Journal of Theoretical and Applied Information Technology
10th April 2016. Vol.86. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

LOSSLESS IMAGE COMPRESSION METHOD USING REVERSIBLE LOW CONTRAST MAPPING (RLCM)

[1]HENDRA MESRA, [2]HANDAYANI TJANDRASA, [3]CHASTINE FATICHAH

[1]Department of Mathematics, Hasanuddin University, Makassar, Indonesia

[2,3]Department of Informatics, Sepuluh Nopember Institute of Technology, Surabaya, Indonesia

E-mail: [1]hendra@unhas.ac.id, [2]handatj@its.ac.id, [3]chastine@cs.its.ac.id

ABSTRACT

This research introduces a new lossless image compression method by a transform function named a Reversible Low Contrast Mapping (RLCM). This function was successful to be implemented as a reversible watermarking method. This proposed method divides an image into fixed-size blocks and groups the blocks into watermark blocks and host blocks. Furthermore, the watermark blocks are embedded in the host blocks. The proposed method does not use any other lossless compression method in its algorithm. Nevertheless, the proposed method can still be combined with other methods. On all the testing images, the compression ratio of the proposed method achieves 1.19 on average. As a comparison, the Huffman compression ratio is 1.08. By combining the method using the Differential coding and the Golomb encoding, the compression ratio of the proposed method increases to 1.32 on average.

Keywords: *Lossless Image Compression, Reversible Watermarking, Reversible Low Contrast Mapping, Differential Encoding, Golomb encoding.*

1. INTRODUCTION

Researches on data compression are still growing today. Besides the classic problems such as the amount of data stored in a storage medium continues to grow, more cost and time are required to transfer large data through the Internet network. The development of compression researches was caused also by the various types of data with different characteristics such as text data, audio, images and videos. The diversity requires different approaches to compress the data. The new data types in other research field such as DNA and X-ray images of medical field are a challenge for researchers to find the best method for compressing the data type.

There were several approaches used in methods of data compression such as Dictionary based compression, Run Length Encoding (RLE), Entropy or Statistical encoding [1]. The compression methods are developed to reduce redundancy of information in data. There are five types of redundancy: Spatial redundancy, Spectral redundancy, Temporal redundancy, Coding redundancy, and Psycho-visual redundancy[1].

Based on the quality of compression result, the compression methods are classified into Lossless and Lossy compression. Lossless compression does not tolerate any loss of information on the data, while lossy compression allows for the lost information with a certain tolerance limits.

In general, researches in lossless compression fields mostly an extension to increase the compression ratio of the existing methods and / or an extension for a specific data type. Some researches that extend the existing methods are a compression method using combination of the Lempel-Ziv-Welch (LZW) method and Bose, Chaudhuri and Hocquenghem (BCH) error correction algorithm [2], a compression method by combining the Huffman and Arithmetic coding on lossless image compression [3], and several studies to improve the compression ratio of Burrows-Wheeler Compression Algorithm (BWCA) [4]–[6]. While some researches that extend the compression methods for particular data types are a method for compression acoustic data using linear prediction [7], compression of floating-point coordinates in geometry data [8]. compression method for medical images by using the integer wavelet transform and predictive coding [9], compression of shape images using chain code by move-to-front transform and adaptive run-length encoding [10],

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

and DNA compression using Burrows-Wheeler Transform [11]. Several other studies on lossless compression methods on images data are discussed in [12].

However, the research to develop a purely new method in lossless compression is very rarely. One of these studies is the research of Burrows and Wheeler in 1994 that developed a compression method using a transform algorithm named Burrows-Wheeler Transform (BWT) [13] which was implemented in the bzip2 compression standard [14]. Nevertheless, the method still using Entropy coding in the algorithm. The research of Dude in 2009 was a furely new compression method. The method used Asymmetric Numeral System (ANS) on the algorithm [15]. This method has a similar compression ratio than Huffman coding.

Mesra et al in 2014 [16] developed a new lossless compression method using a reversible watermarking technique. This method was implemented on the image data. The scheme of the compression method is shown in Figure 1. The method used the Reversible Contrast Mapping (RCM) [17] in its algorithm. The RCM is a reversible watermarking method that has been implemented in some researches in data hiding fields [18]–[20].

