International Journal of Advances in Electronics Engineering

Vol:1 Issue:1 ISSN 2278 - 215X

# JPEG Image Compression using Singular Value Decomposition

Mrs. Rehna V.J
*Department of Telecommunication*
*Atria Institute of Technology*
Bangalore, India
rehna_vj@yahoo.co.in

Mr. Abhranil Dasgupta
*Department of Telecommunication*
*Atria Institute of Technology*
Bangalore, India
abhra.nil007@gmail.com

***Abstract*** **—Computer technology these days is most focused on storage space and speed. Considerable advancements in this direction can be achieved through the usage of digital image compression techniques. In this paper, well studied singular value decomposition based JPEG Image Compression Technique is presented. Singular Value Decomposition is a way of factorizing matrices into a series of linear approximations that expose the underlying structure of the matrix. SVD is extraordinarily useful and has many applications such as data analysis, signal processing, pattern recognition, objects detection and weather prediction. An attempt is made to implement this method of factorization to perform second round of compression on JPEG images to optimize storage space. Compression is further enhanced by the removal of singularity after the initial compression performed using SVD. MATLAB R2010a with image processing toolbox is used as the development tool for implementing the algorithm.**

***Keywords—Singular Value Decomposition, JPEG images, Compression factor, Compression ratio, Rank, Eigen values, Eigen vectors, Singular value.***

## I. INTRODUCTION

Image Compression, an important area in the field of digital image processing, deals with techniques for reducing the storage required for saving an image or the bandwidth required for transmitting it [1]. The objective of Image Compression is to reduce irrelevance and redundancy of the image data thereby optimizing the storage space and increasing the transmission rate over WebPages. Image compression enables image reconstruction. The amount of compression achieved depends on the contents of image data. A typical photographic image can be compressed to about 80% of its original size without experiencing noticeable degradation in the quality. The technique of image compression finds applications in various fields such as Medical Imaging, Museums and Galleries, Web Applications, Telecommunication, Facsimile and Security Industry to name a few.

To date different algorithms have been developed for image compression. These include Predictive Coding, Fractal Compression, Burrows-Wheeler Transform, Wavelet Compression and Embedded Zero Tree Wavelet. Predictive coding refers to the de-correlation of similar neighbouring pixels within an image to remove redundancy. An example of this type is Huffman Coding which is a statistical compression method that converts characters into variable length bit strings. Transform coding includes Burrows-Wheeler transform which is a preprocessing technique which is useful for improving lossless compression. Delta encoding aids in compression of data in which sequential data occurs frequently. Fractal compression is a method used to compress images using fractals. Fractal algorithms convert these parts into mathematical data called fractal codes which are used to recreate the encoded image.Wavelet compression is a form of data compression well suited for image and audio compression. The entire image is treated as a series of wavelets which are the changes from pixel to pixel as measured by the deviation of an individual pixel from zero. EZW is a progressive encoding to compress an image into a bit stream with increasing accuracy. This may be lossy compression. Vector quantization is a technique often used in lossy data compression which requires the development of an appropriate codebook to compress data.

This paper focuses on Singular Value Decomposition (SVD) which is a way of factorizing matrices into a series of linear approximations that expose the underlying structure of the matrix. SVD is commonly used in Object detection, Face recognition, Field matching techniques and Meteorological and Oceanographic data analysis.

Rest of the paper is constituted as follows: Section II deals with Image Compression, section III comprises of Image File Formats, section IV describes the Procedure of SVD, section V deals with Application of SVD in Image Compression followed by Methodology Used in section VI, section VII elaborates the Flow of the Process, Results Obtained is shown in section VIII and finally Conclusion of the paper is presented in section IX.

## II. IMAGE COMPRESSION

The term Compression refers to the process of reducing the amount of data required to represent a given quantity of information. Image Compression aims to reduce the number of bits required to represent an image by removing the redundancies which makes it one of the most useful and commercially successful technologies in the field of Digital Image Processing.

Three principle types of data redundancies[1],[4] that can be identified are:

A. *Coding redundancy:* Coding redundancy consists of variable length code words selected as to match the statistics of the original source. In the case of Digital Image Processing, it is the image itself or the processed version of its pixel values. Examples of image coding schemes that explore coding redundancy are the Huffman codes and the Arithmetic coding technique.

B. *Spatial redundancy:* Spatial reduncancy is sometimes called Interframe redundancy, Geometric redundancy or Interpixel redundancy. Here, because the pixels of most 2-D intensity arrays are correlated spatially that is each pixel is similar to or independent on neighbouring pixels, information is unnecessarily replicated in the representation of the correlated pixels. Examples of this type of redundancy include Constant area coding and many Predictive coding algorithms.

