International Journal of Multimedia and Ubiquitous Engineering
Vol. 10, No. 3 (2015), pp. 299-310
http://dx.doi.org/10.14257/ijmue.2015.10.3.27

Optimization Weighted Matrix of Non-Negative Matrix Factorization for Image Compression

Robin[1] and Suharjito[2]

[1]*Faculty of Information Technology, Mikroskil, Medan, Indonesia*
[2]*Magister of Information Technology, Bina Nusantara University Jakarta, Indonesia*
robin80wong@yahoo.com, harjito@yahoo.com

Abstract

Many methods have been applied in image compression and Non-negative matrix factorization (NMF) is one of some approach which could be applied in image compression. Non-Negative Matrix Factorization (NMF) was a realtively new approach to decompose data into two factors with non-negative entries. This paper shows that the NMF method can be applied in image compression using a model of 2x2 pixels weighted matrix rather than using the model of 200x200 pixels, 10x10 pixels and 4x4 pixels weighted matrix. This paper also shows that the 2x2 Weighted matrix model example of ((65536, 1), (1, 65536)) was the best weighted matrix for NMF image compression. Finally, this research proved that by using the weighted matrix with higher determinant value could gain smaller size compressed image.

Keywords*: image compression, non-negative matrix factorization, matrix model*

1. Introduction

Field of data compression has been a daily necessity in data storage and transmission through computer networking. The aim of image compression is to reduce size of data without reducing or eliminating the original data to gain more effective and efficiency storage space. This is also happening in image compression. An effective image compresion will result in a more efficient image storage and shorter time in transferring image through network [1,2] A common characteristic of most of the images is that the neighboring pixels are correlated and therefore contain redundant information. The foremost task is to find less correlated representation of the image. Redundancy reduction aims at removing duplication from the signal source (image). Irrelevancy reduction omits parts of the signal that will not be noticed by the signal receiver, namely the Human Visual System (HVS) [2]. An Image is a set of graphical data stored in numerical values which is non-negative so that the NMF algorithm can be used to process and compress image [3].

The use of NMF in digital image processing has been carried out by several studies such as for image segmentation [3], for image classification [4], for pattern recognition [5] and for compressing the digital image [6]. The implementation of the NMF algorithm in image compression can be done by determining the right size and the right determinant of weighted matrix [4, 6]. Problems that often arise in the use of NMF is the determination of the weighted matrix in accordance with the characteristics of a digital image. The aim of this paper is to determine the most suitable weighted matrix size and determinant of weighted matrix to be used in image compression.

ISSN: 1975-0080 IJMUE

2. Literature Review

Image compression means reducing the byte size of graphical file without reducing the meaning of the image to unacceptable level [7]. A common characteristic of most of the images is that the neighboring pixels are correlated and therefore contain redundant information. The foremost task is to find less correlated representation of the image. Two fundamental components of compression are redundancy and irrelevancy reduction. Redundancy reduction aims at removing duplication from the signal source (image). These are three types of redundancy can be identified.

- *Spatial Redundancy* or correlation between neighboring pixel values.
- *Spectral Redundancy* or correlation between different color planes or spectral bands.
- *Temporal Redundancy* or correlation between adjacent frames in sequence of images (in video applications) [2]

There are also three types of redundancy can be identified.

- Coding redundancy
  Coding redundancy is present when less than optimal code words are used. Interpixel redundancy results from correlations between the pixels of an image.
- Interpixel redundancy

  An interpixel redundancy can be predicted from the value of neibouring pixels.

- Psychovisual redundancy

Psychovisual redundancy is due to data that is ignored by the human visual system (*i.e.*, visually non essential information) [1].

Most 2-D intensity arrays contain information that is ignored by the human visual system and extraneous to the intended use of the image. It is redundant in the sense that it is not used. Image compression research aims at reducing the number of bits needed to represent an image by removing the spatial and spectral redundancies as much as possible [8].

A matrix named V can be factorize into W and H with equation

$$V \approx W * H \qquad (1)$$

Where V is the original image matrix to be factored; W is the factor matrix represent the weight of V matrix which V matrix will factorized into H matrix. Each of data vector V can be measured with a linear combination of columns of W matrix which is the weight for H matrix components. Therefore W can play role as optimized base for linear approach from the data inside V. Since the relatively few base vectors to be used to represent data vectors, a good approximation can only be achieved if the base vector discover the latent structure in the data [9].

