Online Signature Verification using SVD Method

By

ROUSHANAK RAHMAT (8793)

FINAL REPORT

Submitted to the Electrical & Electronics Engineering Program
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

June 2009

CERTIFICATION OF APPROVAL

Online Signature Verification using SVD Method

By

Roushanak Rahmat (8793)

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

Dr. Nidal Kamel (Project Supervisor)
Ms. Norashikin Yahya (Project Co-Supervisor)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2009

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Roushanak Rahmat

Roushanak Rahmat

ABSTRACT

Online signature verification rests on hypothesis which any writer has similarity among signature samples, with scale variability and small distortion. This is a dynamic method in which users sign and then biometric system recognizes the signature by analyzing its characters such as acceleration, pressure, and orientation. The proposed technique for online signature verification is based on the Singular Value Decomposition (SVD) technique which involves four aspects: 1) data acquisition and preprocessing 2) feature extraction 3) matching (classification), 4) decision making. The SVD is used to find r-singular vectors sensing the maximal energy of the signature data matrix A, called principle subspace thus account for most of the variation in the original data. Having modeled the signature through its r-th principal subspace, the authenticity of the tried signature can be determined by calculating the average distance between its principal subspace and the template signature. The input device used for this signature verification system is 5DT Data Glove 14 Ultra which is originally design for virtual reality application. The output of the data glove, which captures the dynamic process in the signing action, is the data matrix, A to be processed for feature extraction and matching. This work is divided into two parts. In part I, we investigate the performance of the SVD-based signature verification system using a new matching technique, that is, by calculating the average distance between the different subspaces. In part II, we investigate the performance of the signature verification with reduced-sensor data glove. To select the 7-most prominent sensors of the data glove, we calculate the F-value for each sensor and choose 7 sensors that gives the highest F-value.

ACKNOWLEDGEMENTS

One of the greatest pleasures of writing a report is acknowledging the effort of many people whose names may not appear on the cover but without those hardworking, cooperation, friendship and understanding, producing this report could have been impossible:

First and foremost, I would like to thank my supervisor, Dr. Nidal Kamel who found time in a very busy schedule to teach me about concepts of my FYP, monitor my progress and answer my questions. I am also deeply grateful for his advice, encouragement and patience duration my FYP1and FYP2.

Second, I would like to thank my co supervisor, Ms. Norashikin Binti Yahya for her backing, attention and time. I have very much benefited from her professional and personal advices.

Third, I like to express my gratitude to my family, specially my parents for their supports which helped me to develop my confidence to handle my job.

Finally, I would like to express my thanks to internal and external examiners for taking the time to evaluate this report.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SVD: Singular Value Decomposition

FAR: False Accept Rate

FRR: False Reject Rate

EER: Error Equal Rate

IBDG: Image Based Data Glove

PCA: Principle Component Analysis

SF: Similarity Factor

CHAPTER 1
INTRODUCTION

This chapter presents an introduction to signature verification technique which covers the research trend and the past work of online signature verification. This is mostly covered in section 1.1. The problem statement and the objective of this work are elaborated in section 1.2 and 1.3 respectively.

1.1 Background Study

Signature verification is a common and important behavioral biometric to recognize human beings for purpose of establishing their authority. Signature is commonly used to complete an automated matter, gaining physical entry to a protected area or gaining control of a computer. Online methods in signature verification improved the accuracy of the verification due to the dynamic properties of the signatures which are considered in these methods.

A number of biometric techniques have been proposed for personal identification in the past. The technique can be categorized as vision–based and non vision-based. Among the vision-based ones are face recognition, fingerprint recognition, iris scanning and retina scanning. Voice recognition or signature verification are the most widely known among the non-vision based methods. Signature verification requires the use of electronic tablets or digitizers for online capturing and optical scanners for offline conversion.

Signature verification is an important research area in the field of person authentication. The literature on signature verification is quite extensive and shows two main areas of research, offline and online systems. Offline systems deal with a static image of the signature, i.e. the result of the action of signing while online systems work on the dynamic process of generating the signature, i.e. the action of signing itself.

Research on online signature verification systems is widespread while those on offline are not many. Offline signature verification is not a trivial pattern recognition problem when it comes to skilled forgeries. This is because, as opposed to the online case, offline signature lacks any form of dynamic information. In the past some authors have worked in simple forgeries while others have dealt with the verification of skilled forgeries.

In 1999, in the field of dynamic signature verification, Tolba presented the first attempt to use a virtual reality glove as an input device to capture behavioral data from users performing their hand signature [1]. The acquisition data from the glove was based on the optical-fiber sensor situated at each finger, which captured 256 different positions. Results from Tolba showed 0% error rate with 100% confidence by combining 21 correlated features with a 6×10 matrix.

Kamel, Sayeed and Ellis in [2] proposed glove-based approach to online signature verification. The proposed technique is based on the SVD to find r singular vectors sensing the maximal energy of the signature data matrix A, called principle subspace thus account for most of the variation in the original data. Having identified data glove signature through its r-principle subspace, the authenticity can then be obtained by calculating the angles between the different subspaces. This SVD-based signature verification technique reported an EER of 2.46% which demonstrate the good potential in their proposed technique.

This research here will continue [2] which is using SVD technique for glove-based online signature verification. The SVD is used to reduce the dimensionality of the data glove output matrix, and extracting a number of unique features for signature classification (matching). In other words, the algorithm in [2] will be applied to a new data set, which is consisting of five to six sensors instead of fourteen sensors as in [2].

1.2 Problem Statement

Handwriting is a personal biometric that is thought to be unique to an individual and hence can be used to identify a person. As a result, the use of handwritten signatures has been, from early history, a legally accepted means of authenticating various documents. In addition, in some criminal cases, analysis of handwriting is often performed by forensic document examiners to determine the authorship of a questioned document.

This work will investigate the glove base online signature verification technique by using SVD technique. The performance of SVD is being tested against three types of forgeries; skilled, casual and random forgeries. Skilled forgery is produced when the forger has unrestricted access to one or more samples of the writer's actual signatures. A casual forgery is produced when the forger is familiar with the writer's name, but does not have access to a sample of the actual signature. A random forgery can be any random scribble, a genuine signature or a high quality forgery for other writer.

Figure 1 shows one genuine signature (on the right side) and its forger signature (on the left side) which can be categorized as the skilled forgeries.

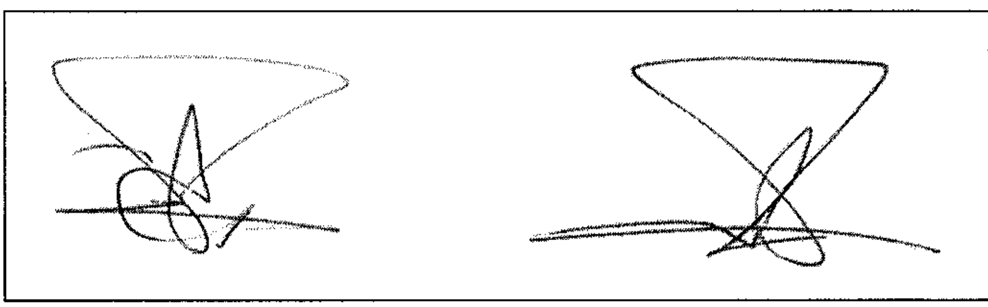


Figure 1: Samples of a genuine signature and its skilled forgery

1.3 Objectives and Scope of Study

As mentioned earlier in [2] the data matrix A, has dimension of fourteen sensors and in the first part of this work, the performance of algorithm in [2] will be investigated on a different set of data. This work will also test a new feature classification technique namely average error distance. This technique has advantage over the one in [2] because it will be able to solve the problem when two subspaces are very close to each other. This scenario will occur if the forged signature is almost the same as the genuine signature.

In the second part, the performance of the signature verification system will be evaluated with reduced-sensor data glove where the SVD method is applied to a new data matrix, A_r. A similar approach as in the previous paragraph will be used in evaluating the reduced-sensor system.

The scopes of study for this project are:

- Dynamic/Online signature verification technique
  - Data acquisition
  - Feature extraction
  - Matching
  - Decision
  - Performance evaluation

- Concept of linear algebra
  - Matrix and linear transformation
  - Vector spaces
  - Eigenvalues an eigenvectors
  - Orthogonal projection

- Philosophy of SVD and its applications in signature verification technique
  - Dimensionality reduction and feature extraction
  - Conceptual relations between SVD and oriented energy

- MATLAB programming

CHAPTER 2

LITERATURE REVIEW

In this chapter, research areas topics related to this thesis are elaborated. This chapter can be divided into three main sections namely 1) linear algebra concepts that is related to SVD, 2) biometric recognition systems and 3) data glove. Section 2.1 - 2.3 cover the fundamental concept of linear algebra. These include matrices and linear transformation, vector space and subspace, and eigenvalue and eigenvector. The general principle of biometric recognition system is covered in section 2.4 and signature verification is an example of such a system. These include the explanation on different classification error and method of assessing the biometric systems.

2.1 Matrices and Linear Transformation

The most common use of linear algebra is to solve systems of linear equations, which arise in applications to such diverse disciplines as physics, biology, economics, engineering and sociology. The systems of linear equations can be written compactly, using arrays called matrices and vectors. More importantly, the arithmetic properties of these arrays called matrices and vectors. This section begins by developing the basic properties of matrices and vectors and will continue by linear transformation, which helps to understand the basic of linear algebra. Most of the materials covered in this section are obtained mainly from [3] and [4].

2.1.1 Matrices and Vectors

Many types of numerical data are best displayed in two-dimensional arrays. For example the sold products of two bookstores from a same company in the same day can be represented by the following table:

store	1	2
Newspaper	7	10
Magazines	16	20
Books	45	45

This information can be easily represented as: $\begin{bmatrix} 7 & 10 \\ 16 & 20 \\ 45 & 45 \end{bmatrix}$

Such a rectangular array of real numbers is called a matrix. It is customary to refer to real numbers as scalars (originally from the word scale) when working with a matrix.

A matrix (plural, matrices) is a rectangular array of scalars. If the matrix has m rows and n columns, the size of the matrix is m by n, written $m \times n$. The scalar in the ith column is called the (i, j)-entry of the matrix. If A is a matrix, its (i, j)-entry can be denoted by a_{ij} and two matrices can be equal if they have the same size and have equal corresponding entries.

2.1.2 Matrix Sums and Scalar Multiplication

Matrices are more than convenient devices for storing information. Their usefulness lies in their arithmetic [4]. A matrix addition is the operation of adding two matrices by adding the corresponding entries together. The sum of two m-by-n matrices A and B, denoted by $A + B$, is again an $m \times n$ matrix computed by adding corresponding elements. For example:

$$\begin{bmatrix} a1 & a2 \\ a3 & a4 \\ a5 & a6 \end{bmatrix} + \begin{bmatrix} b1 & b2 \\ b3 & b4 \\ b5 & b6 \end{bmatrix} = \begin{bmatrix} a1+b1 & a2+b2 \\ a3+b3 & a4+b4 \\ a5+b5 & a6+b6 \end{bmatrix}$$

Subtraction of matrices is similar to their summation, and is possible as long as they have the same dimensions. $A - B$ is computed by subtracting corresponding elements of A and B, and has the same dimensions as A and B.

The scalar multiplication of a matrix $A = (a_{ij})$ and a scalar r gives a product rA of the same size as A. The entries of rA are given by:

$$(rA)_{ij} = r.a_{ij} \tag{1}$$

For example, if

$$A = \begin{bmatrix} a1 & a2 \\ a3 & a4 \\ a5 & a6 \end{bmatrix}$$

Then

$$7A = \begin{bmatrix} 7.a1 & 7.a2 \\ 7.a3 & 7.a4 \\ 7.a5 & 7.a6 \end{bmatrix}$$

The power of linear algebra lies in the natural relations between operations of matrix addition and scalar multiplication. Properties of matrix addition and scalar multiplication [4]:

- $A + B = A + B$ (2)
- $(A + B) + C = A + (B + C)$ (3)
- $A + 0 = A$ (4)
- $A + (-A) = 0$ (5)
- $(st)A = s(tA)$ (6)
- $s(A + B) = sA + sB$ (7)
- $(s + t)A = sA + tA$ (8)

2.1.3 Matrix Transpose

The transpose of a matrix A is another matrix A^T created by any one of the following equivalent actions:

- write the rows of A as the columns of A^T
- write the columns of A as the rows of A^T
- reflect A by its main diagonal (which starts from the top left) to obtain A^T

Formally, the transpose of an $m \times n$ matrix A is the $n \times m$ matrix

$$A_{ij}^T = A_{ji} \quad \text{when} \quad 1 \leq i \leq n \;, 1 \leq j \leq m \tag{9}$$

Properties of the transpose [4]:

- $(A + B)^T = A^T + B^T$ (10)
- $(sA)^T = sA$ (11)

- $(A^T)^T = A$ (12)

2.1.4 *Inverse of a Matrix and Determinant*

In linear algebra, an n-by-n (square) matrix A is called invertible or non-singular if there exists an n-by-n matrix B such that

$$AB = BA = I_n \tag{13}$$

where I_n denotes the $n \times n$ identity matrix and the multiplication used is ordinary matrix multiplication. If this is the case, then the matrix B is uniquely determined by A and is called the inverse of A, denoted by A^{-1}. It follows from the theory of matrices that if

$$AB = I \tag{14}$$

for square matrices A and B, then also

$$BA = I \tag{15}$$

Non-square matrices ($m \times n$ matrices for which $m \neq n$) do not have an inverse. However, in some cases such a matrix may have a left inverse or right inverse. If A is $m \times n$ and the rank of A is equal to n, then A has a left inverse: an $m \times n$ matrix B such that $BA = I$. If A has rank m, then it has a right inverse: an $n \times m$ matrix B such that $AB = I$.

A square matrix that is not invertible is called singular or degenerate. A square matrix is singular if and only if its determinant is zero. The concept of determinant will be explained in the next few paragraphs.

The determinant of a matrix A is denoted by | A |. For example, for matrix

$$A = \begin{bmatrix} a1 & a2 & a3 \\ a4 & a5 & a6 \\ a7 & a8 & a9 \end{bmatrix}$$

the determinant $det(A)$ might be indicated by | A | or more explicitly as

$$|A| = \begin{vmatrix} a1 & a2 & a3 \\ a4 & a5 & a6 \\ a7 & a8 & a9 \end{vmatrix}$$

The 2×2 matrix, $A = \begin{bmatrix} a1 & a2 \\ a3 & a4 \end{bmatrix}$

has determinant $det(A) = ad - bc$ (16)

For determinant of 3×3 matrix, can be found in [4].

