Spring 2007

What Can Your Computer Recognize: Chemical and Facial Pattern Recognition Through the Use of the Eigen Analysis Method

Anthony J. Giordano
Regis University

Thesis written by

Anthony J. Giordano

Approved by

Rebecca G Brewer May 1, 2007

Dr. Rebecca G. Brewer, Assistant Professor of Chemistry, Co-Advisor

Jim May 1, 2007

Dr. James A. Seibert, Associate Professor of Mathematics, Co-Advisor

Accepted by

Thomas 1 MAY 2007

Dr. Thomas G. Bowie, Regis University Honors Program

Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I wish to acknowledge Dr. Seibert and Dr. Brewer for their unceasing aid throughout the course of this endeavor. Their friendship and professional nature meant a great deal to me. Without their continual input, guidance, and advice this project would not have been possible. I wish to also thank Dr. Bowie for his kind words of wisdom and his ability to always know that the project would be completed, even when technical difficulties stood in my path. I am particularly grateful for Dr. Bowie's ability to push me along whenever I became an academic in need of direction. I must also thank them all for being so willing to go through draft after draft of this thesis. As you can see, the final product became quite a tome to read through for each draft and I cannot thank them enough for doing so without hesitation. I also owe a great debt of gratitude to Michael Uhrig for his aid in the mathematical proofs and the face recognition aspect of the project.

Thanks as well to Dr. Suzanne Caulk for the use of her computer and office during a large portion of the research. I am particularly grateful to Dr. Surendra Mahapatro for his ability to help me explore the meaning of research and his thoughtful comments and questions.

I cannot conclude without thanking my parents. Their support has always been a major driving force in my pursuits and I do not think it would have been possible to achieve so many of my goals without their help. Both my mother and father were instrumental in helping me clarify my thoughts and better understand the finished product, for this and many other things I am ever grateful.

1

Introduction

Seeing patterns in the world is part of the human condition. If the numbers $2, 4, 6, 8, \ldots$ are put before someone they will readily recognize the pattern of counting by two and be able to continue the sequence with the number 10, 12, ... Similarly, someone who is moderately acquainted with mathematics would recognize the numbers $0, 1, 1, 2, 3, 5, 8, \ldots$ as the Fibonacci sequence. Yet, patterns are not simply limited to what can be observed within mathematical relationships. Figure 1 shows a pattern made of rope that the human eye and mind can detect as such.

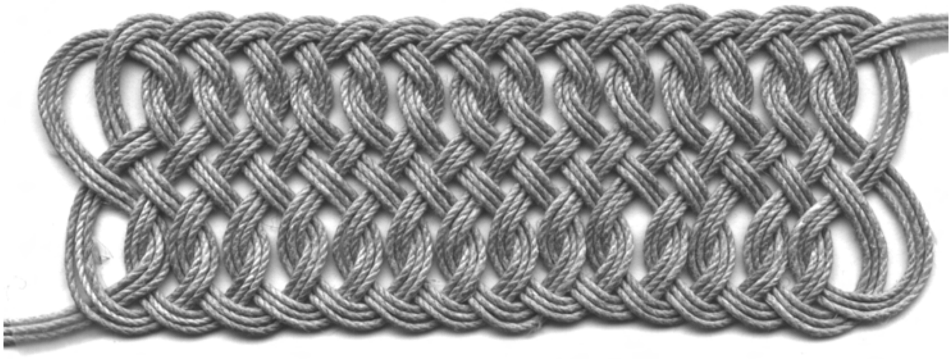

Figure 1: Pattern of a Woven Rope Frieze [1]

Yet, while humans can identify the pattern found within the frieze, a computer could not perform the same recognition with the ease or sophistication inherent to the human mind. Even the seemingly simple act of reading and comprehending the sentences on a page is dataset vector onto the space spanned by the first j eigenvectors. The data originating from outside the database are then reshaped in the same way and projected onto the same subspace. At this point, it is a simple matter of taking the Euclidean distance between the coefficients of the orthogonal projections of the newly tested data and the information which originally comprised the database. A set of closest matches can then be output, leaving the final interpretation of the result up to the human operator. Thus, this system must be considered a supervised form of recognition.

