Image Compression Based on SVD and MPQ-BTC

Nidhal K. El Abbadi[1], Adil Al Rammahi[2], Dheiaa Shakir Redha[2], Mohammed Abdul-Hameed[2]
[1]Computer Science Department, University of Kufa, Najaf, Iraq
[2]Department of Mathematics, University of Kufa, Najaf, Iraq

ABSTRACT

In this paper we will provide a new way of images compression based on two mathematic concepts, these two concepts are Singular Value Decomposition (SVD), and Block Truncation Coding. The input image either is in JPEG format or in BMP format, and the current way suitable for both color and gray scale images. The input image will be compressed first by reducing the image matrix rank, which achieved by using the SVD process, and then the result matrix compressed by using BTC. The proposed algorithm gives further lossless compression to JPEG compression process and the rate of compression reach more than 99%.

Keywords: SVD, BTC, image processing, compression, MPQ-BTC.

1. INTRODUCTION

With the growth of using multimedia such as images, video, graphs, and audio in different application and internet, large amount of data are transmitted through network. Using of these data increase the requirements of storage capacity and transmission bandwidth.
Image compression means minimizing the size of files in bytes without degrading the quality of the image to an unacceptable level. The main task of compression is to remove the redundancy (Nivedita Sonika Jindal. 2012).
Image compression deals with the problem of reducing the amount of data required to represent a digital image. Compression is achieved by the removal of three basic data redundancies: 1) coding redundancy, which is present when less than optimal; 2) Inter-pixel redundancy, which results from correlations between the pixels; 3) Psycho-visual redundancies, which is due to data that is ignored by the human visual (Doaa Mohammed, Abou Chadi, 2011).

There are two types of image compression namely lossy image compression and lossless image compression, the lossless compression mean the possibility to reconstruct the origin image and message without losing any data, while the lossy compression stand for the idea of losing some of information and data after decompressed the image.

Lossless data compression is ideal for text (D.Harihara Santoch, 2011). Lossy compression yields good compression ratio comparing with lossless compression while the lossless compression gives good quality of compressed images. JPEG and Block Truncation Coding (BTC) is a lossy image compression techniques. It is a simple technique which involves less computational complexity. BTC is a recent technique used for compression of monochrome image data. It is one-bit adaptive moment preserving quantizer that preserves certain statistical moments of small blocks of the input image in the quantized output.
There are some papers suggested to use the SVD with other compression techniques such as.
(Awwal et. al. 2014) presented new compression technique using SVD and the wavelet difference reduction WDR. The WDR used for further reduction. This technique tested with other techniques such as WDR and JPEG 2000 and gives better result than these techniques. Using WDR with SVD enhance the PSNR and compression ratio.
(Adiwijaya et. al. 2013) suggested new compression method based on Wavelet- SVD, which used a graph coloring technique in the quantization process. This

technique work well and enhanced the PSNR and compression ratio. The generated compression ratio by this paper ranged between 50–60%, while the average PSNR ranged between 40–80.

2. SINGULAR VALUE DECOMPOSITION (SVD)

SVD is robust and reliable orthogonal matrix decomposition method. Due to SVD conceptual and stability reasons, it becomes more and more popular in signal processing area. SVD is an attractive algebraic transform for image processing. SVD has prominent properties in imaging. Although some SVD properties are fully utilized in image processing, others still needs more investigation and contributed to.
A key property of SVD is its relation to the rank of a matrix and its ability to approximate matrices of a given rank. Digital images are often represented by *low ra*nk matrices and, therefore, able to be described by a sum of a relatively small set of eigen images. This concept rises the manipulating of the signal as two distinct subspaces (Rowayda A. Sadek, 2012).

