그레이레벨 이미지에서의 엔트로피 코딩 성능 향상을 위한 순위 기법

유강수* · 심춘보**

A Ranking Method for Improving Performance of Entropy Coding in Gray-Level Images

Kang Soo You* · Chun-Bo Sim*

요 약

본 논문에서는 엔트로피 부호화기를 통해 그레이레벨 이미지에서의 효율적인 압축 알고리즘을 제안한다. 제안하는 기법의 핵심은 원래의 그레이레벨 이미지 정보를 특정 순위 정보로 변환하는 것이다. 이를 위해 먼저, 그레이레벨 값을 가지는 정보를 부호화하기 전에 이웃하는 주변 픽셀(그레이레벨) 값들에 대해서 상호 발생 빈도수를 계산한다. 그런 후, 이미 계산된 상호 발생 빈도수에 따른 특정 순위를 각 그레이레벨 값에 적용한다. 마지막으로, 엔트로피 부호화기를 통해 순위 정보를 전송하여 압축을 수행한다. 제안하는 기법은, 영상의 통계적 발생 빈도에 따른 정보를 토대로, 그레이레벨 이미지를 순위 영상으로 변환함으로써 기존의 엔트로피 코딩 기법의 성능을 향상시킨다. 시뮬레이션 결과 8비트의 그레이레벨 이미지에 대해서 제안하는 기법이 기존의 엔트로피 부호화기에 비해 최대 37.85%까지 압축 성능을 더 향상시킴을 알 수 있었다.

ABSTRACT

This paper proposes an algorithm for efficient compression gray-level images by entropy encoder. The issue of the proposed method is to replace original data of gray-level images with particular ranked data. For this, first, before encoding a stream of gray-level values in an image, the proposed method counts co-occurrence frequencies for neighboring pixel values. Then, it replaces each gray value with particularly ranked numbers based on the investigated co-occurrence frequencies. Finally, the ranked numbers are transmitted to an entropy encoder. The proposed method improves the performance of existing entropy coding by transforming original gray-level values into rank based images using statistical co-occurrence frequencies of gray-level images. The simulation results, using gray-level images with 8-bits, show that the proposed method can reduce bit rate by up to 37.85% compared to existing conventional entropy coders.

키워드

Gray-Level Images, Ranking Method, Entropy Coding

* 전주대학교 교양학부
** 순천대학교 정보통신공학부, 교신저자

접수일자 : 2007. 12. 10

Ⅰ. Introduction

As telecommunication technologies and digital equipments develop, nowadays, an area the researcher is studying on transmitting and storing image using lossless compression. The amount of data, in fact, may be so big that it results in impractical storage and communication requirements. There are many coding techniques that will effectively reduce the total number of bits needed to represent the data. This accompanying process is generally referred to as compression. An image compression addresses the problem of reducing data information to represent a digital image with removing redundancy of data [1][2].

In actual fact, some of image data such as medical images and precious paintings need to be stored without any loss of information. For the purpose of compressing digital images losslessly and more efficiently, several popular lossless image coding algorithms such as LZW, Arithmetic and Huffman coding have been studied for a long time. These algorithms are called entropy coding [3][4].

These novel entropy coding algorithms are properly operated for textual data like English texts, but they often fail to highly compress natural digital image which have continuos-tone. Because 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 with JPEG-LS using predictive scheme [5][6]. From this point of view, we propose an efficient method to increase compression ratio for gray-level images.

This paper starts out with an introduction to a scheme converting the array of pixel values into a more compressible form so that a large data of digital image can be compressed more effectively by the entropy encoders. The proposed method in this paper is a kind of pre-processing technique to replace each pair of neighboring pixels of an original image by particularly ordered numbers, that is, ranks. The goal of this paper is to enhance statistical characteristic of data redundancy in gray-level images. In the proposed method, however, a little additional data must be transmitted to the decoding entity so that the ranking image data can be reconstructed to the former image.

In spite of required additional information, experimental results show that the proposed scheme can reduce bit rate by up to 37.85% compared to the conventional LZW, Huffman and Arithmetic coding. This is because the ranked image from the proposed scheme has more redundancy of data than the original image. And spatial distribution of the ordered image is so skewed that image data can be compressed more efficiently.

