IMAGE COMPRESSION USING SINGULAR VALUE DECOMPOSITION

INTRODUCTION

- Image processing is any form of information processing in which the input is an **image**.
- Image processing studies how to transform ,store ,retrieval the image.

Digital Image Processing

- **An image can be defined as a two dimension function f (x, y) (2D-image).**
- **where x and y are spatial coordinates, and the amplitude of f at any pair of (x, y) is gray level of the image at that point.**
- **When x, y and the amplitude value of f are finite, discrete quantities, the image is called**

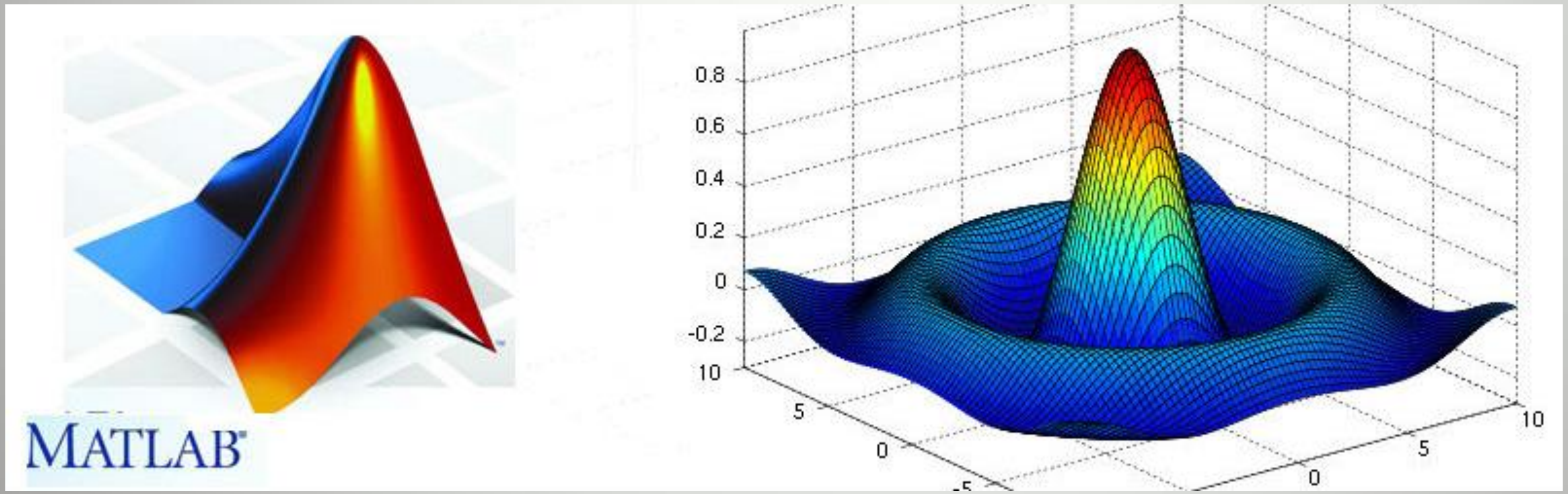
"a digital image".

- **The finite set of digital values is called picture elements or pixels. Typically, the pixels are stored in computer memory as a two dimensional array or matrix of real number.**

Objective of the Project

- The objective of this project is to apply linear algebra “Singular Value Decomposition (SVD)“ to midlevel image processing.

- MATLAB is used as a platform of programming and experiments in this project, since MATLAB is a high-performance In integrating computation, visualization and programming.



Singular Value Decomposition (SVD)

- Singular Value Decomposition (SVD) is said to be a significant topic in linear algebra by many renowned mathematicians. SVD is a factorization of matrix .
- SVD has many practical and theoretical values; special feature of SVD is that it can be performed on any real (m, n) matrix.
- Let's say we have a matrix **A** with m rows and n columns, with rank r and r ≤ n ≤ m. Then the A can be factorized into three matrices: A = (USV^T)
- Where Matrix U is an m × m orthogonal matrix
- And matrix V is an n × n orthogonal matrix
- Here, S is an m × n diagonal matrix with singular values (SV)

on the diagonal. The matrix S can be showed in following

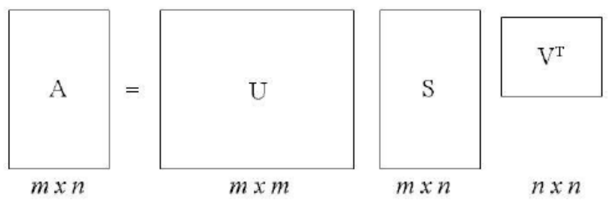


Figure1. Illustration of Factoring *A* to *USV^T*

$$S = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \sigma_{r+1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & \sigma_n \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix}$$

SVD Approach for Image Compression

- Image compression deals with the problem of reducing the amount of data required to represent a digital image.