2.1.5 *Identity Matrices*

In linear algebra, the identity matrix or unit matrix of size n is the $n \times n$ square matrix with ones on the main diagonal and zeros elsewhere. It is denoted by I_n, or simply by I if the size is immaterial or can be trivially determined by the context.

$$I_1 = [1]$$

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$I_n = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}$$

The important property of matrix multiplication of identity matrix is that for $m \times n$ matrix A

$$I_m A = A\ I_n = A \tag{17}$$

The ith column of an identity matrix is the unit vector e_j. The unit vectors are also the eigenvectors of the identity matrix, all corresponding to the eigenvalue 1, which is therefore the only eigenvalue and has multiplicity n. It follows that the determinant of the identity matrix is 1 and the trace is n.

Using the notation that is sometimes used to concisely describe diagonal matrices, which can be written as:

$$I_n = diag(1, 1, \ldots, 1) \tag{18}$$

2.1.6 *Linear Transformation*

In linear algebra, linear transformations can be represented by matrices. If T is a linear transformation mapping R^n to R^m, and x is a column vector with n entries, then

$$T(\vec{x}) = A\vec{x} \tag{19}$$

for some $m \times n$ matrix A, called the transformation matrix of T.

When $S_1$and S_2be subsets of R^n and R^m, respectively. A function f from $S_1$and S_2, written $f : S_1 \rightarrow S_2$, is a rule that assigns to each vector v in S_1 a unique vector $f(v)$ in S_2. The vector $f(v)$ is called the image of v (under f). The set $S_1$is called domain of a function f, and the set S_2 is called the codomain of f. The range of f is defined to be the set of images $f(v)$ for all v in S_1 [4]. As shown in Figure 2, both u and v have w as their image.

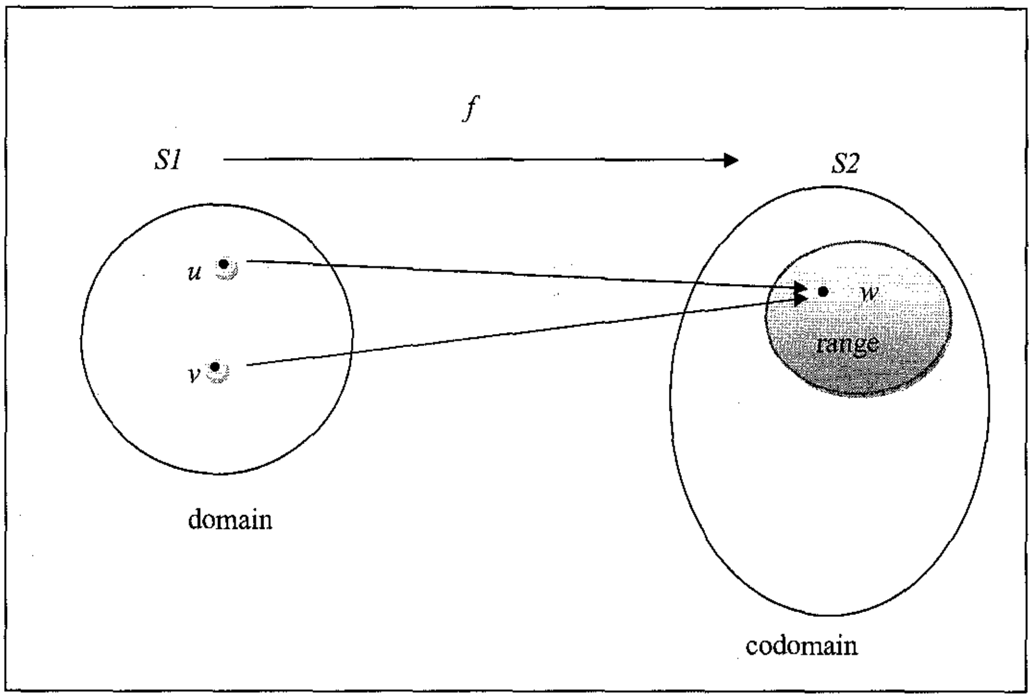


Figure 2: The domain, codomain and range of a function

Besides rotation and projections, some special cased of linear transformations are the geometric transformations, namely, reflections, contractions and dilations.

There are two linear transformations that deserve special attention. The first one is identity transformation $I : R^n \rightarrow R^n$, which is defined by $I(x) = x$ for all x in R^n when I is linear and its range is all of R^n. The second transformation is the zero transformation $T_0 : R^n \rightarrow R^m$, which is defined by $T(x) = 0$ for all x in R^n. Like the identity transformation, T_0 is linear and its range consists precisely of the zero vector. Properties of the linear transformation [4]:

- $T_A\,(\mathbf{u}+\mathbf{v}) \;=\; T_A\,(\mathbf{u}) \;+\; T_A\,(\mathbf{v})$ (20)
- $T_A\,(c\mathbf{u}) \;=\; c\,T_A\,(\mathbf{u})$ for every scalar c (21)
- $T(0) \;=\; 0$ (22)
- $T(-\mathbf{u}) \;=\; -T(\mathbf{u})$for all vectors **u** in R^m (23)
- $T(\mathbf{u}-\mathbf{v}) \;=\; T(\mathbf{u}) \;-\; T(\mathbf{v})$ for all vectors **u** and **v** in R^n (24)
- $T(a\mathbf{u}+b\mathbf{v}) \;=\; aT(\mathbf{u}) \;+\; bT(\mathbf{v})$ for all vectors **u** and **v** in R^n and all scalars a and b (25)

2.2 Vector Space and Subspaces

Many concepts concerning vectors in R^n can be extended to other mathematical systems. A vector space in general, can be considered as a collection of objects that behave as vectors do in R^n. The objects of such a set are called vectors. A vector space is a nonempty set *V* of objects, called vectors, on which are defined two operations, called addition and multiplication by scalars (real numbers).

This material covered from section 2.2 until 2.3 are obtained from [6]. Some of the proofing required for theorems presented here is not being reproduced in this report. For clarification, readers are encouraged to refer to [6] for details on the related theorems and their examples.

A. Subspaces

Vector spaces may be formed from subsets of other vectors spaces. These are called subspaces. A subspace of a vector space *V* is a subset *H* of *V* that has three properties:

- The zero vector of *V* is in *H*
- For each **u** and **v** are in *H*, **u** + **v** is in *H* (In this case *H* is closed under vector addition)
- For each *u* in *H* and each scalar *c*, *c***u** is in *H* (In this case *H* is closed under scalar multiplication)

If the subset *H* satisfies these three properties, then *H* itself is a vector space.

Theorem 1: If $v_1, \ldots, v_p$ are in a vector space *V*, then $Span\{v_1, \ldots, v_p\}$ is a subspace of *V*.

$Span\{v_1, \ldots, v_p\}$ is called the subspace spanned (or generated) by$\{v_1, \ldots, v_p\}$. Given any subspace *H* of *V*, a spanning (or generating) set for *H* is a set $\{v_1, \ldots, v_p\}$in *H* such that $H = Span\,\{v_1, \ldots, v_p\}$.

2.1.7 Null Spaces, Column Spaces, and Linear Transformations

The null space of an $m \times n$ matrix *A*, written as Nul *A*, is the set of all solutions to the homogeneous equation $Ax = 0$. In set notation,

$$Nul\,A = \{x : x\ is\ in\ R^n\ and\ Ax = 0\} \tag{26}$$

A more dynamic description of Nul *A* is the set of all *x* in R^n that are mapped into the zero vector of R^m via a linear transformation $\mapsto Ax$. As illustrated in Figure3:

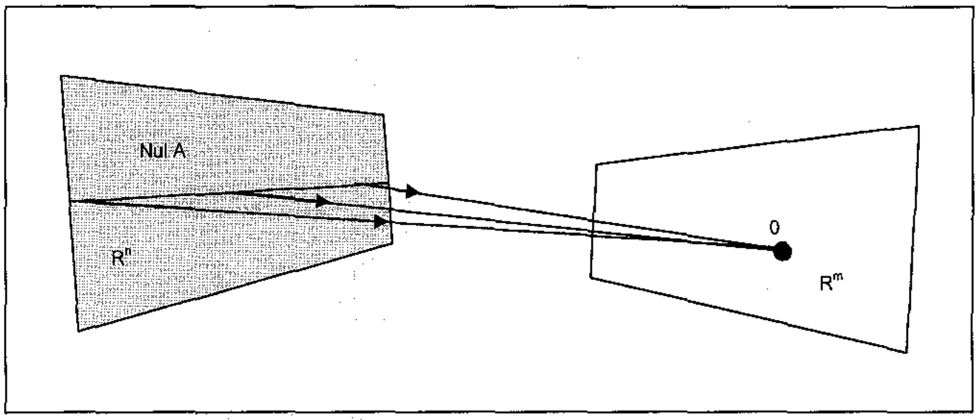


Figure 3: Null space

The term null space is appropriate because the null space of a matrix is a vector space as mentioned in the next theorem.

Theorem 2: The null space of an $m \times n$ matrix *A* is a subspace of R^n. Equivalently, the set of all solutions to a system $Ax = 0$ of *m* homogeneous linear equations in *n* unknowns is a subspace of R^n.

A. *The Column Space of a Matrix*

Another important subspace associated with a matrix is its column space. Unlike the null space, the column space is defined explicitly via linear combination. The column space of an $m \times n$ matrix A ($Col\ A$) is the set of all linear combinations of the columns of A. If $A = [a_1 \ldots a_n]$, then $Col\ A = Span\ \{a_1, \ldots., a_n\}$. Since $Span\ \{a_1, \ldots., a_n\}$is a subspace by Theorem 1, the next theorem follows from the definition of $Col\ A$ and the fact that the columns of A are in R^m.

Theorem 3: The column space of an $m \times n$ matrix A is a subspace of R^m.
Recall that if $Ax = b$, then b is a linear combination of the columns of A. Therefore,

$$Col\ A = \{b : b = Ax \text{ for some } x \text{ in } R^n\} \tag{27}$$

B. *Kernel and Range of a Linear Transformation*

Subspaces of vector spaces other than R^n are often described in terms of a linear transformation instead of a matrix. A linear transformation T from a vector space V into a vector space W is a rule that assigns to each vector **x** in V a unique vector $T(x)$ in W, such that:

i. $T(\mathbf{u} + \mathbf{v}) = T(\mathbf{u}) + T(\mathbf{v})$ for all u,v in V (28)

ii. $T(c\mathbf{u}) = cT(\mathbf{u})$ for all u in tin V and all scalars c (29)

The kernel (or null space) of T is the set of all vectors u in V such that $T(\mathbf{u}) = 0$. The range of T is the set of all vectors in W of the form $T(\mathbf{u})$ where u is in V.

$$\text{So if } T(x) = Ax, Col\ A = \text{Range of } T \tag{30}$$

2.1.8 Linearly Independent Sets; Bases

A set of vectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots., \mathbf{v}_p\}$ in a vector space V is said to be linearly independent if the vector equation

$$c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \ldots. + c_p\mathbf{v}_p = 0 \tag{31}$$

has only the trivial solution $c_1 = 0, \ldots, c_p = 0$.

The set $\{\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_p\}$ is said to be linearly dependent if there exists weights $c_1, \ldots\ldots, c_p$, not all zero, such that (31) holds. In such a case, (31) is called a linear dependence relation among $\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_p$.

Just as in R^n, a set of two vectors is linearly dependent if and only if $\mathbf{v} \neq 0$. Also, a set of two vectors is linearly dependent if and only if one of the vectors is a multiple of the other. And any set containing the zero vector is linearly dependent.

Theorem 4: An indexed set $\{\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_p\}$of two or more vectors, with $\mathbf{v}_1 \neq 0$, is linearly dependent if and only if some vector v_j (with $j > 1$) is a linear combination of the preceding vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{j-1}$.The main difference between linear dependence in R^n and in general vector space is that when the vectors are not n-tuples, the homogeneous equation (30) usually cannot be written as a system of n linear equations. That is, the vectors cannot be made into the columns of a matrix A in order to study the equation $Ax = 0$.

A. A Basis Set

When H is a subspace of a vector space V. An indexed set of vectors $\beta = \{\mathbf{b}_1, \ldots, \mathbf{b}_p\}$ in V is a basis for H if

i. β is a linearly independent set, and

ii. $H = Span\,\{\mathbf{b}_1, \ldots, \mathbf{b}_p\}$.

A basis set is an "efficient" spanning set containing no unnecessary vectors. In Figure 4, the linearly independent sets $\{\mathbf{v}_1, \mathbf{v}_2\}$ and $\{\mathbf{v}_1, \mathbf{v}_3\}$ to both be examples of basis sets or bases (plural for basis) for H, when H is the plane.

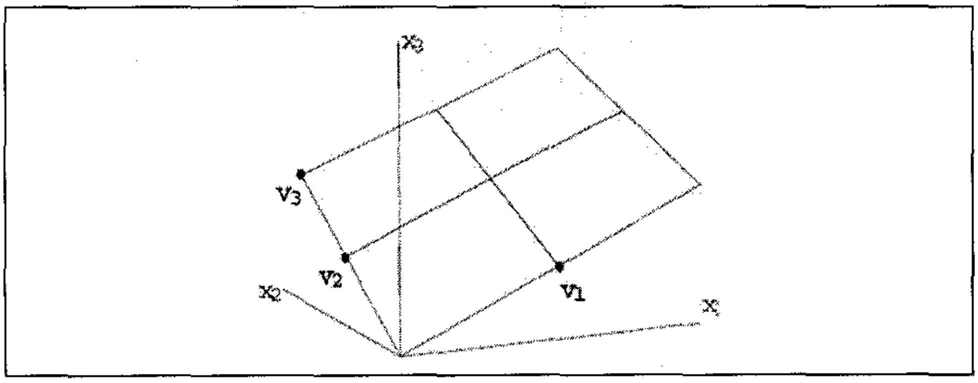


Figure 4: A basic set of plane H

B. The Spanning Set Theorem

A basis can be constructed from a spanning set of vectors by discarding vectors which are linear combinations of preceding vectors in the indexed set.

Theorem 5 (The Spanning Set Theorem): Let $S = \{\mathbf{v}_1, \dots, \mathbf{v}_p\}$ be a set in *V* and let $H = Span\{\mathbf{v}_1, \dots, \mathbf{v}_p\}$.

i. If one of the vectors in *S*, say v_k , is a linear combination of the remaining vectors in *S*, then the set formed from *S* by removing v_k still spans *H*.

ii. If $H \neq \{0\}$, some subset of *S* is a basis for *H*.

2.1.9 The Dimension of a Vector Space

A vector space *V* with a basis β containing *n* vectors is isomorphic to R^n. This section shows that this number *n* is an intrinsic property (called the dimension) of the space *V* that does not depend on the particular choice of basis.

Theorem 6: If a vector space *V* has a basis $\beta = \{\mathbf{b}_1, \dots, \mathbf{b}_p\}$, then any set in *V* containing more than *n* vectors must be linearly dependent.

