This full-text paper was peer-reviewed and accepted to be presented at the IEEE WiSPNET 2016 conference.

Lossy Image Compression using SVD Coding Algorithm

K.M.Aishwarya[1], Rachana Ramesh[2], Preeti.M.Sobarad[3], Dr. Vipula Singh
Department of Electronics and Communication Engineering, R.N. Shetty Institute of Technology (RNSIT), Bangalore, India
[1]aish3095@gmail.com
[2] rach.3395@gmail.com
[3]preeti.m.sobarad@gmail.com

***Abstract*—Singular Value Decomposition (SVD) deals with the decomposition of general matrices which has proven to be useful for numerous applications in science and engineering disciplines. In this paper the method of SVD has been applied to mid-level digital image processing. SVD transforms a given matrix into three different matrices, which in other words, means refactoring the digital image into three matrices. Refactoring is achieved by using singular values, and the image is represented with a smaller set of values. The primary aim is to achieve image compression by using less storage space in the memory and simultaneously preserving the useful features of original image. The experiments with different singular values are performed and the performance evaluation parameters for image compression viz. Compression Ratio, Mean Square Error, PSNR and Compressed Bytes are calculated for each SVD coefficient. The implementation tool for the tests and experiments is MATLAB.**

Keywords*—*Image processing; Image Compression; Singular Value Decomposition*; *Lowest Rank-k Approximation

I. Introduction

The recent developments and improvements in the field of image security have increased awareness on the importance of digital signal processing for image recognition and image compression. These developments have made it essential to reduce the digital information that needs to be stored and transmitted. The reduction in both the storage space capacity of the image and its transmission bandwidth is exploited using image compression. The main advantage of image compression is that the percentage of irrelevance and redundancy is reduced. This also optimizes the storage space and enhances the transmission rate. Image compression enables image reconstruction [6][5]. The digital information contained by the image determines the degree of compression achieved. Singular Value Decomposition (SVD) is one of the most effective tools for image compression and also for biometric recognition such as face recognition.

Image processing exploits the transformation, storage and retrieval of a digital image [2]. The field of image processing is far-fetched and finds its applications in satellite imagery [2], medical imaging [2], object recognition [2] and image enhancement [2][1]. With the advancement in high speed computers and signal processors, image processing has become the most common form of digital signal processing.

Image compression (in this particular case) takes place in the following order in this research:

1. Refactoring of original image using SVD algorithm on the system tests.
2. Perform image reconstruction for the input image using different values of '*K*' (lowest rank approximations or singular value of the matrix).
3. Compute the Compression Ratio (CR), Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) for quantitative and qualitative measurement of the compressed image, as a measure for performance evaluation of the compressed image.
4. Observe the variation in the performance evaluation parameters with varying singular values '*k*'.

II. Image compression

An image can be represented as an *m* by *n* matrix, where *m*, the number of rows, is the pixel height of the image, and *n*, the number of columns, is the pixel width of the image. When a computer creates or stores an image, each and every pixel is assigned a number to represent its relative darkness or brightness [5]. Each value contained inside the matrix decides the brightness of the corresponding displayed pixel. In case of a grayscale image the range of values within the matrix is from 0 (black) to 1 (white), where each number simply represents light or dark. But for colour images, which are much more space-consuming, the computer must split the image into three layers composed of red, green and blue in the image. Each single-colour picture is then calculated much like a grayscale picture based on darkness, and then recombined at the end to reproduce the original image.

In other words, in case of a colour image, each colour pixel is broken down to three primary components: red, green and blue(RGB) [9]. Hence there are three values associated with each pixel, each ranging from 0 (colour is absent) to 1 (completely saturated). These 3 values are assigned to red, green and blue respectively. For instance, if a 9 megapixel grayscale image is considered, it can be represented as a 3000 × 3000 pixels matrix [9]. Since it is a grayscale image, each pixel in the image matrix can be represented by a certain integer whose value falls between 0 and 255 [9]. If a storage space of 1 byte is assigned to each pixel, then the entire image requires approximately 9MB space. For a colour image, this storage space value is larger since it has three components, red, green and blue (RGB). Each component is represented by a matrix, so storing colour images takes three times the space (27Mb) [9].

