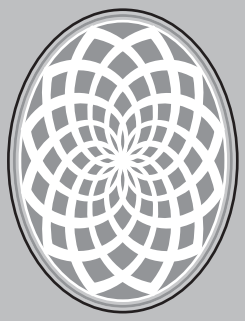


Industrial Mathematics Institute

1997:02

Data compression and elementary encoding of wavelet coefficients

Z. Gao, A. Andreev and R.C. Sharpley

IMI

Preprint Series

Department of Mathematics
University of South Carolina

Data Compression and Elementary Encoding of Wavelet Coefficients

by

Zhenguang Gao, Andrey Andreev, and Robert C. Sharpley [1]

Department of Mathematics, University of South Carolina, Columbia, SC 29208

Abstract— **Two elementary algorithms are introduced for image compression each of which is based on efficient, lossless encoding of quantized bi-orthogonal wavelet coefficients. Application of this type of algorithm is applied to several standard test images using regular and hyperbolic wavelet bases, and comparisons are given to Shapiro's EZW algorithm. Peak signal to noise ratio improvements typically of 0.6–0.8 dB are demonstrated. Generalizations of this type of algorithm to non-square images and higher dimensions are also briefly described.**

1. Introduction

Image compression is important in many applications, such as image transmission, feature extraction and denoising of data [6, 8, 11, 3]. Our original intent was to develop reliable data compression algorithms for three-dimensional time-dependent scalar and vector fields with general index limits (i.e. non-dyadic). In particular, a requirement for 3D data compression arose in our development of utilities for the interactive tracking and steering of remote simulations over small bandwidth or congested communication connections [14, 15]. The underlying rationale is that scientists and practitioners need the capability to quickly assess a simulation and correspondingly terminate or alter its progress. Interactive work on large scale simulators typically require data compression algorithms which permit a careful analysis of three dimensional data at various frequencies and resolutions. Wavelet analysis is a natural framework for these studies.

In our investigations, we developed various encoding algorithms for multi-dimensional data and tested them in the two-dimensional case by comparing the performance with existing algorithms for 2-D images. We obtained the somewhat surprising results that these algorithms performed very well in the tests, although they were much more simple to implement and generalize. The basic steps of our image compression algorithm include (1) application of discrete bi-orthogonal wavelet transformations to an extension of the image data, (2) a one-step quantization according to the precision required, (3) a natural ordering of the coefficients by index, (4) a preprocessing procedure of encoding the space-frequency correlations of the coefficients, followed by (5) an application of a Q-coder algorithm. The reconstruction (to within quantization) involves the simple inversion of these operations in the reverse order.

Steps 1 and 3-5 are generally lossless, but in practice noise may be present in the original image. The information loss occurs in the quantization stage, which may be estimated by approximation results of DeVore *et al.* (see [7] and inequality (21) below). As we show, the combined algorithm is very easy to implement and, compared to other algorithms, is computationally efficient even for multidimensional data.

[1]This work was supported in part by Martin Marietta Subcontract SK966V, DOE Grant DE-FG05-95ER25266, ONR Grant N00014-96-1-1003, and DoD Grant N00014-97-1-0806.

In the following sections we will discuss each of these steps. Section 2 outlines the multiresolution analysis for images based on tensor-products. Section 3 discusses the quantization and error estimate, while Section 4 describes the ordering of the quantized data without storing the location of each datum. We have developed preprocessing algoritms which have many variants, two of these are described in Section 5. In Section 6, we present the results of computational experiments with these encoders and compare their effectiveness with published results of Shapiro and an implementation of that algorithm. We provide concluding remarks and observations in Section 7 which relate the encoding technique with entropy and data complexity. A preliminary version of this research was reported earlier in [12]. We would like to express our thanks to Ronald DeVore for discussions concerning the results of this paper.

2. Wavelet Transforms

In this section we briefly describe biorthogonal locally supported wavelets and the corresponding multiresolution analysis (see [2, 16, 22] for details and additional references). We begin with an outline of the development in the univariate case with an emphasis on how to implement both decomposition and reconstruction of discrete data, and complete the section by describing 2D tensor product bases for images.

2.1 Multiresolution analysis.

A *multiresolution analysis* of $L_2(R)$ is defined as a 'ladder' of closed subspaces V_k which satisfy the following properties:

(1.a) There exists a *scaling* function φ with a non-vanishing integral, so that the closure of the span of its integer translates ($\varphi_j(x) := \varphi(x-j)$) is V_0. For computational purposes we will assume that φ has compact support. Moreover, we assume the collection $\{\varphi_j | j \in \mathrm{Z}\}$ is a *Riesz basis* for V_0: there are constants C_1, C_2 so that if $f \in V_0$, then $f = \sum_j c_j \varphi_j$ and

$$C_1 \, \|\{c_j\}\|_{\ell^2} \le \|f\|_{L_2(R)} \le C_2 \, \|\{c_j\}\|_{\ell^2}$$

(1.b) $$V_k \subset V_{k+1}, \quad k \in \mathrm{Z}$$

(1.c) $$\overline{\bigcup_{k\in\mathrm{Z}} V_k} = L_2(R) \quad , \quad \bigcap_{k\in\mathrm{Z}} V_k = \{0\}$$

(1.d) $$f(x) \in V_k \iff f(2x) \in V_{k+1} \quad \text{for all } k \in \mathrm{Z}$$

Although some of these properties are redundant, this formulation is best for our purposes. For a given scaling function φ, property (1.b) (with $k = 0$) implies that a refinement relation holds

(2) $$\varphi(x) = 2 \sum_j \alpha_j \varphi(2x - j).$$

By property (1.a) the coefficients must be unique. We assume further that the coefficients α_j satisfy the property

$$\sum_{j\in \mathrm{Z}} \alpha_j = 1 \ .$$

In practice, one usually determines coefficients with desirable properties for the resulting scale (and wavelet) functions and show that the above properties are satisfied. As contrasted with the usual orthogonal development of wavelets, the theory of biorthogonal wavelets instead makes use of two ladders of subspaces, one for the primal space and another for its dual. In the case of Hilbert spaces L_2, these spaces may be regarded as coinciding.

2.2 Univariate bi-orthogonal decompositions.

A *biorthogonal wavelet analysis* begins with two dual scale functions $\varphi, \widetilde{\varphi}$ which each provide a multiresolution analysis $V_k, \widetilde{V_k}$, respectively of L_2. Actually, one should think of V_k as providing a decomposition of L_2 and $\widetilde{V_k}$ performing a similar role for its dual. We denote by $\tilde{\alpha}_j$ the coefficients following from property (2) for the scale function $\tilde{\varphi}$. We define corresponding *dual wavelet* functions by

$$\psi(x) = 2\sum_j \beta_j \varphi(2x-j) \quad , \quad \tilde{\psi}(x) = 2\sum_j \tilde{\beta}_j \tilde{\varphi}(2x-j) \tag{3.a}$$

where

$$\beta_j = (-1)^j \tilde{\alpha}_{1-j} \ , \quad \tilde{\beta}_j = (-1)^j \alpha_{1-j}. \tag{3.b}$$

We also assume that the *dual scale functions* $\varphi, \tilde{\varphi}$ have the following properties:

$$\int_R \varphi(x)\ \tilde{\varphi}(x-j)\ dx = \delta_{0j} \tag{4}$$

then it follows that

$$\int_R \psi_{k,j}(x)\ \tilde{\psi}_{l,i}(x)\ dx = \delta_{i,j}\delta_{k,l}. \tag{5}$$

where we are using the standard notation

$$\phi_{k,j}(x) = \sqrt{2^k}\phi(2^k x - j) \tag{6}$$

for the L^2-normalized translated-dilates of a function ϕ. Here the index k denotes the scale and j indicates the integer shift. For our computational purposes we assume that each of the refinement sums in (2) is finite with an even number of symmetric 'filter' coefficients $\alpha_j, \tilde{\alpha}_j$:

$$\varphi(x) = 2\sum_{j=1-u}^{u} \alpha_j\ \varphi(2x-j), \quad \tilde{\varphi}(x) = 2\sum_{j=1-\tilde{u}}^{\tilde{u}} \tilde{\alpha}_j\ \tilde{\varphi}(2x-j). \tag{7}$$

