A Preprocessing Algorithm for Efficient Lossless Compression of Gray Scale Images

Sun-Ja Kim*, Doh-Yeun Hwang **, Gi-Hyoung Yoo**,
Kang-Soo You*** and Hoon-Sung Kwak**

* Department of Industrial Technology, Chonbuk University, Jeonju, Chonbuk, Korea
(Tel : +82-63-270-2417; E-mail: rich816@lycos.co.kr)
**Department of Computer Engineering, Chonbuk University, Jeonju, Chonbuk, Korea
(Tel : +82-63-270-2417; E-mail: ehdus, ghyoo, hskwak@chonbuk.ac.kr)
***Department of Image Engineering, Chonbuk University, Jeonju, Chonbuk, Korea
(Tel : +82-63-270-2417; E-mail: mickey@chonbuk.ac.kr)

Abstract: This paper introduces a new preprocessing scheme to replace original data of gray scale images with particular ordered data so that performance of lossless compression can be improved more efficiently. As a kind of preprocessing technique to maximize performance of entropy encoder, the proposed method converts the input image data into more compressible form. Before encoding a stream of the input image, the proposed preprocessor counts co-occurrence frequencies for neighboring pixel pairs. Then, it replaces each pair of adjacent gray values with particular ordered numbers based on the investigated co-occurrence frequencies. When compressing ordered image using entropy encoder, we can expect to raise compression rate more highly because of enhanced statistical feature of the input image. In this paper, we show that lossless compression rate increased by up to 37.85% when comparing results from compressing preprocessed and non-preprocessed image data using entropy encoder such as Huffman, Arithmetic encoder.

Keywords: lossless Compression, Entropy Coding, Gray Scale Image, Data Redundancy

1. INTRODUCTION

An enormous amount of data is produced when a 2-D light intensity function is sampled and quantized to create a digital image. In fact, the amount of data generated may be so great that it results in impractical storage, processing and communications requirements. For instance, more than 25 gigabytes (25×109 bytes) of data are required to represent the Encyclopedia Britannica in digital form. Because digital images are usually in need of more storage than common data, a number of researches to compress image data efficiently are in progress from around 30 years ago. Image compression addresses the problem of reducing the amount of data required to represent a digital image, removing redundancy of data.

Over the years, the need for image compression has grown steadily and it is necessary to compress multimedia data(that is, the use of digital computers in printing, publishing, video production and dissemination) efficiently. In addition, image compression plays a crucial role in many important and diverse applications, including televideo-conference, remote sensing(the use of satellite imagery for weather and other earth-resource applications), document and medical imaging, facsimile transmission(FAX), and the control of remotely piloted vehicles in military, space, and hazardous waste control applications.

Especially, in the fields of medical imaging, satellite imagery, highly precious image analysis and preservation of the works of artistic value, the lossless compression techniques is necessary to preserve any original data when coding and decoding a digital image. Some typical algorithms such as Run-Length coding, LZW coding, Huffman coding and Arithmetic coding are well-known for lossless compression technique. JPEG-LS, which is standard for lossless and near-lossless compression of still images, is a famous lossless image compression skill in company with CALIC, UCM, and so on [1-2].

Among these lossless techniques, the entropy coding methods such as Huffman and Arithmetic coding are based on the statistical feature of pixel values. They assign a few bits for frequently detected data, otherwise, more bits for rarely occurring data in a image.[2] Because image data is inherently multi-dimensional, however, compression has to be performed under consideration of the spatial feature. Contrary to JPEL-LS which compresses an image using predictive technique, any entropy coders that treat image data as a string of symbols often fail to highly compress natural digital images that have continuous-tone [3-4].

This paper introduces a newly designated scheme rearranging pixel values so that natural digital images can be compressed more efficiently by the entropy coders. The proposed method is a kind of preprocessing technique to replace a original image with a particular ordered image whose form is more compressible. Our goal is to enhance statistical characteristic of gray scale images through proposed preprocessing steps.

