International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol. 10, No. 11 (2017), pp.33-40
http//dx.doi.org/10.14257/ijsip.2017.10.11.04

Image Compression Using Patterns

Vishwanath S. Kamatar[1] and Vishwanath P. Baligar[2]

K.L.E Institute of Technology, Hubli[1]
B.V.B College of Engineering & Technology, Hubli[2]
vishwanath.k@kleit.ac.in, vpbaligar@bvb.edu

Abstract

In this paper, we propose a method for image compression using pattern based approach. These patterns are found for all the blocks of size 4X2 in the image. Patterns are decided on the basis of pixel values. Pattern numbers, value of pixels, and the corresponding errors of each pixel are used to encode. The proposed method ensures improvement in result for near lossless compression of images. Results are also compared with standard compression algorithm, namely JPEG.

Keywords*: compression, near-lossless, patterns*

1. Introduction

Compression of gray scale images has been always a research interest among image processing community. Many methods have been proposed to achieve better compression ratio or to provide a tradeoff between compression ratio and computational complexity, to meet the requirements from several application areas, such as medical imaging, areal imaging, and image archiving. While using digital images, one of the most important limitations is that, the high memory inevitabilities for stowing digital images. As a consequence, images need to be compressed before they are either stored or shared. A number of compression algorithms are available in literature for compressing images. The two classes' of compression are lossless and lossy compression. Lossless compression method reconstructs the image by conserving its original information. For lossy compression, there may be some loss of information in the reconstructed image, which is acceptable. In this paper, we propose image compression algorithm that practices a pattern encoding approach. Proposed algorithm is in near-lossless mode. Firstly, image is divided into a number of non-overlapping blocks of pixels in order to get benefits of block processing. Then for each block, the suggested method finds a particular pattern value. Then the pattern, minimum pixel value of the block, and average error between pixels of the block are encoded. There are several algorithms, which use Huffman coding and combination of Huffman with other coding methods [3, 5, and 6]. Also, different algorithms are available with different coding methods. A novel method uses linear predictors for block of pixels, of images for lossless image compression, which has given best results compared to JPEG-LS [1]. Another method, Near-lossless image compression algorithm offers low average compression rate with high image quality for endoscopic images, provides lossless compression for region of interest (ROI) and high-quality compression for other regions with competitive performance [7]. In the other technique, image-folding method is applied. In this, column folding followed by row folding applied iteratively, until the image size reduced to some predefined value. By using property of adjacent neighbor redundancy for prediction, compression is achieved [8]. A compression method in which a well-known error correcting BCH codes, and followed by RLE is used and the results from this are given as input to second round compression using Huffman

Received (June 25, 2017), Review Result (October 24, 2017), Accepted (October 30, 2017)

ISSN: 2005-4254 IJSIP

coding to get higher compression [2]. In another scheme, the algorithm uses initially encoded several symbols to predict value of current symbol. The algorithm encodes these prediction errors by modified PPM and achieves the compression [10]. One more algorithm uses progressive near-lossless image compression, based on a lossy and near-lossless refinement layered compression scheme [11]. Another compression method known as new lossless compression technique is described and is well suited for asymmetric applications [9]. Though much of the work is focused on techniques in lossless image compression, the need for qualitative improvement necessitates development of some advanced technique. In the present work, by considering pixels within a particular block, we establish a particular pattern, based on the nature of pixel distribution. Then we take advantage of these patterns of pixels of the block.

The next section will give us a clear view about proposed method. Section 3 gives experiments and results. Section 4 gives future scope. Finally, section 5 concludes with discussions of computational efficacy of proposed method.

2. Proposed Method

The proposed method uses pattern concept for compressing the grayscale images. Pattern takes the form of permutations. Permutation is the arrangement of all the members of a set into some sequence or order, or if the set is already ordered, rearrangement of its members. Actually, in the proposed method, patterns are the sequence of pixels with their order of arrangement (arrangement with pixel values). To create patterns, alphabetic characters are used and permutation of alphabetic characters is considered. Proposed algorithm generates patterns before it starts compressing the input image. These patterns are stored in sequential order with indexes as lookup table. Proposed algorithm divides input image into a number of non-overlapping blocks. For each block, algorithm constructs the pattern of the block and searches the exact pattern among patterns, which are already present in the lookup table. If the exact pattern matches the entry in the lookup table, the index of that pattern from the lookup table is considered. The index of lookup table, the minimum pixel value of the block, and the variation used to encode to get bit stream. The same procedure applies for each block. The block to bit stream conversion performed according to a left-to-right, top-to-bottom manner. The organization of encoded bit stream is as follows.

a) Index of the pattern from the lookup table
b) Minimum value of gray pixel from the block
c) Four bits to represent a variation value

Steps (a) to (c) repeated for each block of the input image.

