HIGH PERFORMANCE SCALABLE IMAGE COMPRESSION WITH EBCOT

David Taubman

School of Electrical Engineering,
The University of New South Wales, Sydney, Australia

ABSTRACT

A new image compression algorithm is proposed, based on independent Embedded Block Coding with Optimized Truncation of the embedded bit-streams (EBCOT). The algorithm exhibits state-of-the-art compression performance while producing a bit-stream with a rich feature set, including resolution and SNR scalability together with a random access property. The algorithm has modest complexity and is extremely well suited to applications involving remote browsing of large compressed images. The algorithm lends itself to explicit optimization with respect to MSE as well as more realistic psychovisual metrics, capable of modeling the spatially varying visual masking phenomenon.

1. INTRODUCTION

This paper describes a novel image compression algorithm known as EBCOT: "Embedded Block Coding with Optimized Truncation". The EBCOT algorithm is related in various degrees to much earlier work on scalable image compression. Noteworthy among its early predecessors are Shapiro's EZW (Embedded Zero-Tree Wavelet compression) algorithm [5], Said and Pearlman's SPIHT (Spatial Partitioning of Images into Heirarchical Trees) algorithm [4] and Taubman and Zakhor's LZC (Layered Zero Coding) algorithm [6]. Like each of these, the EBCOT algorithm uses a Wavelet transform to generate the subband samples which are to be quantized and coded. Like each of these, the EBCOT algorithm also generates scalable compressed bit-streams; however, whereas its predecessors typically exhibited one (SNR) or at most two (SNR and resolution) dimensions of scalability, the EBCOT algorithm produces bit-streams which are SNR and resolution scalable and which also have a random access attribute whereby independent portions of the bit-stream correspond to different spatial regions of the image.

A key advantage of scalable compression is that the target bit-rate or reconstruction resolution need not be known at the time of compression. A related advantage is that the image need not be compressed multiple times in order to achieve a target bit-rate. Rather than focusing on generating a single scalable bit-stream to represent the entire image, EBCOT partitions each subband into relatively small blocks of samples and generates a separate highly scalable bit-stream to represent each so-called code-block, B_i. The bit-stream associated with B_i may be independently truncated to any of a collection of different lengths, R_i^n, where the increase in reconstructed image distortion resulting from these truncations is given by D_i^n, with respect to an appropriate distortion metric. The enabling observation leading to the development of the EBCOT algorithm is that it is possible to independently compress relatively small blocks (say 32×32 or 64×64) with an embedded bit-stream consisting of a large number of truncation points, R_i^n, such that most of these truncation points lie on the convex hull of the corresponding rate-distortion curve [7].

The existence of a large number of independent code-blocks, each with many useful truncation points leads to a vast array of options for constructing scalable bit-streams. To efficiently utilize this flexibility, the EBCOT algorithm introduces a novel abstraction between the massive number of code-stream segments produced by the block entropy coding process and the structure of the bit-stream itself. Specifically, the bit-stream is organized into so-called *quality layers*, $\mathcal{Q}_q$. Each code-block, B_i, contributes $R_i^{n_i^q} - R_i^{n_i^{q-1}}$ bytes to quality layer, $\mathcal{Q}_q$, so that layers $\mathcal{Q}_1$ through $\mathcal{Q}_q$ contain the portion of the block's embedded bit-stream up to truncation point n_i^q. The Post-Compression Rate Distortion optimization (PCRD-opt) algorithm described in Section 2 assigns the truncation points, n_i^q, in an optimal manner with respect to any of a variety of relevant distortion metrics so that truncating the final bit-stream to any whole number of quality layers results in a rate-distortion optimal subset of the original bit-stream. Moreover, this property also holds if one or more of the subbands are discarded to reduce the effective image resolution, or if some of the blocks are

This work was largely performed while the author was with Hewlett-Packard Research Laboratories in Palo Alto, California.

discarded to reduce the spatial region of interest.

