# A Low-Complexity Modeling Approach for Embedded Coding of Wavelet Coefficients

Erik Ordentlich, Marcelo Weinberger, Gadiel Seroussi
Computer Systems Laboratory

HPL-97-150
December, 1997

progressive image compression,
Laplacian density,
run-length coding,
rate distortion

We present a new low-complexity method for modeling and coding the bitplanes of a wavelet-transformed image in a fully embedded fashion. The scheme uses a simple ordering model for embedding, based on the principle that coefficient bits that are likely to reduce the distortion the most should be described first in the encoded bitstream. The ordering model is tied to a conditioning model in a way that deinterleaves the conditioned subsequences of coefficient bits, making them amenable to coding with a very simple, adaptive elementary Golomb code. The proposed scheme, without relying on zerotrees or arithmetic coding, attains PSNR vs. bit rate performance superior to that of SPIHT, and competitive with its arithmetic coding variant, SPIHT-AC.

Internal Accession Date Only

© Copyright Hewlett-Packard Company 1997

# 1 Introduction

Progressive image compression refers to the encoding of an image into a bitstream that can be parsed efficiently to obtain lower rate and lower resolution descriptions of the image. Such descriptions are said to be SNR (signal to noise ratio) and resolution scalable. Most state-of-the-art progressive image compression schemes are based on a *wavelet transform* followed by quantization of the transform coefficients. The multi-resolution nature of the wavelet transform leads to resolution scalability in a straightforward way. In this paper we focus on SNR (signal to noise ratio) scalability where the goal is to produce a so called *embedded* bitstream which has the property that the prefixes of the bitstream yield a continuum of lower rate descriptions of the image at the highest possible levels of quality.

An attractive approach for achieving SNR scalability within the wavelet transform framework is to describe the coefficients sequentially by *bitplanes*, from most significant to least. Here, sequential description by bitplanes plays a dual role in quantization and progressive transmission, by realizing a sequence of successively refined uniform quantizers. Describing each coefficient in bitplane order results in a maximal drop of distortion per bit of description. This approach was first proposed in [1], a paper that is often considered a milestone in the development of progressive image compression and wavelet-based image compression as a whole. Since then many of the advances in progressive image compression have centered around improvements to the *lossless* compression of the *binary* bitplane data, and two basic approaches have emerged. These are the zerotree-based approach of [1], which seeks to achieve compression through a carefully designed ordering of coefficients within bitplanes, and the more traditional context modeling approach rooted in the classical modeling/coding dichotomy [2]. The latter was originally investigated in [3] and the best results to date have been reported in [5] using a fairly complex set of contexts. The most significant enhancement of the zerotree approach since [1] is the SPIHT algorithm of [4].

The original motivation for describing the coefficients in bitplane order, namely maximal drop in distortion per bit of description, should also be the guiding principle in ordering bits coming from different coefficients, i.e., information that is likely to reduce distortion the most should be described first. Thus, probabilistic modeling is just as important for optimal embedding as it is for coding. We refer to this observation as the *embedding principle*. In fact, this principle could lead to orderings that do not preserve bitplane sequence, with bits from a lower bitplane being described before a higher bitplane is completed. This approach was investigated in [7], leading to a fairly effective but complex coding algorithm.

In this work, we take the low complexity road, and we build on the embedding principle to construct a simple yet very efficient algorithm for bitplane compression. We observe that the interaction between embedding and conditioning models is a complex one, since reordering for embedding clearly affects causality relations inherent to conditioning models. Therefore, lacking a tractable way to optimize this interaction, we adopt a "greedy" approach that gives preference to ordering, unlike in [6], where the opposite preference is given. This leads to a two-tiered ordering/conditioning model, which results in the generation of independent subsequences of coefficient bits, one subsequence associated with each ordering context. By virtue of the ordering, each subsequence is available for encoding as a contiguous block, thus allowing for the use of a very simple adaptive code, referred to as an *elementary Golomb code*. Thus, in effect, when the ordering model is regarded also as a conditioning model, we have deinterleaved the conditioned subsequences, making an arithmetic coder less of a necessity, a fact that is formally justified in the sequel.

Moreover, as we use a very simple model, our scheme overcomes some of the limitations of the zerotree and traditional context modeling approaches. In particular, the zerotree approach suffers from an entanglement of modeling, coding, and algorithmic issues that makes it difficult to engineer for different applications, some of which might find the necessary data structures problematic in certain memory-limited environments. Also, the zerotree approach does not directly yield resolution-scalable bitstreams. The traditional context modeling approach, in turn, suffers from its reliance on the relatively complex machinery of adaptive arithmetic coding. Despite its simplicity, and without relying on zerotrees or arithmetic coding, our scheme is fully embedded and attains SNR vs. bitrate performance that is superior to that of SPIHT and is competitive with its arithmetic coding version SPIHT-AC.

Section 2 elaborates on probabilistic aspects of the embedding principle. Section 3 then describes the image compression algorithm. This is followed by a redundancy analysis of elementary Golomb codes in Section 4. Section 5 presents experimental compression results and comparisons with SPIHT and SPIHT-AC. Section 6 highlights the low-complexity features of the proposed image compression algorithm, and Section 7 concludes the paper by mentioning some potential enhancements to the algorithm.

