ROYAL UNIVERSITY OF PHNOM PENH

# Using the Singular Value Decomposition (SVD) for Image Compression

*Author:*
TEAV Vuthy
(teavvuthy@yahoo.com)

*Advisor:*
DR. ANGEL R. Pineda
(apineda@fullerton.edu)

September 25, 2010

# Contents

# Abstract

Any $m \times n$ matrix can be factored into the product of an orthogonal matrix times a diagonal matrix times another orthogonal matrix. This is called the Singular Value Decomposition (SVD) of a matrix.

The goal of studying the SVD of a matrix is to create approximations of the full $m \times n$ matrix by only using some of the terms of the diagonal matrix in the decomposition. This approximation of the full matrix is the basis of image compression using SVD, since images can be viewed as matrices with each pixel being an element of a matrix.

In our project, we will prove the theorem of Singular Value Decomposition (SVD), and compute the SVD of a matrix example by calculation and by using *Octave* and MATLAB. We will also explain how the SVD can be applied to compress images, and implement the image compression algorithm developed for a sample image by using *Octave* and MATLAB.



Key Words: SVD, Image Compression.


# 1 Introduction

Among the methods to write a matrix as a product of matrices, Singular Value Decomposition (SVD) is a very useful method [1, 2]. The SVD is applied to the image compression. For example, suppose that a satellite in space is taking photographs of Jupiter to be sent back to the earth. The satellite digitalises the picture by subdividing it into tiny squares called pixels or picture elements. Each pixel is represented by a single number that records the average light intensity in the square.

If each photograph was divided into 400×400 pixels, it would have to send 160,000 numbers to the earth for each picture. This would take great deal of time and would limit the number of photographs that could be transmitted. It is much better, If we can approximate this matrix with a matrix which requires less storage [2].

Suppose we know the SVD of a matrix. The key is in the singular values ( in $\Sigma$). Singular values are arranged in decreasing order so the first singular value is big and the next after is smaller and smaller. Then some singular values are extremely small. In fact, we have formula $\mathrm{A} = U\Sigma V^T$ with U and V are orthogonal matrices and $\Sigma$ is a diagonal matrix. We can do the multiplication as columns times rows :

$$\text{We get A} = U\Sigma V^T = u_1\sigma_1 v_1^T + u_2\sigma_2 v_2^T + \cdots + u_r\sigma_r v_r^T \ (\bullet).$$

Here the u's are columns of U and the v's are columns of V. Any matrix is the sum of r matrices of rank one.

Since the singular values ($\sigma$'s) are arranged in decreasing orders, some $\sigma$'s are extremely small so some terms in ($\bullet$) above are also extremely small and can be

ignored. If we ommit some terms that are very small from (•), we get a matrix that has less storage than matrix A.

If only 30 terms are kept, we send only 30 times 80 numbers instead of 160,000. First, we hardly recognize the picture but as more and more singular values are included, the quality of picture is more and more better.

# 2 Background

## 2.1 Theorem: Singular Value Decomposition (SVD)

Let A be an $m \times n$ matrix with rank r. Then there exists an $m \times n$ matrix

$\Sigma$ where $\Sigma = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}$,

with (m-r) rows having all zero entries, (n-r) columns having all zero entries and D is an $r \times r$ diagonal matrix for some r not exceeding the smaller of m and n.
The diagonal entries in D are the first r singular values of A: $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ and there exist an $m \times m$ orthogonal matrix U and an $n \times n$ orthogonal matrix V such that A = $U\Sigma\ V^T$ [1].

## Proof:

Since we want to get $A^TA = (V\Sigma^TU^T)(U\Sigma\ V^T) = V\Sigma^T\Sigma\ V^T$ and on the other hand, we have $A^TA = (A^TA)^T$ so $A^TA$ is a symmetric matrix then by the definition and theorems of a symmetric matrix implies that $A^TA$ has a complete set of orthonormal eigenvectors $v_i$ which go into the columns of $V$.
( Refer to the reference on pages 405-409 0f [1]).

First, we take { $v_1$, $v_2$, ...,$v_n$ } to be an orthonormal basis of $\mathbb{R}^n$
of eigenvectors of $A^TA$.

And second, let $\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_n$ be associated eigenvalues of $A^TA$. Then, for $1 \leq i \leq$ n , we have:

$$\| Av_i \|^2 = (\ Av_i)^T(Av_i)$$

$$= v_i^TA^TA\ v_i$$

$$= v_i^T(\ A^TA\ v_i\ )$$

$$= v_i^T(\lambda_i v_i), \text{ by the definition of } v_i \text{ being the eigenvectors of } A^TA$$

( $A^TAv_i = \lambda_i v_i$ ).

$$\| Av_i \|^2 = \lambda_i v_i^T v_i$$

$= \lambda_i$, since $v_i$ is a unit eigenvector of $A^TA$ then $v_i^T v_i = 1$.