The abstraction here is that the quality layers are not tied directly to the underlying entropy coding process, which means that the compressor has the freedom to compose any number of layers, using any desired policy (usually induced by an appropriate distortion metric, according to the PCRD-opt algorithm) to determine the relative contributions of each code-block to each layer. At one extreme only a single layer is generated, so that the final bit-stream is not SNR scalable, but exact rate control is achieved in a single pass via the PCRD-opt algorithm. At the other extreme a very large number of layers (typically 50) is generated for optimum performance when the target bit-rate is entirely unknown. Between these extremes, a small number of targeted layers may be generated for applications in which the set of desirable bit-rates can be known ahead of time. To reduce the overhead associated with representing the contents of a large number of layers, EBCOT introduces a second coding engine which operates on the summary information associated with each code-block, as described in Section 4.

2. RATE-DISTORTION OPTIMIZATION

Recall that the image is represented by a collection of code-blocks, B_i, whose embedded bit-streams may be truncated to rates, R_i^n; the corresponding contribution to distortion in the reconstructed image is denoted D_i^n, for each truncation point, n. We are thus assuming that the relevant distortion metric is additive, i.e. $D = \sum_i D_i^{n_i}$, where D is distortion in the reconstructed image and n_i is the truncation point for B_i. If we work with orthogonal (or approximately orthogonal) Wavelet transforms or if quantization errors in distinct code-blocks can be considered uncorrelated then this additivity requirement is satisfied by the MSE metric and by a Weighted MSE (WMSE) metric obtained by scaling the MSE contribution from each subband according to a visual Contrast Sensitivity Function (CSF). It is also satisfied by the very successful visual distortion metric discussed in the appendix. Here, we briefly discuss optimal selection of the n_i so as to minimize distortion, D, subject to a constraint, $R_{\max}$, on the bit-rate for the relevant quality layer.

Now it is easy to see that any set of truncation points, $\{n_i^\lambda\}$, which minimizes

$$(D_\lambda + \lambda R_\lambda) = \sum_i \left(D_i^{n_i^\lambda} + \lambda R_i^{n_i^\lambda} \right) \qquad (1)$$

for some λ is optimal in the sense that the distortion cannot be reduced without also increasing the overall rate. Thus, if we can find a value of λ such that the truncation points which minimize (1) yield the target rate, $R_\lambda = R_{\max}$, then this set of truncation points must be an optimal solution to our rate-distortion optimization problem. We will not generally be able to find a value of λ for which R_λ is exactly equal to $R_{\max}$. Nevertheless, since EBCOT's code-blocks are relatively small (e.g. 32x32 or 64x64) and there are many truncation points for each block, it is sufficient in practice to find the smallest value of λ such that $R_\lambda \leq R_{\max}$.

We conclude this section by observing that the determination of the truncation points, n_i^λ, for any given λ, may be performed very efficiently, based on a small amount of summary information collected during the generation of each code-block's embedded bit-stream. Thus, the rate control operation is applied after compressing all the code-blocks in a single pass; hence the name *Post Compression Rate Distortion optimization* (PCRD-opt). It is clear that we have a separate minimization problem for each code-block, B_i. Briefly, a conventional convex hull analysis yields the set of candidate truncation points, n_{i_j}, for each code-block B_i; we need then only store the rates, $R_i^{n_{i_j}}$ and distortion-rate "slopes", $S_i^{n_{i_j}} = (D_i^{n_{i_{j-1}}} - D_i^{n_{i_j}})/(R_i^{n_{i_j}} - R_i^{n_{i_{j-1}}})$. The convex hull analysis ensures that the slopes are strictly decreasing, so the solution is obtained by setting $n_i^\lambda = \max\{n_{i_j} | S_i^{n_{i_j}} \geq \lambda\}$. For more information on the PCRD-opt algorithm, refer to [7].

