A Hybrid SVD Method Using Interpolation Algorithms for Image Compression

J. Izadian , A. Hosaini *Jalal_Izadian@yahoo.com*
Department of Mathematics, Faculty of Sciences,
Mashhad Branch, Islamic Azad University,
Mashhad, Iran

M. Jalili *Jalili.maryam@yahoo.com*
Department of Mathematics,
Neyshabur Branch, Islamic Azad University,
Neyshabur, Iran

Abstract

In this paper the standard SVD method is used for image processing and is combined with some interpolation methods as linear and quadratic interpolation for reconstruction of compressed image.The main idea of the proposed method is to select a particular submatrix of main image matrix and compress it with SVD method, then reconstruct an approximation of original image by interpolation method. The numerical experiments illustrate the performance and efficiency of proposed methods.

Keywords: SVD Method, Interpolation Method, Image Reconstruction, Compressing Image, Lossless Compression, Lossy Compression.

1. INTRODUCTION

Image compression techniques plays an important role in transmision and storage of image information in computer science and related domains.The goal of image compression is to obtain a representation that minimizes the bits volume while still maintaining the important meaning and the intrinsic structure of the original image. Image compression techniques can be classified into two groups. Lossless compression and lossy compression.

In lossless compression the reconstructed image is identical to the original one and deduces a low compression ratio while lossy compression methods allow a loss in the actual image data. So the original image cannot be created excactly from the compressed image. But these methods deduce high compression ratio. There is many ways to compress depending on the application field. One popular method for compressing an image is the wavelet teqnique [1]. Wavelet functions form an orthonormal basis on which it is possible to project each data set. Another important method is the Singular Value Decomposition (SVD). This technique is based on the factoriziation of the real matrix of image in three matrices that can be used to reconstruct the main image or an approximation of it, for more information see [2,3,4].

In this paper the SVD method is combined with two interpolation processes, that are linear 2D interpolation (triangular interpolation), and bilinear interpolation, to decrease the volume of transfered image . This paper is orgonized as follows. Section 2 is specified to describe the basic concepts of proposed method. Section 3 presents the numerical experiments . Finally section 4 concludes the paper with discussion.

2. THE BASIC CONCEPTS OF METHOD

Suppose that a $m \times n$ pixel gray-scale image is given. Each pixel having some level of black and white given by some integers that can be selected between 0 and 255 or a real number between zero and one. In the case of integer, each integer requires approximately one byte to store. Then the resulting image has approximately $m \times n$ bytes volume. If the image is coloured in (RGB) system the image contains three $m \times n$ matrices for red, green, and blue colours, then $3 \times m \times n$ pixels image. The Singular Value Decomposition (SVD) for a gray-scale image for compressing $m \times n$ pixels image matrix is based on the following decomposition of A,

$$A = USV^T.$$

Actually, recalling from advanced linear algebra one has the following theorem, that is demonstrated in advanced linear algebra (for example see [5,6]).

Theorem 2.1 *Any nonzero real $m \times n$ matrix A with rank $r > 0$, there are an orthogonal $m \times m$ matrix U, and an orthogonal $n \times n$ matrix V such that $U^T AV = S$ is an $m \times n$ "diagonal matrix" of the following form*

$$S = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix},$$

$$D = diag(\sigma_1, ..., \sigma_r),\ \sigma_1 \geq \sigma_2 \geq ... \sigma_r > 0,$$

where $\sigma_1, \sigma_2, ..., \sigma_r$ are non-zero sigular values of A. This factorization is called the Singular Value Decomposition (SVD) of A.

By using above theorem and assuming that

$$U = [u_1\ u_2\ ...u_r\ u_{r+1}...u_m],\quad i = 1,2,...,m,$$

where $u_i, i = 1,...,m$ are the column of U, by considering the orthogonality of U, one has

$$u_i^T u_j = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j. \end{cases}$$

Similarly for orthogonal $n \times n$ matrix V, by considering.

$$V^T = \begin{bmatrix} v_1^T \\ v_2^T \\ \cdots \\ v_n^T \end{bmatrix},$$

where v_i are for $i = 1,...,n$ are columns of V, it yields

$$v_i^T v_j = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j. \end{cases}$$

