IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

IMPLEMENTATION OF IMAGE COMPRESSION USING FAST COMPUTATION OF SVD ON DM642

Dr. H S Prasantha (0000-0003-4739-517X)
Professor
Department of Electronics & Communication Engineering
Nitte Meenakshi Institute of Technology, Bangalore, Karnataka, India

Abstract: Image compression aims at reducing the number of bits required to represent an image by removing the spatial and spectral redundancies as much as possible. The focus of the research work is only on still image compression. The lossy compression methods which give higher compression ratio are considered in the research work. The paper discusses the image compression using SVD (Singular value Decomposition) method on various images of any type and resolution. A novel method is proposed to preprocess the SVD which reduces the computation complexity compared (in terms of time or clock cycles) to SVD. The proposed method termed as modified SVD (MSVD) is compared with the SVD based on computational complexity and memory requirements. The experiments are conducted on TIDSP DM642 to compare modified SVD with the actual SVD.

Index Terms – **Singular Value Decomposition, compression ratio, computation time, modified SVD, block size, DSP Processor, image compression**

I. INTRODUCTION

Image compression is minimizing the size in bytes of a graphics file without degrading the quality of the image to an unacceptable level. The reduction in file size allows more images to be stored in a given amount of disk or memory space. It also reduces the time required for images to be sent over the internet or downloaded from web pages. A common characteristic of most images is that the neighboring pixels are correlated and therefore contain redundant information. The transform coding methods involves greater computations and hence it is required to reduce the computation complexity. Transform coding is used to convert spatial image pixel values to transform coefficient values.

The different transform coding techniques used for image compression includes Discrete Cosine Transform (DCT), Haar transform, Singular Value Decomposition (SVD), Slant transform, Hadamard transform, Kahrunen Loeve Transform (KLT), etc (Dr.Edel Garcia 2006), (Andrew.B.Watson 1994). (Sindhu.M 2009), (Shivali.D.Kulkarni 2008), (Mr.T.Sreenivasulu Reddy 2007), (Sathish.K.Singh 2010). The suitability of the transform is due to energy compaction property (kamrul Hasn Talukdar 2007). Also, suitability of the transforms is due to subjective quality of the decompressed images in terms of PSNR (Peak signal to Noise Ratio) and quality index, computation time and energy compaction property.

A variation to the SVD based image compression technique is proposed to compress the given input image. The variation can be viewed as a preprocessing step in which the input image is permuted as per a fixed, data independent permutation, after which it is fed to the standard SVD algorithm (Abhiram Ranade 2007). The DCT is used to transform the highly correlated blocks of the YCbCr components, while the SVD is used to transform the low correlated blocks (Y. Wongsawat, 2004). A good compromise between the quality and the compression rate factors that are achieved when processing images by the DCT technique are discussed (A. Messaoudi 2005).

Image Compression is minimizing the size of an image without degrading the quality of the image to an unacceptable level. The reduction in file size allows more images to be stored in a given amount of disk or memory space. It also reduces the time required for images/video to be sent over the internet or downloaded from web pages. The SVD is a fundamental concept in science and engineering, and one of the most central problems in numerical linear algebra. It is also known as principal component analysis (PCA) in statistics and the Karhunen-Loeve (KL) or Hotelling expansion in pattern recognition. The beauty of the SVD is that it provides a robust method of storing larger images as smaller square ones. This is accomplished by representing the original image with each succeeding non-zero singular values. To reduce the storage size even further, one may approximate a "good enough" image with using even fewer singular values. SVD, one of the most useful tools of linear algebra is a factorization and approximation technique which effectively reduces any matrix into smaller invertible and square matrix. SVD is preferred over DCT, Haar and other transforms is due to the suitability of SVD even though matrix is not invertible and non-square. Also DCT, Haar transforms are linear whereas SVD is nonlinear transformation.

