Wavelet-Walsh, Quantization and Fractal Coding Transformation Methods to Minimize Image Data Size

Yuli Asriningtias[1], Febri Saputro[2], Emy Setyaningsih[3]
[1,2] *Department of Informatics, Faculty of Information Technology and Electrical Engineering University of Technology Yogyakarta, Indonesia*
[3]*Department of Computer System Institut Sains & Teknologi AKPRIND Yogyakarta, Indonesia*
[1]*yuli_asriningtias@uty.ac.id* [2]*febrisaputro28@gmail.com*, [3]*emypurnomo@akprind.ac.id*

ARTICLE INFO

Article history:
Received June 15, 2018
Revised July 20, 2018
Accepted August 15, 2018

Keywords:
Compression, wavelet-Walsh transformation, scalar quantization, fractal.

ABSTRACT

The development of digital technology requires an efficient and fast process, not only in data transmission but also in data stored. The type of data most often exchanged is digital images, but good quality digital images are large. One way to reduce the size of image data is to use the compression method. A compression method is to reduce or compress the size of data while maintaining the quality of information contained therein. This research proposed the hybrid compression method using Wavelet-Walsh transformation, scalar quantization, and Fractal coding to compress image data greyscale so that it has a smaller size but still with excellent image quality. The test result showed that the average Compression Ratio is 1.32 with the decent reconstruction image result quality that is average value PSNR = 43.31 dB.

I. Introduction

Compression is a technique to reduce the size of data by eliminating data redundancy and strong correlations between adjacent pixels. The compression itself is consists of two methods, namely lossy and lossless. Lossless is a compression technique that can produce the same reconstruction image as the original image without information loss, while lossy is the opposite of the lossless method. The results of lossy compression reconstruction are not the same as the original image because this method allows the loss of some information contained in the image data [1].

Recently the need to store image data in small sizes so that the transmission process becomes faster is increasing. Therefore, the compression technique is hardly needed to minimize image data size. Many studies have discussed the proposed compression method with various advantages and disadvantages [2]. On average, the development of studies about compression merging several processes into one. The effectiveness level is measured using Peak Signal to Noise Ratio (PSNR) value and compression ratio (CR). PSNR is used to measure the relative mistake to signal square average value, which is regulated by the logarithmic scale [3]. CR is used to measure the input ratio towards the compression result output file based on file size before and after the compression process [3].

Several researchers succeeded in developing a compression method by merging several ways. Research conducted by [4] proposes a compression method using a hybrid of wavelet transforms with vector quantization. The experiment results in the proposed methods generated good compression quality and succeeded in minimizing average data redundancy 32%. [5] developed a compression method using a fractal approach that aims to reduce image size and maintain image quality so that it remains high in the decompression process. The test results show that the proposed method can minimize data redundancy around 13.7%, but the image quality is not good because the PSNR value is less than 30 dB, i.e., PSNR = 27.3 dB. [6] also developed a compression method by combining the transformation and coding methods. The technique developed starts with a wavelet and fractal transformation, which aims to reduce the number of block domains. In the next step, the coding process uses the Huffman method. The research result showed a compression ratio which way much higher than other lossless methods. The research conducted by [7] proposed a compression model with a new parallel scheme to overcome the weakness of the fractal algorithm, namely the long compression time. The test results show that the image quality from the reconstruction results is not good, which is shown by the average value of PSNR = 29.9 dB, but the compression ratio is quite efficient with a value of CR = 5.83%.

Based on previous studies, the results of reconstruction and image compression ratio is still less than optimal, so there are opportunities to develop compression models for image data. This paper proposed compression methods that merge the wavelet transformation method with Walsh filter, scalar quantization, and fractal coding. The choice of wavelet transformation algorithm for the transformation process is due to the wavelet transformation characteristics that have a high ratio and speed in the compression process. While the choice of the scalar quantization method, because the quantization process can categorize image intensity, it becomes simpler so that it can reduce the size of the image, and the image quality is still quite good when seen with the ordinary eye. The choice of fractal method for the coding process because fractal compression file sizes are generally very small due to the reduction in the size of the image data using local similarities in the image. Therefore the proposed method can be used to compress images that have high local similarities such as images related to nature.

This paper is organized as follows: Section 2 elaborates on the proposed method, followed by Section 3, that contains the evaluation and analysis. Finally, section 4 concludes this paper.