The paper is organized as follows. Section 2 introduces the existing entropy coding method and section 3 the proposed ranking method. The simulation results are given in section 4 and finally section 5 concludes.

Ⅱ. Entropy in Image

The entropy is term which represents code bit of a input or source symbols and means how input symbols have a partiality. The process of entropy coding can be split in two parts. There are modeling and coding. Modeling assigns probabilities to the symbols, and coding produces a bit sequence from these probabilities. As established in Shannon's source coding theorem, there is a relationship between the symbol probabilities and the corresponding bit sequence. A symbol with probability P gets a bit sequence of length $-\log_b P$. In order to achieve a good compression rate, an accurate estimation of probability is needed [7][8]. Equation (1) is a formula for *Entropy*.

$$Entropy = -\sum_{n} P_n \log_2 P_n \tag{1}$$

Since the model is responsible for the probability of each symbol, modeling is one of the most important tasks in data compression. The entropy coder encodes a given set of symbols with the minimum number of bits required to represent them. As we know, two of the most popular entropy coding schemes are Huffman coding and Arithmetic coding. Lempel-Ziv coding is also an entropy coding scheme. With these entropy coding methods, more frequently appearing symbols are coded with fewer bits per symbol [2][9].

When encoding a sequence of symbols, a fixed length encoder assigns each symbol to the same number of bits, whereas a fixed length coder assigns different number of bits to source symbols. Using the statistical characteristic of input data, the variable length encoder assigns shorter codes to more frequent symbols, usually resulting in higher compression ratio than the fixed length coding. So this efficient variable length encoding method is also called entropy coding.

On the assumption that an arbitrary sequence of data consist 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 and 11 correspond to 0, 1, 2 and 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.

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

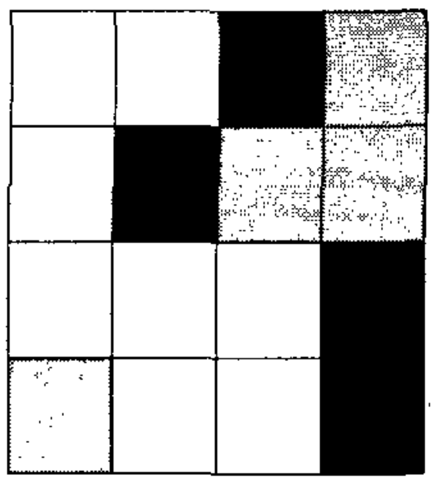
Entropy encoding methods can also be used to compress digital images. They treats multi-dimensional image data as a sequence of pixel values. Furthermore, quantized images generally have a large scale of intensity, so they have more various symbols than common data. These features occasionally cause inefficient compression ratio when applying entropy coding for image data.

From this point of view, we introduce a ranking technique replacing original image data with particularly ordered data so that natural gray-level images can be compressed more efficiently by any entropy encoder. The proposed method enhances the statistical characteristic of input data because it uses much more same values to represent digital images.

Ⅲ. The Proposed Ranking Method

In this paper, we propose the ranking method based on a gray-level image to improve compression performance of lossless image by entropy coding. In order to replace an input image map into a more compressible form, the proposed method starts with counting co-occurrence frequencies of adjacent pixel values in a sequence of the image. Subsequently it replaces original gray-levels with particularly ordered numbers, based on a set of counted co-occurrence frequencies. Lastly, it transmits the transformed ordering data, i.e. ranks and additional information that indicates co-occurrence frequencies to the entropy encoder.

Using these simple steps, we have an intention of enhancing statistical feature of natural digital images which have intensity based gray-levels. The proposed ranking technique performs the following simple and easy steps for increasing data redundancy. If n bits per pixel are used to represent an arbitrary gray-level image, the image's gray-level G is 2^n. For instance, if a gray-level image uses 2 bits per pixel, then gray-level G is $2^2=4$. Figure 1 shows an example of 2 bits gray-level image sized 4×4 using 4 gray-levels; they are '0', '1', '2' and '3'.



