

## Embedded Image-Domain Adaptive Compression of Simple Images\*

Youngjun Yoo

DSP Solutions R&D Center  
Texas Instruments Inc.  
Dallas, TX 75243

Younggap Kwon and Antonio Ortega

Integrated Media Systems Center  
University of Southern California  
Los Angeles, CA 90089

### Abstract

*Popular transform coding techniques provide excellent lossy compression for natural images but fall short with “simple” images, i.e., images represented by only a small portion of the available intensity values. In this paper we propose a novel image-domain compression technique aiming at simple images, or, more generally, simple regions within a compound image. The proposed method is a context-adaptive bit-plane coder based on a binary arithmetic coder. The context information allows us to exploit the strong correlation across the bit-planes of simple images. Given the bit-plane coding structure our coder produces an SNR-scalable embedded bit stream which can be lossless when completely decoded.*

### 1 Introduction

Transform-based image coding methods such as JPEG [1] and wavelet coders [2] are excellent at compressing natural images at various compression rates. However these methods fall short when the histogram of an image has only a small number of active intensity values, i.e., a large portion of intensity values are never used to represent the image. We call such images “simple.” Examples of simple images of interest include: *bi-level images; gray-scale or color images scanned from bi-level images; computer generated graphics with simple textures, cartoons, screendumps, diagrams, etc.*

It is intuitively obvious that simple images are easier to compress than natural images, with high compression rates being achievable even if lossless compression is used. However a transform-domain representation is not necessarily as useful for simple images as it is for natural images. For example, transform coders produce a significant number of coefficients which have to be represented with higher precision than the original pixels. This is especially inefficient when the original image was simple and contained just a few different gray-scale levels. Also strong Markov statistics in simple images, which can facilitate data

source modeling and adaptive coding techniques, are not preserved through transform. Moreover, while the sharp edges in simple images are usually important visual objects requiring lossless or near-lossless compression, lossy transform coders can result in blurring sharp edges.

Our goal in this paper is to introduce an efficient image-domain compression algorithm that is particularly useful for simple images. The algorithm is proposed in the context of lossless and near-lossless compression since simple images are not only easier to compress but visually more susceptible for compression artifacts than natural images. However, even in a lossy compression environment, these types of approaches are useful. In particular, for compound images which contain both natural and simple regions, efficient performance can be achieved even with lossless or near-lossless compression of the simple image regions.

A previous work that treats the simple image/subimage with a dedicated compression algorithm can be found in the course of RICOH’s CREW algorithm development [3, 4]. An early version of CREW depends on wavelet transform to obtain good coding performance for general continuous-tone images [3]. Performance degradation of this wavelet-based coder, when applied to images having “unusual” first-order Markov statistics, is considered in the improved version of CREW [4] which opts for a *binary mode*. CREW’s binary mode is an image-domain bit-plane coding algorithm based on a bi-level image coder similar to JBIG [5], the ISO/ITU bi-level image compression standard. This algorithm is applied to parts of the image which are deemed to be simple based on a suitable segmentation criterion. With this option, CREW has achieved significant gains in compression performance for simple images and compound images containing large simple regions.

Our proposed coder is also based on bit-plane coding in the image domain. Thanks to its bit-plane coding nature, the proposed coder has the important feature of generating an embedded bit stream for the

---

\*This work was supported in part by Texas Instruments.

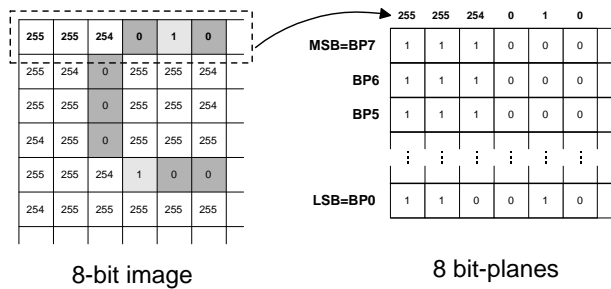


Figure 1: Bit-planes obtained from an 8-bit gray-scale image. In this figure one line of pixels in the image is converted into bit-planes.

compressed image. However there is a vital distinction between our algorithm and CREW’s binary mode in that our algorithm uses a novel context modeling technique to effectively exploit the strong correlation across the bit planes of simple images.