The SVD involves the decomposition of an image represented as matrix A in to U, S and V matrices where $UU^T = I, VV^T = I$, I is a identity matrix, the columns of U are orthonormal eigenvectors of AA^T, the columns of V are orthonormal eigenvectors of $A^T A$ and S is a diagonal matrix containing the square roots of Eigen values from U or V in descending order. The columns of U are the left singular vectors, S has singular values and is diagonal and V^T has rows that are the right singular vectors. The SVD represents an expansion of the original data in a coordinate system where the covariance matrix is diagonal. Decomposition of the image into U, S and V matrix is computationally very complex and for reconstruction multiplication of U, S and V matrix is also very complex. Hence method is proposed to minimize the complexity of the algorithm. A modified method proposed for preprocessing the SVD can be estimated in terms of the memory requirements and the computation time required in comparison with the SVD. The novel method of preprocessing reduces the computational complexity and also provides easy way of implementing SVD with reduced block size.

II. RESEARCH METHODOLOGY

Consider an input image represented as matrix A of dimension $M\ X\ N$ where M is the number of rows and N is the number of columns. Total number of pixels present in the image is MN. Hence in most cases, it is required to use reduced SVD compared to full SVD to achieve image compression by reducing spatial redundancies.

The input image without SVD requires MN entries for storage and with SVD input image requires N^2 entries if $M > N$ and M^2 entries if $M < N$. Applying SVD results in $(2M^2 + M)$ values if $M < N$ and $(2N^2 + N)$ values if $M > N$ for an exact representation of the image. Therefore the approximation must be at most rank ${}^{N}/_{(2M+1)}$ if $M < N$ and ${}^{M}/_{(2N+1)}$ if $M > N$ in order to compress the image otherwise storage requirement will increase. Any video frame or image with dimension $M\ X\ N$ has values $M > N$ due to the fact that width is greater that height (aspect ratio). Hence the input image requires N^2 entries and applying SVD results in $(2N^2 + N)$ values and the approximation must be at most rank ${}^{M}/_{(2N+1)}$ in order to compress the image to reduce the storage space.

The SVD process has higher computational needs and hence it is very slow for images of larger resolutions (for higher values of M and N). If the image is broken in to smaller blocks and if each block is handled separately, overall processing time can be reduced to a greater extent.

In the proposed approach, instead of directly applying SVD on the entire image, the image is segmented into blocks of smaller sub images of size $b_l\ x\ b_l$ where b_l is the block size. The algorithm used to compute SVD is then applied onto these sub-images individually. To reconstruct the image, individual sets of U, S and V matrices are used to re-compute the respective b_l x b_l blocks. These blocks are then arranged and re-placed in their original positions to obtain the complete image. The fundamental concept of the SVD based image compression scheme is to use a smaller number of ranks to approximate the original matrix. The lower rank approximation is sufficient to represent the given image in most cases and hence it is possible to achieve a very high compression ratio.

The Rank 'r' approximation to matrix A is given by $A_r = U_r \Sigma_r V_r^T$, where Σ_r is the top-left $r \times r$ sub-matrix of Σ, U_r consists of the first r columns of U, and V_r^T the first r rows of V^T. The SVD decomposition is advantageous because U_r, Σ_r, V_r provide the best rank r approximation to A in the sense of packing the maximum energy from A. Also even with small r, the approximation A_r gets most of the energy of A and hence the attractiveness of the method.

In the proposed approach, the dimensions of the given image are scaled up to the next integral multiples of b_l, by replicating the last row and/or column required number of times or padding the required number of zeros. Further, it is divided into b_l x b_l blocks and SVD is applied on each block independently. SVD of rank 'r' is computed where 'r' is the rank of each block. The total number of elements in the three component matrices for a rank 'r' approximation of one block of dimensions $b_l \times b_l$ is

$$bl * r + r + bl * r = (2 * bl * r + r)$$

Compared to ${b_l}^2$ pixel values of the original block, we are left with $(2 \times b_l \times r) + r$ number of pixels. Thus for an image of dimension $M \times N$ where M and N are the integral multiples of b_l we need to store

$$\frac{M \times N}{b_l \times b_l} \times (2 \times b_l \times r) + r = (m \times n \times 2 \times b_l \times r) + r$$