III. Redundancies

Three types of redundancies exist when image compression and similar techniques are studied:

A. Coding Redundancy

Coding redundancy is present when less than optimal code words are used. Lookup Tables (LUTs)[3]are used for the implementation of this technique and that makes the coding reversible [3][8]. Huffman coding and Arithmetic Coding techniques are the two most exercised image coding schemes for this technique[3].

B. Interpixel Redundancy

In Interpixel redundancy, a pixel value is predicted based on the values of the neighbouring pixels, by mapping the original 2-D array of pixels into a different format[3][8]. Interpixel redundancy is also called spatial redundancy, inter frame redundancy, or geometric redundancy [3].

C. Psycho-visual Redundancy

From the study of psychophysical aspects of human vision it is known that the human eye is insensitive to certain band of frequencies[8] i.e. it does not respond to all the incoming visual information with equal sensitivity. Some parts of the information come to be of more importance than others. Due to this, there is another type of redundancy that comes into existence, which is called Psycho-visual redundancy[3].

IV. SINGULAR VALUE DECOMPOSITION

Singular Value Decomposition (SVD) deals with the decomposition of general matrices which has proven to be useful for numerous applications in science and engineering disciplines. The SVD is commonly used in the solution of unconstrained linear least squares problems, matrix rank estimation and canonical correlation analysis. Computational science exploits SVD for information retrieval, seismic reflection tomography, and real-time signal processing[2][7].

A. Theory

The goal of SVD is to find the best approximation of the original data points that is of large dimensions, using fewer dimensions. This is possible by identifying regions of maximum variations. So when a high dimensional, highly variable set of data points is taken, SVD is employed to reduce it to a lower dimensional space that exposes the substructure of the original data more clearly and orders it from most variation to the least. In this way, the region of most variation can be found and its dimensions can be reduced using the method of SVD. In other words, SVD can be seen as a method for *data reduction*. [3]

The singular value decomposition is defined as a factorization of a real or complex, square or non-square matrix. Consider a matrix A with *m* rows, *n* columns and rank *r*[2]. Then A can be factorized into three matrices:

$$A = U\Sigma V^T \qquad (1)$$

Or,

$$A = [u_1 \cdots u_r \cdots u_m] \begin{bmatrix} \sigma_1 & & & & \\ & \ddots & & & \\ & & \sigma_r & & \\ & & & \ddots & \\ & & & & 0 \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_r^T \\ \vdots \\ v_n^T \end{bmatrix}$$ [8]

Where U and V are orthonormal matrices and the matrix Σ is a diagonal matrix with positive real entries [7]:

- U is an $m \times m$ orthogonal matrix
- V^T is the conjugate transpose of the $n \times n$ orthogonal matrix [7].
- Σ is an $m \times n$ diagonal matrix with non-negative real numbers on the diagonal which are known as the singular values of A [7].
- The *m* columns of U and *n* columns of V are called the left-singular and right-singular vectors of A respectively.
- The singular values of Σ are arranged as $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r \geq 0$, [7] where the largest singular values precede the smallest and they appear on the main diagonal of Σ.
- The numbers $\sigma_1^2 \geq \ldots \geq \sigma_r^2$ are the eigen values of AA^T and A^TA. [7]

B. Steps to calculate SVD of a matrix

1) First, calculate AA^T and A^TA.
2) Use AA^T to find the eigen values and eigenvectors to form the columns of U: $(AA^T - \lambda I)\ddot{x} = 0$ [3].
3) Use A^TA to find the eigen values and eigenvectors to form the columns of V: $(A^TA - \lambda I)\ddot{x} = 0$.
4) Divide each eigenvector by its magnitude to form the columns of U and V.
5) Take the square root of the eigen values to find the singular values, and arrange them in the diagonal matrix S in descending order: $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r \geq 0$ [3].