Suppose $\{\mathbf{u}_1, \dots, \mathbf{u}_p\}$ is a set of vectors in *V* where $p > n$. Then the coordinate vectors $\{[\mathbf{u}_1]_\beta, \dots, [\mathbf{u}_p]_\beta\}$ are in R^n. Since $p > n$, $\{[\mathbf{u}_1]_\beta, \dots, [\mathbf{u}_p]_\beta\}$are linearly dependent and therefore $\{\mathbf{u}_1, \dots, \mathbf{u}_p\}$ are linearly dependent.

Theorem 7: If a vector space *V* has a basis of *n* vectors, then every basis of *V* must consist of *n* vectors.

Suppose β_1 is a basis for *V* consisting of exactly *n* vectors. Now suppose β_2 is any other basis for *V*. By the definition of a basis, we know that β_1 and β_2 are both linearly independent sets.

If a nonzero vector space V is spanned by a finite set S, then the subset of S is a basis for V, by the Spanning Set Theorem. In this case, *Theorem 6* ensures that the following definition makes sense.

If V is spanned by a finite set, then V is said to be finite-dimensional, and the dimension of V, written as dim V, is the number of vectors in a basis for V. The dimension of the zero vector space $\{0\}$ is defined to be 0. If V is not spanned by a finite set, then V is said to be infinite-dimensional.

The next theorem is a natural counterpart to the Spanning Set Theorem.

Theorem 8: Let H be a subspace of a finite-dimensional vector space V. Any linearly independent set in H can be expanded, if necessary, to a basis for H. Also, H is finite-dimensional and

$$dim\, H \; < \; dim\, V$$

When the dimension of a vector space or subspace is known, the search for a basis is simplified by the next theorem. It says that if a set has the right number of elements, then one has only to show either that the set is linearly independent or that it spans the space. The theorem is of critical importance in numerous applied problems where linear independence is much easier to verify than spanning.

Theorem 9 (The Basis Theorem): Let V be a p - dimensional vector space, $p \geq 1$. Any linearly independent set of exactly p vectors in V is automatically a basis for V. Any set of exactly p vectors that spans V is automatically a basis for V.

2.1.10 Rank

With the aid of vector space concepts, this section takes a look inside a matrix and reveals several interesting and useful relationship hidden in its rows and columns. For instance, imagine placing 2000 random numbers into a 40×50 matrix A and then determining both the maximum number of linearly independent columns in A and the maximum number of linearly independent columns in A^T (rows in A). Remarkably, the two numbers are the same. As we'll soon see, their common value is the rank of the matrix. To explain why, we need to examine the subspace spanned by the rows of A.

The set of all linear combinations of the row vectors of a matrix A is called the row space of A and is denoted by Row A. For example:

$$A = \begin{bmatrix} -1 & 2 & 3 & 6 \\ 2 & -5 & -6 & -12 \\ 1 & -3 & -3 & -6 \end{bmatrix} \quad \text{and} \quad \mathbf{r}_1 = (1, 2, 3, 6),$$

$$\mathbf{r}_2 = (2, -5, -6, -12),$$

$$\mathbf{r}_3 = (1, -3, -3, -6)$$

$Row\ A\ =\ Span\ \{r_1, r_2, r_3\}$ (a subspace of R^4)

While it is natural to express row vectors horizontally, they can also be written as column vectors if it is more convenient. Therefore $Col\ A^T\ =\ Row\ A.$ When the row operation is used to reduce matrix A to matrix B, linear combinations of the rows of A to come up with B should be taken. This process can be reduced and used row operations on B to get back to A. Because of this, the row space of A equals the row space of B.

Theorem 10: If two matrices A and B are row equivalent, then their row spaces are the same. If B is in echelon form, the nonzero rows of B form a basis for the row space of A as well as B.

The rank of A is the dimension of the column space of A.

$$rank\ A\ =\ dim\ Col\ A\ = \text{number of pivot columns of } A\ =\ dim\ Row\ A$$

$$rank\ A\ +\ dim\ Nul\ A\ =\ n \qquad (32)$$

Theorem 11(The Rank Theorem): The dimensions of the column space and the row space of an $m \times n$ matrix A are equal. This common dimension, the rank of A, also equals the number of pivot positions in A and satisfies the equation rank A + dim Nul $A\ =\ n$.

Theorem 12 (The Invertible Matrix Theorem): When A is a square $n \times n$ matrix. Then the following statements are equivalent to the statement that A is an invertible matrix:

- The columns of A form a basis for R^n

- $Col\,A = R^n$
- $dim\,Col\,A = n$
- $rank\,A = n$
- $Nul\,A = \{0\}$
- $dim\,Nul\,A = 0$

2.2 Eigenvectors and Eigenvalues

The basic concepts presented here on eigenvectors and eigenvalues are useful throughout pure and applied mathematics. Eigenvalues are also used to study difference equations and continuous dynamical systems. They provide critical information in engineering design, and they arise naturally in such fields as physics and chemistry. For example:

$$A = \begin{bmatrix} 0 & -2 \\ -4 & 2 \end{bmatrix}, \qquad \mathbf{u} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \text{and} \qquad \mathbf{v} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

the images of u and v under multiplication by A:

$$A\mathbf{u} = \begin{bmatrix} 0 & -2 \\ -4 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -2 \\ -2 \end{bmatrix} = -2 \begin{bmatrix} 1 \\ 1 \end{bmatrix} = -2\mathbf{u}$$

$$A\mathbf{v} = \begin{bmatrix} 0 & -2 \\ -4 & 2 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -2 \\ 6 \end{bmatrix} \neq \lambda\mathbf{v}$$

u is called an eigenvector of A but **v** is not an eigenvector of A, because A**v** is not a multiple of **v.**

An eigenvector of an $n \times n$ matrix A is a nonzero vector **x** such that $Ax = \lambda x$ for some scalar λ. A scalar λ is called an eigenvalue of A if there is a nontrivial solution **x** of $= \lambda x$; such an x is called an eigenvector corresponding to λ. The set of all solutions to$(A - \lambda I)x = 0$ is called the eigenspace of A corresponding to λ.

2.2.1 Inner Product, Length & Orthogonality

Geometric concepts of length, distance, and perpendicularity, which are well known for R^2 and $R^3$are defined here for R^n. All the three notions are defined in terms of the inner product of two vectors. These concepts provide powerful geometric tools for solving many applied problems.

A. *The Inner Product*

The inner product or dot product of $\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}$ and $\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$:

$$\mathbf{u} \cdot \mathbf{v} = \mathbf{u}^{\mathrm{T}} \cdot \mathbf{v} = \begin{bmatrix} u_1 & u_2 & \cdots & u_n \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = u_1 v_1 + u_2 v_2 + \cdots + u_n v_n \quad (33)$$

Theorem 1: Let **u**, **v** and **w** be vectors in R^n, and let c be any scalar. Then

- $\mathbf{u} \cdot \mathbf{v} = \mathbf{v} \cdot \mathbf{u}$ (34)
- $(\mathbf{u} + \mathbf{v}) \cdot \mathbf{w} = \mathbf{u} \cdot \mathbf{w} + \mathbf{v} \cdot \mathbf{w}$ (35)
- $(c\mathbf{u}) \cdot \mathbf{v} = c(\mathbf{u} \cdot \mathbf{v}) = \mathbf{u} \cdot (c\mathbf{v})$ (36)
- $\mathbf{u} \cdot \mathbf{u} \geq 0$, and $\mathbf{u} \cdot \mathbf{u} = 0$ if and only if $\mathbf{u} = 0$ (37)

Combining equations (35) and (36), one can show

$$\left(c_1 u_1 + \cdots + c_p u_p\right) \cdot \mathbf{w} = c_1 (u_1 \cdot \mathbf{w}) + \cdots + c_p (u_p \cdot \mathbf{w}) \quad (38)$$

B. *Length of a Vector*

For $\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$, the length or norm of **v** is the nonnegative scalar $\|\mathbf{v}\|$ defined by

$$\|\mathbf{v}\| = \sqrt{\mathbf{v} \cdot \mathbf{v}} = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2} \text{ and } \|\mathbf{v}\|^2 = \mathbf{v} \cdot \mathbf{v} \quad (39)$$

C. *Distance in R^n*

The distance between **u** and **v** in R^n: $\quad dist\,(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\|$

This agrees with the usual formulas for R^2 and R^3. Let

$$\mathbf{u} = (u_1 + u_2) \quad \text{and} \quad \mathbf{v} = (v_1 + v_2)$$

Then

$$\mathbf{u} - \mathbf{v} = (u_1 - v_1\,, u_2 - v_2)$$

and

$$dist\,(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\| = \|(u_1 - v_1\,, u_2 - v_2)\| = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2} \quad (40)$$

D. Orthogonal Vectors

The concept of perpendicular lines in ordinary Euclidean geometry can also be extended in R^n. Consider R^2 or R^3 and two lines through the origin determined by vector u and v. The two lines shown below are geometrically perpendicular if and only if the distance from **u** and **v** is the same as the distance from **u** to -**v**. This is the same as requiring the squares of the distance to be the same. Now

$$\begin{aligned}[dist\,(\mathbf{u},\mathbf{v})]^2 &= \|\mathbf{u}-(-\mathbf{v})\|^2 = \|\mathbf{u}+\mathbf{v}\|^2 \\ &= (\mathbf{u}+\mathbf{v}).(\mathbf{u}+\mathbf{v}) \\ &= \mathbf{u}.(\mathbf{u}+\mathbf{v}) + \mathbf{v}.(\mathbf{u}+\mathbf{v}) \\ &= \mathbf{u}.\mathbf{u}+\mathbf{u}.\mathbf{v}+\mathbf{v}.\mathbf{u}+\mathbf{v}.\mathbf{v} \\ &= \|\mathbf{u}\|^2+\|\mathbf{v}\|^2+2.\mathbf{u}.\mathbf{v} \end{aligned} \tag{41}$$

Similarly,

$$[dist\,(\mathbf{u},-\mathbf{v})]^2 = \|\mathbf{u}\|^2+\|\mathbf{v}\|^2+2.\mathbf{u}.\mathbf{v} \tag{42}$$

Since $[dist\,(\mathbf{u},-\mathbf{v})]^2 = [dist\,(\mathbf{u},\mathbf{v})]^2$, $\mathbf{u}.\mathbf{v}=0$. Two vectors **u** and **v** are said to be orthogonal (to each other) if $\mathbf{u}.\mathbf{v}=0$. Also note that if **u** and **v** are orthogonal, then $\|\mathbf{u}+\mathbf{v}\|^2 = \|\mathbf{u}\|^2+\|\mathbf{v}\|^2$.

Theorem 2 (The Pythagorean Theorem): Two vectors **u** and **v** are orthogonal if and only if $\|\mathbf{u}+\mathbf{v}\|^2 = \|\mathbf{u}\|^2+\|\mathbf{v}\|^2$.

E. Orthogonal Complements

If a vector **z** is orthogonal to every vector in a subspace W of R^n, then **z** is said to be orthogonal to W (Figure 5). The set of vectors z that are orthogonal to W is called the orthogonal complement of W and is denoted by $W^{\perp}$ (read as "W perp").

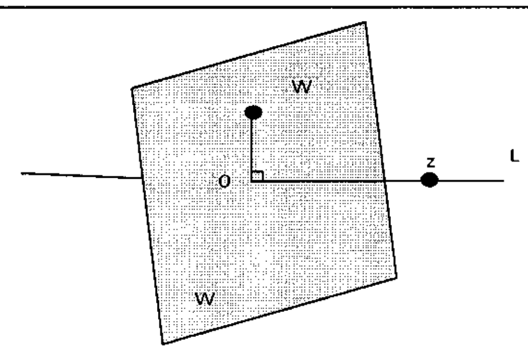


Figure 5: A plane and line through 0 as orthogonal complements

F. Row, Null and Columns Spaces

Theorem 3: Let A be an $m \times n$ matrix. Then the orthogonal complement of the row space of A is the nullspace of A, and the orthogonal complement of the column space is the nullspace of A^T (Figure 6):

$$(Row\ A)^{\perp} = Nul\ A, \quad (Col\ A)^{\perp} = Nul\ A^T \tag{43}$$

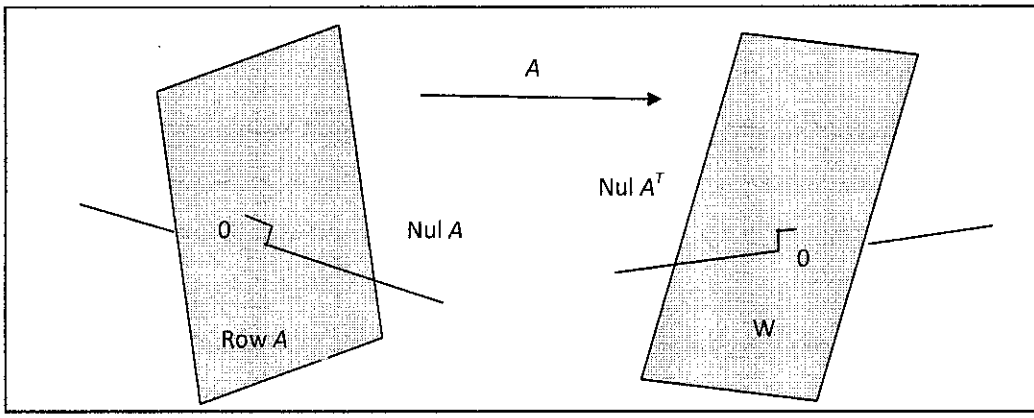


Figure 6: The fundamental subspace determined by an $m \times n$ matrix A

2.2.2 Orthogonal Sets

A set of vectors $\{u_1, u_2, \dots, u_p\}$ in R^n is called an orthogonal set if $u_i \cdot u_j \neq 0$ whenever $i \neq j$. For example, for a set of vector $\{u_1, u_2, u_3\}$, where

$$\mathbf{u_1} = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}, \mathbf{u_2} = \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix} \text{ and } \mathbf{u_3} = \begin{bmatrix} -1/2 \\ -2 \\ 7/2 \end{bmatrix}$$

Then $\mathbf{u_1}.\mathbf{u_2} = 0, \mathbf{u_1}.\mathbf{u_3} = 0,$ and $\mathbf{u_2}.\mathbf{u_3} = 0.$ So each pair of distinct vectors is orthogonal as shown in Figure 7, and so $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ is an orthogonal set.