## 2 The embedding principle

We first review some of the basic elements of bitplane coding of wavelet transform coefficients. The first step is the application of a wavelet transform to the image data to obtain a set of transform coefficients $x_i$, where the index $i$ denotes some scanning order of the transformed image, and its range is omitted for the sake of conciseness. Given $\Delta > 0$, the $x_i$ are then quantized according to $Q_i = \mathrm{sgn}(x_i)\lfloor |x_i|/\Delta \rfloor$. Let $b_{m,i}b_{m-1,i}\ldots b_{0,i}$ denote $|Q_i|$ in binary where $m$ is the smallest integer satisfying $2^{m+1} > |Q_i|$ for all $i$. The goal is to produce a compressed bitstream which describes the quantized coefficients $Q_i$ in bitplane order starting with $b_{m,i}$. That is, $b_{m,i}$ is described for all $i$ before the $b_{m-1,i}$ and so on. Describing the $Q_i$ in this fashion is equivalent to quantizing $x_i$ by a series of successively refined uniform quantizers with step sizes $\Delta_n \stackrel{\triangle}{=} 2^n\Delta,\ n = m, m-1, \ldots, 0$, where $\Delta$ is the finest quantization step size. Notice that we have a priori adopted the practice of ordering by bitplane and that therefore we confine the application of the embedding principle to ordering within complete bitplanes. Let $\mathbf{b}_{n+1,i}$ denote the vector of bitplanes $b_{m,i}\ldots b_{n+1,i}$. Two possibilities exist for encoding each $b_{n,i}$ depending on the values of the previous bitplanes: either $\mathbf{b}_{n+1,i} = 0$, in which case $b_{n,i}$ is called a *significance bit*, or $\mathbf{b}_{n+1,i} \neq 0$, in which case $b_{n,i}$ is called a *refinement bit* [1]. The encoding of significance bits is accompanied by the encoding of the sign of the corresponding coefficient if $b_{n,i} = 1$.

**Optimal ordering.** The embedding principle is best appreciated by way of the following simple scenario. Consider the decomposition of a set $\mathcal{S}$ of coefficient bitplanes into two subsequences $S_j$, $j = 1, 2$, of respective lengths $n_j$. Let $D_j$ be the expected reduction in distortion induced by learning a bit from $S_j$ and let $R_j$ be the expected rate (measured in bits per pixel) required to encode this bit. Suppose further that the sequences of distortion reductions and instantaneous rates are independent and identically distributed (i.i.d.). Under these assumptions, which sequence should be encoded first, $S_1$ or $S_2$? We can answer this question by considering the distortion versus rate behavior for each possibility. We will denote the random behavior of the normalized distortion (distortion per pixel) as a function of rate by $D(R)$. This notation should not be confused with the conventional distortion-rate function which is deterministic. To distinguish the alternatives in question, the function $D(R)$ is denoted by $D^{(1)}(R)$, when $S_1$ is encoded before $S_2$, and by $D^{(2)}(R)$ otherwise.

Let $\alpha_j = n_j/N$, where $N$ is the number of pixels in the image, and let $R_0$ and $D_0$ respectively be the rate (per pixel) and normalized distortion of the description prior to encoding these subsequences, Then, the i.i.d. assumption and the law of large numbers imply that with very high probability $D^{(1)}(R)$, considered over the range $R \in (R_0, R_0 + \alpha_1 R_1 + \alpha_2 R_2)$, will be very close to $D_0 - (D_1/R_1)(R - R_0)$ for $R \in (R_0, R_0 + \alpha_1 R_1)$, and to $D_0 - \alpha_1 D_1 - (D_2/R_2)(R - R_0 - \alpha_1 R_1)$ for $R \in (R_0 + \alpha_1 R, R_0 + \alpha_1 R_1 + \alpha_2 R_2)$. Likewise, a similar approximation to $D^{(2)}(R)$ is obtained by interchanging $R_1$ and $R_2$, $D_1$ and $D_2$, and $\alpha_1$ and $\alpha_2$ in these expressions. Thus, irrespective of the order, the final description ends with high probability at distortion $D_0 - \alpha_1 D_1 - \alpha_2 D_2$ and rate $R_0 + \alpha_1 R_1 + \alpha_2 R_2$. It is clear, however, that if $D_1/R_1 > D_2/R_2$ the $D^{(1)}(R)$ curve (piecewise linear) lies strictly below the alternative curve $D^{(2)}(R)$ for $R \in (R_0, R_0 + \alpha_1 R_1 + \alpha_2 R_2)$ and the reverse is true if $D_2/R_2 > D_1/R_1$. Moreover, it is easy to show that for any strategy based on interleaving $S_1$ and $S_2$, the $D(R)$ curve will, with high probability, lie strictly between the piecewise linear curves $D^{(1)}(R)$ and $D^{(2)}(R)$. Clearly then, the sequence maximizing $D_j/R_j$ should be encoded first. [1]

**Context modeling for ordering and coding.** The above scenario tacitly assumes that the decomposition of $\mathcal{S}$ into $S_1$ and $S_2$ is known a priori to the encoder and the decoder. In practice, however, this decomposition must be based on previously encoded information and the most effective mechanism for isolating sequences of symbols with similar (conditional) statistics is context modeling. Depending on the choice of contexts, the fundamental decodability constraint that context determining information be transmitted first may significantly cut down on the flexibility in optimizing the encoding order for embedding. For example, if the context for determining whether a particular coefficient belongs to $S_1$ or $S_2$ contains

