Compaction of Color Images with Arithmetic Coding

Masahiro Iwahashi and Shun-ichi Masuda

Electronics R&D Laboratories, Nippon Steel Corporation
5-10-1 Fuchinobe, Sagamihara City, Kanagawa 229, Japan

1. INTRODUCTION

In this paper, we will introduce a new compaction method for color images with an arithmetic coding and a Differential Pulse Code Modulation (DPCM). In the field of lossless coding, entropy coding plays an important role. For example, Arithmetic coding and Huffman coding[1] are well-known entropy coding techniques. Arithmetic Coding (AC) is superior in many respects to Huffman coding. AC's performance is optimal without the need for blocking of input data. Its coding abilities are efficient and adaptive[2,3]. In addition to these facts, AC can be performed with no-multiplication by using L-R Arithmetic coding[4] proposed by Langdon etc. The input data for these algorithm shall be represented in the form of binary code sequences. Many applications for a compaction of binary data, black and white image[5], computer data[6,7], have been reported. On the other hand, a natural color image has data represented by a multivalue. Therefore, the input data shall be converted into binary data before feeding to AC. The Multivalue-to-Binary (M-B) Transformation is performed by means of the bit-plane method[8,9] and level-plane method, etc. With the bit-plane method, multivalue data is replaced by a fixed length binary code. Its bit-plane has a binary code sequence suitable for AC. The level -plane method is performed by comparing an input multivalue to some equipped value. As a result of each comparison, 0 or 1 is obtained in accordance with it being equal or not. Each binary code has its own length. It is so called a Variable Length Code (VLC). Some application examples based on this level-plane method and AC have been reported[10,11]. For still picture coding, the Joint Photographic Experts Group (JPEG) of the International Standards Organization (ISO) has adopted the algorithm[12]. A lossless coding based on AC is included in the JPEG's algorithm as an independent function. In this algorithm, input data is coded by DPCM at first. Then the output data is M-B transformed by VLC where a more frequently appearing multi-value is replaced by a shorter binary

0-8194-0742-9/91/$4.00

code. Therefore, the amount of bit data is compacted by this VLC. Furthermore, the binary code is fed to AC so that the data may become more compacted. But the compaction is mainly done by VLC, not by AC. As a result of our simulation, only about 5% of the data is reduced by AC in the case of compaction for natural images. In other words, the conventional method does not fully utilize AC.

Then, we propose a new method in which all of the data compaction depends on AC. In our method, M-B transformation is carried out by using an Entropy-ordered Binary code (EBC). After the transformation, each binary code is linked together to make a binary code sequence for AC. The M-B transformation based on EBC means that a more frequently appearing multi-value is replaced by a less entropy rate binary code. Through this process, a more suitable binary code which has a low entropy rate (including many 0 codes) can be obtained. In this method, all compaction depends on AC, so the AC is fully utilized. In addition to that, the M-B transformation is performed by using fixed length coding instead of variable length coding because the EBC has a fixed length. This means the proposed method has more reduced hardware complexity than a conventional one. As a result of computer simulations, we have confirmed that the compaction efficiency of the proposed method for natural images is better than or equal to that of a conventional one.

2. CODING ALGORITHM

Fig.1 shows the typical configuration of a still picture encoding algorithm. Decoding is performed by an inverse procedure of coding. The coding algorithm is composed of three components, DPCM, M-B transformation and AC. In the following section, two kinds of M-B transformation are described. One of them is a conventional method based on the JPEG algorithm. The other is a new method using Entropy-ordered Binary Code (EBC).

Fig.1. Encoding Algorithm.

