SVD BASED IMAGE COMPRESSION

K. Mounika, D. Sri Navya Lakshmi, K. Alekya

Under the guidance of M.R.N Tagore, Associate Professor

ECE Dept., VVIT, mouni.kalamraju@gmail.com, 9652952453

Abstract- Image compression techniques are the most concerned topics in today's technological developments. Singular Value Decomposition (SVD) is one such image compression technique. This SVD performs its operations on matrices. In this paper, we will discuss how SVD is applied on images, the methodology of image compression using SVD and also the algorithm to compress an image using MATLAB.

Keywords-Image compression, Singular Value Decomposition, Image processing, image as a matrix, image processing, face recognition

INTRODUCTION

Now a day, everyone is fond of selfies. Not only selfies, man wishes to capture all his memorable events. This results in the increase of number of images and videos. It is obvious that a more amount of memory is needed to store all these images and videos. If these images are needed to be transmitted, it even requires large bandwidth. So, there comes the need of image compression techniques. These image compression techniques reduce the storage space occupied by the image without any loss to image quality. Thus the image size can be reduced by selecting proper compression technique depending on the requirement of user or application.

IMAGE COMPRESSION TECHNIQUES

Image compression is a technique in which the storage space of image is reduced without degrading the image quality. This is classified into two types.

Lossy image compression:

In this, the image is compressed such that there is loss in image data, that is, image cannot be reconstructed if once compressed. This technique is best suited for normal photographs where a small loss of fidelity is acceptable. Most of the regular image compression techniques used today are lossy techniques.

SVD is also a lossy image compression technique.

Lossless image compression:

In this, the compressed image is same as that of the original input image. Here, image once compressed can be reconstructed. It is reversible process. This technique is best suited for medical applications etc.,

REPRESENTING IMAGE AS A MATRIX

Every image is represented by pixels. Pixels represent the intensity of image. These pixel values are arranged as a matrix. The matrix representation of an image can be easily obtained using MATLAB.

Syntax for displaying matrix representation of an image is

```
I=imread ('1.jpeg');   % reads input image.
Disp(I);               % displays pixel values of image (as a matrix, arranged in rows and columns)
```

SVD TECHNIQUE

Let A be an $m \times n$ matrix. Performing SVD to A factorizes it into a product of orthogonal matrix, diagonal matrix and another orthogonal matrix.

$$A = USV^T$$

Where,

A is image matrix

U is $m \times m$ matrix

S is $m \times n$ matrix

V is $n \times n$ matrix

Singular Value Decomposition technique splits given matrix into a product of orthonormal matrices and a diagonal matrix. The procedure to perform SVD theoretically is as follows

Find eigen values of the image matrix. Obtain singular values (square root of eigen values).

Place singular values in decreasing order as a diagonal matrix, S matrix

Using image matrix, say A, obtain AA^T and $A^T A$.

Find the eigen vector of above matrices. These vectors become columns of U and V matrices.

Now, using S, U and V matrices, represent A matrix.

IMAGE COMPRESSION USING SVD

After obtaining U, S and V values using above steps,

Eliminate the unnecessary singular values in S matrix.

Obtain compressed image A with the new diagonal matrix obtained after removing some singular values.

MATHEMATICAL ANALYSIS

Let image matrix be $A_{m\times n}$.

Using SVD, A can be represented as

$$A = USV^T$$

$$A = \begin{bmatrix} u_1 & u_2 & \cdots & u_m \end{bmatrix} \begin{bmatrix} s_1 & 0 & \cdots & 0 \\ 0 & s_2 & \cdots & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & s_n \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_n^T \end{bmatrix}$$

The values $s_1 > s_2 > \cdots > s_n > 0$ and as the last values of S are approximately equal to zero, they can be removed. After removing those values resultant A matrix can be represented as

$$A = US^1V^T$$

$$A = \begin{bmatrix} u_1 & u_2 & \cdots & u_m \end{bmatrix} \begin{bmatrix} s_1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & & & & \vdots \\ \vdots & & s_r & & & \vdots \\ \vdots & & & 0 & & \vdots \\ \vdots & & & & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ \\ \vdots \\ v_n \end{bmatrix}$$

In the above matrix, S values after r terms are approximated to zero. So multiplication of the terms greater than r will be zero.