3. Methodology

The first step is starting the research by collecting paper of NMF method theory and research, studying the concept of matrix factorization. NMF method has often applied in image segmentation and bioinformatics. Because the concept of factorization is that an integer value can be factorized into two factors where these factors are always smaller than the integer value itself, the NMF method might be useful in image compression.

After the first step is done, the second step is collecting the images. Because there are several characteristic of images, this research is done by testing the NMF method into images with different characteristic. There are eight characteristic of images in common to be tested in this research. These are the categories.

1. Images with low color value dominant
2. Images with high color value dominant
3. Images with red intensity dominant
4. Images with green intensity dominant
5. Images with blue intensity dominant
6. Images with various color used
7. Images with less color used
8. Grayscale images

The third step is the implementation of the NMF method into the compression using a program which is coded in C++ language. After the implementation process is complete, the next fourth step is testing NMF method into compression.

Testing of NMF method for image compression will be done in several ways. Because of the size of original image (V matrix) to test is 200x200 pixels, the W matrix to be used for image compression in the first testing is 200x200 pixels. Because of counting the determinant of image having size of 200x200 pixels to gain the inverse of W (W^{-1}) might result in an stack overflow error while doing a recursion, the second test to do is to map the original image into many smaller pieces of image and try to compress again using the smaller W matrix. The size of W matrix to try after the first test has fail is began from 10x10 pixels, 4x4 pixels and 2x2 pixels in sequence after each try has fail. The reason why these size of W matrix is chosen is that the length and width of many images can be factorize into those size.

After the determinant of W is gained, the next step is to initialize the element of W matrix randomly and try to get the W matrix with the bigger determinant value. The bigger determinant value is needed to gain smaller W^{-1}. The aim of gaining smaller W^{-1} is to get smaller H matrix (the compression result) by multiplying W^{-1} with V.

A colored pixel is consists of the red, green and blue intensity which each intensity has 1 byte of size (256 variant color for each intensity) so the size of one colored pixel is 3 byte which has color values between (0 to 16777215) where 16777215 is gain from 256*256*256-1. A pixel can be said successfully compressed if only the size of compressed pixels is below 3 bytes. Therefore, the best size for a compressed pixel is 1 byte.

According to equation (1), if the V matrix (original image) and the W (weight) matrix is known, the H matrix can also gained by reversing the equation above into:

$$H=V * W^{-1} \tag{2}$$

To prove that the determinant value can greatly affected the compression result, the function to compute the matrix determinant in C++ language which contains recursion is listed below.

```
long Determinant(long double X[n][n], int Jlh)
{
 int brs,klm,c,i,j;
 long double tmp[n][n];
 long double d;
 d=0;
 if (Jlh==2)
```

```
  d=X[0][0]*X[1][1]-X[0][1]*X[1][0];
 else {
  for(c=0; c<Jlh; c++) {
   brs=0;
   for(i=1; i<Jlh; i++) {
    klm=0;
    for(j=0; j<Jlh; j++) {
     if (j==c) continue;
     tmp[brs][klm]=X[i][j];
     klm++;
    }
    brs++;
   }
   d+=(pow(-1,c)*X[0][c]*Determinant(tmp,Jlh-1));
  }
 }
 return d;
}
```

4. Result and Implementation

The resolution of each the sample image used in this research is 200x200 pixel where the actual size of each sample image is 120000 bytes (200x200 pixels x 3 bytes of RGB channel).

The first try with W matrix of size 200x200 as predicted had resulted in a stack overflow error inside the function of determinant while doing recursions. A large amount of memory is needed to do the recursions from 200 to 3 (200*199*198* ... * 3) because doing recursions from 200 to 3 cost 197 times is beyond the maximum capacity of stack size C++ Builder which is 16777216 bytes (1000000 in hexadecimal).

The second try using 10x10 pixels W matrix doesn't contains any error but compression result of H matrix contains a few negative values. The negative values of the H matrix occur because of multiplying the V matrix with W^{-1} matrix containing negative values has a probability to result in negative values. The negative values in H matrix have caused the 10x10 pixels W matrix cannot be applied in image compression.