2.1. Near- Lossless Method

The following sub section describes the encoding and decoding procedures for Near Lossless method.

2.1.1. Encoding Steps

1. Alienate the input image into a number of non- overlapping blocks.
2. Consider a block and find the pattern for the block.
3. Search for pattern in lookup table and get the index of pattern.
4. Find the minimum pixel value from the block.
5. Find the variation between pixels for the block.
6. Convert the each value from steps 3 to 5 into encoded stream.
7. Repeat the steps 2 to 6 for each block to be encoded.

Repeat the steps 2 to 7 to cover the complete image left-to-right, top-to-bottom manner.

2.1.2. Decoding Steps

1. Read the Index value from bit stream and get the pattern from lookup table.
2. Read the minimum pixel value of the block from bit stream.
3. Read the variation value for the block.
4. Using the minimum pixel value and variation value, reconstruct the pixel values for the block.
5. Using the pattern obtained in step1, rearrange the pixels values reconstructed in the step 4 according to pattern.

Repeat the steps 1 to 5 to construct the blocks of the image according to a left-to right, top-to-bottom manner.

2.2. Near Lossless Method -Detailed Example

An example shows the conversion of block of image into encoded bit stream. Let us consider a block of image of size 4x2.

P[i, j] =

161	162
163	164
165	168
166	167

- First pattern is found, lowest pixel value being A and highest pixel value being H
- Pattern = ABCDEHFG for the above shown block.
- Get the index of the pattern from lookup table (pattern list). Index=5.
- Minimum pixel value of the block= A=161.
- Find the variation between the pixels = (max pixel value - min pixel value)/(No. of pixels-1)

These values are written in sequence as Pattern index=5, Minimum pixel value of the block=161, variation between the pixels=1 as shown in Figure 1.

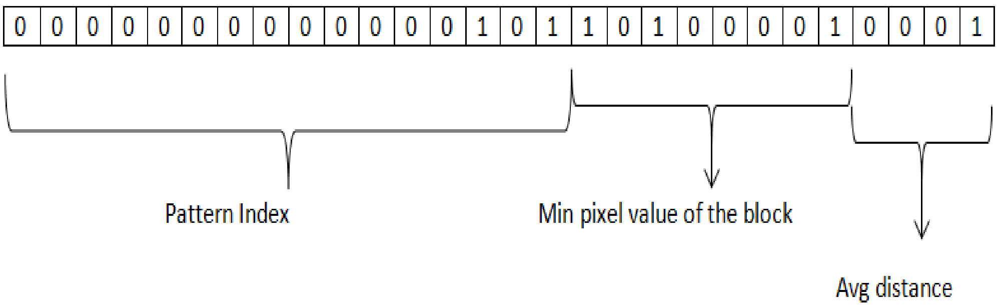

Figure 1. Detailed Example

3. Experiment and Results

To test suggested algorithm we have used several gray-scale images. We have considered only eight images to experiment our algorithm. The data set used for experiment is as shown in the Figure 2. All images used are of 512x 512pixels in size. All images are of bitmap images.

The designed algorithm is a near-lossless method. Here we have carried out 4 categories of experimentation are carried out with the above data set. Categories made depend on size of the blocks of images. The different block sizes used are with size 1 row 8 columns(1x8 pixels), 8 rows 1column(8x1 pixels), 2 rows 4 columns (2x4 pixels) and 4 rows 2 columns (2x4 pixels).

Figure 2. Images Used as Data Sets

The compression ratio is the ratio of original image size to the compressed image size and calculated by the formula.

$$compression\ ratio = \frac{original\ image\ size}{compressed\ image\ size}$$

3.1. Experimental Results for Block Size One Row and Eight Columns (1x8 pixels)

Performance analysis of the algorithm, with a block size of one row and eight columns (1x8 pixels) shown in Table 1. Table depicts size of images before and after the compression and the PSNR (Peak Signal to Noise Ratio) for reconstructed images. Compression ratio is 2.29 always as the algorithm follows unique method of steps to compress images. Reconstructed images are very near to original image but there is a noise in the reconstructed image at edges of the images. Due to this, the PSNR values from the table draw the attention that there is high variation at the edges in the images.

