The Singular Value Decomposition and It's Applications in Image Processing

Christopher Jason Ogden, Tammie Huff
Math 45 — College of the Redwoods

December 18, 1997

Abstract

We use singular value decomposition to approximate large, unmanageable matrices into smaller invertible square matrices. Topics include the mathematics behind the singular value decomposition, the use of `Matlab` with the singular value decomposition, and the application of the singular value decomposition to image compression without significant data loss.

1 Introduction

The singular value decomposition (SVD), one of the most useful tools of linear algebra, is a factorization and approximation technique which effectively reduces any matrix into a smaller invertible and square matrix. The SVD provides where other linear approximation techniques fail. With almost every tool and technique discussed in linear algebra, there is a delimiter, *"...provided that the matrix is both invertible and square."* The singular value decomposition not only approximates this special case scenario, but it will also work it's magic on every other possible scenario. The SVD works wonderfully with both under- and over- determined matrices.

2 Mathematics Behind the SVD

Theorem 1 *There exists matrices U, D, V such that the matrix A, an $m \times n$ matrix with rank r, can be factored into the form $A = UDV^T$; where D is a "diagonal" $m \times n$ matrix with real entries*

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0,\ r = \min(m, n)$$

and U &V are orthogonal matrices such that U is an $m \times m$ matrix and V is an $n \times n$ matrix. The diagonal entries of D, $\{\sigma_1, \sigma_2, \cdots \sigma_r\}$ are called the singular values of the matrix A, the columns of the matrix U are called the left singular vectors of the matrix A, and the column space of the matrix V (or the row space *of V^T) are called the right singular vectors of the matrix A. Furthermore, the matrix A can be condensed as $A_r = U_r \Sigma V_r$, where Σ is an $r \times r$ diagonal matrix with entries $\{\sigma_1,\sigma_2,...,\sigma_r\}$. Further still, the matrix A can be approximated as $A_i = U_i \Sigma_i {V_i}^T$, where i corresponds to the first i rows and i columns of each individual matrix A, U, Σ, V.*

Proof. Given that A is an $m \times n$ matrix in the form

$$A = \begin{bmatrix} \vec{a_1} & \vec{a_2} & \cdots & \vec{a_i} & \cdots & \vec{a_n} \end{bmatrix}$$

such that

$$\vec{a_i} = \begin{bmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{im} \end{bmatrix}$$

Therefore

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1i} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2i} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mi} & \cdots & a_{mn} \end{bmatrix}$$

2.1 Linear Transformations

The process of multiplying any typical vector $\vec{x}$ by a matrix A is known as preforming a linear transformation of $\vec{x}$. If we use the notation $T(\vec{x}) = A\vec{x}$, where $T(\vec{x})$ maps any vector $\vec{x}$ onto the vector $A\vec{x}$ and is denoted by $\vec{x} \to A\vec{x}$, all that is required to maximize the transformation $T(\vec{x})$, is to maximize the quantity $A\vec{x}$.

Exercise 2 *If λ is an eigenvalue of the matrix A, and $\vec{x}$ is the corresponding unit eigenvector, find the unit eigenvector at which $||A\vec{x}||$ is maximized.*

Solution Since $||A\vec{x}||^2$ is maximized for the same $\vec{x}$ as is $||A\vec{x}||$, and since $||A\vec{x}||^2$ is easier to examine, this substitution in the problem will be made.

$$\begin{aligned} ||A\vec{x}||^2 &= (A\vec{x})^T(A\vec{x}) \\ &= \vec{x}^T A^T A\vec{x} \\ &= \vec{x}^T (A^T A)\vec{x} \qquad (1) \end{aligned}$$

$(A^T A)$ is a symmetric matrix as is shown.

$$(A^T A)^T = A^T A^{TT} = A^T A$$

Therefore the problem has become to maximize the quadratic form $\vec{x}^T S\vec{x}$, where S is an invertible square matrix and $\vec{x}$ is as before. It is known that the unit eigenvector $\vec{v_1}$ corresponding to the strictly dominant eigenvalue σ_1 of the matrix S maximizes this equation. This means that $||A\vec{x}||$ is maximized for $\vec{x} = \vec{v_1}$.