Experimental results suggest that the proposed scheme can reduce bit rate by up to 37.85% when compared to a plain Huffman and Arithmetic coding. Furthermore, bit rate can also reduce a little when even transforming a gray scale image into GIF. This is because the ordered image from the proposed scheme has more redundancy of data than the original image. And spatial distribution of the ordered image is so smooth that image data can be compressed more efficiently in spite of additional information. From this simulation results, we know that the preprocessing technique which is proposed for efficient entropy coding can improve performance of lossless compression more effectively.

The rest of the paper is organized as follows. In section 2 we give a brief overview of Huffman coding and Arithmetic coding which are well-known for lossless compression algorithm. In section 3 we show how to rearrange the array of pixel values of original image and how conversion of original pixel values to ranked values can lead to increase compression rate. Then, we present some result from our simulation using Huffman coding and Arithmetic coding to verify proposed method in section 4. Finally, we summarize our results in section 5 and conclude with pointers to further research.

2. AN OVERVIEW OF ENTROPY CODING

In coding (compressing) a set of any symbols, it is well known that certain parts of the input stream appear more frequently than others, so it would be more economic to assign fewer bits as their codes. This is commonly called as not only variable length coding but also entropy coding. As we know, Arithmetic and Huffman algorithms belong to this compression method. Under these entropy coding methods, the more frequently appearing symbols are coded with fewer bits per symbol, and vice versa [2].

When encoding a sequence of symbol, fixed length encoder assigns same number of bits for each symbol. In contrast with fixed length encoding, the variable length encoding system assigns a different number of bits each symbol. It uses the statistical characteristic of input data. The higher probability of occurred symbol, the less bits must be allotted to encode the data more efficiently. Under this variable length coding, compression ratio is usually higher than fixed length coding.

For example, consider an arbitrary sequence of data consisting of 0, 1, 2 and 3. The fixed length encoder may assign 2 bits for each symbol. That is, the encoded code words are 00, 01, 10, 11 correspond to 0, 1, 2, 3, respectively. Table 1 shows the appearance probabilities of each symbol and it also presents the various code words for each symbol assigned by entropy encoder (variable length encoder). In this case, we supposed to use Huffman encoder.

Table 1 An example of entropy coding.

Symbol	Probability	Code Word	Length
0	0.6	0	1
1	0.2	10	2
2	0.1	110	3
3	0.1	111	3

It is natural that the average of bit cost is 2 bits per symbol in fixed length encoding. Under the entropy encoding used in Table 1, however, the average of bit cost is 1.6 bits per symbol. It is the result of $(1\times0.6)+(2\times0.2)+(3\times0.1)+(3\times0.1) = 1.6$ bits per symbol. Therefore, entropy coding is more profitable than fixed length encoding when compressing data composed of above symbols. Following Arithmetic and Huffman coding Algorithm are usually operated effectively for the general data composed of simple characters.

Arithmetic encoder assigns a single code word for an input sequence. In the first step of Arithmetic encoding, the encoder calculates the appearance probabilities of each symbol, and particular intervals from 0 to 1 are partitioned correspond to every symbol presented. In the second step, the intervals become step by step narrowed into subinterval based on probabilities of the input symbols. Upon finishing this recursive sub-partition, the start real number of final subinterval is code word for the input sequence [5].

Huffman encoding also has 2 steps, but it manipulates input stream differently. Huffman encoder counts occurrence frequencies of each symbol in an input sequence. And then, the encoder creates binary tree which has leaf nodes corresponding every presented symbol based on the counted frequencies. The encoder assigns code words to each symbol. The assigned code words have different length. Finally, it encodes the symbols in the input stream separately, sequentially [6].

These entropy encoding methods are also used to compress digital images. This novel technique is properly operated for the general data like a English text, but it often fails to highly compress natural digital images which have continuous-tone. They regard multi-dimensional image data as a sequence of pixel values. Furthermore, quantized images generally have large scale of intensity, so they have more various symbols than common data. These features occasionally cause inefficient compression ratio when applying entropy coding for image data. Because, that is, image data is inherently multi-dimensional and has many different pixel values, the compression performance of entropy encoder is usually lower than others in contrast to JPEG-LS using predictive scheme [3-4].