We get $\| Av_i \|^2 = \lambda_i$ (1)

In general, the singular values of a matrix A are the square roots of the eigenvalues of $A^TA$, denoted by $\sigma_1$, $\sigma_2$,...,$\sigma_n$, and they are arranged in decreasing order. That is $\sigma_i = \sqrt{\lambda_i}$ for $1 \leq i \leq n$. By(1), the singular values of A are the lengths of vectors $Av_1, Av_2, Av_3, \ldots, Av_n$.

Third, let $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \cdots \geq \sigma_r > 0$ be the first r singular values of A ( $A$ is an m $\times$ n matrix), and the next singular values of A are $\sigma_{r+1} = \sigma_{r+2} = \sigma_{r+3} = \cdots = \sigma_n = 0$. The singular values of A denoted by $\sigma_i = \sqrt{\lambda_i}$ for $1 \leq i \leq r$.

From (1) we get $\sigma_i = \sqrt{\lambda_i} = \| Av_i \| > 0$, since $\sigma_i > 0$ for $1 \leq i \leq r$, $\sigma_i = 0$ for $i \geq r+1$, and then we also get $Av_i = 0$ for $i \geq r+1$.

Since $AV_i$ and $AV_j$ are orthogonal for i≠j,
(Because for i≠j, $v_i$ and $v_j$ are orthogonal, we get $(Av_i)^T(Av_j) = v_i^T A^T \mathrm{A} v_j = v_i^T(\lambda_i v_j) = 0$ thus $AV_i$ and $AV_j$ are orthogonal), and for any y in Col(A), we may write as y = Ax and we may also write x= $c_1v_1 + c_2v_2 + \cdots + c_nv_n$, and

$$y = Ax$$

$$= c_1Av_1 + c_2Av_2 + \cdots + c_rAv_r + c_{r+1}Av_{r+1} + \cdots + c_nAv_n$$

$$= \mathrm{c}_1Av_1 + c_2Av_2 + \cdots + c_rAv_r + 0 + \cdots + 0 \qquad (\mathrm{Av}_i = 0 \text{ for } i \geq r+1)$$

Thus y is in span $\{ Av_1, Av_2, \cdots, Av_r \}$, then

$\{ Av_1, Av_2, \cdots, Av_r \}$ is an orthogonal basis for Col(A).

For $1 \leq i \leq r$, define $u_i = \frac{1}{\|Av_i\|} Av_i$

$$u_i = \frac{1}{\sigma_i} Av_i, \text{ since } \sigma_i = \| Av_i \|$$

Then we get $\sigma_i u_i = Av_i$ ( $1 \leq i \leq r$ ) (2)
then $\{ u_1, u_2, u_3, \ldots, u_r \}$ is an orthonormal basis of column space of A.

By Gram-Schmidt, these r orthonormal u's can be extended to a complete orthonormal basis $u_1, u_2, u_3, \ldots, u_m$ then we get an orthonormal basis set $\{ u_1, u_2, u_3, \ldots, u_m \}$ of $\mathbb{R}^m$.

Finally, let U = [ $u_1$ $u_2$ $\cdots u_m$ ] and V = [ $v_1$ $v_2$ $\cdots v_n$ ], then U and V are orthogonal matrices.

Also from (2) we get AV = [ $Av_1$ $Av_2$ $\cdots Av_n$ $0 \cdots 0$ ],

( since we have rank(A) = r ).

$$= [\ \sigma_1 u_1 \quad \sigma_2 u_2 \quad \cdots \sigma_r u_r \quad 0 \cdots 0\ ]$$

Let D be the diagonal matrix with the diagonal entries $\sigma_1, \sigma_2, \sigma_3, \cdots, \sigma_r$, and

$$\text{let } \Sigma = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}$$

$$\text{Then } U\Sigma = [\ u_1 \quad u_2 \quad \cdots u_m\ ] \begin{bmatrix} \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_r \end{bmatrix} & 0 \\ 0 & 0 \end{bmatrix} \quad (3)$$

$$= [\ \sigma_1 u_1 \quad \sigma_2 u_2 \quad \cdots \sigma_r u_r \quad 0 \cdots 0\ ] = AV$$

Then implies $U\Sigma V^T = AVV^T = A$,

( since V is orthogonal matrix, then $VV^T = \mathrm{I}$ )

**Therefore** $A = U\Sigma V^T$ (4)

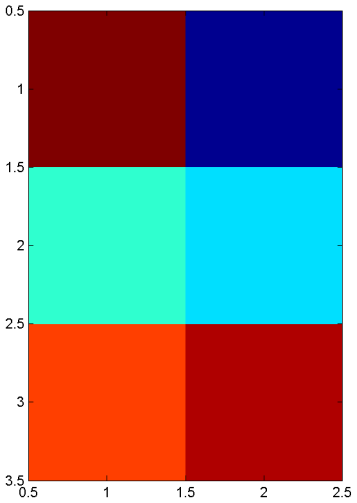
• Specific examples are carried out in Appendix A.