The third try, using the 4x4 pixels W matrix is also result in negative values inside the H matrix which the part of those matrix is shown below.

Table 1. Part of the V Matrix from Index [0][0] to [3][3]

row/column	0	1	2	3
0	133632	265218	396804	594183
1	199425	331011	462597	528390
2	331011	396804	528390	725512
3	462597	528390	725769	857355

Table 2. Part of the W Matrix from Index [0][0] to [3][3]

row/clmn	0	1	2	3
0	65536.0	32768.5	16384.7	1.0
1	32768.5	65536.0	32768.5	16384.7
2	16384.7	32768.5	65536.0	32768.5
3	1.0	16384.7	32768.5	65536.0

Table 3. Part of the W-1 Matrix from Index [0][0] to [3][3]

row/clmn	0	1	2	3
0	2.09*E-5	-1.05*E-5	-1.74*E-5	3.49*E-6
1	-1.05*E-5	2.56*E-5	-9.30*E-6	-1.74*E-6
2	-1.74*E-6	-9.30*E-6	2.56*E-5	-1.05*E-5
3	3.49*E-5	-1.74*E-6	-1.06*E-5	2.09*E-5

Table 4. Part of the H Matrix from Index [0][0] to [3][3]

row/clmn	0	1	2	3
0	1.75	3.24	5.07	8.63
1	-0.18	1.08	1.50	-0.95
2	1.53	1.08	0.93	3.64
3	6.33	7.25	10.24	11.50

Table 1 shows a part of the sample of a 4x4 V matrix while table 2 shows sample of a 4x4 W matrix. Table 3 shows that the inverse of W matrix contains many negative values. As the result, multiplying the V matrix with W^{-1} matrix resulted in an H matrix which contains a few negative values in it. Table 4 shows H[1][0] and H[1][3] contains negative values.

The fourth try, using the 2x2 pixels W matrix is succeeded without errors and even result in a non-negative H matrix which means NMF algorithm can be applied using the 2x2 pixels W matrix which part of those matrix is shown below.

These table below shows the V matrix and W matrix from index [0][0] to [1][1].

Table 5. Sample Data of 2x2 Pixels V Matrix

row/column	0	1
0	133632	265218
1	4677459	454587

Table 6. Sample Data of 2x2 W Matrix

row/column	0	1
0	65536.0	1
1	1	65536.0

Table 7. W-1 matrix from index [0][0] to [1][1]

row/clmn	0	1
0	1.52587891*E-5	-2.3283064*E-10
1	-2.3283064*E-10	1.52587891*E-5

Table 8. The Image Compression H Matrix from Index [0][0] to [1][1]

row/column	0	1
0	2.0390160682	4.0468284490
1	3.0429528958	5.0507652766

Table 5 shows a part of the sample of a 2x2 V matrix, table 6 shows a sample of 2x2 W matrix and Table 7 shows the inverse of W matrix. The H matrix gained by multiplying the V matrix and W^{-1} matrix which is shown in Table 8 doesn't contain any negative values. This means that the 2x2 W matrix can be used in image compression

The fifth try is using the first model of 2x2 pixels W matrix having a large amount value of determinant like the one showed below.

Table 9. Sample of W Matrix, Determinant of W Matrix and the Inverse of the Sample W Matrix

W matrix		Determinant	W^{-1} matrix	
65536	1	4294967295	$1.53x10^{-5}$	$-2.33x10^{-10}$
1	65536		$-2.33x10^{-10}$	$1.53x10^{-5}$

The sixth try is using the second model of 2x2 pixels W matrix having determinant of almost equal to 1 like the one showed below.

Table 10. Another Sample of W Matrix, Determinant of W Matrix and the Inverse of the Sample W Matrix

W matrix		Determinant	W^{-1} matrix	
65536	65535	1.0000305	65534.001	65533.001
65535	65536		65533.001	65534.001

Both of the fifth and the sixth try is using V matrix with the largest value of colored pixel which shown below.

Table 11. Sample of V Matrix

row/column	0	1
0	16777215	16777215
1	16777215	16777215

Result of H matrix from fifth try for each element is below 256 like the table showed below

Table 12. Compression Result Using Sample of W Matrix in Table 9

row/clmn	0	1
0	255.9960785510475	255.9960785510475
1	255.9960785510475	255.9960785510475

While the result of H matrix from sixth try is shown in Table 13:

Table 13. Compression Result using Sample of W Matrix in Table 10

row/column	0	1
0	2198939238408	2198939238408
1	2198939238408	2198939238408

From the implementation above, using the first model W matrix can be used in image compressoin.

