AN EMBEDDED DCT APPROACH TO PROGRESSIVE IMAGE COMPRESSION

Jiankun Li, Jin Li and C.-C. Jay Kuo

Integrated Media Systems Center and Department of Electrical Engineering-Systems
University of Southern California, Los Angeles, CA 90089-2564
email: {jiankunl,lijin,cckuo}@sipi.usc.edu

ABSTRACT

Motivated by Shapiro's embedded zerotree wavelet coding (EZW) and Taubman and Zakhor's layered zero coding (LZC), we propose a layered DCT image compression scheme, which generates an embedded bit stream for DCT coefficients according to their importance. The new method allows progressive image transmission and simplifies the rate-control problem. In addition to these functionalities, it provides a substantial rate-distortion improvement over the JPEG standard when the bit rates become low. For example, we observe a bit rate reduction with respect to the JPEG Huffman and arithmetic coders by about 60% and 20%, respectively, for a bit rate around 0.1bpp.

Keywords: Layered coding, progressive transmission, arithmetic coder, JPEG.

1. INTRODUCTION

The JPEG [1] compression standard is widely adopted in still images coding. The process consists of three stages: the block DCT transform, uniform quantization with a quantization table, and the Huffman (used in the baseline system) or the arithmetic (used in the extended system) entropy coder. In JPEG, one DCT coefficient has to be completely encoded before the coding of the next coefficient. More recently, a concept known as *embedded* or *layered* coding was proposed by Shapiro [3] and further developed by Taubman and Zakhor [4] in the context of wavelet coding. In contrast with the coefficient-by-coefficient approach, they adopted a new approach, in which each coefficient is successively quantized into a certain number of bits. The most significant bits of all coefficients are grouped together to form one layer and encoded first. Then, we move to the layer of the second significant bits and so on. The coding order is consistent with the importance of each bit so that the encoder and the decoder can stop at any time. The embedding property is essential for progressive image transmission. It also greatly simplifies the rate-control problem and allows unequal error protection for robust image transmission.

In this research, we generalize the layered coding to image compression methods based on the block DCT transform. Even though the generalization is not difficult, we feel that it is valuable to have its detailed implementation available. This is the main objective of the work. Besides providing more functionalities, the new coder gives a substantially better rate-distortion performance than the JPEG standard especially at low bit rates.

The work is organized as follows. We introduce the concept of layered coding in Section II. The detailed implementation is presented in Section III. Experimental results are given in Section IV.

2. OVERVIEW OF LAYERED CODING OF DCT COEFFICIENTS

The proposed layered DCT coding adopts the 8×8 block DCT transform which is the same as JPEG. The main differences lie in the quantization and the entropy coding schemes. For a given 8×8 DCT block, we arrange the 63 AC coefficients in a zig-zag order and denote them by $C_1, C_2, \cdots, C_{63}$. Suppose that the AC coefficients after quantization take a value ranging from -32767 to 32767 so that each of them requires 16 bits for representation (including the sign bit). They are labeled with $S, B_1, \cdots, B_{15}$, where S is the sign bit, B_1 the most significant bit (MSB) and B_{15} the least significant bit (LSB). Values in such a bit matrix form a set of *intermediate symbols*. JPEG encodes these symbols in a coefficient-by-coefficient manner, with a 2-D run-level Huffman coder adopted in the JPEG baseline system while an arithmetic coder adopted in the extended system. The JPEG encoding scheme is illustrated with Fig. 1. First, all bits of coefficient C_1 are encoded, then all bits for coefficient C_2, and then C_3 and so on. The arithmetic coder in the JPEG extended

0-7803-3258-X/96/$5.00

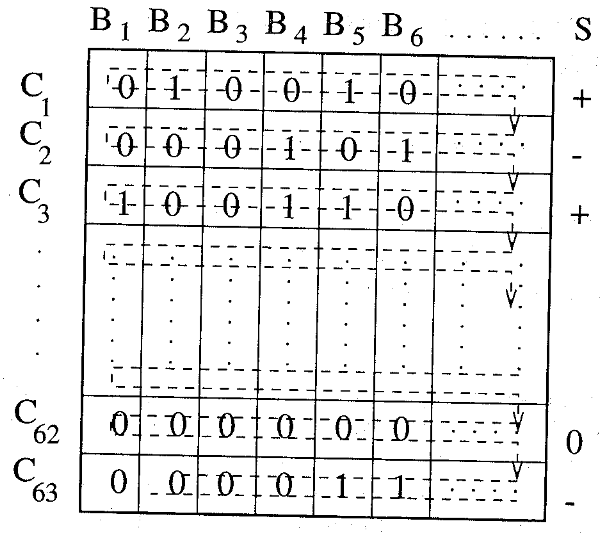

Figure 1: Bit scanning orders for the JPEG.