The scale and wavelet functions φ and ψ generate corresponding spaces V_k, W_k defined by the closed linear spans

$$V_k = \mathrm{span}\{\varphi_{k,j} : j \in \mathrm{Z}\}$$

$$W_k = \text{span}\{\psi_{k,j} : j \in \mathrm{Z}\}$$

Similarly, we denote by $\widetilde{V}_k, \widetilde{W}_k$ the collection of spaces corresponding to $\tilde{\varphi}$ and $\tilde{\psi}$, respectively. It follows from the properties of the dual scale functions and the definitions of the dual wavelets that

$$V_{k+1} = V_k \oplus W_k, \quad \widetilde{V}_{k+1} = \widetilde{V}_k \oplus \widetilde{W}_k. \tag{8}$$

Although W_k may not be the orthogonal complement of V_k, the following relationships hold:

$$\widetilde{W}_k \perp V_k, \quad \widetilde{V}_k \perp W_k. \tag{9}$$

The decomposition (8) is repeated recursively to show for each nonnegative $n \in \mathrm{Z}$

$$\begin{aligned} V_n &= V_{n-1} \oplus W_{n-1} \\ &= V_{n-2} \oplus (W_{n-2} \oplus W_{n-1}) \\ &\vdots \\ &= V_0 \oplus (W_0 \oplus \cdots \oplus W_{n-1}) \ . \end{aligned} \tag{10}$$

Each $f \in V_n$ may then be written uniquely as

$$\begin{aligned} f &= \sum_{j\in\mathrm{Z}} c_j^n \varphi_{n,j} \\ &= \sum_{j\in\mathrm{Z}} c_j^{n-1} \varphi_{n-1,j} + \sum_{j\in\mathrm{Z}} d_j^{n-1} \psi_{n-1,j} \\ &= \sum_{j\in\mathrm{Z}} c_j^{n-2} \varphi_{n-2,j} + \left(\sum_{j\in\mathrm{Z}} d_j^{n-2} \psi_{n-2,j} + \sum_{j\in\mathrm{Z}} d_j^{n-1} \psi_{n-1,j} \right) \\ &\vdots \\ &= \sum_{j\in\mathrm{Z}} c_j^0 \, \varphi_{0,j} + \sum_{k=0}^{n-1} \left(\sum_{j\in\mathrm{Z}} d_j^k \, \psi_{k,j} \right). \end{aligned} \tag{11}$$

which is called its *biorthogonal wavelet decomposition.*

2.3 Decomposition stage for discrete functions.

For an discrete function $\mathbf{c} = \{c_j^n\}_{j=0}^{2^n-1}$ we associate the function $f \in V_n$ defined by

$$f = \sum_{j\in Z} c_j^n \varphi_{n,j}$$

where the coefficients are extended symmetrically at the two boundaries $j = 0$ and $j = 2^n - 1$ by the formula

$$c_j^n = \begin{cases} c_{-1-j}^n, & j = -1, -2, \ldots, -\tilde{u} \\ c_{2^{n+1}-j-1}^n, & j = 2^n, 2^n+1, \ldots, 2^n + \tilde{u} - 1 \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

in order to allow a systematic procedure for producing the next coarser level scale and wavelet coefficients as one procedes recursively through each step represented by the individual lines in (11).

From the biorthogonality, we may reproduce $\mathbf{c}$ through 'sampling' f by integration against the corresponding $\tilde{\psi}_{n,j}$. In particular, as $k = n, n-1, \ldots, 1$ the wavelet and scale coefficients, respectively, at the next coarser level $k-1$ are generated by

$$d_j^{k-1} = \sqrt{2} \sum_{l=1-u}^{u} \tilde{\beta}_l \, c_{2j+l}^k, \quad c_j^{k-1} = \sqrt{2} \sum_{l=1-\tilde{u}}^{\tilde{u}} \tilde{\alpha}_l \, c_{2j+l}^k \tag{13}$$

where $0 \le j \le 2^{k-1} - 1$. Similar to (12) above, we use the symmetric extension of the coefficients at each level:

$$c_j^k = \begin{cases} c_{-1-j}^k, & j = -1, -2, \ldots, -\tilde{u} \\ c_{2^{k+1}-j-1}^k, & j = 2^k, 2^k+1, \ldots, 2^k + \tilde{u} - 1 \end{cases} \tag{14}$$

to provide the necessary terms so that the process may proceed level-by-level. Symbolically this may be represented by the following diagram:

$$\begin{array}{ccccccccccc} \{c_j^n\} & \longrightarrow & \{c_j^{n-1}\} & \longrightarrow & \cdots & \{c_j^k\} & \longrightarrow & \{c_j^{k-1}\} & \longrightarrow & \cdots & \{c_j^1\} \longrightarrow \{c_j^0\} \\ & \searrow & & & & & \searrow & & & & \searrow \\ & & \{d_j^{n-1}\} & & & & & \{d_j^{k-1}\} & & & \{d_j^0\} \end{array} .$$

2.4 Reconstruction stage for discrete functions.

A discrete function is reconstructed as represented by the following diagram by moving from the coarse to the fine levels ($k = 0, 1, \ldots, n-1$):

$$\begin{array}{ccccccccccc} \{c_j^0\} & \longrightarrow & \{c_j^1\} & \longrightarrow & \cdots & \{c_j^{k-1}\} & \longrightarrow & \{c_j^k\} & \longrightarrow & \cdots & \{c_j^{n-1}\} \longrightarrow \{c_j^n\} \\ & \nearrow & & & & & \nearrow & & & & \nearrow \\ \{d_j^0\} & & & & & \{d_j^{k-1}\} & & & & \{d_j^{n-1}\} & \end{array}$$

using the reconstruction filters

$$\begin{aligned} c_{2j}^k &= \sqrt{2} \left(\sum_{l=-\lfloor \frac{u}{2} \rfloor}^{\lfloor \frac{u}{2} \rfloor} \alpha_{2l} \, c_{j-l}^{k-1} + \sum_{l=-\lfloor \frac{\tilde{u}-1}{2} \rfloor}^{\lfloor \frac{\tilde{u}}{2} \rfloor} \beta_{2l} \, d_{j-l}^{k-1} \right) \\ c_{2j+1}^k &= \sqrt{2} \left(\sum_{l=-\lfloor \frac{u}{2} \rfloor}^{\lfloor \frac{u-1}{2} \rfloor} \alpha_{2l+1} \, c_{j-l}^{k-1} + \sum_{l=-\lfloor \frac{\tilde{u}}{2} \rfloor}^{\lfloor \frac{\tilde{u}-1}{2} \rfloor} \beta_{2l+1} \, d_{j-l}^{k-1} \right) \end{aligned} \tag{15}$$

for the range of j from 0 to $2^{k-1} - 1$. Here again we take for the discrete function $\{\mathbf{c}^{k-1}\}$ its even symmetric extension according to (14), but for the wavelet coefficients $\{\mathbf{d}^{k-1}\}$ we must use the odd symmetric extension.