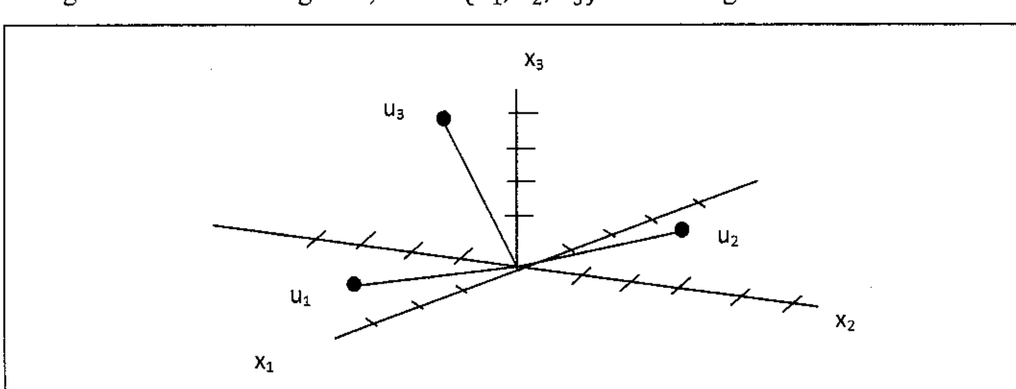


Figure 7: Orthogonal sets

Theorem 4: Suppose S $= \{\mathbf{u_1}, \mathbf{u_2}, \dots, \mathbf{u_p}\}$ is an orthogonal set of nonzero vectors in $R^n$and W, span $= \{\mathbf{u_1}, \mathbf{u_2}, \dots, \mathbf{u_p}\}$. Then *S* is a linearly independent set and is therefore a basis for *W*.

If $0 = c_1\mathbf{u_1} + c_2\mathbf{u_2} + \dots + c_p\mathbf{u_p}$ for some scalars $c_1, \dots, c_p$, then

$$\begin{aligned} 0 = 0.\mathbf{u_1} &= (c_1\mathbf{u_1} + c_2\mathbf{u_2} + \dots + c_p\mathbf{u_p}).\mathbf{u_1} \\ &= (c_1\mathbf{u_1}).\mathbf{u_1} + (c_2\mathbf{u_2}).\mathbf{u_1} + \dots + (c_p\mathbf{u_p}).\mathbf{u_1} \\ &= c_1(\mathbf{u_1}.\mathbf{u_1}) + c_2(\mathbf{u_2}.\mathbf{u_1}) + \dots + c_p(\mathbf{u_p}.\mathbf{u_1}) \\ &= c_1(\mathbf{u_1}.\mathbf{u_1}) \end{aligned} \tag{44}$$

Since $\mathbf{u_1} \neq 0$, $\mathbf{u_1}.\mathbf{u_1} > 0$ which means c_1=0. In a similar manner, $c_1, \dots, c_p$ can be shown to by all 0. So S is a linearly independent set.

An orthogonal basis for a subspace W of R^n is a basis for W that is also an orthogonal set. The next theorem suggests why an orthogonal basis is much nicer than other bases.

Theorem 5: Let $\{u_1, u_2, \dots, u_p\}$ be an orthogonal basis for a subspace W of R^n Then each y in W has a unique representation as a linear combination of $u_1, u_2, \dots, u_p$. In fact, if

$$y = c_1 u_1 + c_2 u_2 + \dots + c_p u_p \tag{45}$$

then

$$c_j = \frac{y.u_j}{u_j.u_j}, \qquad j = 1, \dots, p \tag{46}$$

A. Orthogonal Projections

For a nonzero vector **u** in R^n, when y wanted to be written in R^nas the the following

$$y = (multiple\ of\ u) + (multiple\ a\ vector \perp to\ u) = \hat{y} + z$$

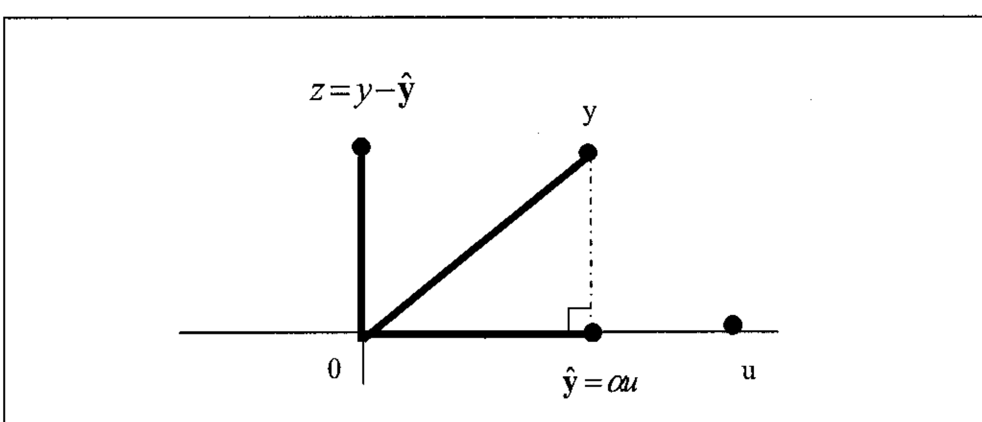


Figure 8: Finding α to make $y - \hat{y}$ orthogonal to u

The vector $\hat{y}$ is called the orthogonal projection of y onto u and the vector z is called the component of y orthogonal to u as illustrated in Figure 8. The vector $\hat{y}$ can be calculated using

$$\hat{y} = \frac{y.u}{u.u} u \tag{47}$$

B. Orthonormal Sets

A set of vectors $\{\mathbf{u_1}, \mathbf{u_2}, \dots, \mathbf{u_p}\}$ in R^n is called an orthonormal set if it is an orthogonal set of unit vectors.

If $W = span\{\mathbf{u_1}, \mathbf{u_2}, \dots, \mathbf{u_p}\}$, then $\{\mathbf{u_1}, \mathbf{u_2}, \dots, \mathbf{u_p}\}$ is an orthonormal basis for W.

Theorem 6: An $m \times n$ matrix U has orthonormal columns if and only if $U^T U = I$.

Theorem 7: Let U be an $m \times n$ matrix with orthonormal columns, and let $\mathbf{x}$ and $\mathbf{y}$ be in R^n. Then

a. $\|U_\mathbf{x}\| = \|\mathbf{x}\|$ (48)

b. $(U_\mathbf{x}) \cdot (U_\mathbf{y}) = \mathbf{x} \,.\, \mathbf{y}$ (49)

c. $(U_\mathbf{x}) \cdot (U_\mathbf{y}) = 0$ if and only if $\mathbf{x} \,.\, \mathbf{y} = 0$. (50)

Properties (a) and (c) say that the linear mapping $\mathbf{x} \longmapsto U_\mathbf{x}$ preserves lengths and orthogonality. These properties are crucial for many computer algorithms.

2.2.3 The Singular Value Decomposition

The singular-value decomposition (SVD) of a matrix is one of the most elegant algorithms in numerical algebra for providing qualitative information about the structure of linear equations [7]. In image processing applications SVD provides a robust method of storing large images into a smaller and more manageable size. This is accomplished by reproducing the original image with each succeeding nonzero singular value. Furthermore, to reduce storage size even further, one may approximate a "good enough" image by using even fewer singular values.

The singular value decomposition of an $m \times n$ matrix A is given by

$$A = U\Sigma V^T \tag{51}$$

where U is an $m \times m$ orthogonal matrix; V an $n \times n$ orthogonal matrix, and Σ an $m \times n$ matrix containing the singular values of $A, \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$ along its main diagonal.

A similar technique, known as the eigenvalue decomposition, also diagonalizes matrix A, but with this case, A must be a square matrix. The EVD diagonalizes A as

$$A = UDV^{-1} \tag{52}$$

where D is a diagonal matrix comprised of the eigenvalues, and V is a matrix whose columns contain the corresponding eigenvectors. However, the EVD can only be applied to square matrixes so for non-square matrixes, we will have to use the SVD.

The next paragraphs will show why the SVD work on $m \times n$ matrix A.

Let assume that the SVD of A is always possible, unlike that of the EVD. The matrix $A^T A$ is symmetric and can be diagonalized. Working with the symmetric matrix $A^T A$, then two conditions must be true;

1. The eigenvalues of $A^T A$ will be real and nonnegative.

2. The eigenvectors will be orthogonal.

Now, for finding the orthogonal matrices U and V that diagonalize a $m \times n$ matrix A. First, if the intent is to factor A as $A = U\Sigma V^T$ then the following must be true.

$$\begin{aligned} A^T A &= (U\Sigma V^T)^T (U\Sigma V^T) \\ &= V\Sigma^T U^T U\Sigma V^T \\ &= V\Sigma^T \Sigma V^T \\ &= V\Sigma^2 V^T \end{aligned} \tag{53}$$

This implies that Σ^2 contains the eigenvalues of $A^T A$ and V contains the corresponding eigenvectors.

Next, is rearranging the eigenvalues of $A^T A$ in order of decreasing magnitude noting that some eigenvalues are equal to zero:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r \geq \lambda_{r+1} = \lambda_{r+2} = \cdots = \lambda_n = 0$$

Then the singular values of A is defined as the square root of the corresponding eigenvalues of the matrix $A^T A$; that is,

$$\sigma_j = \sqrt{\lambda_j}\text{ ,where } j = 1, 2, \ldots, n. \tag{54}$$

After that, the eigenvectors of $A^T A$ are rearranged in the same order as their respective eigenvalues to produce the matrix:

$$V = [v_1, v_2, \ldots, v_r, v_{r+1}, v_{r+2}, \ldots, v_n]$$

When $V_1 = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r]$ is the set of eigenvectors associated with non-zero eigenvalues and $V_2 = [\mathbf{v}_r, \mathbf{v}_{r+1}, \mathbf{v}_{r+2}, \ldots, \mathbf{v}_n]$ is the set of eigenvectors associated with zero eigenvalues. It follows that

$$AV_2 = [A\mathbf{v}_r, A\mathbf{v}_{r+1}, A\mathbf{v}_{r+2}, \ldots, A\mathbf{v}_n] = 0 \tag{55}$$

To find the matrix Σ, let denote Σ_1 as a square $r \times r$ matrix containing the nonzero singular values $\{\sigma_1, \sigma_2, \ldots, \sigma_r\}$ of A along its main diagonal. Therefore matrix Σ may be represented by

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \tag{56}$$

where the singular values along the diagonal are arranged in decreasing magnitude, and the zero singular values are placed at the end of the diagonal. This new matrix Σ, with the correct dimension $m \times n$, is padded with $(m - r)$ rows and $(n - r)$ columns of zeros.

To obtain the orthogonal matrix U, we re-write equation (51) as $AV = U\Sigma$. Expanding the left and the right side of this equation gives

$$A[v_1, \cdots, v_r, v_{r+1}, \cdots, v_n] = [u_1, \cdots, u_r, u_{r+1}, \cdots, u_n] \begin{bmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_r & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{bmatrix}$$

$$[Av_1, \cdots, Av_r, Av_{r+1}, \cdots, Av_n] = [\sigma_1 u_1, \cdots, \sigma_r u_r, 0, \cdots, 0]$$

Therefore,

$$A\mathbf{v}_j = \sigma_j \mathbf{u}_j\text{ , where } j = 1, 2, \ldots, n. \tag{57}$$

In equation (54), σ_j is a scalar, $\mathbf{v}_j$, and $\mathbf{u}_j$ are column vectors and matrix-vector multiplication results in another vector. Therefore, the vector resulting from the multiplication of Av_j is equal to the vector u_j multiplied by the scalar σ_j. In other word, Av_j is a vector lying in the direction of the unit vector u_j with absolute length σ_j. Vector Av_j can be calculated from previously found matrix V. Therefore, the unit vector u_j, is a result of dividing the vector Av_j by its magnitude, σ_j.

$$u_j = \frac{Av_j}{\sigma_j} \tag{58}$$

Equation (54) is restricted to the first r nonzero singular values. The explanation for the zero singular values will be discussed later. This method will determine only part of matrix U. To find the other part where the singular values of A are equal to zero, let defined the matrix U as

$$U = [U_1 \quad U_2] \tag{59}$$

Let

$$U_1 = [u_1, \cdots, u_r] \text{and} U_2 = [u_{r+1}, \cdots, u_m].$$

Then

$$\begin{aligned} AV_1 &= A[v_1, \cdots, v_r] \\ &= [Av_1, \cdots, Av_r] \\ &= [\sigma_1 u_1, \cdots, \sigma_r u_r] \quad \text{from equation (57)} \\ &= [u_1, \cdots, u_r] \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix} \end{aligned}$$

$$\text{i.e } AV_1 = U_1 \Sigma_1 \tag{60}$$

Before proceeding to find the matrix U_2, let consider four fundamental subspaces as illustrated in Figure 9. The null space of matrix A, $N(A)$ denotes the set of all nontrivial (non-zero) solutions to equation $Ax = 0$. Using equation (54) $AV_2 = 0$, it follows that V_2 forms a basis for the $N(A)$. Also because

$$\begin{aligned} A\mathbf{v}_j &= 0\mathbf{u}_j \text{ where } j = r+1, r+2, \cdots, n \\ A\mathbf{v}_j &= 0 \\ \mathbf{v}_j &\in N(A) \end{aligned} \tag{61}$$

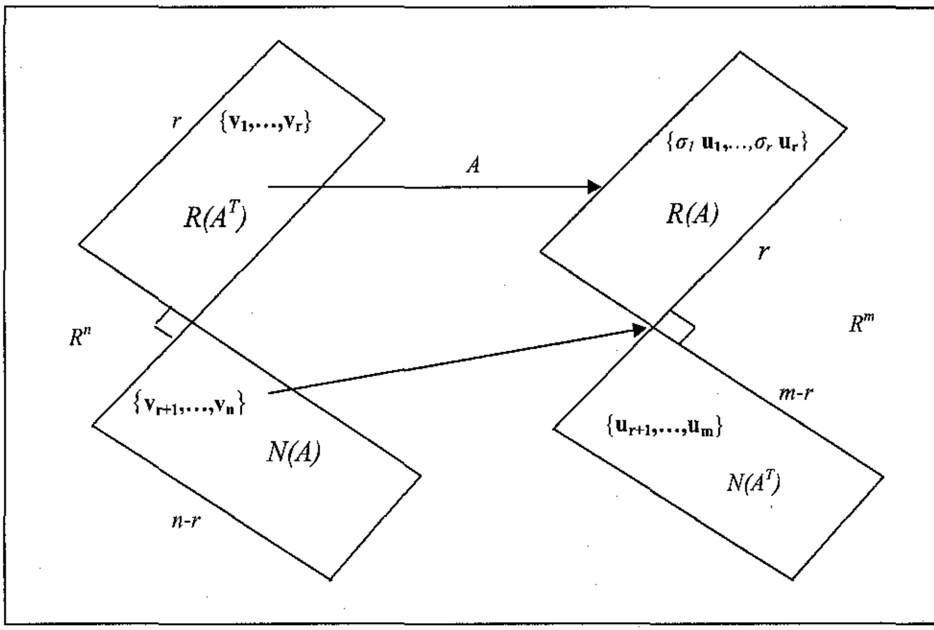


Figure 9: The four fundamental subspaces

As mentioned earlier the orthogonal complement to the $N(A)$ is the $R(A^T)$. Since the columns in the matrix V are orthogonal, the remaining vectors $v_1,\ldots,v_r$ must lie in the subspace corresponding to the $R(A^T)$. From equation (58), $u_j = \frac{A}{\sigma_j} v_j$, this equation holds the valuable information that the column vectors of U, $[u_1,\cdots,u_r]$ are in the columnspace of A. This is because the column vectors of U are linear combinations of the columns of A or, in matrix notation

$$u_j \in R(A) \text{ where } j = r+1,\ r+2,\cdots,n \tag{62}$$

It now follows that $R(A)$ and $N(A^T)$ are orthogonal complements. Since the matrix U is an orthogonal matrix and the first r column vectors of U have been assigned to lie in the $R(A)$, then $[u_{r+1},\cdots,u_m]$ must lie in the $N(A^T)$ which form the matrix U_2. Now the matrix V, the matrix Σ, and the matrix U, the singular value decomposition has been found for any matrix A.