3. EMBEDDED BLOCK CODING

In this section, we describe the actual block coding algorithm, which generates a separate embedded bit-stream for each code-block, B_i. The coder is essentially a bit-plane coder, using similar techniques to those of the LZC algorithm [6]. The key additions are: i) the use of "fractional bit-planes", in which the quantization symbols for any given bit-plane are coded in a succession of separate passes, rather than just one pass; and ii) a simple embedded quad-tree algorithm is used to identify whether or not each of a collection of "sub-blocks" contains any non-zero (significant) samples at each bit-plane. The use of fractional bit-planes is motivated by separate work by Ordentlich et al. [3] and by Li and Lei [2]; its purpose is to ensure a sufficiently fine embedding. The use of a partial embedded quad-tree to identify the significance of each sub-block of samples reduces the model adaptation cost incurred by coding each code-block independently. It may be understood as a type of prior assumption on the joint probability model for significant samples; specifically, the samples which first become significant in bit-plane p, are assumed to be clustered. The initial test for significance at the sub-block level also substantially reduces the number of symbols which must be arithmetically coded, thereby reducing the overall computational complexity.

3.1. Quantization and Significance

Let $s_i[\mathbf{k}] = s_i[k_1, k_2]$ denote the two-dimensional sequence of subband samples belonging to code-block B_i. Let $\chi_i[\mathbf{k}] \in \{1, -1\}$ denote the sign of $s_i[\mathbf{k}]$ and let $\nu_i[\mathbf{k}]$ denote the quantized magnitude, i.e.

$$\nu_i[\mathbf{k}] = \frac{|s_i[\mathbf{k}]|}{\delta_{b_i}}$$

where δ_b is the step-size of the "deadzone" quantizer for subband b and b_i is the subband to which code-block B_i belongs.

Now let P_{b_i} denote the number of bits used to represent these quantized magnitudes; in practice, it is sufficient for the encoder and decoder to agree on a value which is large enough to represent any quantized samples which might arise from the subband. Let $\nu_i^p[\mathbf{k}]$ denote the p'th bit in this P_{b_i}-bit representation, where $0 \leq p < P_{b_i} - 1$. The idea behind bit-plane coding is to encode first the most significant bits, $\nu_i^{P_{b_i}-1}[\mathbf{k}]$, for all samples in the code-block, then the next most significant bits and so on until all bit-planes have been encoded. If the bit-stream is truncated so that some sample is missing p least significant bits, the effect is equivalent to using a coarser deadzone quantizer having step size $2^p \delta_{b_i}$.

In order to efficiently encode $\nu_i^p[\mathbf{k}]$, it is important to exploit previously encoded information about the same sample and neighbouring samples. We do this primarily by means of a binary-valued state variable, $\sigma_i[\mathbf{k}]$, which is initialized to 0, but transitions to 1 when the relevant sample's first non-zero bit-plane value, $\nu_i[\mathbf{k}]$, is encoded. Following convention, we refer to the state, $\sigma_i[\mathbf{k}]$, as the sample's "significance".

3.2. Bit-Plane Coding Primitives

Here we briefly describe the four different primitive coding operations which form the foundation of the embedded block coding strategy. The primitives are used to code new information for a single sample, $s_i[\mathbf{k}]$, in some bit-plane, p. If the sample is not yet significant, i.e. $\sigma_i[\mathbf{k}] = 0$, a combination of the "Zero Coding" (ZC) and "Run-Length Coding" (RLC) primitives is used to code whether or not the symbol becomes significant in the current bit-plane, i.e. whether or not $\nu_i^p[\mathbf{k}] = 1$. If so, the "Sign Coding" (SC) primitive must also be invoked to identify the sign, $\chi_i[\mathbf{k}]$. The RLC primitive is invoked in place of the ZC primitive only when multiple consecutive insignificant samples have entirely insignificant neighbourhoods; its purpose is to reduce the number of coded symbols and hence the computational complexity of the algorithm. If the sample is already significant, the "Magnitude Refinement" (MR) primitive is used to encode the value of $\nu_i^p[\mathbf{k}]$. In every case, a single binary-valued symbol must be encoded, using a common arithmetic coding engine. The probability models used by the arithmetic coder evolve within 18 different contexts: 9 for the ZC primitive; 1 for the RLC primitive; 5 for the ZC primitive; and 3 for the MR primitive. The specific context to be used is determined from the significance and sign information which has already been coded for neighbouring samples. For more information, the reader is referred to [7].