## 2.2 Images as matrices

Each image contains pixels. Pixels are small adjoining squares in a matrix across the length and width of the digital image. Pixels are so small that we don't see the actual pixels when the image is on our computer monitor. If we have an image with m×n pixels (or m×n cells), then we can view the image as a matrix with m rows and n columns because the m×n matrix has m×n elements too and each pixel of the image represents each element of the matrix.

**For example :**

We have a very simple image as the following :
(The image below is obtained by reading its matrix as image in MATLAB using codes in Appendix B)




The image above, we saw that it has 6 cells or 6 pixels and it is the 3×2 image so we can view the image as the 3×2 matrix which means that the matrix contains 6 elements too. By using the Matlab codes for viewing matrix as image in Appendix B, we have the matrix that represents our image above as the following

$$\begin{pmatrix} 0.9572 & 0.1419 \\ 0.4854 & 0.4218 \\ 0.8003 & 0.9157 \end{pmatrix}$$

# 3 Methods

For the method of image compression, we use the programs in Appendix B and, we also use the property and formulas as the following :

## 3.1 Property of the SVD

The m×n matrix A can be written as the sum of rank-one matrices.

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T \qquad (5)$$

Where r is the rank of A, and $u_i$ and $v_i$ are the ith columns of U and V, respectively [3].

### Proof:

By formula (3) in the proof of SVD theorem above, we get:

$$\mathrm{A} = U\Sigma V^T = [\ u_1\ u_2\ ...\ u_r\ ...\ u_m\ ] \begin{bmatrix} \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_r \end{bmatrix} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_r^T \\ \vdots \\ v_n^T \end{bmatrix}$$

$$= [\ \sigma_1 u_1 \quad \sigma_2 u_2 \quad \cdots \sigma_r u_r \quad 0 \cdots 0\ ] \begin{bmatrix} v_1^T \\ \vdots \\ v_r^T \\ \vdots \\ v_n^T \end{bmatrix}$$

$$= \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_r u_r v_r^T.$$

$$Thus : A = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

This property is known as the low-rank approximation property of the SVD. We get the best least squares approximation to A of rank $k \leq r$ by keeping only the first k terms of $\mathrm{A} = \sum_{i=1}^{r} \sigma_i u_i v_i^T$ and the other terms left are ignored [3].

## 3.2 Dimension reduction

What is the purpose of transforming the matrix A into $U\Sigma V^T$ ? We want to approximate the $m \times n$ matrix A by using far fewer entries than in the original matrix. By using the rank of a matrix, we remove the information that is not needed (the dependent entries) when $r \leq m$ or $r \leq n$.

$$\mathrm{A} = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_r u_r v_r^T + \sigma_{r+1} u_{r+1} v_{r+1}^T + \cdots$$

Since the singular values are always greater than zero. Adding on the dependant terms where the singular values are equal to zero does not effect the image. Remove the terms at the end of the equation zero out, leaving us with:

$$\mathrm{A} = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_r u_r v_r^T. \qquad (5)$$

One way to compress the image A is to approximate A by a matrix of smaller rank.

If $k < r$ then the closest approximation to A, ( rankA = r) - by a matrix of rank k that is the truncation of (5) to the first k terms :

$$A \approx \mathrm{A}_k = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_k u_k v_k^T \qquad (6)$$

**Proof:**

We have $\mathrm{A} = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_k u_k v_k^T + \sigma_{k+1} u_{k+1} v_{k+1}^T + \cdots + \sigma_r u_r v_r^T$

and $A_k = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_k u_k v_k^T$

then $A - A_k = \sigma_{k+1} u_{k+1} v_{k+1}^T + \cdots + \sigma_r u_r v_r^T$

and $\| A - \mathrm{A}_K \| = \sqrt{\sigma_{k+1}^2 u_{k+1}^2 (v_{k+1}^T)^2 + \cdots + \sigma_r^2 u_r^2 (v_r^T)^2}$

since $u_i^2 = \| u_i \|^2 = 1$, $(v_i^T)^2 = \| v_i^T \|^2 = 1$ (u and v are unit vectors).

Then $\| A - \mathrm{A}_K \|^2 = \sigma_{k+1}^2 + \sigma_{k+2}^2 + \cdots + \sigma_r^2$.

$$\| A - \mathrm{A}_K \| = \sqrt{\sigma_{k+1}^2 + \sigma_{k+2}^2 + \cdots + \sigma_r^2} \leq (r-k)\sigma_{k+1}$$

Since the sigular values are arranged in dereasing order, when k goes nearly to r, $\sigma_{k+1}$ is very small (about near zero).