2.5 Images and Tensor Products of Wavelet Bases

A square image can be represented by its pixel values c_{ij}, $(i, j = 0, 1, \ldots, 2^n - 1)$ where all these values are integers in the interval [0, 255]. If we write these pixel values in a matrix M_0 and apply

the scaling and wavelet transforms (13) to each row of M_0, we obtain matrices M_1 and M_2 of scale and wavelet coefficients in the x-variable with y held fixed:

$$\left[\begin{array}{ccc} M_0 & M_1 & M_2 \end{array}\right].$$

In each of $M1$ and $M2$ we arrange the coefficients first according to scale (finest to lowest) and then by position (left to right). Applying the univariate wavelet transform next to the columns of M_0, M_1, and M_2, we obtain six additional matrices $M_3, \ldots, M_8$:

$$\left[\begin{array}{ccc} M_0 & M_1 & M_2 \\ M_3 & M_4 & M_5 \\ M_6 & M_7 & M_8 \end{array}\right], \tag{16}$$

where M_1, M_2 have dimension $2^n \times (2^n - 1)$, M_3, M_6 have dimension $(2^n - 1) \times 2^n$, and the remainder have dimension $(2^n - 1) \times (2^n - 1)$. Any $2^n \times 2^n$ linearly independent basis elements form a basis for a two dimensional wavelet transform. Although it is an abuse of terminology, we say that the coefficients are linearly independent when the corresponding collections of scaling and/or wavelet functions are linearly independent. Two bases generally used are the hyperbolic and regular bases. To form the *hyperbolic* decomposition, one uses all the coefficients in the matrix M_8 (i.e, tensor products of wavelets from all scales), the last row in M_5 (all wavelets in x tensored with the coarsest scale function in y), the last column in M_7 (all wavelets in y tensored with the coarsest scale function in x) and the coefficient in the lower right corner of M_4 (product of the coarsest scale functions in x and y). The *regular* basis decomposition is the more standard one employed in image processing and uses all the coefficients corresponding to the functions $\varphi(x)\psi(y)$, $\psi(x)\varphi(y)$, $\psi(x)\psi(y)$ scaled appropriately with square support, (i.e., corresponding elements in M_5, M_7 and M_8), together with the additional coarsest scale coefficient in the lower right corner of M_4. These two bases will be compared in our computational experiments in Section 6.

3. Quantization.

After an application of wavelet transforms, we obtain an indexed collection of real-valued coefficients. In the bivariate case, if $N = 2^n$ and a function f is represented as

$$f(x, y) = \sum_{i,j=0}^{N-1} c_{ij}^n \, \varphi(2^n x - i)\varphi(2^n y - j)$$

then, for convenience, we may write it as

$$f(x, y) = \sum_{k=0}^{N^2-1} a_k \, \phi_k(x, y) \tag{17}$$

where the functions ϕ_k are chosen to form either a hyperbolic or regular basis, and are normalized in the Lebesgue space L^p for a specific p in the interval $(0, \infty)$. In the seminal paper ([7], see also [6]) on the application of nonlinear approximation to wavelet compression, DeVore *et al* established, among many other results, that if f belongs to the Besov smoothness space $B_q^\alpha\left(L^q(I)\right)$ and is

represented as in equation (17) and if $\tilde{f}$ has a similar representation with coefficients $\tilde{a}_k$ which satisfy

$$|a_k - \tilde{a}_k| < \delta^{1/q}, \tag{18}$$

then

$$\|f - \tilde{f}\|_{L^p} = \mathcal{O}(\delta^{\alpha/2}). \tag{19}$$

Here α is a measure for the smoothness of f (see [7] for a complete explanation).

Based on this estimate, uniformly scaled integers will be used to approximate the real-valued coefficients in this representation according to a given precision ϵ. Let $[a]$ denote the closest integer to a real number a. For any $\epsilon > 0$, we have:

$$\left| \; a - 2\epsilon \, [a/2\epsilon] \; \right| \leq \epsilon.$$

If we apply this with $a = a_k$, and define the approximation $\tilde{f}$ by

$$\tilde{f}(x, y) = 2\epsilon \sum_{k=0}^{N^2-1} [a_k/2\epsilon] \, \phi_k(x, y) \tag{20}$$

then the coefficients $\tilde{a}_k$ of $\tilde{f}$ satisfy inequality (18) and therefore the approximation result given in (19) implies that

$$\|f - \tilde{f}\|_{L^p} = \mathcal{O}\left(\epsilon^{\alpha q/2}\right). \tag{21}$$

This process is referred to as *quantization* and ϵ is called the *threshold*. Note that all coefficients of f whose absolute value is smaller than the threshold are quantized to be zeros.

4. Ordering of Coefficients.

Many compression algorithms are based upon the assumption that a majority of the data after quantization are zero. In [7] an algorithm was described which discarded quantized coefficients whose magnitude was smaller than the threshold. The remaining quantized coefficients were arranged in decreasing order and their indices were recorded. A careful description was presented of the types of quantization at each scale level and methods for packing the location of the quantized coefficients to reduce the number of bits per pixel required to transmit the compressed image.

We avoid the overhead of ordering the coefficients and attempt to minimize the effort in the transmission of the locations of the significant coefficients. Our approach uses the observation that the data are naturally arranged in a rough increasing order when viewed relative to their locations within the matrices described in Section 2.4. This rough estimate for a given function f is indicated by the following elementary inequality

$$\left| \int f\phi \right| \leq |b - a|^{1/2} \, \|f - P\|_2 \tag{22}$$

where ϕ is normalized in L^2, has m vanishing moments and is supported in $[a,b]$ and P is any polynomial of degree at most m on $[a,b]$. The inequality follows from the standard approximation argument. Since ϕ has m vanishing moments, $\int P\,\phi = 0$ and

$$\begin{aligned}\left|\int f\phi\right| &= \left|\int (f-P)\phi\right| \le \|f-P\|_{L^q[a,b]}\,\|\phi\|_{L^p} \\ &\le |b-a|^{1/q}\,\|f-P\|_\infty.\end{aligned}$$

Roughly speaking, inequality (22) says that the smaller the support of of a wavelet, the smaller the coefficient of f corresponding to that wavelet. In the matrix of coefficients (16), however, the support of each coefficient is directly related to their matrix positions, with the upper left corner coefficients having the smallest support and those in the lower right corner the largest. Therefore, as we index through the appropriate sub-matrices (for a given basis), we move from smaller to larger supports to ensure that generally coefficients with small values occur first.

5. Encoding of Quantized Coefficients.

In this section, we describe two relatively simple approaches for encoding the quantized wavelet coefficients resulting from the decomposition procedure described in the last section. We use to our advantage the spatial-frequency localization of features to further condense the significant information extracted by the decomposition procedure. The methods we describe are both lossless encoders which pre-process the data to enhance the subsequent application of standard encoders, such as Q-coder [17], for example. The first approach, which we refer to as "**interleaving**," compresses the rows (or columns) of the array of wavelet coefficients by recursively interleaving adjacent coefficients in each row of the coefficient matrix and only retaining the significant bits. The second approach, refered to as "**bit-stream encoding**," makes use of the higher correlations which exist in multivariate, logically rectangular data.

5.1 The Interleaving Method.

Many wavelet-based compression algorithms rely upon the assumption that a majority of the quantized data vanish. Improving the approximation to f requires decreasing the threshold to produce additional nonzero coefficients. Nevertheless, many of the quantized coefficients remain zero, and the ones no longer zero will be small integers.

The first encoding method uses this empirical observation and the fact that small signed integers require a small number of bits for their representation. The algorithm condenses pairs of these integers to encoded integers, using the natural ordering within the matrices (16) as described in the previous section. The procedure is continued until it is no longer possible, at which time both the index and value of the coefficient which failed to be condensed in this manner are recorded. We iterate this procedure until it is no longer efficient. Our computational experiments with images have shown that a fixed value of four iterations provides near optimal results on standard test images, but there are many variations that may be applied to tune this type of algorithm depending on the class of images being analyzed.