Where $m = M/b_l$ and $n = N/b_l$

Thus, an equivalent rank 'R' of the entire image would be such that the total number of elements in the three component matrices should equal to the number of elements when rank 'r' approximations of individual blocks of size $b_l \times b_l$ are considered. This implies that

$$(M * R + R + N * R) = \frac{M}{b_l} * \frac{N}{b_l} * (2 * b_l * r + r)$$

$$(M + N + 1)R = \big(m * n * (2 * b_l + 1)\big)r$$

$$R = r \times \frac{(m*n*(2*bl+1))}{(M+N+1)}$$

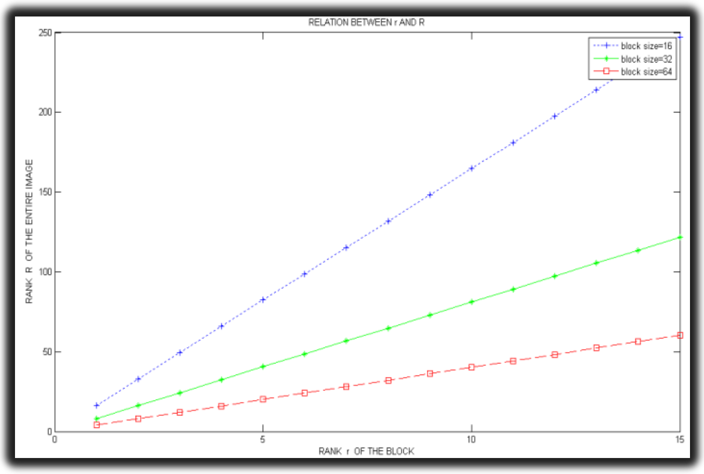

Figure 1: Relation between rank r and R for block sizes of 16, 32 and 64

The figure 1 shows the relation between r and R for different block sizes such as 16, 32 and 64. In the computation of SVD of rank 'R' approximation for the entire image, the rank 'R' varies linearly in relation to rank 'r' approximation for each block of the image as shown in figure 1.

If the dimension $M < N$, the decomposition is done using reduced SVD to obtain the U, S and $V^{T.}$ The compression ratio is given by

$$compression\ ratio = \frac{M*N}{R(2*M+1)}$$

If the dimension $M > N$, the decomposition is done using reduced SVD to obtain the U, S and $V^{T.}$ The compression ratio is given by

$$compression\ ratio = \frac{M*N}{R(2*N+1)}$$

If the dimension $M = N$, the decomposition is done using full SVD to obtain the U, S and $V^{T.}$ The compression ratio is given by

$$compression\ ratio = \frac{M*N}{R(2*N+1)}$$

Table 1: The relation between r and R for different ranks for a block size of 32

Rank r of the block, Block size=32	**Rank R of the entire image**	**Compression ratio**
5	41	3.12
7	57	2.24
9	73	1.75
11	90	1.42
12	98	1.3

Table 2: The relation between r and R for different ranks for a block size of 64

Rank r of the block, Block size=64	Rank R of the entire image	Compression ratio
10	41	3.12
14	57	2.24
18	73	1.75
22	89	1.43
23	93	1.37

The table 1 shows the relation between r and R for different ranks for a block size of 32 and compression ratios and the table 2 shows the relation for a block size of 64 for different ranks with compression ratios. The table shows the result for a PSNR of 30 to 45 dB (PSNR of range 30 to 45 dB indicates very good quality). During the process of reconstruction, the values of U, S and V for each block are used to reconstruct the block and then these $m \times n$ blocks are arranged in the same fashion as they are segmented from the original image matrix. Thus the entire image is reconstructed from the decomposed values. This reconstructed version of the original image is similar to an SVD approximation of rank R applied to the entire image at once.