JPEG-LS [6] and CALIC [7] are other well-known image-domain algorithms that offer good lossless compression performance for the general class of continuous-tone images. However, as these algorithms have been developed with more emphasis on natural image compression, they are not expected to take the full advantage of simple images. Moreover both JPEG-LS and CALIC are not able to generate embedded bit streams while our proposed coder enables progressive transmission or SNR scalability of the compressed image through the embedded bit stream.

The paper is organized as follows. In Section 2 we describe our image coding algorithm, named *Embedded Image-Domain Adaptive Compression* (EIDAC). As a means to increase the compression efficiency of EIDAC, *histogram compaction* is also discussed in this section. Experiments and results are presented in Section 3.

## 2 Coding of simple images

### 2.1 Description of EIDAC algorithm

Given an input image, the compression algorithm processes the bit planes of the image successively, from the most significant bit (MSB) plane to the least significant bit (LSB) plane, thus achieving an embedded bit stream. See Fig. 1 for an example of the bit planes obtained from an 8-bit image. An adaptive binary arithmetic coder is used to compress each bit plane. The conditional probability model of the adaptive arithmetic coder is instrumental for good compression performance and it is described in detail below.

For an input bit to the binary arithmetic coder in

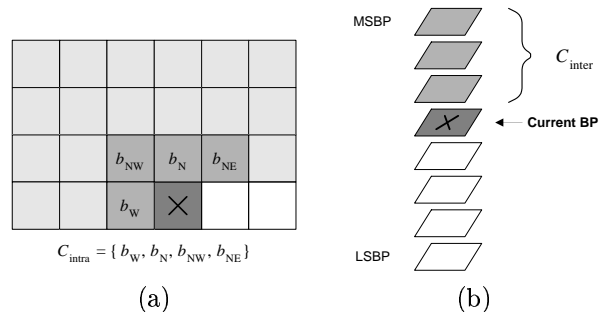


Figure 2: Causal contexts to generate the conditional probability model for an input bit to the binary arithmetic coder in EIDAC. Two sets of bits, (a)  $C_{intra}$  and (b)  $C_{inter}$ , are used to exploit the correlations among neighboring bits within the current bit-plane and across the bit-planes, respectively.

the current bit plane, the context model is composed of two sets of bits: one set containing neighboring bits in the same bit plane,  $C_{intra}$ , and the other containing bits from the other bit planes,  $C_{inter}$ . To avoid the use of overhead we use causal contexts, i.e., only the bits from the causal past, which are thus known to the encoder and the decoder, are used. An example of causal context model is shown in Fig. 2.

The reason for considering two separate sets of context bits is clear. First, the strong spatial correlation among image pixels also extends to some extent to its bit planes and thus using  $C_{intra}$  will be beneficial. Second,  $C_{inter}$  is particularly useful because we are running our algorithm on simple images. Consider as an example a bi-level image scanned as an 8-bit image as shown in Fig. 1. In this case the two more likely intensity levels are going to be  $0000\ 0000_2$  and  $1111\ 1111_2$ . There are possibly a few levels that differ only in some of the LSB’s from these two predominant levels. Thus, if a particular pixel’s MSB is 1, the less significant bits would also be 1 with high likelihood. This motivates the fact that, for simple images, gain can be achieved by using information about the previously scanned bit-planes and thus justifies the potential benefits of  $C_{inter}$ . Note that the situation is similar even if the original images are not scanned from a bi-level source. Since the number of gray-scale levels is much less than the maximum of 256 in those simple images, for each given pattern of, say, 4 MSB’s, the number of possible combinations of 4 LSB’s is likely to be less than 16. This can result in high compression ratios when our context modeling method is used.