The interleaving process is simply to take two signed integers (denoted, for example, by c_1 and c_2) which each require l bits for their representation, and pack them into $2l$ bits, but we ensure that the resulting integer (which we call $cond(c_1,c_2)$) is still relatively small. One method for

implementing this is to use the highest and lowest order bits of the integer $cond(c_1, c_2)$ for the sign bits of c_1 and c_2, respectively. The sgn of c is defined to be '0' if c is nonnegative, and '1' otherwise. Next we define for any signed integer c, $val(c) = |c| - sgn(c)$. The positive integers $val(c_1)$ and $val(c_2)$ are packed into the remaining bits of $cond(c_1, c_2)$ by using the lowest odd bits for c_1 and the lowest even bits for c_2. It is clear that if

$$-2^{l-1} \le c_i < 2^{l-1}, \quad (i = 1, 2)$$

then

$$-2^{2l-1} \le cond(c_1, c_2) < 2^{2l-1}.$$

For the purposes of illustration, let $c_1 = 2$ and $c_2 = -4$, then $sgn(c_1) = 0$, $sgn(c_2) = 1$ and, in binary form, $val(c_1) = 010$, $val(c_2) = 011$. In this case, $val\,(cond(c_1, c_2)) = 00011011$ and $sgn\,(cond(c_1, c_2)) = 0$.

This process on roughly-ordered quantized coefficients may be summarized as follows: Proceeding through the appropriate matrices in (16), we find the first integer coefficient which is outside the range $[-128, 127]$ and record its index n_1. We then condense the pairs of coefficients up to that index by interleaving adjacent pairs. If n_i is smaller than a given value (i.e. coarse enough, so that few coefficients remain), we terminate the process, otherwise we repeat the procedure on the resulting data set until stopping criteria are satisfied.

The final step of the algorithm applies Q-Coder[2] [17] to the encoded data as well as to the exceptional data and their locations, in order to obtain a final compressed data file. The reconstruction process is straightforward and well-defined, since we keep the record of n_i. The algorithm as described is clearly lossless up to the original quantization procedure. Variants of the complete procedure also provide lossless compression of data with progressive transmission. The algorithms may be further tuned using various additional quantization methods preserving the approximation error and varying the stopping criteria according the classes of image data.

5.2 The Bit-Stream Method.

The "bit-stream" encoding method relies on the spatial and frequency correlations among the quantized wavelet coefficients, which are extracted by the wavelet decomposition process, and should be expected to consistently perform better than the interleaving method just described. For simplicity, we consider the regular wavelet basis and indicate how to modify the algorithm for the hyperbolic basis at the end of this section. We briefly describe the general procedure and then use an example to illustrate the details of the encoding process. We will begin with a matrix $\mathbf{A}$ listing only quantized regular basis wavelet coefficients (i.e., any of the matrices M_5, M_7, or M_8, corresponding respectively to $\varphi\psi$, $\psi\varphi$, or $\psi\psi$). From $\mathbf{A}$ we generate a vector of nonzero coefficients $\mathbf{c}$, and three bitstream linear arrays which encode the index positions: the linear array $\mathbf{S}$ will encode the strongly significant spatial correlations and implicitly contains some frequency correlations, $\mathbf{F}$ encodes the frequency correlations, and $\mathbf{R}$ denotes the array which encodes weakly significant spatial correlations.

Since the majority of the coefficients of $\mathbf{A}$ are zero, we write each nonzero coefficient into $\mathbf{c}$ according to its index position (ordered first by rows, next by columns, and so on), after subtracting one from each positive coefficient in order to slightly decrease its size for efficient packing. We use

[2]Our version of Q-coder was implemented by V. Zanev [23].

$\mathbf{X}$ to denote the "indicator matrix" of $\mathbf{A}$, i.e. the characteristic function of the support set of indices of nonzero coefficients of $\mathbf{A}$. For example, if we take the 7×7 matrix:

$$\mathbf{A} = \begin{array}{|c|c|c|c|c|c|c|}\hline 0 & 0 & 1 & -1 & & & \\ \hline 0 & 0 & -3 & 5 & & & \\ \hline 1 & 0 & 0 & 0 & & & \\ \hline 0 & 4 & 6 & 7 & & & \\ \hline & & & & 0 & -3 & \\ \hline & & & & 2 & 1 & \\ \hline & & & & & & 5 \\ \hline \end{array}, \tag{23}$$

the corresponding indicator matrix $\mathbf{X}$ is given by

$$\mathbf{X} = \begin{array}{|c|c|c|c|c|c|c|}\hline 0 & 0 & 1 & 1 & & & \\ \hline 0 & 0 & 1 & 1 & & & \\ \hline 1 & 0 & 0 & 0 & & & \\ \hline 0 & 1 & 1 & 1 & & & \\ \hline & & & & 0 & 1 & \\ \hline & & & & 1 & 1 & \\ \hline & & & & & & 1 \\ \hline \end{array} \tag{24}$$

and the coefficient matrix with positive coefficients reduced by one becomes

$$\tilde{\mathbf{A}} = \begin{array}{|c|c|c|c|c|c|c|}\hline 0 & 0 & 0 & -1 & & & \\ \hline 0 & 0 & -3 & 4 & & & \\ \hline 0 & 0 & 0 & 0 & & & \\ \hline 0 & 3 & 5 & 6 & & & \\ \hline & & & & 0 & -3 & \\ \hline & & & & 1 & 0 & \\ \hline & & & & & & 4 \\ \hline \end{array} \tag{25}$$

which is encoded into the array $\mathbf{c}$, corresponding to the "nonzero coefficients" of $\mathbf{A}$,

$$\mathbf{c} = (0 \ -1 \ -3 \ \ 4 \ \ 0 \ \ 3 \ \ 5 \ \ 6 \ -3 \ \ 1 \ \ 0 \ \ 4).$$

Obviously, $\mathbf{A}$ may be recovered from $\mathbf{c}$ and the indicator matrix $\mathbf{X}$.

The correlations within the matrix $\mathbf{X}$ are encoded by using two types of mappings, one for locally encoding spatial correlations within a given frequency level:

adjacent symbol pairs from matrix $\mathbf{X}_j^{(i)}$	encoded in the matrix $\mathbf{X}_j^{(i+1)}$	encoded in the bitstream array $\mathbf{S}$	encoded in the bitstream array $\mathbf{R}$
0 0	0		
0 1	1	0	0
1 0	1	0	1
1 1	1	1	

(26)

and another for the correlations among adjacent frequencies:

symbols in submatrices from two levels of $\mathbf{X}_j^{(2)}$	encoded in the matrix $\mathbf{X}_j$	encoded in the array $\mathbf{F}$
0 0	0	
0 1	1	0
1 0	1	-1
1 1	1	1

(27)

To illustrate the encoding procedure, we continue with the 7×7 example matrix $\mathbf{X}$ displayed in (24). We set $\mathbf{X}_0 = \mathbf{X}$ and work our way up from the finest level (i.e. highest frequency) to the coarsest by a recursive procedure. The finest level indices for $\mathbf{X}_0$ are those in its 4×4 upper left submatrix. We apply the mapping (26) first to adjacent pairs in the rows of this submatrix, to obtain

$$\mathbf{X_1^0} := \mathbf{X_0} = \begin{array}{|c|c|c|c|c|c|c|}\hline 0&0&1&1&&&\\\hline 0&0&1&1&&&\\\hline 1&0&0&0&&&\\\hline 0&1&1&1&&&\\\hline &&&&0&1&\\\hline &&&&1&1&\\\hline &&&&&&1\\\hline\end{array} \implies$$

$$\mathbf{X_1^{(1)}} = \begin{array}{|c|c|c|c|c|}\hline 0&1&&&\\\hline 0&1&&&\\\hline 1&0&&&\\\hline 1&1&&&\\\hline &&0&1&\\\hline &&1&1&\\\hline &&&&1\\\hline\end{array}, \quad \mathbf{S} = (11001), \quad \mathbf{R} = (10).$$

We next apply this procedure to the columns of $\mathbf{X_1^{(1)}}$, appending to the appropriate bitstreams $\mathbf{S}$ and $\mathbf{R}$ to obtain

$$\mathbf{X_1^{(1)}} = \begin{array}{|c|c|c|c|c|}\hline 0&1&&&\\\hline 0&1&&&\\\hline 1&0&&&\\\hline 1&1&&&\\\hline &&0&1&\\\hline &&1&1&\\\hline &&&&1\\\hline\end{array} \implies \mathbf{X_1^{(2)}} = \begin{array}{|c|c|c|c|c|}\hline 0&1&&&\\\hline 1&1&&&\\\hline &&0&1&\\\hline &&1&1&\\\hline &&&&1\\\hline\end{array}, \quad \mathbf{S} = (11001\ 110), \quad \mathbf{R} = (10\ 0).$$

