LEVEL-EMBEDDED LOSSLESS IMAGE COMPRESSION

Mehmet Celik, A.Murat Tekalp

University of Rochester
Dept. of Electrical and Computer Eng.
Rochester, NY, 14627-0126

Gaurav Sharma

Xerox Corporation
MS0128-27E, 800 Phillips Rd.,
Webster, NY, 14580

ABSTRACT

A level-embedded lossless compression method for continuous-tone still images is presented. Level (bit-plane) scalability is achieved by separating the image into two layers before compression and excellent compression performance is obtained by exploiting both spatial and inter-level correlations. A comparison of the proposed scheme with a number of scalable and non-scalable lossless image compression algorithms is performed to benchmark its performance. The results indicate that the level-embedded compression incurs only a small penalty in compression efficiency.

1. INTRODUCTION

Although most image processing applications can tolerate some information loss, in several areas–such as medical, satellite, and legal imaging– lossless compression algorithms are preferred. CALIC [1], JPEG-LS [2], and JPEG2000 [3] are among well-known lossless image compression algorithms. Among these CALIC provides best compression ratios over typical images, whereas, JPEG-LS is a low complexity alternative with competitive efficiency. The JPEG2000 standard, on the other hand, is a wavelet-based technique, which provides a unified approach for lossy-to-lossless compression.

Generation of an embedded bit-stream, where a lower quality image can be reconstructed with only a part of the bit-stream, is referred as scalable compression. In this paper, we propose a specific instance of scalable compression called level-embedded compression. Level-embedded scalability refers to bit-plane scalability in the image pixel value domain. The method is useful in several applications, where data is acquired by a capture device with a high dynamic range or bit-depth. A lower bit-depth representation is often sufficient for most purposes and the higher bit-depth data is only required for specialized analysis/enhancement or archival purposes. If the full bit-depth image is stored in a conventional lossless compressed stream, a subsequent truncation of lower order bits requires a decompression and reconstruction of the image prior to truncation. If on the other hand, the compression scheme (and the corresponding bit stream) is level-embedded, the truncation can effectively be performed in the bit stream itself by dropping the segment of the stream corresponding to the truncated lower levels. The latter option is often much more desirable because of its memory and computational simplicity, which translate to lower power, time, and resource requirements.

JPEG2000 offers scalability in resolution and distortion by allowing reconstruction of lower resolution and/or lower signal-to-noise-ratio (SNR) images. The scalability in JPEG2000 is, however, different from the scalability provided by level embedded compression. Scalability in JPEG2000 is implemented in the wavelet transform coefficient domain. Truncation of bit-planes in the wavelet transform coefficient domain does not, in general, correspond to the proposed level embedded scalability in the image pixel value domain. In legal applications, the level embedded scalability may therefore be more acceptable because the potential for spatial artifacts may cast doubts on the veracity of photographic evidence. The bit-depth truncation in level-embedded compression is analogous to using an acquisition device with a lower resolution A/D converter. It also offers tight per pixel maximum absolute error bounds and is guaranteed to not produce any spatial artifacts. JPEG-LS, in its near-lossless compression mode, provides per pixel maximum absolute error guarantees without introducing any spatial artifacts, as in level-embedded compression. In this mode, however, JPEG-LS provides only lossy compression and not an embedded lossless stream.

Level-embedded compression may be achieved through independent compression of individual bit-planes as in JBIG [4]. This process, however, causes a significant penalty in compression performance over non-level-embedded methods because it fails to exploit correlations between the different bit-planes of an image. In this paper, we propose an alternative method for achieving level embedded compression which significantly reduces the penalty in compression performance by exploiting the correlations.