Note that, for the simple context modeling exam-

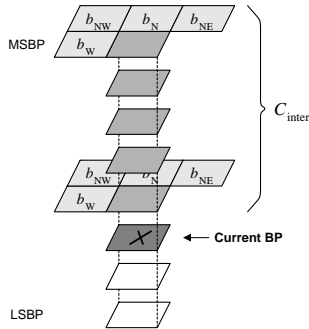


Figure 3:  $C_{inter}$  employed for the experiment in Sec. 3. Although we can include more bits from the upper bit planes for more compression gain, we limit the  $C_{inter}$  bits only to the ones shown in this figure to avoid excessive computation and memory requirements.

ple of Fig. 2, we use only 4 bits for  $C_{intra}$  and up to 7 bits for  $C_{inter}$ . Though this context model can be good enough for bi-level or near-bi-level images, a more sophisticated context model can be employed to fully exploit the available information. For instance, any bits in the *upper* bit planes (i.e., the bit planes already encoded) can be used to define  $C_{inter}$  while a larger causal neighborhood can be considered to define  $C_{intra}$ . In fact, for images with strong spatial correlation, we can achieve more compression gain by including the surrounding pixels’ upper bit plane information. However we cannot arbitrarily increase the number of bits used for context modeling due to the associated increase of algorithm complexity—especially due to exponential increase of memory requirement. For the experiment in the later section, we employ a more complex  $C_{inter}$  as shown in Fig. 3.

Once the context model has been selected, the compression algorithm processes each bit plane in the usual raster scan order. The only necessary side information is the dimension of the input image. Further compression can be achieved by using *histogram compaction* as described below.

## 2.2 Histogram compaction

If the number of active intensity levels is small, we can use side information to specify what pixel values are in use and represent each pixel intensity with a reduced number of bits. For example, suppose that only 29 active pixel values are used in a given image. Then we can represent the original image using only 5 bits per pixel (i.e. we need to code 5 bitplanes, instead of 8) after specifying those 29 levels explicitly as side information. The decision on whether to use his-

togram has to take into account the trade-off between bits added by overhead and bits saved by removing redundant bit planes. For example, suppose we specify the active pixel values with a 256-bit overhead (for 8-bit gray-scale images) where the  $i$ -th bit is set to 1 if the grayscale level  $i - 1$  is present in the original image or set to 0 otherwise. Obviously, histogram compaction would not be worth considering if the bit plane reduction savings are less than the 256 bits of overhead.

## 3 Experiments and results

We implemented the EIDAC algorithm using 4 bits for  $C_{intra}$  as in Fig. 2: the adjacent bits to the current bit in W, N, NW, and NE directions. The number of bits used in  $C_{inter}$  depends on the bit plane and can be as many as 15 bits. When encoding the  $k$ -th bit plane BP $k$ ,  $k = 0, \dots, 7$ , we consider  $7 - k$  bits from the upper bit planes at the pixel location for the current bit, where BP7 denotes the MSB plane and BP0 the LSB plane.  $C_{inter}$  also includes 4 bits from each of BP7 and BP $k + 1$  at the pixel locations for the  $C_{intra}$  bits. See Fig. 3 for the  $C_{inter}$  bits.

We start by emphasizing that EIDAC, in its version without histogram compaction, generates an embedded bit stream as shown in Table 1 and Fig. 4. Embedded coding enables both lossy compression and lossless compression, and is a convenient feature for a number of applications in addition to facilitating rate control.

The compression performance of EIDAC and other lossless image-domain coders for the test images are summarized in Table 2, where we use the test images used by the Piecewise-Constant Image Coder (PWC) [8], a lossless algorithm aimed at compression of computer graphics images. These images are available on the Internet [9] and were converted first to grayscale, as our current version of EIDAC can only handle 8-bit gray-scale images.

To the best of our knowledge PWC is the best coder to be found in the literature for compression of simple images so that this coder serves here as a benchmark. The results in the column labeled I-JBIG are obtained by applying JBIG independently to the the bit planes of the test images. In the next column are the compression results using CompuServe’s Graphical Interchange Format (GIF). We also consider Context-based Adaptive Lossless Image Coder (CALIC), one of the best lossless coding algorithms for general gray-scale images [10]. Note that I-JBIG and EIDAC are the only embedded schemes among those in Table 2.