At this stage we have encoded the horizontal and vertical spatial correlations within the current frequency level. The next step of the process is to encode frequency information from the current

level to the next coarsest using the mapping (27). This is accomplished by comparing the two 2×2 submatrices, position by position, from the two levels (with finer first):

$$\mathbf{X_1^{(2)}} = \begin{array}{|c|c|c|c|c|}\hline 0&1&&&\\\hline 1&1&&&\\\hline &&0&1&\\\hline &&1&1&\\\hline &&&&1\\\hline\end{array} \Longrightarrow \mathbf{X_1^{(3)}} = \begin{array}{|c|c|c|}\hline 0&1&\\\hline 1&1&\\\hline &&1\\\hline\end{array}, \quad \mathbf{S} = (11001\ 110), \quad \mathbf{R} = (10\ 0), \quad \mathbf{F} = (111).$$

The matrix $\mathbf{X_0}$ is now reduced (i.e., reduced to $\mathbf{X_1} := \mathbf{X_1^{(3)}}$) from having three frequency levels to two, together with the bitstreams **S**, **R**, and **F**. Now we iterate these three steps to further reduce the number of submatrices in **X** while appending to the bitstreams. It is clear that the process is invertible at each stage and therefore the encoding is lossless. The full recursive process with our simple example is to encode the horizontal spatial correlations as step 1,

$$\mathbf{X_1} = \begin{array}{|c|c|c|}\hline 0&1&\\\hline 1&1&\\\hline &&1\\\hline\end{array} \Longrightarrow \mathbf{X_2^{(1)}} = \begin{array}{|c|c|}\hline 1&\\\hline 1&\\\hline &1\\\hline\end{array}, \quad \mathbf{S} = (11001\ 110\ 01), \quad \mathbf{R} = (10\ 0\ 0), \quad \mathbf{F} = (111),$$

to encode the vertical spatial correlations as step 2,

$$\mathbf{X_2^{(1)}} = \begin{array}{|c|c|}\hline 1&\\\hline 1&\\\hline &1\\\hline\end{array} \Longrightarrow \mathbf{X_2^{(2)}} = \begin{array}{|c|c|}\hline 1&\\\hline &1\\\hline\end{array}, \quad \mathbf{S} = (11001\ 110\ 01\ 1), \quad \mathbf{R} = (10\ 0\ 0), \quad \mathbf{F} = (111),$$

and to encode corresponding frequency correlations in step 3:

$$\mathbf{X_2^{(2)}} = \begin{array}{|c|c|}\hline 1&\\\hline &1\\\hline\end{array} \Longrightarrow \mathbf{X_2} = \boxed{1}, \quad \mathbf{S} = (11001\ 110\ 01\ 1), \quad \mathbf{R} = (10\ 0\ 0), \quad \mathbf{F} = (111\ 1).$$

At the last stage, we append to **S** the singleton entry of $\mathbf{X_{n-1}}$ if **n** is the number of frequency levels. To reconstruct the matrix **A**, we need only the bitstreams **S**, **R**, and **F**, together with the integer coefficient array **c**. The original thresholded matrix of wavelet coefficients is obtained from **A** and the threshold constant ϵ. Standard arithmetic encoders are used to compress both **c** and the bitstreams **S** and **F**. The bitstream **R** has entries which are equally likely and therefore further compression should not be expected to be effective on this array.

We wish to emphasize that this algorithm is quite simple to extend in order to handle higher dimensional data. After computing $\mathbf{X_{j+1}^{(2)}}$ which encodes horizontal and vertical correlations, we may apply the procedure again to obtain "depth" correlations. Another simple modification is to handle all three matrices M_5, M_7, and M_8, simultaneously, by consolidating the scaling-wavelet correlations componentwise into an additional bitstream **SW**. The algorithm is also easily modified to handle rectangular data by using a rectangular "template" to combine level to level instead of the square "template" described above. When the dimensions are not a power of two, we use our extension process to provide the additional data to allow the algorithm to proceed.

Finally, fixed bases representations other than "regular" bases can be handled with again only slight modifications. The algorithm we have described for the "regular" basis uses wavelet (and

scaling) basis elements of the same frequency to determine spatial correlations. Recall from section 2 that the "hyperbolic" basis consists of all wavelet coefficients plus two "wings" from the matrices M_5 and M_7. We consider at each frequency level k those coefficients for which at least one of the two frequencies of the tensor product factors is of level k. We reduce this template first in the horizontal direction and then in the vertical using the operation (26) as was used for the pairs of regular basis coefficients. As before we can then compare the resulting reduced k-th level to next coarsest level component by component using the operation in (27).

We finish our description of the "bit-stream encoding" by stating a couple of observations which are not apparent in our contrived 7×7 example indicator matrix displayed in (24). In our experiments with standard test images, the bitstreams **S** consist primarily of zeros, since for these images the pair (11) appears infrequently compared to the pairs (10) and (01) and therefore the bitstream should compress significantly. A wavelet coefficient will normally be large when an edge or sudden change occurs. Depending on the classification of the image, these changes typically occur over lower dimensional sets which lead to high occurances of the patterns (10) and (01).

Shapiro observed in his quadtree formulation for the regular wavelet basis, that typically if a coefficient is zero, then with high probabilty the quadtree branch below that node would consist of zeros. Our experiments have shown that approximately 10% of the time, if an entry in the coefficient array vanishes, then the coefficients in the higher frequencies which would be combined in the reduction process with this coefficient (i.e. whose indexing dyadic squares are contained in the indexing dyadic square of this coefficient) will all be zero. Since our process essentially builds the graph structure of the nonzero entries, then the zero entries are essentially ignored, which indicates that compression should be effective.

6. Computational Results.

In this section, we present computational results comparing an implementation (matching the published results in [20]) of Sharpiro's EZW algorithm, the simplified interleaving algorithm, and the bit-stream encoding algorithm described in this paper. We apply these algorithms to both regular and hyperbolic wavelets using the 6-10 biorthogonal filters (see Table I for the coefficients). In each of these experiments, the resulting files are passed through the Q-coder algorithm to finalize the compression. We have chosen four standard test images (see Figure 1) in order to compare the compression rates and the PSNR (Peak Signal to Noise Ratio) of the methods. The PSNR is considered to be a measure of image quality and is given by the formula

$$PSNR = 20\ \log_{10}\left(\frac{255}{\|f-\tilde{f}\|_{\ell_2}}\right)$$

where f is the original image, and $\tilde{f}$ is the reconstructed image from the compressed data, and ℓ_2 is the least squares norm. The four test images range from relatively smooth to rough images (Lena, Barbara, City, Destroyer). All experiments are performed by encoding and then decoding 512×512 square images for comparison. As mentioned above, we have implemented our own version of Shapiro's EZW algorithm which performs according to the published results in [20], while the general Q-coder was implemented by our colleague Zanev [23] as described in [17].

We argue that our algorithms, using both regular and hyperbolic wavelets, are at least comparable to Shapiro's EZW algorithm and are easy to implement in both software and hardware.

Furthermore, the methods are efficient and easily extendible to arbitrary $m \times n$ rectangular images index limits and, perhaps more importantly, to higher dimensions.