3.2. Experimental Results for Block Size Eight Rows and One Column (8x1 pixels)

Table 2 shows performance analysis of the algorithm with a block size of one row and eight columns (1x8 pixels). Reconstructed images are very nearer to original image but there is a noise in the edges of the reconstructed images. By PSNR values from Table 2, there is slight variation in PSNR values when compared with previous experimental results from Table 1. In this experiment, higher value of PSNR means reconstructed images are very similar to original images compared to previous experiment. From this, we conclude that block structure used also influences compression of images, and column major block structure is better when compared with row major block structure.

Table 1. Performance Analysis of Lossy Algorithm with Block Size 1x8 Pixels

Image	Bits before compression	Bits after compression	PSNR
Lena	2097152	917504	29.8603
Barbara	2097152	917504	29.7803
cameraman	2097152	917504	28.4469
Peppers	2097152	917504	29.8956
Mandrill	2097152	917504	30.2355
Living room	2097152	917504	29.0216
Jet plane	2097152	917504	29.3111
House	2097152	917504	33.8955

Table 2. Performance Analysis of Lossy Algorithm with Block Size 8x1 Pixels

Image	Bits before compression	Bits after compression	PSNR
Lena	2097152	917504	32.6957
Barbara	2097152	917504	32.1867
cameraman	2097152	917504	31.0795
Peppers	2097152	917504	30.3013
Mandrill	2097152	917504	30.1816
Living room	2097152	917504	29.9191
Jet plane	2097152	917504	28.7102
House	2097152	917504	33.3178

3.3. Experimental Results for Block Size Two Rows and Four Columns (2x4 pixels)

Performance analysis of the algorithm with a block size of two rows and four columns (2x4 pixels) shown in Table 3. In this experiment, instead of using row wise or column wise (one-dimensional) structure we grouped the pixels together into rectangular(two dimensional) structure. An idea behind this is that, as the nearest pixels grouped together, there will be correlation between the adjacent pixels, and there will be minute difference between adjacent pixel values. This helps in compression as the neighboring blocks will have almost same pattern indexes and reduces the difference between pixels. Due to this, the reconstructed image will be very much similar to original image as compared with previous experiment results. From Table 3, PSNR values are larger than previous results, due to the reconstructed images quality is increased and still there is small noise (mosaic structure) at the edges of reconstructed images.

Table 3. Performance Analysis of Lossy Algorithm with Block Size 2x4 Pixels

Image	Bits before compression	Bits after compression	PSNR
Lena	2097152	917504	34.0975
Barbara	2097152	917504	32.4100
cameraman	2097152	917504	33.8853
Peppers	2097152	917504	33.6578
Mandrill	2097152	917504	32.5528
Living room	2097152	917504	32.1896
Jetplane	2097152	917504	33.5917
House	2097152	917504	40.7882

3.4. Experimental Results for Block Size Four Rows and Two Columns (4x2 pixels)

Table 4 shows performance analysis of the algorithm with block size of two rows and four columns (2x4 pixels). In this experiment, everything is same as previous experiment but only change is that, column major is used. As we know that block structure also influences compression quality, here also reconstructed image has less noise (mosaic structure) at the edges in the reconstructed images as compared with previous experiment results. From Table 4, PSNR values are comparatively higher than all previous experimental results. Higher the value of PSNR, the quality of the reconstructed images will be good. Thus from these experiments it can be concluded that when using blocks approach for compression, column major block structure suits well.

Table 4. Performance Analysis of Lossy Algorithm with Block Size 4x2 Pixels

Image	Bits before compression	Bits after compression	PSNR	Compression Ratio
Lena	2097152	917504	35.3852	2.2857
Barbara	2097152	917504	32.6257	2.2857
cameraman	2097152	917504	35.0766	2.2857
Peppers	2097152	917504	33.5914	2.2857
Mandrill	2097152	917504	32.3440	2.2857
Living room	2097152	917504	32.4584	2.2857
Jet plane	2097152	917504	33.0625	2.2857
House	2097152	917504	39.9878	2.2857

Figure 3 shows reconstructed images using proposed method.

Figure 3. Reconstructed Images using Proposed Method

3.5. Comparison Study of Proposed Algorithm with JPEG Compression Algorithm

Proposed algorithm compared with standard JPEG compression algorithm, which is lossy in nature. Proposed algorithm ensures almost the same image quality as provided by the JPEG compression algorithm. Table 5 shows compression ratios for both proposed algorithm and JPEG compression algorithm. The table shows that compression ratio of JPEG algorithm is higher than proposed algorithm because, JPEG standard algorithm has complexity as it uses DCT, quantization and entropy coding. Proposed algorithm is very simple compared to standard JPEG algorithm.