C. *Irrelevant information:* Most 2-D intensity arrays contain information that is ignored by the human visual system. Image and video compression techniques aim at eliminating or reducing any amount of data that is psycho visually redundant. Most of the image coding algorithms in use today exploit this type of redundancy, such as the discrete cosine transform based algorithm at the heart of the JPEG encoding standard.

The usual steps involved in compressing an image are [5], specifying the rate and distortion parameters for the target image, dividing this image data into various classes, based on their importance. Then dividing the available bit budget among these classes, such that the distortion is minimum. Next step involves quantizing each class separately using the bit allocation information followed by encoding of each class using an entropy coder and write to the file.

There are many approaches to Image Compression but they can be categorized into two fundamental groups: Lossless and Lossy.

In Lossless Compression, also known as reversible compression, the reconstructed image after compression is numerically identical to the original image on a pixel-by-pixel basis.

In Lossy Compression, also known as irreversible compression, the reconstructed image contains degradation related to the original image. As a result significantly higher compression can be achieved as compared to Lossless Compression [2].

## III. IMAGE FILE FORMATS

In the context of Digital Imaging, an image file format is a standard way to organize and store image data. It defines how the data is arranged and the type of compression that is used. There are several formats using which image files can be compressed [6],[7],[8] These include:

A. *BMP (Bitmap)*: Windows Bitmap or BMP files are image files within the Microsoft Windows Operating System. BMP files are not very popular as they do not scale or compress the images well. Being oversized, this format is not web friendly.

B. *GIF (Graphics Interchange Format)*: GIF is a popular image format on the internet because its file size is relatively small compared to other image compression types. GIF is most suitable for graphics, animations, diagrams and cartoons.

C. *PNG (Portable Network Graphics)*: This format is designed specifically for web applications. This format is lossless so it does not lose quality and detail after image compression. PNG format is not suitable for large images because they tend to generate a very large file.

D. *TIFF (Tagged Image File Format)*: It is recommended especially for text, black and white images. TIFF is very flexible; it can be lossy or lossless. It is a rich format and is supported by many imaging programs. It is the standard format for printing, scanned documents and optical character recognition since it does not have any artifacts. Drawbacks of this format include long transfer time, huge disc space consumption and slow loading time.

E. *PPM (Portable Pix Map)*: It is a very old image format that can represent any ordinary colour image. PPM files are basically plain text files making it one of the simplest formats. The PPM format is not intended to be an archival format, so it does not need to be too storage efficient.

*F. PGM (Portable Grayscale Map):* PGM format represents a grayscale graphic image. It is designed to be extremely easy to learn and write programs.

*G. JPEG (Joint Photographic Expert Group):* JPEG file format differs from other file formats as it is lossy. JPEG's compression technology reduces the true quality of the image in order to achieve its striking file size reduction. JPEG was designed specifically for use with highly detailed or photorealistic images, and is typically applied to rendered images and digitized photographs. It is not suitable for use with rough drafts, line drawings, screen captures and other image types which use sharply defined lines and coloured images.

In this paper, SVD based image compression on JPEG image is proposed to reduce the file size further, in addition to the compression already achieved by JPEG format which is used as the input for compression [9].

## IV. The SVD Process

In linear algebra, the Singular Value Decomposition (SVD) is a factorization of a real or complex matrix. SVD is effective compared to other linear approximation techniques [10],[11]. SVD has many practical and theoretical applications like scientific computing, signal processing, automatic control along with Image Compression. One special feature of SVD is that it can be performed on any real mxn matrix. It factorizes matrix A into three matrices U, S and V, such that,

$$A = USV^T \quad (1)$$

where, U is a left singular matrix and V is the right singular matrix and S is a diagonal matrix. The columns of U and V are represented as $u_i$ and $v_i$ respectively and the diagonal elements $s_i$ of S matrix are called singular values. The singular vectors form orthonormal basis, and the relation,

$$Av_i = s_i u_i \quad (2)$$

shows that each right singular vector is mapped on to the corresponding left singular vector. The singular values are arranged on the main diagonal in such an order

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \sigma_4 \ldots\ldots\sigma_r \geq \sigma_{r+1} = \ldots = \sigma_p = 0, \quad (3)$$

where, r is the rank of matrix A, and $(_p)$ is the smaller of the dimensions m or n. Rank is the number of linearly independent rows and columns of the input matrix.

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 compatible than original image [12].