(a) Original

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

(b) Gray-level

Fig. 1 Gray-level image with 2 bits

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

(a) CNT

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

(b) ONT

Fig. 2 The CNT and ONT tables of Fig. 1

3.1 Encoding Part

As mentioned above, the proposed method as a ranking for the gray-level images enables these kinds of map to be compressed more efficiently. In the first step, the ranker transforms a two dimensional input image into a sequence S which has the lowest gray-level as the first entry. If the lowest gray-level is zero in an input image sized $M \times N$, the first entry S_0 in '0' in the sequence $S=(S_0, S_1, S_2, ..., S_{M \times N})$ and the next elements can get by scanning the gray-level from the top-left corner to the bottom-right corner on the image map. Therefore the sequence for the image of Figure 1 has '0' as the first entry and includes the gray-level values appeared an image as neighboring elements in Figure 1. The sequence S is *(0, 0, 3, 0, 3, 2, 3, 2, 3, 3, 1, 1, 1, 1, 2, 0, 3)*.

In the second step, the ranker counts the frequencies of diagrams in the sequence S, which we will call co-occurrence frequencies form now on. And then, the co-occurrence frequencies are stored in the counted number table (CNT) sized $G \times G$. As stated above, G is the gray-level of the image. The slots of G_i-row and G_j-column in the created CNT have the co-occurrence frequencies corresponding to the pairs of S_i and S_{i+1} appearing in sequence S. The counted CNT from the sequence S of the image in Figure 1 is the same as that in Figure 2(a) as an example.

Now, in the third step, the ordered number table (ONT) sized $G \times G$ is created. The ONT has the ordered number when sorting each rows of CNT in descending order. That is, the ONT is simple obtained by ordering each rows of CNT. When any same value is detected in a row, the ordered numbering may be considered to grant priority. In this paper, we choose the ordered numbering where a value has a higher priority than any numbers to its right in the same row. Therefore, a weight function w(j) can be defined as equation (2) and applied to determine priority by equation (3).

$$w_i(j) = (255 - |i-j|)/256 \tag{2}$$

$$\widetilde{ONT_i}(j) = ONT_i(j) + w_i(j) \tag{3}$$

Through this method, Figure 2(b) is obtained from Figure 2(a). In the first row of CNT in Figure 2(a). the 2nd element '3' is the highest occurrence frequency, so the ordered number is '1' in Figure 2(b). On the other hand, '0' is the lowest co-occurrence frequency, so the ordered numbers are '3' and '4' as shown in Figure 2(b).

Lastly, the ranker generates a new sequence (i.e. ranked image) S' referencing the entry of ONT corresponding to each pairs of neighboring gray-levels which appear in one dimensional sequence S. The entries of S' are the ordered numbers of G_i-row and G_j-column of the generated ONT corresponding to the pairs of S_i and S_{i+1} in the sequence S. Figure 3 shows the result data which will be submitted to the entropy encoder and Figure 2(b) is additional information (ONT) to restore the transformed data. So S' can be transformed to two dimensional image map sized $M \times N$ due to having entries same as original number of pixels. Because of the data redundancy of this transformed ranked image, we can expect that the compression performance will be improved when using an entropy encoder. Although we have a problem with CNT when encoding a small sized image, the proposed ranking method operates well for general sized gray-level images.

As can be seen in Figure 3, there are more identical values than the case for the original image data. The reason why a pair of neighboring pixel values, which have the same appearance probability, can be represented as the same rank. It means to enhance the statistical feature of input image so that the image data can be more compressible form by the entropy encoder. In addition, we can also reduce bits required for ONT in Figure 2(b). The ranker transmits 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

Fig. 3 The Result of Ranked Image

The ranked image data is compressed using an Arithmetic encoding. The reason that Arithmetic encoding is chosen is because it shows a better compression ratio than Huffman coding or Arithmetic coding when the symbols (i.e., ranks) to compress have skewed distributions or small variance.