Here, S is $m \times n$ diagonal matrix with singular values of A on the diagonal of D in rectangular matrix S. The matrix S can be presented by the following matrix:

$$S = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 & 0 \\ 0 & \sigma_2 & & & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & \dots & \sigma_r & 0 \\ 0 & \dots & \dots & 0 & 0 \end{bmatrix}$$

For $i = 1,2,...,n$, σ_i are singular values of matrix A. It is proved in Theorem(2.1) that $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r > 0$ and $\sigma_{r+1} = \sigma_{r+2} = ... = \sigma_n = 0$. For $i = 1,2,...,m$, σ_i are called singular values of matrix A^T .The vector v_i for $i=1,2,...,n$ are called the right singular vector of A , and u_i for $i=1,2,...,m$. are the left singular vectors of A (see[4,5]). then

$$A = USV^T = [u_1 \quad u_2 \quad \dots \quad u_{m-1} \quad u_m]S\begin{bmatrix} v_1 \\ v_2 \\ . \\ . \\ . \\ v_n \end{bmatrix} = u_1\sigma_1 v_1^T + u_2\sigma_2 v_2^T + \dots + u_r\sigma_r v_r^T .$$

When compressing the image, the sum is not performed to the very last . The singular values with small enough values are dropped. A matrix of rank k is obtained by truncating these sums after the first k terms, denoted by

$$A_k = \sigma_1 u_1 v_1^T + \sigma_1 u_2 v_2^T + \dots + \sigma_k u_k v_k^T , \qquad (1)$$

is chosen as approximation of A . The total storage volume for A_k will be $k(m+n+1)$. This matrix is an approximation of A , that can be used as an approximation of compressed image A . For having a criterion for storage volume decrease, the compression ratio is defined as follows:

$$C_R = \frac{mn}{k(m+n+1)}.$$

We also use the Schur norm to measure the quality of obtained image A_k , called Mean Square Error (MSE), which is introduced by following expression:

$$MSE = \frac{1}{mn}\left\| A - A_K \right\|_2 = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(a_{ij} - a_{ij}^{(k)})^2.$$

where $A=(a_{ij})_{m\times n}$ and $A_k=(a_{ij}^{(k)})_{m\times n}$. An error value can be defined related to sum of σ_i by following relation:

$$e_k = 1 - \frac{\sum_{i=1}^{k}\sigma_i}{\sum_{i=1}^{r}\sigma_i},$$

A typical choice of k is so that the storage space required for A_k will be less than $\frac{1}{5}$. For having an superior compression ratio, one can utilize a submatrix of A for small k and use an interpolation process to reconstruct A_k .

Consider m and n are even, and $\widetilde{A}$ is a submatrix of order $\frac{m}{2}\times\frac{n}{2}$ of A , by omitting the element of A which are situated on rows or column of odd numbers. This matrix is approximated by SVD method with a suitable value of k . Then an interpolation method is used to reconstruct an approximation of A_k , that can be chosen as approximation of A_k as obtained image. In linear case, consider the chosen submatrix of A as:

$$\widetilde{A}_k = (\widetilde{a}_{ij}^{(k)})_{\frac{m}{2}\times\frac{n}{2}},$$

where m and n are even numbers and the elements of $_k$ can be determined from $A_k = (a_{ij}^{(k)})_{m\times n}$ by the following relations:

$$(\tilde{a}_{i,j}^{(k)}) = a_{2(i-1)+1,2(j-1)+1}^{(k)}, \quad (\tilde{a}_{i,j+1}^{(k)}) = a_{2(i-1)+1,2(j-1)+3}^{(k)},$$

$$(\tilde{a}_{i+1,j}^{(k)}) = a_{2(i-1)+3,2(j-1)+1}^{(k)}, \quad (\tilde{a}_{i+1,j+1}^{(k)}) = a_{2(i-1)+3,2(j-1)+3}^{(k)},$$

$$i = 1,2,\ldots,\frac{m}{2}, \ \ j = 1,2,\ldots,\frac{n}{2}.$$

The matrix $\tilde{A}_k$ is of order $\frac{m}{2}\times\frac{n}{2}$, therefore it has $\frac{3}{4}$ less element than A_k. This matrix is chosen as the compression of A that results a compression ratio given by

$$C_R = \frac{mn}{k(\frac{m}{2}+\frac{n}{2}+1)} = \frac{2mn}{k(m+n+2)},$$

Therefore, in this method the compression ratio is doubled in comparison with A_k. The matrix $\tilde{A}_k$ is supposed as a base for interpolating method to reconstruct an approximation of A_k. In the linear case, this matrix is calculated with the following equations:

$$A_k^*; \ \ A_k, \ A_k^* = (a_{ij}^*)_{m\times n},$$

$$a_{2(i-1)+1,2(j-1)+1}^* = \tilde{a}_{ij}^{(k)}, \quad a_{2(i-1)+3,2(j-1)+1}^* = \tilde{a}_{i+1,j}^{(k)},$$

$$a_{2(i-1)+1,2(j-1)+3}^* = \tilde{a}_{i,j+1}^{(k)}, \quad a_{2(i-1)+3,2(j-1)+3}^* = \tilde{a}_{i+1,j+1}^{(k)},$$

$$a_{2(i-1)+2,2(j-1)+1}^* = (\tilde{a}_{i+1,j}^{(k)} + \tilde{a}_{i,j}^{(k)})/2, \ \ a_{2(i-1)+1,2(j-1)+2}^* = (\tilde{a}_{i,j}^{(k)} + \tilde{a}_{i,j+1}^{(k)})/2,$$

$$a_{2(i-1)+3,2(j-1)+2}^* = (\tilde{a}_{i+1,j}^{(k)} + \tilde{a}_{i+1,j+1}^{(k)})/2, a_{2(i-1)+2,2(j-1)+3}^* = (\tilde{a}_{i,j+1}^{(k)} + \tilde{a}_{i+1,j+1}^{(k)})/2,$$

$$a_{2(i-1)+2,2(j-1)+2}^* = [\tilde{a}_{i+1,j}^{(k)} + \tilde{a}_{i,j+1}^{(k)} + \tilde{a}_{i,j}^{(k)} + \tilde{a}_{i+1,j+1}^{(k)}]/4.$$

In order to describe the hybrid SVD using bilinear interpolation, the following submatrix of A_k is considerd

$$\tilde{A}_k = (\tilde{a}_{ij}^{(k)})_{\frac{m}{3}\times\frac{n}{3}},$$

where A_k is considered a $m\times n$ matrix which m and n are the integers that are divisible by 3. Then $\tilde{A}_k$ can be defined as follows

$$A_k = (a_{ij}^{(k)})_{m\times n},$$

$$(\tilde{a}_{i,j}^{(k)}) = a_{3(i-1)+1,3(j-1)+1}^{(k)}, \quad (\tilde{a}_{i,j+1}^{(k)}) = a_{3(i-1)+1,3(j-1)+4}^{(k)},$$

$$(\tilde{a}_{i+1,j}^{(k)}) = a_{3(i-1)+4,3(j-1)+1}^{(k)}, \quad (\tilde{a}_{i+1,j+1}^{(k)}) = a_{3(i-1)+4,3(j-1)+4}^{(k)},$$

$$i = 1,2,\ldots,\frac{m}{3},\ \ j = 1,2,\ldots,\frac{n}{3}.$$

Since the matrix $\tilde{A}_k$ is a matrix of order $\frac{m}{3}\times\frac{n}{3}$, the compression ratio is obtained by

$$C_R = \frac{mn}{n(\frac{m}{3}+\frac{n}{3}+1)} = \frac{3mn}{k(m+n+3)},$$

Now for each 4×4 block of matrix A_k which is reduced to 2×2 block matrix $\tilde{A}_k$, the following bilinear interpolation is applied on $[0,1]^2$,

$$\tilde{P}^{(i,j)}(x,y) = \sum_{r=1}^{2}\sum_{s=1}^{2} L_{r,s}(x,y)\tilde{a}_{i+(r-1),j+s-1}$$

where $L_{r,s}$ is defined by the following equations:

$$L_{11}(x,y) = \frac{(x-1)}{(0-1)}\frac{(y-1)}{(0-1)},\quad L_{12}(x,y) = \frac{(x-1)}{(0-1)}\frac{(y-0)}{(1-0)},$$

$$L_{21}(x,y) = \frac{(x-0)}{(1-0)}\frac{(y-1)}{(0-1)},\quad L_{22}(x,y) = \frac{(x-0)}{(1-0)}\frac{(y-0)}{(1-0)}.$$

or

$$L_{11}(x,y) = (x-1)(y-1),\ L_{12}(x,y) = y(1-x),$$

$$L_{21}(x,y) = x(1-y),\ L_{22}(x,y) = xy.$$

The matrix A_k^* of order $m\times n$ which is an approximation of A_k is given as follows:

$$A_k^* = (a_{ij}^*)_{m\times n} \cong (a_{ij}^{(k)})_{m\times n},$$

where a_{ij}^* can be determined by the following formulae

$$a^*_{3(i-1)+1,3(j-1)+1} = \tilde{a}^{(k)}_{i,j},\quad a^*_{3(i-1)+4,3(j-1)+1} = \tilde{a}^{(k)}_{i+1,j},$$

$$a^*_{3(i-1)+1,3(j-1)+4} = \tilde{a}^{(k)}_{i,j+1},\quad a^*_{3(i-1)+4,3(j-1)+4} = \tilde{a}^{(k)}_{i+1,j+1},$$

$$a^*_{3(i-1)+2,3(j-1)+1} = \tilde{P}^{(i,j)}(\frac{1}{3},0) = \frac{2}{3}\tilde{a}^{(k)}_{ij} + \frac{1}{3}\tilde{a}^{(k)}_{i+1,j},$$

$$a^*_{3(i-1)+3,3(j-1)+1} = \tilde{P}^{(i,j)}(\frac{2}{3},0) = \frac{1}{3}\tilde{a}^{(k)}_{ij} + \frac{2}{3}\tilde{a}^{(k)}_{i+1,j},$$

$$a^*_{3(i-1)+1,3(j-1)+2} = \tilde{P}^{(i,j)}(0,\frac{1}{3}) = \frac{2}{3}\tilde{a}^{(k)}_{ij} + \frac{1}{3}\tilde{a}^{(k)}_{i,j+1},$$

$$a^*_{3(i-1)+1,3(j-1)+3} = \tilde{P}^{(i,j)}(0,\frac{2}{3}) = \frac{1}{3}\tilde{a}^{(k)}_{ij} + \frac{2}{3}\tilde{a}^{(k)}_{i,j+1},$$

$$a^{*}_{3(i-1)+4,3(j-1)+2} = \widetilde{P}^{(i,j)}(1,\frac{1}{3}) = +\frac{2}{3}\widetilde{a}^{(k)}_{i+1,j} + \frac{1}{3}\widetilde{a}^{(k)}_{i+1,j+1},$$

$$a^{*}_{3(i-1)+4,3(j-1)+3} = \widetilde{P}^{(i,j)}(1,\frac{2}{3}) = +\frac{1}{3}\widetilde{a}^{(k)}_{i+1,j} + \frac{2}{3}\widetilde{a}^{(k)}_{i+1,j+1},$$

$$a^{*}_{3(i-1)+2,3(j-1)+4} = \widetilde{P}^{(i,j)}(\frac{1}{3},1) = +\frac{2}{3}\widetilde{a}^{(k)}_{i,j+1} + \frac{1}{3}\widetilde{a}^{(k)}_{i+1,j+1},$$

$$a^{*}_{3(i-1)+3,3(j-1)+4} = \widetilde{P}^{(i,j)}(\frac{2}{3},1) = +\frac{1}{3}\widetilde{a}^{(k)}_{i,j+1} + \frac{2}{3}\widetilde{a}^{(k)}_{i+1,j+1},$$

$$a^{*}_{3(i-1)+2,3(j-1)+2} = \widetilde{P}^{(i,j)}(\frac{1}{3},\frac{1}{3}) = \frac{4}{9}\widetilde{a}^{(k)}_{ij} + \frac{2}{9}\widetilde{a}^{(k)}_{i,j+1} + \frac{2}{9}\widetilde{a}^{(k)}_{i+1,j} + \frac{1}{9}\widetilde{a}^{(k)}_{i+1,j+1},$$

$$a^{*}_{3(i-1)+2,3(j-1)+3} = \widetilde{P}^{(i,j)}(\frac{1}{3},\frac{2}{3}) = \frac{2}{9}\widetilde{a}^{(k)}_{ij} + \frac{4}{9}\widetilde{a}^{(k)}_{i,j+1} + \frac{1}{9}\widetilde{a}^{(k)}_{i+1,j} + \frac{2}{9}\widetilde{a}^{(k)}_{i+1,j+1},$$