C. Properties of SVD

SVD has many properties and attributes. A selective few of the properties have been listed here[1]:

1) The singular values 'σ' are unique, unlike, the matrices U and V, which are not unique.[1]
2) The singular values of a rectangular matrix A are equal to the square roots of the eigen values $\lambda_1, \lambda_2 \ldots \lambda_m$ of the matrix A^TA.[1]
3) Mathematically, the rank of the matrix A is the number of its non-zero positive singular values; rank (A) = r, $r \leq m$.
4) Since $AA^T = U\Sigma\Sigma^{(T)}U^{(T)}$, so U diagonalizes AA^T and $u_{(i)}$ s are the eigenvectors of AA^T.
5) Since $A^TA = V\Sigma^{(T)}\Sigma V^{(T)}$, V diagonalizes A^TA and the $v_{(j)}$s are the eigenvectors of A^TA [1].
6) If A has rank 'r' then $v_j, v_j, \ldots, v_r$ form an orthonormal basis for range space of A^T, $R(A^T)$, and $u_j, u_j, \ldots, u_r$ form an orthonormal basis for range space A, R(A) [1].

D. SVD Approach to Image Compression

SVD divides a square matrix into two orthogonal matrices(U, V) and a diagonal matrix (∑).So the original matrix is rewritten as a sum of much simpler rank-one matrices. SVD is applied on an image matrix A to decompose it into 3 different matrices U, ∑ and V. But applying SVD alone does not compress the image. To compress an image, after applying SVD, only a few singular values have to be retained while other singular values have to be discarded [1]. All the singular values are arranged in descending order on the diagonal of ∑matrix [1]. The discarding of the SV's follows the fact that the first singular value on the diagonal of ∑contains the greatest amount of information and subsequent singular values contain decreasing amounts of image information [1].

Thus, negligible amount of information is contained in the lower SV's. So, they can be positively discarded after performing SVD, simultaneously avoiding significant image distortion [1].Furthermore, property3 of SVD (section V '*C*') says that '*the number of non-zero singular values of A is equal to the rank of A*'. In cases where the lower order singular values after the rank of the matrix are not zero, the discarding can still be done since they have negligible values and are treated as noise [1].

SVD image compression process can be illustrated by implementing the following algorithm, where a given matrix A is expressed as follows:

$$A = U\Sigma V^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T \quad (2)$$

Now the image matrix 'A' can be represented by the outer product expansion [2]:

$$A_k = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_r u_r v_r^T \quad (3)$$

When performing image compression, the sum is not performed to the very last Singular Values(SV's)[2] ;theSV's with small enough values are dropped. The values falling outside the required rank are equated to zero. The closet matrix of rank k [2] is obtained by truncating those sums after the first k terms:

$$A_k = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_k u_k v_k^T \quad (4)$$

The total storage for A_k will be:

$$A_k = k(m + n + 1) \quad (5)$$

The value of integer k can be chosen less than n[2]. This won't bring any significant change in the image under consideration. Hence the digital image corresponding to A_k will still have very close resemblance to the original image [2]. As different values of k are chosen, it is observed that each value of k pertains to each corresponding image with their corresponding storage capacities. Further approximation in the image matrix can be achieved by dropping more singular terms of the matrix A, thereby reducing the storage space of the image on the computer and achieving disk space optimization[2].

E. SVD Image Compression Measures

To measure the performance of the SVD compression, the quantitative and qualitative measurement of the compressed image is found by calculating the following 3parameters:

1. Compression Ratio (C_R)

Compression Ratio is defined as the ratio of file sizes of the uncompressed image to that of the compressed image[4]:

$$C_R = m * n / (k(m + n + 1)) \quad (6)$$

2. Mean Square Error (MSE)

MSE is defined as square of the difference between pixel value of original image and the corresponding pixel value of the compressed image averaged over the entire image[1].

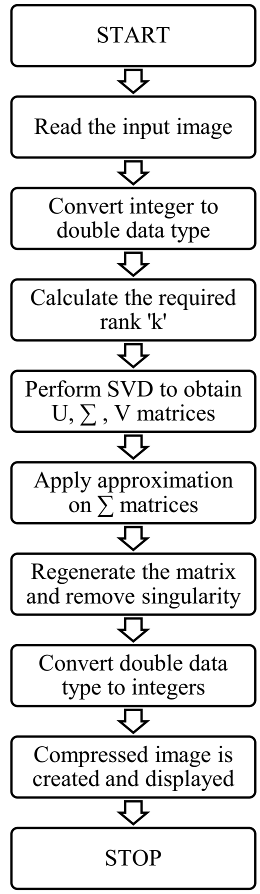

Fig1. Flowchart of computing SVD

Mean Square Error (MSE) is computed to measure the quality difference between the original image A and the compressed image A_k, using the following formula[3]:

$$MSE = \frac{1}{mn} \sum_{y=1}^{m} \sum_{x=1}^{n} \left(f_A(x,y) - f_{A_k}(x,y)\right) \quad \text{-------------} \quad (7)$$