The ranked image is achieved from following algorithm;

```
input file as [a, Map] type
    a = double(a); sa = size(a);
    saa = sa(1)*sa(2); % 1-D original image
cm(a(1)+1,1) = 1;       % count horizontal
                          changed value
    for k=2:saa
        cm(a(k)+1,a(k-1)+1) =
                        cm(a(k)+1,a(k-1)+1) + 1;
    end
[null,cr] = sort(cm);   % make a rank
for k=1:256              % make CNT
    [ia,ib] = find(cr == k);
    CNT(k,:) = ia';
        end
b = zeros(1,saa);        % generate ranked image
    for k=2:saa
        b(k) = CCT(a(k)+1,a(k-1)+1);
    end
d = arithenco(b,cnt);    % arithmetic encoding
                              of ranked image
```

3.2 Decoding Part

Compressed images are perfectly reconstructed in reverse order of encoding steps. At first, the ranked image data is uncompressed from entropy decoding. To reconstruct original image from the ranked image, and then, our decoder generates ONT from CNT and references ONT with ranks.

The encoded data is reconstructed from following algorithm;

```
bhat = arithdeco(d, CNT, saa);      % arithmetic
                                       decoding
ahat = zeros(1,saa);          % reconstructing
                                original image
ahat(1) = cr(bhat(1),1);
    for k=2:saa
        ahat(k) = cr(bhat(k),ahat(k-1));
    end
ahat = unit8(reshape(ahat, sa(1), sa(2))-1);
```

Ⅳ. Simulation Results

Some of images with various sizes (512×512, 720×576, 722×471, 768×512, 2048×2560) have been used for simulation. Of course, the tested images have 8 bits gray-levels. We compared the performance of the proposed method with entropy coding methods such as LZW, Huffman and Arithmetic encoding.

At first, we encode the image using LZW, Huffman and Arithmetic encoders. Subsequently we encode the image using above encoder after the proposed ranking method. For the purposed of comparison among three kinds of compression results, we use the value of bits per pixel (bpp). The bpp values are calculated including uncompressed addition information. Since the calculation of bpp for the proposed method needs to consider the size of compressed image, addition information and ONT, this simulation takes into account the size of ONT to store or transmit.

Table 2, Table 3 and Table 4 show the simulation results, in terms of byte about three different methods, the conventional encoding and proposed methods, respectively. As tables demonstrates, the proposed method shows an average compression gain of 9.71%, 29.26% and 29.47% for LZW, Huffman and Arithmetic encoding. For 8 bits gray-level images, the proposed method has lower bpp and better bit rates are achieved with the proposed method than with other conventional encoding methods.

Figure 4(a), Figure 4(b) and Figure 4(c) describe original images, "lena" sized 512×512, "tulips" sized 768×512 and "peppers" sized 2048×2560. Ranked images are obtained from the original images as in Figure 5(a), Figure 5(b) and Figure 5(c) using the proposed method, respectively. The ranked image appears to be much smoother than the original image. This is expected to greatly facilitate the subsequent entropy coding.

The histogram distribution of the intensity per gray-levels in "lena", "tulips" and "woman" are shown in Figure 6(a), Figure 6(b) and Figure 6(c). And the histogram of the distribution of rank which is transformed by the proposed method is shown in Figure 7(a), Figure 7(b) and Figure 7(c).

Table. 2 Simulation results in LZW encoding with ranked images

Images	LZW (bpp)		saved bits (%)
	conven-tional	the proposed	
anemone	7.09	6.91	2.51
bike	6.71	5.77	13.96
boy	5.09	4.89	3.92
cafe	8.32	7.32	12.01
goldhill	7.61	6.71	11.85
lena	6.94	6.34	8.71
monarch	5.75	5.36	6.82
peppers	6.78	6.54	3.48
tulips	6.44	5.37	16.59
woman	7.33	6.07	17.27
average	6.80	6.13	9.71

Table. 3 Simulation results in Huffman encoding with ranked images

Images	Huffman (bpp)		saved bits (%)
	conven-tional	the proposed	
anemone	7.30	5.75	21.27
bike	7.06	4.86	31.17
boy	6.43	4.08	36.61
cafe	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
peppers	7.26	5.42	25.37
tulips	7.19	4.50	37.49
woman	7.29	4.93	32.32
average	7.16	5.08	29.26