2.4 Biometric Recognition Systems

This section explains some general principles of biometric recognition systems, describes different classification errors and explains how the quality of two systems can be compared objectively.

2.4.1 Identification versus Verification

The process of trying to find out a person's identity by examining a biometric pattern calculated from the person's biometric features is called identification. Identification or verification are two different modes for biometric recognition system. The material in this chapter is obtained from [1].

The real difference between Identification or verification is that, for identification case, the system is trained with the patterns of several persons. A biometric template is calculated for each of the persons, in this training stage. The identified pattern should be matched against every known template and be checked for similarity with the template. The system assigns the pattern to the person with the most similar biometric template. In this case a threshold value should be introduced to the system to prevent the error. If this level is not reached, the pattern is rejected but in verification case, the pattern is only compared with the person's individual template. It is checked whether the similarity between pattern and template is sufficient to provide access to the secured system or area.

2.4.2 Thresholding (False Acceptance / False Rejection)

In any bioidentification system, the level of similarity between patterns can be expressed in scores. Access to the system is granted only, if the score for a trained person (identification) or the person that the pattern is verified against (verification) is higher than a certain threshold. The sores of patterns from persons known by the system should be always higher than the impostors' scores, but in some systems it is not true in all cases. It is because however the classification threshold is chosen, some classification errors occur. These kinds of errors or mistakes can cause false acceptance or false rejection.

For example as shown in the following figure, the belonging scores would be somehow distributed around a certain mean score. This is depicted in the first image on the left side. A gaussian normal distribution is chosen in this example. Depending on the choice of the classification threshold, between all and none of the impostor patterns are falsely accepted by the system. The threshold depending fraction of the falsely accepted patterns divided by the number of all impostor patterns is called False Acceptance Rate (FAR).

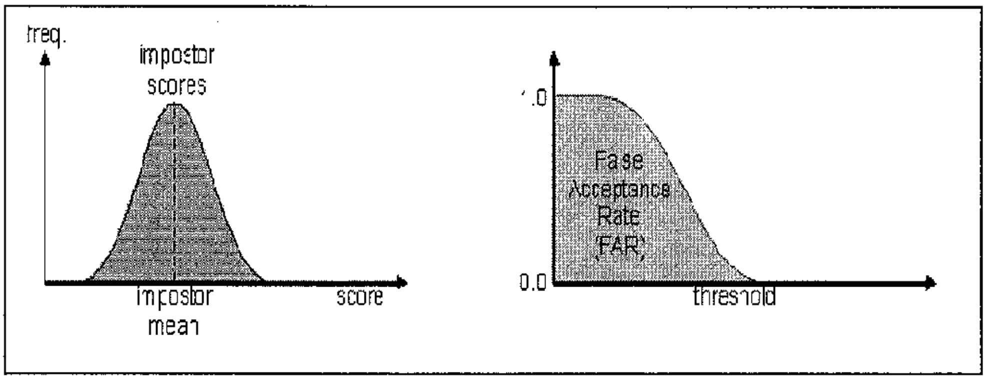


Figure 10: FAR value for varying threshold and score distribution[1]

Similar to the impostor scores, the client pattern's scores can vary around a certain mean value. The mean score of the client patterns is higher than the mean value of the impostor patterns, as shown in the left of the following two images. If a classification threshold that is too high is applied to the classification scores, some of the client patterns are falsely rejected. Depending on the value of the threshold, between none and all of the client patterns will be falsely rejected. The fraction of the number of rejected client patterns divided by the total number of client patterns is called False Rejection Rate (FRR). According to the FAR, its value lies in between zero and one. The image on the right shows the FAR for a varying threshold for the score distribution shown in the image on the left.

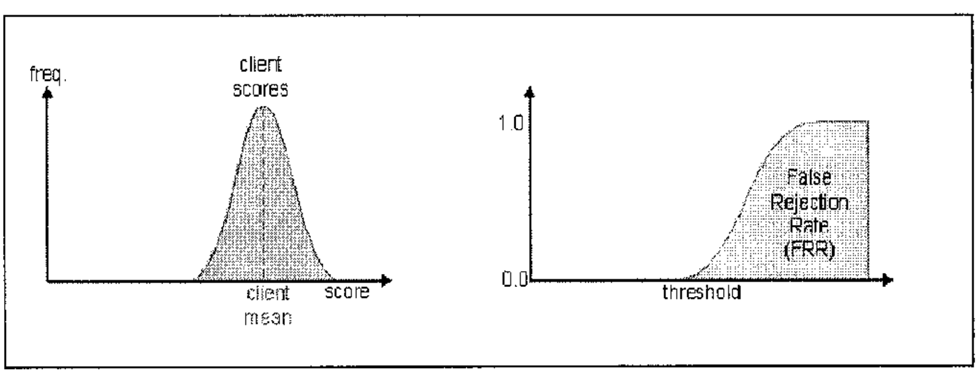


Figure 11: FRR value for varying threshold and score distribution [1]

The choice of the threshold value becomes a problem if the distributions of the client and the impostor scores overlap, as shown in the next image on the left. The value of the FAR and the FRR at overlap point is of course the same for both of them, is called the Equal Error Rate (EER). The lower the EER is, the better is the system's performance, as the total error rate which is the sum of the FAR and the FRR at the point of the EER decreases. In theory this works fine, if the EER of the system is calculated using an infinite and representative test set, which of course is not possible under real world conditions. To get comparable results it is therefore necessary that the EERs that are compared are calculated on the same test data using the same test protocol.

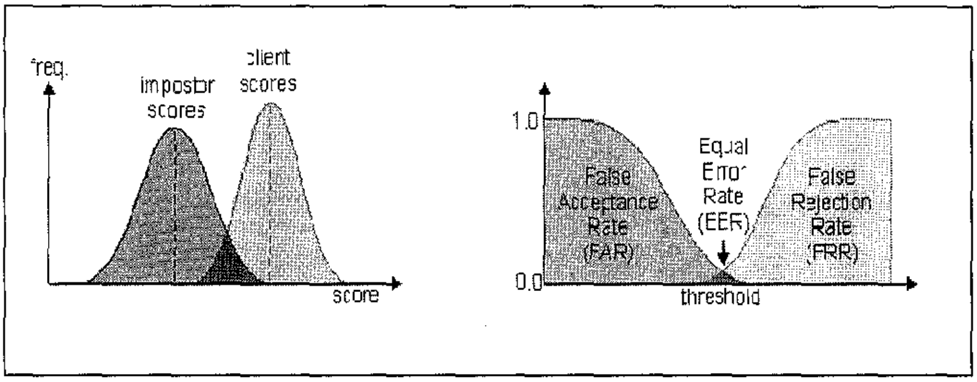


Figure 12: EER value for varying threshold and score distribution [1]

CHAPTER 3
METHODOLOGY

This chapter presents methodology of online signature verification technique. An overview of the signature verification system is described in section 3.1. Next, in section 3.2, the motivation of selecting SVD for signature verification is being highlighted. In section 3.3, the link between the SVD and the concept of oriented energy distribution is being explained. Then, in section 3.4, a method to determine the r-most prominent sensors (out of 14 sensors) is presented. Finally, the procedure of the experiment is covered in section 3.4

3.1 Signature Verification System

Online signature verification system involves 4 main steps; 1) data acquisition and preprocessing (input device), 2) feature extraction, 3) matching (classification), and 4) decision making. Figure 11 shows the general on-line signature verification process. Here, step 2-4 is implemented on MATLAB software. The subsequent paragraphs explain the details on each step.

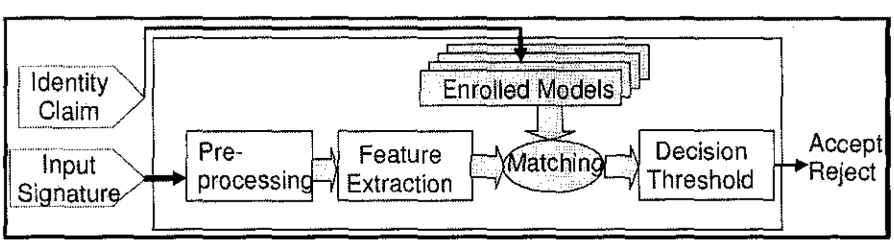


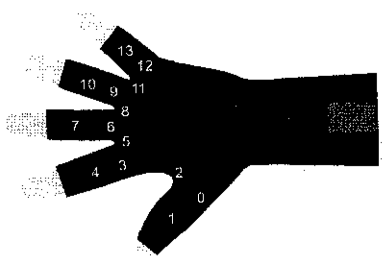
Figure 13: Online signature verification system [2]

Input device:

The input device used is data glove with 14 sensors. The data glove is a new measurement in the field of virtual reality environments, initially designed to satisfy the stringent requirements of modern motion capture and animation professionals. It provides ease of use, a small form factor, and multiple application drivers. The dynamic features of the data glove provide information on:

- Patterns distinctive to an individuals' signature and hand size
- Time elapsed during the signature process
- Hand trajectory dependent rolling

Thus, while most input devices offer few degrees of freedom, the data glove is unique in offering multiple degrees of freedom in that it provides data on both the dynamics of the pen motion during the signature and the individual's hand shape. Figure 12 and Table 1; show the 5DT Data Glove 14 Ultra with the location of the sensors. The description of data glove is given in Appendix A.



Sensor	Description
0	Thumb Flexure (lower joint) or (Thumb Near)
1	Thumb Flexure (second joint) or (Thumb Far)
2	Thumb-index finger abduction or (Thumb/Index)
3	Index finger flexure (at knuckle) or (Index Near)
4	Index finger flexure (second joint) or (Index Far)
5	Index-middle finger abduction or (Index/Middle)
6	Middle finger flexure (at knuckle) or (Middle Near)
7	Middle finger flexure (second joint) or (Middle Far)
8	Middle-ring finger abduction or (Middle/Ring)
9	Ring finger flexure (at knuckle) or (Ring Near)
10	Ring finger flexure (second joint) or (Ring Far)
11	Ring-little finger abduction or (Ring/Little)
12	Little finger flexure (at knuckle) or (Little Near)
13	Little finger flexure (second joint) or (Little Far)

Figure 14: Sensor mapping for 5DT Data Glove Ultra 14 [2]

Feature extraction:

The features extracted from the data glove are considered as dynamic features as explained in the last paragraph.

Matching:

Matching technique is use to measure the similarity between the claimed identity model and the input features. The matching technique used for this project is by calculating the average distance between tried signature principal subspace and the authentic one. This is known as structural matching.

Decision:

Once a similarity measure is obtained, the decision implies the computation of a decision threshold. If the similarity is grater than a threshold, the decision is ACCEPT, otherwise it is REJECT.

3.1.1 Tools and Equipment required

A. MATLAB Software

MATLAB is a numerical computing environment and programming language. Created by The MathWorks, MATLAB allows easy matrix manipulation, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs in other languages. Although it is numeric only, an optional toolbox interfaces with the Maple symbolic engine, allowing access to computer algebra capabilities. The MATLAB codes used for this project are given in Appendix B.

B. 5DT 14 Ultra Data Glove

The glove used in this project is a wired glove use for virtual reality environments. Various sensor technologies are used to capture physical data such as bending of fingers. Often a motion tracker, such as a magnetic tracking device or inertial tracking device, is attached to capture the global position / rotation of the glove. These movements are then interpreted by the software that accompanies the glove, so any movement can be represented as numerical data. Gestures can then be categorized into useful information, such as to recognize sign language or other symbolic functions. The full description of data glove is given in Appendix A.

More generally, the energy E_Q measured in a subspace $Q \subset \boldsymbol{R}^m$ is defined as

$$E_Q[A] = \sum_{k=1}^{n} \|P_Q(\mathbf{a}_k)\|^2. \tag{65}$$

where $P_Q = (\boldsymbol{a}_k)$ denotes the orthogonal projection of $\{\boldsymbol{a}_k\}$ into the subspace Q and $\|.\|$ denotes the Euclidean norm. In other words, the oriented energy of a vector sequence $\{\boldsymbol{a}_k\}$, measured in the direction $\boldsymbol{q}$ (subspace Q) is the energy of the signal, projected orthogonally on to the vector $\boldsymbol{q}$(subspace Q).

3.2.2 The Singular Value Decomposition (SVD)

The SVD for real matrices is based upon the following theorem [4],[5]:

Theorem 1. For any real $m \times n$ matrix $\boldsymbol{A}$, there exists a real factorization

$$\boldsymbol{A} = \boldsymbol{U}_{m\times m} \cdot \boldsymbol{S}_{m\times n} \cdot \boldsymbol{V}_{n\times n}^{T} \tag{66}$$

in which the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are real orthonormal, and matrix $\boldsymbol{S}$ is real pseudo-diagonal with nonnegative diagonal elements.

The diagonal entries σ_i of $\boldsymbol{S}$ are called the singular values of the matrix $\boldsymbol{A}$. It is assumed that they are sorted in non-increasing order of magnitude. The set of singular values $\{\sigma_i\}$ is called the singular spectrum of matrix $\boldsymbol{A}$. The columns $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ of $\boldsymbol{U}$ and $\boldsymbol{V}$ are called the left and right singular vectors of matrix $\boldsymbol{A}$ respectively. The space $S_U^r = span[\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_r]$ is called the r-th left principal subspace. In a similar way, the r-th right singular subspace is defined. Proofs of the above classical existence and uniqueness theorems are found in [13].

Lemma 1. *The number of non zero singular values, equals the algebraic rank of the matrix A.*

Lemma 2. *Via the SVD, any matrix A can be written as the sum of three rank-one matrices*

$$\boldsymbol{A} = \sum_{i=1}^{r} (\boldsymbol{u}_i . \sigma_i . \boldsymbol{v}_i^T) \tag{67}$$

where $(\boldsymbol{u}_i, \sigma_i, \boldsymbol{v}_i)$ is the i-th singular triplet of matrix $\boldsymbol{A}$.