EIDAC is used with and without histogram compaction (EIDAC and EIDAC<sub>H</sub>, respectively, in the

Reconstruction Level	BP7	BP6	BP5	BP4	BP3	BP2	BP1	BP0
Data Size	3,100	6,026	8,505	8,686	8,873	9,026	9,167	9,348

Table 1: Transmitted data size (in bytes) for progressive image reconstruction experiment for `books`.  $BP_k$  corresponds to reconstruction down to the  $k$ -th bit plane. Note that BP7 represents a bi-level reconstruction image and BP0 the lossless reconstruction.

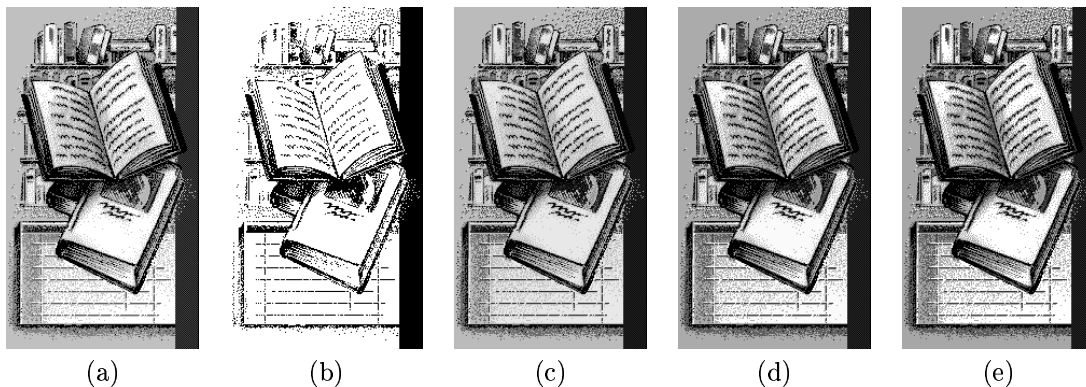


Figure 4: Progressive reconstruction example associated with Table 1. (a) The original image `books` (b) Black-and-white reconstruction corresponding to BP7 (c) 3-bit reconstruction from BP5 (PSNR = 24.43dB) (d) 5-bit reconstruction from BP3 (PSNR = 36.68dB) (e) 7-bit reconstruction from BP1 (PSNR = 49.59dB)

Image	RAW	I-JBIG	GIF	CALIC	PWC	EIDAC	EIDAC <sub>H</sub>
<code>benjerry</code>	27,960	6,876	4,386	6,094	<b>2,408</b>	4,027	3,541
<code>books</code>	56,922	14,767	11,117	23,229	8,601	9,348	<b>8,577</b>
<code>ccitt01</code>	598,500	6,991	9,255	7,914	<b>5,078</b>	6,073	5,784
<code>cmpndd</code>	393,216	73,861	71,825	68,704	<b>49,498</b>	58,242	–
<code>cmpndn</code>	393,216	61,663	58,780	55,564	<b>37,043</b>	45,968	–
<code>flax</code>	17,340	1,037	799	759	129	176	<b>95</b>
<code>gate</code>	60,912	25,974	22,244	24,700	<b>15,301</b>	19,827	19,732
<code>music</code>	12,321	2,245	1,955	2,340	<b>710</b>	1,493	890
<code>netscape</code>	61,200	19,898	14,730	19,915	<b>9,796</b>	13,130	11,096
<code>pattern</code>	18,495	6,201	1,729	2,040	1,074	1,209	<b>1,067</b>
<code>sea_dusk</code>	157,300	2,172	6,219	1,736	<b>622</b>	2,416	2,340
<code>stone</code>	22,477	5,042	4,703	14,755	3,981	3,417	<b>2,780</b>
<code>sunset</code>	307,200	79,251	94,248	77,049	<b>48,702</b>	66,022	–
<code>winaw</code>	294,345	25,987	18,562	34,045	<b>11,414</b>	14,580	12,796
<code>yahoo</code>	27,140	8,831	6,930	7,131	<b>4,178</b>	5,590	–
Total	2,448,544	340,796	327,482	345,975	198,535	251,518	244,520

Table 2: Comparison of compressed file sizes (in bytes) by various lossless compression algorithms for the (gray-scale-converted) PWC test image set. Note that images `cmpndd`, `cmpndn`, `sunset` and `yahoo` have more than 128 levels of gray and therefore no histogram compaction is performed. For these images the rate achieved without histogram compaction is used to compute the totals.