(a) lena (b) tulips (c) woman

Fig. 4 Gray-level images

Table. 4 Simulation results in Arithmetic encoding with ranked images

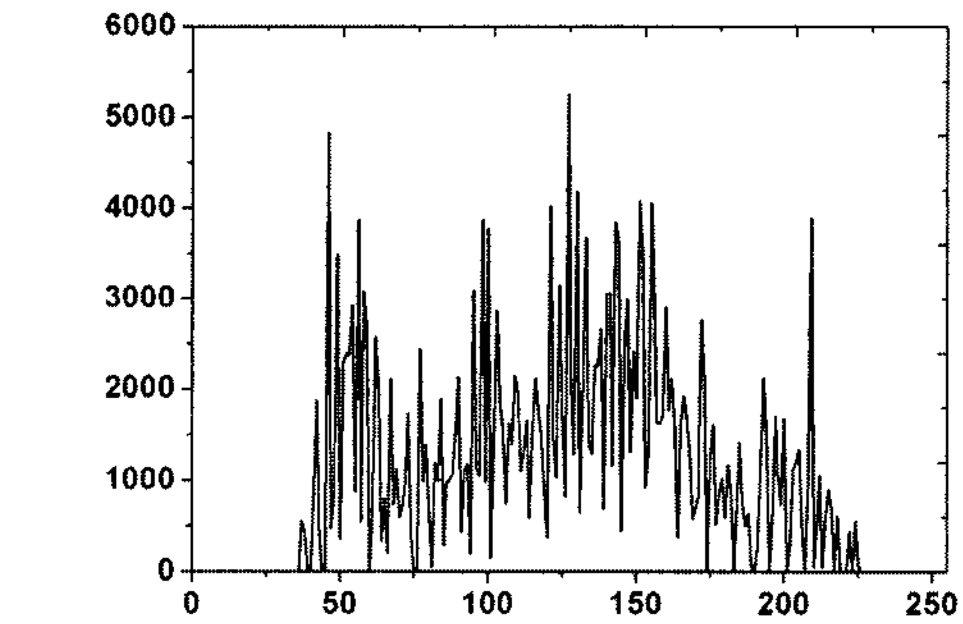
Images	Arithmetic (bpp)		saved bits (%)
	conven-tional	the proposed	
anemone	7.25	5.70	21.36
bike	7.02	4.82	31.36
boy	6.38	4.03	36.82
cafe	7.56	5.85	22.65
goldhill	7.47	5.45	27.10
lena	7.08	5.24	25.99
monarch	6.78	4.50	33.54
peppers	7.20	5.34	25.76
tulips	7.14	4.44	37.85
woman	7.25	4.91	32.32
average	7.11	5.03	29.47