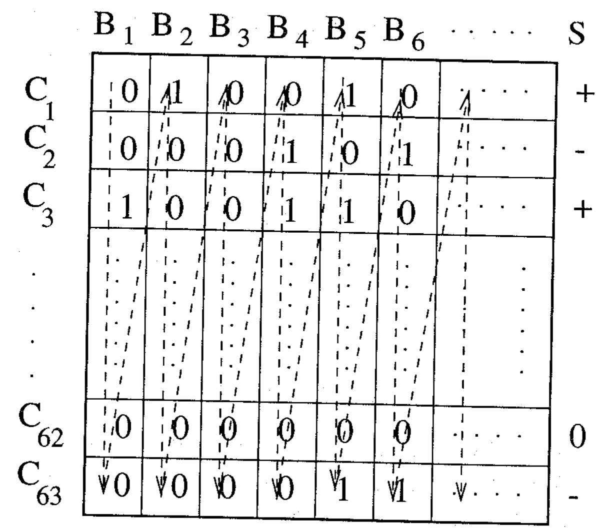

Figure 2: Bit scanning orders for the layered DCT coding scheme.

system adopts a two-way scan. The first scan starting from LSB B_{15} to MSB B_1 locates the most significant bit B_i for the current coefficient C_j. The second scan records bits B_i, B_{i+1}, $\cdots$, B_{15} and the sign bit S. For more details of JPEG, we refer to [1]. One major issue with JPEG is the rate control problem. It is in general difficult to estimate the number of coding bits generated for a given Q factor. Note also that truncating the compressed bit stream at an arbitrary point is equivalent to deleting the bottom portion of the image, since it is encoded sequentially from one block to another.

In this research, we demonstrate that by adopting a different coding order for intermediate symbols, we convert the conventional JPEG coder into an embedded coder called the *layered DCT coder.* Consider the grouping of all bits B_i of coefficients C_j, $1 \leq j \leq 63$, into layer L_i. Now, the new coder first encodes the most significant layer L_1, then layers L_2, L_3 and so on. Within each layer L_i, the coding follows the coefficient order, i.e. starting with coefficient C_1, then C_2, C_3, $\cdots$, C_{63}. This coding order is illustrated in Fig. 2. One important advantage of the layer-by-layer coding is that the resulting bit stream has the embedding property. Since the output bit stream is organized in an order of decreasing importance, rate-control can be easily achieved by simply truncating the bit stream according to the desired coding budget, or the desired coding quality. The layered DCT coding is also suitable for progressive transmission for the more important coding bit is always transmitted prior to the less important bit. Another advantage of the layer-by-layer coding is that the new scheme provides a better rate-distortion trade-off especially at low bit rates, which will be demonstrated in section IV.

3. DETAILED IMPLEMENTATION

We describe the detailed implementation of the proposed layered DCT compression algorithm in this section. The overall framework of our algorithm is depicted in Fig. 3. The distinctive feature of the layered DCT coder is that the quantization and the entropy coding procedures are carried out successively.

Step 1: Block DCT transform and coefficient scaling

As done in JPEG, the input image is partitioned into 8×8 blocks, and the block DCT transform is applied to each block. The DCT coefficients C'_j are scaled with a standard JPEG quantization table Q, i.e. $C_j = \frac{C'_j}{Q_j}$, $j = 0, \cdots, 63$. The scaling is performed to emphasize the visual importance of low frequency components.

Step 2: Coding of DC coefficients

DC coefficients of neighboring blocks are highly correlated. It can be either encoded with a differential layer coding [4] or encoded in the same way as the JPEG arithmetic coder. It is observed that the performance of the two schemes are very similar. In the experiment reported in Section IV, the latter approach is adopted.

Step 3: Successive quantization of AC coefficients

The main differences between the proposed method and JPEG are in the quantization scheme and the entropy coder. After the DCT transform and scaling,

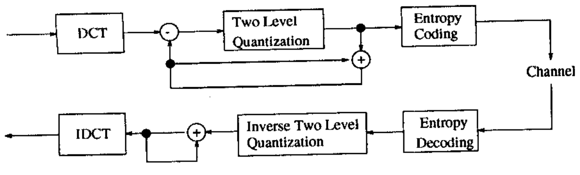

Figure 3: The framework of layered DCT coding.

JPEG applies an one-step quantization which maps each DCT coefficient to a value in a finite index set. The value is then converted to an intermediate symbol and encoded by an entropy coder. In the proposed new scheme, we adopt a successive quantization procedure which is achieved not in one step but with several successive steps. Roughly speaking, at layer i, the DCT coefficient is only quantized up to the precision of the *significant threshold* T_i. Then, the quantization result of layer i is refined with a smaller significant threshold T_{i+1} at layer $i+1$.