Table 5. Comparison of Proposed Algorithm with JPEG Compression Algorithm

Image	Bits before compression	Bits after compression proposed algorithm	Compression Ratio for proposed algorithm	Bits after compression JPEG algorithm	Compression Ratio for proposed algorithm
Lena	2097152	917504	2.2857	786432	2.6666
Barbara	2097152	917504	2.2857	786432	2.6666
cameraman	2097152	917504	2.2857	786432	2.6666
Peppers	2097152	917504	2.2857	786432	2.6666
Mandrill	2097152	917504	2.2857	786432	2.6666
Living room	2097152	917504	2.2857	786432	2.6666
Jetplane	2097152	917504	2.2857	786432	2.6666
House	2097152	917504	2.2857	786432	2.6666

4. Further Research

The algorithm developed is based on the idea of patterns those behave like wavelets, as these patterns will be almost same for neighboring blocks. The discussion of performance analysis in the preceding section suggests that, still there is a scope to improve the algorithm to obtain better results. Further, refinement of this approach can reduce the number bits to be stored, by using variable length codes for encoding the block. The proposed method is for gray scale images. The same can be extended to compress color images with appropriate modification to algorithm.

5. Conclusion

Simple still image compression algorithm using patterns is presented. The proposed algorithm is for gray scale images and it is a near-lossless compression scheme. The developed algorithm assures less complexity. Proposed algorithm tested with images, for different block structure. Based on the thorough analysis, we conclude that the block structure we used for compression also influences the compression ratio. The experiments show that 4x2 pixels block structure will be more suitable for compression of images. The performance of the proposed algorithm compared with standard JPEG algorithm. The proposed algorithm gives good results, which are comparable with standard algorithms. Proposed method also ensures ease in implementation, as it is computationally efficient.

References

[1] V. P. Baligar, L. M. Patnaik and G. R. Nagabhushana, "Low complexity and high fidelity image compression using fixed threshold method", International Journal of Information Sciences, vol. 176, issue. 6, **(2006)**, pp. 664-675.
[2] R. Al-Hashemi and I. W. Kamal, "A New Lossless Image Compression TechniqueBased on Bose, Chandhuri and Hocquengham (BCH) Codes", International Journal of Software Engineering and Its Applications, July, vol. 5, no. 3, **(2011)**.
[3] J. H. Pujar and L. M. Kadlaskar, "A new lossless method of image compression and decompression using Huffman coding techniques", Journal of Theoretical and Applied Information Technology, **(2005)**.
[4] V. Biala and D. Kaur, "Lossless Compression Technique Using ILZW with DFRLC", International Journal of Computer Science Issues, vol. 9, issue 3, no. 3, **(2012)**.
[5] M. K. Mathur, S. Loonker and D. Saxena, "Lossless Huffman coding technique for image compression and reconstruction using binary trees", International Journal of Comp. Tech. Appl (IJCTA), vol. 3, issue no. 1, **(2011)**, pp. 76-79.
[6] C. Saravanan and R. Ponalagusamy, "Lossless Grey-scale Image Compression using Source symbols Reduction and Huffman Coding", International Journal of Image Processing (IJIP), vol. 3, no. 5.
[7] E. Kannan and G. Murugan, "Lossless Image Compression Algorithm For Transmitting Over Low Bandwidth Line", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2. no. 2, **(2012)**.
[8] S. Yerva, S. Nair and K. Kutty, "Lossless image compression based on data folding", International Conference on Recent Trends in Information Technology (ICRTIT), vol., no., **(2011)**, pp.999, 1004.
[9] N.D. Memon and K. Sayood, "Asymmetric lossless image compression", Proceedings Data Compression Conference, vol., no., **(1995)** pp.457.
[10] M. Kitakami and K. Tai, "Lossless Image Compression by PPM- Based Prediction Coding", Data Compression Conference, **(2009)**, vol., no., **(2009)**, pp.452, 452.
[11] A. Krivoulets, "A method for progressive near-lossless image compression", International Conference on Image Processing, vol.2, no., **(2003)**, pp., 185-8189.
[12] V. S. Kamatarn and V. P. Baligar, "Simple and low complexity scheme for lossless compression of gray scale images", International Journal of Applied Engineering Research, ISSN 0973-4562, vol. 10, no.86, **(2015)**.