3.3. Fractional Bit-Planes and Scanning Order

For each bit-plane, p, the coding proceeds in a number of distinct passes, which we identify as "fractional bit-planes". In this work we consider a total of four such passes, $\mathcal{P}_1^p$, $\mathcal{P}_2^p$, $\mathcal{P}_3^p$ and $\mathcal{P}_4^p$ and we identify the truncation points, R_i^n, with the length of the arithmetic code word required to uniquely represent the symbols coded in each successive coding pass. In this way there are four truncation points for each bit-plane. The reason for introducing multiple coding passes is to ensure that each code-block has a finely embedded bit-stream. The coding passes are as follows:

Forward significance propagation pass, $\mathcal{P}_1^p$

Scans each sub-block from top to bottom using the ZC and RLC primitives to code the significance of each currently insignificant sample, whose neighbourhood significance pattern suggests that the sample is very likely to become significant. As new samples become significant they increase the likelihood that subsequent samples will have the appropriate neighbourhood significance pattern, thereby generating a propagating wave of significance decisions.

Reverse significance propagation pass, $\mathcal{P}_2^p$

Similar to $\mathcal{P}_1^p$, except that the samples and sub-blocks within the block are scanned in reverse order and the neighbourhood significance pattern is designed to include more samples in the pass.

Magnitude refinement pass, $\mathcal{P}_3^p$

Uses the MR primitive to refine the magnitude of all significant samples, for which information has not been coded in the previous two passes.

Normalization pass", $\mathcal{P}_4^p$

Uses the ZC and RLC primitives to code the significance of all samples not considered in the previous three passes.

$\mathcal{S}^{P_i-1}$	$\mathcal{P}_4^{P_i-1}$	$\mathcal{P}_1^{P_i-2}$	$\mathcal{P}_2^{P_i-2}$	$\mathcal{P}_3^{P_i-2}$	$\cdots$
$\cdots$	$\mathcal{S}^p$	$\mathcal{P}_4^p$	$\mathcal{P}_1^{p-1}$	$\mathcal{P}_2^{p-1}$	$\mathcal{P}_3^{p-1}$	$\cdots$
$\cdots$	$\mathcal{S}^0$	$\mathcal{P}_0^p$			

Figure 1: *Appearance of coding passes and quad-tree codes within each code-block's embedded bit-stream.*

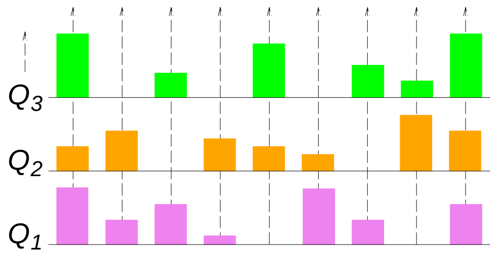

Figure 2: *Progressive appearance of embedded code-block bit-streams in quality layers.*

Figure 1 illustrates the appearance of information within each code-block's embedded bit-stream. Here, $\mathcal{S}^p$ denotes the quad-tree code, identifying the sub-blocks which are significant in bit-plane p. Notice that $\mathcal{S}^p$ appears immediately before the final coding pass, $\mathcal{P}_4^p$, not the initial coding pass, $\mathcal{P}_1^p$, for the bit-plane. This means that all samples in those sub-blocks which become significant for the first time in bit-plane p, are ignored until pass $\mathcal{P}_4^p$. It can and indeed often does happen that one or more of the leading coding passes indicated in Figure 1 is empty, since no sub-blocks are significant at all. These empty passes consume no bits, since the bit-plane, $p_i^{\max}$, in which each code-block, B_i, first becomes significant is identified via a separate mechanism in the "second tier" coding engine discussed in Section 4.