III. IMPLEMENTATION DETAILS

The experiments are conducted by considering different images of various resolutions. The proposed method is compared with SVD using different parameters. The SVD and the proposed method are ported on a digital signal processor DM642 to estimate the computational complexity and memory requirements (memory required to fit the code). The sample of the result obtained is displayed for the further discussion. Also, for different set of input images, different ranks are considered for the discussion. The experiments are conducted by selecting block size of 2, 4, 8, 16, 32 and 64 for an input image. The similar experiments can be conducted for different resolution.

The comparison parameters considered are MSE and PSNR are given by

Mean Square Error (MSE): The Mean Square Error measures the difference between the frames which is usually applied to Human Visual System. It is based on pixel-pixel comparison of the image frames.

$$d(X,Y) = \frac{\sum_{i-1}^{m,n} {}_{j-1}(X_{i,j} - Y_{i,j})^2}{mn} \quad (1)$$

Peak Signal to Noise Ratio (PSNR): PSNR is measured on a logarithmic scale and depends on the mean squared error (MSE) of between an original and an impaired image or video frame, relative to $(2^n - 1)^2$ (the square of the highest-possible signal value in the image, where n is the number of bits per image sample).

$$PSNR_{db} = 10log_{10}\left(\frac{(2^n-1)^2}{MSE}\right) \quad (2)$$

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

The fundamental concept of the SVD based image compression scheme is to use a smaller number of ranks to approximate the original image. A sample of the result is displayed for an input image of dimension 256 X 256 for block size of 32 and 64.

Case 1: Block size: 32

Table 3: The comparison parameters for rank 5 to 12

Rank	MSE	PSNR	Compression ratio	Elapsed time (sec)
5	46.727	31.435	3.12	0.75001
7	18.633	35.427	2.24	0.98283
9	7.4972	39.38	1.75	1.4136
11	2.8986	43.508	1.42	1.8890
12	1.7214	45.772	1.3	2.0299

For block size of 32 for the given set of input image, rank 5 to 12 gives a PSNR of 30 to 45 dB. The table 3 shows the output result in terms of objective measures for a block size of 32. The different parameters are selected to measure the quality of the images for different ranks.

Case 2: Block size: 64

Table 4: The comparison parameters for rank 8 to 22

Rank	MSE	PSNR	Compression ratio	Elapsed time (sec)
10	39.7367	32.138	3.12	0.5905
14	15.9475	36.103	2.24	0.6997
18	5.9818	40.362	1.75	0.8034
22	2.2134	44.680	1.43	1.0035

For block size of 64 for the given set of input image, rank 8 to 22 gives a PSNR of 30 to 45 dB. The table 4 shows the output result in terms of objective measures for a block size of 64. The different parameters are selected to measure the quality of the images for different ranks.

Table 5: Comparison parameters for r = 33 to 90 to get PSNR 30 - 45 dB

Rank	MSE	PSNR	Compression ratio	Elapsed time (sec)
35	57.019	30.570	3.6500	0.4243
50	24.169	34.298	2.5550	0.6572
70	7.6763	39.279	1.8250	0.9276
90	2.0876	44.934	1.4194	1.1340

The experiments are conducted for a given input image by using direct SVD and the table 5 shows the result of direct SVD for different rank to achieve a PSNR of 30 to 45 dB.

Table 6: Comparison of modified SVD with SVD for PSNR=30 dB

Block size	Rank	Compression ratio	computation time (sec)	Memory Usage
Modified SVD Block size 32	5	3.12	16	10604
Modified SVD Block size 64	8	3.87	10	11920
Direct SVD	33	3.87	24	10316

Computation complexity and memory complexity is found by porting proposed algorithm and direct SVD on TIDSP DM642. Table 6 shows the profiling result of proposed method and direct SVD obtained for the given image. Table 6 indicates that the computation time (computation complexity) can be reduced by block processing of SVD as it consumes 0.666 times that of direct SVD to achieve a PSNR of 30 dB and gives a compression ratio of 3.12 for each block if block size is 32. To achieve a PSNR of 30 dB, rank of 5 is required if the block size is 32 and 8 is required if the block size is 64. A rank of 33 is required if direct SVD is employed which gives a compression ratio of 3.87. Hence decrease in the block size, also decreases the rank but the computation time required for computation increases. Selection of block size is very important criteria in modified SVD. Parallel implementation of reduced block size is possible on TI DSP which reduces the computation time further. The proofing result gives the estimation of clock cycles and computation time for different function used in implementation of proposed method and the direct SVD.