The RCM compression method does not use any other compression method in its algorithm. The method partitions an image into blocks. Then, the blocks divide into the watermark blocks and the host blocks. The compression step is performed by embedding the watermark blocks into the host blocks. Although the method divides an image into some blocks but the compression process still uses a linear scan method to embed the watermark. So the method does not use the relation of all pixels on the blocks such as in spatial based methods. In the study, testing was conducted on the method for determining the compression ratio when the image is divided into different block sizes. Figure 2 shows the comparison of the compression ratio of the method in each block size and its comparison with Huffman compression.

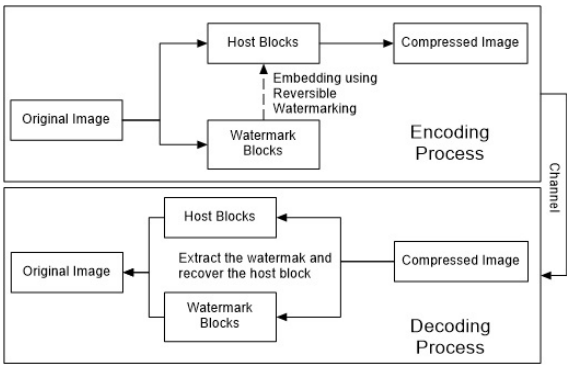

Figure 1. The RCM compression scheme.

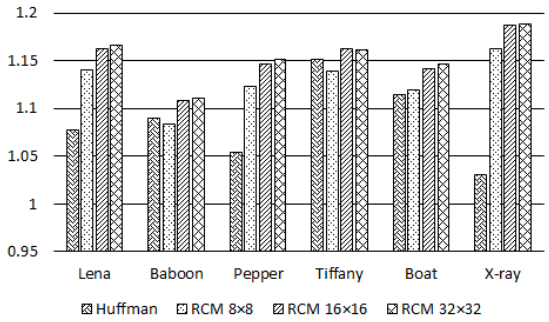

Figure 2. Comparison the Compression Ratio of RCM and Huffman Compression.

The proposed compression method also uses watermarking scheme as in [16] but it uses another transformation function. This research uses a function that was used in [21] as a reversible watermarking method. For simplifying, the function is named as Reversible Low Contrast Mapping (RLCM) in this research.

Several recent researches on watermarking methods by using the RCM function were focused on the generalized of the transform function. Chen et al developed a general function of the RCM [22], Maity et al used a modified of the RCM as a watermarking method [23], and Maity and Maity developed a generalized M-ary RCM forward transform [24]. Some variant of the functions has the same characteristics to the RLCM but they still used the basic rules of the RCM watermarking algorithm. Whereas the RLCM uses a different approach in the algorithm.

Journal of Theoretical and Applied Information Technology
10th April 2016. Vol.86. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2. REVERSIBLE LOW CONTRAST MAPPING (RLCM)

RLCM is an integer transform function was defined by:

$$x' = \left\lfloor \frac{3x - y}{2} \right\rfloor \text{ and } y' = \left\lfloor \frac{3y - x}{2} \right\rfloor \quad (1)$$

The transform function has a special property if the values of x and y are positive integers, that is the both value of x' and y' are an odd number or an even number. This property can be used for embedding a watermark by changing the Last Significant Bits (LSB) of y' and using the LSB of x' as a control bit in detection and recovery process.

The inverse of this function was defined by:

$$x = \left\lfloor \frac{3x' + y'}{4} \right\rfloor \text{ and } y = \left\lfloor \frac{3y' + x'}{4} \right\rfloor \quad (2)$$

Equation (2) are the inverse function of the RLCM if the values of x' or y' from (1) are not modified. By using the special property, although the LSB of y' was changed for embedding a watermark, the original value of y' can be restored by using the LSB of x'. Therefore, (2) can be used to get the original values of x and y.

In eight bits image domain (D), the values of x and y are in the range 0-255, therefore, to prevent overflow and underflow, all pixel pairs must be in the image domain that is $(x, y) \in D$. The value of x and y in the image domain if the following conditions are satisfied:

$$0 \le \left\lfloor \frac{3x - y}{2} \right\rfloor \le 255 \text{ and } 0 \le \left\lfloor \frac{3y - x}{2} \right\rfloor \le 255$$

2.1. Embedding Watermark Algorithm

Such as in the RCM watermarking method, the critical point of the RLCM function is on the boundary of the domain, that is where $\lceil (3x - y) / 2 \rceil = 0$, $\lceil (3x - y) / 2 \rceil = 255$, $\lceil (3y - x) / 2 \rceil = 0$, and $\lceil (3y - x) / 2 \rceil = 255$.