In Table II, we compare compression rates of the Lena test image for Shapiro's method in [20], our implementation of his method, our elementary interleaving encoding, and our bit-stream encoding using both regular and hyperbolic wavelets. Table III provides similar information about the Barbara test image. These two images were selected since they are standard and were used as test images in [20]. The reasoning behind the inclusion of the results from our implementation of Shapiro's method is to show that it performs consistent with published results in [20], for both the Lena and Barbara images. We can then get an indication of the performance of that method on other less smooth test images for which test results are not available in [20]. Tables IV–V compares our implementation of Shapiro's method against the encoding applied to both Regular and Hyperbolic biorthogonal wavelet decompositions for the City and Destroyer test images. The computational results for bit-stream encoding against Shapiro's method is also presented in Plots 1–3. The results of Table II are presented in Plot 1 which plots the PSNR against compression rate, which ranges from 8 to 256. Plots 2–3 presents the corresponding information for the Barbara, City and Destroyer test images, respectively. It is clear from these plots and tables that all six methods are comparable with some advantage of the interleaving encoding for the Destroyer image. Figure 2 compares the compressed Lena images by the five methods against the original image using a compression rate of 128. Figure 3 performs the same role for the rougher Barbara image. These figures give a qualitative measure of compression for the three methods holding the compression ratio fixed.

The images we use were also selected since they have varying degrees of smoothness in Besov spaces and the nonlinear approximation theory of DeVore [10] provides a basis for analysis of these results. From the tables and plots we can see that generally the bit-stream method gives the highest compression while the interleaving method gives the lowest, with the exception for the Destroyer image. It is well known that the regular basis works better than the hyperbolic basis for the 'smooth' Lena image, while hyperbolic basis is a better choice for 'rougher' Barbara image. In combination with our algorithms, both interleaving and bit-stream encoding, this property is retained.

To indicate the strong potential of using the encoders and multidimensional biorthogonal wavelets for data compression, we apply these methods to a relatively rough contaminant concentration front invading the groundwater in a porous media (see Figure 4). The scalar data field is evaluated on a nonuniform grid with $137 \times 129 \times 11$ vertices. In this experiment, we use the hyperbolic basis with the symmetric 6-10 filter in each coordinate and the interleaving method for encoding the significant coefficients. We have rendered this scalar concentration field according to the color map located at the bottom of each image. In this figure, we have compared the renderings of the original file against those reconstructed from compressed files with compression rates of 172, 493, and 1200, respectively. Similar compression rates have been obtained for compression of the grid data, with no visual loss of information. In practice, the original grid data are generated from scattered data whose measurements contain varying degrees of error. One may then perhaps argue that the compressed (denoised) grid provides at least as good a representation of the physical grid as the original file. These same algorithms have been applied to video and to time-dependent three-dimensional fields and will be reported on elsewhere.

7. Remarks

As stated in the introduction, our original intent for this study was the development of reliable data compression algorithms for three-dimensional scalar and vector fields with general index limits (i.e. non-dyadic) for use in interactive steering of simulations on remote massively parallel supercomputers [14, 15]. In this report we have described two methods for encoding the biorthogonal wavelet coefficients and have demonstrated that they generally outperform other compression methods. We also observe that bit-stream encoding is typically the superior of the two methods for most images. Our algorithms, though primarily tested for 2 dimensional square images, have natural extensions to higher dimensions for any logically rectangular data. These algorithms also are well suited for use with adaptive bases, for example the minimal entropy basis [4] and adaptive nonlinear bases [18].

Shannon's entropy may be used to understand the effectiveness of the bitstream encoding. If S is a random variable which takes k values $v_1, v_2, \ldots, v_k$ with corresponding probabilities $p_1, p_2, \ldots, p_k$, then the *entropy* of S may be defined as

$$H(S) = -\sum_{i=1}^{k} p_i \log_2 p_i$$

and is considered a measure of the complexity of S. The Noiseless Coding Theorem of information theory [1, 19] guarantees that the minimal expected length of encoded data among all uniquely decipherable binary encoding schemes may essentially be obtained by an extension of Huffman encoding. Shannon's theory states that if the size of the data is n, then $nH(S)$ is the expected number of bits required to represent S and is then called the *complexity* of S. This branch of information theory also provides prefix encoders, such as Huffman encoding, which on average provide minimal length encoding of the data (i.e. $nH(S)$). Datum in the data sets are treated as values generated by repeated application of a random variable, that is, it is assumed that the values are produced by independent, identically distributed random variables. In practice, the probabilities are then estimated by the frequency table of the data. The bitstream preprocessor may be regarded as a prefix encoder which extracts and encodes significant wavelet correlations by means of a local space-frequency Huffman procedure (see definitions (26) and (27)) and thereby prepares the data for an enhanced application of standard data compression algorithms such as that which is incorporated in Q-coder.

Theoretically, we may describe the performance of the bitstream algorithm as follows. Given a binary data set S with complexity $nH(S)$, if we apply one step bitstream encoding on S, then S will be represented as a 4-ary data set S' whose size is $\frac{n}{2}$. The complexity of S' is $\frac{n}{2}H(S')$. Under these conditions, one can show [13] that $\frac{n}{2}H(S') \leq nH(S)$ and that bitstream encoding decreases the complexity of the original data. This result not only assures the effectiveness of the bitstream encoding, but also provides a parameter[13], which depends on relative ratios of the frequencies of the four symbols, to measure its performance.

Typical data which we have encountered in large scale scientific simulations and medical image processing may be quite sizable and has motivated further study of parallel versions of our algorithms, as well as progressive transmission. These efforts will be reported on elsewhere.

References

[1] D. Applebaum, *Probability and Information: An Integrated Approach*, Cambridge University Press, New York, 1996.

[2] A. Cohen, I. Daubechies and J.-C. Feauveau, Biorthogonal Bases of Compactly Supported Wavelets, *Comm. Pure and Applied Math.* **45**(1992), 485-560.

[3] R.R. Coifman and D.L. Donoho, "Translation-Invariant De-Noising", in *Wavelets and Statistics (Villard de Lans, 1994)*, Anestis Antoniadis and Georges Oppenheim, ed., Lecture Notes in Statist. **103**, Springer Verlag, New York, 1995.

[4] R. R. Coifman and M.V. Wickerhauser, *Entropy Based Algorithms for Best Basis Selection*, IEEE Transactions on Information Theory **3**(1992), 713–718.

[5] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, Philadelphia, 1992.

[6] R. A. DeVore, B. Jawerth, and B. J. Lucier, Data Compression Using Wavelets: Error, Smoothness, and Quantization, in *DCC '91, Data Compression Conference*, J.A. Storer *et al.* (ed.), 186–195, IEEE Computer Society Press, Los Alamitos, CA, 1991.

[7] R. A. DeVore, B. Jawerth, B. J. Lucier, Image Compression Through Wavelet Transform Coding, *IEEE Trans. Inf. Theory* **38** (1992), 719–746.

[8] R. A. DeVore and B.J. Lucier, Classifying the Smoothness of Images: Theory and Applications to Wavelet Image Processing, in *Proceedings of ICIP-94*, vol. II, IEEE Computer Soc. Press, 1994, pp. 6–10.

[9] R. A. DeVore, S.V Konyagin, and V.N. Temlyakov, Hyperbolic Wavelet Approximation, *Constructive Approximation* **14**(1998), (to appear).

[10] R.A. DeVore, Nonlinear Approximation, *Acta Numerica*, (1998), 99 pages (to appear).

[11] D. Donoho, Denoising by Soft Thresholding, *IEEE Trans. Inform. Theory* **41**(1995), 613–627.

[12] Z. Gao, A. Andreev and R.C. Sharpley, An Elementary Compression Method on Wavelet Coefficients, *IMI Report* **97:02**, Department of Mathematics, University of South Carolina, Columbia, SC, January 1997.

[13] Z. Gao, *Wavelet Transforms and Data Compression*, Ph.D. thesis, University of South Carolina, (in preparation).

[14] Kaulgud, A. and R.C. Sharpley, An Interactive Tracking/Steering Library, *IMI Report* **95:10**, Department of Mathematics, University of South Carolina, Columbia, SC, August 1995.

