7-16-2007

SVD and PCA in Image Processing

Wasuta -. Renkjumnong

SINGULAR VALUE DECOMPOSITION AND PRINCIPAL COMPONENT ANALYSIS IN IMAGE PROCESSING

by

WASUTA RENKJUMNONG

Under the Direction of Marina Arav

ABSTRACT

The Singular Value Decomposition is one of the most useful matrix factorizations in applied linear algebra, the Principal Component Analysis has been called one of the most valuable results of applied linear algebra. How and why principal component analysis is intimately related to the technique of singular value decomposition is shown. Their properties and applications are described. Assumptions behind this techniques as well as possible extensions to overcome these limitations are considered. This understanding leads to the real world applications, in particular, image processing of neurons. Noise reduction, and edge detection of neuron images are investigated.

Keywords: Principal component analysis, Singular value decomposition, Image Processing, Noise reduction, Edge detection, Image compression, Eigenvalues, Eigenvectors, Covariance matrix, Diagonalization, Column space, Orthonormal basis, Orthogonal vectors, Span

SINGULAR VALUE DECOMPOSITION AND PRINCIPAL COMPONENT ANALYSIS IN IMAGE PROCESSING

by

WASUTA RENKJUMNONG

A Thesis Presented in Partial Fulfillment of the Requirements for the Degree of

Master of Science

in College of Arts and Sciences

Georgia State University

2007

SINGULAR VALUE DECOMPOSITION AND PRINCIPAL COMPONENT ANALYSIS IN IMAGE PROCESSING

by

WASUTA RENKJUMNONG

Major Professor:	Marina Arav
Committee:	Frank Hall
	Zhongshan Li
	Saeid Belkasim
	Michael Stewart

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

August 2007

ACKNOWLEDGEMENTS

The author wishes to gratefully acknowledge the assistance of Dr. Marina Arav, Dr. Frank J. Hall, Saeid Belkasim, Zhongshan Li, and Michael Stewart without whose guidance this thesis would not have been possible. He would also like to thank Drs. Mihaly Bakonyi, George Davis, Lifeng Ding, Yu-Sheng Hsu, Gengsheng Qin, Nikitta Patterson, Michael Stewart, Joseph Walker, Dr. Yichuan Zhao ,and Mr. Tsegaselassie Workalemahu for support and encouragement in his course work and in his research towards this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

1. Introduction

Singular values and the singular value decomposition play an important role in high-quality statistical computations and in schemes for data compression based on approximating a given matrix with one of lower rank. They also play a central role in the theory of theory of unitarily invariant norms. Many modern computational algorithms are based on singular value computations because the problem of computing the eigenvalues of a general matrix (like the problem of computing the eigenvalues of a Hermitian matrix) is well-conditioned. The original motivation for the study of singular values is the following. Nineteenth-century differential geometers and algebraists wanted to know how to determine whether two real bilinear forms

$$\varphi_A(x,y)=\sum_{i,j=1}^{n} a_{ij}x_iy_j \quad \text{and} \quad \varphi_B(x,y)=\sum_{i,j=1}^{n} b_{ij}x_iy_j, \tag{1}$$
$$A=[a_{ij}], \quad B=[b_{ij}]\in M_n(\mathbb{R}), x=[x_i], y=[y_i]\in\mathbb{R}^n,$$

were equivalent under independent real orthogonal substitutions, that is, whether there are real orthogonal $Q_1, Q_2 \in M_n(\mathbb{R})$ such that $\varphi_A(x,y)=\varphi_B(Q_1x, Q_2y)$ for all $x, y \in \mathbb{R}^n$. One could approach this problem by finding a canonical form to which any such bilinear form can be reduced by orthogonal substitutions, or by finding a complete set of invariants for a bilinear form under orthogonal substitutions. In 1873, the Italian differential geometer E. Beltrami did both, followed, independently, by the French algebraist C. Jordan in 1874.

Beltrami discovered that for each real $A \in M_n(\mathbb{R})$ there are always real orthogonal $Q_1, Q_2 \in M_n(\mathbb{R})$ such that

$$Q_1^TAQ_2=\Sigma=\operatorname{diag}(\sigma_1(A),\dots,\sigma_n(A)), \tag{2}$$

is a nonnegative diagonal matrix, where $\sigma_1(A)^2 \geq \cdots \geq \sigma_n(A)^2$ are the eigenvalues of AA^T (and also of A^TA). Moreover, he found that the (orthonormal) columns

of Q_1 and Q_2 are eigenvectors of AA^T and A^TA, respectively. Although Beltrami proposed no terminology for the elements of his canonical form, this is what we now call the *singular value decomposition* for a real square matrix; the *singular values* of A are the numbers $\sigma_1(A) \geq \cdots \geq \sigma_n(A) \geq 0$. The diagonal bilinear form

$$\varphi_{Q_1^T A Q_2}(\xi, \eta) = \sigma_1(A)\xi_1\eta_1 + \cdots + \sigma_n(A)\xi_n\eta_n, \tag{3}$$

gives a convenient canonical form to which any real bilinear form $\varphi_A(x, y)$ can be reduced by independent orthogonal substitutions, and the eigenvalues of AA^T are a complete set of invariants for this reduction.

Jordan came to the same canonical form as Beltrami, but from a very different point of view. He found that the (necessarily real) eigenvalues of the $2n$-by-$2n$ real symmetric matrix

$$\begin{bmatrix} O & A \\ A^T & O \end{bmatrix}, \tag{4}$$

are paired by sign, and that its n largest eigenvalues are the desired coefficients $\sigma_1(A), \ldots, \sigma_n(A)$ of the canonical form (3). Jordan's proof starts by observing that the largest singular value of A is the maximum of the bilinear form x^TAy subject to the constraints $x^Tx = y^Ty = 1$. The block matrix (4) has been rediscovered repeatedly by later researches, and plays a key role in relating eigenvalue results for Hermitian matrices to singular value results for general matrices.

Apparently unaware of the work of Beltrami and Jordan, J. J. Sylvester (1889/90) discovered, and gave yet a third proof for, the factorization (2) for real square matrices. He termed the coefficients $\sigma_i(A)$ in (3) the *canonical multipliers* of the bilinear form $\varphi_A(x, y)$. The leading idea for Sylvester's proof was "to regard a finite orthogonal substitution as the product of an infinite number of infinitesimal ones."

In 1902, L. Autonne showed that every nonsingular complex $A \in M_n$ can be written as $A = UP$, where $U \in M_n$ is unitary and $P \in M_n$ is positive definite.

Autonne returned to these ideas in 1913/15 and used the fact that A^*A and AA^* are similar to show that any square complex matrix $A \in M_n$ (singular or nonsingular) can be written as $A = V\Sigma W^*$, where $V, W \in M_n$ are unitary and Σ is a nonnegative diagonal matrix; he gave no name to the diagonal entries of Σ. Autonne recognized that the positive semidefinite factor Σ is determined essentially uniquely by A, but that V and W are not, and he determined the set of all possible unitary factors V and W associated with A. He also recognized that the unitary factors V and W could be chosen to be real orthogonal if A is real, thus obtaining the theorem of Beltrami, Jordan, and Sylvester as a special case; however, Autonne was apparently unaware of their priority for this result. He realized that by writing $A = V\Sigma W^* = (VW^*)(W\Sigma W^*) = (V\Sigma V^*)(VW^*)$ he could generalize his 1902 polar decomposition to the square singular case. Although Antonne did not consider the singular value decomposition of a nonsquare A in his 1915 paper, the general case follows easily from the square case.

In his 1915 paper, Autonne also considered special forms that can be achieved for the singular value decomposition of A under various assumptions on A, for example, unitary, normal, real, coninvolutory ($\bar{A} = A^{-1}$), and symmetric. In the latter case, Autonne's discovery that a complex square symmetric A can always be written as $A = U\Sigma U^T$ for some unitary U and nonnegative diagonal Σ gives him priority for a useful (and repeatedly rediscovered) result often attributed in the literature to Schur (1945) or Takagi (1925), see [6]. To be precise, one must note that Autonne actually presented a proof for the factorization $A = U\Sigma U^T$ only for nonsingular A, but his comments suggest that he knew that the assumption of nonsingularity was inessential. In any event, the general case follows easily from the nonsingular case.

The pioneering work on the singular value decomposition in Autonne's 77-page 1915 paper seems to have been completely overlooked by later researchers. In the

classic 1993 survey [11], Antonne's 1915 paper is referenced, but the singular value decomposition for square complex matrices is stated so as to suggest (incorrectly) to the reader that Autonne had proved it only for nonsingular matrices.

In 1930, Wintner and Murnaghan rediscovered the polar decomposition of a nonsingular square complex matrix as well as a special case (nonsingular A) of Autonne's observation that one may always write $A = PU = UQ$ (the same unitary U) for possibly different positive definite P and Q; they also noted that one may choose $P = Q$ if and only if A is normal. Wintner and Murnaghan seemed unaware of any prior work on the polar decomposition.

In another 1930 paper, Browne cited Autonne's 1913 announcement and used his factorization $A = V\Sigma W^*$ (perhaps for the first time) to prove inequalities for the spectral radius of Hermitian and general square matrices. Browne attached no name to the diagonal entries of Σ, an referred to them merely as "the square roots... of the characteristic roots of AA^*."

A complete version of the polar decomposition for a rectangular complex matrix was published in 1935 by Williamson, who credited both Autonne (1902) and Wintner-Murnaghan (1930) for prior solution of the square nonsingular case; Autonne's 1915 solution of the general square case is not cited. Williamson's proof, like Autonne's, starts with the spectral decompositions of the Hermitian matrices AA^* and A^*A. Williamson did not mention the singular value decomposition, so there is no recognition that the general singular value decomposition for a rectangular complex matrix follows immediately from his result.

Finally, in 1939, Eckart and Young gave a clear and complete statement of the singular value decomposition for a rectangular complex matrix, crediting Autonne (1913) and Sylvester (1890) for their prior solutions of the square complex and real cases. The Eckart-Young proof is self-contained and is essentially the same as Williamson's (1930); they do not seem to recognize that the rectangular case follows

easily from the square case. Eckart and Young give no indication of being aware of Williamson's result and do not mention the polar decomposition, so there is no recognition that their theorem implies a general polar decomposition for rectangular complex matrices. They give no special name to the nonnegative square roots of the "the characteristic values of AA^*," and they view the factorization $A = V\Sigma W^*$ as a generalization of the "principle axis transformation" for Hermitian matrices.

While algebraists were developing the singular value and polar decompositions for finite matrices, there was a parallel and apparently quite independent development of related ideas by researchers in the theory of integral equations. In 1907, E. Schmidt published a general theory of real integral equations, in which he considered both symmetric and nonsymmetric kernels. In his study of the nonsymmetric case, Schmidt introduced a pair of integral equations of the form

$$\varphi(s) = \lambda \int_a^b K(s,t)\psi(t)dt \quad \text{and} \quad \psi(s) = \lambda \int_a^b K(t,s)\varphi(t)dt, \tag{5}$$

where the functions $\varphi(s)$ and $\psi(s)$ are not identically zero. Schmidt showed that the scalar λ must be real since λ^2 is an eigenvalue of the symmetric (and positive semidefinite) kernel

$$H(s,t) = \int_a^b K(s,\tau)K(t,\tau)d\tau.$$

If one thinks of $K(s,t)$ as an analog of a matrix A, then $H(s,t)$ is an analog of AA^T. Traditionally, the "eigenvalue" parameter λ in the integral equation literature is the reciprocal of what matrix theorists call an eigenvalue. Recognizing that such scalars λ together with their associated pairs of functions $\varphi(s)$ and $\psi(s)$ are, for many purposes, the natural generalization to the nonsymmetric case of the eigenvalues and eigenfunctions that play a key role in the theory of integral equations with symmetric kernels, Schmidt called λ an "eigenvalue" and the associated pair of functions $\varphi(s)$ and $\psi(s)$ "adjoint eigenfunctions" associated with λ.