$$a^{*}_{3(i-1)+3,3(j-1)+2} = \widetilde{P}^{(i,j)}(\frac{2}{3},\frac{1}{3}) = \frac{2}{9}\widetilde{a}^{(k)}_{ij} + \frac{1}{9}\widetilde{a}^{(k)}_{i,j+1} + \frac{4}{9}\widetilde{a}^{(k)}_{i+1,j} + \frac{2}{9}\widetilde{a}^{(k)}_{i+1,j+1},$$

$$a^{*}_{3(i-1)+3,3(j-1)+3} = \widetilde{P}^{(i,j)}(\frac{2}{3},\frac{2}{3}) = \frac{1}{9}\widetilde{a}^{(k)}_{ij} + \frac{2}{9}\widetilde{a}^{(k)}_{i,j+1} + \frac{2}{9}\widetilde{a}^{(k)}_{i+1,j} + \frac{4}{9}\widetilde{a}^{(k)}_{i+1,j+1},$$

In fact, one has

$$a^{*}_{3(i-1)+(1+r),3(j-1)+(1+s)} = \widetilde{P}^{(i,j)}(\frac{r}{3},\frac{s}{3}) \quad r,s = 0,1,2,3$$

Then, A^{*} is obtained as suitable approximation of A_k.

3. NUMERICAL EXPERIMENTS

In this section some test examples are presented. In these examples we compare visually and numerically the compressing. The reconstructed image and the compression ratio and, the mean square error (MSE) in each case are presented. In the tables, L_I , and BI , are used for linear and bilinear interpolation, respectively.

3.1 Example

In this example we consider a 144×300 pixels black and white image from MATLAB gallery. The initial and final images are given in Figure 1.(a), and (b), the results for simple SVD linear interpolation and bilinear interpolation are presented in Figures 2. and 3, respectively. In Table 3.1. numerical results for three methods are given.

$method$	k	e	CR	MSE	cpu time
SVD	9	0.4304	10.78658	1.2458×10^-4	8.7622
SVD	16	0.3384	6.0674	8.0855× 10^-5	8.6436
SVD	25	0.258	3.8831	5.5209× 10^-5	8.0967
SVD	36	0.1889	2.6966	3.9410 ×10^-5	9.4412
SVD,LI	9	0.3873	10.7143	2.5198 ×10^-4	8.1953
SVD,LI	16	0.2860	6.00268	1.6349 ×10^-4	8.4375
SVD,LI	25	0.1985	3.8571	1.1551 ×10^-4	8.7658
SVD,LI	36	0.1228	2.6786	8.1363 ×10^-5	11.4910
SVD,BI	9	0.3335	95.3642	4.001 ×10^-4	7.3140
SVD,BI	16	0.2308	53.6424	2.5966 ×10^-4	7.9940
SVD,BI	25	0.1579	34.3311	1.6774 ×10^-4	11.3992
SVD,BI	36	0.0454	23.8411	1.0217 ×10^-4	12.6754

TABLE 3.1: The results for black and white image.

[original] [final]

FIGURE 1: The results of SVD method for k=36.

[original] [final]

FIGURE 2: The results of SVD and linear interpolation method for k=36.

[original] [final]

FIGURE 3: The results of SVD and bilinear interpolation method for k=36.

3.2 Example

In this example a $3\times384\times512$ pixel colored image from MATLAB gallery is selected. The initial and final images are given in Figure 4. and the results for SVD linear interpolation , and bilinear interpolation are shown in Figure 4-5, respectively. The numerical results are presented in Table 3.2.

$method$	M	e	CR	MSE	$time$
SVD	9	0.4913	24.3538	5.9974×10^-5	25.9324
SVD	16	0.3969	13.6990	3.3733 ×10^-5	26.5473
SVD	25	0.32242	8.7674	1.9898 ×10^-5	26.3382
SVD	36	0.2685	6.0884	1.3075 ×10^-5	28.4017
SVD,LI	9	0.7508	24.2726	2.2499 ×10^-4	11.3439
SVD,LI	16	0.7158	13.6533	2.0673 ×10^-4	11.8093
SVD,LI	25	0.7084	8.7381	2.0631 ×10^-4	12.4191
SVD,LI	36	0.6945	6.0681	2.0611×10^-4	14.2736
SVD,BI	9	0.7370	217.0066	7.0313 ×10^-4	8.7288
SVD,BI	16	0.7001	122.0662	6.9817 ×10^-4	10.8634
SVD,BI	25	0.6922	78.1224	6.9801 ×10^-4	12.1687
SVD,BI	36	0.6535	54.2517	6.978 ×10^-4	13.1736

TABLE 3.2: The results for black and white.