A relationship between σ and λ can be obtained by taking this step just slightly farther. We notice that

$$A\vec{x} = \lambda\vec{x}$$

$$(A^T A)\vec{v} = \sigma\vec{v}$$

Therefore

$$\begin{aligned} ||A\vec{x}|| &= ||\lambda\vec{x}|| \\ &= |\lambda|\ ||\vec{x}|| \\ &= |\lambda| \end{aligned}$$

And substituting back into equation (1),

$$\begin{aligned} |\lambda|^2 &= \vec{x}^T(A^T A)\vec{x} \\ \vec{x}|\lambda|^2 &= \vec{x}\vec{x}^T(A^T A)\vec{x} \\ |\lambda|^2\vec{x} &= (A^T A)\vec{x} \end{aligned}$$

but $\vec{x} = \vec{v}$, so

$$\begin{aligned} |\lambda|^2\vec{v} &= \sigma\vec{v} \\ |\lambda|^2\vec{v}\vec{v}^T &= \sigma\vec{v}\vec{v}^T \\ |\lambda|^2 &= \sigma \end{aligned}$$

2.2 Invertible Square Matrix

Let $S = (A^T A)$ such that S is an $n \times n$ invertible square matrix in the form

$$S = \begin{bmatrix} \vec{s_1} & \vec{s_2} & \cdots & \vec{s_i} & \cdots & \vec{s_n} \end{bmatrix}$$

where

$$\vec{s_i} = \begin{bmatrix} s_{i1} \\ s_{i2} \\ \vdots \\ s_{in} \end{bmatrix}$$

Therefore S is an $n \times n$ matrix in the form

$$S = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \vdots & \vdots & & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nn} \end{bmatrix}$$

We have just shown that $||A\vec{x}||$ is maximized for $\vec{x} = \vec{v_1}$ where $\vec{v_1}$ is the eigenvector of S corresponding to the strictly dominant eigenvalue σ_1. Since $\{\vec{v_1}, \vec{v_2}, ..., \vec{v_n}\}$ is an orthonormal set of the unit eigenvectors of S, then the matrix $V = [\vec{v_1}, \vec{v_2}, ..., \vec{v_n}]$ is an $n \times n$ orthogonal matrix comprised of the unit eigenvectors of S in order of descending dominance. In addition, the eigenvalues $\{\sigma_1, \sigma_2, ..., \sigma_n\}$ of S are put into the 'diagonal' entries of the padded matrix D.

It is at this point that we must remember the goal which we are trying to achieve. This is, that given any $m \times n$ matrix A, there exists a factorization $A = UDV^T$. We know the format of D, V, and A, so a matrix U can be obtained in terms of the remaining three matrices in the very least.

$$\begin{aligned} A &= UDV^T \\ U &= AVD^{-1} \end{aligned} \tag{2}$$

We can consider the matrix D as an $m \times n$ matrix in the form

$$D = \begin{bmatrix} \vec{d_1} & \vec{d_2} & \cdots & \vec{d_i} & \cdots & \vec{d_n} \end{bmatrix}$$

where the vector $\vec{d_i}$ is such that the only non-zero entry of the vector is the i^{th} term, which is the inverse of the i^{th} singular value of the matrix A. Now the product VD^{-1} can be expressed as

$$\begin{aligned} VD^{-1} &= \begin{bmatrix} \vec{v_1} & \vec{v_2} & \cdots & \vec{v_n} \end{bmatrix} \begin{bmatrix} \vec{d_1}^{-1} \\ \vec{d_2}^{-1} \\ \vdots \\ \vec{d_n}^{-1} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{\sigma_1}\vec{v_1} & \frac{1}{\sigma_2}\vec{v_2} & \cdots & \frac{1}{\sigma_n}\vec{v_n} \end{bmatrix} \end{aligned}$$

Therefore VD^{-1} is an $n \times n$. The product of the left multiplication of A, an $m \times n$ matrix, to VD^{-1}, an $n \times m$ matrix, yields that U is an $m \times m$ matrix in the form

$$U = \left[\frac{1}{\sigma_1}A\vec{v_1}, \frac{1}{\sigma_2}A\vec{v_2}, ..., \frac{1}{\sigma_n}A\vec{v_n}\right]$$