2.1 DPCM

Generally, image data has a strong correlation between neighboring pixels. Using this character, the value of a pixel can be predicted from the pixel neighborhoods. The output value of DPCM (prediction error) is close to zero if the predictor has high precision. In this method, the prediction error is calculated by the following equations.

$$
\begin{aligned}
y(m,n) &= x(m,n) - 0.5\,\{\, x(m-1,n) + x(m,n-1) \,\} &&, m>0 \text{ and } n>0 ,\\
y(m,n) &= x(m,n) - x(m-1,n) &&, m>0 \text{ and } n=0 ,\\
y(m,n) &= x(m,n) - x(m,n-1) &&, m=0 \text{ and } n>0 ,\\
y(m,n) &= x(m,n) &&, m=0 \text{ and } n=0 ,\\
&(m=0,1,2,\ldots,M-1,\ n=0,1,2,\ldots,N-1) && \qquad (1)
\end{aligned}
$$

Where x(m,n) denotes the value of a pixel and y(m,n) denotes a prediction error. The precision of the predictor may be improved by using more neighborhoods or selecting another value of coefficients. The above equation is optimally selected in view of hardware complexity. The output of this process has a very weak correlation, so it is like a zero-memory source. And most of these output values are zero. These characters are used in the following signal processing.

2.2 Multivalue-to-Binary (M-B) Transformation

The outputs of DPCM are represented by a multi-value with a sign. It is necessary for these values to be transformed into binary codes to use AC. The binary transformed codes are linked to each other to make a binary bit stream to be fed to AC.

2.2.1 JPEG Method

In this section, a JPEG method[11] is explained as a conventional M-B transformation. Table 1 shows the correspondence between a multi-value and a binary code. The transformation is performed by comparing input data values with some standard values. If the input value is more close to zero, the transformation is performed by fewer comparisons. And a comparison has one bit of output. Therefore, the whole bit amount of data is reduced by this process. Furthermore, the binary data is fed to AC to obtain a more reduced bit-stream. But the output code is contains many 0 signs which are the same as 1 signs. In other words, the output has an entropy rate close to 1. This means that more compaction by AC can't be expected. This fact will be shown as a simulation result described in section 3.1.

2.2.2 Proposed Method

Then, we propose a new M-B transformation using an Entropy-ordered Binary Code (EBC). The key to utilizing the capability of AC is consideration of the entropy rate. If the value of input data is close to zero, it is replaced by a low entropy rate binary code, which includes many 0 signs. Table 2 shows the correspondence between a multi-value and EBC. The entropy rate of EBC should be gradually increased according to the absolute value of the multivalued input. The code length of Table 2 is fixed for 7 bits. The multi-valued data is replaced by fixed binary data. From the viewpoint of simple implementation, fixed length coding is more desirable than VLC. The entropy rate is defined for zero-memory source by

$$E = -\sum_{i=0}^{1} P_i \log_2 P_i \tag{2}$$

where P_i denotes probability of sign i (i=0,1). The entropy rate in table 2 is calculated as the rate of a binary code length and the number of included 0 signs. For example, consider the case of a 7-bit binary code. The first code which has the lowest entropy rate is "0000000" with a 0 entropy rate. The next group which includes only one bit of "1" signs consisting of "0000001," "0000010,"..., etc. It has 7 kinds of elements and its entropy rate is defined by eq.(2) where $P_0=6/7$ and $P_1=1/7$. The i-th group of L bits EBC has an entropy rate depicted as

$$E = -(i/L)\log_2(i/L) - \{(L-i)/L\}\log_2\{(L-i)/L\} \; . \tag{3}$$

The number of group elements is calculated by

$$M = {}_L C_i = L(L-1)\ldots(L-i+1)/i! \tag{4}$$

where ${}_L C_i$ denotes a binomial coefficient. It should be noticed that a longer bit length of the code follows a lower entropy rate in the second or the third group of EBC. But the code length shall be decided by a dynamic range of input data. In addition to that , we compared two cases of EBC allocation. One of them is the case where an absolute value of a multivalue is assigned by EBC and another single bit is prepared as a sign. Another case is shown in table 2. As a result of image data compaction, more effective performance is obtained by the latter method.

Table 1. Variable Length Code.

INPUT	OUTPUT	E
0	1	0.00
1	001	0.92
-1	011	0.92
2	0001	0.81
-2	0101	1.00
3	000010	0.65
-3	010010	0.92
4	000011	0.92
-4	010011	1.00
5	00000100	0.54
-5	01000100	0.81
:	:	:

Table 2. Entropy-ordered Binary Code.