Picard (1910) further developed Schmidt's theory of nonsymmetric kernel but,

at least in the symmetric case, refers to Schmidt's "eigenvalues" as singular values (valeurs singulières). Perhaps in an effort to avoid confusion between Schmidt's two different uses of the term "eigenvalue," later researchers in integral equations seem to have adopted the term "singular value" to refer to the parameter λ in (5). In a 1937 survey, for example, Smithies refers to "singular values, i.e., E. Schmidt's eigen-values of the nonsymmetric kernel, and not the eigen-values in the ordinary sense." Smithies also notes that he had been "unable to establish any direct connection between the orders of magnitude of the eigen-values and the singular values when the kernel in not symmetric." Establishing such a connection seems to have remained a theme of Smithies' research for many years, and Smithies' student Chang (1949) succeeded in establishing an indirect connection : Convergence of an infinite series of given powers of the singular values of an integral kernel implies convergence of the infinite series of the same powers of the absolute values of the eigenvalues. Weyl (1949) then showed that there was actually a direct inequality between partial sums of Chang's two series, and the modern theory of singular value inequalities was born.

Although it is not at all concerned the singular value decomposition or polar decomposition, a seminal 1939 paper of Von Neumann made vital use of facts about singular values in showing that a norm $||A||$ on $M_{m,n}$ is unitarily invariant if and only if it is a symmetric gauge function of the square roots of "the proper values of AA^*". Despite his familiarity with the integral equation and operator theory literature, Von Neumann never uses the term "singular value," and hence his pages are speckled with square roots applied to the eigenvalues of AA^*. His primary tools are duality and convexity, and since the basic triangle inequality for singular value sums was not discovered until 1951, Von Neumann's proof is both long and ingenious.

During 1949-50, a remarkable series of papers in the *Proceeding of the National*

Academy of Science (U.S.) established all of the basic inequalities involving singular values and eigenvalues. Let $\lambda_1, \ldots, \lambda_n$ and $\sigma_1, \ldots, \sigma_n$ denote the eigenvalues and singular values of a given square matrix, ordered so that $|\lambda_1| \geq \cdots \geq |\lambda_n|$ and $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$. Apparently motivated by Chang's (1949) analytic results in the theory of integral equations, Weyl (1949) showed that $|\lambda_1 \cdots \lambda_k| \leq \sigma_1 \cdots \sigma_k$ for $k = 1, \ldots, n$ and deduced that $\varphi(|\lambda_1|) + \cdots + \varphi(|\lambda_k|) \leq \varphi(\sigma_1) + \cdots + \varphi(\sigma_k)$ for $k = 1, \ldots, n$, for any increasing function φ on $[0, \infty)$ such that $\varphi(e^t)$ is convex on $(-\infty, \infty)$; Weyl was particularly interested in $\varphi(s) = s^p, \quad p > 0$, since this special case completely explained (and enormously simplified) Chang's results. Ky Fan (1949, 1950) introduced initial versions of variational characterizations of eigenvalue and singular value sums that became basic for much of the later work in this area. Pólya (1950) gave an alternative proof of a key lemma in Weyl's 1949 paper. Pólya's insight led to extensive and fruitful use of properties of doubly stochastic matrices and majorization in the study of singular value inequalities. None of these papers uses the term "singular value;" instead, they speak of the "two kinds of eigenvalues" of a matrix, namely, the eigenvalues of A and those of A^*A.

In a 1950 paper, A. Horn begins by saying, "In this note I wish to present a theorem on the singular values of a product of completely continuous operators on Hilbert space .... The singular values of an operator K are the positive square roots of the eigen-values of K^*K, where K^* is the adjoint of K." He then shows that

$$\sigma_1(AB) \cdots \sigma_k(AB) \leq \sigma_1(A) \cdots \sigma_k(A)\sigma_1(B) \cdots \sigma_k(B),$$

for all $k = 1, 2, \ldots$.

In the following year, Ky Fan (1951) extended the work in his 1949-50 papers and obtained the fundamental variational characterization of singular value sums that enabled him to prove basic inequalities. He also revisited Von Neumann's characterization of all unitarily invariant norms and showed how it follows easily

from his new insights. A novel and fundamental feature of Fan's variational characterizations of singular values is that they are quasilinear functions of A itself, not via A^*A. They are surely the foundation for all of the modern theory of singular value inequalities.

In 1954, A. Horn proved that Weyl's 1949 inequalities are sufficient for the existence of a matrix with prescribed singular values and eigenvalues, and stimulated a long series of other investigations to ascertain whether inequalities originally derived as necessary conditions are sufficiently strong to characterize exactly the properties under study. In this 1954 paper, which, unlike his 1950 paper, is clearly a paper on matrix theory rather than operator theory, Horn uses "singular values" in the context of matrices, a designation that seems to have become the standard terminology for matrix theorists writing in English. In the Russian literature one sees singular values of a matrix or operator referred to as *s-numbers* [5]; this terminology is also used in [13].

In this thesis, we provide a detailed overview of the basic singular values of matrices while focusing on the SVD and PCA as the most computational methods. We discuss how SVD and PCA are related and how to apply both SVD and PCA to the real world. For example, we applied the techniques to Lena's famous image and a neuron image to compare how both techniques can effectively decompose and highlight details of the images. MATLAB program was used to perform these tasks.

2. Singular Value Decomposition

As in [6] and [7], we start by defining key terms used throughout this thesis for clarity and cohesiveness.

2.1 Definition

We let M_n denote the set of all $n \times n$ complex matrices. We note that some authors use the notation $\mathbb{C}^{n\times n}$. Now let $A \in M_n$. Then a nonzero vector $x \in \mathbb{C}^n$ is said to be an eigenvector of A corresponding to the eigenvalue λ, if

$$Ax = \lambda x$$

The set containing all of the eigenvalues of A is called the *spectrum of* A and is denoted, $\sigma(A)$.

An *inner product* on a vector space V is an operation on V that assigns to each pair of vectors x and y in V a real number $\langle x, y\rangle$ satisfying the following conditions:
(i) $\langle x, x\rangle \geq 0$ with equality if and only if $x = 0$.
(ii) $\langle x, y\rangle = \langle y, x\rangle$ for all x and y in V.
(iii) $\langle \alpha x + \beta y, z\rangle = \alpha\langle x, z\rangle + \beta\langle y, z\rangle$ for all x, y, z in V and all scalars α and β.

A vector space V with an inner product is called an *inner product space.*

Given a vector w with positive entries, we could also define an inner product on $\mathbb{R}^n$ by

$$\langle x, y\rangle = \sum_{i=1}^{n} x_i y_i w_i.$$

The entries w_i are referred to as *weights.*

Let $v_1, v_2, ..., v_n$ be vectors in an inner product space V. If $\langle v_i, v_j\rangle = 0$ whenever $i \neq j,$ then $\{v_1, v_2, \ldots, v_n\}$ is said to be an *orthogonal set* of vectors.

An *orthonormal* set of vectors is in an orthogonal set of unit vectors. The set $\{u_1, u_2, \ldots, u_n\}$ will be orthonormal if and only if

$$\langle u_i, u_j \rangle = \delta_{ij},$$

where

$$\delta_{ij} = \begin{cases} 1, & \text{if} \quad i = j, \\ 0, & \text{if} \quad i \neq j \end{cases} .$$

Given any orthogonal set of nonzero vectors $\{v_1, v_2, \ldots, v_n\}$, it is possible to form an orthonormal set $\{u_1, u_2, \ldots, u_n\}$ by defining

$$u_i = (\frac{1}{||v_i||})v_i \quad \textit{for} \quad i = 1, 2, \ldots, n.$$

An $n \times n$ matrix A matrix $D = [d_{ij}] \in M_n$ is called a *diagonal matrix*, if $d_{ij} = 0$ whenever $i \neq j$.

An $n \times n$ matrix A is said to be an *orthogonal matrix* if the column vectors of A from an orthonormal set in $\mathbb{R}^n$.

An $n \times n$ matrix A is orthogonal if and only if $A^T A = I$. Then A is invertible and $A^{-1} = A^T$.

Let A and B be $n \times n$ matrices. B is said to be *similar* to A if there exists a nonsingular matrix S such that $B = S^{-1}AS$.

An $n \times n$ matrix A is said to be *diagonalizable* if there exists a nonsingular matrix X and a diagonal matrix D such that

$$X^{-1}AX = D.$$

We say that X *diagonalizes* A.

Let $A \in M_n$, then A is diagonalizable, if A is similar to a diagonal matrix.

A *quadratic form* on $\mathbb{R}^n$ is a fuction Q defined on $\mathbb{R}^n$ whose value at a vector x in $\mathbb{R}^n$ can be computed by an expression of the form $Q(x) = x^T Ax$, where A is an $n \times n$ symmetric matrix. The matrix A is called the *matrix of the quadratics form.*

The vector space $\mathbb{R}^{m\times n}$ is given R and S in $\mathbb{R}^{m\times n}$, we can define an inner product by

$$\langle R, S\rangle = \sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij}s_{ij}. \tag{6}$$

For the vector space $\mathbb{R}^{m\times n}$ the norm derived from the inner product (6) is called the *Frobenius norm* and is donated by $||\cdot||_F$. Thus if $G \in \mathbb{R}^{m\times n}$, then

$$||G||_F = (\langle G, G\rangle)^{1/2} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} g_{ij}^2}.$$

The diagonalization theorems play a part in many interesting applications. Unfortunately, as we know, not all matrices can be factored as $A = PDP^{-1}$ with D diagonal. However, a factorization $A = QDP^{-1}$ is possible for any $m \times n$ matrix A. A speacial factorization of this type, called is the singular value decomposition, is one of the most useful matrix factorizations in applied linear algebra.

The singular value decomposition is based on the following property of the ordinary diagonalization that can be imitated for rectangular matrices: The absolute values of the eigenvalues of a symmetric matrix A measure the amounts that A stretches or shrinks certain vectors (the eigenvectors). If $Ax = \lambda x$ and $||x|| = 1$, then

$$||Ax|| = ||\lambda x|| = |\lambda| \cdot ||x|| = |\lambda|. \tag{7}$$

If λ_1 is the eigenvalue with the greatest magnitude, then a corresponding unit eigenvector v_1 identifies a direction in which the stretching effect of A is greatest. That is, the length of Ax is maximized, when $x = v_1$, and $||Av_1|| = |\lambda_1|$, by (7). This description of v_1 and $|\lambda_1|$ has an analogue for rectangular matrices that will lead to the singular value decomposition.

Most numerical calculations involving an equation $Ax = b$ are as reliable as possible when the SVD of A is used. The two orthogonal matrices U and V do not affect lengths of vectors or angles between vectors. Any possible instabilities in numerical calculations are identified in Σ. If the singular values of A are extremely large or small, roundoff errors are almost inevitable, but an error analysis is aided by knowing the entries in Σ and V. If A is an invertible $n \times n$ matrix, then the ratio σ_1/σ_n of the largest and smallest singular values gives the condition number of A.

In practical work, we might occasionally encounter a "nearly singular" or *ill-conditioned* matrix - an invertible matrix that can become singular if some of its entries are changed ever so slightly. In this case, row reduction may produce fewer than n pivot positions, as a result of roundoff error. Also, roundoff error can sometimes make a singular matrix appear to be invertible.

Some matrix programs will compute a condition number for a square matrix. The larger the condition number, the closer the matrix is to being singular. The condition number of the identity matrix is 1. A singular matrix has an infinite condition number. In extreme cases, a matrix program may not be able to distinguish between a singular matrix and an ill-conditioned matrix. Matrix computations can produce substantial error when a condition number is large.

Let A be an $m \times n$ matrix. Then A^TA is symmetric and can be orthogonally diagonalized. Let $\{v_1, \ldots, v_n\}$ be an orthonormal basis for $\mathbb{R}^n$ consisting of eigenvectors of A^TA, and let $\lambda_1, \ldots, \lambda_n$ be the associated eigenvalues of A^TA. Then, for $1 \leq i \leq n$, we have

$$||Av_i||^2 = (Av_i)^TAv_i = v_i^TA^TAv_i = v_i^T(\lambda_i v_i) = \lambda_i v_i^T v_i = \lambda_i||v_i||^2 = \lambda_i. \qquad (8)$$

Hence the eigenvalues of A^TA are all nonnegative. By renumbering, if necessary,

we may assume that the eigenvalues are arranged in the nonincreasing order:

$$\lambda_1 \geq \lambda_2 \geq \ldots \lambda_n \geq 0.$$

The singular values of A are the square roots of the eigenvalues of $A^T A$, denoted by $\sigma_1, \ldots, \sigma_n$, and they are arranged in nonincreasing order. That is, $\sigma_i = \sqrt{\lambda_i}$ for $1 \leq i \leq n$. Remark by (8), the singular values of A are the lengths of the vectors $Av_1, \ldots, Av_n$.