II. Proposed Method

A. *Compression Scheme*

Proposed compression method scheme consists of four parts showed in figures 1

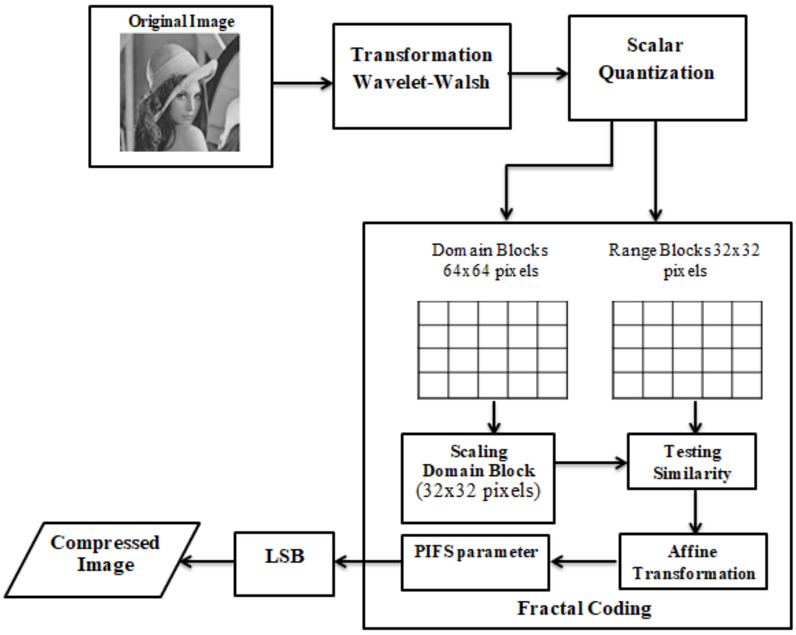

Fig 1: Compression Scheme

The compression process stage is as follows:

Step 1: Walsh transformation method is done by dividing an image into a 4 x 4 size sub-image. There are 16 matrix filters on the modified Walsh transformation filter (each filter has 4 x 4 size). Each different matrix value is appropriate with the upper left coordinate on the 4 x 4 sub-image, which will be processed. For example, to get the image on the coordinate (0,0), it uses the first image matrix, which operated with 4 x 4 sub-image; it also goes for coordinate (0,1) second matrix filter and so forth. To get an 8-bit (0-225) value, so when the re-resolution process occurs, the matrix does not result on out of range value, so the Walsh transformation filter uses the equation (1).

$$NewWalsh\, Filter = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \tag{1}$$

Step 2: Perform the scalar quantization process by reducing the number of bits by each level of image intensity. If an 8-bit image is quantized, the image size will change to 7-bits. Rules for grouping image intensities using equation (2)

$$m(i,j) = \begin{cases} (\mathrm{Im}(i,j)+1)/2 & if\ \mathrm{Im}(i,j) \bmod 2 = 0 \\ \mathrm{Im}(i,j)/2 & if\ \mathrm{Im}(i,j) \bmod 2 \neq 0 \end{cases}. \tag{2}$$

Step 3: The process of fractal coding is done by divide the image into several blocks of the same size and not overlap each other, called the range block. Furthermore, it also made several domain blocks whose size is two times the range block. The formation of a domain block can be overlap or not overlap. The advantage of forming not overlap domain blocks is that the number of domain blocks is fewer, so the matching time is faster. But the results are not as good as using overlap domain blocks. If used domain blocks overlap, the number of domain blocks will increase so that the possibility of local self-similarity is also high, and the matching time will be longer.

This study uses overlapping domain blocks while the range block is part of a block whose divide does not overlap. So to encode the image F, we divide it into range blocks $R_1, R_2, \ldots, R_i \ldots, R_n$, such that

$$f = R_1 \cup R_2 \cup \ldots \cup R_n \tag{3}$$

and

$$R_i \cap R_j = 0\ with\ i \neq j\,. \tag{4}$$

That is, the range blocks cover the whole image and do not overlap[8].

After that, the image is divided into domain blocks $D_1, D_2, \ldots, D_j, \ldots D_m$. For each range block R_i, we find a contractive transform W_i and a domain block D_j, so that it meets equation (5)

$$R_i \approx W_i(D_j). \tag{5}$$

The combination of $W_i, W_2, \ldots, W_i, \ldots W_n$ is called Partitioned Iterated Function System (PIFS) W. If W is simpler than the original image, we can encode f into W and achieve certain compression. The size of the range block is B x B pixels, whereas, for the domain block, the size is 2B x 2B pixels (twice the size of the range block), and they overlap every B pixels in both x-direction and y-direction. The choice of size for the range block determines the results of the compression ratio and the quality of the reconstructed image. If the quality of the reconstructed image is high, the range block uses a small size (4x4 pixels and below), but the achievable compression ratio will not be very high. Using a small range block makes it easy to match the domain block. Conversely, if used a range block with a large size (8x8 pixels and above), then the matching process in the domain block is more complicated, although it will produce a high compression ratio. Next, measure the similarity between domain blocks and range blocks is measured using the Root Mean Square (RMS). The smaller RMS measure, the more alike the two image blocks. The level of similarity of image blocks u and v of size B xB pixels is defined using equation (6)