[original] [final]

FIGURE 4: The results for SVD method for k=36.

[original] [final]

FIGURE 5: The results for SVD and linear interpolation method.

[original] [final]

FIGURE 6: The results for SVD and bilinear interpolation method.

3.3 Example

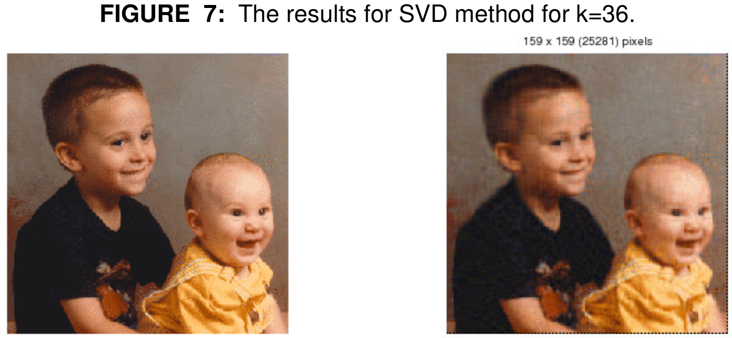
In this example an other colored image is chosen from MATLAB gallery which has $3\times318\times318$ pixels. The initial and final images are given in Figures 7-9. and the results for simple SVD linear interpolation, and bilinear interpolation are shown in the same Figures, respectively. The numerical results are presented in Table 3.3.

method	M	m_i	n_i	e	CR	MSE	*time*
SVD	9	3	3	0.4304	17.6389	8.8662× 10^-5	17.8987
SVD	16	4	4	0.3384	9.9212	4.4780 ×10^-5	18.5721
SVD	25	5	5	0.2580	6.3500	2.9478× 10^-5	25.36016
SVD	36	6	5	0.1889	4.4097	1.968 ×10^-5	32.3716
SVD,LI	9	3	3	0.7493	17.5562	3.5891× 10^-4	10.3739
SVD,LI	16	4	4	0.7192	9.8754	3.5681 ×10^-4	11.7131
SVD,LI	25	5	5	0.6943	6.3202	3.5630× 10^-4	13.90725
SVD,LI	36	6	5	0.6738	4.3891	3.5615 ×10^-4	17.5310
SVD,BI	9	3	3	0.7330	155.7969	0.0016	9.1011
SVD,BI	16	4	4	0.6995	87.6358	0.0016	10.1484
SVD,BI	25	5	5	0.6720	56.0869	0.0016	12.1943
SVD,BI	36	6	5	0.6535	38.9491	0.0016	13.17828

Table 3.3: The results for colour image .

[original] [final]

FIGURE 7: The results for SVD method for k=36.



[original] [final]

FIGURE 8: The results for SVD and linear interpolation method for k=36

[original] [final]

FIGURE 9: The results of SVD and bilinear interpolation method for k=36

4. CONCLUSION

In this paper two hybrid SVD methods, using linear interpolation, and bilinear interpolation are presented. The results show the preference of bilinear interpolation combined with SVD. The linear method is also economically acceptable.

5. REFERENCES

[1] J. Stoer, R. Bulirsch. Intoduction to Numerical Analysis. translated by R.Bartels , W.Gautschi. C.Witzgall. Springer-Verlag, second edition, (1991).

[2] H. Abdi. “Singular Value Decomposition *(SVD) and Generalized Singular Value Decomposition (GSVD).”* In N.J. Salkind (Ed.): Encyclopedia of Measurement and Statistics. Thousand Oaks (CA): Sage. pp. 907-912, 2007.

[3] R. Ashino, A. Morimoto, M. Nagase, R.Vaillancourt. “Image Compression with Multiresulotion Singular Value Decomposition and Other Methods.” Mathematical and Computer Modelling. Vol. 41(6-7). pp. 773-790, 2005.

[4] P.Dostert. “An application of linear algebra to image compression.” The university of arizona. July 2009.

[5] M.Sing Song. “Wavelet *Image Compression.” Mathematical subject classification*, 1991.

[6] M.Sing Song. “Wavelet Image Compression”, Southern Illinois University Edwardsville, Edwardsville, IL62026, USA, 2004.