Theorem 2.4 *If S is a subspace of $\mathbb{R}^n$, then $dim S + dim S^{\perp} = n$. Furthermore, if $\{x_1, \ldots, x_r\}$ is a basis for S and $\{x_{r+1}, \ldots, x_n\}$ is a basis for $S^{\perp}$, then $\{x_1, \ldots, x_r, x_{r+1}, \ldots, x_n\}$ is a basis for $\mathbb{R}^n$.*

As in [14], in many applications it is necessary to either determine the rank of a matrix, or to determine whether the matrix is deficient in rank. Theoretically, we can use Gaussian elimination to reduce the matrix to row echelon form and then count number of nonzero rows. However, this approach is not practical when working in finite-precision arithmetic. If A is rank deficient and U is the computed echelon form, then, because of roundoff errors in the elimination process, it is unlikely that U will have the proper number of nonzero rows. In practice, the coefficient matrix A usually involves some error. This may be due to errors in the data or to the finite number system. Thus, it is generally more practical to ask whether A is "close" to a rank deficient matrix. However, it may well turn out that A is close to being rank deficient and the computed row echelon from U is not.

We assume throughout this thesis that A is an $m \times n$ matrix with $m \geq n$. (This assumption is made for convenience only; all the results will also hold if $m < n$.) We will present a method for determining how close A is to a matrix of smaller rank. The method involves factoring A into a product $U\Sigma V^T$, where U is an $m \times m$ orthogonal matrix, V is an $n \times n$ orthogonal matrix, and Σ is an $m \times n$ matrix

whose off-diagonal entries are all 0's and whose diagonal elements satisfy

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0,$$

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \ldots & 0 \\ 0 & \sigma_2 & \ddots & \vdots \\ & & \ddots & 0 \\ \vdots & & \ddots & \sigma_n \\ & & & 0 \\ & & & \vdots \\ 0 & & \ldots & 0 \end{bmatrix}.$$

The σ_i's determined by this factorization are unique and are called the singular values of A. The factorization $U\Sigma V^T$ is called the Singular Value Decomposition (SVD) of A. We will show that the rank of A equals the number of nonzero singular values, and that the magnitudes of the nonzero singular values provide a measure of how close A is to a matrix of lower rank.

The following theorem shows that such a decomposition is always possible.

Theorem 2.5 (The SVD Theorem) *If A is an $m \times n$ matrix, then A has a singular value decomposition.*

Proof. A^TA is a symmetric $n \times n$ matrix. Therefore, its eigenvalues are all real and it has an orthogonal diagonalizing matrix V. Furthermore, its eigenvalues must all be nonnegative. To see this, let λ be an eigenvalue of A^TA and x be an eigenvector belonging to λ. It follows that

$$||Ax||^2 = x^TA^TAx = \lambda x^Tx = \lambda||x||^2.$$

Hence,

$$\lambda = \frac{||Ax||^2}{||x||^2} \geq 0.$$

We may assume that the columns of V have been ordered so that the corresponding eigenvalues satisfy

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0.$$

The singular values of A are given by

$$\sigma_j = \sqrt{\lambda_j}, \qquad j = 1, \ldots, n.$$

Let r denote the rank of A. The matrix $A^T A$ will also have rank r. Since $A^T A$ is symmetric, its rank equals the number of nonzero eigenvalues. Thus

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r \geq 0 \quad \text{and} \quad \lambda_{r+1} = \lambda_{r+2} = \ldots = \lambda_n = 0.$$

The same relation holds for the singular values

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r \geq 0 \quad \text{and} \quad \sigma_{r+1} = \sigma_{r+2} = \ldots = \sigma_n = 0.$$

Now let

$$V_1 = (v_1, \ldots, v_r), \quad V_2 = (v_{r+1}, \ldots, v_n)$$

and

$$\Sigma_1 = \begin{bmatrix} \sigma_1 & 0 & \ldots & 0 \\ 0 & \sigma_2 & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ 0 & & \ldots & \sigma_r \end{bmatrix}. \tag{9}$$

Thus Σ_1 is an $r \times r$ diagonal matrix whose diagonal entries are the nonzero singular values $\sigma_1, \ldots, \sigma_r$. The $m \times n$ matrix Σ is then given by

$$\Sigma = \begin{bmatrix} \Sigma_1 & O \\ O & O \end{bmatrix}.$$

The column vectors of V_2 are eigenvectors of $A^T A$ belonging to $\lambda = 0$. Thus

$$A^T A v_j = 0, \qquad j = r+1, \ldots, n$$

and, consequently, the column vectors of V_2 form an orthonormal basis for $N(A^T A) = N(A)$. Therefore,

$$AV_2 = 0$$

and, since V is an orthogonal matrix, it follows that

$$I = VV^T = V_1V_1^T + V_2V_2^T,$$

$$A = AI = AV_1V_1^T + AV_2V_2^T = AV_1V_1^T. \tag{10}$$

So far we have shown how to construct the matrices V and Σ of the singular value decomposition. To complete the proof, we must show how to construct an $m \times m$ orthogonal matrix U such that

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

or, equivalently,

$$AV = U\Sigma. \tag{11}$$

Comparing the first r columns of each side of (11), we see that

$$Av_j = \sigma_j u_j, \qquad j = 1, \ldots, r.$$

Thus, if we define

$$u_j = \frac{1}{\sigma_j} Av_j, \qquad j = 1, \ldots, r \tag{12}$$

and

$$U_1 = (u_1, \ldots, u_r),$$

then it follows that

$$AV_1 = U_1\Sigma_1. \tag{13}$$

The column vectors of U_1 form an orthonormal set since

$$u_i^T u_j = (\frac{1}{\sigma_i} v_i^T A^T)(\frac{1}{\sigma_j} Av_j) = \frac{1}{\sigma_i\sigma_j} v_i^T(A^TAv_j) = \frac{\sigma_j}{\sigma_i} v_i^T v_j = \delta_{ij},$$

where $1 \leq i \leq r, \quad 1 \leq j \leq r$.

It follows from (12) that each u_j, $1 \leq j \leq r$, is in the column space of A. The dimension of the column space is r, so $u_1, \ldots, u_r$ form an orthonormal basis for $R(A)$. The vector space $R(A)^{\perp} = N(A^T)$ has dimension $m - r$. Let $\{u_{r+1}, u_{r+2}, \ldots, u_m\}$ be an orthonormal basis for $N(A^T)$ and set

$$U_2 = (u_{r+1}, u_{r+2}, \ldots, u_m),$$

$$U = [U_1 \quad U_2].$$

It follows from Theorem 2.4 that $u_1, \ldots, u_m$ form an orthonormal basis for $\mathbb{R}^m$. Hence U is an orthogonal matrix. We still must show that $U\Sigma V^T$ actually equals A. This follows from (13) and (10) since

$$U\Sigma V^T = [U_1 \quad U_2] \begin{bmatrix} \Sigma_1 & O \\ O & O \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_1 V_1^T = A V_1 V_1^T = A. \quad \blacksquare$$

Observations: let A be an $m \times n$ matrix with a singular value decomposition $U\Sigma V^T$.

1. The singular values $\sigma_1, \ldots, \sigma_n$ of A are unique; however, the matrices U and V are not unique.

2. Since V diagonalizes $A^T A$, it follows that the v_j's are eigenvectors of $A^T A$.

3. Since $AA^T = U\Sigma\Sigma^T U^T$, it follows that U diagonalizes AA^T and that the u_j's are eigenvectors of AA^T.

4. Comparing the jth columns of each side of the equation

$$AV = U\Sigma,$$

we get

$$Av_j = \sigma_j u_j, \qquad j = 1, \ldots, n.$$

Similarly,

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

and hence

$$A^T u_j = \sigma_j v_j \qquad \text{for} \qquad j = 1, \ldots, n,$$

$$A^T u_j = 0 \qquad \text{for} \qquad j = n+1, \ldots, m.$$

The v_j's are called the *right singular vectors* of A, and the u_j's are called the *left singular vectors* of A.

5. If A has rank r, then

(**i**) $v_1, \ldots, v_r$ form an orthonormal basis for $R(A^T)$.

(**ii**) $v_{r+1}, \ldots, v_n$ form an orthonormal basis for $N(A)$.

(**iii**) $u_1, \ldots, u_r$ form an orthonormal basis for $R(A)$.

(**iv**) $u_{r+1}, \ldots, u_m$ form an orthonormal basis for $N(A^T)$.

6. The rank of the matrix A equal to the number of its nonzero singular values (where singular values are counted according to multiplicity). A similar assumption about eigenvalues is not true. For example, the matrix

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

has rank 3 even though all of its eigenvalues are 0.

7. In the case that A has rank $r < n$, if we set

$$U_1 = (u_1, u_2, \ldots, u_r), \qquad V_1 = (v_1, v_2, \ldots, v_r)$$

and define Σ_1 as in equation (9), then

$$A = U_1 \Sigma_1 V_1^T. \tag{14}$$

The factorization (14) is called the *compact form of the singular value decomposition* of A. This form is useful in many applications, including digital image processing.

If A is an $m \times n$ matrix of rank r and $0 < k < r$, we can use the singular value decomposition to find a matrix in $\mathbb{R}^{m \times n}$ of rank k that is closest to A with respect

to the Frobenius norm. Let $\mathcal{M}$ be the set of all $m \times n$ matrices of rank k or less. It can be shown that there is a matrix X in $\mathcal{M}$ such that

$$||A - X||_F = \min_{S \in \mathcal{M}} ||A - S||_F. \tag{15}$$

Assuming that the minimum is achieved, we will show how such a matrix X can be derived from the singular value decomposition of A. The following lemma will be useful.

Lemma 2.6 *If A is an $m \times n$ matrix and Q is an $m \times m$ orthogonal matrix, then*

$$||QA||_F = ||A||_F.$$

Proof.

$$||QA||_F^2 = ||(Qa_1, Qa_2, \ldots, Qa_n)||_F^2 = \sum_{i=1}^{n} ||Qa_i||_2^2 = \sum_{i=1}^{n} ||a_i||_2^2 = |||A||_F^2. \quad \blacksquare$$

It follows from the lemma that, if A has singular value decomposition $U\Sigma V^T$, then

$$||A||_F = ||\Sigma V^T||_F.$$

Since

$$||\Sigma V^T||_F = ||(\Sigma V^T)^T||_F = ||V\Sigma^T||_F = ||\Sigma^T||_F,$$

it follow that

$$||A||_F = (\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_1^2)^{1/2}.$$

Theorem 2.7 *Let $A = U\Sigma V^T$ be an $m \times n$ matrix, and let $\mathcal{M}$ denote the set of all $m \times n$ matrices of rank k or less, where $0 < k < rank(A)$. If X is a matrix in $\mathcal{M}$ satisfying* (15), then

$$||A - X||_F = (\sigma_{k+1}^2 + \sigma_{k+2}^2 + \cdots + \sigma_n^2)^{1/2}.$$

In particular, if $A' = U\Sigma' V^T$, *where*

$$\Sigma' = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 & \dots & 0 \\ 0 & \ddots & & \vdots & & \vdots \\ \vdots & & \sigma_k & & & \\ 0 & \dots & & 0 & \dots & 0 \\ \vdots & & & \vdots & & \vdots \\ 0 & \dots & & 0 & \dots & 0 \end{bmatrix} = \begin{bmatrix} \Sigma_k & O \\ O & O \end{bmatrix},$$

then

$$||A - A'||_F = (\sigma_{k+1}^2 + \cdots + \sigma_n^2)^{1/2} = \min_{S \in \mathcal{M}} ||A - S||_F.$$

Proof. Let X be a matrix in $\mathcal{M}$ satisfying (15) Since $A' \in \mathcal{M}$, it follows that

$$||A - X||_F \leq ||A - A'||_F = (\sigma_{k+1}^2 + \cdots + \sigma_n^2)^{1/2}. \tag{16}$$

We will show that

$$||A - X||_F \geq (\sigma_{k+1}^2 + \cdots + \sigma_n^2)^{1/2},$$

and hence that equality holds in (16). Let $Q\Omega P^T$ be the singular value decomposition of X.

$$\Omega = \begin{bmatrix} \omega_1 & & 0 & \dots & 0 \\ & \omega_2 & & & \vdots \\ 0 & & \ddots & & 0 \\ \vdots & & & \omega_k & \\ 0 & \dots & 0 & & 0 \end{bmatrix} = \begin{bmatrix} \Omega_k & O \\ O & O \end{bmatrix}.$$

If we set $B = Q^T A P$, then $A = QBP^T$, and it follows that

$$||A - X||_F = ||Q(B - \Omega)P^T||_F = ||B - \Omega||_F.$$

Let us partition B in the same manner as Ω.