From this point of view, we introduce a preprocessing technique replacing original image data with particular ordered data so that natural gray scale images can be compressed more efficiently by any entropy encoders. We deal with the efficient way to successfully increase compression ratio for gray scale images. The proposed method enhances the statistical characteristic of input data because it uses much more same values to represent digital images.

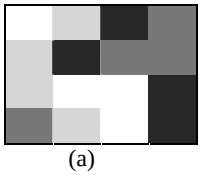
3. THE PROPOSED PREPROCESSING SCHEME

In this paper, we propose the method preprocessing an input gray scale image in order to improve performance of lossless compression by entropy coding. To transform an input image map into the more compressible form, our method starts with ***CA*** (Counted Array) which has same sized rows and columns as gray scale. It consists of co-occurrence frequencies for the pair of pixels appeared in the sequence of an input image.

Next, another same sized ***OA*** (Ordered Array) is generated by sorting each row of ***CA***, having the ranked number. Finally, each pixel value of the original image is replaced by the ordered values corresponding to the values of ***OA***. That is, the preprocessor replaces original neighboring two gray values with particular ordered number based on a set of counted occurrence frequencies. Then, it transmits the transformed ordered image data and additional information, ***CA*** to entropy encoder.

After preprocessing, the ordered image can be compressed more efficiently by any entropy encoders. However, ***CA*** has to be transmitted to the receivable entity with encoded data to convert the ordered image into the original image with any loss. Despite this additional information, compression rate is much higher than plain entropy methods encoding images without pre-process. Our goal, using this preprocessing technique, is to enhance statistical feature of natural digital images which have continuous-tone. To increase data redundancy, this preprocessing technique operates as following simple and easy steps.

If n bits per pixel are used to represent arbitrary gray scale image, as we know, G (the gray scaled input image) is 2^n. For instance, if 2 bits per pixel were used for a gray scale image, G is $2^2=4$. Fig. 1 shows an example of 2 bits gray scale image using following specification; gray scale G is 4 (from 0 to 3), size (row×column) is 4×4.



(a)

3	2	0	1
2	0	1	1
2	3	3	0
1	2	3	0

(b)

Fig. 1 An example of 4×4 sample image with 4 gray scale. (a) shows gray scale image; (b) shows an each pixel value.

The proposed technique functions as preprocess for the gray scale images that have this kind of image map to be compressed more efficiently. In the first step, the preprocessor transforms a two-dimensional input image into a sequence **S** which has the lowest gray value as the first entry. If the lowest gray value is zero in an input image sized M×N, the first entry S_0 is 0 in the sequence **S** = (S_0, S_1, S_2, … , $S_{M\times N)}$ and the next elements can get scanning the gray values from top-left to bottom-right on the image map. Therefore, the sequence for the image of Fig. 1 has 0 as the first entry and includes the gray values appeared in image as next elements. That is, **S** = (0, 3, 2, 0, 1, 2, 0, 1, 1, 2, 3, 3, 0, 1, 2, 3, 0).

Next, the preprocessor counts co-occurrence frequencies of adjacent two pixel values in the sequence **S**. And then, investigated co-occurrence frequencies is stored in ***CA*** sized G×G. As stated above, G is gray scale of the image. The slots of G_i-row and G_j-column in the created ***CA*** have the co-occurrence frequencies correspond to the pairs of S_i and S_{i+1} appeared in sequence **S**. For example, ***CA*** obtained from sequence **S** of the image in Fig. 1 is same as (a) of Fig. 2.

Now, in the third step, ***OA*** sized G×G is created and the ***OA*** has the ordered number as a result of sorting each row of ***CA*** with descending order. That is, the ***OA*** is just led by ordering each row of ***CA***. When any same values are detected in a row, the number of methods may be considered to grant priority. In this paper, we choose the method that previous values have more high priority simply. Through this introduced method, (b) of Fig. 2 is calculated from (a) of Fig. 2. In the second row of (a), 2 is the highest occurrence frequency, so the ordered number is 1 in (b). On the contrary, 0 is the lowest occurrence frequency, so the values are 4 in (b). According to selected priority rule, 1 of third slot means 3, 1 of final slot means 4 respectively.