The process of Singular Value Decomposition begins by selecting the matrix A which has m rows and n columns. Now, matrix A is factorized into three matrices U, S and $V^T$.

Generation of matrix V involves the follows steps:

Pre-multiplying both sides of the equation $A = USV^T$ by $A^T$ yields,

$$A^TA = (USV^T)^T (USV^T) = VS^TU^TUSV^T \quad (4)$$

$U^TU$ gives the identity matrix and $S^TS=S^2$ since S is a diagonal matrix. On substituting these values in the above equation gives,

$$A^TA = VS^2V^T \quad (5)$$

Eigen values and Eigen vector of matrix $A^TA$ are needed to find V and S matrices.

The Eigen vectors of a square matrix are the non-zero vectors that, after being multiplied by a matrix, remain proportional to the original vector i.e., they change only in magnitude, not in direction. For each Eigen vector, corresponding Eigen value is the factor by which the Eigen vector changes when multiplied by the matrix. The mathematical interpretation of this idea is as follows [13],[14]:

If A is a square matrix, of size nxn and $\lambda$ is an associated Eigen value such that

$$Av_i = \lambda v_i \quad ; \quad i=l, 2, 3, \ldots, n \quad (6)$$

Then v is called an Eigen vector of matrix A, associated with Eigen value $\lambda$.

The above equation can be rewritten as

$$Av_i = \lambda I v_i \quad (7)$$

where I is the identity matrix of size nxn. The size of this identity matrix has to be the same as that of the matrix for which the Eigen values and Eigen vectors have to be calculated. Equation (7) reduces to

$$(A-\lambda I)\, v_i = 0 \quad (8)$$

The Eigen values $\lambda$ of matrix A are those real numbers for which the homogenous system defined by

equation (8) has a non-zero solution and the Eigen vector of matrix A associated with λ are the non-zero solutions of this system. Equation (8) has a non-zero solution if its coefficients' matrix is noninvertible and this is possible if its determinant is equal to zero, i.e.

$$|A-\lambda I| = 0 \qquad (9)$$

The equation (9) is said to be the characteristic equation of matrix A. In the context of SVD compression, the Eigen values are the square of the elements of S (the singular values), and the Eigen vector consists of columns of matrix V (the right singular vectors).

The next step is to find U:

Post-multiplying both sides of the equation $A = USV^T$ by $A^T$ gives,

$$AA^T = (USV^T)(USV^T)^T = USV^TVS^TU^T \qquad (10)$$

Here $V^TV$ gives the Identity matrix and $S^TS=S^2$ since S is a diagonal matrix. On substituting these values in equation (10) yields,

$$AA^T = US^2U^T \qquad (11)$$

Again the Eigen vectors are calculated, but this time for the matrix $AA^T$. These are the columns of U (the left singular vector). The process of calculating Eigen values and Eigen vector follow the same steps as explained previously in the calculation of matrix V.

Once U, S and V matrices are obtained, matrix A can be generated which is represented as the product of matrices U, S and $V^T$.

$$A = \begin{bmatrix} u_1 & \cdots & u_r & \cdots & u_m \end{bmatrix} \begin{bmatrix} \sigma_1 & & & & \\ & \ddots & & & \\ & & \sigma_r & & \\ & & & \ddots & \\ & & & & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ \vdots \\ V_r^T \\ \vdots \\ V_n^T \end{bmatrix}$$

where, U is m × m matrix, S is m × n matrix and V is n × n matrix.

## V. SVD In Image Compression

The matrix A (m × n) is approximated by using far fewer entries than in the original matrix. When the rank r < m or r < n, the redundant information is removed. Here, rank is the total number of non-zero diagonal elements of the S-matrix. These are also called as singular values which are arranged in the decreasing order along the main diagonal. The values which fall outside the required rank are equated to zero as shown in equation (12) [15],[16].

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T \ldots + \sigma_r u_r v_r^T + 0 u_{r+1} v_{r+1}^T + \ldots \qquad (12)$$

Since the singular values are always greater than zero, adding on the dependant terms where the singular values are equal to zero does not affect the quality of the image. Therefore the terms at the end of the equation zero out yielding,

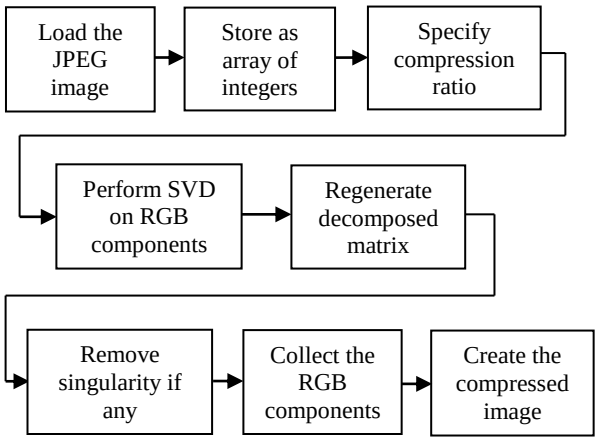
$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \ldots + \sigma_r u_r v_r^T \qquad (13)$$

We can further approximate the matrix by leaving off more singular terms of the matrix A [15],[17]. This further reduces the amount of space required to store the image on a computer hence, optimizing the disc space.