2. LEVEL EMBEDDED COMPRESSION ALGORITHM

We first describe the algorithm[1] for the case of two embedding levels: a base layer corresponding to the higher levels and a residual layer comprising of the lower levels. The method is subsequently generalized to multiple levels in Section 2.4. The image is first separated into the base layer and a residual layer. The base layer is obtained by dividing each pixel value by a constant integer L ($B_L(s) = \left\lfloor \frac{s}{L} \right\rfloor$). L specifies the amplitude of the enhancement layer, which is the remainder, which is also called *the residual* ($r = s - L \left\lfloor \frac{s}{L} \right\rfloor$). We also call the quantity $L \left\lfloor \frac{s}{L} \right\rfloor$ as the quantized pixel, $Q_L(s)$. Note that the use of a power of 2 for L corresponds to partitioning of the images into more significant and less significant bit planes, and other values generalize this notion to a partitioning into higher and lower levels. Since the resulting base layer, i.e. the most significant levels of the image, is coded without any reference to the enhancement layer and its statistics closely resemble that of the full bit-depth image, any lossless compression algorithm can be used for the base layer. In this paper, CALIC [1] is used for base layer compression. The compression of the enhancement layer is outlined in more detail below.

[1] Additional details of the algorithm can be found in [5].

Since the enhancement layer, or the residual signal, represents the lowest levels of a continuous-tone image, its compression is a challenging task. For small values of L, the residual typically has no structure, and its samples are virtually uniformly distributed and uncorrelated from sample to sample. If the rest of the image information is used as side-information, however, significant coding gains can be achieved, by exploiting the spatial correlation among pixel values and the correlation between high and low levels (bit-planes) of the image.

The proposed method is inspired by the CALIC algorithm [1]. The method is comprised of three main components: *i)* prediction, *ii)* context modeling and quantization, *iii)* conditional entropy coding. The prediction component reduces spatial redundancy in the image. The context modeling stage further exploits spatial correlation and the correlation between different image levels. Finally, conditional entropy coding based on selected contexts translates these correlations into smaller code-lengths. The algorithm is presented below in pseudo-code.

1. $\hat{s}_O$ = Predict Current Pixel();
2. d, t = Determine Context D,T($\hat{s}_O$);
3. $\dot{s}_O$ = Refine Prediction($\hat{s}_O, d, t$);
4. θ = Determine Context $\Theta(\dot{s}_O)$;
5. If ($\theta \geq 0$),
   Encode/Decode Residual(r_O, d, θ);
   else,
   Encode/Decode Residual($L - 1 - r_O, d, |\theta|$);

2.1. Prediction

Prediction is based on a local neighborhood of a pixel which consists of its 8-connected neighbors, denoted by standard map directions: W, NW, N,... The residual samples are encoded and decoded in the raster scan order, i.e. left-to-right and top-to-bottom. This order guarantees that residuals at positions W, NW, N, NE have already been reconstructed when the center residual, r_O, is being decoded. In addition, all quantized pixel values of the image, $Q_L(\mathbf{s})$, are known as side-information. We define a reconstruction function $f(.)$, which gives the best known value of a neighboring pixel, exact value ($Q_L(s) + r$) if known, or the quantized value plus $\frac{L}{2}$ (to compensate for the bias in the truncation $Q_L(.)$).

$$f(s_k) = \begin{cases} s_k & \text{if } k \in \{W, NW, N, NE\}, \\ Q_L(s_k) + \frac{L}{2} & \text{otherwise.} \end{cases} \quad (1)$$

A simple, linear prediction for the current pixel value is calculated using the nearest, 4-connected neighbors of a pixel.

$$\hat{s}_O = \frac{1}{4} \sum_{k \in \{W,N,E,S\}} f(s_k). \quad (2)$$

Since this predictor is often biased, resulting in a non-zero mean for the prediction error, $s_O - \hat{s}_O$, we refine this prediction and remove its bias using a feed-back loop, on a per-context basis as in [1]. The refined prediction is calculated as,

$$\dot{s}_O = round(\hat{s}_O + \bar{\epsilon}(d, t)), \quad (3)$$

where $round(\ )$ is the integer round, and $\bar{\epsilon}(d, t)$ is the average of the prediction error ($\epsilon = s_O - \dot{s}_O$) over all previous pixels in the given context (d, t). The resulting predictor $\dot{s}_O$ is a context-based, adaptive, nonlinear predictor.