$$d(u,v)=\sqrt{\sum_{i,j}(u(i,j)-v(i,j))^2}\ . \tag{6}$$

where summation is for i = 0 to B−1 and j = 0 to B–1. Before each block (Ri) in the range block (Wi) is matched, each block in the domain block (Dj) must first be scaled to ½ the part. The scaling process makes it easy to calculate the distance between domain blocks and range blocks. The scaling process is by changing the 2x2 pixel block to one pixel by calculating the average value of the four pixels. If the range blocks and domain blocks have a high similarity, the affine transformation will map the domain blocks to range blocks. This research will do compression on grayscale images, so it needs to extend the form of an affine transformation. If the pixel depth at position (x, y) as z = f (x, y), then the affine transformation is extended using the equation (7)[8]

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = W_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix} \tag{7}$$

if simplified, it becomes equation (8)

$$v_i \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix}. \tag{8}$$

With the W_i mapping above, the intensity of each pixel is also scaled and shifted using equation (9)

$$z' = s_i z + o_i \tag{9}$$

The parameter s_i represents the pixel contrast factor. If s_i is 0, then the pixel darkens. If s_i is equal to 1, then the contrast does not change, whereas if the value is between 0 to 1, then the contrast decreases, and if it is greater than 1, then the contrast will increase. The o_i parameter represents the pixel brightness offset. A positive o_i value will brighten the image, and if a negative o_i value will darken. Both the s_i and o_i parameters can accurately map grayscale domain blocks to grayscale block ranges as well.

The parameters e_i and f_i express the shift of the left corner of the domain block to the left corner of the corresponding block range. Whereas si and oi are calculated using the regression formula in equation (10) and (11)

$$E = \sum_{i=1}^{n} (d'_i - r_i)^2 = \sum_{i=1}^{n} (s.d_i + o - r_i)^2 \tag{10}$$

$$DRMS = \frac{\sqrt{E}}{n} \tag{11}$$

The search sequence continues for the next range blocks until all range blocks have been paired with the domain block with its affine transformation. The result of the compression process is many local Iterated Function System (IFS) called PIFS. All PIFS parameters are packaged and saved in an external file. The PIFS parameters that need to be stored are only the parameters e_i, f_i, s_i, o_i and the type of symmetric operation for each block range. In reality, the parameters e_i and f_i are replaced by the position of the domain block mapped to the block range. While the parameters a_i, b_i, c_i and d_i do not need to be stored because the value is fixed at ½ for a_i and d_i and 0 for b_i and c_i.

Step 4: Hide the range block, which is similar to the domain block (parameters $e_i, f_i, s_i,$ and o_i) in the top-left coordinate using the Least Significant Bit (LSB) method. By changing the binary matrix value at index 0 with the binary bit data to be protected. The process for hiding the PIFS parameter uses a matrix of compression results. The coordinate hiding by the LSB method can increase decompression speed by not effecting the matrix value LSB process, and it is because the intensity value only increases one or decrease 1.

B. Decompression Scheme

The decompression method scheme is shown in Figure 2.

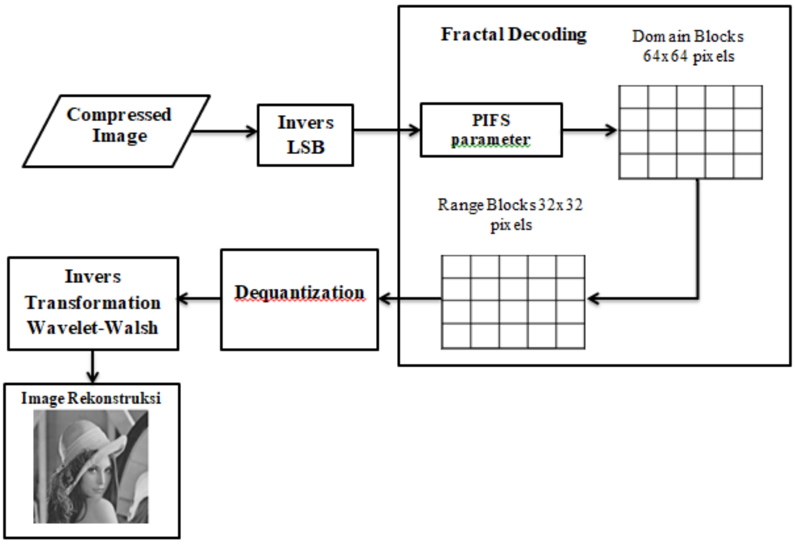

Fig 2: Decompression Scheme

The decompression process stage is as follow:

Step 1: The method of LSB inverse is used to return the information which is hidden on the compressed image matrix. That information is in the form of x and y coordinate (upper left) from domain block, which is useful for the decompression process, especially in the fractal decoding process part.

Step 2: The decoding process is the same as the coding process using fractals in the compression process. The difference lies in the division of domain blocks that use PIFS parameter information that was previously hidden using the LSB method.

Step 3: The purpose of the dequantization process is to return to the 8-bit form again, using the rules in equation (12)

$$\mathrm{Im}(i,j) = \begin{cases} (m(i,j)+1)*2 & if\ m(i,j)\bmod 2 = 0 \\ m(i,j)*2 & if\ m(i,j)\bmod 2 \neq 0 \end{cases}. \tag{12}$$

Step 4: The method of inverse Wavelet-Walsh transformation uses the same process as the compression process.

III. Evaluation and Analysis

A test towards the proposed compression method performance is implemented using Windows 8.1, NetBeans application IDE 7.0.1 operated on the laptop with CPU specification using AMD processor A8 and 4GB RAM.

The performance of the proposed method will be evaluated by implementing the method to two group grayscale images with different characteristics. The test image is a Lena-grey.bmp image with 512x512 size which is divided into two groups, they are:

a. Contrast level
   Image group based on the contrast that is (in sequence) high contras, medium contrast and low contrast.

Fig 3: Lena Image With Different Contrast.

b. Brightness level
   Image group based on the brightness that is (in sequence) high brightness, medium brightness and low brightness.

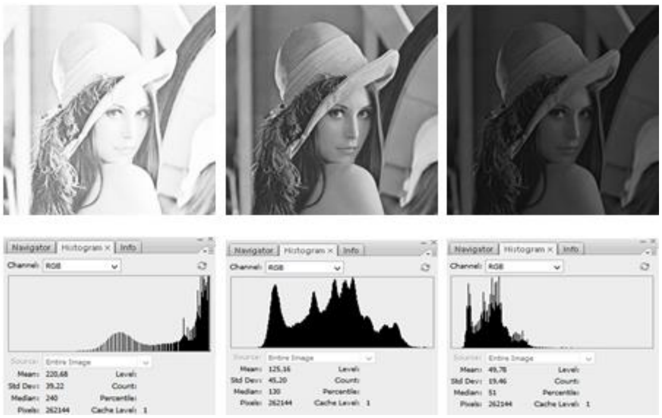

Fig 4: Lena Image With Different Brightness

The proposed method performance to count the efficiency level of a compression method measured based on CR value using equation (13).

$$CR = \frac{b}{b'} \tag{13}$$

with b is the bit amount before the compression process and b' is the bit amount after the compression process.

The test result towards various images with different contrast levels and brightness levels resulted in average CR 1.32 as seen in table 1. If it is examined from CR value so the proposed method is definitely effective if it is used on a high contrast image.

Table 1 Compression Ratio result

Image Name	Size	Compression Ratio
LenagreyHighContrast.bmp	512 x 512	1.32
LenagreyMediumContrast.bmp	512 x 512	1.19
LenagreyLowContrast.bmp	512 x 512	1.28
LenagreyHighBrightnes.bmp	512 x 512	1.17
LenagreyMediumBrightnes.bmp	512 x 512	1.19
LenagreyLowBrightnes.bmp	512 x 512	1.76
Average		**1.32**

While to measure image quality of decompression result, it is measured quantitatively using PSNR measurement. Compression technique is said to be a good technique if it can get a small Mean Squared Error (MSE) value and high PSNR value, which means that the error or mistakes from this compression technique is quite small and reconstruction result image towards original image has high similarity. MSE is patterned with equation (14).

$$MSE = \frac{1}{M \times N} \sum_{y=1}^{M} \sum_{x=1}^{N} [f_1(x,y) - f_2(x,y)]^2 \quad (14)$$

With f1 (x,y) and f2 (x,y) is the original image and decompression result image with its own size m x n. Signal ratio value towards the peak noise or PSNR value can be counted with equation (15).

$$PSNR = 10 * \log_{10} \frac{255^2}{MSE} \ . \quad (15)$$

Table 2. MSE and PSNR results.