If a pair (x, y) is at a critical point, the extraction process will result in a fault value of (x, y), so the watermark will not be extracted correctly and the original image will not be recovered. To avoid the condition, the value of y must be changed in the watermarking algorithm by using an equation:

$$y_{max} = \arg \max_{y} \left(|x - y_0|, |x - y_1| \right) \quad (3)$$

The y_{max} is the value of y after its LSB is changed to "0" (y_0) or changed to "1" (y_1) so that the difference of x and y becomes maximum.

The algorithm for embedding the watermark described as follows:

1) Partition the image into pairs of pixels (x, y) in non-overlapping manner.
2) For all pairs (x, y)
   a. Compute the y_{max} by using (3).
   b. If $(x, y_{max}) \in D$
      1. Compute x' and y' by (1)
      2. Compute y'_{max} from x' and y' by using (3)
      3. If $(x', y'_{max}) \in D$ then y' can be used to embed a watermark.
   c. If $(x, y_{max}) \notin D$
      1. Set $x' = x$ and $y' = y$
      2. The LSB of y' must be saved as a recovery bit.
3) Merge watermark bits and recovery bits.
4) Embed the composite watermark into y' that satisfied the condition in 2.b.3 by changing its LSB.

Based on the algorithm, the domain of all pairs (x, y) is divided into three regions such as shown in Figure 3a. The region of A is the Embeddable region, all pairs in this region satisfy the condition of $(x, y_{max}) \in D$ and $(x', y'_{max}) \in D$, so that all pairs in the region can be used to embed a watermark. The region of B is the Changeable region, all pairs in this region satisfy only one condition that is $(x, y_{max}) \in D$ but do not satisfy the condition of $(x', y'_{max}) \notin D$. All pairs in this region are transformed by (1) but they cannot be used to embed a watermark. And the region C is the non-Embeddable region, all pairs in this region do not satisfy the both condition, so the LSB of y' in this region must be saved for recovery process. As a comparison, the regions of the RCM algorithm is shown in Figure 3b. The RCM divides the domain into the Embeddable region (A) and the non-Embeddable region (C).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

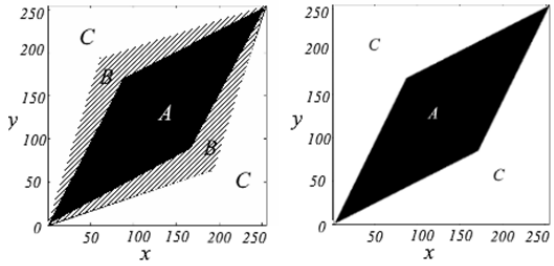

Figure 3. The comparison of domain regions on (a) The RLCM algorithm and (b) The RCM algorithm.

2.2. Extraction and Recovery Algorithm

In regard to the embedding watermark algorithm, the extraction and recovery process are developed as the reverse steps of the algorithm. The algorithm is described as follows:

1) Partition the watermarked image into pairs of pixels (x', y')
2) For all pairs (x', y')
   a. Compute the y'_{max} by using (3).
   b. If $(x', y'_{max}) \in D$
      1. Extract the LSB of y' as a bit watermark
      2. Set the LSB of y' = the LSB of x'
      3. Transform (x', y') by using (2).
   c. If $(x', y'_{max}) \notin D$
      1. If the LSB of x' = the LSB of y' then transform (x', y') by using (2).
      2. If the LSB of $x' \neq$ the LSB of y' then saves this pair position for the recovery process.
3) Separate the watermark bits and the recovery bits.
4) Recover the image by change the LSB of y' at the pair was found in step 2.c.2 uses the recovery bits.
5) Reconstruct the watermark data.

3. LOSSLESS COMPRESSION SCHEME

The proposed method uses a similar scheme with the RCM compression. The compression scheme divides the image into fixed-size blocks, then the blocks are divided into host block and watermark block. The compression process is performed by embedding the watermark blocks into the host blocks.