INPUT	OUTPUT	E
0	0000000	0.00
1	0000001	0.58
-1	0000010	0.58
2	0000100	0.58
-2	0001000	0.58
3	0010000	0.58
-3	0100000	0.58
4	1000000	0.58
-4	0000011	0.86
5	0000101	0.86
-5	0000110	0.86
:	:	:

2.3 Arithmetic Coding (AC)

AC is a high performance entropy coder for zero memory source based on Elias's coding. The basic concept[12] of this algorithm is explained by the use of figure 3. First, an interval [0,1) on the number line is divided into P_0:P_1. Then two kinds of intervals $[0,P_0)$ and $(P_0,1)$, represented by A_0 and A_1 respectively, are obtained. Second, if an input bit is "0," the interval A_0 is selected as a new initial interval. Next, the selected interval is divided into two parts and the new initial interval is selected according to the next bit of the input binary code. This procedure is continued until the final bit of data is fed to AC. At the end of this procedure, the interval has the size $P_0^{N0}\ P_1^{N1}$, where N_i represents the number of i (i=0,1) in the input bit stream. The location of this interval in the number line is uniquely identified by the input bit stream. The smallest data length to represent the location is

$$[\, - N_0 \log_2 P_0 \ - \ N_1 \log_2 P_1 \,] + 1 \qquad \text{bits.} \qquad (5)$$

When the number N signs involved in a binary bit stream increases, N_0/N and N_1/N is close to P_0 and P_1 respectively. Then, we can find that an increase of N leads to a compaction rate, the value of eq.(5) divided by N, close to entropy rate. Therefore AC has high performance in compacting binary zero-memory source. Many algorithms are reported to develop above the principle for hardware implementation with reduced complexity. In this report, the algorithm described in ref.8 is used as an AC algorithm.

3. SIMULATION RESULTS

In this section, the proposed algorithm and the JPEG algorithm have been simulated using some standard pictures. The simulation has been done in order to analyze encoding efficiency in the case of applying image compaction.

3.1 JPEG method

Fig.2 shows the compaction rate of the JPEG algorithms. For example, in the case of applying it to the picture "goldhill," the compaction rate for VLC is about 0.77. On the other hand, a compaction rate for AC is about 0.94. The product is the overall compaction rate. This data shows that data compaction has been mainly achieved in the VLC process, but not in the AC process. Only 5% of the VLC output is compacted in the AC process. This means that the coding capability of AC is not fully utilized.

3.2 Proposed method

A compaction rate of the JPEG algorithms and the proposed algorithm are summarized in Fig.3. This figure shows that the compaction rate of the proposed algorithm is better than that of the JPEG algorithm. The superiority is clear in the case of picture "Barbara." The compression rate of the JPEG algorithm goes down in the case of "Barbara," because "Barbara" contains fairly detailed texture. The reduction of the proposed algorithm isn't so remarkable in comparison with the JPEG results. This shows that the proposed transformation is well fitted for AC and the coding ability of AC is well utilized.

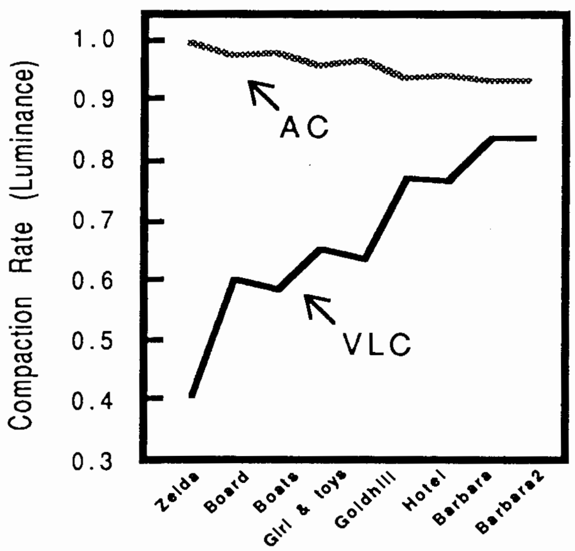

Fig.2 Compaction Rate of the JPEG Algorithm.