Image Name	Size	MSE	PSNR (dB)
LenagreyHighContrast.bmp	512 x 512	2.97	43.41
LenagreyMediumContrast.bmp	512 x 512	3.57	42.60
LenagreyLowContrast.bmp	512 x 512	2.70	43.83
LenagreyHighBrightnes.bmp	512 x 512	4.49	41.61
LenagreyMediumBrightnes.bmp	512 x 512	3.57	42.60
LenagreyLowBrightnes.bmp	512 x 512	1.69	45.83
Average		**3.17**	**43.31**

The test result of all images shows that the proposed methods are quite useful, it is proven from the quality of reconstruction result image, which is quite high that is PSNR average value: 43.31 dB, as showed in table 2. The proposed method also works optimally if it is implemented towards image with quite bright quality. This can be seen from the resulted PSNR value is rather high than PSNR value on the image with other quality.

The compression and decompression time average is also quite fast, that is 8.63 second and 0.71 seconds as presented in table 3.

Table 3. Compression and decompression time result.

Image Name	Size	Compress (second)	Decompress (second)
LenagreyHighContrast.bmp	512 x 512	11.64	0.99
LenagreyMediumContrast.bmp	512 x 512	6.05	0.67
LenagreyLowContrast.bmp	512 x 512	11.57	0.94
LenagreyHighBrightnes.bmp	512 x 512	11.63	0.80
LenagreyMediumBrightnes.bmp	512 x 512	6.05	0.67
LenagreyLowBrightnes.bmp	512 x 512	4.86	0.20
Average		**8.63**	**0.71**

If Lena image with medium contrast characteristic or medium brightness level compared to Lossless Wavelet Fractal (WFC) which proposed by [5], so the CR value is not way much different than the proposed method. Whereas based on the reconstruction result image quality, the method proposed by the author is quite decent, which is PSNR = 42.60 dB. While on the Lossless WFC, the reconstruction result is still under the method which proposed as presented in table 4

Table 4. Comparison Result CR and PSNR

Proposed Method		**Lossless WFC**	
CR	PSNR(Db)	CR	PSNR(Db)
1.19	42.60	1.38	39.61

IV. Conclusion

The compression method proposed by merging Wavelet transformation with Walsh filter, scalar quantization, and fractal coding has decent performance. It can be seen from the PSNR average = 43.31 dB above PNSR value required compressed image reconstruction result quality that is 30 dB. The proposed method can minimize the average image size of CR = 1.32. The test result showed that the proposed method would be optimal if it is implemented to compress images with low brightness level characteristics. Further research is recommended to test the technique in color images.

V. Reference

[1] B. Xiao, G. Lu, Y. Zhang, W. Li, and G. Wang, "Lossless image compression based on integer Discrete Tchebichef Transform," *Neurocomputing*, vol. 214, pp. 587–593, Nov. 2016.

[2] E. Setyaningsih and A. Harjoko, "Survey of Hybrid Image Compression Techniques," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 4, p. 2206, Aug. 2017.

[3] I. M. Pu, *Fundamental Data Compression*. Jordan Hill, Oxford OX2 8DP 30 Corporate Drive, Suite 400, Burlington, MA 01803: Elsevier, 2006.

[4] H. B. Kekre, P. Natu, and T. Sarode, "Color Image Compression Using Vector Quantization and Hybrid Wavelet Transform," *Procedia Comput. Sci.*, vol. 89, pp. 778–784, 2016.

[5] S. Liu, Z. Zhang, L. Qi, and M. Ma, "A fractal image encoding method based on statistical loss used in agricultural image compression," *Multimed. Tools Appl.*, vol. 75, no. 23, pp. 15525–15536, Dec. 2016.

[6] F. Mekhalfa, M. R. N. Avanaki, and D. Berkani, "A Lossless hybrid wavelet-fractal compression for welding radiographic images," *J. Xray. Sci. Technol.*, vol. 24, no. 1, pp. 107–118, Mar. 2016.

[7] A.-M. H. Y. Saad and M. Z. Abdullah, "High-speed implementation of fractal image compression in low cost FPGA," *Microprocess. Microsyst.*, vol. 47, pp. 429–440, Nov. 2016.

[8] G. Lu, "Fractal-Based Image and Video Compression," in *The Transform and Data Compression Handbook*, 1st ed., K. R. Rao and P. C. Yip, Eds. CRC Press LLC, 2001, pp. 313–343.