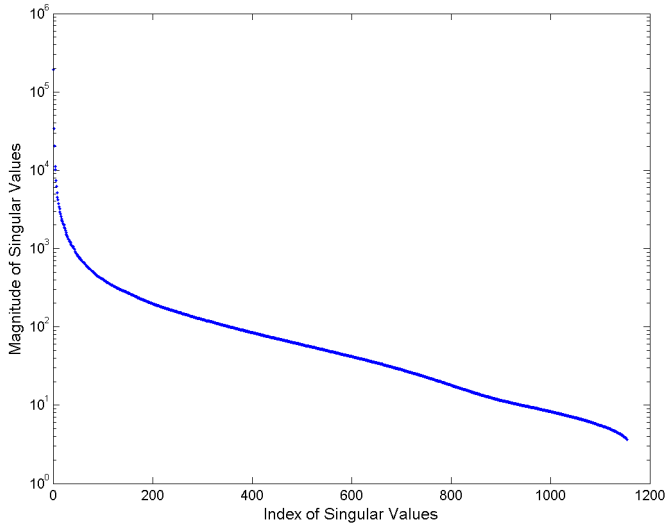
Then $\| A - \mathrm{A}_K \| \leq$(r - k)$\sigma_{k+1} \approx$ a very small number.

Thus: $A \approx A_k$ ( $k < r$).

# 4 Results and Discussion

**•Dimension reduction**

Look at the example of singular value plotting of the image (Angkor.jpg) that we use the program for singular value plotting in Appendix B below :




We see that the singular values decrease rapidly. We can further approximate a matrix by leaving off more singular terms at the end. Since the singular values are arranged in decreasing order, the last terms will have the least effect on the overall image. Doing this reduces the amount of space required to store the image on a computer.

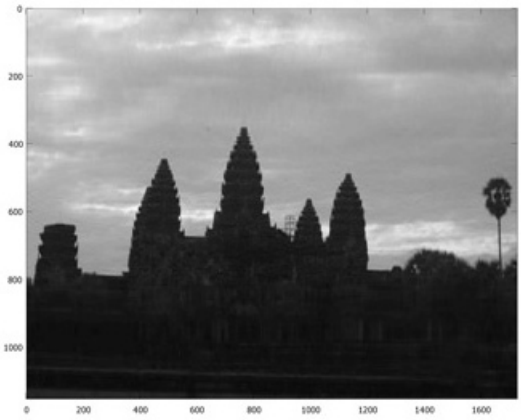
**•Image processing**

**• Method for computing the element numbers (pixels) of an image in the image compression, and computing the percentage of storage space of the image:**

A full m×n image has m×n numbers. In the image compression, we use one (1×m) singular vector u, and one (1×n) singular vector v. For a singular value, we use (m+n) numbers. Thus if we use k singular values, we store k×(m+n) numbers.
A percentage of storage space in the image compression is the numbers that we store in our compression over the total in the image = k×(m+n)/(m ×n)%.

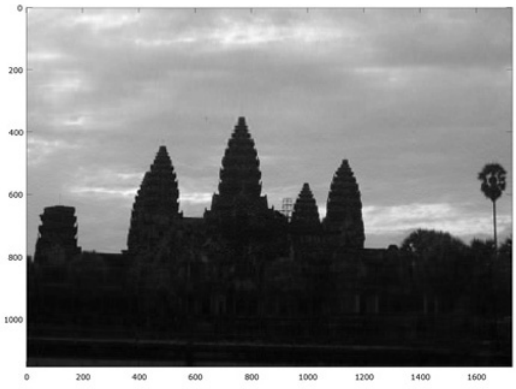
For the pictures in this presentation, we used the program for image compression in the Appendix B.
In the first example below, the original image has size 1728×1154 pixels.


( 13 terms )




( 49 terms )

The first thirteen terms actually give the shape of image but it is not a good approximation yet. This takes up about 98.12% less storage space than the original image.

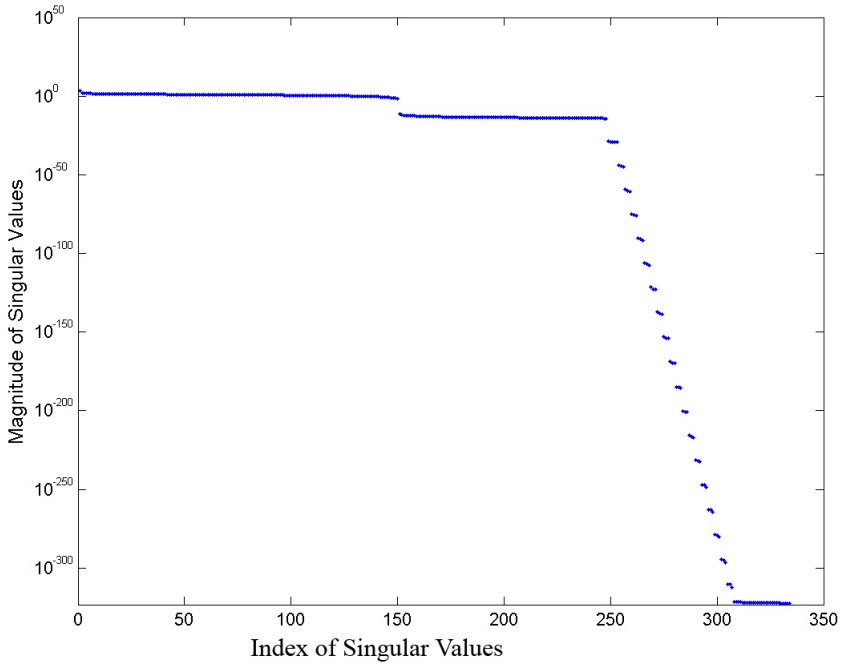



( 60 terms )