Lemma 3. *Frobenius norm of $m \times n$ matrix A of rank r*

$$||A||_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2 = \sum_{k=1}^{r} \sigma_k^2 \tag{68}$$

where σ_k are the singular values of A.

The total energy in a vector sequence $\{a_k\}$ associated with matrix A as defined in definition 1, is equal to the energy in the singular spectrum.

The smallest non-zero singular value corresponds to the distance in Frobenius norm, of the matrix to the closest matrix of lower rank. This property makes SVD attractive for approximation and data reduction purposes.

3.2.3 Conceptual Relations between SVD and Oriented Energy)

We are now in the position to establish the link between the singular value decomposition and the concept of oriented energy distribution. Define the unit ball UB in $\boldsymbol{R}^m$ as UB = $\{\boldsymbol{q} = \boldsymbol{R}^m | \|\boldsymbol{q}\|_2 = \boldsymbol{1}\}$.

Theorem 2. *Consider a sequence of m-vectors $\{a_k\}$, $k = 1, 2, \ldots, n$ and the associated $m \times n$ matrix A with SVD as defined in Eq. (4) with $n \geq m$. Then,*

$$E_{\mathbf{u}_i}[A] = \sigma_j^2 \tag{69}$$

$$\forall \boldsymbol{q} \in \text{UB: if} \quad \boldsymbol{q} = \sum_{i=1}^{m} \gamma_i . \mathbf{u}_i, \text{ then}$$

$$E_{\boldsymbol{q}}[A] = \sum_{i-1}^{m} \gamma_i^2 . \sigma_i^2 \tag{70}$$

Proof. Trivial from Theorem 1.

The oriented energy measured in the direction of the i-th left singular vector of the matrix A, is equal to the i-th singular value squared. The energy in an arbitrary direction $\boldsymbol{q}$ is the linear combination of 'orthogonal' oriented energies associated with the left singular vectors. If the matrix A is rank deficient, then there exist directions in $\boldsymbol{R}^m$ that contain no energy at all.

With the aid of theorem 2, one can easily obtain, using the SVD, the directions and spaces of extremal energy, as follows:

Corollary 1. *Under the assumptions of Theorem 2*:

1. $\mathbf{max}_{q \in UB} E_{\boldsymbol{q}}[\boldsymbol{A}] = E_{\boldsymbol{u}_1}[\boldsymbol{A}] = \sigma_1^2,$ (71)

2. $\mathbf{min}_{q \in UB} E_{\boldsymbol{q}}[\boldsymbol{A}] = E_{\boldsymbol{u}_\mathrm{m}}[\boldsymbol{A}] = \sigma_\mathrm{m}^2,$ (72)

3. $\mathbf{max}_{Q^r \subset R^m} E_{Q^r}[\boldsymbol{A}] = E_{S_U^r}[\boldsymbol{A}] = \sum_{i=1}^{r} \sigma_i^2$, (73)

4. $\mathbf{min}_{Q^r \subset R^m} E_{Q^r}[\boldsymbol{A}] = E_{(S_U^{m-r})^\perp}[\boldsymbol{A}] = \sum_{i=m-r+1}^{m} \sigma_i^2$ (74)

where 'max' and 'min' denote operators, maximizing or minimizing overall *r*-dimensional subspaces Q^r of the space $\boldsymbol{R}^m$. S_U^r is the *r*-dimensional principal subspace of matrix A while $(S_U^{m-r})^\perp$denotes the *r*-dimensional orthogonal complement of S_U^{m-r}.

Property (71), (72), (73), and (74) follow immediately from the SVD theorem 1 and from Theorem 2. In words, properties (71) and (72) relate the SVD to the minima and maxima of the oriented energy distribution. In fact, it can be shown that extrema occur at each left singular direction.

The *r*-th principal subspace S_U^r is, among all *r*-dimensional subspaces of $\boldsymbol{R}^m$, the one that senses a maximal oriented energy (property 73). Properties (73) and (74) show that the orthogonal decomposition of the energy via the singular value decomposition is canonical in the sense that it allows subspaces of dimension *r* to be found where the sequence has minimal and maximal energy. This decomposition of the ambient space, as direct sum of a space of maximal and minimal energy for a given vector sequence, leads to very interesting rank consideration.

By establishing this link between the oriented energy and SVD, we proved that the first *r*-left singular vectors sensing the maximal energy of glove data matrix *A*, and thus account for most of the variation in the original data. This means that with an $m \times n$ data matrix that is usually overdetermined with much more samples (columns) than channels (rows), $n >> m$, the singular value decomposition allows most of signature characteristics to be compressed into *r* vectors.

3.3 Reference Signature

During the enrollment stage, 200 sample signatures from each writer to be enrolled are collected and the average distance between the principle vectors are computed. Based on these distance, a reference signature is selected as the one that presents minimal distance to the others.

3.4 Matching

Having modeled the signature through its principal subspace S_U^r in Section A, the authenticity of the tried signature can be determined by calculating average distance between the principal subspace and the reference or template signature. This angle is referred to as similarity factor (SF) and given in percent. A complete description of the singular value decomposition (SVD)-based algorithm for computing cosines of principal angles between two subspaces can be found in [9], [10].

3.5 The *r*-most Prominent Sensors

The previous section described a method of compressing the storage space required for the signature by representing the signature with *r* largest singular vectors. Using this technique, the redundancy in the data matrix, *A* has been reduced. In the next section, the *q*-most significant sensors (out of 14 sensors) will be used for feature extraction and matching. The selection of the *q* sensors can be obtained by calculation of F value which will be explained in the succeeding paragraph.

If the reference pattern of the *j* sensor of the signer *p*, $s_{rj}^{(p)}(nT)$, given by

$$s_{rj}^{(P)}(nT) = \frac{1}{N}\sum_{i=1}^{N} s_{ij}^{(P)}(nT) \tag{75}$$

where $s_{ij}^{(P)}(nT)$ is the i^{th} signal of sensor *j* of the signer *p* and N is the number of signatures [12]. T is sampling period and *n* is the time index. With 14 sensor data glove, a 14 reference patterns are collected for each signature.

The intra writer variance $V_{intra,j}$ and the inter writer variance $V_{inter,j}$ of the signals of sensor *j*, are calculated as follows,

$$V_{intra,j} = \sum_{i=1}^{N} \sum_{n} \left(S_{ij}^{(P)}(nT) - S_{rj}^{(P)}(nT) \right)^2 \tag{76}$$

$$V_{inter,j} = \sum_{p=1}^{P} \sum_{n} \left(S_{ij}^{(P)}(nT) - \overline{S_{rj}(nT)} \right)^2 \tag{77}$$

where $\overline{s_{rj}(nT)}$ is the average reference pattern of P signers of sensor j, given as

$$S_{rj}(nT) = \frac{1}{p} \sum_{P=1}^{P} s_{ij}^{(P)}(nT) \tag{78}$$

Next, the F value, defined as

$$F_j = \frac{V\ inter,j}{V\ intra,j} \tag{79}$$

is used to indicate the amount of individuality provided by each sensor. The higher the F value is, the higher the amount of individuality provided by the sensor, and vice versa.

3.6 Experiment Procedure

In order to develop the online signature verification system a series of experiments was performed on the signature database and the result of each experiment is analyzed. The database contained 20 genuine signature from 3 signers and 200 forgery signature for each signer. From here on, the genuine signature samples set will be referred as Genuine 1, 2 and 3 while forgery signature sample set will be referred as Forgery 1, 2 & 3. The 5-steps experiments are:

- Step 1: To calculate the energy of all 14 sensors on every genuine sample. The result of this experiment will be used to reduce the redundancy in the data matrix, A which in turn save the storage space and computational time.
- Step 2: To calculate the average distance between singular vectors of different genuine signatures. This calculation is required in order to select a reference signature to be stored in signature verification system.
- Step 3: To calculate the similarity factor between the reference signature (obtained in step 2) with all forgery signatures. Based on this calculation, the threshold value for the signature verification system can be determined.
- Step 4: To determine the 7-most prominent sensors by calculating the F value.
- Step 5: To repeat step 2 and 3 to the new database formed from step 4.

CHAPTER 4
RESULT AND DISCUSSION

In this chapter, the results of the experiment for signature verification system are presented in two different sections. In section 4.1, the experimental results on 14-sensor data matrix, A are presented on the calculation of oriented energy, reference signature and decision threshold. Next, in section 4.2, the results on reduced-sensors case are presented covering the calculation of reference signature and decision threshold.

4.1 14- Sensors Signature Verification System

In the first set of experiment, the technique for signature verification system for 14-sensor matrix is presented. It involves plotting the energy content in data matrix, A as a function of the number of singular vector. This will be used to reduce the redundancy on A which in turn reduce the storage and computational requirement in the signature verification system. In the second experiment, with A being represented with less singular vectors, the reference/template signature is being determined. Finally, the decision threshold for the signature verification system is decided based on the evaluation of similarity factor between the template signature and all the respective forgeries samples.

4.1.1 Oriented Energy

The plot of the energy content in the first 14 singular vectors of the data glove output matrix A is shown in Figure 15 . The graph shows that most of the signature energy is compressed in the first 5 singular vectors of data matrix A. In other word, it is sufficient to store only the first 5 singular vectors of the signature instead of storing all. This property is useful because it allows for a substantial reduction in the amount of storage space for enrollment as well as the computational time for matching. At the same time, this will help reducing the differences between different attempts of the same signature due to the emotional state of the signing person.

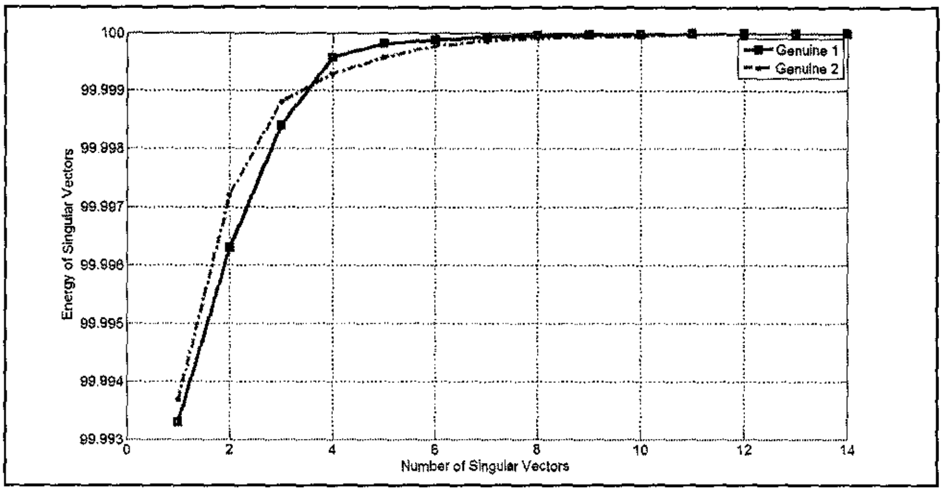


Figure 15: Amount of energy as a function of the first 14 singular vectors

4.1.2 Reference Signature

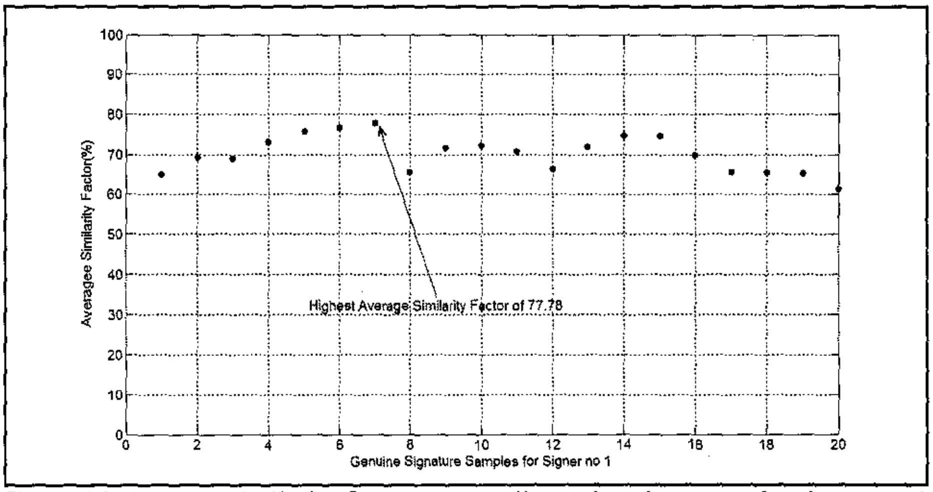


Figure 16: Average similarity factor among all genuine signatures for signer no. 1

Having model the signature through its 5-left-singular vectors of its data matrix A, the selection of reference signature from the database was done by calculating the average distance between singular vectors of different trials. This calculation result for genuine 1 is shown in Figure 16. From the figure, we can see that signature no7 can be selected as the reference because it has the highest average similarity factor of 77.78%.

4.1.3 Decision Threshold

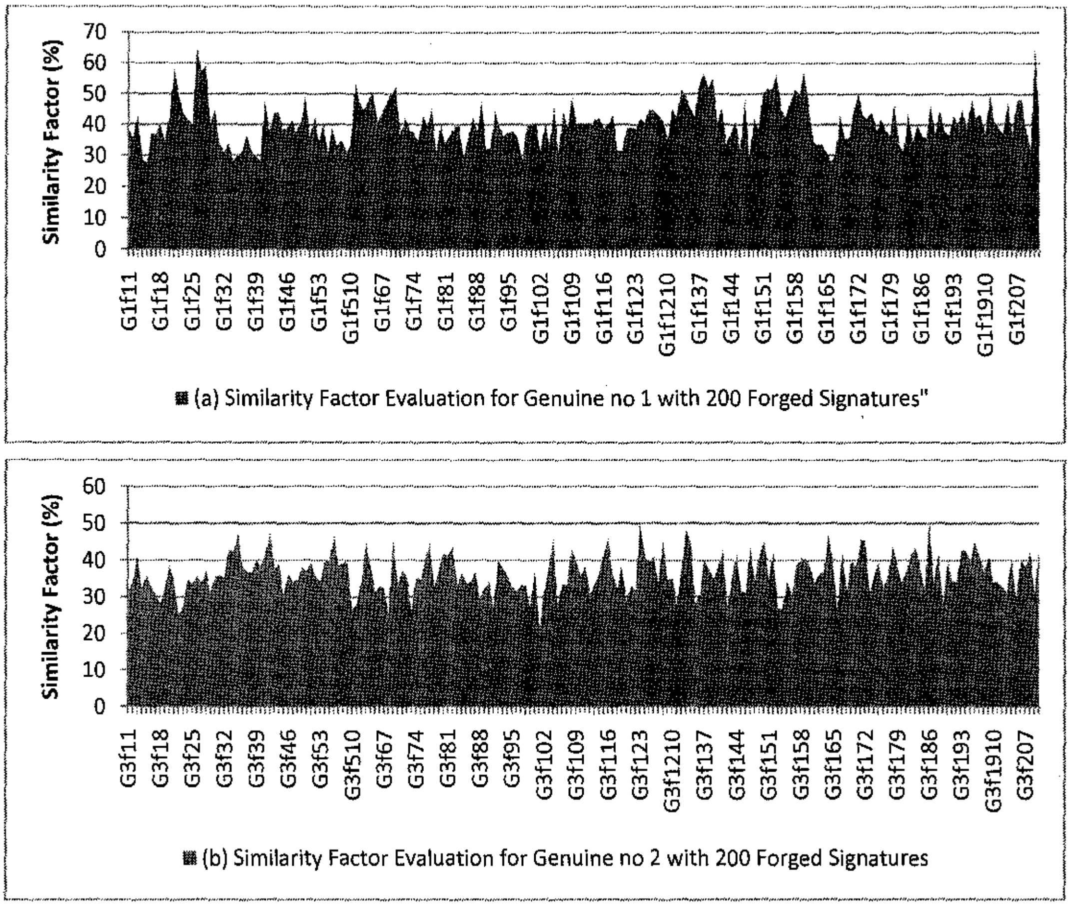


Figure 17.: Similarity factor plots for (a) Genuine 1, (b) Genuine 2

Threshold value for signers is selected based on the calculation of similarity factor between the reference signatures obtained in section 4.1.2 with all forgery signatures. Figure 17 (a-b) shows the plot of similarity factor value between the reference and 200 forgeries signatures for 3 signers. It is clear from the graph that if the decision threshold for forgery is set at any value above 70%, the proposed technique would recognize forgery signature with very low false acceptance rate (FAR) and false rejection rate (FRR).