## VI. Methodology Used

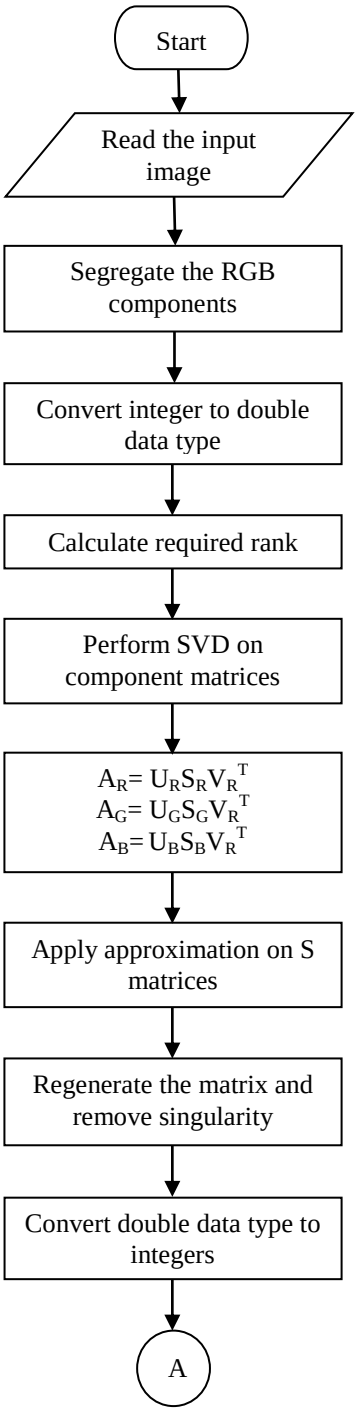
Singular Value Decomposition technique discussed in the previous section is implemented to compress JPEG images further. The process required to accomplish this is as follows:

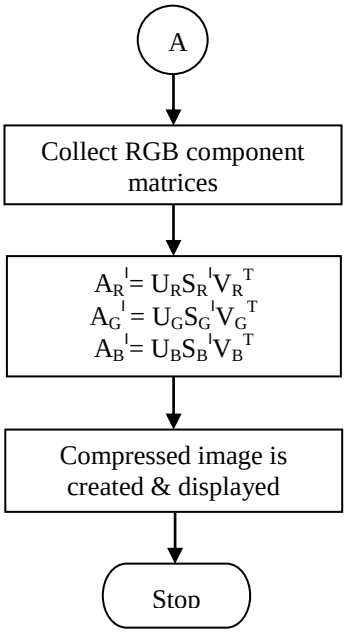
Initially the JPEG image which has to be compressed is given as an input to the processor. This input image is stored as an array of integers. Before getting on with the process of compression, the amount of compression that has to be achieved for the input JPEG image is specified through the compression ratio. Compression ratio is defined as the ratio of file sizes of the uncompressed image to that of the compressed image. Compression is then achieved by performing Singular Value Decomposition on RGB components of the input JPEG image [18]. The resultant decomposed matrix is regenerated by decoding the bit stream.



The next step is to remove singularity if any. Removing singularity is nothing but removing the redundant pixels having same frequencies. This not only helps in reducing the file size but also maintains the quality of the image. Finally the reformed RGB components are collected and combined to create and display the compressed image.

## VII. FLOW DIAGRAM




The flow of JPEG image compression using SVD technique is as shown above. The input image is read by the Matlab software and stored as an array of integers. This image is segregated into RGB or red, green, blue component matrices. The pixel values of these matrices are converted to double data type for higher accuracy. After determining the rank of the original matrix, compression factor, which is the inverse of compression ratio, is specified. The rank required for the SVD process is calculated by dividing the original rank by the compression factor. In the next step, SVD is performed on the component RGB matrices using the new rank obtained thus yielding the matrices $A_R$, $A_G$ and $A_B$ [19]. The S matrices of these component matrices are approximated by equating the values which fall outside the new rank to zero. This results in three new S matrices $S_R^I$, $S_G^I$ and $S_B^I$ respectively. Using these S matrices, the RGB matrices are regenerated. Suppose there are two consecutive values having almost the same frequencies, higher of the two values are equated to 1, thus singularity is achieved. The next step involves conversion of double data type back to integers. The RGB component matrices are collected thus yielding $A_R^I$, $A_G^I$ and $A_B^I$ matrices. These matrices are combined to create and display the compressed output image.