2.3 Consolidating a Matrix Using SVD

Definition 3 *The rank of a matrix is the number of linearly independent vectors in the column space of the matrix.*

Exercise 4 *Letting A, U, D, V be as before, show that if the rank of the matrix is less than the smallest dimension of the matrix, a smaller matrix A_r can be computed.*

Solution If we let $r = rank(A)$ and define Σ to be an $r \times r$ diagonal matrix where the main diagonal of the matrix are the first r eigenvalues of the matrix S in order of descending dominance, then Σ would be in the form

$$\Sigma = \begin{bmatrix} \sigma_1 & & & 0 \\ & \sigma_2 & & \\ & & \ddots & \\ 0 & & & \sigma_r \end{bmatrix}$$

and the values $\{\sigma_1, \sigma_2, ..., \sigma_r\}$ would be called the singular values of the matrix A. If we define the matrix D to be an $m \times n$ matrix with containing Σ and augmented with zeros so as to give it the same dimensions as the matrix A, D would be in the form

$$D = \underbrace{\begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix}}_{(m-r) \text{ rows of zeros}} \qquad (n-r) \text{ columns of zeros}$$

If we let $V_r = [\vec{v_1}, \vec{v_2}, ..., \vec{v_r}]$ and let the remaining vectors $\left\{\vec{v_{r+1}}, \vec{v_{r+2}}, ..., \vec{v_n}\right\}$ be the basis for the matrix V_{n-r}, then V can be written as

$$V = [V_r, V_{n-r}]$$

A similar argument can be made for U.

$$U = [U_r, U_{m-r}]$$

Therefore, it can be shown that the matrix A can be written as a smaller $r \times r$ matrix A_r.

$$\begin{aligned} A &= UDV^T \\ A &= [U_r, U_{m-r}] \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_r{}^T \\ V_{n-r}{}^T \end{bmatrix} \\ A &= U_r \Sigma V_r{}^T = A_r \end{aligned}$$

2.4 Approximating a Matrix Using SVD

It can be shown from the previous example that the matrix A which equals the matrix A_r can be expressed as a linear combination of the singular values of the matrix A and the corresponding vectors $\vec{u_i}$ and $\vec{v_i}^T$

$$\begin{aligned} A &= UDV^T \\ A &= \vec{u_1}\vec{d_1}\vec{v_1}^T + \vec{u_2}\vec{d_2}\vec{v_2}^T + \cdots + \vec{u_i}\vec{d_i}\vec{v_i}^T + \cdots + \vec{u_n}\vec{d_n}\vec{v_n}^T \\ A &= \sigma_1\vec{u_1}\vec{v_1} + \sigma_2\vec{u_2}\vec{v_2} + \cdots + \sigma_i\vec{u_i}\vec{v_i} + \cdots + \vec{u_n}\vec{d_n}\vec{v_n}^T \end{aligned}$$

We know that the terms $\left\{\sigma_1\vec{u_1}\vec{v_1}, \sigma_2\vec{u_2}\vec{v_2}, ..., \sigma_n\vec{u_n}\vec{v_n}\right\}$ are in order of dominance from greatest to least. Therefore, an approximation of the matrix A can be achieved by reducing the number of iterations involved in the linear combination.

$$A_i = \sigma_1\vec{u_1}\vec{v_1} + \sigma_2\vec{u_2}\vec{v_2} + \cdots + \sigma_i\vec{u_i}\vec{v_i}$$

3 `MATLAB` and the SVD

It's quite obvious that the mathematics behind the singular value decomposition would become extraordinarily involved rather quickly. For this reason, once the mathematics have been understood, it is a good idea to use a mathematics software. `Matlab` is one example that works quite nicely. To find out more about these commands and others while working in `Matlab` use the `help` command. For example, if the command is `linspace(0,5)`, type `help linspace` to find out more about the `linspace` command.