The use of Singular Value Decomposition (SVD) in image compression has been widely studied (D.Harihara Santoch,2011& Rowayda A. Sadek 2012), if the image when considered as a matrix, has low rank, or can be approximated sufficiently well by a matrix of low rank, then SVD can be used to find this approximation, and further this low rank approximation can be represented much more compactly than the original image.
Every real matrix $A_{m\times n}$ can be decompose into a product of three matrices $A=USV^T$, where U and V are orthonormal matrices, where $U^TU=I$, $V^TV=I$, and S=diagonal $(s_1,s_2,\ldots)$. The diagonal entries of S are called the singular values of A and the columns of U are calls the right singular vectors of A, the columns of V are calls the left singular vectors of A. This decomposition is known as the SVD of A. SVD is one of the most useful tools of linear algebra with several applications to multimedia including image compression (Aleksandr et al. 2006)

The $m_{\times}n$ matrix A can be written as the sum of rank-one matrices.

$$A=\sum_{i=1}^{r} \sigma_i u_i v_i^T$$

Where r is the rank of A, and u_i and v_i are the ith columns of U and V, respectively (Angel R. Pineda and Teav Vuthy, 2010)

This property is known as the low-rank approximation property of the (SVD). We get the best least squares approximation to A of rank $K \leq r$ by keeping only the first k terms of a matrix A and the other terms left are ignored (Angel R. Pineda and Teav Vuthy, 2010)

The rank of matrix A is equal to the number of its nonzero singular values". In many applications, the singular values of a matrix decrease quickly with increasing rank. This propriety allows us to reduce the noise or compress the matrix data by eliminating the small singular values or the higher ranks.

When an image is SVD transformed, it is not compressed, but the data take a form in which the first singular value has a great amount of the image information. With this, we can use only a few singular values to represent the image with little differences from the original.

The rank of A is the maximal number of linearly independent column vectors in A, i.e. the maximal number of linearly independent vectors among {a1; a2; : : : ; an}. If A = 0, then the rank of A is 0.
SVD can offer low rank approximation which could be optimal sub rank approximations by considering the largest singular value that pack most of the energy contained in the image (Meftah M. Almrabet et al. 2009) (S.Vimala, et al. 2011).
SVD shows how a matrix may be represented by a sum of rank-one matrices. The approximation a matrix A can be represented as truncated matrix Ak which has a specific rank k.

3. DIMENSION REDUCTION

One way to compress the image A is to approximate A by a matrix of smaller rank. If k < r then the closest approximation to A, (rank A = r)

$$A\approx A_K=S_1U_1V_1^T+S_2U_2V_2^T+\ldots.+S_k\,U_kV_k^T$$

Since the singular values are arranged in descending order, when k goes nearly to r, S_{k+1} is very small (about near zero).Then $S_{k+1}U_kV_k^T$ is very small and in this case the effects of it on the matrix is trivial.

Thus $A=A_K$ ($k\leq r$) where r is a rank of matrix A

For example when we take Lena image, the dimensions of this matrix image is (512×512), therefore rank of this matrix equals to 512, since the matrix can be written as a linear combination based on the singular values, the singular values are in descending order and not all with zero values. The first borders are significant in the

construction of the matrix. **Fig.1**. shows how the images resolution affected with reducing the image rank.

Fig.1. Lena image with different rank

4. BLOCK TRUNCATION CODING

Block Truncation Coding (BTC) works by dividing the image into small sub blocks and then reducing the number of gray levels within each block. This reduction is performed by a quantizer that adapts to the local image statistics. The basic form of BTC divides the whole image into ***N*** blocks and codes each block using a two-level quantizer (a, and b).

For each block the pixel values will be changed to (0, or 1) depending on the mean value ($\overline{X}$), the pixels value greater than or equal to mean replaced by "1", while the pixel values less than the mean ($\overline{X}$) replaced by "0". The collection of 1s and 0s for each block is called a bit plane, B. In BTC, two Statistical moments (a) and (b) are computed using the relations (1) and (2) and are preserved along with the bit plane for reconstructing the image. The compressed image is transmitted or stored as a set {B, a, b}

$$a=\overline{X}-\sigma\sqrt{\frac{n^+}{n^-}} \qquad (1)$$

$$b=\overline{X}+\sigma\sqrt{\frac{n^-}{n^+}} \qquad (2)$$

Where, n^+ is the number of pixel values greater than or equal to $\overline{X}$ and n^- is the number of pixels whose gray levels are less than the mean $\overline{X}$.

While in the reconstructing the image, the ***0*** in the bit plane is replaced by **a,** and the 1 in the bit plane is replaced by b.