( Original image )

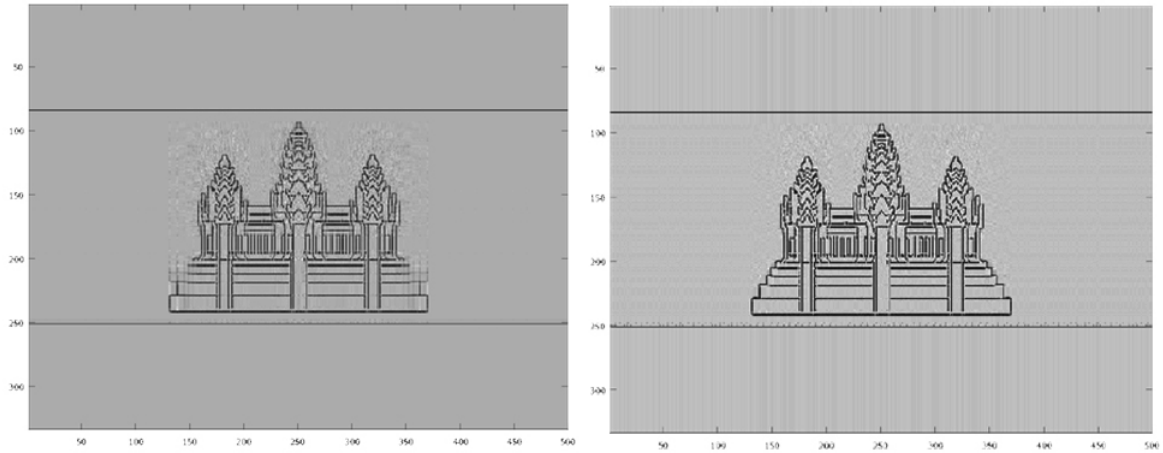
Finally, by the first sixty terms give a near perfect image, and yet reguires 91.32% less storage space than the original image. This is very much less storage space than the original image. This is very good !

**Now look at the second example of the image processing of the image named Cambodia-flag.gif(size 500×334) as the following :**
• The plot of singular values :



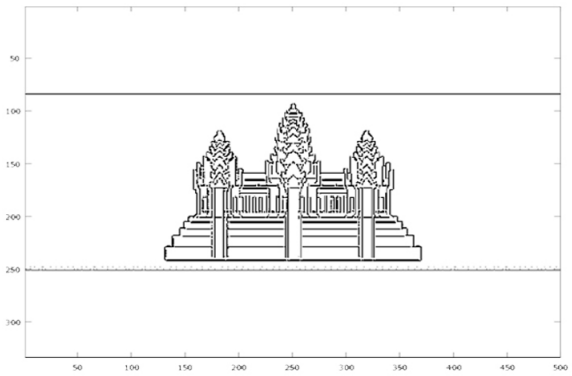

In the plot above, we observe that if we look at the image with 150 singular values, it should look very similar to the original image.

• Image processing :

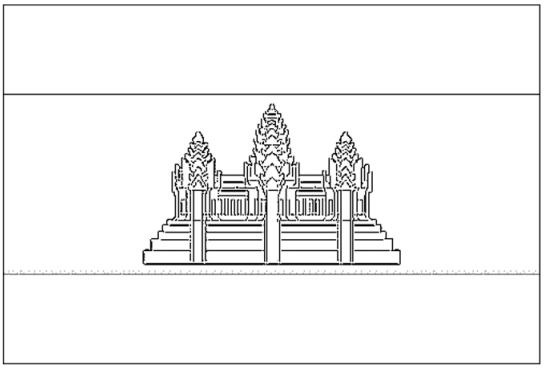


( 30 terms ) ( 60 trems )

For the first thirty terms and sixty terms, they are not good approximation but we can know what the pictures are.
For the first thirty terms, it takes 85% less storage space than the original. And for the first sixty terms, it takes 70% less storage space than the original.




( 150 terms )



( Original image )

Finally, the first 150 terms give a near perfect image, and yet reguires 25% less storage space than the original.
This is very good !

# 5 Conclusions

Using (SVD) for image compression can be a very useful tool to save storage space. We are able to get an image that is indistinguishable from the original image, but only using about 75% of the original storage space( for the second example above).
The Singular Value Decomposition is not only used for image compression, it has many other useful applications.

# 6 Future work

There are still other things relevent to the topic of the thesis left such as :

- We haven't studied other applications of the SVD. For example, using the SVD for statistical applications to find relations between data [3].
- We haven't studied the other tools that are used for image compression either. For example, the Discrete Cosine Transformation for image compression [3].