2.2. Context Modeling and Quantization

Typical natural images exhibit non-stationary characteristics with varying statistics in different regions. If the pixels can be partitioned into a set of *contexts*, such that within each context the statistics are fairly regular, the statistics of the individual contexts (e.g. probability distributions) may be exploited in encoding the corresponding pixels (residuals) using conditional entropy coding. If chosen appropriately, contexts can yield significant improvements in coding efficiency. Increasing number of contexts better adapt to the local image statistics hence improve the coding efficiency. Since the corresponding conditional statistics often have to be learned on-the-fly observing the previously encoded (decoded) symbols, convergence of these statistics and thereby efficient compression is delayed when a large number contexts are used. The reduction in compression efficiency due to large number of contexts is known as the *context dilution* problem.

As a first step, we adopt a variant of d and t contexts from [1], which are defined as follows

$$\Delta = \sum_{k \in \{W,NW,N,NE,E,SE,S,SW\}} \frac{1}{8} |f(s_k) - \hat{s}_O|, \quad (4)$$

$$d = Q(\Delta), \quad (5)$$

$$t_k = \begin{cases} 1 & \text{if } f(s_k) > \hat{s}_O, \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

$$t = t_W \| t_N \| t_E \| t_S, \quad (7)$$

where t is obtained by concatenating the individual t_k bits (16 values), and $Q(\Delta)$ is a scalar non-uniform quantizer with 8 levels, whose thresholds are experimentally determined so as to include an approximately equal number of pixels in each bin[2]. The context d corresponds to local activity as measured by the mean absolute error of the unrefined predictor Eqn. 2 and t corresponds to a texture context[3].

Typically, the probability distribution of the prediction error, $\epsilon = s - \dot{s}$, can be approximated fairly well by a Laplacian distribution with zero mean and a small variance which is correlated with the context d [6, pp. 33],[7]. Here, we assume that the prediction error distribution $p(\epsilon|d)$ is exactly Laplacian. The arguments and the ensuing conclusions and techniques, however, are largely applicable even when the true distributions deviate from this assumption. Fig. 1.a shows a plot of the probability mass function (*pmf*) $p(\epsilon|d)$ under this assumption. Given $\dot{s}$, the conditional probability distribution of pixel values $p(s = \dot{s} + \epsilon|d, \dot{s})$ is obtained by shifting the prediction error distribution $p(\epsilon|d)$ by $\dot{s}$ (Fig. 1.b).

In order to obtain residual's probability distribution from pixel statistics and to exploit the knowledge of the quantized pixel $Q_L(s)$, we introduce an additional context, θ, which is used only in the coding process and not in prediction.

Note that the known quantized value $Q_L(s)$ may be used as an additional context directly. A known quantized pixel value, $Q_L(s)$, limits the possible values of the pixel s to the range $[Q_L(s), Q_L(s) + L)$. This is illustrated in Fig. 1.b as the region between the two vertical broken lines. The conditional probability mass function $p(r|d, \dot{s}, Q_L(s))$ can therefore be obtained by normalizing this segment of the probability mass function to sum

[2] For the experimental results of Section 3, the quantizer $Q(\ )$'s threshold are $\{1, 2, 3, 4, 6, 10, 15\}$

[3] In order to avoid context-dilution during coding, t contexts are used only during prediction and not while coding.

up to 1 (see Fig 1.c). Entropy coding the residual using this conditional *pmf* restricts the symbol set required thereby improving compression. Note, however, that there are typically a large number of possible values for $Q_L(s)$, which would cause significant context dilution. The characteristics of the Laplacian distribution, however, allow for a significant reduction in the number of these contexts.