5. MOMENT PRESERVING QUANTIZER TECHNIQUES *MPQ-BTC*

The difference between BTC and MPQ-BTC is in the way of writing relations (1), and (2). Terms of mathematical operation are conducted in the same style, but the relations (1) and (2) are rewritten in the following form

$$a=\overline{X}-\sigma\sqrt{\frac{n^+}{n^-}} \qquad (3)$$

$$b=\overline{X}+\sigma\sqrt{\frac{n^+}{n^-}} \qquad (4)$$

6. METHODS AND MATERIALS

The SVD and MPQ-BTC algorithm combined together to compress the image in this paper, according to following steps

The input image which has to be compressed is in the JPEG format or may be in the BMP format and compress with JPEG. This image stored as array of integers.

SVD will be computing for the input image (A = USV), and then compress the image by reducing the image rank, we choose the best k≤ r where r is the rank of image matrix. Image compressed by rebuild the image matrix ($A = USV^T$) with new reduced rank.

The result image from the SVD compression process will be divided to non overlapping square blocks; For each block we measuring the mean and the standard deviation, and then determine the values of (a, and b) as in relations (3, and 4). Each pixel value greater than the mean, will be replaced by the value (1) while the pixels value less than the mean will be replaced by the value (0), as in the following example for block 4x4 (mean for this example equal 142.18).

$$A=\begin{bmatrix}172 & 169 & 155 & 137\\ 169 & 160 & 144 & 122\\ 163 & 150 & 132 & 108\\ 151 & 136 & 114 & 93\end{bmatrix} \rightarrow \begin{bmatrix}1 & 1 & 1 & 0\\ 1 & 1 & 1 & 0\\ 1 & 1 & 0 & 0\\ 1 & 0 & 0 & 0\end{bmatrix}$$

a=115, b=168

The bit plain matrix B (x, x) which is the same size of the block (where x is the number of rows and columns

for square matrix) will be reconstructed to new size depending on the origin block size (the result matrix will be rectangle matrix with (x/2) rows and (2x) columns.

2×2 block size will be reconstructed to size 1×4. And 4×4 block size will be reconstructed to size 2×8, and so on. So the above bit plane matrix becomes:

$$C=\begin{bmatrix}1 & 1 & 1 & 0 & 1 & 1 & 0 & 0\\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0\end{bmatrix}$$

At this case each 8 bits will be converted to one byte or to the corresponding decimal value inserted in the one dimension matrix D, which also included the value of (a, and b) at the last two rows.

$$D = \begin{bmatrix}236\\232\\168\\115\end{bmatrix}$$

This process repeats for all the image blocks.

6.1 DECOMPRESSION PROCESS

The decompression process is the inverse of compression process:

1. The compressed image will be divided into one dimension blocks according to the following relation

((n + 2) / 2) + 1. (Where n is the block size during the compression process).

2. Convert the values of each byte for each block (except the last two rows) to the equivalent binary numbers.

3. Reconstruct the two dimension square matrix according to block size (n).

4. Each 0 in the resulted binary matrix will be replaced by the value a, while the value 1 replaced by b.

For the above example the recompressed matrix will be

$$B = \begin{bmatrix}168 & 168 & 168 & 115\\ 168 & 168 & 168 & 115\\ 168 & 168 & 115 & 115\\ 168 & 115 & 115 & 115\end{bmatrix}$$

7. RESULTS

1. The proposed algorithm tested with many images some of them showed in **Fig. 2,** it is clear the decompression images are perceptually similar to origin images, note the rank in this test was 100, and the block size was 4x4.

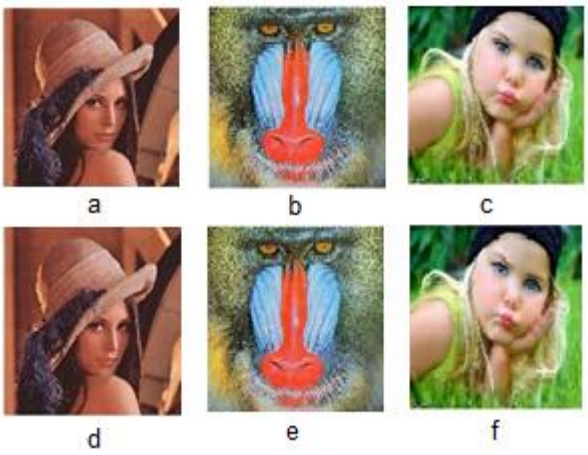

Fig. 2: (a, b, c) images before compression. (c, d, e) images after decompression.