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix},$$

B_{11} is a $k \times k$ block, B_{12} is $k \times (n-k)$, B_{21} is $(n-k) \times k$, *and* B_{22} is $(n-k) \times (n-k)$.

It follows that

$$||A - X||_F^2 = ||B_{11} - \Omega_k||_F^2 + ||B_{12}||_F^2 + ||B_{21}||_F^2 + ||B_{22}||_F^2.$$

We claim $B_{12} = 0$. If not, then define

$$Y = Q \begin{bmatrix} B_{11} & B_{12} \\ 0 & 0 \end{bmatrix} P^T.$$

The matrix Y is in $\mathcal{M}$ and

$$||A - Y||_F^2 = ||B_{21}||_F^2 + ||B_{22}||_F^2 < ||A - X||_F^2.$$

This contradicts the definition of X. Therefore, $B_{12} = 0$. In a similar manner it can be shown that B_{21} must equal 0. If we set

$$Z = Q \begin{bmatrix} B_{11} & 0 \\ 0 & 0 \end{bmatrix} P^T$$

then $Z \in \mathcal{M}$ and

$$||A - Z||_F^2 = ||B_{22}||_F^2 \leq ||B_{11} - \Omega_k||_F^2 + ||B_{22}||_F^2 = ||A - X||_F^2.$$

It follows from the definition of X that B_{11} must equal Ω_k. If B_{22} has singular value decomposition $U_1 \Lambda V_1^T$, then

$$||A - X||_F = ||B_{22}||_F = ||\Lambda||_F.$$

Let

$$U_2 = \begin{bmatrix} I_k & O \\ O & U_1 \end{bmatrix} \quad \text{and} \quad V_2 = \begin{bmatrix} I_k & O \\ O & V_1 \end{bmatrix}.$$

Now

$$U_2^T Q^T A P V_2 = \begin{bmatrix} \Omega_k & O \\ O & \Lambda \end{bmatrix}$$

$$A = (QU_2) \begin{bmatrix} \Omega_k & O \\ O & \Lambda \end{bmatrix} (PV_2)^T,$$

and hence it follows that diagonal elements of Λ are singular values of A. Thus

$$||A - X||_F = ||\Lambda||_F \geq (\sigma_{k+1}^2 + \cdots + \sigma_n^2)^{1/2}.$$

It follows from (16) that

$$||A - X||_F = (\sigma_{k+1}^2 + \cdots + \sigma_n^2)^{1/2} = ||A - A'||_F. \quad \blacksquare$$

If A has singular value decomposition $U\Sigma V^T$, then we can think of A as a product of $U\Sigma$ times V^T. If we partition $U\Sigma$ into columns and V^T into rows, then

$$U\Sigma = (\sigma_1 u_1, \sigma_2 u_2, \ldots, \sigma_n u_n),$$

and we can represent A by an outer product expansion

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_n u_n v_n^T. \tag{17}$$

If A is of rank n, then

$$A' = U \begin{bmatrix} \sigma_1 & & 0 & \ldots & 0 \\ & \sigma_2 & & & \vdots \\ 0 & & \ddots & & 0 \\ \vdots & & & \sigma_{n-1} & \\ 0 & \ldots & 0 & & 0 \end{bmatrix} V^T = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_{n-1} u_{n-1} v_{n-1}^T$$

will be the nearest matrix of rank $n - 1$ that is closest to A with respect to the Frobenius norm. Similarly,

$$A'' = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_{n-2} u_{n-2} v_{n-2}^T$$

will be the nearest matrix of rank $n - 2$, and so on. In particular, if A is a nonsingular $n \times n$ matrix, then A' is singular and $||A - A'||_F = \sigma_n$. Thus σ_n may be taken as a measure of how close a square matrix is to being singular. In most practice applications, matrix computations are carried out by computers using finite precision arithmetic. If the computation involve a nonsingular matrix that is very close to being singular, then matrix will behave computationally exactly like a singular matrix. In this case, computed solutions to linear systems may have no

digits of accuracy whatsoever. More generally, if an $m \times n$ matrix A is close enough to a matrix of rank r, where $r < \min(m, n)$, then A will behave like a rank r matrix in finite precision arithmetic. The singular values provide a way of measuring how close a matrix is to matrices of lower rank; however, we must clarify what we mean by very close. We must decide how close is close enough. The answer depends on the machine precision of the computer that is being used.

Machine precision can be measured in terms of the unit roundoff error for the machine. Another name for the unit roundoff is the *machine epsilon*. To understand this concept, we need to know how computers represent numbers. If the computer uses number base beta and keeps track of n digits, then it will represent a real number x by a *floating-point number*, denoted $fl(x)$, of the form $\pm 0.d_1 d_2 \ldots d_n \times \beta^k$, where the digits d_i are integers with $0 \leq d_1 < \beta$. For example, $-0.87654321 \times 10^{25}$ is an eight-digit, base 10 floating-point number, and $0.110100111001 \times 2^{-9}$ is a twelve-digit, base 2 floating-point number. It turns out that the machine epsilon, ϵ, is the smallest floating-point number that will serve as a bound for the relative error wherever we approximate a real number by a floating-point number; that is, for any real number x,

$$|\frac{fl(x) - x}{x}| < \epsilon \tag{18}$$

For eight-digit, base 10, floating-point number arithmetic, the machine epsilon is 5×10^{-8}. For twelve-digit, base 2, floating-point arithmetic ,the machine epsilon is $(\frac{1}{2})^{-12}$, and in general for n-digit base β arithmetic, the machine epsilon is $\frac{1}{2} \times \beta^{-n+1}$.

In light of (18) the machine epsilon is the natural choice as a basic unit for measuring roundoff errors. Suppose that A is a matrix of rank n, but k of its singular values are less than a "small" multiple of the machine epsilon. Then A is close enough to matrices of rank $n - k$ so that it is impossible to tell the difference when doing floating-point arithmetic. In this case we would say that A

has *numerical rank* $n - k$. The multiple of the machine epsilon that we use to determine numerical rank depends on the dimensions of the matrix and so on its largest singular value. The following definition of numerical rank is the one used by the software package MATLAB. In fact, the MATLAB command *rank(A)* will computer the numerical rank of A, rather than the exact rank.

Definition 2.3

The numerical rank of an $m \times n$ matrix is the number of singular values of the matrix that are greater than $\sigma_1 \max(m, n)\epsilon$, where σ_1 is the largest singular value of A and ϵ is the machine epsilon.

3. Principal Component Analysis

3.1 Historical Remarks.

Factor analysis had its start at the beginning of the twentieth century with the efforts of psychologists to identify the factor or factors that make up intelligence. The person most responsible for pioneering this field was the psychologist, Charles Spearman. In a 1904 paper, Spearman analyzed a series of exam scores at a preparatory school, using a correlation matrix.

Based on the observed correlations, Spearman concluded that the test results provided evidence of common basic underlying functions. Further work by psychologists to identify the common factors that make up intelligence has led to development of an area of study known as factor analysis.

Predating Spearman's work by a few years is a 1901 paper [17] by Karl Pearson analyzing a correlation matrix derived from measuring seven physical variables for each of 3000 criminals. This study contains the roots of a method popularized by H. Hotelling in a well-known paper [8] published in 1933. An effective way to suppress redundant information and provide in only one or two composite data most of the information from the initial data is also called principal component analysis.

To see the basic idea of this method, assume that a series of n aptitude tests are administered to a group of m individuals and that deviations from the mean for tests form the columns of an $m \times n$ matrix A. Although, in practice, column vectors of A are positively correlated, the hypothetical factors that account for the scores should be uncorrelated. Thus, we wish to introduce mutually orthogonal vectors $y_1, y_2, \ldots, y_r$ corresponding to the hypothetical factors. We require that the vectors span $R(A)$, and hence the number of vectors, r, should be equal to the rank of A. Furthermore, we wish to number these vectors in decreasing order of variance.

The first principal component vector y_1 should account for the most variance. Since y_1 is in the column space of A, we can represent it as the product Av_1 for some $v_1 \in \mathbb{R}^n$, The covariance matrix is

$$S = \frac{1}{n-1} A^T A,$$

and the variance of y_1 is given by

$$\text{var}(y_1) = \frac{(Av_1)^T Av_1}{n-1} = v_1^T S v_1.$$

The vector v_1 is chosen to maximize $v^T S v$ over all unit vectors v. This can be accomplished by choosing v_1 to a unit eigenvector of $A^T A$ belonging to its maximum eigenvalue λ_1. The eigenvectors of $A^T A$ are the right singular vectors of A. Thus, v_1 is the right singular vector of A corresponding to the largest singular value $\sigma_1 = \sqrt{\lambda_1}$. If u_1 is the corresponding left singular vector, then

$$y_1 = Av_1 = \sigma_1 u_1.$$

The second principal component vector must be of the form $y_2 = Av_2$. It can be shown that the vector that maximizes $v^T S v$ over all unit vectors that are orthogonal to v_1 is just the second right singular vector v_2 of A. If we choose v_2 in this way and u_2 is the corresponding left singular vector, then

$$y_2 = Av_2 = \sigma_2 u_2,$$

and since

$$y_1^T y_2 = \sigma_1 \sigma_2 u_1^T u_2 = 0.$$

it follows that y_1 and y_2 are orthogonal. The remaining y_i's are determined in a similar manner.

In general, the singular value decomposition solves the principal component problem. If A has rank r and singular value decomposition $A = U_1 \Sigma_1 V_1^T$ (in

compact form), then the principal component vectors are given by

$$y_1 = \sigma_1 u_1, \quad y_2 = \sigma_2 u_2, \quad \ldots, \quad y_r = \sigma_r u_r.$$

The left singular vectors $u_1, \ldots, u_n$ are the normalized principal component vectors. If we set $W = \Sigma_1 V_1^T$, then

$$A = U_1 \Sigma_1 V_1^T = U_1 W.$$

The columns of the matrix U_1 correspond to the hypothetical intelligence factors. The entries in each column measure how well the individual students exhibit that particular intellectual ability. The matrix W measures to what extent each test depends on the hypothetical factors.

3.2 PCA

PCA is a technique for simplifying a data set, by reducing multidimensional data sets to lower dimensions for analysis.

PCA is an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. PCA can be used for dimension reduction in a data set while retaining those characteristics of the data set that contribute most to its variance, by keeping lower-order principal components and ignoring higher-order ones. Such low-order components often contain the "most important" information of the data. But this is not necessarily the case, depending on the application. PCA involves a mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called "principal components". The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible.

We discuss change of variable in a quadratic form as following. If x represent a variable vector in $\mathbb{R}^n$, then a *change of variable* is an equation of the form

$$x = Py, \quad \text{or equivalently,} \quad y = P^{-1}x, \tag{19}$$

where P is an invertible matrix and y is a new variable vector in $\mathbb{R}^n$. Here y is the coordinate vector of x relative to the basis of $\mathbb{R}^n$ determined by the columns of P.

If the change of variable (19) made in a quadratic form x^TAx, then

$$x^TAx = (Py)^TA(Py) = y^TP^TAPy = y^T(P^TAP)y,$$

and the new matrix of the quadratic from is P^TAP. If P orthogonal diagonalize A, then $P^T = P^{-1}$ and $P^TAP = P^{-1}AP = D$. The matrix of the new quadratic form is diagonal.

The column of P in the theorem are called the *principal axes* of the quadratic form x^TAx. The vector y is the coordinate vector of x relative to the orthonormal basis of $\mathbb{R}^n$ given by these principal axes.