V. SUMMARY AND CONCLUSIONS

The experimental results indicate that the proposed method performs better than the SVD in terms of computation complexity by reducing the computational time. Also the proposed method can be modified to achieve higher compression ratio. The methodology is discussed as follows to achieve a higher compression ratio. For $n \, x \, n$ block with rank R requires $2nR$ elements. If the block is split in to four quarters block of size ${}^{n}/_{2} \; X \; {}^{n}/_{2}$ with each sub block having rank R_1, R_2, R_3, R_4 respectively, the number of storage element is

$n(R_1+R_2+R_3+R_4)$

$$n(R_1 + R_2 + R_3 + R_4) < 2nR$$

Hence $(R_1 + R_2 + R_3 + R_4) < 2R$

The key to SVD compression is to use low rank approximation to the image. For the less complicated images, lower rank is sufficient to accurately represent it. For highly complicated images, higher rank is necessary for accurate representation.

REFERENCES

1. H. S. Prasantha, H. L. Shashidhara, and K. N. Balasubramanya Murthy. Image compression using SVD. In Proceedings of the International Conference on Computational Intelligence and Multimedia Applications, pages 143–145. IEEE Computer Society, 2007.
1. Abhiram Ranade, *et. al.*, "A Variation on SVD Based Image Compression", Image Vision Computation", 25(6), 20017, pp.771-777, 2007
2. Messaoudi, *et. al.*, "A Good Compromise between Images Quality and Compression Rate of the DCT Technique", GVIP Conference, CICC, Cairo, Egypt., 2005
3. Dr. Edel Garcia, "Singular Value Decomposition (SVD)" A Fast Track Tutorial Published on September 11 2006.
4. Niranjan Damera Venkata*, "*Image Quality Assessment Based on a Degradation Model", IEEE transactions on image processing, Vol. 9, No. 4, 2000.
5. Satu Jumisko, *et .al.*, "Evaluation of Subjective Video Quality of Mobile Devices", MULTIMEDIA '05, Proceedings of the 13th annual ACM international conference on Multimedia, 2005
6. Gunasheela K S, H S Prasantha, "Satellite image compressiondetailed survey of the algorithms", Proceedings of ICCR in LNNS Springer, 2017, vol. 14, pp. 187-198.
7. Raghavendra.M.J, Prasantha.H.S and S.Sandya, "Image Compression Using Hybrid Combinations of DCT SVD and RLE", International Journal of Computer Techniques, Volume 2 Issue 5-2015.
8. Gunasheela K S, H S Prasantha, "Compressive sensing for image compression: survey of algorithms", Proceedings of Emerging Research in Computing, Information, Communication and Applications, ERCICA, Springer publication, Bengaluru, 2018
9. K N Shruthi, B M Shashank, Y. SaiKrishna Saketh, H.S Prasantha and S. Sandya, "Comparison Analysis Of A Biomedical Image For Compression Using Various Transform Coding Techniques", IEEE, pp. 297-303, 2016
10. Raghavendra.M.J, Prasantha.H.S and S.Sandya, "DCT SVD Based Hybrid Transform Coding for Image Compression", International Journal of Recent and Innovative Trends in computing and communication. 2015
11. Prasantha H.S., Shashidhara H.L, Balasubra manya Murthy, K N: Comparative analysis of different interpolation schemes in image processing. In: International Conference on Advanced Communication Systems (ICACS), India, pp. 17–24 (2007)
12. Prasantha H.S, Shashidhara H.L and K.N.Balasubramanya Murthy, "Image Scaling comparison using Universal Image Quality Index", International Conference on Advances in computing, control and Telecommunication Technologies (ICACCTT), India, pp.859-863, Dec 2009.