3. Peak Signal to Noise Ratio (PSNR)

Peak signal-to-noise ratio (PSNR) is defined as the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. PSNR is usually expressed in terms of the logarithmic decibel scale to accommodate signals with a wide range.

In lossy compression, the quality of compressed image is determined by calculating PSNR. The signal in this case is the original data, and the noise is the error introduced by compression. The PSNR (in dB) is given by the equations [3]:

$$PSNR = 10.\log_{10}\left(\frac{MAX_I^2}{MSE}\right) \quad (8)$$

$$= 20.\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \quad (9)$$

$$= 20.\log_{10}(MAX_I) - 10.\log_{10}(MSE) \quad (10)$$

Here, MAX_I is the maximum possible pixel value of the image[3].

F. SVD v/s Memory (Storage Space Calculation)

The amount of memory required by a non-compressed image I needs to be found out first. Considering an $m \times n$ pixel grayscale image, the values stored for such an image is mn, where one value is assigned for each pixel.

$$I_M = m\,n$$

On the contrary, a rank k SVD approximation of I consists of three matrices, U ,∑ and V; so the method for computation of the number of values for SVD compressed image is trickier[5].After SVD is performed, the number of values required to store U is calculated. U is an $m \times m$ matrix, where only the first k columns are desired. Hence U can be considered to bean $m \times k$ matrix with mk values.

$$U_M = mk$$

Similarly, for matrix V only the first k columns are considered (which become the rows of V^T),so V is stored as an $n \times k$ matrix containing nk values.

$$V_M = nk$$

Lastly, due to the consideration of only the first k columns of U and V, only the first k singular values of the ∑matrix are considered. So, $\sum_M = k$ Now the rank k approximation requires a total number of A_M values which is computed as [5]:

$$A_M = U_M + V_M + \textstyle\sum_M \quad (11)$$

$$A_M = mk + nk + k$$

$$A_M = k\,(m + n + 1) \quad (12)$$

G. Best Rank - k Approximations

Most images aren't perfect geometric shapes. Many images that we come across have full rank, i.e. to compress those images using SVD, it is observed that their best rank approximation parameter 'k' will nearly be equal to n (where n is the smaller dimension of the original matrix I).Solving for A_M in this case [5].

$$A_M = n\,(m + n + 1) \quad (13)$$

$$A_M = mn + n^2 + n \quad (14)$$

But for a general consideration, the compressed image A_M occupies more space than the original image ($I_M = mn$), which is contradictory to the purpose of SVD compression [5]. In order to save memory, in other words to reduce the storage space during SVD, the value of k has limit restrictions on it. Hence $A_M \le I_M$:

$$A_M < I_M$$

$$k(m+n+1) < mn \quad (15)$$

$$k < \frac{mn}{m+n+1} \quad (16)$$

The same rule for computation of k applies to colour images [5].Incase of colouring images $I_M = 3mn$,while

$$A_M = 3(U_M + V_M + \textstyle\sum_M)$$

$$A_M = 3k(m + n + 1) \quad (17)$$

Thus,

$$k < \frac{3mn}{3(m+n+1)}$$

Hence,

$$k < \frac{mn}{m+n+1}$$

Theorem:A best k-rank approximation $A_k$is given by zeroing out the $(r-k)$trailing singular values of A, that is:-

$$A_k = U\Sigma_k V^T, \Sigma_k = \mathrm{diag}(\sigma_1, \cdots, \sigma_k, 0, \cdots, 0) \quad (18)$$

The minimal error is given by the Euclidean norm of the singular values that have been zeroed out in the process:

$$\|A - A_k\|F = \sqrt{\sigma_{k+1}^2 + \cdots + \sigma_r^2} \quad (19)$$

Let the SVD of A be:

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T \quad (20)$$

For k = {1,2…r}, let

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T \quad (21)$$

be the sum truncated after k terms. The rank of the SVD compressed image A_k is k. Furthermore, A_k is the best rank k approximation to A when the error is measured in either the 2-norm or the Frobenius norm [5].