Figure 1 to Figure 8 below shows the comparison of compression result size of 30 200x200 pixels 24-bit color images (120000 bytes) for each image categories in JPEG and NMF in bytes.

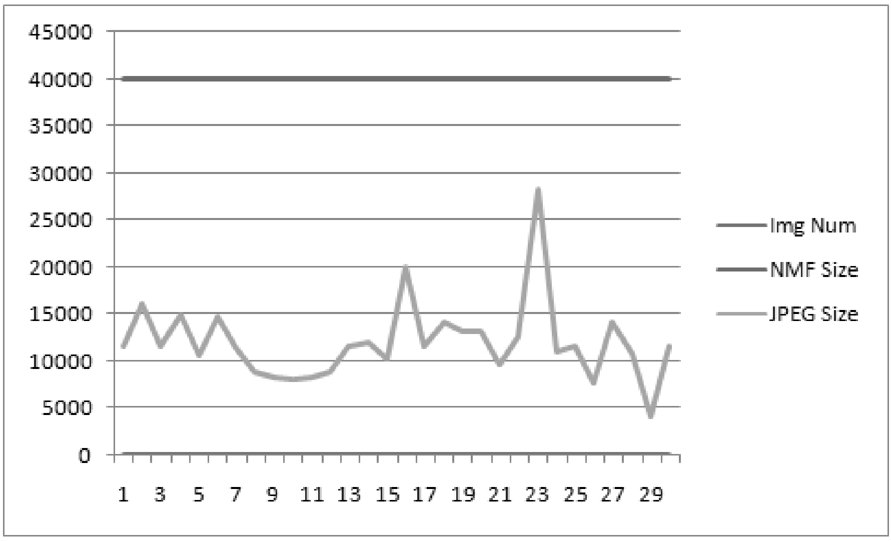

Figure 1. Compression Result of 30 Low Color Value Images Using JPEG and NMF

The various size of compressed low color value image using JPEG shows in Figure 1 is caused by various color value contain in each image while the compression result using NMF is always 40000 bytes.

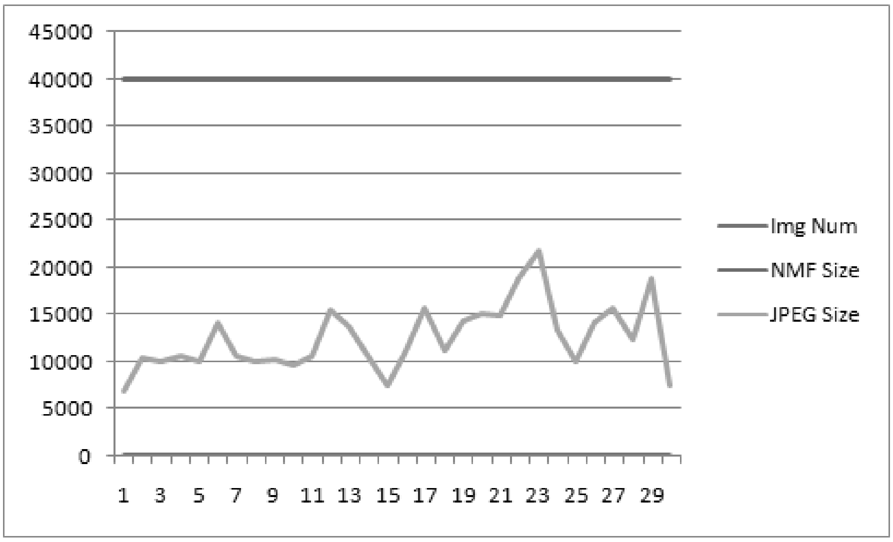

Figure 2. Compression Result of 30 High Color Value Images Using JPEG and NMF

The various size of compressed high color value image using JPEG shows in Figure 2 might also caused by various color value contain in each image while the compression result using NMF is always 40000 bytes.