2. The compression ratio and PSNR of images varied when compression images by reducing the image rank in the SVD process as showed in **Fig. 3,** and **Fig. 4.**

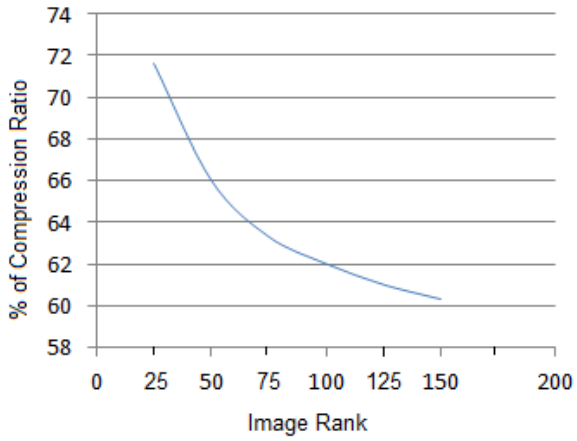

Fig. 3: Relation between image rank and compression ratio.

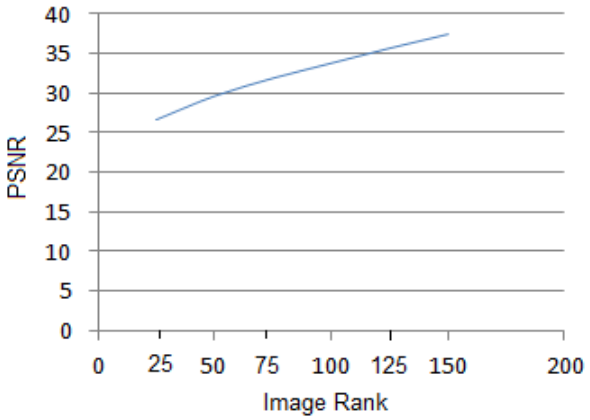

Fig. 4: Variation of PSNR with image rank.

3. Compression ratio and PSNR for images when applying only the MPQ-BTC method showed in **Fig. 5.**

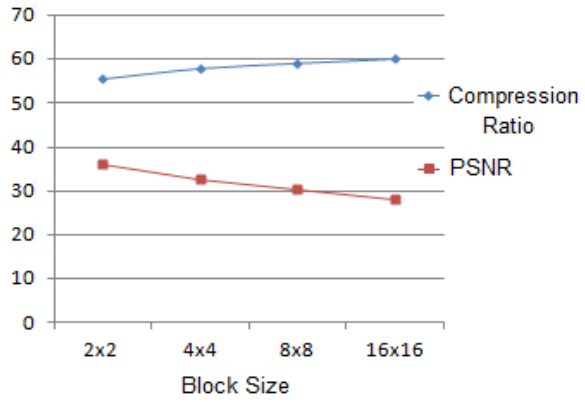

Fig. 5: showed the compression ratio and PSNR when compression images with MPQ-BTC algorithm.

4. Appling the proposed algorithm on (JPEG) gray scale image give the compression ratio begin with 95.4% according to block size, as shown in **Fig. 6**, the compression and decompression time shows in **Fig. 7,** while the PSNR, bit per pixel (bpp), and mean square error (MSE) shows in **Table 1**.

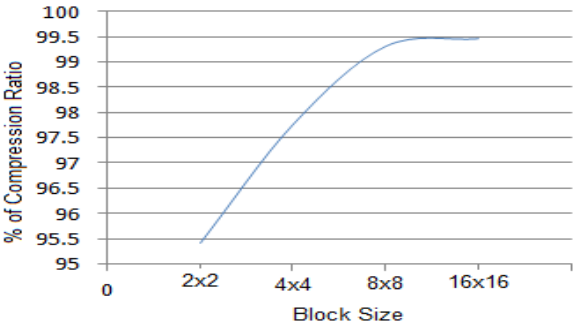

Fig. 6: Compression ratio for gray scale images.