TABLE1. PERFORMANCE MEASUREMENT OF SVD FOR DIFFERENT VALUES OF '*K*'

No. of Singular Values used 'k'	Performance Evaluation Parameters			
	Compression Ratio (CR)	MSE (in dB)	PSNR (in dB)	Bytes of Compressed Image (B)
2	61.046	40.61	25.04	30796
12	23.588	39.42	28.17	31679
22	12.101	38.09	30.32	31699
52	5.72	38.047	32.39	31781
112	2.01	37.44	33.46	31815
R=202	0.580	-20.93	83.88	31840
251	0.564	-20.72	86.42	31862
260	0.552	-20.69	86.437	31905
262	0.544	-20.66	86.453	31926
264	0.536	-20.66	86.453	32124

[Bytes of input image: 124358]

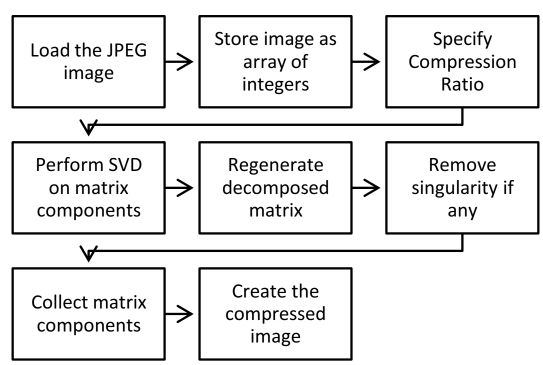

Fig.2 Representation of implemented algorithm

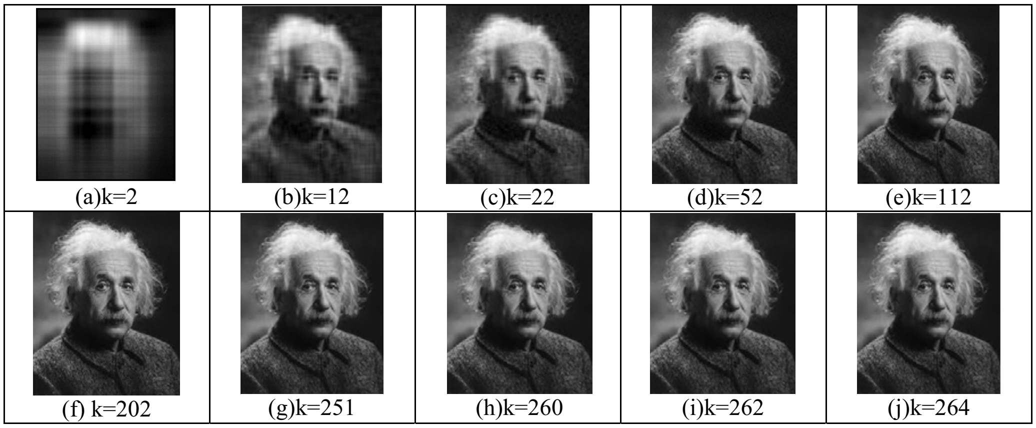

Fig 3. SVD compression for different Singular Values'k'