In the RCM compression method, implementation of a shift operation increased the compression ratio of the method on an image with light or dark intensity. Therefore, this proposed method implements the shift operation as a step in the algorithm. The shift operation is defined by function:

$$x_{new} = (x - \lfloor \bar{x} \rfloor + 128) \bmod 256 \quad (4)$$

where x is pixel value and $\lfloor \bar{x} \rfloor$ is the rounding down of average pixel values of the image.

The inverse of this function is used for decompression process, the inverse function is defined by:

$$x_{new} = (x + \lfloor \bar{x} \rfloor - 128) \bmod 256 \quad (5)$$

All blocks are sorted based on the embedding capacity in the RCM compression method. The step can increase the compression ratio, but it needs more space to save the information of original order of all blocks thereby the increasing is insignificant. Besides, the ordering step increases the algorithm complexity. Therefore, the proposed method does not involve it in the algorithm.

3.1. Compression Algorithm

The main process of the proposed method is the same as the RCM compression, except for the sorting step, the embedding watermark algorithm, and the algorithm for extraction and recovery. The lossless compression algorithm is described as follows:

a. Shift the pixel values of the image using (4)
b. Partition the image into n blocks.
c. Select the first block as a host block and the last block as a watermark block.
d. Transform the watermark block into watermark bits.
e. Embed the watermark bits into the host block using the embedding watermark algorithm iteratively until the block is full and then save the iterations number of the block.
f. If the host block is full, select the second block and set it as a new host block. Embed the remaining bits to the host block.
g. If the remaining bits are empty, select a block at the n-1 position, transform the block into watermark bits and embed it into the host block.
h. Repeat the process until the position of host block and watermark block are the same.
i. Save the number of host blocks, the pixel values of the host blocks, and the iteration number of all host blocks.

Based on the algorithm, the output of the proposed method consists of the image size, the value of $\lfloor \bar{x} \rfloor$, the number of all blocks, and the number of host blocks as a single data item, and the iteration number and the pixel values of each host block as an array data.

Journal of Theoretical and Applied Information Technology
10th April 2016. Vol.86. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.2. Decompression Algorithm

The decompression process uses all output of compression result. The algorithm is described as follows:

a. Select first block of host blocks.
b. Read the iteration number of the block.
c. Extract the watermark from the block by using the extraction and recovery algorithm.
d. Separate the watermark bits and the recovery bits and recover the block by using the recovery bits.
e. Select the next host blocks and then repeat step (c) and (d). Perform this step to the remaining host blocks.
f. Reconstruct the watermark blocks from the watermark bits.
g. Reconstruct the image based on the information of image size.
h. Shift the pixel values to the original value by using (5)

4. EXPERIMENTAL RESULT

As well as the study in RCM compression, this study uses six gray-level images as the testing images such as shown in Figure 4. This study performs the testing by using three different block sizes that is 8×8, 16×16 and 32×32.

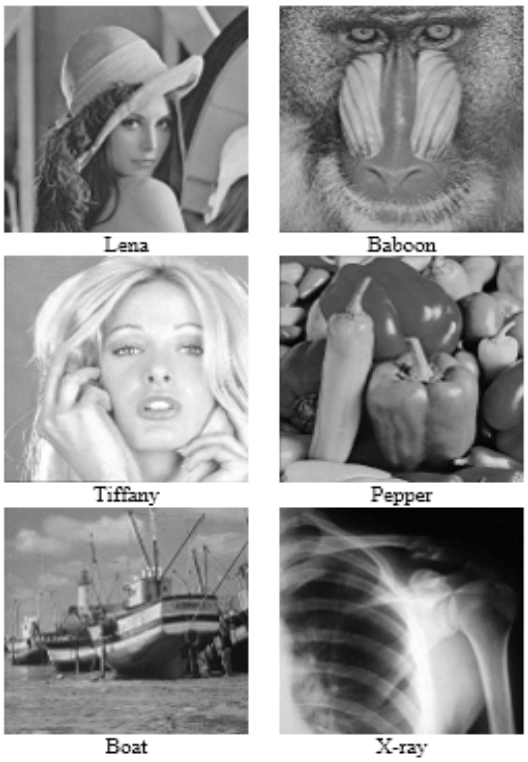

Figure 4. Sample of the test images

To measure the performance of the proposed method, this study uses the Compression Ratio (CR) that is expressed by equation:

$$CR = \frac{n_1}{n_2}$$

where n_1 is the file size of the original image and n_2 is the size of the compressed image.