(a) lena (b) tulips (c) woman

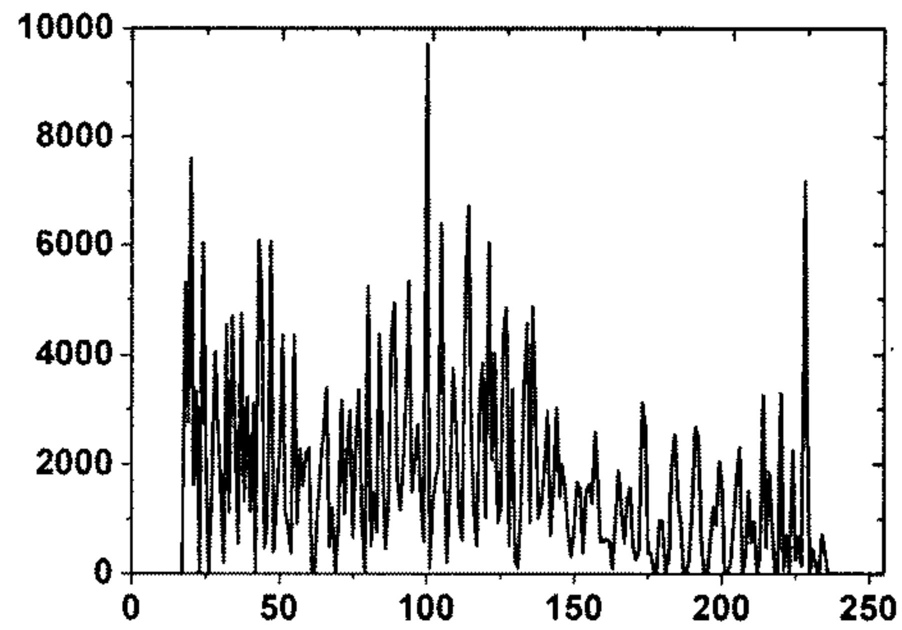
Fig. 5 Ranked images

As shown in Figure 6 and Figure 7, each histogram is concentrated in lower ranked 100 bins, i.e., higher ranks, and therefore this makes that we can expect higher efficiency in the compression ratio. Similar improvements are observed with other images. As shown in Figure 5 and Figure 7, the simulation results show that the bit cost is reduced effectively when encoding ranked images in table 1, table 2 and table 3.

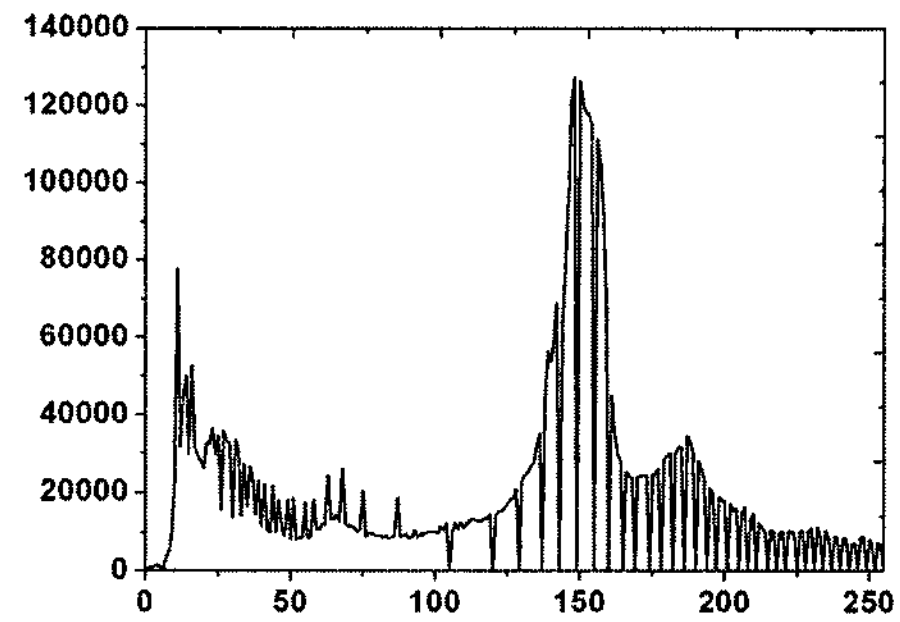
This simulation results obtained using 8 bit gray-level images show that the proposed method can reduce the bit rate by up to 37.85 % than other conventional methods. Further, we note that the proposed ranking method for efficient lossless image compression can improve compression performance in the entropy coders.