Exercise 5 *Create a random* 8×10 *matrix* A *with a rank of* 6 *and integer values ranging from* -64 *to* 64*. Using* ***`Matlab`****'s* ***`svd`*** *command, find the matrices* U*,* D*, & V corresponding to* A*.*

To create a matrix of random integers, the easiest way is to use the `randint` command (you must have ATLAST for this to work). The command with these parameters reads:

```
>> A=randint(8,10,64,6)
```

Then, all that is required is the ability to type the next line into the computer:

```
>> [U,D,V]=svd(A)
```

If you wish, it is possible to confirm the rank of the matrix A by typing the command:

```
>> rank(A)
```

Or by typing

```
>> diag(D)
```

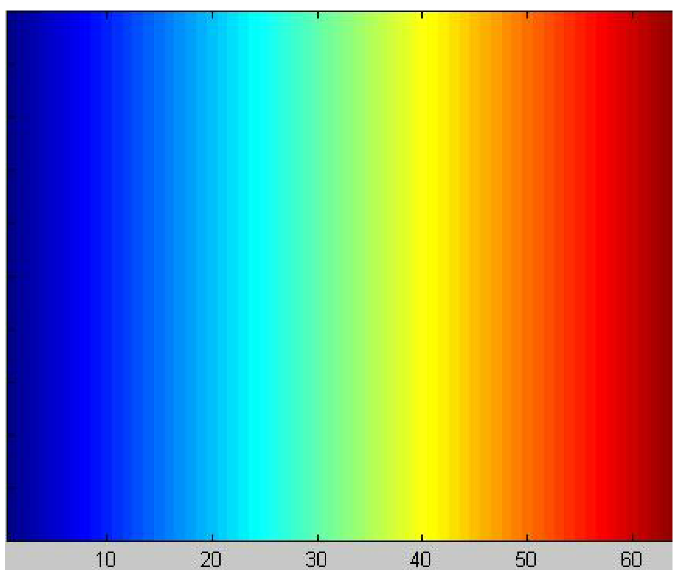
and seeing that two of the singular values of A are zero, or so nearly zero that they are negligible. And, that's it. The matrix A was inputted, and the matrices U, D, & V were computed.

4 Image Processing and the SVD

In order to understand the following section, a brief discussion about how `Matlab` constructs images from matrices is necessary. Basically, each entry in the matrix corresponds to a small square of the image. The numerical value of the entry corresponds to a color in `Matlab`. The color spectrum can be seen in `Matlab` by typing the following commands, and the pictured screen will appear (1).

```
>> C=1:64;
```

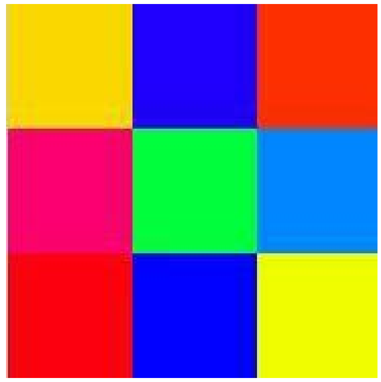
>> image(C)



With this in mind, entering a 3×3 matrix of random integers should give a picture of nine square blocks comprising one large block. Furthermore, the color of each individual block will correspond to the color pictured at that numerical value in figure (1).

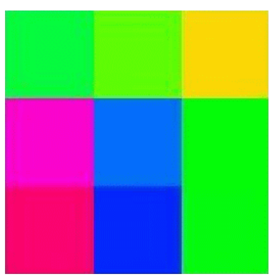
```
>> S=randint(3,3)
```

>> image(S)

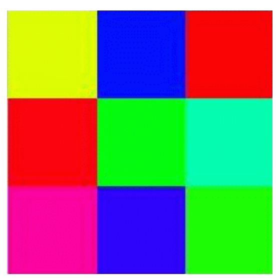


It has been shown that any matrix S can be approximated using the a lesser number of iterations when calculating the linear combinations defining S. This can be shown using the `Matlab` command (available through ATLAST) `svdimage`. `svdimage` is an interactive program which shows the original image produced by S and the image produced by the approximated A with the shown number of iterations. The following demonstrates (figures will vary due to the use of random integers).