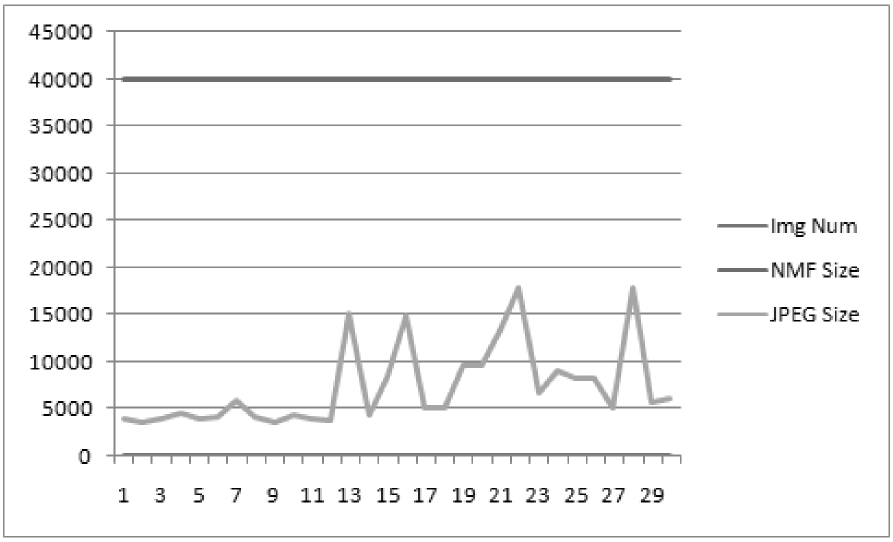

Figure 3. Compression Result of 30 Red Intensity Dominant Images Using JPEG and NMF

Figure 3 shows the compressed red intensity dominant image using JPEG resulted in a various size and the compression resulted using NMF is always 40000 bytes.

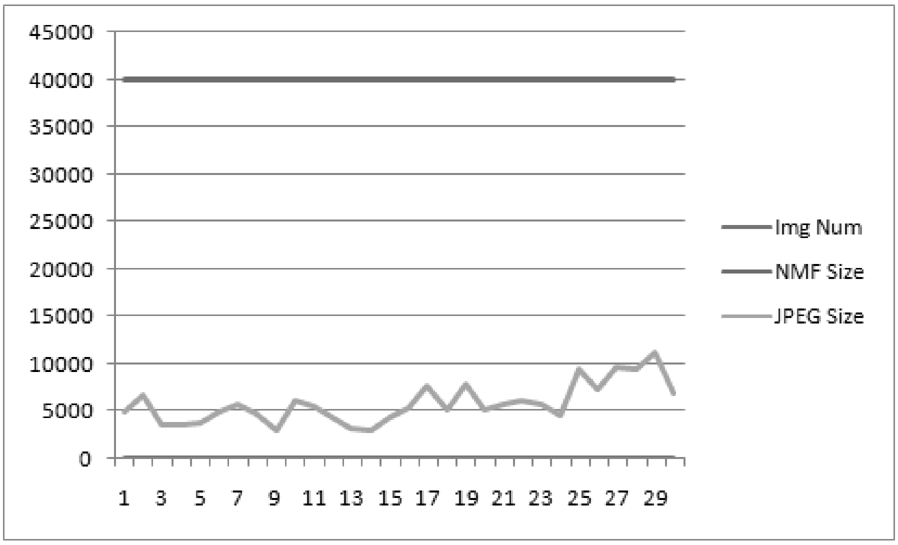

Figure 4. Compression Result of 30 Green Intensity Dominant Images Using JPEG and NMF

Figure 4 shows the compressed green intensity dominant image using JPEG also resulted in a various size and the compression result using NMF is always 40000 bytes.