(a) lena



(b) tulips

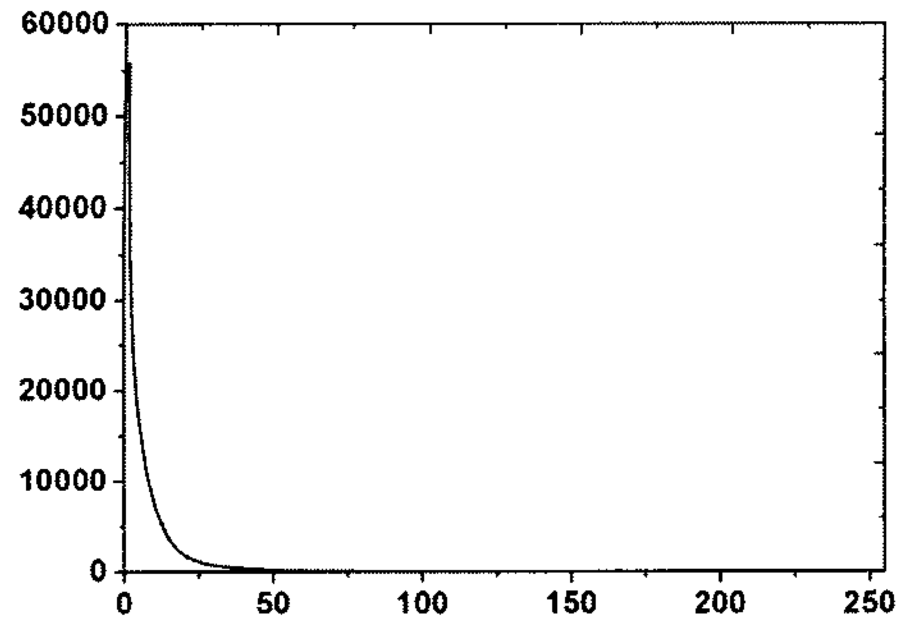


(c) woman

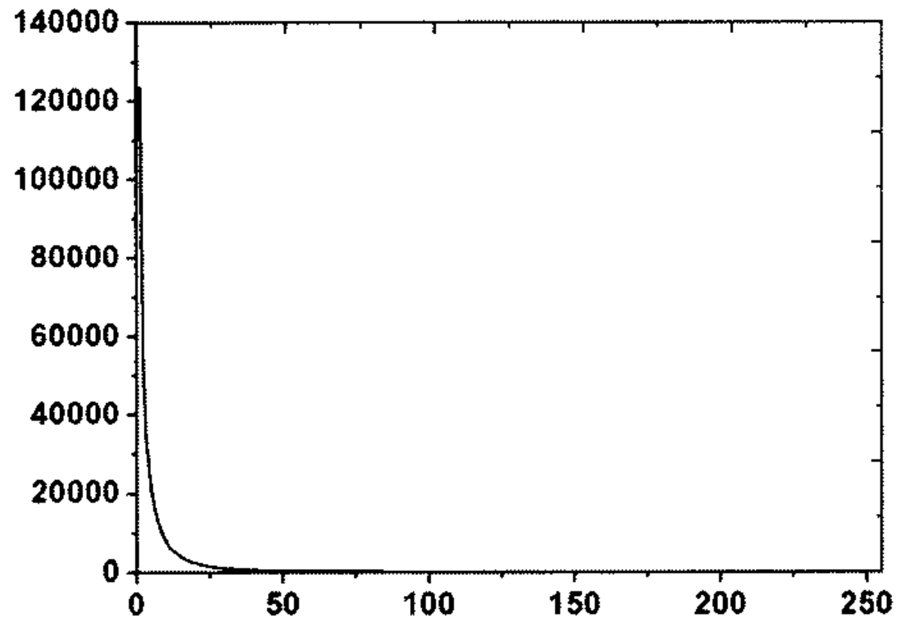
Fig. 6 Gray-level histogram

Ⅴ. Conclusions

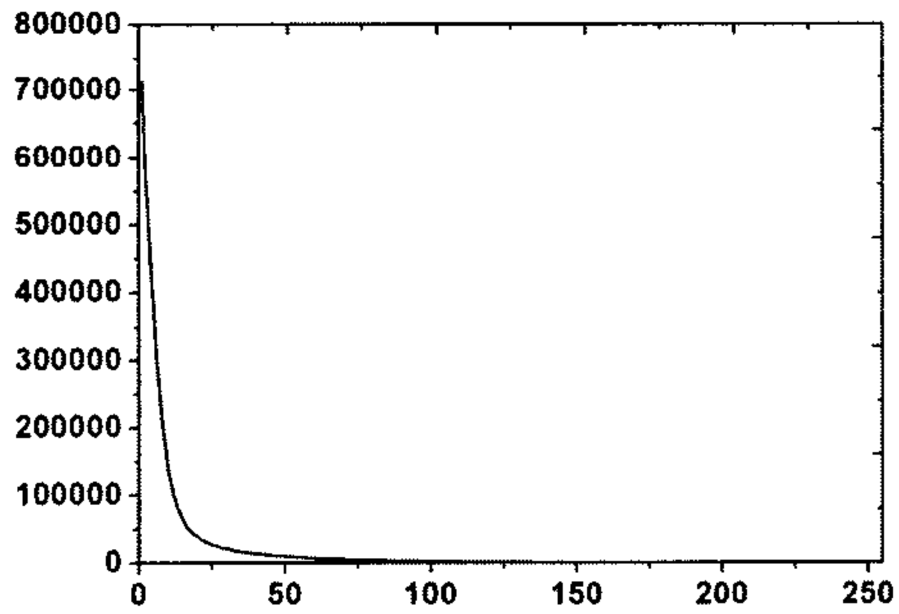
In this paper, we introduced a ranking method in order to achieve better performance of lossless image compression. The proposed method transforms an original gray-level image into a more compressible form based on



(a) lena



(b) tulips



(c) woman

Fig. 7 Ranked histogram

co-occurrence frequencies of two adjacent gray-levels. The ranked image is compressed more efficiently by the entropy encoder because data redundancy is increasing and spatial distribution of replaced image pixels is concentrated in certain ranks.

This method can be applied in various fields such as medical imagery used to preserve any original data and any application that needs to transmit digital images as fast as possible. When transforming and restoring an input image, the improved method operating without additional information may be considered. It can reduce the amount of data, thereby providing better compression rate.

All images as in Figure. 5 using the proposed method appear to be much smoother than the original image. This is expected to greatly facilitate the subsequent entropy coding.

References

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

[2] K. Sayood, 'Introduction to Data Compression', 2nd Ed., Morgan Kaufmann. 2000.

[3] T. A. Welch, "A Technique for High Performance Data Compression," IEEE Computer, Vol.17, No.6, pp.8-19, 1984.

[4] L. H. Witten, R. M. Neal and J. G. Cleary, "Arithmetic Coding for Data Compression," Communications of the ACM, Vol.30, No.6, pp.520-540, 1987.

