon coding the high correlation of neighboring pixels, in addition to the similarities of different bands, are utilized by context dependent entropy coding. The Horizon context dependent coding is especially attractive for encoding the low order bits, which must be encoded in a lossless system. Moreover the coding can start with the Zerotree, or some other spectral context model, and change to Horizon after any number of bit-planes.

4.1 Zerotree

The most important part of the Zerotree embedded coding of the wavelet coefficients is a prediction method which, according to Shapiro, is based on "the basic hypothesis - if a coefficient at a coarse scale is insignificant with respect to a threshold then all of its descendants are also insignificant." The descendants are defined with respect to a tree structure defined on the wavelet coefficients, which takes advantage of the similarity of the bands at different resolutions [Sha93]. The other part of the Zerotree embedded coding is a bit-significance embedding method to encode signed integers. In the so-called *dominant pass* the integers in sign-magnitude form, are encoded one bit at a time from the MSB to LSB, by the prediction method of Zerotree, until the first "on" bit is detected. The sign is encoded at this time, which is the logical embedding order of the sign bit. The remaining of the bits are encoded without the Zerotree prediction, in the so-called *subordinate pass*.

4.2 Horizon: A Spatial-Spectral Context Model

Horizon context model addresses the bit-significance embedded encoding of the wavelet coefficients by a binary entropy coder. There are three basic contexts designed for embedded encoding of signed integers. Hence these can be described independently of any wavelet system. Recall that in the bit-significance embedded coding of signed integers, the sign bit is encoded with the first "on" bit (starting from the MSB). Therefore, prior to the occurrence of the sign bit, the set of events consists of 0,1,-1. After the sign bit is encoded the set of events is 0,1. The first context is used to encode the event of zero bit vs. non-zero bit when the sign bit has not yet been encoded. The second context is used to encode the sign bit if the previous event was non-zero. Context three is used to encode the zero bit vs. the one bit if the sign bit is already encoded. In the terminology of Shapiro the first two contexts are used during the dominant pass and the third context during the subordinate pass. Notice also that the second context is used at most once for every integer.

The basic contexts of the Horizon model can be extended for the coding of the wavelet coefficients. Briefly, the fact that the wavelet transform is localized both in space and frequency makes it possible to use regional contexts in the same band as well as between bands contexts. The regional contexts in the same band are similar to the JBIG or the lossless JPEG systems. The between band spectral contexts can use modeling such as the tree structure used in Zerotree, or use the so-called similar coefficients from the other bands, provided that the system stays causal.

5 Computation in Software and Hardware

CREW is suitable for implementation in both software and hardware. It can be performed in two passes where the first pass generates all the transform coefficients and the second pass embeds and encodes them, or it can be performed in a unique one pass mode with memory management.

The filters chosen in CREW are easy to implement for both encoding and decoding. The implementation of the forward transform is immediate from Eq. 3.6. Figure 5.1 shows a hardware implementation of the inverse transform (the more interesting case). As in the forward case, four additions/subtractions are required. A total of four multiplications/divisions are hardwired shifts in hardware and three shift instructions

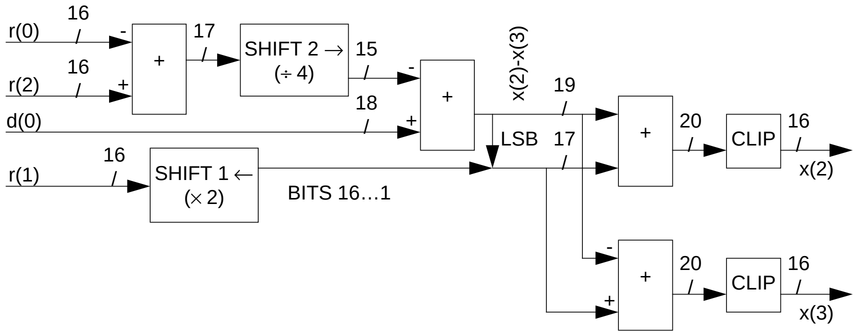

Figure 5.1: An Implementation of the Inverse RTS-Transform

in software. After $2 \times r(1)$ is computed (bits 16...1), its least significant bit (LSB) is taken from the computed value of x(2)-x(3). This operation has zero gate cost in hardware and is two logic operations in software.

For lossless decompression, the clip operation in Figure 5.1 simply shifts its input right by one (divides by two) and may drop (or otherwise ignore) the three most significant bits. In the lossy case, where quantization can cause the reconstructed value to be out of range, the three most significant bits must be checked and out of range results must be changed to the minimum or maximum allowable value.

For images where a full frame can fit in memory allowing two pass implementation, memory/data flow management is not a difficult issue. Even for 1024×1024 16 bit medical images (2 Mbytes in size), requiring a full frame buffer is probably reasonable. However larger images (for example A4, 400 DPI 4-color images are about 50 Mbytes), performing the wavelet transform with a limited amount of line buffer memory is desirable. A one pass method reduces the memory required by about a factor of 100 compared to using a full frame buffer for this example.

Because only the high pass filter is overlapped, the largest filter support region is defined by a cascade of low pass filters followed by a high pass filter. For a four level decomposition, the largest support region is $(2^3 \times 6) \times (2^3 \times 6) = 48 \times 48$ pixels, as shown in Figure 5.2. Note that for computational efficiency, redundant calculations due to overlap are done only once. Thus, only 16×16 new pixels are used in calculations for each region.

At the moment CREW uses the binary adaptive arithmetic coder known as Q-coder.