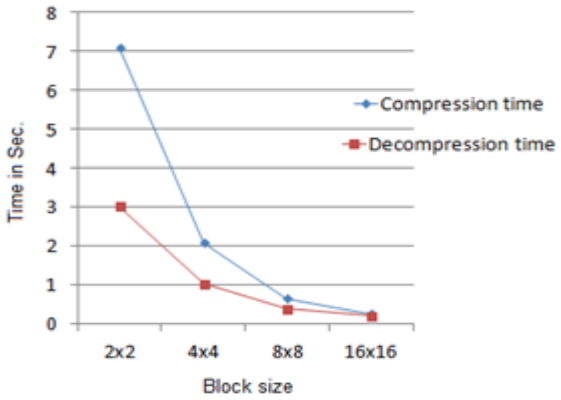

Fig. 7: shows the compression and decompression time for gray images.

Table 1. showes some of compressed image proprties with differen block size.

Block size	bpp	PSNR			Average of PSNR	MSE
		R	G	B		
2x2	0.29	35.57	36.38	35.48	35.81	2.81e-4
4x4	0.21	33.87	34.71	34.14	34.24	3.77e-4
8x8	0.13	31.77	32.88	32.72	32.46	5.70e-4
16x16	0.06	29.24	30.57	30.93	30.25	9.57e-4

5. The algorithm applied for color images and the compression ratio showed in **Fig. 8**, while the PSNR showed in **Fig. 9**.

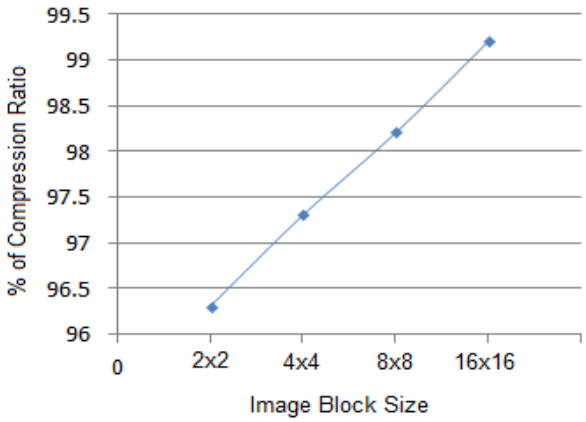

Fig. 8: compression ratio for color images compressed with proposed algorithm.

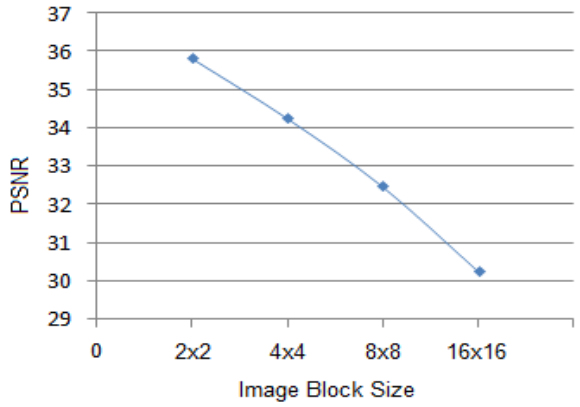

Fig. 9: PSNR for decompressed color images when compressed by proposed algorithm.

6. Changing the rank of image during the SVD process did not affect the compression ratio too much when applying the proposed copmression algorithm, it is almost the same compression ratio as shown in **Fig. 10**, but it was little bit changing in PSNR as shown in **Fig. 11**.