G_i \ G_j	0	1	2	3
0	0	3	0	1
1	0	1	3	0
2	2	0	0	2
3	2	0	1	1

G_i \ G_j	0	1	2	3
0	3	1	4	2
1	3	2	1	4
2	1	3	4	2
3	1	4	2	3

Fig. 2 The relative arrays of the sample input image in Fig. 1.
(a) shows the ***CA***; (b) shows the ***OA***.

Finally, the preprocessor creates a new sequence **S´** using the entry of ***OA*** corresponding each pair of neighboring gray values appeared in one dimensional sequence **S**. The entries of **S´** are the ordered numbers of G_i-row and G_j-column of the created ***OA*** correspond to the pairs of S_i and S_{i+1} appeared in sequence **S**. **S´** can be transformed to two-dimensional image map sized M×N due to having entries same as original number of pixels.

Fig. 3 (a) shows the result data which will be submitted to entropy encoder and (b) is additional information, ***CA*** to restore the transformed data. Because of the high data redundancy of this transformed image, we can expect that the compression performance will be improved when using entropy encoder. Although the amount of ***CA*** may be problem when encoding a small sized image, the proposed preprocessing technique operates well for general sized gray scale images.

See the Fig. 3 (a), there are much more same value than an original image data. This is because the pair of neighboring pixel value, which has the same appearance probability, can be represented same ordered number. It means enhancing the statistical feature of input image so that the image data can be more compressible form by entropy encoder. And we can also reduce bits required for ***CA*** in Fig. 3 (b). The preprocessor transmit only non-zero frequencies using data structure such as sparse matrix. This is because there are generally too many zero frequencies and we can ignore them.

2	2	1	1
1	1	1	2
1	2	3	1
1	1	2	1

(a)

0	3	0	1
0	1	3	0
2	0	0	2
2	0	1	1

(b)

Fig. 3 Preprocessed data of Fig. 1.
(a) shows ordered data ***OA***
(b) shows additional information ***CA***.

Fig. 4 is representing the proposed compression system. This system creates ***CA*** and ***OA*** based on statistical feature of input gray values appeared in **S**. And then, it replaces an original image data S with transformed image data **S´** using the values of ***OA***. **S´** is finally compressed by entropy encoder. The compressed stream includes ***CA*** in order to restore transformed image data. The number of used gray scale G can be different in every image, so the size of ***CA*** depends on the number of gray scale of input image. Sometimes, the size of ***CA*** may be inefficient depending on the image size and gray scale, especially when the original image is small.

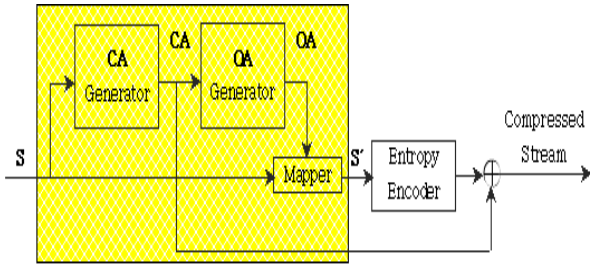


Fig. 4 The proposed encoding system

In Fig. 5, the preprocessor makes ***OA*** based on received ***CA*** and entropy coder decode the compressed sequence of ordered image data. Then, It restores decoded sequence **S´** into the original gray scale image **S**. It can be realized by reversing the preprocessing on encoder. In spite of this additional data, in comparison with compressing gray scale images without preprocess, we know that the compression rate can be increased by the proposed method. In this scheme, the restored image has the same resolution and size as the original gray scale image due to the lossless nature of compression.

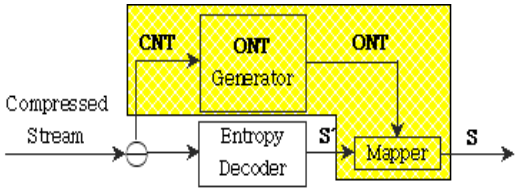


Fig. 5 The proposed decoding system.

