A Lossy Image Codec Based on Index Coding*

(Extended Abstract)

Jun Tian and Raymond O. Wells, Jr.
Computational Mathematics Laboratory
Rice University
Houston, Texas 77251-1892
USA
juntian@rice.edu, wells@rice.edu

September 28, 1995

*in *Proc. IEEE DCC*, Snowbird, Utah, 1996

Abstract

In this paper we propose a new lossy image codec based on index coding. Both J. Shapiro's embedded zerotree wavelet algorithm, and A. Said and W. A. Pearlman's codetree algorithm use spatial orientation tree structures to implicitly locate the significant wavelet transform coefficients. Here a direct approach to find the positions of these significant coefficients is presented. The new algorithm combines the discrete wavelet transform, differential coding, variable-length coding of integers, ordered bit plane transmission, and adaptive arithmetic coding. The encoding can be stopped at any point, which allows a target rate or distortion metric to be met exactly. The bits in the bit stream are generated in the order of importance, yielding a fully embedded code to successively approximate the original image source; thus it's well suited for progressive image transmission. The decoder can also terminate the decoding at any point, and produce a lower (bit) rate reconstruction image. Our algorithm is very simple in its form (which will make the encoding and decoding very fast), requires no training of any kind or prior knowledge of image sources, and has a clear geometric structure. The image coding results of it are quite competitive with almost all previous reported image compression algorithms on standard test images.

Contents

List of Figures

1 Introduction

In this paper, a new lossy image codec based on index coding is presented. It contains the following features:

- A discrete wavelet transform which removes the spatial and spectral redundancies of digital images quite well.
- Index coding which represents the positions of significant wavelet transform coefficients very efficiently.
- Ordered bit plane transmission which provides a successive approximation of image sources and facilitates progressive image transmission.
- Adaptive arithmetic coding which requires no training of any kind or prior knowledge of image sources.

2 Index Coding

The following problem is considered, for a finite index set $\mathcal{I} = \{i : i \in \mathbf{N}, i \leq M\}$, where M is a positive number, assume $\mathcal{S}$ is a subset of $\mathcal{I}$, how to code $\mathcal{S}$ efficiently?

Our solution is to combine two ideas, differential coding [3], and variable-length coding of integers [2]. Suppose that the elements of $\mathcal{S}$ are ordered in a monotonically increasing order. Taking the difference of two adjacent elements will produce the *difference set* $\mathcal{S}'$ of $\mathcal{S}$. For example, if

$$\mathcal{S} = \{1, 2, 5, 36, 42\}, \tag{1}$$

then

$$\mathcal{S}' = \{1, 1, 3, 31, 6\}. \tag{2}$$

And it is straightforward to get $\mathcal{S}$ from the difference set $\mathcal{S}'$ by taking the partial sums of $\mathcal{S}'$. Thus $\mathcal{S}$ and $\mathcal{S}'$ contain the same information. Can the difference set $\mathcal{S}'$ be further reduced? P. Elias described several variable-length representations of integers in [2], among which the coding $C_{\hat{\beta}}$ has the shortest length, and we will choose $C_{\hat{\beta}}$ to code $\mathcal{S}'$. By definition, $C_{\hat{\beta}}$ is the binary coding of an integer, with the most significant bit removed. For example, $C_{\hat{\beta}}(19_{10}) = C_{\hat{\beta}}(10011_2) = 0011$, where the subindex 10 means the decimal representation, and the subindex 2 means the binary representation. The $C_{\hat{\beta}}$ coding result of the difference set $\mathcal{S}'$ will be called the *reduced set* $\mathcal{S}''$ of $\mathcal{S}$. Thus for $\mathcal{S}$, $\mathcal{S}'$ in (1) and (2), we have

$$\mathcal{S}'' = \{, , 1, 1111, 10\}. \tag{3}$$

Note that there are no coded symbols before the first two commas "," in $\mathcal{S}''$. Since all elements in $\mathcal{S}'$ are positive integers ($\mathcal{S}$ is monotonically increasing), the inverse operation of $C_{\hat{\beta}}$ will be adding a "1" as the most significant bit in the binary representation. Applying this inverse operation on $\mathcal{S}''$ will produce $\mathcal{S}'$. Thus $\mathcal{S}''$ and $\mathcal{S}'$ contain the same information. It follows that $\mathcal{S}$ and the reduced set $\mathcal{S}''$ also contain the same information. So we will use $\mathcal{S}''$ to represent $\mathcal{S}$. In practice, one will need some end of message symbol to separate different elements when the reduced set $\mathcal{S}''$ is used, like the comma "," in (3).