A Quadratic form $\mathcal{Q}$ is:

a. positive definite, if $\mathcal{Q}(x) > 0$ for all $x \neq 0$,

b. negative definite, if $\mathcal{Q}(x) < 0$ for all $x \neq 0$,

c. indefinite if $\mathcal{Q}(x)$ assumes both positive and negative values.

Also, $\mathcal{Q}$ is said to be positive semidefine if $\mathcal{Q}(x) \geq 0$ for all x, and $\mathcal{Q}$ is negative semidefine if $\mathcal{Q}(x) \leq 0$ for all x.

Theorem 3.1 (The Principal Axe Theorem.) *Let A be an $n \times n$ symmetric matrix. Then there is an orthogonal change of variable, $x = Py$, that transforms the quadratic form x^TAx into a quadratic form y^TDy with no cross-product term.*

Theorem 3.2 (Quadratic Forms and Eigenvalues) *Let A be an $n \times n$ symmetric matrix. Then a quadratic form x^TAx is :*

a. *positive define if and only if the eigenvalues of A are all positive,*

b. *negative define if and only if the eigenvalues of A are all negative, or*

c. *indefinite if and only if A has both positive and negative eigenvalues.*

Proof. By the Principal Axes Theorem, there exists an orthogonal change of variable $x = Py$ such that

$$\mathcal{Q}(x) = x^T Ax = y^T Dy = \lambda_1 y_1^2 + \lambda_2 y_2^2 + \cdots + \lambda_n y_n^2, \tag{20}$$

where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of A. Since P is invertible, there is a one-to-one correspondence between all nonzero x and all nonzero y. Thus the values of $\mathcal{Q}(x)$ for $x = 0$ coincide with the values of the expression on the right side of (20), which is obviously controlled the signs of the eigenvalue $\lambda_1, \ldots, \lambda_n$, in the three ways described in the theorem 3.1. ■

Theorem 3.3 *If Q is any matrix, the matrices Q^TQ and QQ^T are both symmetric.*

Let us examine the transpose of each in turn.

$$(QQ^T)^T = Q^{TT}Q^T = QQ^T,$$

$$(Q^TQ)^T = Q^TQ^{TT} = Q^TQ.$$

Theorem 3.4 *A matrix is symmetric if and only if it is orthogonally diagonalizable.*

If Q is orthogonally diagonalizable, then Q is a symmetric matrix. By hypothesis, orthogonally diagonalizable means that there exist some E such that $Q = EDE^T$, where D is a diagonal matrix which diagonalizes Q. Let us compute Q^T.

$$Q^T = (EDE^T)^T = E^{TT}D^TE^T = EDE^T = Q.$$

Evidently, if Q is orthogonally diagonalizable, it must also be symmetric.

Theorem 3.5 *A symmetric matrix is diagonalized by a matrix of its orthonormal eigenvectors.*

Restated in math, let Q be a square $n \times n$ symmetric matrix with associated eigenvectors $\{e_1, e_2, \ldots, e_n\}$. Let $E = [e_1 e_2 \ldots e_n]$ where the i^{th} column of E is the eigenvectors e_i. This theorem asserts that there exists a diagonal matrix D where $Q = EDE^T$.

Theorem 3.6 *The inverse of an orthogonal matrix is its transpose.*

Show that if Q is an orthogonal matrix, then $Q^{-1} = Q^T$. We now show that $Q^T Q = I$, where I is the identity matrix. $Q^T Q$ is exact description of the identity matrix. The definition of Q^{-1} is $Q^{-1}Q = I$. Therefore, because $Q^T Q = I$, it follows that $Q^{-1} = Q^T$.

3.1 Mean and Covariance.

For principal component analysis, let $[X_1 \quad \cdots \quad X_N]$ be a $p \times N$ matrix of observations, such as described above. The *sample mean*, M, of the observation vectors $X_1, \ldots, X_N$ is given by

$$M = \frac{1}{N}(X_1 + \cdots + X_N).$$

For $k = 1, \ldots, N$, let

$$\hat{X}_k = X_k - M.$$

The columns of the $p \times N$ matrix

$$A = [\hat{X}_1 \quad \hat{X}_2 \quad \cdots \quad \hat{X}_N],$$

have zero sample mean, and A is said to be in *mean-deviation form.*

The covariance matrix is the $p \times p$ matrix S defined by

$$S = \frac{1}{N-1} AA^T.$$

Since any matrix of the form AA^T is positive semidefinite, so is S, (see Theorem 3.2).

For an example 1, let the observation vectors are

$$X_1 = \begin{bmatrix}1\\2\\1\end{bmatrix}, \quad X_2 = \begin{bmatrix}4\\2\\13\end{bmatrix}, \quad X_3 = \begin{bmatrix}7\\8\\1\end{bmatrix}, \quad X_4 = \begin{bmatrix}8\\4\\5\end{bmatrix}.$$

We can compute the sample mean and the covariance matrix. The sample mean is

$$\frac{1}{4}(\begin{bmatrix}1\\2\\1\end{bmatrix} + \begin{bmatrix}4\\2\\13\end{bmatrix} + \begin{bmatrix}7\\8\\1\end{bmatrix} + \begin{bmatrix}8\\4\\5\end{bmatrix}) = \frac{1}{4}\begin{bmatrix}20\\16\\20\end{bmatrix} = \begin{bmatrix}5\\4\\5\end{bmatrix}.$$

We can subtract the sample mean from $X_1, \ldots, X_4$ to obtain

$$\hat{X}_1 = \begin{bmatrix}-4\\-2\\-4\end{bmatrix}, \quad \hat{X}_2 = \begin{bmatrix}-1\\-2\\8\end{bmatrix}, \quad \hat{X}_3 = \begin{bmatrix}2\\4\\-4\end{bmatrix}, \quad \hat{X}_4 = \begin{bmatrix}3\\0\\0\end{bmatrix},$$

and,

$$B = \begin{bmatrix}-4 & -1 & 2 & 3\\-2 & -2 & 4 & 0\\-4 & 8 & -4 & 0\end{bmatrix}.$$

The sample covariance matrix is

$$S = \frac{1}{3}\begin{bmatrix}-4 & -1 & 2 & 3\\-2 & -2 & 4 & 0\\-4 & 8 & -4 & 0\end{bmatrix}\begin{bmatrix}-4 & -2 & -4\\-1 & -2 & 8\\2 & 4 & -4\\3 & 0 & 0\end{bmatrix} = \frac{1}{3}\begin{bmatrix}30 & 18 & 0\\18 & 24 & -24\\0 & -24 & 96\end{bmatrix} = \begin{bmatrix}10 & 6 & 0\\6 & 8 & -8\\0 & -8 & 32\end{bmatrix}.$$

To discuss the entries in $S = [s_{ij}]$, let X represents a vector that varies over the set of observation vectors and denote the coordinates of X by $x_1, \ldots, x_p$. Then x_1, for example, is a scalar that varies over the set of first coordinates of $X_1, \ldots, X_N$. For $j = 1, \ldots, p$, the diagonal entry s_{jj} in S is called the variance of x_j.

The variance of x_j measures the spread of the values of x_j. In an example 1, then the variance of x_1 is 10 and the variance of x_3 is 32. The fact that 32 is more than 10 indicates that the set of third entries in the response vectors contains a wider spread of values than the set of first entries.

The *total variance* of the data is the sum of the variances on the diagonal of S. In general, the sum of the diagonal entries of a square matrix S is called the *trace*

of the matrix, written $\text{tr}(S)$. Thus

$$\{ \text{ total variance } \} = \text{tr}(S)$$

The entry s_{ij} in S for $i \neq j$ is called the covariance of x_i and x_j. If the covariance between x_1 and x_3 is zero because the $(1,3)$-entry in S is zero. Statisticians say that x_1 and x_3 are *uncorrelated*. Analysis of the multivariate data in $X_1, \ldots, X_N$ is greatly simplified when most or all of the variables $x_1, \ldots, x_p$ are uncorrelated, that is, when the covariance matrix of $X_1, \ldots, X_N$ is diagonal or nearly diagonal.

3.2 Principal Component Analysis

For simplicity, assume that the matrix $[X_1 \quad \cdots \quad X_N]$ is already in mean-deviation form. The goal of principal component analysis is to find an orthogonal $p \times p$ matrix $P = [u_1 \quad \cdots \quad u_p]$ that determines a change of variable, $X = PY$, or

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & \cdots & u_p \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{bmatrix}$$

with the property that the new variables $y_1, \ldots, y_p$ are uncorrelated and are arranged in order of decreasing variance.

The orthogonal change of variable $X = PY$ means that each observation vector X_k receives a "new name," Y_k, such that $X_k = PY_k$. Notice that Y_k is the coordinate vector of X_k with respect to the columns of P, and $Y_k = P^{-1}X_k = P^T X_k$ for $k = 1, \ldots, N$.

It is not difficult to verify that for any orthogonal P, the covariance matrix of $Y_1, \ldots, Y_N$ is $P^T SP$. So the desired orthogonal matrix P is one that makes $P^T SP$ diagonal. Let D be a diagonal matrix with the eigenvalues $\lambda_1, \ldots, \lambda_p$ of S on the diagonal, arranged so that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$, and let P be an orthogonal matrix whose columns are the corresponding unit eigenvectors $u_1, \ldots, u_p$. Then $S = PDP^T$ and $P^T SP = D$.

The unit eigenvectors $u_1, \ldots, u_p$ of the covariance matrix S are called the *principal components* of the data (in the matrix of observations). The *first principal component* is the eigenvector corresponding to the largest eigenvalue of S, the *second principal component* is the eigenvector corresponding to the second largest eigenvalue, and so on.

The first principal component u_1 determines the new variable y_1 in the following way. Let $c_1, \ldots, c_p$ be the entries in u_1. Since u_1^T is the first row of P^T, the equation $Y = P^T X$ shows that

$$y_1 = u_1^T X = c_1 x_1 + c_2 x_2 + \cdots + c_p x_p.$$

Thus y_1 is a linear combination of the original variables $x_1, \ldots, x_p$, using the entries in the eigenvector u_1 as weights. In a similar fashion, u_2 determines the variable y_2, and so on.

4. Application to Image Processing

As in [1], [19], [12], [10], and [15], we would like to define how images are represented in the computer. We will introduce some basic properties of images. We will also introduce MATLAB for image compression.

4.1 Digital Images

A digital image is defined as a 2D function, $f(x, y)$, where x and y denote spatial coordinates and f is called *intensity* or *gray level* at the point of (x, y). Digitization, or digitizing is the process of turning an analog signal into a digital representation of that signal. The term is often used for converting analog signal, such as printed photos into discrete values. In this case, like in normal digital photos, sampling rate refers to the resolution of the image (often measured in dots per inch). Digitizing is the primary way of storing images in form suitable for computer processing.

4.2 Digital Images Display

A computer display is an interface between the computer and the operator. Although there are other interfaces (such as a printer) the main link to the operator is usually a display monitor. To connect the computer's output to the monitor, the video adapter converts the computers instructions to a form that tells the monitor what to display.

Cathode ray tube or CRT (Display Monitor) is the picture tube of a monitor. The back of the tube has a negatively charged cathode, or electron gun. The electron gun shoots electrons down the tube and onto a positively charged screen. The screen is coated with a pattern of red, green and blue phosphor dots that will glow when struck by the electron stream. Each cluster of three dots is one pixel (picture element) A typical computer screen area has a picture of 768×1204 pixels.

The image on the monitor screen is made up from thousands of such tiny dots. The distance between pixels on a computer monitor screen is called its dot pitch

and is measured in millimeters. Most monitors have a dot pitch of .28 millimeters or less.

4.3 Pixel and Images

Short for Picture Element, a pixel is a single point in a graphic image. Graphics monitors display pictures by dividing the display screen into thousands (or millions) of pixels, arranged in rows and columns. The pixels are so close together that they appear connected.

The number of bits used to represent each pixel determines how many colors or shades of gray can be displayed. For example, in 8-bit color mode, the color monitor uses 8 bits for each pixel, making it possible to display 2 to the 8th power (256) different colors or shades of gray.