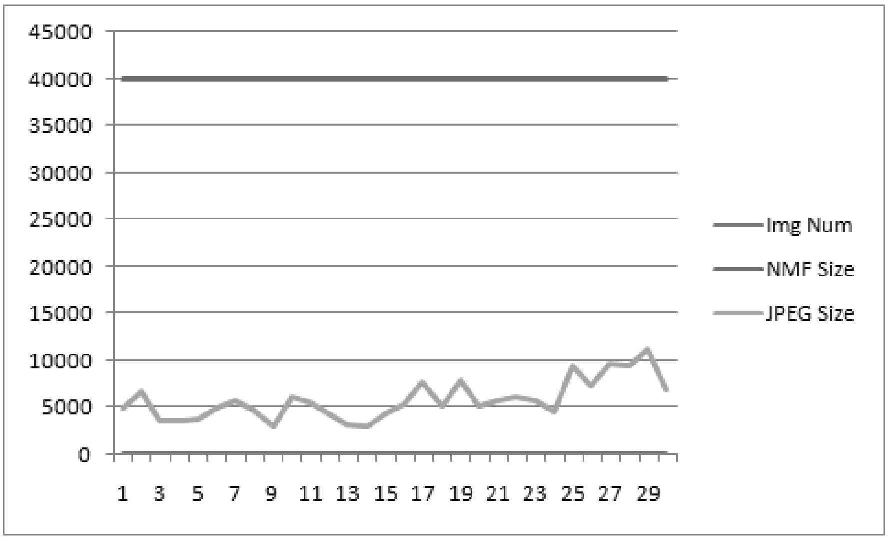

Figure 5. Compression Result of 30 Blue Intensity Dominant Images Using JPEG and NMF

Figure 5 shows the compressed blue intensity dominant image using JPEG resulted in a various size and the compression result using NMF is always 40000 bytes.

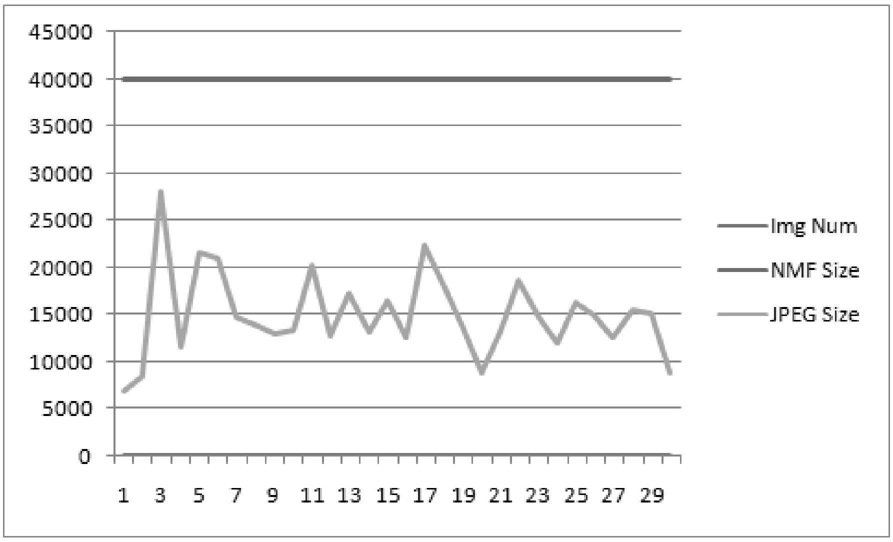

Figure 6. Compression Result of 30 Various Colors Images Using JPEG and NMF

Figure 6 shows that image with many different color value almost contains no redundant information causing bigger size of compressed image using JPEG while the NMF is still resulted in 40000 bytes.