If m=n, the above matrix can be represented as

$$A = [\, s_1u_1 \;\; s_2u_2 \;\ldots\; s_ru_r \;\; 0 \ldots . \;\; 0 \,] \begin{bmatrix} v_1{}^T \\ v_2{}^T \\ \vdots \\ v_n{}^T \end{bmatrix}$$

$$= s_1u_1v_1{}^T + s_2u_2v_2{}^T + \ldots.. + s_ru_rv_r{}^T.$$

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

We know that, rank of a singular matrix is equal to the number of non zero singular values. The rank of above matrix will now be obviously reduced as the number of S values is approximated to 'r' terms.

Therefore, size of the matrix is reduced, which in turn reduces the memory occupied by the image.

Thus,

- From the above analysis, the matrix A can be approximated by adding only the first few terms (r terms) of the series. As r increases, the image quality increases, but at the same time, the amount of memory needed to store the image also increases.
- Optimum r value should be selected such that there is no damage to image quality and at the same time storage space occupied by image is reduced. Thus, selection of r value plays an important role in performance of Singular Value Decomposition (SVD) technique.

IMPLEMENTING SVD BASED IMAGE COMPRESSION USING MATLAB

METHODOLOGY USED

Initially the JPEG image which has to be compressed is given as an input. This input image is stored as an array of integers. Required 'r' value should be specified. Compression is then achieved by performing Singular value decomposition (SVD) on RGB components of the input JPEG image. The resultant decomposed matrix is regenerated after approximating S matrix.

SVD FUNCTIONS

Singular value decomposition of symbolic matrix can be easily done using MATLAB build in function 'svd'. This function decomposes the given matrix into three matrices.

Syntax

```
sigma = svd(X)
[U,S,V] = svd(X)
[U,S,V] = svd(X,0)
[U,S,V] = svd(X,'econ')
```

ALGORITHM

Step-1:

Read the image (input image).

syntax:

```
img=imread('filename.jpg');
```

Step-2:

Split the input image (colour image) into R, G, B channels.

Syntax:

```
red = img(:,:,1); % Red channel
green = img(:,:,2); % Green channel
blue = img(:,:,3); % Blue channel
```

Step-3:

Decompose each component using Singular Value Decomposition

Syntax:

```
[u,s,v]=svd(I);
```

Step-4:

Select r value and discard the diagonal value of S matrix not required.

Construct the image using the selected singular values.

Syntax:

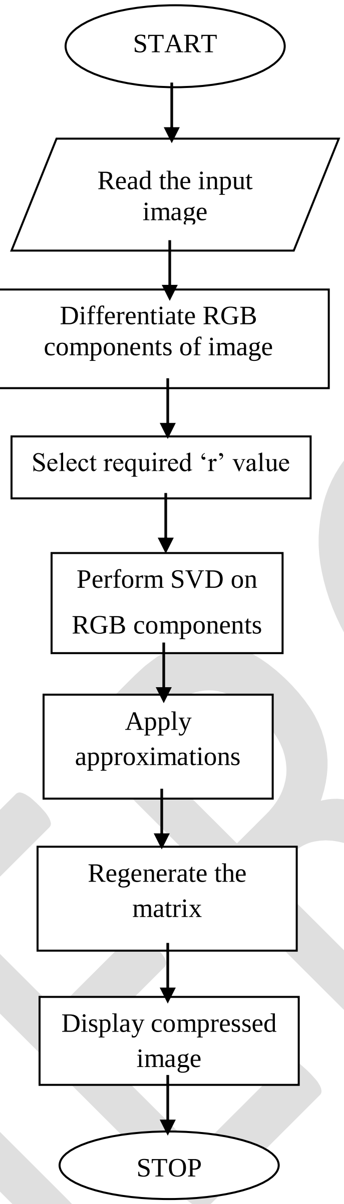
```
for j=1:r
    c=c+s(j,j)*u(:,j)*v(:,j).';
end
```

- The r-value in the m-file represents the number of iterations taken on each layer used in the resulting decomposition. This is actually the rank of the SVD matrix. By increasing the rank we can increase clarity until an optimal image is reached.

Step-5:

Display the compressed image.