On color monitors, each pixel is actually composed of three dots – a red, a blue, and a green one. Ideally, the three dots should all converge at the same point, but all monitors have some convergence error that can make color pixels appear fuzzy.

The quality of a display system largely depends on its resolution, how many pixels it can display, and how many bits are used to represent each pixel. VGA systems display 640 by 480, or about 300,000 pixels. In contrast, SVGA systems display 800 by 600, or 480,000 pixels. True Color systems use 24 bits per pixel, allowing them to display more than 16 million different colors.

In this simple case of black and white pictures each pixel can be one of two states. This can be represented by a binary number. If a picture is a gray-scale then every pixel represents the intensity using a real number 0 (black) to the largest value (infinity). A convenient is way of representing real number the range is using $[0, 1]$. However the display device is only capable of displaying a finite number of different intensities, so we quantize the pixel into discrete levels. A common quantization used is 256 levels of intensity ($256 = 2^8$), which is convenient value in computers as each a pixel can be represented by a single 8-bit byte. At this stage

a further level of approximation has been added.

Color signals are more complicated as the information is multidimensional color space used by most computers is a 3-dimensional RGB shorthand for Red, Green, Blue space.

RGB is a convenient color model for computer graphics because the human visual system works in a way that is similar - though not quite identical - to an RGB color space. The most commonly used RGB color spaces are sRGB and Adobe RGB (which has a significantly larger gamut). Adobe has recently developed another color space called Adobe Wide Gamut RGB, which is even larger, in detriment of gamut density.

Each pixel is a 3-dimensional vector with each element representing the intensity of the primary color component red, green and blue. Most computer's color monitors use an RGB color space. Printer generally use a Cygan, Magenta, and Yellow space color. Not all color information can be encoded in 3 dimensions. Now, It is very convenient to convert RGB color image by using MATLAB. In the next section, we would like to introduce MATLAB.

4.4 Images Compression

Once a color space has been established, every pixel is stored as a vector in that space. If the the picture is $m \times n$ pixels in an RGB color space, then we need an array of $3mn$ elements to represent it. A typical picture size is 480×640 for medium resolution. This needs 921600 element to store this picture. This is approximately .09 Megabytes. A high resolution picture of 768×1024 will need 2.25 Megabytes.

Compressing an image is significantly different than compressing raw binary data. Of course, general purpose compression programs can be used to compress images, but the result is less than optimal. This is because images have certain statistical properties which can be exploited by encoders specifically designed for them. Also, some of the finer details in the image can be sacrificed for the sake

of saving a little more bandwidth or storage space. This also means that lossy compression techniques can be used in this area.

4.5 Images Processing Using MATLAB

If an image is stored as a JPEG-image on your disc we first read it into MATLAB. However, in order to start working with an image, for example perform a wavelet transform on the image, we must convert it into a different format. This section explains four common formats.

Intensity images are equivalent to a "gray scale image" and this is the image we will mostly work with in this project. It represents an image as a matrix where every element has a value corresponding to how bright/dark the pixel at the corresponding position should be colored. There are two ways to represent the number that corresponds to the brightness of the pixel: The double class (or data type). This assigns a floating number ("a number with decimals") between 0 and 1 to each pixel. The value 0 corresponds to black and the value 1 corresponds to white. The other class is called uint8 which assigns an integer between 0 and 255 to represent the brightness of a pixel. The value 0 corresponds to black and 255 to white. The class uint8 only requires roughly 1/8 of the storage compared to the class double. On the other hand, many mathematical functions can only be applied to the double class. We will see later how to convert between double and uint8 (unsigned 8-bit integer).

4.6 Binary Image

In a binary image, each pixel assumes one of only two discrete values. Essentially, these two values correspond to on and off. A binary image is stored as a two-dimensional matrix of 0's (off pixels) and 1's (on pixels). A binary image can be considered a special kind of intensity image, containing only black and white. Other interpretations are possible, however; you can also think of a binary image

as an indexed image with only two colors. A binary image can be stored in an array of class double or uint8. However, a uint8 array is preferable, because it uses far less memory. In the Image Processing Toolbox, any function that returns a binary image returns it as a uint8 logical array. The toolbox uses the presence of the logical flag to signify that the data range is [0,1]. If the logical flag is off, the toolbox assumes the data range is [0,255].

4.7 Indexed Image

This is a practical way of representing color images. (In this thesis we will mostly work with gray scale images). An indexed image stores an image as two matrices. The first matrix has the same size as the image and one number for each pixel. The second matrix is called the color map and its size may be different from the image. The numbers in the first matrix is an instruction of what number to use in the color map matrix.

4.8 RGB Image

This is another format for color images. It represents an image with three matrices of sizes matching the image format. Each matrix corresponds to one of the colors red, green or blue and gives an instruction of how much of each of these colors a certain pixel should use. RGB Images like an indexed image, an RGB image represents each pixel color as a set of three values, representing the red, green, and blue intensities that make up the color. Unlike an indexed image, however, these intensity values are stored directly in the image array, not indirectly in a colormap.

In MATLAB, the red, green, and blue components of an RGB image reside in a single m-by-n-by-3 array. m and n are the numbers of rows and columns of pixels in the image, and the third dimension consists of three planes, containing red, green, and blue intensity values. For each pixel in the image, the red, green, and blue elements combine to create the pixel's actual color.

For example, to determine the color of the pixel (112,86), look at the RGB triplet stored in (112,86,1:3). Suppose (112,86,1) contains the value 0.1238, (112,86,2) contains 0.9874, and (112,86,3) contains 0.2543. The color for the pixel at (112,86) is: 0.1238 ,0.9874, and 0.2543. An RGB array can be of class double, in which case it contains values in the range [0,1], or of class uint8, in which case the data range is [0,255].

4.9 Multi-frame Image

In some applications we want to study a sequence of images. This is very common in biological and medical imaging where we might study a sequence of slices of a cell. For these cases, the multi-frame format is a convenient way of working with a sequence of image.

4.10 Images and Linear Transformations

A color image is mathematically defined by three $m \times n$ matrices corresponding to red, green and blue.

Changing bases plays very important role in the image analysis, as in geometry where changing the origin and the coordinate axes simplify equations. Changing bases is the unified approach that explain all the transforms that appear in this thesis. If V and W are of finite-dimension, and one has chosen bases in those spaces, then every linear transformation from V to W can be represented as a matrix; this is useful because it allows concrete calculations. Conversely, matrices yield examples of linear transformations: if A is a real m-by-n matrix, then the rule $f(x) = Ax$ describes a linear transformation $R^n \to R^m$

Let $\{v_1 \dots v_n\}$ be a basis for V. Then every vector v in V is uniquely determined by the coefficients $c_1, \dots, c_n$ in

$$c_1 v_1 + \cdots + c_n v_n.$$

If $f : V \to W$ is a linear transformation

$$f(c_1v_1 + \cdots + c_nv_n) = c_1f(v_1) + \cdots + c_1f(v_n),$$

which implies that the function f is entirely determined by the values of $f(v_1), \ldots, f(v_n)$.

Now let $\{w_1, \ldots, w_n\}$ be a basis for W. Then we can represent the values of each $f(v_j)$ as

$$f(v_j) = a_{1j}w_1 + \cdots + a_{mj}w_m.$$

Thus, the function f is entirely determined by the values of $a_{i,j}$.

If we put these values into an m-by-n matrix M, then we can conveniently use it to compute the value of f for any vector in V. For if we place the values of $c_1, \ldots, c_n$ in an n-by-1 matrix C, we have $MC = f(v)$.

A single linear transformation may be represented by many matrices. This is because the values of the elements of the matrix depend on the bases that are chosen.

5. Image Compression Using SVD

A Video image or photograph such as Lena's image can be digitized by breaking it up into a rectangular array of cells (or pixels) and measuring the gray level of each cell. This information can be stored and transmitted using an $n \times n$ matrix A. The entries of A are nonnegative numbers corresponding to the measures of the gray levels. Because the gray level of any one cell generally turn out to be close to the gray levels of its neighboring cells, it is possible to reduce the amount of storage necessary from n^2 to multiple of n. Generally, the matrix A will have small singular values. As in [14], consequently, A can be approximated by a matrix of much lower rank.

If A has singular value decomposition $U\Sigma V^T$, then A can be represented by the outer product expansion

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_n u_n v_n^T.$$

The closest matrix of rank k is obtained by truncating this sum after the first k terms:

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

The total storage for A_k is $k(2n+1)$. We can chose k to be considerably less than n and still have the digital image corresponding to A_k very close to the original. For example, data compression will be used on the following problem. We are supposed to send Lena's images from Thailand to America. The computer digitizes the picture by subdividing it into tiny square called pixels or picture elements. Each pixel is represented by a single number that records the average light intensity in that square. If each photograph were divided into 256×256 pixels, it would have to send 66,536 numbers to Thailand for each picture. (This amounts to a $256D$ matrix.) This would take a great deal of time and would limit the number of

photographs that could be transmitted. It is sometimes possible to approximate this matrix with a 'simple' matrix which requires less storage.

Let us start by looking at the Lena's picture of only 256 pixels. The gray level specification for each square is determined by the corresponding entry in the matrix A. To reproduce this picture exactly, we need to keep track of all 256 ($= 16 \times 16$) entries of A. However, using SVD analysis of A, we can approximately reproduce this image keeping track of much less data. To compress the data, we set the smaller singular values to zero. Using only the first 3 singular values of A the data transmits the following data for the compressed image: the first 3 rows of U, the first 3 singular values and the first 3 rows V. And then the recipient in Thailand form the 16D matrix.

$$A_3 = \sigma_1 u_1^T v_1 + \sigma_2 u_2^T v_2 + \sigma_3 u_3^T v_3$$

The data transmits $3(2 \times 16 + 1) = 99$ elements. This is a lot less than the 256 elements. This can save great deal of time.

As in [9] and [10], we want to compare how well the data transmits using SVD or PCA method. We will compare our experiment by Frobenius norm, which is a good way of measuring error. We also want to measure the compression efficiency. Last, but not least the Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) is most interesting to use for comparison in our experiment. In this section, we mathematically define those methods which we use for comparing PCA and SVD for image compression.

In many cases, it is useful to associate an energy to an image. If we want to compress an image, we usually want to remove as much data as possible without losing too much energy of the image. The difference in energy between the original image and the compressed image is one possible way to design an error measure when compressing images. However, it is important to notice that this measure

is not always appropriate as an error measure. In some cases the energy error measure might be small, while the visual error is terrible. One difficult problem in image processing is to find a good way of measuring error so that the error measure "agrees with our eyes". In other words, "what you see is what you get". Let a digital image be represented as the $m \times n$ matrix M and denote its elements as M_{ij}, $i = 1, 2, 3, \ldots, m$ and $j = 1, 2, 3, \ldots, n$. Frobenius norm of the matrix M is

$$||M||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |M_{ij}|^2}. \tag{21}$$

This is not the most common matrix norm, and not always very convenient, but it is related to the so-called L^2-norm for functions (l^2-norm for sequences). The L^2-norm (and the l^2-norm)is related to the concept of energy and for this reason the Frobenius matrix norm is sometimes useful in image processing. Note that in our case when we work with matrices representing images our matrix elements are real. Let M_c represent a compressed version of the image M. We define the relative (Frobenius) error E_{rel} as

$$E_{rel} = \frac{||M - M_c||_F}{||M||_F}. \tag{22}$$

When we compress data, a measurement of the error that the compression introduces is needed. We also need to think of the algorithm to measure how efficient our compression is in terms of information we need to store. Let n_M be a number representing the the number of 'memory units' we need to represent an image M and let n_{M_c} be a number representing the number of memory units we need to represent a compressed version of the image. (Here M_c denotes the compressed version of the image M).

Lossless compression involves with compressing data which, when decompressed, will be an exact reproduction of the original data. This is the case when binary data such as executables, documents, etc. are compressed. They need to be exactly reproduced when decompressed. On the other hand, images (and music too) need

not be reproduced 'exactly'. An approximation of the original image is enough for most purposes, as long as the error between the original and the compressed image is tolerable.