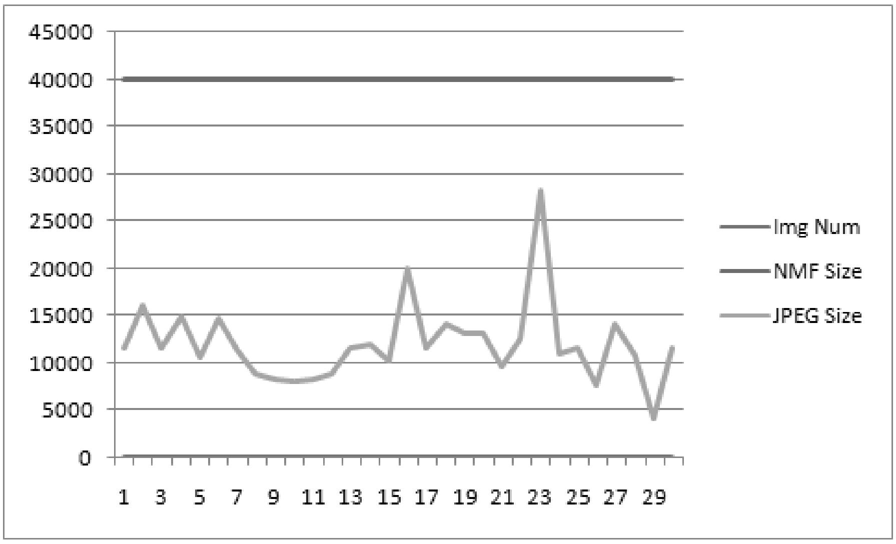

Figure 7. Compression Result of 30 Less Colors Images using JPEG and NMF

Figure 7 shows that compressed less colors image using JPEG is also resulted in a various size while the compression result using NMF is always 40000 bytes.

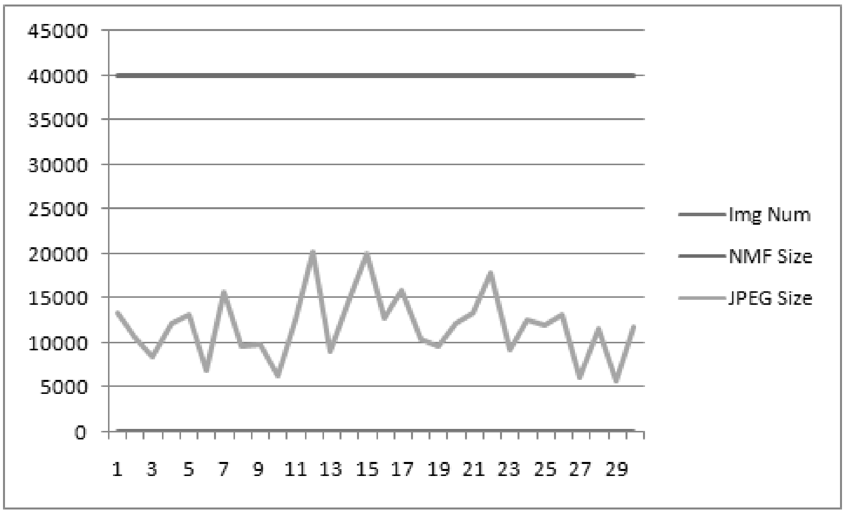

Figure 8. Compression Result of Grayscale Images Using JPEG and NMF

Figure 8 shows that compressed grayscale images using JPEG result in various size while the NMF is still resulted in 40000 bytes.

Table 14 shows comparison of average image size after compression between the NMF algorithm and Winzip for 8 image categories where each categories to compare contains 30 images to test.

Table 14. Compression Result Using Sample of W Matrix in Table 9 (1 Kilobytes = 1024 bytes)

Image Category	Average image size after compressed using (bytes)		Average compression efficiency (%)	
	JPEG	NMF	JPEG	NMF
Low colors	11939.430	40000	90.05	66.667
High colors	12503.630	40000	89.58	66.667
Red intensity	7278.567	40000	93.93	66.667
Green intensity	7832.267	40000	93.47	66.667
Blue intensity	5814.200	40000	95.15	66.667
Variation colors	14942.270	40000	87.55	66.667
Less colors	11939.433	40000	90.05	66.667
Grayscale	11871.730	40000	90.10	66.667

The size of the original image of 200x200 pixels is 120000 bytes because each pixel contains 3 bytes of the combination of red, green & blue color value (200x200x3 = 120000). Table 14 shows that image compression using sample of W matrix from table 9 for the different type of image with the same size always result in the same size of 40000 bytes. It is because of the efficiency of NMF compression is 66.67% because each pixel with size of 3 bytes can be compressed into 1 byte.

Figure 9 shows comparison of compression process time on 30 1600x1200 pixels 24 bit color images using the NMF and JPEG in milliseconds.