[15] L. Scott Johnson, A. Kaulgud, R.C. Sharpley, R.E. Ewing, Z. Leyk, J. Pasciak, M. Celia, and J.R. Brannan, *Integration of Contaminant Transport Simulators on Parallel Machines with a Graphical User Interface for Remote Interactive Modeling,* in "Proceedings of the 1997 Simulation Multiconference," Atlanta, April 1997, Soc. for Computer Simulation International, San Diego, p. 319–324.

[16] D. B. Meade and D. Mitchum, *WV - A User-Friendly Image Processing Package Using Wavelets*, Ver. 1.0, University of South Carolina, June 1994.

[17] W.B. Pennebaker, J.L. Mitchell, G.G. Langdon, Jr., and R.B. Arps, An Overview of the Basic Principles of the Q-Coder Adaptive Binary Arithmetic Encoder, *IBM J. Res. Dev.* **32** (1988), 717–727.

[18] P. Petrushev, personal communication.

[19] S. Roman, *Introduction to Coding and Information Theory*, Undergraduate Texts in Mathematics, Springer-Verlag, New York, 1996.

[20] J.M. Shapiro, An Embedded Hierarchical Image Coder Using Zerotrees of Wavelet Coefficients, J. A. Storer and M. Cohn (ed.), *IEEE Computer Society Press*, Los Alamitos, CA, 1993.

[21] R. N. Williams, *Adaptive Data Compression*, Kluwer Academic Publishers, Dordrecht, 1994.

[22] Z. Yang, *Wavelets and Image Compression*, Ph.D. thesis, 1995, University of South Carolina.

[23] V. Zanev, A Software Implementation of Q-Coder, University of South Carolina, 1995.

l	$\tilde{\alpha}_l$	$\tilde{\beta}_l$
-4	0	0.01337437
-3	0	0.00494231
-2	-0.09127176	-0.04754360
-1	0.03372823	0.09432042
0	0.55754352	0.43490656
1	0.55754352	0.43490656
2	0.03372823	0.09432042
3	-0.09127176	-0.04754360
4	0	0.00494231
5	0	0.01337437

TABLE I. 6–10 Biorthogonal Wavelet Filter Coefficients for Computational Experiments (see equation (13)).

Compression Rate	Shapiro PSNR (dB)	USC Shapiro PSNR (dB)	Regular Condensation PSNR (dB)	Hyperbolic Condensation PSNR (dB)	Regular Bit Stream PSNR (dB)	Hyperbolic Bit Stream PSNR (dB)
8	39.55	39.16	39.22	38.93	39.57	39.24
16	36.28	36.05	35.88	35.82	36.65	36.33
32	33.17	32.83	33.15	32.82	33.72	33.45
64	30.23	30.01	29.91	29.78	30.89	30.51
128	27.54	27.66	27.66	27.16	28.21	27.90
256	25.38	25.33	25.26	25.07	25.86	25.53

TABLE II. Quantitative Results Comparing Peak-Signal to Noise Ratios (PSNR) for 512×512 Lena Image Using Various Methods with Compression Rates from 8–256.

Compression Rate	Shapiro PSNR (dB)	USC Shapiro PSNR (dB)	Regular Condensation PSNR (dB)	Hyperbolic Condensation PSNR (dB)	Regular Bit Stream PSNR (dB)	Hyperbolic Bit Stream PSNR (dB)
8	35.14	33.64	35.10	35.20	35.59	35.65
16	30.53	29.09	30.55	30.49	30.81	30.96
32	26.77	26.09	26.66	27.04	27.22	27.50
64	24.03	24.22	24.05	24.18	24.43	24.69
128	23.10	22.76	22.83	22.71	23.10	23.00
256	21.94	21.92	21.68	21.50	22.14	21.92

TABLE III. Quantitative Results Comparing Peak-Signal to Noise Ratios (PSNR) for 512×512 Barbara Image Using Various Methods with Compression Rates from 8–256.

Compression Rate	USC Shapiro PSNR (dB)	Regular Condensation PSNR (dB)	Hyperbolic Condensation PSNR (dB)	Regular Bit Stream PSNR (dB)	Hyperbolic Bit Stream PSNR (dB)
8	24.04	24.72	24.19	25.02	24.55
16	22.03	22.25	21.81	22.43	21.98
32	20.33	20.63	20.26	20.77	20.39
64	19.14	19.42	19.17	19.53	19.24
128	18.40	18.44	18.26	18.57	18.40
256	17.60	17.70	17.50	17.81	17.61

TABLE IV. Quantitative Results Comparing Peak-Signal to Noise Ratios (PSNR) for 512×512 City Image Using Various Methods with Compression Rates from 8–256.

Compression Rate	USC Shapiro PSNR (dB)	Regular Condensation PSNR (dB)	Hyperbolic Condensation PSNR (dB)	Regular Bit Stream PSNR (dB)	Hyperbolic Bit Stream PSNR (dB)
8	36.89	40.85	40.67	40.38	40.56
16	33.31	34.78	35.35	34.62	35.31
32	31.42	32.08	32.50	31.97	32.51
64	29.15	30.69	30.86	30.78	30.92
128	26.26	28.39	28.40	28.48	28.52
256	24.10	25.64	25.88	25.70	26.23

TABLE V. Quantitative Results Comparing Peak-Signal to Noise Ratios (PSNR) for 512×512 Destroyer Image Using Various Methods with Compression Rates from 8–256.

(a)

(b)

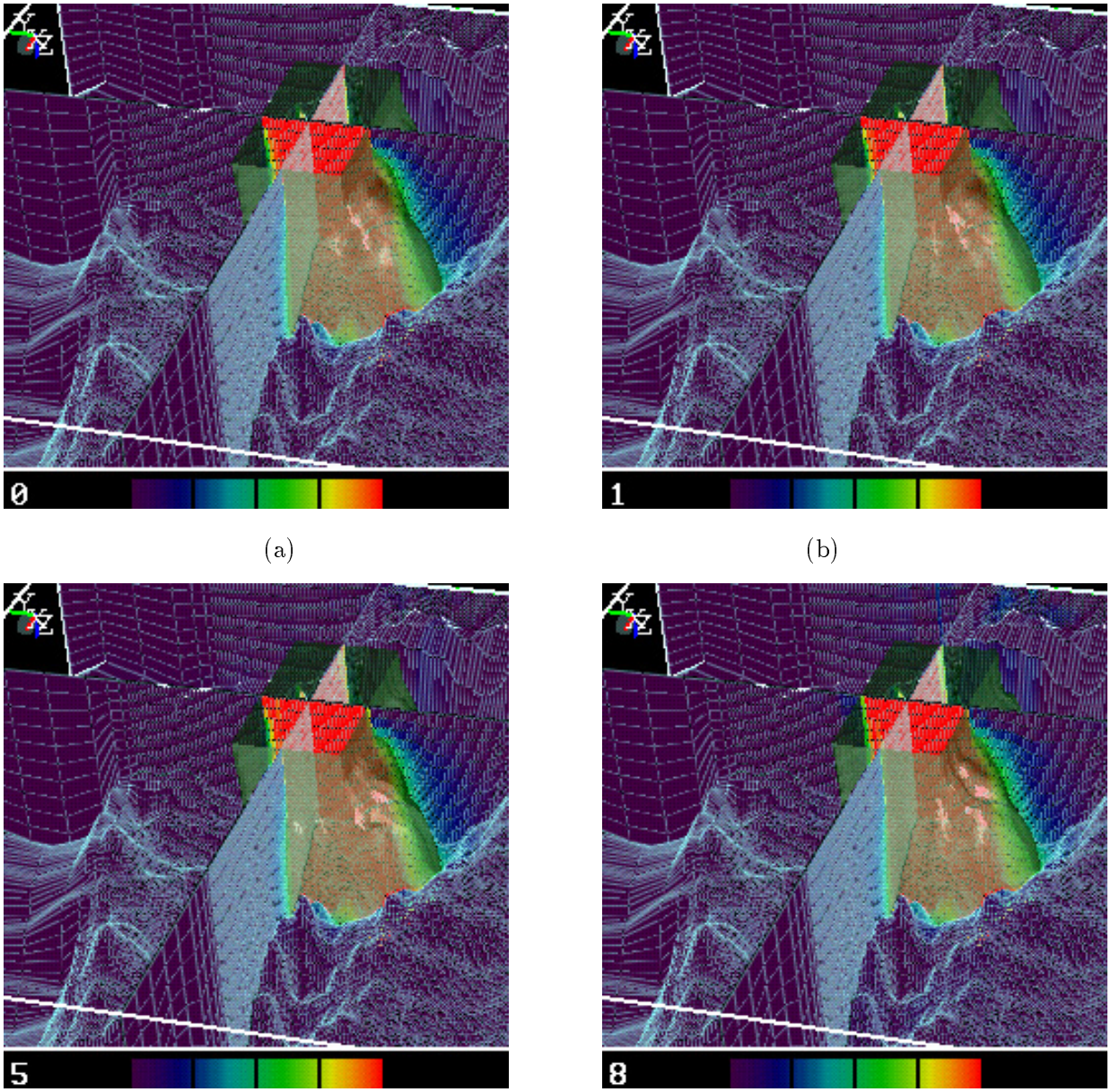
(c)