Table 1 describes the comparison of compression ratio of the Huffman method and the proposed method on all testing images and block sizes. The table shows that the block size 16×16 achieves a higher compression ratio to Pepper, Tiffany, Boat, and X-ray images. Whereas the Lena and Baboon images archive the higher compression ratio on the block size 32×32. The difference of the two block sizes is small and insignificant.

Table 1. Comparison of proposed method on the three block sizes and the Huffman method.

No	Images	Compression Ratio			
		Huff.	8×8	16×16	32×32
1	Lena	1.078	1.201	1.211	1.212
2	Baboon	1.090	1.125	1.136	1.137
3	Pepper	1.054	1.180	1.190	1.190
4	Tiffany	1.151	1.197	1.206	1.201
5	Boat	1.114	1.160	1.170	1.168
6	X-ray	1.031	1.235	1.245	1.242
Average		1.086	1.183	1.193	1.192

The comparison of the best result of the proposed method, the Huffman encoding and the best compression ratio on the RCM compression is shown in Figure 5. The proposed method yields a higher compression ratio than the other method for all test images. On the comparison to the RCM compression, the proposed method has a higher compression method because the difference of (x', y') of the RCM transform is three times larger than the initial value, that is $|x' - y'| = 3|x - y|$, whereas the difference of the RLCM is $|x' - y'| = 2|x - y|$. The small difference will increase the iteration number of a block, so the proposed method can embed more data than the RCM compression. Figure 6 shows the comparison of the average number of iterations on the both methods at the best value of the compression ratio.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

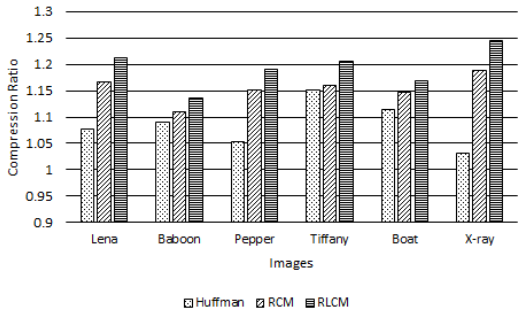

Figure 5. Comparison Ratio Compression of the Huffman, the RCM and the RLCM Compression

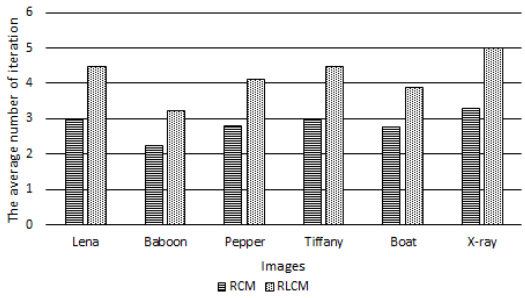

Figure 6. Comparison the average iteration number of the RCM and the RLCM compression.

The compression ratio of the proposed method can be increased by involving the other lossless compression method in the algorithm. This research has also been tested by engaging the Differential coding and the Golomb encoding to compress the watermark blocks before it is embedded into the host blocks. The compression process of watermark blocks is shown in Figure 7. In the golomb coding, the value of *m* was determined by average of *C* values. All of *m* value of the watermark blocks must be saved for extraction and recovery process.

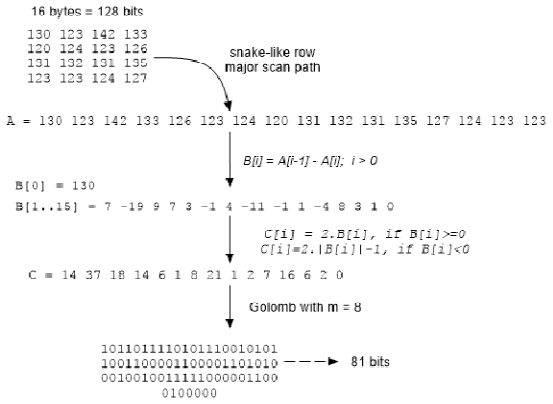

Figure 7. Sample of compression process using the Differential coding and the Golomb encoding on a watermark block.

Table 2 shows the compression ratio of the modified method. The method gives a higher compression ratio than the original method such as shown in Figure 8. The reduction of watermark blocks data size by the Differential coding and the Golomb encoding increases the compression ratio because there are more watermark blocks can be embedded into the host blocks. On all test images, the 16×16 block size gave high compression ratio than the others. In this method, the small size of block can embed more data than the bigger size. But the small size block need more space to save the number of iteration of all host blocks and the value of *m* of the watermark blocks.