To initialize the process of successive quantization, we set all coefficients as insignificance and search the whole image for the maximum absolute value of the AC coefficients, which is denoted by T_0. Then, we apply the significant identification rule with significant threshold $T_1 = T_0/2$ to construct layer L_1. That is, for each scaled AC coefficient C_j, if its magnitude is greater than T_1, we use symbol 1 to encode its significance, and then record its sign S. Otherwise, we generate symbol 0 to indicate that it is still insignificant. Note that only the significant symbol needs a sign. In terms of mathematics, we have

$$\begin{array}{llll} C_j > T_1, & B_{j,1} = 1 & S_j = + & E_{j,1} = C_j - 1.5T_1, \\ C_j < -T_1, & B_{j,1} = 1 & S_j = - & E_{j,1} = -C_j - 1.5T_1, \\ \text{otherwise}, & B_{j,1} = 0 & & E_{j,1} = C_j. \end{array}$$

where symbol $E_{j,1}$ in the last column denotes the quantization residue at layer L_1. For each advanced layer L_{i+1}, $i = 1, 2, \cdots$, we refine the significant threshold by half, i.e. $T_{i+1} = T_i/2$, and quantize all AC coefficients accordingly. If the AC coefficient to be coded is insignificant in all previous layers, the significance identification rule is applied. Otherwise, the refinement quantization rule is applied. These two rules can be summarized as follows.

1. rule of significance identification:

$$\begin{array}{llll} E_{j,i-1} > T_i, & B_{j,i} = 1 & S_j = + & E_{j,i} = E_{j,i-1} - 1.5T_i, \\ E_{j,i-1} < -T_i, & B_{j,i} = 1 & S_j = - & E_{j,i} = -E_{j,i-1} - 1.5T_i, \\ \text{otherwise}, & B_{j,i} = 0 & & E_{j,i} = E_{j,i-1}, \end{array}$$

2. rule of refinement quantization:

$$\begin{array}{lll} E_{j,i-1} \geq 0, & B_{j,i} = 1 & E_{j,i} = E_{j,i-1} - T_i, \\ E_{j,i-1} < 0, & B_{j,i} = 0 & E_{j,i} = -E_{j,i-1} - T_i. \end{array}$$

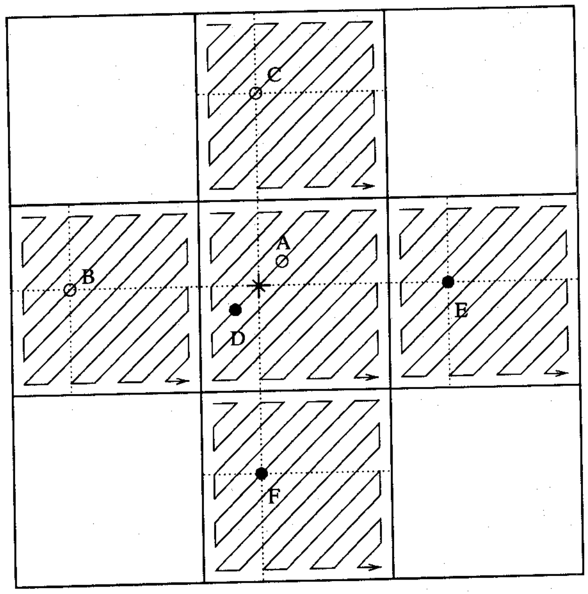

Figure 4: Illustration of significant symbol coding context, where $*$ is the current coding position, $\circ$ is the significant symbol of current layer and $\bullet$ is the significant symbol in the previous layer.

The generated symbols $B_{j,i}$ and S_j provide a binary representation of the coefficient C_j normalized by T_0 as shown in Fig. 2. These symbols are generated with a decreasing order of importance and will be coded layer by layer in the next step.

Step 4: Context adaptive arithmetic coding

All symbols generated in successive quantization is binary, which can be coded efficiently by an arithmetic coder. Besides, we can predict the location of the significant coefficient with a context adaptive arithmetic coder used in the JPEG extended system. We also encode the refinement symbol and the sign with a specific context. The context adaptive arithmetic coder is a highly efficient entropy coder. Its average coding rate is close to the entropy of the source with low computational complexity. Its implementation only requires addition and shifting operations. There is no need for training or assuming the initial probability distribution so that it is parameter free. The source probability distributions p_0 and p_1 are estimated on the fly and implemented with a look-up table. The source probability distribution is represented as an 8 bit (1 byte) status of the coder, which includes 7 bits for the probability table and 1 bit for the most frequently appeared symbol (MFS). The small coder status enables the construction of a parallel coder for a compound source. We refer to [2] for more details.