Two of the error metrics used to compare the various image compression techniques are the Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR). The MSE is the cumulative squared error between the compressed and the original image, whereas PSNR is a measure of the peak error . The mathematical formula for the two are

$$MSE = \frac{1}{mn}\sum_{x=0}^{m-1}\sum_{y=0}^{n-1}[M(x,y) - M'(x,y)]^2, \tag{23}$$

$$PSNR = 20log10(\frac{255}{\sqrt{MSE}}). \tag{24}$$

where $M(x,y)$ is the original image, $M'(x,y)$ is the approximated version (which is actually the decompressed image) and m,n are the dimensions of the images. A lower value for MSE means lesser error, and as seen from the inverse relation between the MSE and PSNR, this translates to a high value of PSNR. Logically, a higher value of PSNR is good because it means that the ratio of Signal to Noise is higher. Here, the 'signal' is the original image, and the 'noise' is the error in reconstruction. So, if we find a compression scheme having a lower MSE (and a high PSNR), we can recognize that it is a better one. In the next section, we display some result for the image compression using SVD.

6. Edge Detection Neuron Images and Noise Reduction Using SVD

6.1 Edges Detection

As in [4], in an image, an edge is a curve that follows a path of rapid change in image intensity. Edges are often associated with the boundaries of objects in a scene. Edge detection is used to identify the edges in an image. This function looks for places in the image where the intensity changes rapidly, using one of these two criteria:

1. Places where the first derivative of the intensity is larger in magnitude than some threshold.

2. Places where the second derivative of the intensity has a zero crossing.

Edge provides a number of derivative estimators, each of which implements one of the definitions above. For some of these estimators, you can specify whether the operation should be sensitive to horizontal edges, vertical edges, or both. edge returns a binary image containing 1's where edges are found and 0's elsewhere.

If we want to emphasize edges in an image, it is necessary for some of the weights in the filter to be negative. In particular, sets of weights that sum to zero produce, as output, differences in pixel values in the input image. A first-derivative row filter gives, at a position in the output image, the difference in pixel values in columns on either side of that location in the input image. Therefore the output from the filter will be large in magnitude (either negative or positive) if there is a significant difference in pixel values to left and right of pixel location. One set of weight, which also involves some smoothing in average over 3 rows, is given by

$$w = \frac{1}{6}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}.$$

This choice of weight gives a separable filter, achieves some noise reduction and, in

a sense to be explained below, estimates the first derivative of the image intensity in the row direction.

To show that the output does estimate the first derivative, we consider the image intensity $f(y,x)$ to be specified for continuously varying row index y and column index x, and to be differentiable. By taking a Taylor series expansion, f near (i,j) can be approximated by f_{ij}, together with a sum of partial derivatives of f with respect to y and x evaluated at (i,j):

$$f_{i+k,j+1} \approx f_{ij} + k\frac{\partial f_{ij}}{\partial y} + l\frac{\partial f_{ij}}{\partial x} + k^2\frac{\partial^2 f_{ij}}{2\partial y^2} + l^2\frac{\partial^2 f_{ij}}{2\partial x^2} + kl\frac{\partial^2 f_{ij}}{\partial y \partial x}.$$

It is found, after messy algebra in which most terms cancel, that for the above array of weights,

$$\sum_{k=-1}^{1}\sum_{l=-1}^{1} w_{kl} f_{i+k,j+1} \approx \frac{\partial f_{ij}}{\partial x},$$

i.e. a first derivative in the direction.

Now we can derive Sobel's Gradient filter. First of all we return to the calculus notation, the maximum gradient at a point (i,j) is given by

$$\sqrt{(\frac{\partial f_{ij}}{\partial x})^2 + (\frac{\partial f_{ij}}{\partial y})^2}.$$

We can design a filter to estimate this gradient by replacing the partial derivatives above by estimates of them : the outputs from first-derivative row and column filter,

$$\widehat{\frac{\partial f_{ij}}{\partial x}} = \frac{1}{6}(f_{i-1,j+1} + f_{i,j+1} + f_{i+1,j+1} - f_{i-1,j-1} - f_{i,j-1} - f_{i+1,j-1})$$

and

$$\widehat{\frac{\partial f_{ij}}{\partial y}} = \frac{1}{6}(f_{i+1,j-1} + f_{i+1,j} + f_{i+1,j+1} - f_{i-1,j-1} - f_{i-1,j} - f_{i-1,j+1}).$$

This estimate of the maximum gradient is known as *Prewitt's filter*. Larger pixel values of the filter output are shown darker in the display, and zero values are shown as white.

6.01 Sobel's filter

It is very similar, except that in estimating the maximum gradient it gives more weight to pixels nearest to (i, j), as follows:

$$\widehat{\frac{\partial f_{ij}}{\partial x}} = \frac{1}{8}(f_{i-1,j+1} + 2f_{i,j+1} + f_{i+1,j+1} - f_{i-1,j-1} - 2f_{i,j-1} - f_{i+1,j-1})$$

and

$$\widehat{\frac{\partial f_{ij}}{\partial y}} = \frac{1}{8}(f_{i+1,j-1} + 2f_{i+1,j} + f_{i+1,j+1} - f_{i-1,j-1} - 2f_{i-1,j} - f_{i-1,j+1}).$$

For both Prewitt's and Sobel's filter, the direction of maximum gradient can also obtained as

$$\phi_{ij} = tan^{-1}(\widehat{\frac{\partial f_{ij}}{\partial y}} / \widehat{\frac{\partial f_{ij}}{\partial x}}),$$

measured clockwise from horizontal the direction of increasing x.

6.2 Noise Reduction

Neuron images are affected to some extent by *noise* that is unexplained variation in data: disturbances in image intensity that are not either interpretable or not of interest. We try to find a way to simplify if this noise can be *filtered out*. In an analogous way, filters are used in chemistry to free liquids from suspended impurities by passing them through a layer of sand or charcoal. SVD and PCA method can accentuate features of interest in data and also aid to visual interpretation of images.

The present project relates to digital image processing, and more particularly to an improved method of removing noise from an image by using singular value decomposition and principal component analysis in Chapter 2, 3, 4, and 5. Digital images captured by any electronic devices such as a camera is often corrupted with noise. In the case of cameras for use with video recorders, a charge-coupled device (CCD) is an image sensor, consisting of an integrated circuit containing an array of linked or coupled, light-sensitive capacitors. This device is also known as a Color-Capture Device, which is known to introduce a significant amount of noise.

There are two methods of removing noise from images. These can be divided into linear and non-linear filtering techniques. In a linear filtering technique a pixel in an image is replaced by a weighted sum of nearby pixels. An isolated pixel value generated by a noise event will be altered to a value that is nearer to that of the nearby pixels when this averaging operation is applied; hence, the noise is reduced. Unfortunately, this type of filter also blurs edges that are not generated by noise. Hence, overall picture quality is reduced as well. Non-linear methods, in principle, can remove noise without significantly altering the overall picture quality.

Briefly and in general terms, a noise filter removes noise from an image by means of singular value decomposition and principal component analysis. The unfiltered image is provided in the form of a numerical array to a digital processing device such as MATLAB. The SVD and PCA transform is repeatedly applied to the image, each time using a different value of rank, to determine the degree of compressibility provided by each such value. The results of these transforms are compared to PCA transform. The SVD transform is then applied to all the rank using MATLAB. Finally, the SVD and PCA, techniques are shown to have a close result to reduce noise of the image and to provide the filtered image. Therefore, it is the objective of the section to provide an improved image and an improved method of filtering noise from images. Another objective of this section is to provide an image noise filter and filtering method that does not significantly degrade the image.

7. Experimental Result of SVD

7.1 The SVD and PCA Approximation of an Image.

The image processing techniques described in Chapter 4, 5, and Chapter 6 are applied to Lena's image and a neuron image of a Cray fish. We will reconstruct the favorite Lena's image using SVD and PCA of rank 2, 5, 10, 25, 50, and 256 (original image) and we will compare the results to the original image. Initially ,we use Lena's image with gray-scale. We applied different ranks values to the picture, 256×256 pixels each. The computer took five minutes (on Mobile Intel (R) Pentium (R) 4-M CPU 2.20 Gigahertz and 1.00 Gigahertz RAM) to load all images into its memory and perform the required calculation.

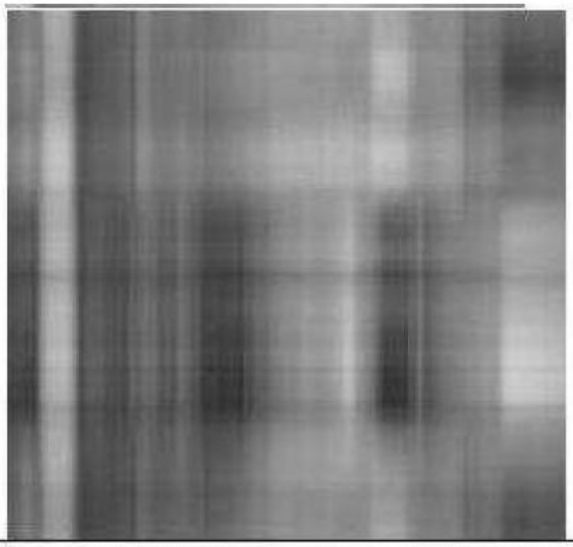
Then we calculated the relative (Frobenius) error (E_{rel}) and Peak Signal to Noise Ratio (PSNR). Mean Square is a product of Peak Signal to Noise Ratio. MATLAB program was used to calculate the error as in Figure 1. We used MATLAB to reconstruct Lena's image by using PCA, as shown in Figure 2. PCA method of rank 2-25 have higher PSNR than SVD, as shown in Figure 1. But SVD method of rank 50-256 are doing better job for compressing Lena's image, as shown in Figure 4(a). In this task, we simply need to get the lowest value of relative (Frobenius) error. SVD method gives slightly better result than PCA method. The relative error of the image being measured is shown in Figure 4(b). We used edge detection to highlight the edge to measure the amount of details in each image as shown, in Figure 5(g).

Figure 1

I=132.5377 (original)

Method	Ic	Erel	MSE	PSNR
SVD rank 2	127.6614	0.0368	0.0194	65.2616
PCA rank 2	127.4055	0.0387	0.0188	65.3877
SVD rank 5	129.8232	0.0205	0.0109	67.7696
PCA rank 5	129.5301	0.0227	0.0098	68.2174
SVD rank 10	131.1686	0.0103	0.0055	70.7200
PCA rank 10	131.0619	0.0111	0.0054	70.8171
SVD rank 25	132.0320	0.0047	0.0020	75.0314
PCA rank 25	131.9204	0.0047	0.0020	75.0770
SVD rank 50	132.3699	0.0013	0.0007	79.8167
PCA rank 50	132.2481	0.0022	0.0008	79.2752
SVD rank 256	132.5377	0.0000	0.0000	infinity
PCA rank 256	132.5377	0.0000	0.0000	infinity

Figure 2a, 2b, 2c and 2d



(a) Reconstruct the favorite Lina's image using PCA of rank 2

(b) Reconstruct the favorite Lina's image using PCA of rank 5

(c) Reconstruct the favorite Lina's image using PCA of rank 10

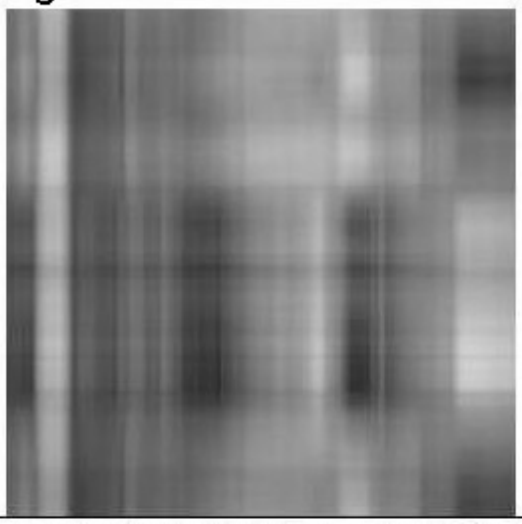
(d) Reconstruct the favorite Lina's image using PCA of rank 25