3 Outline of the Algorithm

After taking the discrete wavelet transform of an image, all wavelet transform coefficients will be ordered in such a way that the coefficients at coarser scale will come before the coefficients at finer scale. A typical scanning pattern [5] is indicated in Figure 1. In a wavelet decomposition domain with N scales, the scan begins at LL_N, then HL_N, LH_N, HH_N, at which point it moves on to scale $N-1$, etc. And a wavelet transform coefficient x is defined as *significant* with respect to a threshold T if $|x| \geq T$, otherwise x is said to be *insignificant*.

The wavelet transform coefficients are stored in three ordered sets, the set of significant coefficients (**SCS**), the temporary set containing significant coefficients found in a given round (**TPS**), and the set of insignificant coefficients (**ICS**). The initial **ICS** contains all the wavelet transform coefficients with the order shown in Figure 1, while **SCS** and **TPS** are empty. And the initial threshold T is chosen such that $|x_j| < 2T$ for all the wavelet transform coefficients x_j, and for some j_0, $|x_{j_0}| \geq T$. Output the initial threshold T.

First we have a sorting pass. In the sorting pass, all significant coefficients in **ICS** with respect to T will be moved out and put into **TPS**. Let $\mathcal{S}$ be the indices (in **ICS**) of these significant coefficients. Output the reduced set $\mathcal{S}''$ of $\mathcal{S}$. Instead of using "," as the end of message symbol to separate different elements in $\mathcal{S}''$, we will take the signs (either "+" or "−") of these significant coefficients as the end of message symbol. For example, if $\mathcal{S} = \{1, 2, 5, 36, 42\}$ as in (1), and the signs of these four significant coefficients are "$+-++-$", then the encoding output $\mathcal{S}''$ will be "$+-1+1111+10-$". Then update the

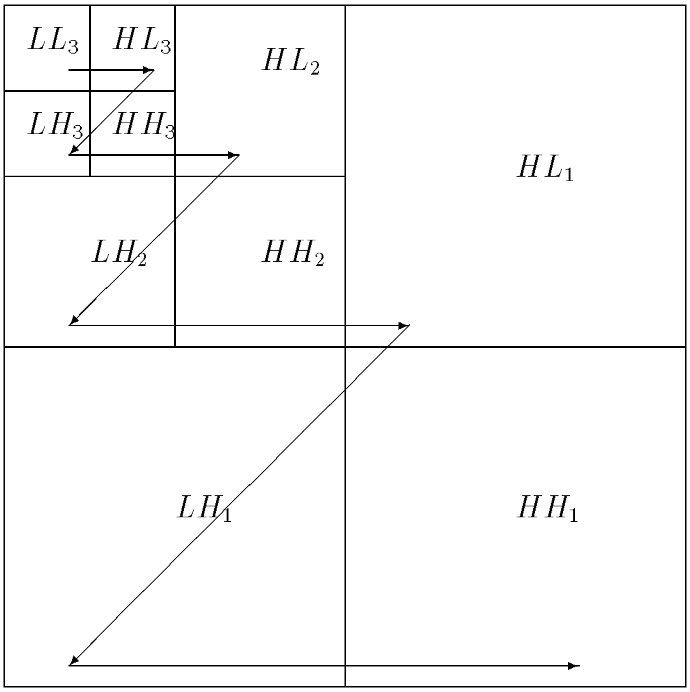

Figure 1: Scanning order of the wavelet transform coefficients with 3 scales.

indexing in **ICS**, for example, if x_3 is moved to **TPS**, then all coefficients after x_3 in **ICS** will have their indices subtracted by 1, and so on.

Right after the sorting pass, we have a refinement pass. In the refinement pass, the magnitudes of coefficients in **SCS** will have an additional bit of precision. For example, if the magnitude of a coefficient in **SCS** is known to be in $[32, 64)$, then it will be decided at this stage whether it is in $[32, 48)$ or $[48, 64)$. And a "0" symbol will indicate it is in the lower half $[32, 48)$, while a "1" symbol will indicate it is in the upper half $[48, 64)$. Output all these refinement values "0" and "1". Then append **TPS** to the end of **SCS**, reset

TPS to the empty set, and divide T by 2. Another round begins with the sorting pass.