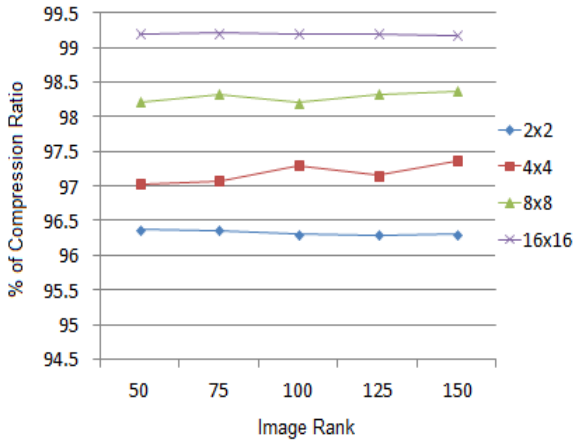

Fig. 10: compression ratio with different image rank when compressed images with proposed algorithm.

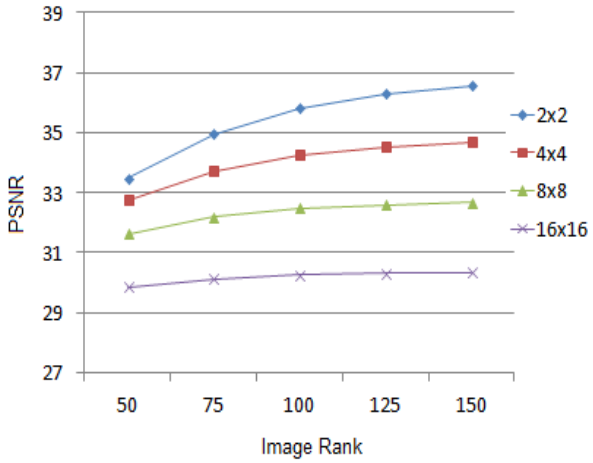

Fig. 11: PSNR for decompressed images when compressed with proposed algorithm using different image rank.

7. We test the proposed algorithm with bmp images, these images compressed with three steps, first we compressed it with JPEG algorithm, and then compressed with SVD (rank equal 100), finally compressed with MPQ-BCT for different block size. The compression ratio were very high as shown in **Fig. 12**, also we test the compression and decompression time as in **Fig. 13**, and PSNR as in **Fig. 14**.

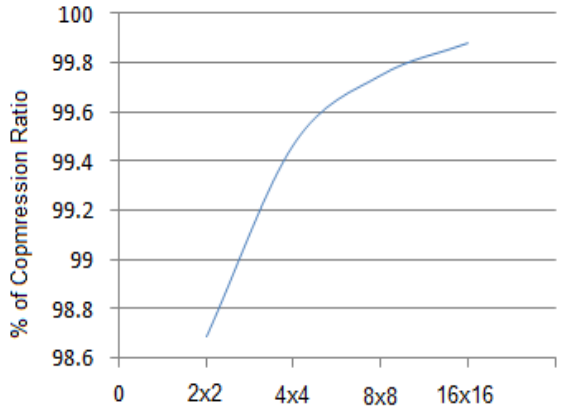

Fig. 12: total compression ratio when the input images was bmp images (rank=100).

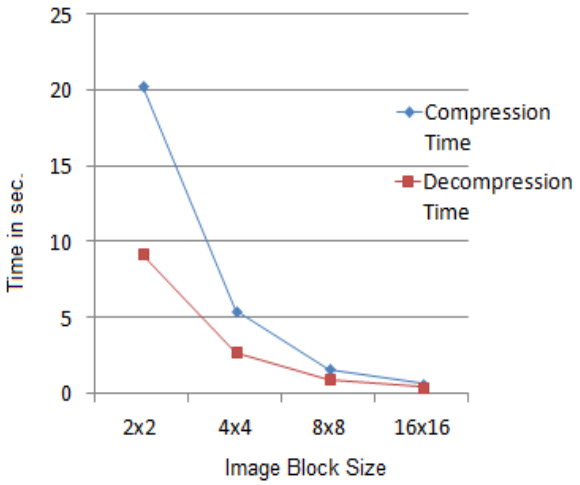

Fig. 13: compression and decompression time for bmp images.