V. Experiment and Result Analysis

Initially the processor is fed with the JPEG image which has to be compressed [8]. This input image is stored as an array of integers. For performing compression in MATLAB, the array of integers is converted to double data type. Now, prior to the compression process, the user needs to specify the amount of compression that is desired. This is achieved by specifying the Compression Ratio [8] for the particular input JPEG image. Singular Value Decomposition is performed to refactor the input image matrix and is then applied separately to the matrix components. The resultant decomposed matrix is regenerated by decoding the bit stream. SVD compression technique is applied to the input image for different singular values and the compressed images are created and displayed in Fig.3. Considering different values of *k*, the result of SVD compression is depicted in the images displayed above. Fig.3(*f*) shows the original image. The displayed image is extremely blurred when the singular value is chosen to be 3, as shown in Fig.3(*a*). Alternatively, it means that only first 2 eigen values of ∑ matrix are considered for image reconstruction. The image obtained by applying SVD using *k=12* as the singular value (shown in Fig.3*(b)*) shows comparatively less signs of distortion than Fig.3*(a)*. By observing figures 3(*a)* to 3(*j)*, conclusion can be drawn that with increase in the value of *k*(i.e. number of Eigen values used for reconstruction of the compressed image), the compressed image approaches the original image with negligible MSE. The rank of the input image matrix is 202 as observed from the tabular column. Following inference can be drawn on the basis of the above table and plots:

1. The Eigen values used in the reconstruction of the compressed image are represented by the parameter '*k*'.
2. Smaller values of *k* imply greater compression ratio (i.e. less storage space is required) but a deterioration in the image quality (i.e. larger MSE values& smaller PSNR values).
3. Thus, it is necessary to strike a balance between storage space required and image quality for good image compression. From the above observations, it is found that optimum compression results are obtained when MSE of the compressed image is just less than or equal to 38dB (i.e. MSE ≤ 38dB). In our case, this is obtained when value of *k* is 52.
4. Thus, it is necessary to strike a balance between storage space required and image quality for good image compression. From the above observations, it is found that optimum compression results are obtained when MSE of the compressed image is just less than or equal to 38dB (i.e. MSE ≤ 38dB). In our case, this is obtained when value of *k* is 52.
5. When *k* is equal to the rank of the image matrix (202 here), the reconstructed image is almost same as the original one. And as *k* is increased further, there is anegligible decrease in the MSE values. This means that improvement in the image qualityis very negligible.

VI. Conclusion

In performing SVD compression for JPEG images the values of Compression Ratio and their variation with corresponding singular values (SVD coefficients) are observed and their relation is concluded to be a decreasing exponential function. More compression ratio can be achieved for smaller ranks. On the other hand, the computation time for the compressed images is the same for all the values of *k* taken. It was also found that the fewer the singular values were used, the smaller the resulting file size was. An increase in the number of SVD coefficients causes an increase in the resulting file size of the compressed image. , As the number of SVD coefficients nears the rank of the original image matrix, the value of Compression Ratio approaches one. From the observations recorded it can be seen that the Mean Square Error decreases with increase in the number of SVD coefficients, unlike PSNR which varies inversely with the value of 'k'. Therefore, an optimum value for 'k' must be chosen, with an acceptable error, which conveys most of the information contained in the original image, and has an acceptable file size too.

VII. Applications

1. An optimum value of Compression Ratio (C_R) is characteristic to any image compression technique to make the compressed image well adapted to statistical variations [2]. SVD has proven to be advantageous in this aspect.
2. The range and null space of a matrix are important quantities in linear algebraic operations, which are explicitly defined by SVD, through the left and right singular vectors (U & V resp.). Vector U has vanishing singular values of original image that span its 'null space'. Vector V contains the non-zero singular values of original image that span the 'range'.
3. Noise reduction is also one of the many applications of SVD. In this paper A stands for an image matrix. Likewise, if A represents a noisy signal, then on computation of SVD, small singular values of A can be discarded. The discarded SV's mainly represent noise. Hence, the compressed signal A_k represents a noise-filtered signal.
4. SVD also finds its application in the area of Face Recognition

VIII. Future work

The future work will focus on the use of Wavelet Difference Reduction (WDR) and Adaptively Scanned Wavelet Difference Reduction (ASWDR) for further compressing of the image. WDR offers high compression of the overall system. ASWDR adapts the scanning procedure used by WDR in order to predict locations of the significant transform values at half thresholds. The study will also focus on image reconstruction and face recognition

References

[1] Samruddhi Kahu, Reena Rahate, "Image Compression using singular Value Decomposition", *International Journal of Advancements in Research and Technology,* Volume 2, Issue 8, August-2013.

[2] Lijie Cao, "Singular Value Decomposition Applied To Digital Image Processing*", Division of Computing Studies, Arizona State university Polytechnic Campus, Arizona*.