# Appendix A

## Examples of the Singular Value Decomposition (SVD)

**Example 1** *Find a singular value decomposition of* $A = \begin{bmatrix} 2 & 0 \\ 1 & 2 \\ 0 & 1 \end{bmatrix}$

*Solution:*

*First, compute:* $A^TA = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 1 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 5 & 2 \\ 2 & 2 \end{bmatrix}$

*Find the eingenvalues of* $A^TA$*:*

$|\lambda I\text{-}A^TA| = \begin{vmatrix} \lambda - 5 & -2 \\ -2 & \lambda - 5 \end{vmatrix} = (\lambda - 5)^2 \text{ - } 4 = \lambda^2 - 10\lambda + 21 = 0$

*We get* $\lambda_1 = 7$ *;* $\lambda_2 = 3$ *are the eigenvalues of* $A^TA$*.*

*Find unit eigenvectors corrresponding to* $\lambda_1 = 7$ *;* $\lambda_2 = 3$ *:*

*For* $\lambda_1 = 7$ *:* $(\lambda_1 I \text{ - } A^TA)X = 0$ *;* $\begin{pmatrix} 2 & -2 \\ -2 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = 0$

$\begin{cases} x_1 - x_2 = 0 \\ -x_1 + x_2 = 0 \end{cases}$ *or* $x_1 - x_2 = 0$ *take* $x_1 = x_2 = 1$

$x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$*. Then the unit eigenvector belongs to* $\lambda_1 = 7$ *is:*

$v_1 = \frac{x}{\|x\|} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$.

*For* $\lambda_2 = 3$*:* $(\lambda_2 I \text{ - } A^TA)Y = 0$*;* $\begin{pmatrix} -2 & -2 \\ -2 & -2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = 0$*.*

*We get* $y_1 + y_2 = 0$

*take* $y_1 = -1$ *,* $y_2 = 1$ *then* $y = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$*. The unit eigenvector belongs to* $\lambda_2 = 3$ *is:* $v_2 = \frac{y}{\|y\|} = \begin{pmatrix} \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$.

*Since* $v_1.v_2 = -\frac{1}{2} + \frac{1}{2} = 0$*, thus* $\{ v_1 , v_2 \}$ *is an orthonormal basis of* $\mathbb{R}^2$ *of unit eigenvectors of* $A^TA$*.*

$Av_1 = \begin{bmatrix} 2 & 0 \\ 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{2}{\sqrt{2}} \\ \frac{3}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$.

$$Av_2 = \begin{bmatrix} 2 & 0 \\ 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{-2}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

*We have the singular values of A:* $\sigma_1 = \sqrt{\lambda_1} = \sqrt{7}$ *and* $\sigma_2 = \sqrt{\lambda_2} = \sqrt{3}$. *For* $1 \leq i \leq r$ *where* $r = min\ \{\ 3\ ,\ 2\ \} = 2$*:*

*define* $u_i = \frac{1}{\|Av_i\|} Av_i = \frac{1}{\sigma_i} Av_i$ *(* $\|\ Av_i\ \| = \sigma_i$ *from the proof of SVD theorem ).*

*We get* $u_1 = \frac{1}{\sigma_1} Av_1 = \frac{1}{\sqrt{7}} \begin{bmatrix} \frac{2}{\sqrt{2}} \\ \frac{3}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{2}{\sqrt{14}} \\ \frac{3}{\sqrt{14}} \\ \frac{1}{\sqrt{14}} \end{bmatrix}$

$$u_2 = \frac{1}{\sigma_2} Av_2 = \frac{1}{\sqrt{3}} \begin{bmatrix} \frac{-2}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{-2}{\sqrt{6}} \\ \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{6}} \end{bmatrix}$$

*Then* $\{\ u_1\ ,\ u_2\ \}$ *is an orthonormal basis of* $\mathbb{R}^2$. *We need to extend this set to* $\{\ v_1\ ,\ v_2\ ,\ v_3\ \}$ *an orthonormal basis of* $\mathbb{R}^3$ *because U is a* $3 \times 3$ *matrix. We need one more orthonormal vector which is orthogonal to* $u_1$ *and* $u_2$.

*That is* $\begin{cases} u_1^T X = 0 \\ u_2^T X = 0 \end{cases} \begin{cases} 2x_1 + 3x_2 + x_3 = 0 \\ -2x_1 + x_2 + x_3 = 0 \end{cases}$

*We get* $x = \begin{pmatrix} 1 \\ -2 \\ 4 \end{pmatrix}$. *The unit eigenvector is*

$w = \frac{x}{\|x\|} = \begin{bmatrix} \frac{1}{\sqrt{21}} \\ \frac{-2}{\sqrt{21}} \\ \frac{4}{\sqrt{21}} \end{bmatrix}$. *By Gram-Schmidt:* $u_3 = w$ - $\frac{w.U_1}{u_1 u_1} u_1 - \frac{w.U_2}{u_2 u_2} u_2 = w$.