Table 2. Comparison of the modified method on the three block sizes

No	Images	Compression Ratio		
		8×8	16×16	32×32
1	Lena	1.333	1.344	1.339
2	Baboon	1.161	1.173	1.173
3	Pepper	1.291	1.304	1.304
4	Tiffany	1.326	1.335	1.325
5	Boat	1.247	1.260	1.260
6	X-ray	1.494	1.501	1.478
Average		1.309	1.320	1.313

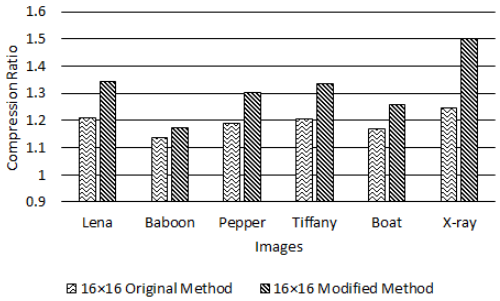

Figure 8. Comparison of compression ratio of the modified and the original method on the 16×16 block size.

5. DISCUSSION

This research used Differential coding and Golomb encoding to compress the watermark blocks. The using of other compression method on the blocks will result different compression ratio. There are many compression methods can be used to change the differential coding and the golomb coding. Besides, a watermark block can be viewed as an image therefore the image lossless compression method can also be used.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Besides compressing the watermark block, the compression ratio can also be increased by compressing the compressed image. Directly compression on the compressed image does not increase the compression ratio significantly. Figure 9 shows a block after compression process. By using S-Transform [25], the pixel values of the *l* block can still be used to embed a watermark and the *h* block can be compressed then becomes a watermark bits. The compression process can still be continued by using multi resolution method. Besides this way, the compression process can also be directly used at the S-Transform results.

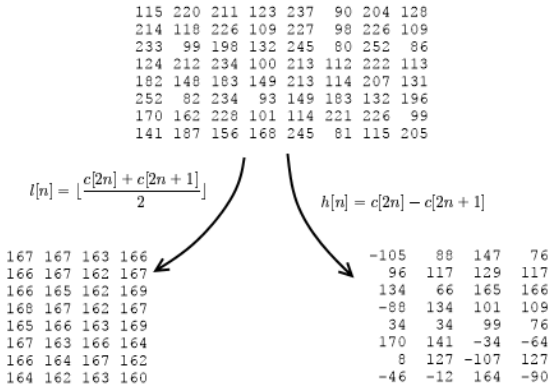

Figure 9. Sample of S-Transform process on a compressed block

6. CONCLUSION

The RLCM compression obtained higher compression ratio than the Huffman encoding, and the RCM Compression although the method does not involve other compression method in the algorithm. Based on the block size, in average, the 16×16 size of block gave a higher compression ratio than the others.

The compression ratio of the proposed method can still be increased by combining the method with other compression method. By using the differential coding and the Golomb encoding, the modified method obtained compression ratio about 1.32 in average. The compression ratio can still be increased by further compression to the compressed image such as by using the S-Transform.

The disadvantages of the proposed method are not all pixels in a block are used to embed the watermark because the original watermarking method only uses the *y* value of (*x*, *y*). Besides, not all pairs get their maximum capacity because the algorithm restrict the iteration number of a block although there are some pixel pairs can still be used to embed a watermark.

REFERENCES:

[1] S. Rao and P. Bhat, "Evaluation of lossless compression techniques," in *Communications and Signal Processing (ICCSP), 2015 International Conference on*, 2015, pp. 1655–1659.

[2] A. Alarabeyyat, S. Al-Hashemi, T. Khdour, M. H. Btoush, S. Bani-Ahmad, R. Al-Hashemi, S. Bani-Ahmad, and others, "Lossless Image Compression Technique Using Combination Methods," *J. Softw. Eng. Appl.*, vol. 5, no. 10, p. 752, 2012.

[3] P. G. Howard and J. S. Vitter, "Parallel lossless image compression using Huffman and arithmetic coding," *Inf. Process. Lett.*, vol. 59, no. 2, pp. 65–73, 1996.

[4] E. Syahrul, "Lossless and nearly-lossless image compression based on combinatorial transforms," Université de Bourgogne, 2011.