[1] In [6], the expectation of the ratio of distortion reduction to bitrate rather than the ratio of expectations is proposed as the ordering criterion. Clearly, it is the latter criterion which is more relevant from an operational point of view.

coefficients from $S_2$, these latter coefficients must be encoded first, irrespective of embedding considerations. Thus, the encoding of $S_1$ and $S_2$ may have to be interleaved based on the causal succession of contexts. Such a choice of contexts might be optimal for the overall compression of $S_1$ and $S_2$ but the resulting interleaved encoding leads to suboptimal embedding. As it seems intractable to find the best combination, the compromise we strike is to establish a *two-stage hierarchy* of contexts. The first stage of this hierarchy is a function only of previously encoded information outside of $\mathcal{S}$ and is used to determine $S_1$ and $S_2$ (and possibly more sequences) for ordering purposes. The second stage reverts to ordinary context modeling constrained by the selected ordering. It turns out that when we specialize this approach to the bitplane problem, where $\mathcal{S}$ corresponds to an entire bitplane, the bulk of the compression gain for most natural images is obtained through the first stage in the hierarchy. This fortuitous result opens the door to very low-complexity compression since each $S_j$ can be compressed independently under the i.i.d. assumption, a task for which full fledged adaptive arithmetic coding might be less of a necessity, as shown in Section 4.

**Application of the embedding principle.** Let us specialize the discussion further to the problem of encoding the $n$-th bitplane $b_{n,i}$ after $\mathbf{b}_{n+1,i}$ have been encoded for all $i$. The set $\mathcal{S}$ of coefficient bitplanes corresponds to all of the $\{b_{n,i}\}$. We assume the average squared error distortion measure and that the wavelet transform is sufficiently near orthogonal that reductions in distortion in the transform domain correspond to equivalent reductions in distortion in the image domain, an important consideration in applying the above scenario.

The first stage of the two-stage context hierarchy described above decomposes $\mathcal{S}$ only on the basis of the values of $\mathbf{b}_{n+1,i}$ into a collection of subsequences $S_j$ of coefficients having similar distortion reduction per rate of description ( $D_j/R_j$ ) statistics. The sequences are then encoded separately in order of decreasing values of $D_j/R_j$. Next, we investigate these statistics and show that under general conditions the embedding principle leads to ordering significance bits according to their probability of being one. The analysis also finds conditions under which significance bits should be encoded before refinement bits.

We start with the refinement bits. Prior to learning a refinement bit $b_{n,i}$ it is known that $x_i$ belongs to an interval of width $\Delta_{n+1} = 2^{n+1}\Delta$. Learning $b_{n,i}$ halves the width of this uncertainty interval to $\Delta_n$. Let $\mathcal{I}$ denote the conditioning information $\{\mathbf{b}_{n+1,i}\,, \mathrm{sgn}(x_i)\}$, and let $\mathcal{I}^{(k)}$ denote the refined information $\{b_{n,i} = k, \mathbf{b}_{n+1,i}\,, \mathrm{sgn}(x_i)\}$ for $k = 0, 1$. Assuming reconstruction at the mean, it can be shown that the expected drop in distortion due to refinement is

$$D_{ref} = E^2(x_i|\mathcal{I}^{(0)})(1 - p_{ref}) + E^2(x_i|\mathcal{I}^{(1)})p_{ref} - E^2(x_i|\mathcal{I})$$

where $p_{ref} = \mathrm{Prob}(b_{n,i} = 1|\mathcal{I})$ and $E(\cdot|\cdot)$ denotes conditional expectation. It has been found empirically that, conditioned on $\mathcal{I}$ (and no additional context), if $\mathbf{b}_{n+1,i} \neq 0$ then $x_i$ is nearly uniformly distributed, namely $p_{ref} \approx 1/2$. In this case $D_{ref} = \Delta_n^2/4$ and $R_{ref}$, the average rate of description per bit, is $H(1/2) = 1$, where $H(\cdot)$ is the binary entropy function.

Prior to learning a significance bit $b_{n,i}$, in turn, it is known that $x_i \in (-\Delta_{n+1}, \Delta_{n+1})$. This is refined to $x_i \in (-\Delta_n, \Delta_n)$ if $b_{n,i} = 0$, and to $x_i \in \mathrm{sgn}(x_i)[\Delta_n, \Delta_{n+1})$ otherwise, where $\Delta_n = \Delta_{n+1}/2$. Letting $p_j = \mathrm{Prob}(b_{n,i} = 1 | \mathbf{b}_{n+1} = 0)$ for significance bits from subsequence $S_j$, and assuming that the coefficients from this subsequence are distributed according to a density $f_j$ which is symmetric around zero, it can be shown that for reconstruction at the mean the average distortion reduction is $D_j = p_j E^2_{f_j}(x_i | \mathcal{I}^{(1)})$, where again $\mathcal{I}^{(1)}$ denotes the conditioning information $(b_{n,i} = 1, \mathbf{b}_{n+1,i} = 0, \mathrm{sgn}(x_i))$. Furthermore, assuming efficient compression of $b_{n,i}$ and no compression of the sign ( as the additional savings do not justify the increase in complexity), the average rate of description per bit is $R_j = p_j + H(p_j)$, where $H(p_j)$ bits are required on average for the $b_{n,i}$ and one bit is required with probability $p_j$ for the sign. Hence, we have the following proposition.