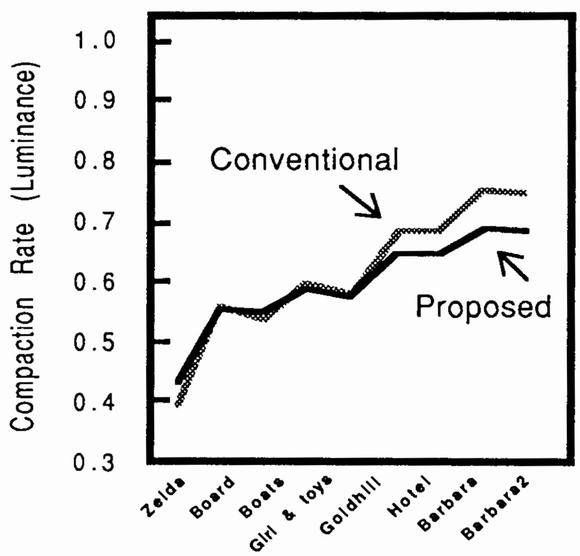

Fig.3 Comparison Between Proposed Method and Conventional Method.

4. CONCLUSIONS

A new compaction algorithm for color picture coding has been described. Replacing the JPEG binary transformation with the proposed transformation, the picture compaction algorithm based on the arithmetic coding has achieved both higher compaction ability and implementation simplicity. The new method shows very satisfying results which are better than the existing JPEG algorithm. Moreover, the new algorithm achieves implementation simplicity, because the binary transformation process is based on fixed length coding.

5. REFERENCES

[1] Huffman, D.A. "A method for the construction of minimum-redundancy codes," Proc. Inst. Electr. Radio Eng. 40,9 (September 1952)

[2] Langdon, G.G. "An Introduction to Arithmetic Coding," IBM J. R&D, Vol.28,No. 2, pp135-149, (March 1984)

[3] Pennebarker, Mitchell, Langdon and Arps, "An overview of the basic principle of the Q-Coder Binary Arithmetic Coder", IBM J. R&D, Vol.32, pp717-726 (1988)

[4] J.Rissanen,A. and Mohiuddin,K.M. "A Multiplication-Free Multialphabet Arithmetic Code," IEEE Trans. on Communication, COM-37, No.2 , pp.93-pp.98,(February 1989)

[5] Glen, Langdon and Rissanen, "Compression of Black White Imaes with Arithmetic Coding," IEEE Trans. on Communication, COM-29, No.6, pp858-867 (June 1981)

[6] Witten,I.H., Neal,R.M. and Cleary,J.G. "Arithmetic Coder for Data Compression," Communication of the ACM, No.6, Vol.30, (June 1987)

[7] Ramabadran and Cohn, "An Adaptive Algorithm for Compression of Computer Data," IEEE Trans. on Communication, COM-37, No.4, pp317-324 (April 1989)

[8] Schwartz,J.W. and Barker,R.C "Bit-plane Encoding: A technique for source encoding," IEEE Trans. Aerosp. & Electron. Syst., AES-2, 4, pp.385-392 (July 1966)

[9] Spencer, D.R. and Hung,T. "Bit-plane Encoding of Continuous-tone Pictures," Symposium on Computer Processing in Communications, Polytechnic Institute of Brooklyn, New York (1969)

[10] Gall,D.L. and Tabatabai,A. "Subband Coding of Digital Images Using Symmetric Short Kernel Filters and Arithmetic Coding Techniques," Proc. ICASSP 88, pp.761-pp.764, (1988)

[11] "Draft (Revision 5) of the JPEG algorithm," JPEG, JPEG-8-R5 (December 15, 1989)

[12] Elias,P. and Abramson,N. "Information Theory and Coding," McGraw-Hill Book Co., Inc., New York, (1963)