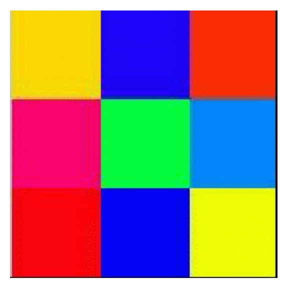
```
>> [U,D,V]=svd(S)
>> svdimage(S,U,D,V)
```



One Iteration



Two Iterations



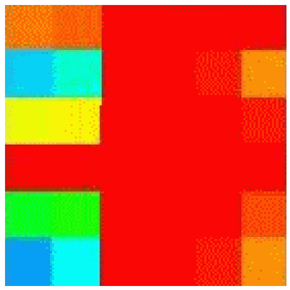
Three Iterations

The original image was not obtained until the third iteration, but this can be explained by typing

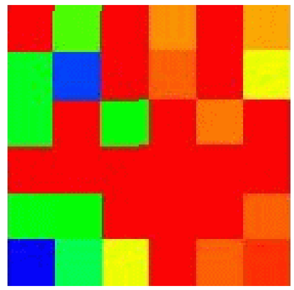
```
>> rank(A)
```

Of course, realize that in the scope of a more detailed image, say 512×512 the discoloring of a pixel here or there might not be of concern. Use `Matlab` to construct a 6×6 matrix of rank 4. This time allow color values from between -64 and 64. The original image should be achieved by the fourth approximation. Meaning that the 4×4 matrix S_r of only sixteen entries gives the same image as the 6×6 matrix of 36 entries. A reduction of more than 50% in size.

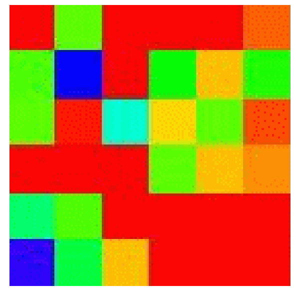
```
>> S=randint(6,6,64,4)
>> [U,D,V]=svd(S)
>> svdimage(S,U,D,V)
```



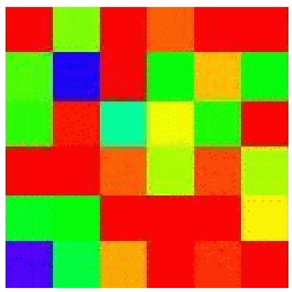
One Iteration



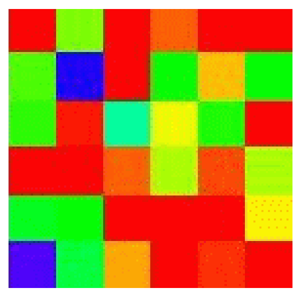
Two Iterations



Three Iterations



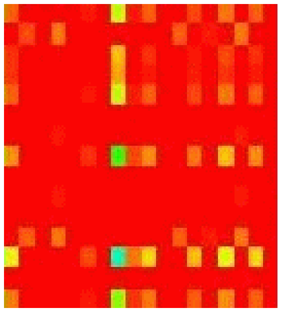
Four Iterations



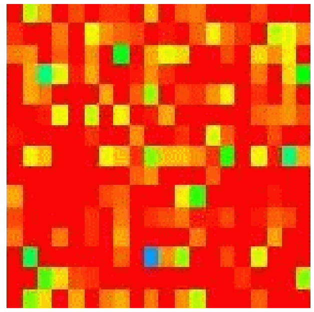
Original Image

Up until now, the matrix S has been an invertible square matrix. It is possible to show that the matrix A, an $m \times n$ matrix, can be approximated using the same techniques. Using `Matlab`, construct A to be a 15×20 matrix of random integers ranging from -64 to 64, with a rank of 12. An exact representation of the original image should be obtained by the twelfth iteration. Bear in mind however, that depending on the desired quality of the picture, and the number of pixels used to construct the image, that it is possible for an approximation of only ten iterations to be 'good enough'.

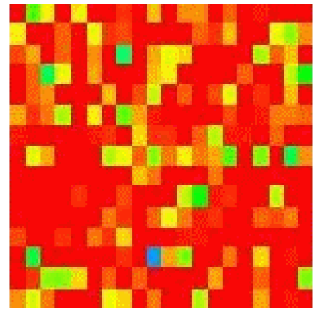
```
>> A=randint(15,20,64,12)
>> [U,D,V]=svd(A)
>> svdimage(A,U,D,V)
```