**Proposition 1** *If the coefficients corresponding to significance bits $b_{n,i}$ from sequence $S_j$ are distributed according to the symmetric density $f_j$, then the average distortion reduction per bit of description of the $b_{n,i}$ is given by*

$$\frac{D_j}{R_j} = \frac{p_j E^2_{f_j}(x_i | \mathcal{I}^{(1)})}{p_j + H(p_j)}.$$

It can be shown that the quantity $p/(p + H(p))$ is monotonically increasing in $p$. Therefore, if the $f_j$ are such that $E_{f_j}(x_i | \mathcal{I}^{(1)})$ is non-decreasing in $p_j$, then $p_j$ serves as a convenient criterion for ordering sequences of significance bits. In particular, the sequences $S_j$ with larger $p_j$ should be encoded first. It turns out that the family of generalized Gaussian distributions (with fixed exponent), and, in particular, the family of Laplacian distributions, has this property. Closed form expressions of $E_{f_j}$ can be derived for this family as a function of $p_j$.[2] The most general conditions under which a family of distributions exhibits the monotonicity property are under investigation.

If, as in [7], we use the simplifying assumption that $x_i$ conditioned on $\mathcal{I}^{(1)}$ is uniformly distributed, in which case $D_j/R_j$ reduces to

$$\frac{D_j}{R_j} = \frac{9\Delta_n^2 p_j}{4(p_j + H(p_j))},$$

the monotonicity property applies, since $E_{f_j}(x_i | \mathcal{I}^{(1)})$ is independent of $p_j$.

As for refinement bits versus significance bits, comparing $D_j/R_j$ to $D_{ref}/R_{ref}$ under the uniform assumption for each case, one finds that $D_j/R_j$ exceeds $D_{ref}/R_{ref}$ for $p_j$ larger than about .01, indicating that for these values of $p_j$ significance bits should be encoded before refinement bits. Under the Laplacian assumption on the distribution of the significance bits,

[2] A similar analysis can be carried out for reconstruction at the midpoint of the uncertainty interval, as opposed to the mean.

this threshold is somewhat larger, which suggests the possibility that it may be beneficial to encode refinement bits before subsequences of significance bits with very small $p_j$.

To summarize, the above analysis motivates the use of $p_j$ as a criterion for ordering the encoding of significance bits as well as the already common practice of encoding significance information before refinement information, except when $p_j$ is very small. Notice that the ordering problem is thus simpler than the coding problem: only the relative ordering of the $p_j$'s must be estimated or guessed and not the actual values themselves.

# 3 Algorithm

In this section we describe an algorithm for coding bitplanes of wavelet coefficients that is motivated by the consideration of the previous section, and by a low-complexity goal. As a result of the above discussion, we build a hierarchy of a few simple contexts, the first level of which classifies the $b_{n,i}$ into subsequences $S_{n,1}, \ldots, S_{n,K}$ based only on bitplanes $m$ through $n+1$. We model the subsequences $S_{n,j}$ as being mutually independent, encode them separately, and order the resulting bitstreams based on an anticipated relative ordering of $D_j/R_j$.

We now specify the subsequence classification. Let coefficient $f(i)$ be the parent of coefficient $i$ where we refer to the usual parent-child relationship [1] among wavelet coefficients,[3] and let $N(i)$ denote an as yet unspecified collection of spatially contiguous neighbors of $i$ from the same frequency band as $i$. We determine four subsequences $S_{n,j},\ j = 1, \ldots, 4$, which are, respectively:

1. Non-zero neighbor subsequence: all coefficients $i$ with $\mathbf{b}_{n+1,i} = 0$ and $\mathbf{b}_{n+1,k} \neq 0$ for at least one $k$ in $N(i)$.
2. Non-zero parent subsequence: all coefficients $i$ with $\mathbf{b}_{n+1,i} = 0$, $\mathbf{b}_{n+1,k} = 0$ for all $k$ in $N(i)$, and $\mathbf{b}_{n+1,f(i)} \neq 0$.
3. Run-subsequence: consists of all coefficients $i$ with $\mathbf{b}_{n+1,i} = 0$, $\mathbf{b}_{n+1,k} = 0$ for all $k$ in $N(i)$, and $\mathbf{b}_{n+1,f(i)} = 0$.
4. Refinement subsequence: all coefficients $i$ with $\mathbf{b}_{n+1,i} \neq 0$.

Let $p_{n,j} \overset{\triangle}{=} \mathrm{Prob}(b_{n,i} = 1)$ for subsequence $S_{n,j}$. Subsequence $S_{n,3}$ is referred to as the run-subsequence as we expect $p_{n,3}$ to be very small and, therefore, $S_{n,3}$ will exhibit long runs of zeros. In fact, we would expect a priori that $p_{n,1} \geq p_{n,2} \geq p_{n,3}$. This was indeed found to be the case for all the images examined. Thus, Proposition 1 and the ensuing discussion suggest

[3] The parent of coefficient $i$ is that coefficient from the next decomposition level in the same spatial orientation which has a basis function maximally overlapping that of coefficient $i$. This leads to each parent having four children.

that $S_{n,j}$ be encoded in the order $S_{n,1}, S_{n,2}$, and $S_{n,3}$, followed by the refinement bits $S_{n,4}$. It turns out that in some cases, especially at low bitrates, encoding $S_{n,4}$ before $S_{n,3}$ leads to better embedding, as suggested by the small value of $p_{n,3}$ and the results of the previous section. For a negligible compression overhead and a modest increase in complexity, the encoding order can be made adaptive based on learning the relative ordering of the $p_{n,j}$.