Since the Laplacian distribution decreases exponentially about its peak at $\dot{s}$, the conditional *pmf* $p(r|d, \dot{s}, Q_L(s))$ can be determined from the relative positions of $\dot{s}$ and $Q_L(s)$. For instance, if $\dot{s} \leq Q_L(s)$, the peak is at $r = 0$ and the *pmf* decreases exponentially and is identical for all cases corresponding to $\dot{s} \leq Q_L(s)$ (e.g. Fig 1.b&c). This allows all the cases corresponding to $\dot{s} \leq Q_L(s)$ to be combined into a single composite context. Similarly, if $\dot{s} \geq Q_L(s) + L - 1$, the peak is at $r = L - 1$ and the distribution increases exponentially, which may all be combined into a single context as well. In other cases, when $Q_L(s) < \dot{s} < Q_L(s) + L - 1$, the peak is at $r = \dot{s} - Q_L(\dot{s})$. Although total number of contexts after the above reductions is not large, it can be reduced further, if the symmetry of the Laplacian is exploited. In particular, the distributions with peaks at r_θ and $L - 1 - r_\theta$ are mirror images of each other. If the residual values are re-mapped (flipped $r_{new} = L - 1 - r_{old}$) in one of these two contexts, the resulting distributions will be identical. As a result, we can merge these contexts without incurring any penalty.

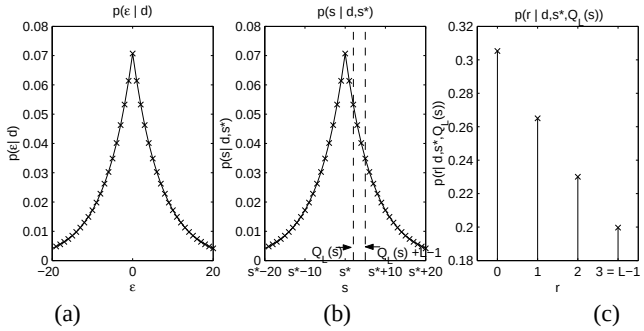

Fig. 1. *a)* Prediction error PMF, $p(\epsilon|d)$, under Laplacian assumption ($\sigma_d = 10$). *b)* Corresponding pixel PMF $p(s = \dot{s} + \epsilon|d, \dot{s})$. *c)* Conditional PMF of the residual ($L = 4$), $p(r|d, \dot{s}, Q_L(s))$

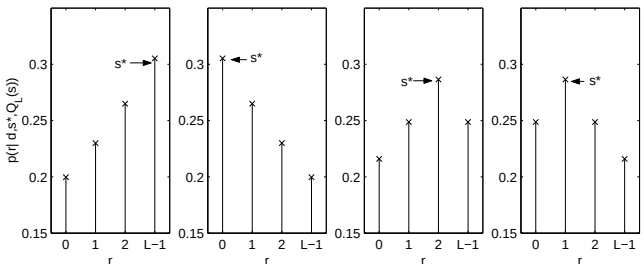

Fig. 2. Conditional PMFs $p(r|d, \dot{s}, Q_L(s))$ for contexts $\theta = \{\pm 1, \pm 2\}$ ($L = 4$). Symmetric contexts are merged by remapping the residual values.

The θ contexts differentiate between statistically different (after incorporating all symmetries) residuals using the knowledge of $\dot{s}$ and $Q_L(s)$. This enables the conditional entropy coder to adapt to the corresponding probability distributions in order to achieve higher compression efficiency. Minimizing the number of such contexts allows the estimated conditional probabilities to converge to the underlying statistics faster.

Finally, we have empirically determined that assigning a separate θ context to the cases $\dot{s} = Q_L(s)$ and $\dot{s} = Q_L(s) + L - 1$ further enhances the compression efficiency. These cases have been formerly included in the context where $\dot{s} \leq Q_L(s)$ and $\dot{s} \geq Q_L(s) + L - 1$. We believe that the rounding in Eqn. 3 partially randomizes the prediction when $Q_L(s) \approx \dot{s}$ and causes this phenomenon. The number of θ contexts and (d, θ) coding contexts become $\left\lfloor \frac{L+1}{2} + 1 \right\rfloor$ and $8 \left\lfloor \frac{L+1}{2} + 1 \right\rfloor$, respectively.