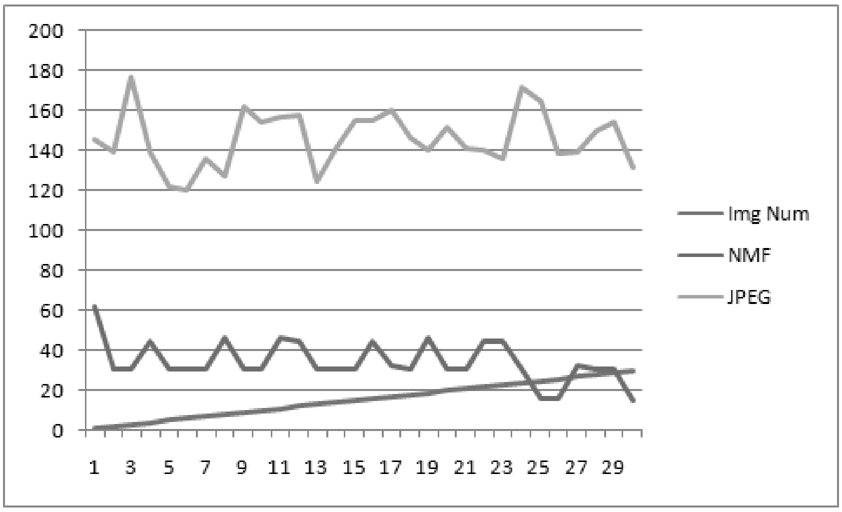

Figure 9. Comparison of Compression Process Time of JPEG and NMF

5. Conclusion

The implementations above prove the following conclusion.

1. As shown in table 12, determinant of W matrix greatly affected image compression result. The greater value of the W determinant is, and the smaller compression result can be gained.
2. W matrix which element value is gained randomly might have less value of the W determinant. The W^{-1} matrix gain from W matrix with less value of determinant can result in a higher matrix value. Therefore, multiplying the original image V matrix with W^{-1} matrix can result in bigger compressed image H matrix value.

3. As show in figure 9, compressed size of JPEG is smaller than NMF while the compression process time of NMF is faster than JPEG.

References

[1] J.H. Pujar and M.L. Kadlaskar, "A New Lossless Method of Image Compression and Decomposition Using Huffman Coding Techniques", Journal of Theoretical and Applied Information Technology, **(2010)**, p. 6.
[2] Shindu and Rajkamal , "Images and Its Compression Techniques – A Review", International Journal of Recent Trends in Engineering, vol 2, no. 4, **(2009)**, pp. 1-5.
I. Buciu, "Non-negative Matrix Factorization, A New Tool for Feature Extraction: Theory and Application", International Journal of Computers, Communications & Control, **(2008)**, p. 8.
[3] Z. Yuan and E. Oja, "Projective Nonnegative Matrix Factorization for Image Compression and Feature Extraction", Neural Networks Research Centre, **(2005)**.
[4] Z. Yang, Z. Yuan, and J. Laaksonen, "Projective Non-Negative Matrix Factorization With Applications To Facial Image Processing", International Journal of Pattern Recognition and Artificial Intelligence, vol. 21, no. 8, **(2007)**, pp. 1353–1362.
[5] S. Inuganti and V. Gampala, "Image compression using Constrained Non-Negative Matrix Factorization", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 10, **(2013)**, pp. 498-503.
[6] D. Gupta, S. Pardeep, Nivedita , and S. Sharma, "A Comparative Study of Image Compression Between Singular Value Decomposition, Block Truncating Coding, Discrete Cosine Transform and Wavelet", International Journal of Computer Science a 100 nd Network Security, **(2012)**, p. 7.
[7] S. Dhawan, "A Review of Image Compression and Comparison of its Algorithm", IJECT vol. 2, no. 1, **(2011)**, pp. 1-5.
[8] D. D. Lee and H. S. Seung, "Algorithms for Non-negative Matrix Factorization", **(2000)**, pp. 1-7.

Authors

Robin is a lecturer at bachelor degree in STMIK Mikroskil. He received under graduated degree in information technology from Faculty of Information Technology, Mikroskil at 1999. He received master degree in information technology from Faculty of Information Technology, Binus University 2014. His interest research are computer science and image compression

Suharjito is the Head of Information Technology Department in Binus Graduate Program of Bina Nusantara University. He received under graduated degree in mathematics from The Faculty of Mathematics and Natural Science, Gadjah Mada University, Yogyakarta, Indonesia in 1994. He received master degree in information technology engineering from Sepuluh November Institute of Technology, Surabaya, Indonesia in 2000. He received the PhD degree in system engineering from the Bogor Agricultural University (IPB), Bogor, Indonesia in 2011. His interest research are intelligent system, Fuzzy system, image processing, image compression and software project estimation supply chain risk management