For the second level of the context hierarchy, it was found to be beneficial for some images to further decompose the run subsequence $S_{n,3}$ within each level of the wavelet decomposition based on $b_{n,f(i)}$, the value of the $n$-th bitplane of the parent of $i$. Thus, $S_{n,3}^{(l,0)}$ consists of all coefficients in $S_{n,3}$ and level $l$ of the wavelet decomposition with $b_{n,f(i)} = 0$, and $S_{n,1}^{(l,3)}$ consists of those coefficients with $b_{n,f(i)} = 1$. We have added the superscript $l$ to emphasize that causality constraints require that these subsequences be encoded in order of decreasing $l$.

Once identified, the subsequences are compressed using the adaptive codes of Section 4 except for $S_{n,4}$, the refinement subsequence, which is appended to the bitstream uncoded. The adaptive codes are built on an extended alphabet in which the value $b_{n,i} = 1$ for a significance bit always marks the end of a symbol. The sign of the coefficient whose one-valued bit terminated the extended symbol is appended uncoded to the bitstream following the encoding of the symbol.

Specific features incorporated into the coder used to obtain the figures in Table 1 are:

1. 9-7 biorthogonal wavelet decomposition [8] with 6 levels.

2. The neighborhood $N(i)$ of the $i$-th coefficient consists of the eight coefficients spatially adjacent to $i$.

3. The scanning of the subsequences is as follows. Each band at level $l$ is partitioned into macro-blocks of size $2^{7-l} \times 2^{7-l}$. Within each macro-block, the coordinates of the $i$-th coefficient in the scan are obtained by deinterleaving the odd and even bits of $i$. This is sometimes called the zig-zag scan. Within each level the order of the bands with respect to orientation is high horizontal-low vertical, low vertical-high horizontal, high vertical-high horizontal. The levels are scanned from highest level (lowest frequency) to lowest level (highest frequency).

4. The mean of the lowest frequency band coefficients is subtracted prior to quantization. The bitplanes of these coefficients are packed uncoded along with the sign information when a coefficient bitplane is non-zero for the first time.

5. The reconstruction value of a coefficient is always in the middle of the most recently available uncertainty interval.

Separate bitstreams are generated for each bitplane and each subsequence of coefficients within each bitplane. The sub-bitstreams can be rearranged offline (on byte boundaries) for optimal embedding.

**Complexity tradeoffs.** Lower complexity variants that we investigated include a simplified subsequence classification based on blocks. The idea is to first group the coefficients into 2x2 blocks of spatially contiguous coefficients having the same parent. The classification rule is then the same as above except $N(i)$, the neighborhood of the $i$-th coefficient, now refers to the other 3 coefficients in $i$'s block. This reduces the memory access required for subsequence classification to retreiving the parent and then one access per coefficient (as opposed to at least 3 per coefficient in an efficient implementation of the above scheme). This simplification has a cost of about .1dB on most natural images and a somewhat higher cost for some artificial images such as compound documents. We also investigated changing the scanning of the subsequences to raster, as opposed to the scan order of item 3 above. Raster scan may be necessary in memory-limited applications. The cost of using a raster scan can vary, but it typically ranges between 0 and .1 dB. For artificial images the difference is often more significant both positively and negatively.

**Reconstruction at the mean.** It may be possible to improve the reconstructed image quality by determining the quantizer reconstruction values adaptively, as opposed to always using the midpoint of the uncertainty interval. The minimum squared error reconstruction value for a set of coefficients known to lie in a quantization interval is well known to be the sample mean: however, the decoder must be informed of this quantity, or must estimate it from previously decoded information. We now describe a low-complexity scheme for adapting the reconstruction values of significant coefficients based on estimating the sample means at the encoder.

To reduce the number of quantities which must be estimated (and hence save on side-information and computation) we again rely on the observation that the conditional empirical distributions of wavelet transform coefficients are well approximated by Laplacian distributions. Thus, the model we assume is that wavelet transform coefficients are distributed according to a mixture of Laplacians, where one component of the mixture dominates in each of the sets $\pm[0, \Delta_0), \pm[\Delta_0, \Delta_1), \pm[\Delta_1, \Delta_2), \ldots, \pm[\Delta_m, \Delta_{m+1})$, where, as above, $\Delta_n = 2^n \Delta$. Let $I_n$ denote the set $\pm[\Delta_n, \Delta_{n+1})$. The idea is that the encoder determines the Laplacian density that best approximates the distribution of wavelet coefficients falling in each $I_n$. A sufficient statistic for determining this density is $\mu_n$, the sample mean of the absolute values of the coefficients belonging to $I_n$. Note that, since $I_n$ contains those coefficients for which the $n$-th bitplane is the *first* non-zero bitplane, $\mu_n$ need not be specified until the $n$-th bitplane is encoded: prior to this all of these coefficients are quantized and reconstructed at zero. Thus, each $\mu_n$ is added to the bitstream as side-information just prior to the encoding of the corresponding bitplane.