2.3. Conditional Entropy Coding

At the final step, residual values are entropy coded using estimated probabilities conditioned on different contexts. In order to improve efficiency, we use a context-dependent adaptive arithmetic coder. In a context-dependent adaptive entropy coder, the conditional probability distribution of residuals in each coding context (d, θ) is estimated from previously encoded(decoded) residual values. That is, the observed frequency of each residual value in a given context approximates its relative probability of occurrence. These frequency counts are passed to an arithmetic coder which allocates best code-lengths corresponding to given symbol probabilities.

2.4. Multi-level Embedded Coding

The above description outlined level embedded compression for two levels, a base layer and a single enhancement level. Multi-level embedded coding can be obtained as a straightforward extension by applying the algorithm recursively. In the first stage, the image is separated into a base layer B_1 and an enhancement layer r_1 using level L_1. In the second stage, the base layer B_1 is further separated into a base layer B_2 and enhancement layer r_2 using a (potentially different) level L_2. The process is continued for additional stages as desired. Each enhancement layer r_i is compressed using the corresponding base layer B_i, and last base layer B_n is compressed as earlier.

3. EXPERIMENTAL RESULTS

We evaluated the performance of the proposed scheme using the six 512×512 8-bit gray-scale images seen in Fig. 3. Although the algorithm works for arbitrary values of the embedding level L, in order to allow comparison with bit-plane compression schemes, here we concentrate on bit-plane embedded coding, which corresponds to using $L = 2$. Furthermore, the recursive scheme outlined in Sec. 2.4 is used to obtain multi-level embeddings with more than one enhancement layer, each consisting of a bit-plane. The number of enhancement layers, i.e. embedded bit-planes, is varied from 1 through 7. One (1) enhancement layer corresponds to the case where the LSB-plane is the enhancement layer and 7 MSB-planes form the base layer. Likewise, seven (7) enhancement layers correspond to a fully scalable bit-stream, where all bit-planes can be reconstructed consecutively, starting with the most significant and moving down to the least significant. As indicated earlier, in each case, the corresponding base layer is compressed using CALIC algorithm.

Fig. 3. Test images used for experiments. Each image is 512×512 in size and has 256 gray levels (8-bits).

In Table. 1, the performance of the proposed algorithm is compared with that of state-of-the-art lossless compression methods. (More results can be found in [5].) The methods included in this benchmarking include the regular (non-embedded) lossless compression methods: CALIC, JPEG2000, JPEG-LS, and gray-coded JBIG -"JBIG(gray)"; and embedded compression using JBIG (independent bit-planes), and the level-embedded scheme proposed in this paper. The different level embeddings are denoted as L.E. 1, L.E. 2, ... , L.E. 7 for the cases corresponding to 1, 2, ... 7 enhancement layers. In our experiments, CALIC provided the best compression rates for non-embedded compression. Therefore, in Table. 1, we tabulate results for all non-embedded schemes and the level-embedded scheme proposed here as the percentage increases in bit-rate with respect to the CALIC algorithm.

From the table, it is apparent that JPEG-LS and JPEG2000 offer fairly competitive performance to CALIC with only modest increases in bit rate. Nonetheless, just like CALIC these methods are not bit-plane scalable. JPEG2000 provides resolution and distortion scalability but not bit-plane scalability. In its default mode, JBIG provides bit-plane scalability, however at a significant loss of coding efficiency (almost a 35% increase in bit rate over CALIC, on average). The performance of JBIG is significantly improved when pixel values are gray-coded prior to separation into bit-planes. This corresponds to the row labeled "JBIG(gray)" in the table. However, in this case the resulting compressed bitstream is no longer bit-plane scalable for the original image data. The level embedded compression scheme does significantly better than JBIG. For a small number of embedding levels the penalty is quite small with up to 4 enhancement layers requiring under 8% increase in bit-rate over CALIC.