4.2 Reduced Sensors Signature Verification System

In the second set of the experiment, the 7-most prominent sensors in the data matrix are to be determined as a method to reduce the amount of sensor required for signature verification. Once the value of 7-most prominent sensors is known the SVD technique is applied to determine the reference signature and decision threshold value.

4.2.1 7-most Prominent Sensors

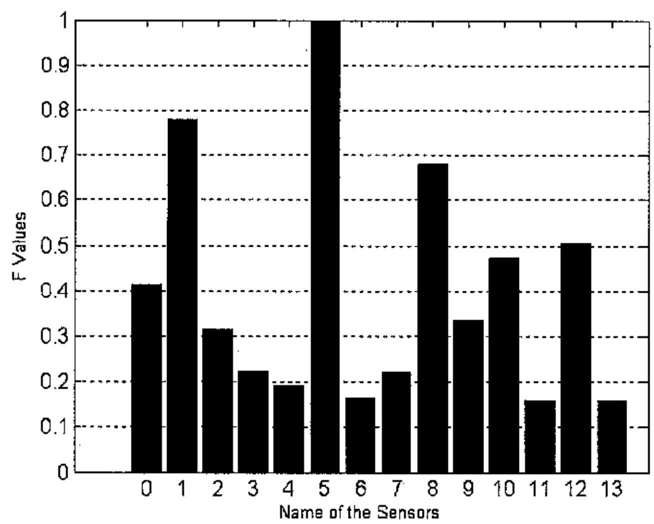


Figure 18: F values for 14 sensors of the data glove

For 20 samples of genuine signature, the reference patterns are calculated for each sensor. Next, the intra variance and inter variance for each sensor are calculated and the F values for the 14 sensors are obtained as shown in Figure 18. A set of 7-most prominent sensors (5, 1, 8, 12, 10, 0, 9) arranged in decreasing F-value are given in Table 1. The new set of data matrix A_r, with only 7 sensors will be used in the next experiment which is on selecting reference signature and threshold value.

Table 1: 7 reduced sensors of data glove

Sensor	Description	F value
5	Index-middle finger abduction	1
1	Thumb flexure (second joint)	0.778428
8	Middle-ring finger abduction	0.679025
12	Little finger flexure (at knuckle)	0.50505
10	Ring finger flexure (second joint)	0.4713
0	Thumb flexure (lower joint)	0.412622
9	Ring finger flexure (at knuckle)	0.334443

4.2.2 Reference Signature

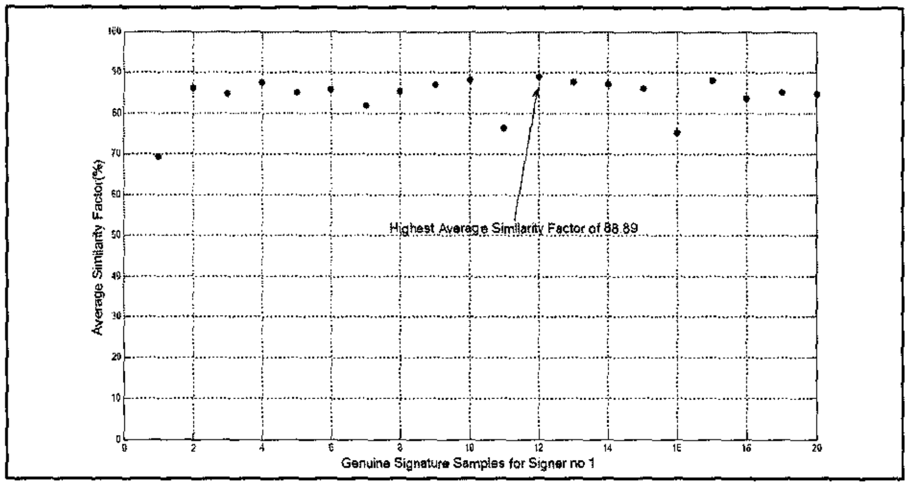
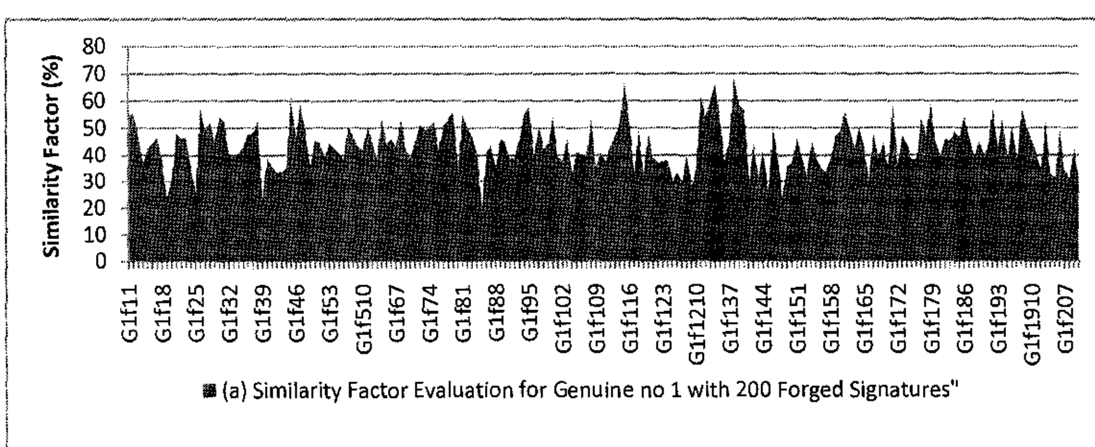


Figure 19.:Figure 16. Average similarity factor among all genuine signatures for signer no. 1

By reducing the sensors, the size of the reduced data matrix, A_r is equal to $m \times 7$. The selection of reference signature from the database was done as in section 4.1.2, that is, by calculating the average distance between singular vectors of different trials. This calculation result for genuine 2 is shown in Figure 19 which shows that signature 12 can be selected as the reference because it has the highest average similarity factor of 88.89%. The lower value of similarity factor is expected because as we physically reduced the data matrix by removing 7 sensors, we are at the same time throwing away some information content of A.

4.2.3 Decision Threshold



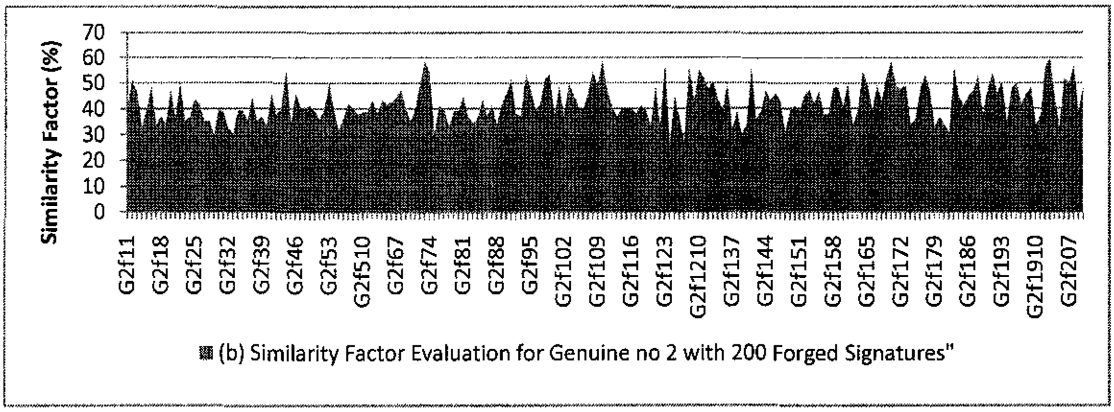


Figure 20: Similarity factor plots for (a) Genuine 1, (b) Genuine 2

The value of decision threshold for the reduced sensors experiment is equal to 75% as shown in Figure 20. The threshold is expected to be higher than in section 4.1.3 for the same reason mentioned in the last section, that is, the reduced sensors data, A_r carry less information than the original A.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

This chapter summarized the finding of this work and give recommendation on further work in this area.

5.1 Conclusion

An approach to signature verification problem with data glove as input device to online signature verification system is presented. The technique is based on the singular value decomposition in finding a set of singular vectors sensing the maximum energy of the signature. This limited set of vectors is referred to as the principal subspace of data glove output matrix A and used to model the signature. The average distance between the different principal subspaces is used for signature classification. In the first experiment, the optimal value of r as a compromise between the reduced dimensionality and truncated energy of matrix A was found. Next, the decision threshold is sought and the value of 70% for the similarity factor was found to give sufficiently low FRR and FAR. In the second experiment, we attempted to test the performance of the signature verification system by reducing the number of sensors in the data glove by half. The selection of 7 most prominent sensors was done based on the calculation of F- value for each sensor. Next the decision threshold is sought and the value of 1.8% EER was found to give sufficiently low FRR and FAR.

This work has demonstrated the effectiveness of data glove as an input device for the system. The system has the potential to offer a high level of security for special applications, including banking and electronic commerce. Off course, to reduce the cost of the glove, the number of sensors on the data glove can be reduced without affecting its performance in signature verification. This will result in lower cost of the system and make it available for the average consumer.

5.2 Recommendation

In this work, we tested the performance of the signature verification system with the data glove sensors being reduced to half. The performance of the system can be further experimented by evaluating it at difference number of reduced sensors.

From this test, the optimum number of sensors required for signature verification system can be determined which will reduce the cost of the system without affecting its performance.

REFERENCES

[1] http://www.bioid.com/sdk/docs/About_EER.htm

[2] Nidal S. Kamel, Senior Member, Shohel Sayeed, Member and Grant A. Ellis, Senior Member,"*Glove-Based Approach to Online Signature Verification*", IEEE transaction on pattern analysis and machine intelligence, vol. 30, NO. 6, June2008

[3] http://en.wikipedia.org/wiki/, Oct 2008

[4] L. E. Spence, A. J. Insel , and S. H. Friedberg, *Elementary Linear Algebra: A Matrix Approach*, Prentice Hall, 1999

[5] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, 4th edition, pp.823

[6] D. C. Lay, *Linear Algebra and Its Applications*, 3rd ed.,Pearson Education.

[7] V.C. Klema and A.J. Laub "*The Singular Value Decomposition: Its Computation and Some Applications*," IEEE Trans Autom. Control, vol. AC-25, pg. 164-176

[8] Ed. F Deprettere., *"SVD and Signal Processing: algorithms, applications and architectures"*, North-Holland Publishing Co., 1989.

[9] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[10] Z. Drmac, "On Principal Angles between Subspaces of Euclidean Space,"SIAM J. Matrix Analysis and Applications, vol. 22, pp. 173-194, 2000..

[11] A. S. Tolba, " GloveSignature: A Virtual-Reality-Based System for Dynamic Signature Verification," Digital Signal Processing Academic Press, vol. 9, pp. 241-266, 1999.

[12] Mingming, M. and Wijesoma, W. (2000). Automatic Online Signature Verification based on Multiple Models. In CIFEr.'01: Computational Engineering in Financial Engineering Conference (pp. 30-33), IEEE.

[13] SVD and Signal Processing: Algorithms, Applications and Architectures, F. Deprettere, ed., North-Holland Publishing Co., 1989.

APPENDICES

APPENDIX A
DATA GLOVE

Expensive high-end wired gloves can also provide haptic feedback, which is a simulation of the sense of touch. This allows a wired glove to also be used as an output device. Traditionally, wired gloves have only been available at a huge cost, with the finger bend sensors and the tracking device having to be bought separately.

Concerned about the high cost of the most complete commercial solutions, Pamplona et al. propose a new input device: an image-based data glove (IBDG). By attaching a camera to the hand of the user and a visual marker to each finger tip, they use computer vision techniques to estimate the relative position of the finger tips. Once they have information about the tips, they apply inverse kinematics techniques {GirardMaciejewski1985} in order to estimate the position of each finger joint and recreate the movements of the fingers of the user in a virtual world. Adding a motion tracker device, one can also map pitch, yaw, roll and XYZ-translations of the hand of the user, (almost) recreating all the gesture and posture performed by the hand of the user in a low cost device.

One of the first wired gloves available to home users was the Nintendo Power Glove. This was designed as a gaming glove for the Nintendo Entertainment System. It had a crude tracker and finger bend sensors, plus buttons on the back. In 2001, Essential Reality made a similar attempt at a cheap gaming glove, this time for the PC: the P5 Glove. However, this peripheral never really became popular among gamers. Ironically, even specialized stores are now selling the older and less performant Power Glove for a higher price than the more sophisticated P5 Glove.

Wired gloves are often called "datagloves" or "cybergloves", but these two terms are trademarks, belonging to Sun Microsystems (which acquired the patent portfolio of VPL Research Inc. in February 1998) and Immersion Corporation (which acquired Virtual Technologies, Inc. and its patent portfolio in September 2000) respectively.

An alternative to wired gloves is to use a camera and computer vision to track the 3D pose and trajectory of the hand, at the cost of tactile feedback.

5DT Data Glove

The 5DT Data Glove Ultra has been designed to satisfy the stringent requirements of modern Motion Capture and Animation Professionals. It offers comfort, ease of use, a small form factor and multiple application drivers. The high data quality, low cross-correlation and high data rate make it ideal for realistic real time animation.