Fix a set $I_n$, and let $I_n^+$ be the postive half of $I_n$. Bitplanes $n$ through $j$ ($j \leq n$, as bitplanes are described in decreasing order) refine $I_n^+$ into $2^{n-j}$ sub-intervals indexed by $k$, and reconstruction values $r_{k,j,n}$ must be determined for each sub-interval. For $j = n$ there is only a single sub-interval, namely $I_n^+$, and the decoder sets $r_{0,n,n} = \mu_n$: this is the minimum squared error optimum value, assuming the coefficients are symmetrically distributed about zero. For $j < n$, the decoder determines the $r_{k,j,n}$, as the conditional expected values, relative to the corresponding sub-intervals, of the Laplacian density having $\mu_n$ as its conditional expected absolute value relative to $I_n$. An interesting and useful property of the Laplacian (and exponential) density is that the conditional densities relative to two intervals having the same length and lying on the same side of the origin are simply shifted versions of the same density. This implies that the difference between the reconstruction values $r_{j,k,n}$ and the left edge of each sub-interval in $I_n^+$ is a constant $\delta_{j,n}$. It suffices, therefore, to compute only the $\delta_{j,n}$, and then to add these offsets to the readily computed sub-interval edges to obtain the reconstruction values. The Laplacian assumption implies that the negative half of $I_n$ is treated symmetrically.

It can be shown that a real parameter $x$ relates $\mu_n$ and $\delta_{j,n}$ through the equations

$$\frac{\mu_n - \Delta_n}{\Delta_n} = \frac{1}{\ln x} - \frac{1}{x-1} \quad , \tag{1}$$

and

$$\frac{\delta_{j,n}}{\Delta_j} = \frac{1}{2^{j-n} \ln x} - \frac{1}{x^{2^{j-n}} - 1} \quad .$$

Solving these equations for $\delta_{j,n}$ in terms of $\mu_n$ is only possible numerically. In practice, this computation can be accelerated using a lookup table containing values of the quantity $\delta_{j,n}/\Delta_j$, and indexed by quantized values of the quantity $\min((\mu_n - \Delta_n)/\Delta_n, 1/2)$ and $n - j$. The minimum is required since the right side of (1) is bounded above by 1/2. Intuitively, this arises from the fact that the conditional mean of a Laplacian density relative to any positive interval is smaller than the midpoint of the interval.

We remark that there are many possible extensions to the above parameterized reconstruction value scheme. For example, we can increase the number of sub-intervals for which explicit sample means are encoded, and we can further group coefficients based on blocks, bands, and/or contexts, and encode sample means for each group.

# 4 Elementary Golomb codes

By the modeling considerations discussed in sections 2 and 3, our goal is to encode independent binary subsequences, which are modeled as i.i.d. using a low-complexity adaptive code. It was found empirically that for $S_{n,1}, S_{n,2}$, and $S_{n,3}$, the subsequences that are actually coded, the probability $p_{n,j}$ of a one satisfies $p_{n,j} < .5$. Therefore, in the subsequent analysis of binary data compression, we can assume that in all cases the probability $q$ of a zero satisfies $q > 0.5$. For a positive integer parameter $m$, let $EG_m$ denote a variable-to-variable length code defined over the extended alphabet $\{1, 01, 001, \ldots, 0^{m-1}1, 0^m\}$, where the notation $0^\ell$ denotes a sequence of $\ell$ zeros. Under $EG_m$, the extended symbol $0^m$ is encoded with a 0, while $0^\ell 1$, $0 \leq \ell < m$, is encoded with a 1 followed by the binary representation of $\ell$ (using $\lfloor \log m \rfloor$ bits if $\ell < 2^{\lceil \log m \rceil} - m$ and $\lceil \log m \rceil$ bits otherwise, where logarithms are taken to the base 2). By considering a concatenation of extended input symbols, it is easy to see that $EG_m$ is equivalent to a *Golomb* code [9] of order $m$ applied to the sequence of *zero-run lengths*. However, $EG_m$ is defined over a finite alphabet which provides for better adaptation, as shown below. In fact, these codes were introduced in [10], [11], and [12] in the context of adaptive run-length coding. We will refer to $EG_m$ as an *elementary Golomb code* of order $m$.

In this section we present new properties of elementary Golomb codes which provide insight into their well-known efficiency for encoding i.i.d. binary sequences over a surprisingly wide range of values of $q$. We also elaborate on adaptive strategies for the case in which $m$ is a power of 2. First, notice that by the equivalence with Golomb codes, we can apply results from [13] to show that the order $m$ of the best elementary Golomb code for a given value of $q$ is the unique positive integer satisfying

$$q^m + q^{m+1} \leq 1 < q^m + q^{m-1} \ . \tag{2}$$

**Proposition 2** *For $q$ and $m$ satisfying (2), the Huffman code for a Tunstall extension with $m$ symbols constructed over the binary alphabet is $EG_m$.*

Since a Tunstall extension [14] minimizes the bound $1/E[L]$ on the redundancy of a Huffman code constructed over symbols of expected length $E[L]$, Tunstall-Huffman combinations have been proposed as good approximations for the (open) problem of finding the best *dual-tree* code [15, 16]. Thus, Proposition 2 states that, in a sense, $EG_m$ is the best variable-to-variable code with $m$ symbols for the range of probabilities for which $m$ is optimal.