## VIII. Results Obtained

Original Bitmap Image | JPEG Compressed Image

CF: 8 | CF: 12

CF: 15 | CF: 20

Fig. SVD compressed images. Top right: SVD compressed image with a compression factor 8, Top left: Compressed image with compression factor of 12, Bottom left: Compressed image with compression factor of 15, Bottom right: Compressed image with compressed factor of 20

As shown above, the bitmap image having a file size of 768 KB, when compressed to JPEG format, has a file size of 99.5 KB. This image after being compressed using SVD technique has a file size of 35.8 Kb for a compression factor of 8; for a compression factor of 12, it reduces to 32.8 KB, for a compression factor of 15, it comes down to 31.1 KB and finally for a compression factor of 20, the file size further reduces to 28.5 KB. Thus it can be noticed that as the compression factor is increased, the file size of the compressed image reduces. Also there is considerable degradation in the quality of the image as the compression factor is increased. Further an optimum value of compression factor should be chosen to provide a trade-off between the reduction in file size and quality of the image.

## IX. Conclusion

Using SVD, further 25% compression is obtained in addition to 40% achieved by JPEG format [20]. Higher compression ratio is achieved due to additional compression; without compromising much on the quality of the image. Hence the image obtained is almost indistinguishable from the original image which uses only 35% of the original storage space.

## Acknowledgment

Authors extend thanks to Dr. Shivkumar, Principal, Atria Institute of Technology, Bangalore and Prof. Vasanthi. S, Head of the Department of Telecommunication Engineering, Atria Institute of Technology, Bangalore for their support and co-operation in carrying out the work successfully in the institute.

## References

[1] Rafael C.Gonzales and Richard E.Woods, Digital image processing, Pearson Education, 2001, 2nd edition.
[2] Rafael C.Gonzales, Richard E.Woods and Steven L.Eddins, Digital image processing using matlab, Gatemarks Publications, 2nd edition.
[3] Anil K.Jain, Fundamentals of Digital Image Processing, Pearson Education, 2001.
[4] www.*encyclopedia.jrank.org* < *Contributed Topics from FJ*
[5] www.debugmode.com/imagecmp/
[6] www.scantips.com/basics09.html
[7] *netpbm.sourceforge.net/doc/ppm.html*
[8] www.cic.edu/conferences_events/workshop/ *.../baker_supplemental1.pdf*
[9] Stewart, G. W. (1993), "On the Early History of the Singular Value Decomposition", *SIAM Review* 35 (4): 551–566, doi:10.1137/1035134,http://citeseer.ist.psu.edu/stewart92early.html. www.cofc.edu/~langvillea/...**SVD**Module/*p4module.html*
[10] www.uwlax.edu/faculty/will/**svd**/index.html
[11] Strang, Gilbert. Introduction to linear algebra. Wellesley-Cambridge Press, 1998.
[12]"Image Compression using Singular Value Decomposition", Adam Abrahamsen and David Richards, December 14, 2001.
[13] *www1.mengr.tamu.edu/**EigenvaluesEigenvectors**.pdf*
[14] G. H. Golub and C. F. V. Loan. *Matrix Computations*. The John Hopkins University Press, 2007.
[15] H. C. Andrews and C. L. Patterson. Singular Value Decomposition (SVD) Image Coding. *IEEE Transactions on Communications*, 24:425–432, April 1999.
[16] C. S. M. Goldrick, W. J. Dowling, and A. Bury. Image coding using the singular value decomposition and vector quantization. In *Image Processing and Its Applications*, pages 296–300. IEE, 2000.
[17]Kalman, Dan. " Singularly valuable Decomposition". The college mathematics journal. Vol 27_N0.1_Jan 1998, 2-23.
[18] *simplephotoshop.com › Photoshop tutorials › Color&Tone*
[19] P. Waldemar and T. A. Ramstad. Image compression using singular value decomposition with bit allocation and scalar quantization. In *Proceedings of NORSIG Conference*, pages 83–86, 1996
[20] W. B. Pennebaker and J. L. Mitchell. *JPEG still image data compression standard*. Van Nostrand Reinhold, 1993.