Figure 2e and 2f

(e) Reconstruct the favorite Lina's image using PCA of rank 50

(f) Reconstruct the favorite Lina's image using PCA of rank 256

Figure 3a and 3b



(a) Reconstruct the favorite Lina's image using SVD of rank 2

(b) Reconstruct the favorite Lina's image using SVD of rank 5

Figure 3c, 3d, 3e and 3f

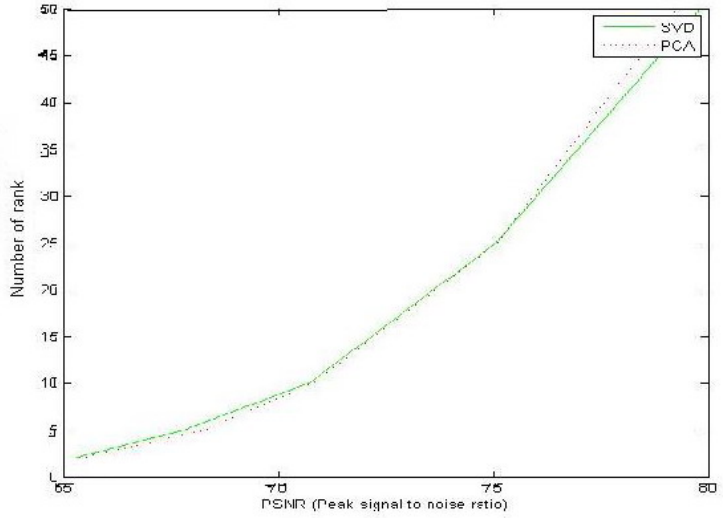
(c) Reconstruct the favorite Lina's image using SVD of rank 10

(d) Reconstruct the favorite Lina's image using SVD of rank 25

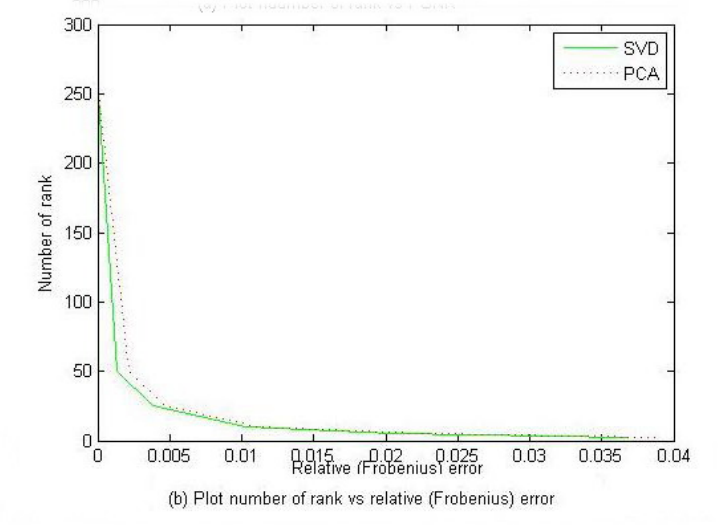
(e) Reconstruct the favorite Lina's image using SVD of rank 50

(f) Reconstruct the favorite Lina's image using SVD of rank 256

Figure 4a and 4b.

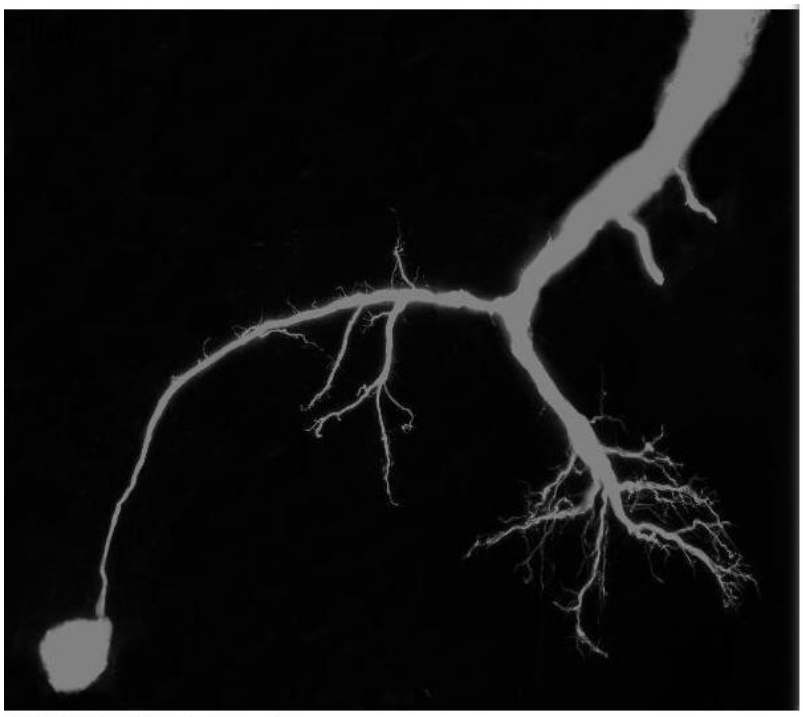


(a) Plot nuumber of rank vs PSNR

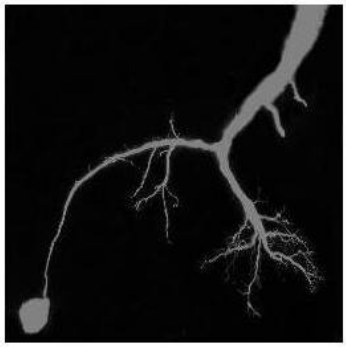


(b) Plot number of rank vs relative (Frobenius) error

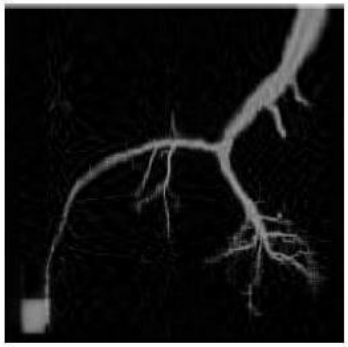
Figure 5a, 5b, 5c and 5d



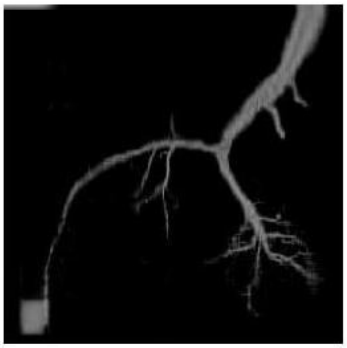
(a) Original Neuron image of Clay fish



(b) The neuron image resize to 256x256 pixel

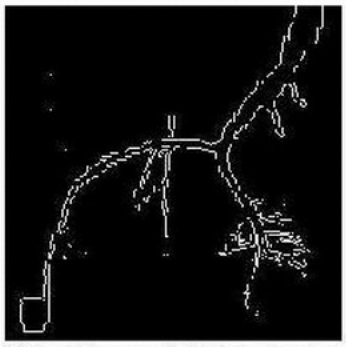


(c) The neuron image applied PCA rank 25

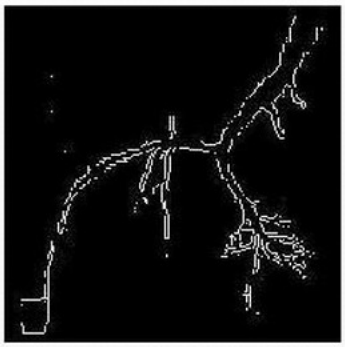


(d) The neuron image applied SVD rank 25

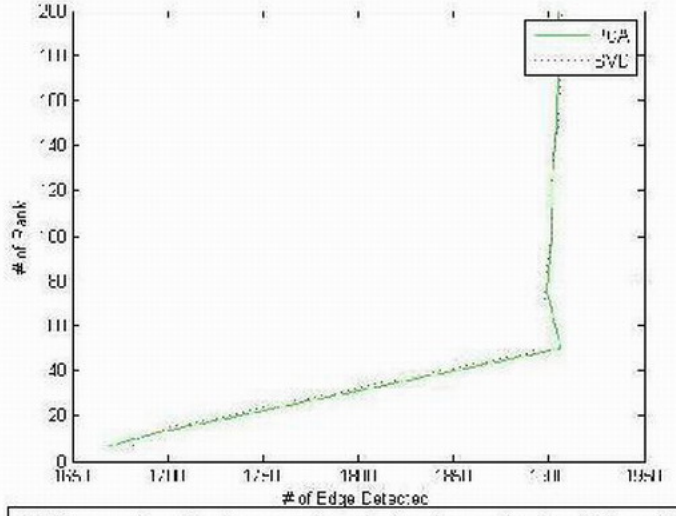
Figure 5e, 5f and 5g.



(c) The (c) image applied Sobel's edge detection



(d) The (d) image applied Sobel's edge detection



(g) Plot number of rank vs number of edge detected using SVD and PCA

8. Conclusion

The singular value decomposition is a main tool for performing principal component analysis in practical applications. If B is a $p \times N$ matrix of observations meandeviation form, and if $A = (1/\sqrt{N-1})B^T$, the A^TA is the covariance matrix, S. The squares of the singular values of A are the p eigenvalues of S, and the right singular vectors of A are the principal components of the data.

The PCA and SVD techniques of decomposition used here are extremely successful. We found that the PCA method is slightly better than SVD method, but iterative calculation of the SVD of A is faster and more accurate than an eigenvalue decomposition of S.

The 'signal' is the original Lena's image and the 'noise' is the error from the image reconstruction. We find a compression scheme that having a lower MSE means a higher PSNR. As shown in Figure 1, PCA method gives better results than SVD method. In addition, it is extremely important for Lena's image to be oriented in order to have the most efficient results. Although the rank of the images is increased, we were still able to obtain a successful output in comparison to that of the Lena's image.

PCA method gave more details than SVD method after we used Sobel's edge detection to highlight details. It is extremely helpful in reducing the noise and highlight details of images.

Further research in this area could be focusing on testing other images or RGB color images or finding the most effective method to decompress images. We would like to see the most effective method to decompose digital images that takes small amount of time to calculate and give us more details of the images.

References

[1] S. Belkasim, Advance Digital Image Processing, 2007.

[2] L. David, Linear Algrebra and It's Applications, Addison-Wesley, NY, edit. 7, pp.441-486, 2000.

[3] B. Christopher, Neural Network for Pattern Recognition, section 8.6 pp 310-319, 1996.

[4] C. A. Glasbey and G. W. Horgan Image Analysis for The Biological Sciences pp. 23-91, 1995

[5] I. Gohberg and M.G. Krein, Introduction to the Theory of Linear Nonselfadjoint Operatiors. Translations of Mathematical Monographs, Vol. 18, American Mathematical Society, Providence, R.I., 1969.

[6] R. A. Horn and C. R. Johnson. Matrix Analysis. Cambridge University Press, Cambridge, 1985.

[7] R. A. Horn and C. R. Johnson. Matrix Analysis. Cambridge University Press, Cambridge, pp.134-144, 1991.

[8] H. Hotelling, Analysis of a Complex of Statistical Variables in Principal Components, Journal of Educational Psychology, 26, 1933.

[9] D. Kalman, A Singularly Valuable Decomposition: The SVD of a Matrix, The College Mathematics Journal, Vol. 27, NO. 1, January 1996.

[10] S. Kumar, An Introduction to Image Compression S, 2001-2003.

[11] C. C. MacDuffee. The Theory of Matrices. J. Springer, Berlin, 1933; Chealsea, New York, 1973-5.

[12] A. E. Maxwell, Multivariance Analysis in Behavioral Research, Chapman and Hall, London, 1977.

[13] A. Pietsch. Eigenvalues and s-Numbers. Cambridge University Press, Cambridge, 1987.

[14] J. L. Steven, Linear Algebra with Applications, edit.6, pp 236-237, and pp 367-378, 2002.

[15] K. Sandberg, Introduction to Image Processing in MALAB 1, Department of Applied Mathematics, University of Colorado at Boulder.

[16] J. Shlens, A Tutorail on Principal Component Analysis, version 2, pp 1-11, 2005.

[17] C. Spearman, General Intelligence, Objectively Determined and Measured, American Journal of Psychology, 15, 1904.

[18] W. Todd, Introduction to The Singular Value Decompossition, Davidson college,*www.davidson.edu/academic/math/will/svd/index.html* 1999.

[19] A. B. Watson, Digital Images and Human Version, The Massachusetts Instution of Technology Press, 1993.

[20] A. G. Akritas and G. I. Malaschonok, Application of Singular-value Decomposition (SVD), Tambov University, Tambov, Russia, 2004.