When $q$ is unknown *a priori*, elementary Golomb codes are superior to Golomb codes in that the value of $m$ can be adapted *within a run*, based on the current estimate of $q$. However, to design an adaptive strategy for $EG_m$ based on (2) can be a complex procedure. By reducing the family of codes to the case where $m = 2^g$, the redundancy is still very small for the ranges of interest, while the adaptation turns out to be extremely simple. In addition, the code words for the extended symbols ending in a one are all $g+1$ bits long and can be easily implemented.

**Proposition 3** *Let $m = 2^g$. The order $m$ of the best elementary (power-of-2) Golomb code for a given value of $q$ is the unique positive integer satisfying*

$$q^m \leq \phi < q^{m/2} \tag{3}$$

*where $\phi \triangleq (\sqrt{5} - 1)/2$ (inverse of the golden ratio).*

By Proposition 3, the values of $q$ for which there is a transition point between optimal codes $EG_{2^g}$ are $\phi$, $\phi^{1/2}$, $\phi^{1/4}$, $\ldots$, $\phi^{2^{-\ell}}$, $\ldots$ . By use of tools developed in [17] it can be shown that these values are also local maxima for the minimum relative redundancy of the codes. These maxima are decreasing with the order of the codes, so that the worst case redundancy is attained at $q = \phi \approx 0.618$ and it equals $(1/H(\phi)) - 1 \approx 4\%$.

As for adaptivity, notice that, with $\psi$ denoting the golden ratio, the transition points satisfy

$$\frac{q}{1-q} = \frac{1}{\psi^{2^{-g}} - 1} \triangleq \frac{2^g}{\ln \psi} - \frac{1}{2} + \gamma(g) \tag{4}$$

where $\gamma(g)$ can be shown to be a *decreasing* function of $g$ that ranges between $\psi + (1/2) - (1/\ln \psi) \approx 0.04$ $(g = 0)$, and $0$ $(g \to \infty)$. Since $\psi \approx 1.618$ and $1/\ln \psi \approx 2.078$, (4) implies that $q/(1-q)$ is within 4% of $2^{g+1} - (1/2) + (1/8)$ for every $g \geq 0$. Since $q/(1-q)$ can be estimated as the ratio of the number of zeros to the number of ones in the sequence, it follows that the optimal $g$ can be adaptively estimated through simple shift and add operations. As in [18], provisions should be taken to implement a forgetting factor for the remote past. More complex adaptation strategies for dual-tree codes are proposed in [19].

The above low-complexity adaptive coding scheme is used for all the mutually independent subsequences described in Section 3, except for $S_{n,3}^{(l,0)}$. In this case, we use a variation of this scheme including a dual mode adaptation with a “short run” mode and a “long run” mode. The coder always starts up in the short run mode, and when a run of zeros longer than a pre-set constant (say, 8) is encountered a transition is made into the long run mode which completes the encoding of the run. This method can be viewed as a simple two-state variation of elementary Golomb codes, for cases where an i.i.d. model is inappropriate.

# 5 Results

Table 1 shows PSNR[4] figures for the proposed low complexity coder on a sample of the 8bpp luminance images from the ISO/JPEG2000 test set. The three rows of PSNR’s for each image are respectively for the proposed coder, SPIHT-AC, and SPIHT. The numbers for SPIHT-AC and SPIHT were obtained using the executables kindly made available by W.A. Pearlman on his website. As can be seen, our coder rarely performs worse than .1dB below SPIHT-AC, and it is typically between .3 and .6dB above SPIHT. For example, for .5bpp, our coder’s PSNR is on the average .04dB below that of SPIHT-AC, and .5dB above that of SPIHT.

# 6 Complexity issues

The algorithm of Section 3 has some clear complexity advantages. Arithmetic coding is avoided as is the extensive memory usage of SPIHT and other zerotree-based coders. Unlike

[4] Defined as $10\log_{10}(255^2/\epsilon)$, where $\epsilon$ is the mean of the squares of the pixel errors.

| | | Output bit rate (bpp) | | | | |
|---|---|---|---|---|---|---|
| **Image** | | .0625 | .125 | .25 | .5 | 1.0 |
| **aerial2** | ∗ | 24.55 | 26.44 | 28.45 | 30.57 | 33.28 |
| | † | 24.63 | 26.52 | 28.49 | 30.60 | 33.32 |
| | ‡ | 24.35 | 26.15 | 28.18 | 30.23 | 32.86 |
| **bike** | ∗ | 23.37 | 25.85 | 29.09 | 33.01 | 37.66 |
| | † | 23.44 | 25.89 | 29.12 | 33.01 | 37.70 |
| | ‡ | 22.86 | 25.29 | 28.51 | 32.36 | 36.98 |
| **cafe** | ∗ | 18.97 | 20.61 | 22.95 | 26.39 | 31.59 |
| | † | 18.95 | 20.67 | 23.03 | 26.49 | 31.74 |
| | ‡ | 18.69 | 20.28 | 22.49 | 25.85 | 30.89 |
| **cats** | ∗ | 28.07 | 29.80 | 32.55 | 36.90 | 43.55 |
| | † | 28.15 | 29.88 | 32.62 | 36.89 | 43.60 |
| | ‡ | 27.78 | 29.42 | 32.07 | 36.28 | 42.79 |
| **finger** | ∗ | 20.39 | 21.94 | 24.33 | 27.79 | 31.42 |
| | † | 20.32 | 21.87 | 24.25 | 27.67 | 31.35 |
| | ‡ | 20.10 | 21.65 | 23.84 | 27.17 | 30.79 |
| **txtur1** | ∗ | 18.07 | 19.04 | 20.40 | 22.31 | 25.40 |
| | † | 18.05 | 19.11 | 20.41 | 22.44 | 25.69 |
| | ‡ | 17.92 | 18.84 | 20.13 | 22.02 | 24.92 |
| **woman** | ∗ | 25.40 | 27.28 | 29.84 | 33.46 | 38.24 |
| | † | 25.43 | 27.33 | 29.95 | 33.59 | 38.28 |
| | ‡ | 25.07 | 26.91 | 29.43 | 32.93 | 37.73 |