[5] 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," IEEE Trans on Geoscience And Remote Sensing, Vol.39, No.10, pp.2298-2306, 2001.

[6] N. Memon and R. Rodila, "Transcoding GIF Images to JPEG-LS," IEEE Trans. on Consumer Electronics, Vol.43, No.3, pp.423-429, 1997.

[7] R. Hashemian, "Direct Huffman coding and decoding using the table of code-lengths," Proc. of Int., ITCC2003, Information Technology: Coding and Computing [Computers and Communications], pp.237-241, 2003.

[8] A. Moffat, 'Compression and Coding Algorithms,' Klower Academic Publishers, 2002.

[9] D. L. Duttweiler and C. Chamzas, "Probability Estimation in Arithmetic and Adaptive Huffman entropy Coders," IEEE Trans. on Image Processing, Vol.4, No.3, pp.237-246, 1995.

저자소개

Kang Soo You

Received his M.S. degrees in Dept. of Computer Engineering and Ph.D. degrees in Dept. of Image Engineering from Chonbuk National University, Jeonju, Korea in 1994 and 2006, respectively. In 2006, he joined the School of Liberal Arts, Jeonju University, Korea, where he is presently a full time instructor.
His research interests include Image Processing, Multimedia, and Computer Vision, etc.

Chun-Bo Sim

Received his M.S. and the Ph.D. degrees in Department of Computer Engineering from Chonbuk National University, Jeonju, Korea in 1998 and 2003, respectively. From 2004 to 2005, he joined the Division of Computer & Information Engineering, Catholic University of Pusan, Korea, where he worked as a full time instructor. In 2005, he joined the Division of Information & Communication Engineering, Sunchon National University, Korea, where he is presently an assistant professor.
His research interests include Multimedia Processing, Multimedia Database, Ubiquitous Computing, etc.