*Finally, let* $U = [\ u_1\ u_2\ u_3\ ] = \begin{bmatrix} \frac{2}{\sqrt{14}} & \frac{-2}{\sqrt{6}} & \frac{1}{\sqrt{21}} \\ \frac{3}{\sqrt{14}} & \frac{1}{\sqrt{6}} & \frac{-2}{\sqrt{21}} \\ \frac{1}{\sqrt{14}} & \frac{1}{\sqrt{6}} & \frac{4}{\sqrt{21}} \end{bmatrix}$

$V = [\ v_1\ v_2\ ] = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$ *and* $\Sigma = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \sqrt{7} & 0 \\ 0 & \sqrt{3} \\ 0 & 0 \end{bmatrix}$

*Then* $A = U\Sigma V^T$

$$\begin{bmatrix} 2 & 0 \\ 1 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{2}{\sqrt{14}} & \frac{-2}{\sqrt{6}} & \frac{1}{\sqrt{21}} \\ \frac{3}{\sqrt{14}} & \frac{1}{\sqrt{6}} & \frac{-2}{\sqrt{21}} \\ \frac{1}{\sqrt{14}} & \frac{1}{\sqrt{6}} & \frac{4}{\sqrt{21}} \end{bmatrix} \begin{bmatrix} \sqrt{7} & 0 \\ 0 & \sqrt{3} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}.$$

*Note: the SVD is not unique for a given matrix A. In defining* $Av_1 = \sigma u_1$ *( see in the proof of SVD theorem ), for example, replacing* $v_1$ *by -*$v_1$ *and* $u_1$ *by -*$u_1$ *does not change the equality, but changes the matrices U and V.*

*So in accordance with the nonuniqueness of the SVD, others perfectly good SVD for the matrix A are :*

$$\begin{bmatrix} 2 & 0 \\ 1 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{-2}{\sqrt{14}} & \frac{-2}{\sqrt{6}} & \frac{1}{\sqrt{21}} \\ \frac{-3}{\sqrt{14}} & \frac{1}{\sqrt{6}} & \frac{-2}{\sqrt{21}} \\ \frac{-1}{\sqrt{14}} & \frac{1}{\sqrt{6}} & \frac{4}{\sqrt{21}} \end{bmatrix} \begin{bmatrix} \sqrt{7} & 0 \\ 0 & \sqrt{3} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

$$or \begin{bmatrix} 2 & 0 \\ 1 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{-2}{\sqrt{14}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{21}} \\ \frac{-3}{\sqrt{14}} & \frac{-1}{\sqrt{6}} & \frac{-2}{\sqrt{21}} \\ \frac{-1}{\sqrt{14}} & \frac{-1}{\sqrt{6}} & \frac{4}{\sqrt{21}} \end{bmatrix} \begin{bmatrix} \sqrt{7} & 0 \\ 0 & \sqrt{3} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}.$$

*The last result of SVD for the matrix A is the same as the result of SVD computed by Octave below.*

## Computing the SVD by using Octave:

• *For computing the SVD of a matrix A in our example above , we write the command in Octave as the following:*

```
> A = [ 2 0 ; 1 2 ; 0 1 ]
> A = 2  0
      1  2
      0  1
> [ U , S , V ] = svd(A) % in order to compute the SVD of

                        %  matrix A.
```

*Then we get the answers as the following:*

```
U =
  -0.53452   0.81650   0.21822
  -0.80178  -0.40825  -0.43644
  -0.26726  -0.40825   0.87287
S =

Diagonal Matrix

   2.6458         0
        0    1.7321
        0         0

V =

  -0.70711   0.70711
  -0.70711  -0.70711
```

**Example 2** *Find the best rank-one approximation of the matrix* $A = \begin{bmatrix} 2 & 0 \\ 1 & 2 \\ 0 & 1 \end{bmatrix}$.

*Solution:*

$$\begin{bmatrix} 2 & 0 \\ 1 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{-2}{\sqrt{14}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{21}} \\ \frac{-3}{\sqrt{14}} & \frac{-1}{\sqrt{6}} & \frac{-2}{\sqrt{21}} \\ \frac{-1}{\sqrt{14}} & \frac{-1}{\sqrt{6}} & \frac{4}{\sqrt{21}} \end{bmatrix} \begin{bmatrix} \sqrt{7} & 0 \\ 0 & \sqrt{3} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{-2}{\sqrt{14}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{21}} \\ \frac{-3}{\sqrt{14}} & \frac{-1}{\sqrt{6}} & \frac{-2}{\sqrt{21}} \\ \frac{-1}{\sqrt{14}} & \frac{-1}{\sqrt{6}} & \frac{4}{\sqrt{21}} \end{bmatrix} \left( \begin{bmatrix} \sqrt{7} & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{3} \\ 0 & 0 \end{bmatrix} \right) \begin{bmatrix} \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}$$

$$= \sqrt{7} \begin{bmatrix} \frac{-2}{\sqrt{14}} \\ \frac{-3}{\sqrt{14}} \\ \frac{-1}{\sqrt{14}} \end{bmatrix} \begin{bmatrix} \frac{-1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} + \sqrt{3} \begin{bmatrix} \frac{2}{\sqrt{6}} \\ \frac{-1}{\sqrt{6}} \\ \frac{-1}{\sqrt{6}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 \\ \frac{3}{2} & \frac{3}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} + \begin{bmatrix} 1 & -1 \\ \frac{-1}{2} & \frac{1}{2} \\ \frac{-1}{2} & \frac{1}{2} \end{bmatrix}$$