Features:

- Advanced Sensor

This new range of data gloves from 5DT features a completely redesigned sensor technology. The new sensors make these gloves more comfortable and give more consistent data across a large range of hand sizes. Cross-correlation has been reduced significantly.

- Bluetooth Wireless Option

A wireless option is available, based on the latest Bluetooth technology for high bandwidth, wireless connectivity up to a range of 20m. The wireless kit can run for 8 hours off a single battery pack. The battery pack can be exchanged in seconds when necessary.

- 5 and 14 Sensor Gloves Available

The 5DT Data Glove Ultra range is available in a 5 sensor and 14 sensor configuration with a host of options such as right- and left-handed versions.

- Cross-Platform SDK

The glove SDK is available for Windows as well as Linux and UNIX. It is also possible to interface the glove without the SDK since it has an open-source communications protocol.

- Interface Options protocol

The Ultra range of data gloves now comes standard with a USB interface, eliminating the need for an external power supply. An open-source, open-platform serial interface is available for workstation or embedded applications.

- Wide Application Support

The 5DT Data Glove is now supported in most of the leading 3D modeling and animation packages.

APPENDIX B
MATLAB CODING

The MATLAB software was the most important software which used in this work, for calculating SVD. The data from data glove is a matrix in $n \times 14$ dimension, n is the sampling time which differs from signature by signature, for example one signature's data is shown in the Table 5 which has dimension as 61×14.

Table 2: Data generated by data glove

ms	Glove 1 Thumb Ne:	Glove 1 Thumb Far	Glove 1 Thumb/Ind	Glove 1 Index Near	Glove 1 Index Far	Glove 1 Index/Midt	Glove 1 Middle Ne:	Glove 1 Middle Far	Glove 1 Middle/Rin	Glove 1 Ring Near	Glove 1 Ring Far	Glove 1 Ring/Little	Glove 1 Little Near	Glove 1 Little Far
0	566	1511	1845	1222	2124	2734	1465	1679	2736	1917	2138	3141	1698	1900
16	566	1512	1846	1222	2124	2736	1464	1678	2736	1916	2138	3140	1698	1900
32	567	1513	1847	1222	2123	2736	1463	1677	2736	1915	2138	3138	1698	1901
48	568	1514	1848	1223	2124	2736	1461	1676	2736	1914	2139	3136	1698	1902
64	568	1513	1849	1223	2123	2736	1459	1676	2736	1913	2139	3135	1697	1903
80	568	1512	1851	1222	2123	2736	1457	1675	2737	1911	2139	3135	1696	1902
96	567	1512	1851	1222	2122	2736	1456	1674	2737	1910	2138	3135	1696	1901
112	568	1511	1852	1223	2122	2736	1455	1673	2738	1910	2138	3135	1696	1901
128	569	1511	1854	1222	2122	2736	1453	1672	2739	1909	2138	3134	1696	1902
144	570	1511	1854	1222	2122	2737	1451	1672	2740	1907	2138	3132	1695	1901
160	570	1510	1855	1221	2122	2737	1449	1672	2742	1905	2138	3129	1694	1900
176	571	1510	1856	1221	2123	2737	1448	1672	2744	1904	2138	3128	1694	1901
192	571	1510	1857	1222	2123	2737	1447	1671	2745	1903	2137	3128	1694	1903
208	571	1511	1858	1223	2125	2737	1446	1672	2745	1901	2137	3126	1694	1904
224	571	1513	1858	1224	2126	2741	1443	1672	2742	1900	2137	3124	1694	1905
240	572	1515	1860	1225	2128	2743	1443	1674	2740	1899	2138	3122	1695	1905
256	573	1520	1869	1233	2135	2750	1444	1678	2737	1899	2140	3122	1696	1909
272	572	1523	1883	1236	2137	2751	1445	1680	2736	1899	2141	3124	1696	1911
288	572	1526	1890	1239	2140	2752	1445	1682	2736	1900	2142	3126	1697	1914
304	572	1528	1894	1241	2143	2752	1446	1684	2736	1900	2144	3127	1697	1915
320	571	1531	1898	1244	2145	2752	1446	1686	2736	1901	2145	3129	1698	1918
336	571	1540	1903	1248	2149	2753	1447	1689	2736	1903	2147	3130	1699	1923
352	571	1543	1906	1249	2151	2754	1448	1691	2735	1905	2148	3133	1699	1925
368	571	1546	1907	1249	2152	2755	1449	1693	2731	1905	2150	3133	1700	1927
384	571	1548	1908	1250	2153	2757	1449	1694	2731	1906	2151	3134	1700	1927
400	571	1550	1909	1250	2154	2757	1449	1696	2729	1906	2152	3134	1700	1928
416	571	1551	1911	1250	2154	2758	1449	1696	2728	1907	2152	3134	1700	1929
432	571	1551	1912	1251	2154	2758	1449	1697	2729	1907	2152	3135	1700	1929
448	570	1551	1910	1252	2154	2758	1449	1697	2729	1908	2152	3135	1700	1929
464	568	1548	1905	1255	2153	2757	1449	1696	2731	1908	2152	3135	1699	1929
480	565	1542	1888	1255	2151	2755	1447	1695	2735	1909	2152	3135	1698	1927
496	561	1531	1854	1249	2145	2755	1446	1692	2736	1909	2151	3135	1695	1922
512	558	1520	1834	1239	2141	2755	1447	1688	2736	1909	2149	3135	1692	1912
528	556	1514	1829	1230	2138	2757	1447	1684	2737	1909	2147	3133	1689	1906
544	556	1510	1827	1224	2136	2758	1446	1683	2739	1909	2145	3133	1688	1903
560	557	1507	1828	1221	2135	2754	1444	1681	2744	1909	2144	3130	1687	1903
576	560	1507	1831	1220	2134	2753	1443	1680	2746	1908	2143	3128	1687	1903
592	561	1507	1833	1219	2133	2752	1442	1679	2746	1907	2143	3124	1687	1903
608	563	1508	1835	1220	2133	2752	1440	1679	2748	1906	2143	3120	1687	1903
624	564	1508	1837	1220	2133	2752	1439	1678	2749	1905	2142	3120	1687	1903
640	566	1509	1839	1220	2133	2752	1439	1677	2750	1905	2142	3120	1688	1903
656	567	1510	1841	1219	2133	2752	1437	1677	2750	1904	2141	3120	1688	1902
672	568	1510	1843	1218	2133	2752	1437	1676	2751	1903	2140	3119	1688	1902
688	568	1511	1845	1218	2133	2752	1435	1676	2751	1901	2140	3112	1688	1901
704	569	1512	1846	1218	2133	2752	1434	1675	2751	1901	2140	3108	1688	1901
720	569	1513	1846	1218	2134	2752	1433	1676	2752	1900	2140	3105	1689	1900
736	570	1514	1847	1217	2135	2752	1432	1677	2751	1899	2140	3103	1689	1899
752	571	1515	1848	1217	2135	2752	1431	1678	2749	1898	2140	3103	1689	1900
768	572	1518	1849	1219	2136	2752	1431	1679	2741	1898	2141	3103	1690	1902
784	572	1519	1849	1219	2137	2752	1431	1681	2737	1899	2142	3103	1691	1905
800	573	1520	1849	1218	2138	2752	1432	1682	2737	1899	2142	3106	1692	1908
816	574	1521	1848	1217	2138	2752	1433	1683	2736	1900	2143	3109	1694	1910
832	574	1522	1847	1216	2138	2753	1433	1683	2736	1900	2144	3112	1695	1910
848	575	1523	1847	1213	2139	2753	1434	1685	2732	1900	2144	3120	1697	1912
864	575	1523	1846	1213	2140	2753	1434	1685	2728	1901	2145	3120	1699	1913
880	576	1524	1845	1212	2140	2753	1435	1685	2728	1902	2145	3121	1701	1915
896	576	1525	1844	1211	2140	2753	1435	1685	2727	1902	2146	3128	1702	1917
912	575	1524	1843	1211	2140	2753	1435	1686	2726	1902	2146	3130	1704	1919
928	576	1524	1842	1211	2140	2753	1434	1686	2725	1902	2146	3131	1705	1921
944	575	1525	1842	1211	2140	2753	1434	1686	2726	1902	2146	3133	1705	1923
960	575	1524	1842	1212	2140	2753	1435	1686	2726	1903	2147	3133	1706	1924

In SVD matrices, the right singular vector was the matrix that is used for signature verification. The right singular vector can be calculated by inserting the signature data as a matrix to MATLAB command window, such as a=[the data], next for finding SVD, the following codes shall be written command window:

```
>> [a1,a2,a3]=svd(a);

>> a3

a3 =
```

-0.0756	0.0099	-0.0993	-0.0371	0.5495	0.0857	0.2169	-0.3239	-0.1243	0.0129	-0.3657	0.5576	-0.1794	0.1707
-0.2021	0.3110	-0.1795	-0.1879	-0.0258	0.3870	0.2618	0.2783	-0.5837	-0.1825	0.3372	0.0731	0.0707	-0.0222
-0.2470	0.7353	0.2132	0.3076	0.3188	-0.0577	-0.3132	0.1146	0.0859	0.1019	-0.0689	-0.1257	-0.0275	-0.0542
-0.1630	0.3088	0.1912	0.1811	-0.5704	-0.1993	0.4768	-0.3991	-0.0596	0.0265	-0.0889	0.0453	-0.2037	0.0354
-0.2838	0.0781	-0.3465	-0.0056	-0.1633	0.0705	-0.1939	-0.0297	0.3296	-0.4810	0.0146	-0.0048	-0.0513	0.6167
-0.3653	-0.1411	-0.3993	0.1514	-0.1205	-0.0635	-0.4501	-0.3889	-0.2221	0.3173	0.2776	0.1333	-0.0744	-0.2037
-0.1918	-0.0680	0.5351	-0.0645	-0.0636	0.4038	-0.1093	-0.2870	0.1874	-0.0160	0.2038	0.2701	0.5055	0.0359
-0.2235	0.0451	-0.1863	-0.1519	-0.1774	0.2795	-0.0345	0.0516	0.2678	-0.2764	-0.4434	0.0629	-0.0412	-0.6530
-0.3638	-0.3772	0.0413	0.6847	0.1618	-0.0771	0.2904	0.2529	0.0202	-0.2151	0.0877	0.0250	0.0879	-0.1083
-0.2531	-0.1862	0.2078	-0.0496	-0.0731	0.4521	0.0215	0.2519	0.1533	0.4248	-0.0006	-0.0738	-0.5859	0.1715
-0.2848	-0.0814	-0.1377	-0.0353	-0.0577	0.0598	0.0766	0.0254	-0.2179	0.3477	-0.5381	-0.3600	0.4821	0.2424
-0.4154	-0.1805	0.4055	-0.3955	-0.0113	-0.4730	-0.2225	0.1639	-0.2848	-0.2331	-0.1145	0.0784	-0.1383	0.0072
-0.2252	-0.0540	0.0062	-0.2520	0.3956	0.0147	0.2588	-0.4323	0.1261	-0.1333	0.2358	-0.5983	-0.1007	-0.1221
-0.2538	0.1239	-0.2169	-0.3054	0.0462	-0.3345	0.3138	0.2572	0.4469	0.3565	0.2420	0.2672	0.2063	-0.0429

After finding right singular vectors for the specific signatures that we want to measure their similarity factor, we shall save their right singular vectors in an M-file and then apply the cosine value between them, such as following M-file:

```
echo off
clear all
close all

a=[-0.1194    -0.5598    -0.6836     0.1954     0.2068    -0.3289   0.1258
   -0.2831     0.3363    -0.3510    -0.7952     0.1088    -0.1778   0.0883
   -0.3497     0.3652    -0.3732     0.3102    -0.6881     0.0949   0.1623
   -0.5077    -0.0962    -0.0179     0.0101     0.3800     0.7254   0.2491
   -0.5002     0.3604     0.2232     0.4085     0.4067    -0.4596   0.1628
   -0.4084    -0.4454     0.1541    -0.0897    -0.3108     0.0663   0.7086
   -0.3295    -0.3204     0.4430    -0.2414    -0.2558    -0.33070.5995];

b=[-0.1184    -0.2679    -0.7534     0.2602     0.3833    -0.3120   0.1860
   -0.2784     0.4783    -0.2557    -0.7547     0.1907    -0.1285   0.0766
   -0.3510     0.4366    -0.3740     0.3479    -0.6310     0.1634   0.0217
```

```
    -0.5094    -0.1530     0.0770    -0.0186     0.2902     0.7001   0.3695
    -0.4994     0.3053     0.4387     0.3962     0.3379    -0.4332   0.0783
    -0.4099    -0.4933    -0.0490    -0.1014    -0.1049     0.0364   0.7507
    -0.3292    -0.3858     0.1622    -0.2723    -0.4574    -0.42470.5028];

c=[-0.1198    -0.3981    -0.5601     0.6669    -0.0899     0.2350   0.0734
    -0.2827     0.4397    -0.7255    -0.3054     0.1956    -0.1128   0.2369
    -0.3448     0.4096     0.1321     0.2111    -0.7929    -0.0227   0.1491
    -0.5080    -0.0557     0.0299     0.0061     0.0806    -0.4879   0.7024
    -0.5037     0.2670     0.3581     0.3292     0.5250     0.3677   0.1660
    -0.4080    -0.5786     0.1083    -0.1425    -0.0211    -0.3509   0.5857
    -0.3296    -0.2668    -0.0399    -0.5373    -0.2061     0.65960.2287];

d=[-0.1213    -0.3046     0.1772    -0.8248     0.1509    -0.3913   0.0708
    -0.2845     0.5073     0.7817     0.0377     0.1767     0.0119   0.1333
    -0.3455     0.4272    -0.2374    -0.3287    -0.7236     0.0676   0.0748
    -0.5095    -0.1350    -0.0408    -0.0557     0.1894     0.5468   0.6185
    -0.4981     0.2612    -0.5025     0.1039     0.5010    -0.3433   0.2269
    -0.4093    -0.5562     0.1213     0.0867    -0.1205     0.2688   0.6435
    -0.3314    -0.2676     0.1797     0.4345    -0.3479    -0.59410.3514];

s   = 5; % number of subspaces, the first 5 singular vectors in each
singular matrix

U1  = a; % U1 can be assigned to any of the vectors above
U2  = b; % U2 can be assigned to any of the vectors above

for u = 1:s
   d  = U1(:,u).'*U2(:,u); % d is the inner product of U1 and U2
   D1 = sqrt(sum(U1(:,u).^2)); % D1 is the Norm of U1
   D2 = sqrt(sum(U2(:,u).^2)); % D2 is the Norm of U2
   dis(u) = d/( D1*D2); % dis(u)measures the cosine value between
two singular matrix
end
x = mean(abs(dis))*100 % x equals to the mean value of dis(u)by 100
times
```