FLOW DIAGRAM



MEMORY UTILIZATION

Let $A_{m \times n}$ be an image matrix. This image matrix will contain a total of $m \times n$ pixels. Assume, each pixel will occupy a memory location. So, the input image occupies mn memory locations. This can be mathematically shown as,

$$A_M = mn$$

After performing SVD,

U matrix is of size $m \times m$

V matrix is of size $n \times n$

With r approximations,

U matrix is of size $m \times r$

V matrix is of size $n \times r$

According to definition of SVD,

The image matrix can be represented as a product of orthogonal matrix times a diagonal matrix times another orthogonal matrix.

Thus,

$$A = USV^T$$

So, total memory locations occupied by this decomposed matrix with r approximations is

$$A_M = U_M + V_M + S_M$$

$$A_M = mr + nr + r$$

$$A_M = r(m + n + 1)$$

For high resolution images,

$$mn >> r(m + n + 1)$$

Hence, memory occupied is reduced ie., image is compressed.

APPLICATIONS

SVD approach can be used in image processing, image compression, face recognition, water marking, data retrieval etc., SVD is most widely used in face recognition, noise reduction in images, image de-blurring, signal processing etc., Research is still going on different applications of SVD on digital image processing.

RESULTS

Outputs for r=70 for given image

The size of input image is 768×1024. It occupies 6291456 bytes.

After compression, the resultant image occupies 2359296 bytes.

Clearly, image is compressed without any loss to image quality.

Input image

Output for r=70

Outputs for different r values

Input image

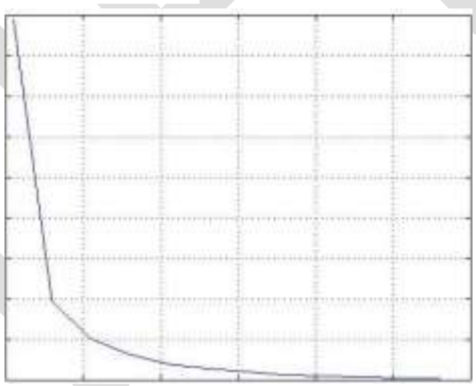
with r=5 values

with r=10 values

with r=30 values

with r=50 values

Error between compressed image and original image



r values

graph between r values and error between compressed image and original image

CONCLUSION

Singular Value Decomposition (SVD) is a simple, robust and reliable technique. This SVD technique provides stable and effective method to split the image matrix into a set of linearly independent matrices. SVD provides good compression ratio and also a practical solution to image compression problem. The results shown above clearly displays the compressed outputs for different r values. Thus, selection of r value plays a crutial role in this SVD based image compression technique.

REFERENCES:

[1] Using the Singular Value Decomposition (SVD) for Image Compression by TEAV Vuthy –Royal University of Phnom Penh.

[2] Golub, Gene H.; Kahan, William (1965). "Calculating the singular values and pseudo-inverse of a matrix". *Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis* 2.

[3] L. David, Linear Algebra and It's Applications, Addison-Wesley, NY, edit. 7.

[4] Singular Value Decomposition Applied To Digital Image Processing by Lijie Cao – Arizona State University Polytechnic Campus, Arizona.

[5] The Singular Value Decomposition and It's Applications in Image Processing irtfweb.ifa.hawaii.edu

[6] SVD and PCA in Image Processing- Wasuta-Renkjumnong, Georgia State University,2007. scholarworks.gsu.edu/math_theses

[7] . David, Linear Algebra and It's Applications, Addison-Wesley, NY, edit. 7.

[8] Abrahamsen and Richards, Image Compression Using Singular Value Decomposition, December 14, 2001 <http://online.redwoods.cc.ca.us/instruct/darnold/laproj/Fall2001/AdamDave/textwriteup.pdf>

[9] Arnold, An Investigation into using Singular Value Decomposition as a method of Image Compression, September 2000 <http://www.cs.bgu.ac.il/~na031/Arnold.pdf>

[10] Buhr and White, Digital Image Compression via Singular Value Decomposition, May 3, 2006

[11] M. Berry, S. Dumais and G. O'Brien. *Using linear algebra for intelligent information retrieval*. SIAM Review, 37(4):573--595, 1995.

[12] An svd based grayscale image quality measurement by shnayderman, guesev, esjucuigky