- Compression is achieved by the removal of three basic data redundancies:

1) coding redundancy, which is present when less than optimal
2) interpixl redundancy, which results from correlations between the pixels
3) psychovisual redundancies, which is due to data that is ignored by the human visual

Image Compression Using SVD

➢Introduction

We often need to transmit and store the images in many applications. Smaller the image, less is the cost associated with transmission and storage. So we often need to apply data compression techniques to reduce the storage space consumed by the image. One approach is to apply Singular Value Decomposition (SVD) on the image matrix.

In this method, digital image is given to SVD. SVD refactors the given digital image into three matrices.

Singular values are used to refactor the image and at the end of this process,

image is represented with smaller set of values, hence reducing the storage space required by the image.

Goal here is to achieve the image compression while preserving the important features which describe the original image.

SVD can be adapted to any arbitrary, square, reversible and non-reversible matrix of $m \times n$ size.

Compression ratio and **Mean Square Error** is used as performance metrics.

- When an image is SVD transformed, it is not compressed, but the data take a form in which the first **singular value** has a great amount of the image information. With this, we can use only a few singular values to represent the image with little differences from the original. To illustrate the SVD image compression process, we show detail procedures:

$$A = USV^T = \sum_{i=1}^{r} \sigma_i \mathrm{u}_i \mathrm{v}_i^T$$

- That is A can be represented by the outer product expansion:

$$A = \sigma_1 \mathrm{u}_1 \mathrm{v}_1^{\ T} + \sigma_2 \mathrm{u}_2 \mathrm{v}_2^{\ T} + \cdots + \sigma_r \mathrm{u}_r \mathrm{v}_r^{\ T}$$

- When compressing the image, the sum is not performed to the very last SVs, **the SVs with small enough values are dropped**. (Remember that the SVs are ordered on the diagonal.) The **closet matrix** of rank **k** is obtained by truncating those sums after the first k terms:

$$A_k = \sigma_1 \mathrm{u}_1 \mathrm{v}_1^{\ T} + \sigma_2 \mathrm{u}_2 \mathrm{v}_2^{\ T} + \cdots + \sigma_k \mathrm{u}_k \mathrm{v}_k^{\ T}$$

Image Compression Measures

- To measure the performance of the SVD image compression method, we can computer the compression factor and the quality of the compressed image. Image compression factor can be computed using the Compression ratio:

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

- To measure the quality between original image A and the compressed image AK , the measurement of Mean Square Error (MSE) can be computed:

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

Image Comperssion Using SVD

➢Applying SVD In Image Compression

- Let A be the matrix of the given image. It can be expressed as
- $A = USVT$; summation ranging from 1 to r.
- Expanding the summation,
- $A = \mu1u1v1T + \mu2u2v2T + \ldots + \mu rurvrT$
- As we have seen, it is not necessary to compute the summation to the very last of the values in the

matrices. Smaller values are dropped before the summation. After truncating the matrix, we get the following summation:

- $Ak = \mu1u1v1T + \mu2u2v2T + \ldots + \mu kukvkT$

Storage required by the reduces matrix Ak is k(m+n+1) units. Value of k will be closer to n, but less than n. Hence resulting image will have a good resolution and it is not distorted too much. It is evident that storage space required for the compressed image depends on the value of k. So the value of k can be adjusted as per the requirement. This process is illustrated in Figure 2

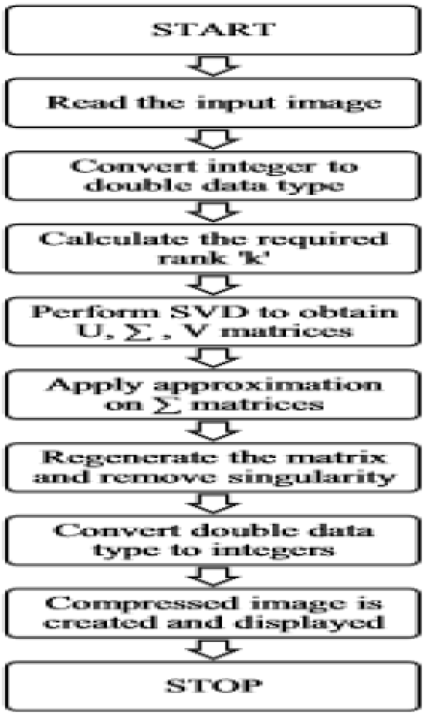


Figure 2. Image compression using SVD

Image Compression Using SVD

➢Experiment

Figure 4 shows the compression of images with different values of k

Figure 4. Compressions for different values of k.

Image Compression Using SVD

➢Results

Compression ratio: It is the ratio of memory required to store the real image to the memory required to store new image (changed).it calculates the extent to which pixels in image are changed (compressed). It is calculated as CR = m×n/(k×(m+n+1)).

Table 1. Compressed image sizes corresponding to different values of k.

K	CR	Size of compressed image(in Kb)
5	174.55	106.6
10	87.27	124.3
40	21.81	159
80	10.9	176.1
100	8.72	180
800	1.09	215.8

Image Compression Using SVD

➢Observations OF The Experiment :

1- Using less singular value (smaller K), **the better compression ratio** is achieved

2- However, the more singular value is used (larger K), quality measurement MSE is smaller (**better image quality**), and the reconstructed images are more equal to the original, but using more **storage space**.