*Notice how the original matrix A is seperated into a larger contribution plus a smaller contribution, because of the different sizes of the singular values. The best rank-one approximation of a matrix A is given by the first rank one matrix:*

$$\begin{bmatrix} 1 & 1 \\ \frac{3}{2} & \frac{3}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

*The second matrix provides smaller correction to matrix A. This is the main idea behind the dimension reduction and compression of the SVD.*

***For using MATLAB [4] in order to get the rank one approximation to matrix A, we do as the following steps:***

```
>> A =[2 0 ; 1 2 ; 0 1]

A =

     2     0
     1     2
     0     1

>> [u,s,v] = svd(A,0), sigma = diag(s) %to find SVD of A and singular values of A.

u =
```

```
   -0.5345    0.8165
   -0.8018   -0.4082
   -0.2673   -0.4082


s =

    2.6458         0
         0    1.7321


v =

   -0.7071    0.7071
   -0.7071   -0.7071


sigma =

    2.6458
    1.7321

>> E1 = sigma(1)*u(:,1)*v(:,1)' % to find rank one approximation to matrix A.

E1 =

    1.0000    1.0000
    1.5000    1.5000
    0.5000    0.5000
```

# Appendix B

**Matlab codes for viewing matrix as image:**

```
figure;                       % Matrices vs. images
m = rand(3,2);               % choose elements of 3x2 matrix at random
imagesc(m)                    % Plot matrix as image
axis image;                   % Show pixel coordinates as axes
print -dpng '3x2Imagetake.png' % to save the image as 3x2Imagetake.png
```

**The program used in Octave for image compression (file's name: Angkor.jpg)**

```
A = imread('Angkor.jpg'); % First of all, we command Octave to read our image.

imagesc(A); % After that, we command Octave to show our original image.

[u,s,v]=svd(A); % Then, we command Octave to compute svd of our image.

C=zeros(size(A)); % to produce a matrix C of zeros of the same size as A.

for j=1:k  % we start iterating on k=2,3,... until we get the
           % appropriate approximation of SVD of our image.

C=C+s(j,j)*u(:,j)*v(:,j).'; % This is the formula of SVD. Matrix C is
                        % written as sum of k matrices of rank one including rank one
       % matrix associated with the largest singular value and the one associated with
       %the smallest singular value. it is applied  to the image compression.

end

C=floor(C); % We command Octave to make the full matrix C.

colormap(gray) % Note that the image should be black and white because
               %  if it is a color image, it is really three matrices, one for
              % each primary color( red, blue, and green ).
               % We set the color map to be grayscale.

imagesc(C) % We command Octave to show our image compression.

print -dpng Angkor.png % to save the image that has been scaned as file's name:Angkor.png.
```

**The program used in MATLAB for plotting the singular values of image [4]( file's name: Angkor.jpg )**

```
[A,map]=imread('Angkor.jpg') % to read image into MATLAB.

colormap(gray) % the reason to put colormap(gray),
         % please read the code for image compression above.

B=im2double(A,'indexed'); % to double image A in order
                          % to get a better image.

axis image, axis off  % in order to put axis for our
                      % plotting.

[U,S,V]=svd(B,0); % to compute SVD of image B.

m=diag(S); % at here, m is the singula values of B =elements
           % in the diagonal of matrix S.

semilogy(m,'.') % this is the code to plot the sigular
```

```
                % values vs their indices.
xlabel('Index of Singular Values','FontSize',12)      %to put label for x-axis.
ylabel('Magnitude of Singular Values','FontSize',12)  % to put label for y-axis.
print -dpng 'singularplotangkor.png' % to save the plot.
```

Note: For using the programs in the appendix B, the image should be black and white (2D image). If it is the color image (RGB image), we can not use those codes above since in RGB images there exist three indexed images. In order to use those codes, we have to convert the color image (RGB image) into black and white image or convert the color image to be monochrome.


## Acknowledgements

I would like to express my deep thanks and gratitude to :

Dr. Angel R. Pineda for his guidance and comments.

And Emily Bice for her monitoring the progress of thesis.


## References


[1] Linear Algebra And Its Applications, 3rd ed, *David C Lay*, Addison-Wesley Publishing Company, 2002.

[2] Linear Algebra And Its Applications, 3rd ed, *Gilbert Strang*, Thomson Learning, Inc, 1988.

[3] Numerical Analysis *Timothy Sauer*, George Mason University, Pearson Education, Inc, 2006.

[4] MATLAB An Introduction With Applications, 2nd ed, *Amos Gilat*, John Wiley & Sons, Inc, 2005.