[5] J. Abel, "Improvements to the Burrows-Wheeler compression algorithm: After BWT stages," *ACM Trans Comput. Syst.*, 2003.

[6] M. Effros, "PPM performance with BWT complexity: A new method for lossless data compression," in *Data Compression Conference, 2000. Proceedings. DCC 2000*, 2000, pp. 203–212.

[7] S. I. Ao and International Association of Engineers, Eds., *Lossless Compression Method for Acoustic Waveform Data Based on Linear Prediction and Bit-recombination Mark Coding*. Hong Kong: IAENG, International Association of Engineers, 2013.

[8] M. Isenburg, P. Lindstrom, and J. Snoeyink, "Lossless compression of predicted floating-point geometry," *Comput.-Aided Des.*, vol. 37, no. 8, pp. 869–877, Jul. 2005.

[9] T. G. Shirsat and V. K. Bairagi, "Lossless medical image compression by integer wavelet and predictive coding," *ISRN Biomed. Eng.*, vol. 2013, 2013.

[10] B. Žalik and N. Lukač, "Chain code lossless compression using move-to-front transform and adaptive run-length encoding," *Signal Process. Image Commun.*, vol. 29, no. 1, pp. 96–106, Jan. 2014.

[11] D. Adjeroh, Y. Zhang, A. Mukherjee, M. Powell, and T. Bell, "DNA sequence compression using the Burrows-Wheeler Transform," in *Bioinformatics Conference,*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2002. Proceedings. IEEE Computer Society, 2002, pp. 303–313.

[12] M. Rehman, M. Sharif, and M. Raza, "Image compression: A survey," *Res. J. Appl. Sci. Eng. Technol.*, vol. 7, no. 4, pp. 656–672, 2014.

[13] M. Burrows and D. J. Wheeler, "A block-sorting lossless data compression algorithm," *SRC Res. Rep. 124*, 1994.

[14] D. Salomon, *Data compression: the complete reference*. Springer Science & Business Media, 2004.

[15] J. Duda, "Asymmetric numeral systems," *ArXiv Prepr. ArXiv09020271*, 2009.

[16] H. Mesra, H. Tjandrasa, and C. Fatichah, "A new approach for lossless image compression using Reversible Contrast Mapping (RCM)," in *Information, Communication Technology and System (ICTS), 2014 International Conference on*, 2014, pp. 71–76.

[17] D. Coltuc and J.-M. Chassery, "Very fast watermarking by reversible contrast mapping," *Signal Process. Lett. IEEE*, vol. 14, no. 4, pp. 255–258, 2007.

[18] H. K. Maity and S. P. Maity, "Joint robust and reversible watermarking for medical images," *Procedia Technol.*, vol. 6, pp. 275–282, 2012.

[19] Y. Yang, X. Sun, H. Yang, C.-T. Li, and R. Xiao, "A contrast-sensitive reversible visible image watermarking technique," *Circuits Syst. Video Technol. IEEE Trans. On*, vol. 19, no. 5, pp. 656–667, 2009.

[20] Z. Liu, X. Sun, Y. Liu, L. Yang, Z. Fu, Z. Xia, and W. Liang, "Invertible transform-based reversible text watermarking," *Inf. Technol. J.*, vol. 9, no. 6, pp. 1190–1195, 2010.

[21] Hendra, "Reversible Watermarking Using Integer Transform," Thesis, Universitas Gadjah Mada, 2008.

[22] X. Chen, X. Li, B. Yang, and Y. Tang, "Reversible image watermarking based on a generalized integer transform," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, 2010, pp. 2382–2385.

[23] D. Maiti, S. P. Maity, and H. Maity, "Modification in contrast mapping: Reversible watermarking with performance improvement," in *Signal Processing and Communications (SPCOM), 2012 International Conference on*, 2012, pp. 1–5.

[24] S. P. Maity and H. K. Maity, "M-ary reversible contrast mapping in reversible watermarking with optimal distortion control," in *Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2013 Fourth National Conference on*, 2013, pp. 1–4.

[25] S. Sahni, B. C. Vemuri, F. Chen, C. Kapoor, C. Leonard, and J. Fitzsimmons, "State of the art lossless image compression algorithms," in *IEEE Proceedings of the International Conference on Image Processing*, 1997, pp. 948–952.