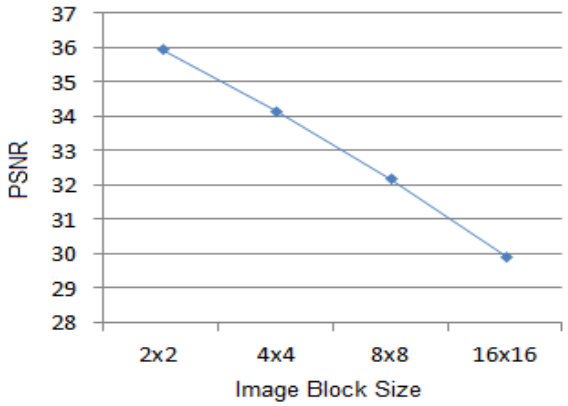

Fig. 14: PSNR for decompressed bmp images.

8. The proposed algorithm compared with different techniques and gives very good results, as shown in **Table 2** (Afshan Mulla et al. 2013) (Sachin Dhawan. 2011).

Table 2: comparing proposed algorithm with different compression technique.

Algorithms		**Compression Ratio**	**PSNR**
DWT		>>35	34.66
K-Means		<25	27.36
3D Spiral JPEG		<=60	31.73
Wavelet		>>32	32.47
JPEG		<=50	29.64
VQ		<32	N/A
Fractel		>=16	N/A
Proposed Algorithm	Image not already compressed	>>150	34.16
	Compressed image with JPEG (further compression)	>25	34.24

8. CONCLUSIONS

We introduced in this paper new compression algorithm based on combine both SVD and MPQ for the compressed JPEG images. Using these techniques further compression is obtained in addition to compression achieved by JPEG format. The contribution of this paper is higher compression ratio is achieved due to additional compression; without compromising much on the quality of the image.
In the current proposed algorithm, we advice to use image rank = 100 for compression with SVD. Also we suggest using 4x4 block size for the MPQ-BCT compression technique.
The proposed algorithm compared with other algorithm and give very promised results.

REFERENCES

Adiwijaya, M Maharani, B K Dewi, F A Yulianto and B Purnama, Adiwijaya, 2013, Digital Image Compression using Graph Coloring Quantization Based on Wavelet-SVD, Journal of Physics: Conference Series 423, doi:10.1088/1742-6596/423/1/012019.

Afshan Mulla, Namrata Gunjikar, Radhika Naik, 2013, Comparison of different Image Compression Techniques, International Journal of Computer Applications, vol. 70, issue 28, pp. 7-12, DOI: 10.5120/12253-7934.

Aleksandr Shnayderman, Alexander Gusev, and Ahmet M. Eskicioglu, 2006, An SVD-Based Gray scale Image Quality Measure for Local and Global Assessment, IEEE Transactions on Image Processing journal, Volume 15 Issue 2, Page 422-429, doi:10.1109/TIP.2005.860605.

Angel R. Pineda and Teav Vuthy, 2010, Using the Singular Value Decomposition (SVD) for Image Compression, thesis introduced to Royal University of Phnom Penh.

Awwal Mohammed Rufai, Gholamreza Anbarjafari, Hasan Demirel, 2014, Lossy image compression using singular value decomposition and wavelet difference reduction, Digital Signal Processing, Volume 24, Pages 117–123, doi.org/10.1016/j.dsp.2013.09.008.

Doaa Mohammed, Abou Chadi, 2011, Image Compression Using Block Truncation Coding,

journal in selected areas of telecommunication (JSAT).

Meftah M. Almrabet , Amer R. Zerek, Allaou Chaoui and Ali A. Akash, 2009 , Image Compression using BlocK Truncation Coding, IJ-STA, Vol 3, No.2, pp.1046-1053.

Nivedita Sonika Jindal, 2012, Performance Analysis of SVD and SPIHT Algorithm for Image Compression Application, International Journal of Advanced Research in Computer Science and Software Engineering**,** Volume 2, Issue 2.

Rowayda A. Sadek. 2012, SVD Based Image Processing Applications: State of The Art, Contributions and Research Challenges, *(IJACSA)* International Journal of Advanced Computer Science and Applications, Vol. 3, No. 7.

S.Vimala, P.Uma, B.Abidha, 2011, Improved Adaptive Block Truncation Coding for Image Compression, International Journal of Computer Applications, Volume 19–No.7.

Sachin Dhawan, 2011, A Review of Image Compression and Comparison of its Algorithms, IJECT VOL. 2, ISSUE 1, pp.22-26.