4. LAYER REPRESENTATION

Recall that the final bit-stream is composed of a collection of quality layers, $\mathcal{Q}_q$. Together, layers $\mathcal{Q}_1$ through $\mathcal{Q}_q$ contain the initial $R_i^{n_i^{\lambda_q}}$ bytes of each code block, B_i, where λ_q is the rate-distortion slope for $\mathcal{Q}_q$ and $\{n_i^\lambda\}$ denotes the set of rate-distortion optimal block truncation points corresponding to slope λ, as explained in Section 2. In many applications, the number of layers and their slopes, λ_q, might be optimized to yield R-D optimal performance at a set of target bit-rates. In any event, layer $\mathcal{Q}_q$ contains the incremental contributions, $L_i^q = R_i^{n_i^{\lambda_q}} - R_i^{n_i^{\lambda_{q-1}}} \geq 0$, from each code-block's embedded bit-stream. As illustrated in Figure 2, some code-blocks might contribute no bytes at all to some layers. Along with these incremental contributions, the length of the segment, L_i^q and the number of new coding passes, $N_i^q = n_i^{\lambda_q} - n_i^{\lambda_{q-1}}$, in the segment must be identified for each code-block, B_i. Finally, if B_i makes its first non-empty contribution to quality layer $\mathcal{Q}_q$ then the most significant bit-plane, $p_i^{\max}$, must also be identified, as mentioned in Section 3.3.

Unfortunately, in this brief discourse it is not possible to explain the various methods used to signal these different types of block summary information. It is, however, important to note that two quantities exhibit substantial spatial redundancy: the most significant bit-plane, $p_i^{\max}$; and the index, q_i, of the quality layer to which B_i first makes a non-empty contribution. To capture this redundancy, the EBCOT algorithm employs a type of embedded quad-tree code, which is driven from the leaves instead of the root. To see why this is important, note that $p_i^{\max}$ must be identified only in the first quality layer to which B_i makes a contribution, which is generally different for different code blocks from the same subband. We build a separate quad-tree for each subband, but scan from the leaves toward the root sending only the minimal subset of the quad-tree code which is required by blocks which become significant in the relevant quality layer. For more information, the reader is referred to [7].

5. EXPERIMENTAL RESULTS

Table 1 provides numerical results to illustrate the performance of the proposed EBCOT algorithm under a variety of conditions. Average PSNR results are presented for the three most popular JPEG2000 test images, "Bike", "Cafe" and "Woman", each 2560x2048 pixels in size. The first column of PSNR results corresponds to the well known SPIHT [4] algorithm with the arithmetic coding option for maximum performance. The next three columns of PSNR results are obtained with the EBCOT algorithm with various numbers of quality layers in the bit-stream. With only one quality layer, the minimum overhead is incurred in specifying the layer contributions of each code-block, but the bit-stream is not SNR scalable. With five quality layers, optimized for the specific bit-rates at which we are reporting PSNR results, we obtain some degree of SNR scalability with little sacrifice in performance. In the generic case, 50 quality layers are generated, covering bit-rates spanning several orders of magnitude, where the R-D slope values, λ_q, are not optimized for any particular bit-rates. This models applications in which nothing is known about the bit-rates of interest. The bit-stream is simply truncated to each of the rates at which we measure PSNR.

The last column of PSNR results reported in Ta-

Table 1: *Average PSNR results, measured in dB, for the "Bike", "Cafe" and "Woman" test images.*

bpp	Spiht	1 layer	5 layers	Generic	Visual
0.0625	22.61	22.89	22.89	22.82	22.40
0.125	24.63	24.94	24.93	24.86	24.44
0.25	27.37	27.73	27.71	27.61	27.21
0.5	31.03	31.50	31.46	31.34	31.02
1.0	35.91	36.41	36.37	36.20	35.71

ble 1 is obtained using the visual distortion metric described in the Appendix instead of the MSE metric used for the previous three columns. Of course, the PSNR decreases somewhat because we are no longer optimizing for MSE which is equivalent to PSNR. On the other hand, the visual quality is observed to increase dramatically; in the case of the "Woman" image, the visual improvement is comparable to a saving of approximately 50% in bit-rate. For more comprehensive results, the reader is again referred to [7]. The MSE-optimal results are obtained with 64x64 code-blocks, while the best visual results have 32x32 blocks.