4. PERFORMANCE EVALUATION

In order to evaluate the real effectiveness of the proposed method, we have tested with 8 bit gray scale images. To compare the performance of preprocessing, we use two entropy coding method; one is Arithmetic coding and another is Huffman coding. The tested gray scale images have various size from 512×512 to 2048×2560. First, we encode the images using Arithmetic and Huffman encoder without preprocess. And then, we also encode the image using above encoder after proposed preprocessing.

For the purpose of comparison between two kinds of compression results, we use the value of bits per pixel (bpp). That is, bpp is the measure of performance evaluation. As we well know, this bpp indicates multiply 8 by the reciprocal compression ratio (CR). The CR is obtained dividing original image size into compressed image size. To make a long story short, bpp = 8/CR. The bpp values are calculated including the size of CNT in only preprocess algorithm.

Fig. 6 displays "boy" sized 768×512. The original image is shown in (a) of Fig. 6, and (b) of Fig. 6 shows the preprocessed image obtained from the proposed algorithm on the original image. The image that has ordered numbers appears to be much smoother than the original image. This is expected to greatly facilitate the subsequent entropy coding.

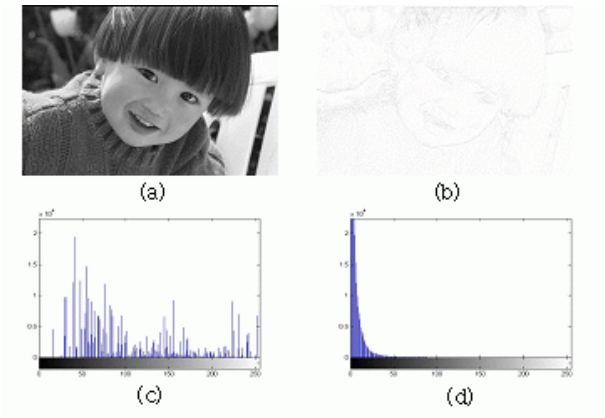


Fig. 6 An example of tested image - "boy". (a) An original image; (b) The preprocessed image; (c) A histogram of the original image; (d) A histogram of preprocessed image.

The histogram distribution of the intensities of "boy" is shown in Fig. 6 (c) and Fig. 6 (b) shows the histogram of the distribution of ordered numbers for the preprocessed "boy". As can be seen Fig. 6 (b), the histogram is condensed to lower ordered number, therefore this makes that we can expect the higher efficiency in the compression ratio. Similar improvements are observed with other images.

Table 2 includes the original input gray scale images. Tested images have 8 bit gray scale and have various size. Table 3 shows the simulation results using Huffman encoding, in terms of bpp about two different methods, the plain encoding and proposed technique respectively. Table 4 shows the simulation results by Arithmetic encoding as well. As Table 3 and Table 4 demonstrate, for 8 bits gray scale images, the proposed preprocessing technique reduces the bit rates when compared with the another plain encoding schemes.

As we knew from histograms of Fig. 6, simulation results show that bits cost is reduced effectively when encoding preprocessed images in Table 2. This simulation results using 8 bit gray scale images show that the proposed method can reduce bit rate by up to 37.85%. From this simulation results, we know that the preprocessing technique for efficient lossless image compression can improve compression performance in the entropy coders. Fig. 7 illustrates the graph of bpp comparison from Table 3 and Table 4.

Table 2 Tested images by proposed scheme.

images	width	height	size
airplane	512	512	262,144
anemone	722	471	340,062
arial	735	493	362,355
bike3	781	919	717,739
boat	512	512	262,144
boy	768	512	393,216
café	2048	2560	5,242,880
goldhill	720	576	414,720
lena	512	512	262,144
monarch	768	512	393,216
tulips	768	512	393,216
woman	2048	2560	5,242,880

Table 3 Simulation results using Huffman coding.