Image		CALIC	PWC	EIDAC
aerial2	(362,144)	6.29	6.34	<b>6.24</b>
bike	(327,780)	5.27	5.34	<b>4.98</b>
cafe	(327,680)	6.23	6.45	<b>5.93</b>
goldhill	(262,144)	<b>4.63</b>	5.37	5.38
lena	(262,144)	<b>4.11</b>	4.97	4.90
woman	(327,680)	<b>5.17</b>	5.57	5.39
Average		5.31	5.69	5.46

Table 3: Lossless compression rates (bits/pixel) for a set natural 8-bit images. In the parentheses are the raw image file sizes in bytes.

table). Even without histogram compaction, EIDAC can outperform the other compression schemes except PWC. By using histogram compaction, EIDAC's achieves better compression performance than PWC for some of the test images. It can be observed that EIDAC<sub>H</sub> tends to outperform PWC for the simpler images in the test set.

Table 3 provides results for a set of natural images. Whereas neither EIDAC nor PWC were specifically designed with natural images in mind, this comparison is useful to evaluate the robustness of these algorithms when used on images that are not as simple as expected. It can be seen that EIDAC, the only embedded algorithm considered, provides competitive performance, and even outperforms CALIC for certain images. It can also be seen that it is more robust than PWC to the lack of simple of characteristics in the source images.

As a summary we have proposed EIDAC as an efficient image-domain algorithm for compression of simple images. Our experimental results show that EIDAC outperforms other schemes such as GIF or CALIC, but not PWC, which is also specifically targeted at simple images. However, an important feature of EIDAC is its ability to produce embedded bit streams, unlike CALIC, GIF, or PWC. The EIDAC algorithm has been proposed to the ongoing JPEG 2000 standardization effort, as a part of the algorithm contribution of [11]. EIDAC is used to complement an embedded wavelet coder for tiling-based coding of compound images where classification is used to determine which parts of the image are simple. We are currently exploring more sophisticated context modeling for both  $C_{intra}$  and  $C_{inter}$  as well as techniques for histogram compaction that preserve embedded coding property of EIDAC.

## References

- [1] W. Pennebaker and J. Mitchell, *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, 1994.
- [2] Y. Yoo, A. Ortega, and B. Yu, "Image subband coding using context-based classification and adaptive quantization," *IEEE Trans. on Image Proc.*, Jun. 1997. Submitted.
- [3] A. Zandi, M. Boliek, E. L. Schwartz, M. J. Gormish, and J. D. Allen, "CREW lossless/lossy image compression - Contribution to ISO/IEC JTC 1.29.12." ISO/IEC JTC1/SC29/WG1 N196, Jun. 1995.
- [4] A. Zandi, M. Boliek, E. L. Schwartz, and A. Keith, "CREW second evaluation - ISO/IEC JTC1/SC29/WG1." ISO/IEC JTC1/SC29/WG1 N252, Nov. 1995.
- [5] ITU-T Recommendation T.82, "Information Technology - Coded Representation of Picture and Audio Information - Progressive Bi-level Image Compression," 1993.
- [6] ISO/IEC JTC1/SC29/WG1, "JPEG-LS part-2 WD." ISO/IEC JTC1/SC29/WG1 N938, 1998.
- [7] X. Wu and N. Memon, "Context-based, adaptive, lossless image codec," *IEEE Trans. on Commun.*, vol. 45, pp. 437-444, Apr. 1997.
- [8] P. J. Ausbeck Jr, "Context models for palette images," in *Data Compression Conference*, (Snowbird, Utah), Mar. 1998.
- [9] <http://sipi.usc.edu/~younggap/EIDAC/testimage/>.
- [10] <http://www.csd.uwo.ca/~wu/index.html>.
- [11] J. Liang, C. Chrysafis, A. Ortega, Y. Yoo, K. Ramchandran, and X. Yang, "The predictive embedded zero-tree wavelet (PEZW) coder: A highly scalable image coder for multimedia application." ISO/IEC JTC1/SC29/WG1 N803, Nov. 1997.