The proposed method incurs a penalty which increases roughly linearly with increase in the number of enhancement layers (embedded bit-planes). In a hypothetical application, where 2 bit-planes are embedded, for instance, to truncate 8-bits to 6-bits in a digital camera, the increase in bit-rate is 3% on the average. This number is quite competitive with the non-scalable JPEG-LS and CALIC algorithms in view of the added functionality. It is also better than the corresponding rate for the JPEG2000 algorithm. When all bit-planes are embedded the penalty increases to 15%. This is significantly better than the JBIG algorithm in its bit-plane scalable mode. However, it is considerably worse than the JPEG2000, where alternate scalability is provided. The degradation at higher levels of embedding is not a major concern because most applications of level-embedded compression are likely to require only a small number of embedded bit planes.

4. CONCLUSIONS

We present a level-embedded lossless image compression method, which enables bit-plane scalability, or more generally level scalability. In situations, where the resulting compressed bit-stream needs to be truncated to produce a lower bit rate (and lower quality) image, the proposed scheme guarantees freedom from compression induced spatial artifacts and tight bounds on per pixel maximum error, making it especially suitable in certain medical and legal imaging applications. Experimental results comparing the method with state-of-the-art lossless compression methods indicate that level scalability is achieved with only a small penalty in the compression efficiency over regular (non level-embedded) compression schemes.

5. REFERENCES

[1] X. Wu, "Lossless compression of continuous-tone images via context selection, quantization, and modelling," *IEEE Trans. on Image Proc.*, vol. 6, no. 5, pp. 656–664, May 1997.

[2] ISO/IEC 14495-1, "Lossless and near-lossless compression of continuous-tone still images- baseline," 2000.

[3] ISO/IEC 15444-1, "Information technology–JPEG 2000 image coding system–part 1: Core coding system," 2000.

[4] ISO/IEC 11544, "Information technology - coded representation of picture and audio information - progressive bi-level image compression," 1993.

[5] M.U. Celik, G. Sharma, and A.M. Tekalp, "Gray-level-embedded lossless image compression," *to appear in EURASIP Image Comm.*

[6] N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*, Prentice Hall, Englewood Cliffs, NJ, 1984.

[7] M. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS," *IEEE Trans. on Image Proc.*, vol. 9, pp. 1309–1324, Aug. 2000.

Table 1. Performance of level-embedded compression scheme against different lossless compression methods. Percent increase with respect to CALIC is indicated.

Image		Avg.	F-16	Mand	Boat	Barb	Gold	Lena
Comp. Method		Best lossless compression rate (Baseline)						
CALIC (bpp)		4.40	3.54	5.66	4.15	4.42	4.58	4.08
		Percent Increase in bit-rate *wrt* baseline						
Regular	CALIC	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	JPEG2000	5.2	7.6	4.0	6.2	4.6	4.6	5.2
	JPEG-LS	3.1	1.9	2.8	2.4	6.2	1.8	3.4
	JBIG(gray)	15.0	17.5	11.2	15.8	17.6	13.7	15.8
Embedded	JBIG	35.5	46.6	26.2	35.8	36.3	33.6	39.7
	L.E. 1	1.1	2.0	0.2	1.6	1.4	0.7	1.1
	L.E. 2	3.0	4.1	0.9	4.1	4.1	2.2	3.7
	L.E. 3	5.1	7.0	2.2	6.3	6.3	4.7	5.8
	L.E. 4	7.8	10.6	3.4	9.9	10.1	6.6	8.6
	L.E. 5	10.5	13.7	5.3	12.4	14.0	8.5	11.7
	L.E. 6	12.8	15.9	6.6	14.5	17.5	10.7	14.4
	L.E. 7	14.9	18.8	7.6	16.4	20.0	12.6	17.5