For significance identification, our coding context

Image	PSNR	Rate of JPEG		Layered DCT		
		Huff.	Arith.	Rate	J-Huff.	J-Arith.
Lena	24.23	0.125	0.054	0.041	67.2%	24.1%
	30.84	0.25	0.198	0.192	23.2%	3.0%
	34.76	0.5	0.463	0.453	9.4%	2.4%
	37.92	1.0	0.954	0.940	6.0%	1.5%
Boat	24.26	0.15	0.078	0.064	57.3%	18.1%
	27.54	0.25	0.184	0.181	27.6%	1.6%
	31.02	0.5	0.437	0.491	1.8%	-12.4%
	34.56	1.0	0.931	0.902	9.8%	3.1%

Table 1: Performance Comparison for Lena and Boat.

consists of 6 bits 'A' to 'F' which are illustrated in Fig. 4. For each circle position, we use 1 bit to represent the current significance status of the symbol. Among the 6 bit context, 'A' and 'D' are frequency prediction points inside the current coding block, 'B', 'C' and 'E', 'F' are the spatial prediction points of the neighboring blocks. 'A', 'B' and 'C' are coded before the current coding coefficient '*' and therefore belong to the current coding layer. In contrast, 'E', 'D' and 'F' are coded after the current coding coefficient '*' and belong to the previous coding layer. The 6-bit context classifies the coding of current coefficient '*' into $2^6 = 64$ categories, and a separate adaptive arithmetic coder is assigned to each category. The layer-by-layer coding turns out to be more efficient than the coefficient-by-coefficient JPEG coding especially at low bit rates. This can be explained by the fact that we only predict whether a bit is significant or not in the layer-by-layer coding. Compared to the prediction of the coefficient value, this task is easier and can be done more accurately.

4. EXPERIMENTAL RESULTS

Experiments are conducted to compare the new layered DCT coder with the JPEG standard using the Huffman and the arithmetic entropy coders. The images used in the experiments are Lena and Boat of size 512 × 512. For fair comparison, we strip the header of the JPEG coded bit stream. We use the same quantization factor Q for JPEG Huffman and arithmetic coders, which results in the same coding image quality for both coders. Due to the embedding property of the proposed layered DCT coder, we can truncate the coded bit stream when it reaches the same PSNR values as JPEG. By doing so, we can compare the bit rates for the three coding schemes with the same quality. The results are shown in Table 1. We list the coding bit rate reduction with respect to JPEG Huffman and arithmetic coders in the last two columns of the table. The rate-distortion

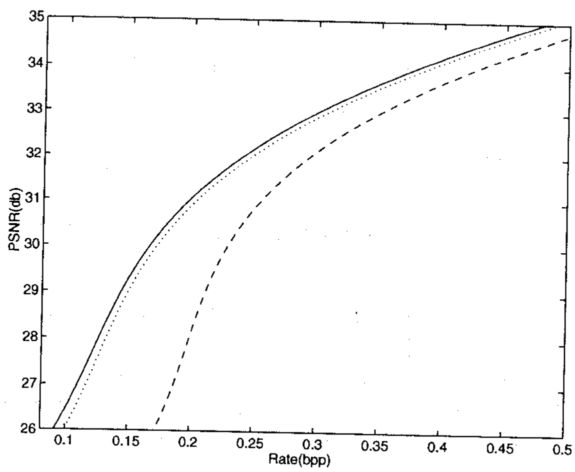

Figure 5: Rate-distortion performance of Lena using (a) JPEG Huffman coder (dashed line), (b) JPEG arithmetic coder (dotted line) and (c) the proposed layered DCT coder (solid line).

tradeoff curves for Lena are also depicted in Fig. 5.

We see from the experimental results that the layered DCT coder significantly outperforms the JPEG Huffman coder and also outperforms the JPEG arithmetic coder in most cases. The improvement is more substantial when the bit rates become lower. For example, it outperforms the JPEG Huffman and arithmetic coders by about 60% and 20%, respectively, when the bit rate is around 0.1 bpp. In addition to the superior rate-distortion performance, the layered DCT coder possesses the embedding property which makes progressive image transmission and rate-control easier to attain.

5. REFERENCES

[1] W. B. Pennebaker, *JPEG Still Image Data Compression Standard*, New York: Van Nostrand Reinhold, 1993.

[2] D. L. Duttweiler and C. Chamzas, "Probability estimation in arithmetic and adaptive Huffman entropy coders," *IEEE Trans. on Image Processing*, Vol. 4, pp. 237–246, Mar. 1995.

[3] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. on Signal Processing*, Vol. 41, No. 12, pp. 3445–3462, 1993.

[4] D. Taubman and A. Zakhor, "Multirate 3-D Subband Coding of Video," *IEEE Trans. on Image Processing*, Vol. 3, No. 3, pp. 572–588, 1994.