[3] Abhishek Thakur, Rajesh Kumar, Amandeep Bath, Jitender Sharma, "Design of image compression algorithm using MATLAB", *IJEEE,* Vol. 1, Issue 1, p-ISSN: 1694-2426.

[4] Mrak M., Grgic S. and Grgic M., PictureQuality Measures in image compression systems, *IEEE EUROCON, Ljubljana, Eslovenia,* September 2003.

[5] Dan Kalman "A Singularly Valuable Decomposition: the SVD of a matrix", The American University, Washington D.C, February 13, 2002.

[6] M. Grossber, I. Gladkova, S. Gottipat , M. Rabinowitz, P. Alabi, T. George1, and A.Pacheco, "A Comparative Study of Lossless Compression Algorithms on Multi-Spectral Imager Data", *Data Compression Conference, IEEE, pp:447, 2009.*

[7] Stefany Franco, Dr. Tanvir Prince, Ildefonso Salva, and Charlie Windolf, "Mathematics of Image Compression", *Journal of Student Research 2014, Vol. 3, Issue 1, pp: 46-62.*

[8] Rehna V.J. , Jeyakumar M.K. , "Singular Value decomposition based image coding for achieving additional compression to JPEG images", *International Journal of image processing and vision sciences (IJIPVS)* Volume-1 Issue-1, 2012.

[9] Prasantha H S, Shashidhara H L, Balasubramanya Murthy K N, "Image Compression using SVD", *International Conf. on Computational Intellignece and Multimedia Applications, 2007.*

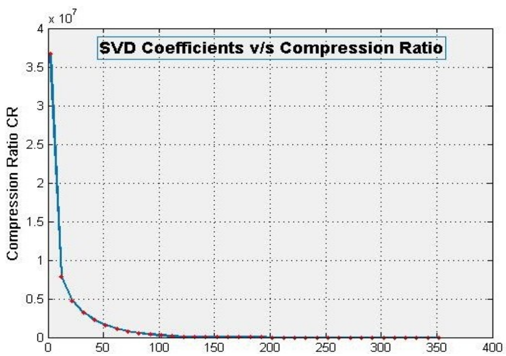

Fig 4. Plot of SVD Coefficients v/s Compression Ratio

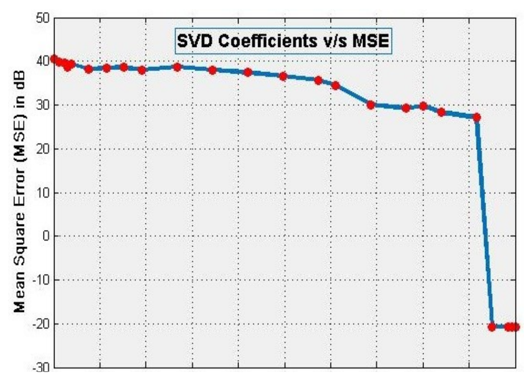

Fig 5. Plot of SVD Coefficients v/s MSE

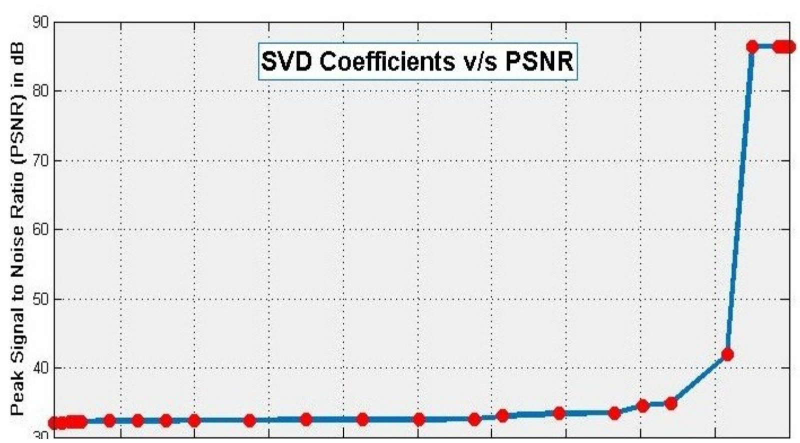

Fig 6. Plot of SVD Coefficients v/s PSNR