6. CONCLUSIONS

The EBCOT algorithm offers state-of-the-art compression performance with a rich set of bit-stream features, including resolution scalability, SNR scalability and the random access property. Remarkably, all features can co-exist within a single bit-stream without making substantial sacrifices in compression efficiency. On a fundamental level, EBCOT makes two significant contributions to the current state of image compression. The first contribution is the generation of finely embedded bit-streams independently for each block of subband samples, thereby enabling the use of PCRD optimization; this in turn enables the successful exploitation of visual masking, whose spatially varying nature hampers its exploitation in conventional compression frameworks. The second contribution is EBCOT's introduction of abstract quality layers which are not directly related to the structural properties of the underlying entropy coder. This endows the encoder with tremendous flexibility in structuring bit-streams with appropriate properties. As a result of these properties, EBCOT has been selected as the framework for the evolving JPEG2000 image compression standard.

7. REFERENCES

[1] I. Höntsch and L. Karam, "APIC: Adaptive Perceptual Image Coding based on Subband Decomposition with Locally Adaptive Perceptual Weighting," *Proc. Int. Conf. Image Proc.* Vol. 1, pp. 37-40, 1997.

[2] J. Li and S. Lei, "Rate-Distortion Optimized Embedding," *Picture Coding Symposium (Berlin).* pp. 201-206, September 10-12 1997.

[3] E. Ordentlich, M. Weinberger and G. Seroussi, "A Low-Complexity Modeling Approach for Embedded Coding of Wavelet Coefficients," *Data Compression Conference (Snowbird).* pp. 408-417, March 1998.

[4] A. Said and W. Pearlman, "A New, Fast and Efficient Image Codec based on Set Partitioning in Hierarchical Trees," *IEEE Trans. on Circuits and Systems for Video Technology.* Vol. 6, No. 3, pp. 243-250, June 1996.

[5] J. M. Shapiro, "An Embedded Hierarchical Image Coder using Zerotrees of Wavelet Coefficients," *Data Compression Conference (Snowbird).* pp. 214-223, 1993.

[6] D. Taubman and A. Zakhor, "Multi-Rate 3-D Subband Coding of Video," *IEEE Trans. Image Proc.* Vol. 3, No. 5, pp. 572-588, September, 1994.

[7] D. Taubman, "High Performance Scalable Image Compression with EBCOT," *submitted to IEEE Trans. Image Proc.* March, 1999. Also available via http://maestro.ee.unsw.edu.au/~taubman.

A. VISUAL DISTORTION METRICS

It is well known that MSE is a poor model for visual distortion. Extensions to visually weighted MSE based on the CSF are also widely known. More generally, we may consider spatially varying distortion metrics which attempt to exploit the masking phenomenon. Previous attempts to do so, such as that in [1], have been hampered by causality constraints and non-scalable bit-streams. The EBCOT algorithm and the PCRD-opt strategy of Section 2 provide an excellent context in which to exploit spatially varying visual phenomena such as masking. In our work, the conventional MSE distortion metric is modified according to

$$D_i^n = w_{b_i}^2 \sum_{\mathbf{k}} \frac{\left(\hat{s}_i^n[\mathbf{k}] - s_i[\mathbf{k}]\right)^2}{\left(V_i[j,k]\right)^2}$$

where $V_i[\mathbf{k}]$ is a computationally efficient approximation to the "visual masking strength" at sample $s_i[\mathbf{k}]$, w_{b_i} is the L2-norm of the Wavelet basis functions for the relevant subband. The formulation is normalized so that this compensation for masking effects does not introduce any global weighting to the different subbands. Global weightings based on the Human Visual System's CSF can additionally be included and have been shown to give complementary improvements to image quality. For more information on the visual masking metric, the reader is referred to [7].