Table 1: PSNR (dB) vs. output bit rate for proposed coder (∗), SPIHT-AC (†), and SPIHT (‡), on images from the JPEG2000 benchmark set.

zerotree-based coders, the encoder can generate the bitstreams for all bitplanes simultaneously in a single pass through the transform coefficients, when offline reordering of the bitstreams is possible. Furthermore, when sufficient memory resources are available, the decoding of long runs of zeros can be accelerated by the use of lists to remember the locations of significant coefficients, a technique borrowed from SPIHT. Coders based on context-adaptive arithmetic coding, on the other hand, are not as amenable to this technique.

The complexity advantages of the proposed coding algorithm stem from an explicit approach to the optimal embedding problem, which motivates the two-stage hierarchy of contexts which, in turn, allows for the low-complexity encoding of deinterleaved subsequences of coefficients with similar statistics.

# 7 Conclusion

We have proposed a new low-complexity method for coding the bitplanes of a wavelet-transformed image (actually the algorithm, to a large extent, can be tailored to work with any type of transform). In terms of PSNR our coder is competitive with SPIHT-AC and superior to SPIHT. We believe that the principal advantage of the proposed coder over zerotree-based schemes is that it embodies a cleaner separation between modeling, coding, and algorithmic components. This seems very useful from an engineering perspective since it greatly simplifies the problem of tuning the algorithm for particular applications (hardware, software) and performance vs. complexity tradeoffs. This also opens the possibility for further modeling improvements.

Potential enhancements to the proposed coder which are under investigation include incorporating further context modeling for coding within the run subsequence $S_{n,3}$, as the i.i.d. assumption seems to be weakest for this subsequence; further subdividing the run subsequence based on proximity to non-zero coefficients; and classifying subsequences based on larger blocks in the block version of the algorithm.

# 8 References

[1] J. M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12), December 1993.

[2] J. Rissanen and G. G. Langdon, Jr. Universal modeling and coding. *IEEE Trans. Info. Theory*, 27(1):12–23, January 1981.

[3] D. Taubman and A. Zakhor. Multirate 3-D subband coding of video. *IEEE Transactions on Image Processing*, 3(5), September 1994.

[4] A. Said and W. A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(3), 1996.

[5] X. Wu. High-order context modeling and embedded conditional entropy coding of wavelet coefficients for image compression. *31st Asilomar Conference on Signals, Systems and Computers*, October 1997.

[6] J. Li and P.-Y. Cheng and C.-C. Kuo. On the improvements of embedded zerotree wavelet (EZW) coding. *SPIE,* Taiwan, 1995.

[7] J. Li and S. Lei. Rate-distortion optimized embedding. *Picture Coding Symposium,* Berlin, Germany, pp. 201–206, Sep. 10-12, 1997.

[8] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Transactions on Image Processing*, 1(2), April 1992.

[9] S. W. Golomb. Run-length encodings. *IEEE Trans. Inform. Theory*, IT-12:399–401, July 1966.

[10] J. Teuhola. A compression method for clustered bit-vectors. *Information Processing Letters*, 7:308–311, October 1978.

[11] R. Ohnishi, Y. Ueno, and F. Ono. The efficient coding scheme for binary sources. *IECE of Japan*, 60-A:1114–1121, December 1977. (In Japanese).

[12] G. G. Langdon, Jr. An adaptive run-length coding algorithm. *IBM Technical Disclosure Bulletin*, 26:3783–3785, December 1983.

[13] R. Gallager and D.C. Van Voorhis. Optimal source codes for geometrically distributed integer alphabets. *IEEE Trans. Inform. Theory*, IT-21:228–230, March 1975.

[14] B. P. Tunstall. *Synthesis of Noiseless Compression Codes.* PhD thesis, Georgia Inst. Technol., Atlanta, 1968.

[15] F. Fabris. Variable-length-to-variable-length source coding: A greedy step-by-step algorithm. *IEEE Trans. Inform. Theory*, IT-38:1609–1617, September 1992.

[16] G. H. Freeman. Divergence and the construction of variable-to-variable-length lossless codes by source-word extensions. In *Proc. of the 1993 Data Compression Conference*, pages 79–97, Snowbird, Utah, USA, March 1993.

[17] G. Seroussi and M. J. Weinberger. On adaptive strategies for an extended family of Golomb-type codes. In *Proc. of the 1997 Data Compression Conference*, pages 131–140, Snowbird, Utah, USA, March 1997.

[18] ISO/IEC JTC1/SC29 WG1 (JPEG/JBIG). Information technology - lossless and near-lossless compression of continuous-tone still images, final committee draft FCD14495-1 (JPEG-LS), 1997.

[19] P. R. Stubley. Adaptive variable-to-variable length codes. In *Proc. of the 1994 Data Compression Conference*, pages 98–105, Snowbird, Utah, USA, March 1994.