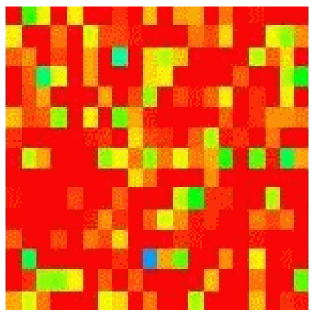
One Iteration



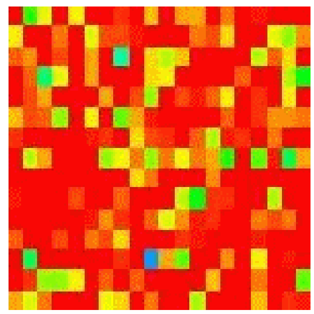
Five Iterations



Ten Iterations



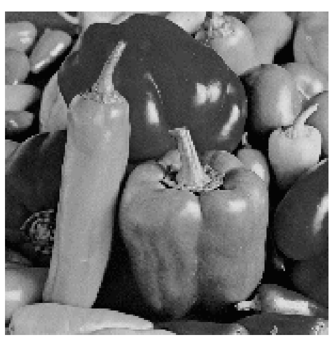
Twelve Iterations



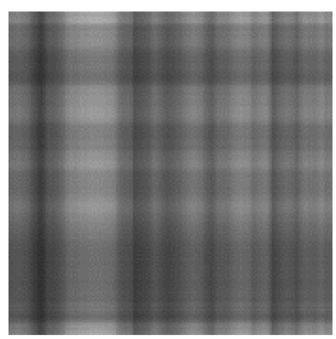
Original Image

Again, the approximation of A_{12} having only 144 entries is an exact duplicate of the original matrix A containing 300 entries, again a reduction of more than half. Notice also that our 'good enough' matrix A_{10} is a matrix of 100 entries, one third the size of the original.

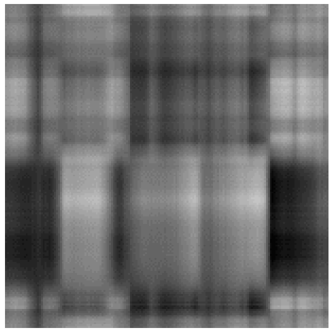
The following image is a 512×512 matrix.



Original Image

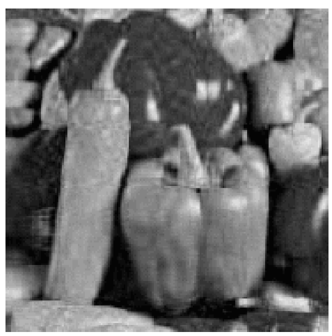


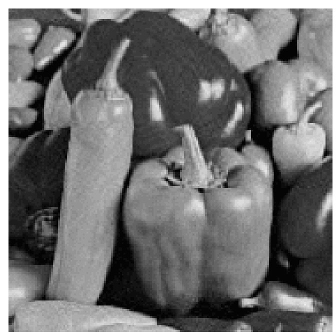
One Iteration



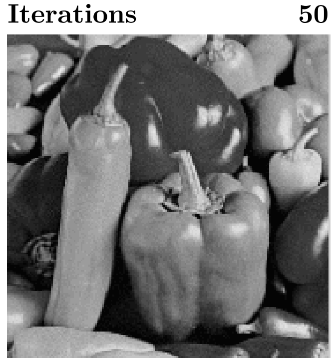
Two Iterations **Ten Iterations**

As you can see, after only 10 iterations you can already tell what the image is.





25 Iterations **50 Iterations**



75 Iterations

By 25 iterations the picture is clearly evident. By 75 iterations we have essentially the original image. A 75x75 matrix, with 5625 entries is significantly reduced compared to a 512x512 matrix with 262,144 entries.

5 Conclusion

Singular value decomposition has shown to be useful in linear algebra. When applied to image processing, a matrix can be compressed to a significantly smaller sized matrix, and portray almost an identical image. This saves a lot of room!

6 Bibliography

Lay, David C. **Linear Algebra and Its Applications**. Addison Wesley Longman, Inc., 1997

Leon, Steve. **Atlast:Computer Exercises for Linear Algebra**. Prentice-Hall, Inc., 1996,

Mulcahy, Colm & Rossi, John. "Atlast:A Fresh Approach to Singular Value Decomposition"