images	Huffman(bpp)		saved bits(%)
	plain	preprocessed	
airplane	6.49	5.25	19.18
anemone	7.30	5.75	21.27
arial	7.36	6.88	6.55
bike3	6.89	4.81	30.19
boat	7.06	6.48	8.21
boy	6.43	4.08	36.61
café	7.59	5.88	22.63
goldhill	7.51	5.50	26.82
lena	7.14	5.30	25.75
monarch	6.83	4.57	33.14
tulips	7.19	4.50	37.49
woman	7.29	4.93	32.32

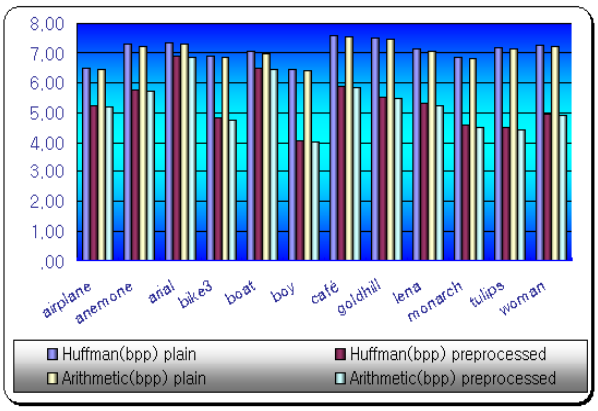


Fig. 7 A graph comparing each bpp from Table 3 and 4.

Table 4 Simulation results using Arithmetic coding

images	Arithmetic(bpp)		saved bits(%)
	plain	preprocessed	
airplane	6.44	5.19	19.39
anemone	7.25	5.70	21.36
arial	7.31	6.83	6.57
bike3	6.85	4.76	30.56
boat	7.00	6.41	8.36
boy	6.38	4.03	36.82
café	7.56	5.85	22.65
goldhill	7.47	5.45	27.10
lena	7.08	5.24	25.99
monarch	6.78	4.50	33.54
tulips	7.14	4.44	37.85
woman	7.25	4.91	32.32

5. CONCLUSION

In this paper, we introduced a preprocessing technique in order to improve the performance of entropy coding for gray scale images. The proposed method transformed an original image map into more compressible form based on occurrence frequencies of adjacent two gray scale values. The preprocessed data is compressed more efficiently by entropy encoder because data redundancy is increasing and spatial distribution of replaced image data is concentrated.

However, the proposed method had disadvantage that the additional data depending on gray scale of input images should be submitted to the receiver. After performance evaluation with 8 bits gray scale images, we confirmed that bit rate was reduced up to 37.85% by Arithmetic and Huffman encoder as compared with entropy encoding results which didn't use the proposed preprocessing method. As stated above, these evaluated values included additional data, ***CA***.

The proposed preprocessing method is easy to implement because of simplicity of algorithm and has practical time complexity. Furthermore, the amount of data that will be transmitted can be reduced considerably due to improved compression performance. Therefore, this method can be applied in various fields of not only medical imagery that is necessary to preserve any original data but also the application that needs to transmit digital images as fast as possible.

When transforming and restoring a input image, the improved method operating without additional information may be considered. It can reduce amount of data and compression rate will increase more highly, especially in small images.

REFERENCES

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd Ed. Prentice Hall, 2002.

[2] Khalid Sayhood, *Introduction to Data Compression*, Morgan Kaufmann, 2000.

[3] S. D. Rane and G. Sapiro, "Evaluation of JPEG-LS, the New Lossless and Controlled-Lossy Still Image Compression Standard, for Compression of High-Resolution Elevation Data", *Geoscience and Remote Sensing, IEEE Transactions on*, Vol. 39, No. 10, 2001.

[4] N. Memon and R. Rodila, "Transcoding GIF images to JPEG-LS", *Consumer Electronics, IEEE Transactions on*, Vol. 43, Issue. 3, pp. 423-429, 1997.

[5] M. Rabbani and R. Joshi, "An overview of the JPEG 2000 still image compression standard", *Signal Processing : Image Communication*, Vol. 17, pp. 3-48, No. 1, 2002.

[6] R. Hashemian, "Direct Huffman coding and decoding using the table of code-lengths", Information Technology: Coding and Computing [Computers and Communications]. Proceedings. ITCC 2003. International Conference on, pp. 237 – 241, 2003.