6 Experimental Results

Two sets of images are used for experimentation, a class of 512×512 USC gray scale 8-bit deep images, and a class of medical images of different modalities. Medical images “cr”, “dsa”, “xray” are 1024×1024, and are 10 bits deep. Images “ct” and “mri” are 512×512 and are 12 bits deep. Tables 1 and 2 are the lossless results, for the medical and the USC images respectively. The results related to the USC images (Table 2) are compared with JPEG lossless (with QM-coder), and bit-plane JBIG of the gray coded image [AT94]. The results on medical imaging (Table 1) are compared with DPCM which uses three neighbor pixels for prediction and Huffman

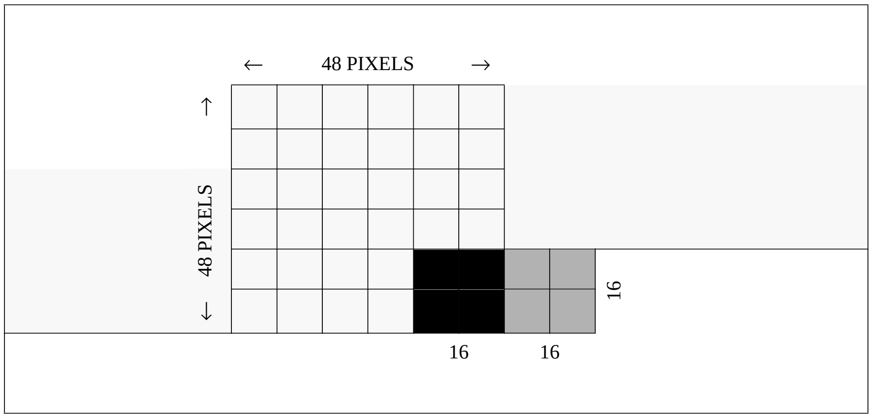

Figure 5.2: Image and Coefficients in line Buffer; 4-level pyramid

codes the prediction errors. Table 3 is the results of lossy compression of the USC images. The MSE results are compared with JPEG lossy with arithmetic coding at the same compression ratio. JPEG lossy with arithmetic coding was chosen because of its superior rate/distortion over baseline JPEG. In each case JPEG is used roughly at low, and high compression ratio. The truncation of CREW's compressed bit stream was used to decompress at exactly the same ratio. As can be observed state of the art lossy and lossless performance is achieved for all cases.

7 Conclusion

CREW is essentially an exact reconstruction transform scheme with good energy compaction. Moreover, it is specially designed so that the transform coefficients have small redundancy and are easy to entropy code. Since CREW has good compression efficiency at any level of quantization it is well suited for embedded coding. Embedded coding of the coefficients makes the quantization level a function of the length of the coded stream. Hence, quantization is performed with truncation. Without truncation the image is recovered losslessly.

Horizon context dependent entropy coding, together with the Shapiro's Zerotree is used to achieve such a system. In addition to the lossless and lossy state-of-the-art compression performance with a single system, efficient software and hardware implementations are also possible. Other features of CREW are multi-resolution and progressive capabilities.

TABLE 1
Compression ratio of lossless compression of Medical images.

compression method	cr	ct	dsa	mri	xray
CREW	2.43	5.26	2.89	3.23	2.58
DPCM	2.34	3.95	2.64	2.86	2.41
JBIG	2.25	4.92	2.72	2.68	2.46

TABLE 2
Compression ratio of lossless compression of USC images.

compression method	couple	crowd	lax	lena	man	woman1	woman2	average
CREW	1.63	1.88	1.34	1.84	1.69	1.66	2.37	1.73
JPEG	1.54	1.87	1.31	1.72	1.64	1.58	2.28	1.66
JBIG	1.53	1.75	1.31	1.69	1.59	1.58	2.10	1.62

TABLE 3
Mean Square Error (MSE) of lossy compression of USC images.

compression method		couple	crowd	lax	lena	man	woman1	woman2
low ratio	CREW	23.69	17.29	54.72	14.69	22.20	22.41	6.59
	JPEG	29.62	20.10	87.80	17.08	29.98	33.08	6.55
high ratio	CREW	42.17	30.65	99.70	21.15	40.08	38.30	9.73
	JPEG	49.86	36.10	137.00	27.74	48.84	52.29	11.32

References

[AT94] R. Arps and T. Truong. Comparison of international standards for lossless still image compression. *Proceedings of the IEEE*, 82(6):889–899, June 1994.

[Lan91] G. Langdon. Sunset: a hardware algorithm for lossless compression of gray scale images. In *Medical Imaging V: Image Capture, Formatting, and Display*, volume 1444, pages 272–282. SPIE, March 1991.

[LGT88] D. Le Gall and A. Tabatabai. Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques. In *International Conference on Acoustics, Speach and Signal Processing*, pages 761–765, New York, 1988. IEEE.

[SA90] E. Simoncelli and E. Adelson. Subband Transforms. In J. Woods, editor, *Subband Coding*, chapter 4. Kluwer Academic Publishers, Norwell, Massachusetts, 1990.

[SAAJ91] I. Shah, O. Akiwumi-Assani, and B. Johnson. A chip set for lossless image compression. *IEEE Journal Solid-State Circuits*, 26(3):237–244, 1991.

[Sha93] J. Shapiro. An embedded hierarchical image coder using zerotrees of wavelet coefficients. *Proc. IEEE Data Compression Conference*, pages 214–223, 1993.

[SP93] A. Said and W. Pearlman. Reversible image compression via multiresolution representation and predictive coding. In *Visual Communications and Image Processing*, volume 2094, pages 664–674. SPIE, November 1993.

[Spe93] D. Speck. Low-complexity subband coding for image compression. Technical Report UCSC-CRL-94-03, Computer Research Laboratory, University of California at Santa Cruz, May 1993.

[VBL94] J. Villasenor, B. Belzer, and J. Liao. Filter evaluation and selection in wavelet image compression. In *Data Compression Conference*, Snowbird, Utah, 1994. IEEE.