The resulting symbol stream in the sorting pass and refinement pass will be further coded by adaptive arithmetic coding [6]. And the encoding can be stopped at any point, which allows a target rate or distortion metric to be met exactly.

4 Coding Results

A novel implementation of this new algorithm has been applied to several test images, including "Barbara", "Goldhill", "Lena". The Cohen-Daubechies-Feauveau 9/7-tap filters [1] with six scales were used. To make a direct comparison with other coders, the compression performance is measured by the peak signal to noise ratio (PSNR), though some other criterion might well be more preferable depending on the context.

We compared the coding results with J. Shapiro's embedded zerotree wavelet algorithm (EZW) [5], and A. Said, W. A. Pearlman's codetree algorithm (SPC) [4] . The PSNR versus bit rate on the 8 bit per pixel (bpp), 512×512, grayscale "Lena" image is plotted in Figure 2. In Figure 2, the top curve is Said and Pearlman's SPC algorithm. And our new algorithm is about 0.3 dB higher in PSNR than Shapiro's EZW algorithm. However, a better implementation of this new algorithm may possibly increase the coding performance, which we will report later. Some reconstructed images using this new algorithm are shown in Figure 3. The main advantage of this new algorithm over SPC and EZW is its simple form. It is a direct approach

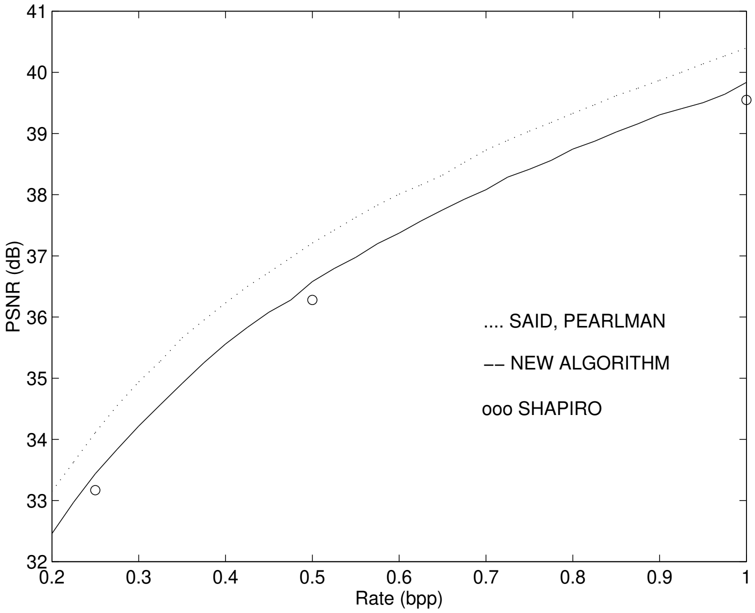

Figure 2: Coding Performance of the New Algorithm on "Lena", Compared with SPC and EZW

to locate the significant coefficients, while both SPC and EZW use spatial orientation tree structures to implicitly find the significant coefficients.

References

[1] A. Cohen, I. Daubechies, and J.-C. Feauveau, *Biorthonormal Bases of Compactly Supported Wavelets*, Commun. Pure Appl. Math., vol. XLV, pp. 485-560, 1992.

[2] P. Elias, *Universal Codeword Sets and Representations of the Integers*, IEEE Trans. Informat. Theory, vol. 21(2), pp. 194-203, Mar. 1975.

[3] D. Gottlieb, S. A. Hagerth, P. G. H. Lehot, and H. S. Rabinowitz, *A Classification of Compression Methods and Their Usefulness for a Large Data Processing Center*, Nat. Com. Conf. 44, pp. 453-458, 1975.

[4] A. Said, and W. A. Pearlman, *A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees*, submitted to IEEE Trans. Circ. and Syst. Video Tech., 1995.

[5] J. M. Shapiro, *Embedded Image Coding Using Zerotrees of Wavelet Coefficients*, IEEE Trans. Signal Processing, vol. 41, pp. 3445-3462, Dec. 1993.

[6] I. H. Witten, R. Neal, and J. G. Cleary, *Arithmetic Coding for Data Compression*, Commun. ACM, vol. 30, pp. 520-540, Jun. 1987.

(a) Original

(b) 8:1 Compression, PSNR = 39.84 dB

(c) 16:1 Compression, PSNR = 36.59 dB

(d) 32:1 Compression, PSNR = 33.44 dB

Figure 3: Reconstructed Images of 8 bpp, 512×512, grayscale "Lena" Using the New Algorithm