(d)

Figure 1. Original test images used in computational experiments: (a) Lena, (b) Barbara, (c) City, and (d) Destroyer.

Figure 2. Compression of Lena Image with a compression rate of 128 comparing methods: (a) original image, (b) Shapiro (South Carolina implementation), (c) Hyperbolic wavelets with interleaving, (d) Regular wavelets with interleaving, (e) Hyperbolic wavelets with bit-stream and (f) Regular wavelets with bit-stream.

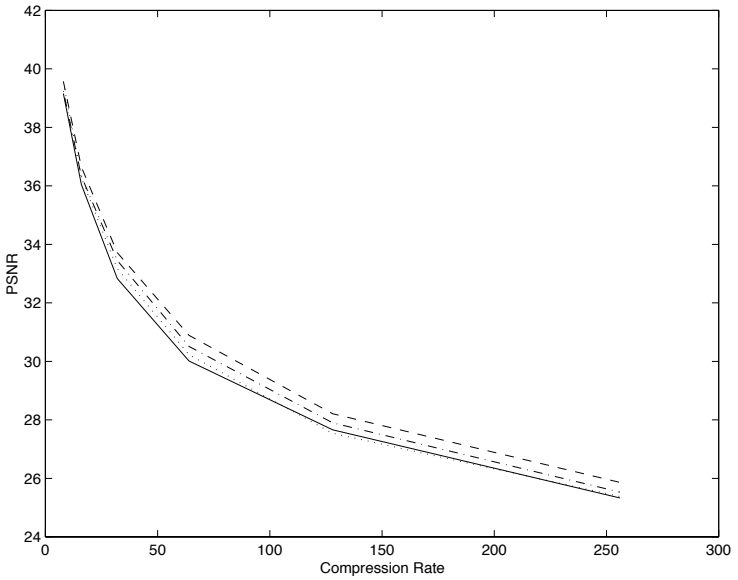
Figure 3. Compression of Barbra Image with a compression rate of 128 comparing methods: (a) original image, (b) Shapiro (South Carolina implementation), (c) Hyperbolic wavelets with interleaving, (d) Regular wavelets with interleaving, (e) Hyperbolic wavelets with bit-stream and (f) Regular wavelets with bit-stream.



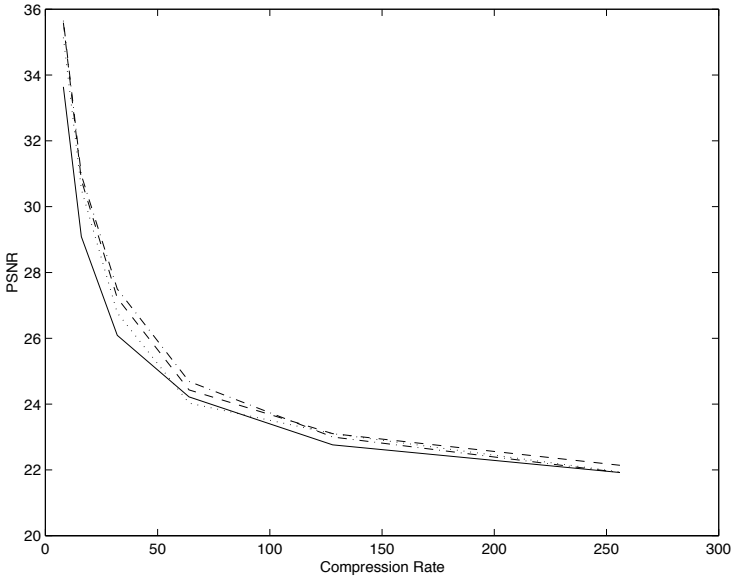
(a) (b)

(c) (d)

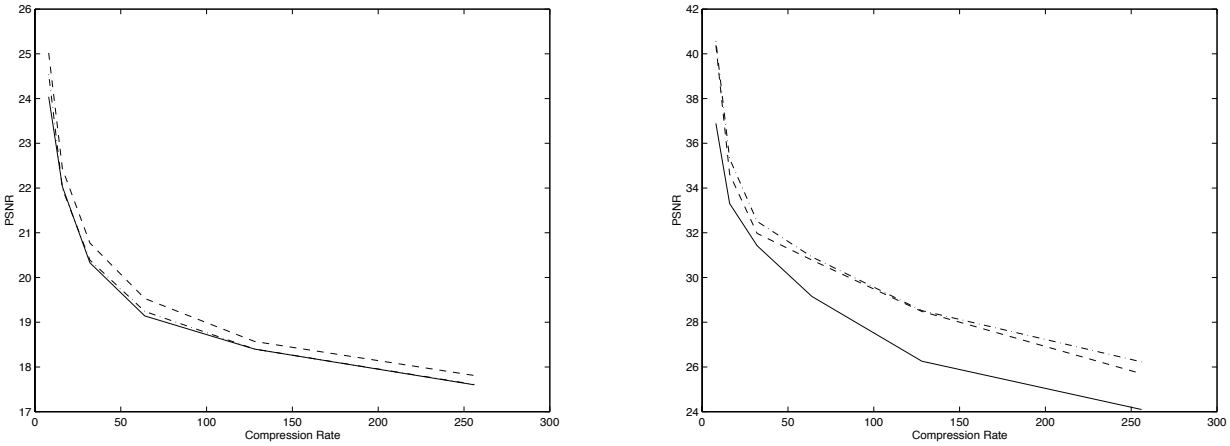
Figure 4. Comparision of 3D data compression, using the interleaving encoding, to original 3D scalar field with varying compression rates: (a) Original data, (b) 172 times compression, (c) 493 times compression, and (d) 1200 times compression. All images are rendered with the same orthogonal slices and an isosurface of .3 of maximum concentration percentage.



Plot 1. Plots of PSNR (Peak Signal to Noise Ratio) as a function of the compression rate for the **Lena** test image using Shapiro's EZM algorithm (dotted), the USC version of Shapiro EZM algorithm (solid) and the elementary encoding applied to regular (dashed) and hyperbolic (dash-dotted) wavelets.



Plot 2. Plots of PSNR (Peak Signal to Noise Ratio) as a function of the compression rate for the **Barbara** test image using Shapiro's EZM algorithm (dotted), the USC version of Shapiro EZM algorithm (solid) and the elementary encoding applied to regular (dashed) and hyperbolic (dash-dotted) wavelets.



(a) (b)

Plot 3. Plots of PSNR (Peak Signal to Noise Ratio) as a function of the compression rate using the USC version of Shapiro's method (solid) and our encoding algorithm applied to regular (dashed) and hyperbolic (dash-dotted) wavelets for: (a) **City** image and (b) **Destroyer** image.