The particular applications addressed here revolve around face recognition methodologies and chemometric analysis of compounds. The immediate thought regarding these two applications is their apparent disconnection from one another, yet such a conclusion could not be further from the truth. In fact, the use of these two seemingly disparate applications will aid in demonstrating the true robustness of the KL system and how an application of pattern recognition in one particular area does not mean it must be limited to that field. The KL technique was originally tested by using it within the context of face recognition. Yet, there is no reason that the same principle cannot be expanded into chemometrics. The idea behind this field of chemistry, which was introduced by Svante Wold and Bruce R. Kowalski early in the 1970s, is to obtain relevant information from experimental data and be able to display the usually large amount of data in a manner that is both meaningful and useful [6, p. 5-6]. A method commonly discussed in the literature is principle component analysis, or PCA, which is essentially the singular-value decomposition or Karhunen-Loéve technique. Consequently, the mathematics behind each application is the same and the only modifications required are to the programming of the software.

2

Linear Algebra Applications to Face Recognition

In order to understand how both the face recognition and chemical recognition databases function it is important to discuss the math behind the KL expansion method. This process relies on properties of the singular-value decomposition (SVD) of a matrix. The SVD of any matrix A provides us with a way to decompose that matrix to the form

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

where if A is $p \times n$ then U is a $p \times p$ orthogonal matrix whose columns provide a basis for the columnspace of A and V is an $n \times n$ matrix whose columns provide a basis for the rowspace of A. For our purposes only a portion of the SVD will be used.

We should note here that the following propositions are all well established already. However, for the sake of completeness and to provide the reader with a greater understanding of this area we determined it would be beneficial to include all of the proofs. Most of the following proofs can be found in [15], although for proof of Proposition 2.5 a good source would be [14]. The purpose of the first six propositions is to establish certain properties of the $n \times n$ matrix A^TA, namely that it has a set of eigenvectors which form a basis for $\mathbb{R}^n$.

Proposition 2.1. *Given any matrix A, the matrix A^TA is symmetric.*

Proof. Define the matrix A as the set of vectors

$$A = \begin{pmatrix} \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_n \\ \\ \end{pmatrix}$$

and

$$A^T = \begin{pmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \vdots \\ \mathbf{v}_n^T \end{pmatrix}.$$

Then the ij^{th} entry of A^TA is

$$\mathbf{v}_i^T\mathbf{v}_j = \mathbf{v}_i \cdot \mathbf{v}_j.$$

Similarly, the ji^{th} entry is

$$\mathbf{v}_j^T\mathbf{v}_i = \mathbf{v}_j \cdot \mathbf{v}_i.$$

Since the dot product of two vectors is commutative, $\mathbf{v}_i \cdot \mathbf{v}_j = \mathbf{v}_j \cdot \mathbf{v}_i$. □

Proposition 2.2. *For any symmetric matrix A, the eigenvectors from different eigenspaces are always orthogonal.*

Proof. Let $\mathbf{x}_a$ and $\mathbf{x}_b$ be two eigenvectors of A corresponding to two distinct eigenvalues λ_a and λ_b, respectively. Then,

$$\lambda_a\mathbf{x}_a \cdot \mathbf{x}_b$$

$$= (A\mathbf{x}_a)^T\mathbf{x}_b$$

$$= \mathbf{x}_a^T A^T \mathbf{x}_b$$

$$= \mathbf{x}_a^T A \mathbf{x}_b$$

$$= \mathbf{x}_a^T \lambda_b \mathbf{x}_b$$

$$\lambda_a \mathbf{x}_a \cdot \mathbf{x}_b = \lambda_b \mathbf{x}_a \cdot \mathbf{x}_b$$

$$(\lambda_a - \lambda_b)\mathbf{x}_a \cdot \mathbf{x}_b = 0.$$

Since λ_a and λ_b are distinct, $\lambda_a - \lambda_b \neq 0$ and $\mathbf{x}_a \cdot \mathbf{x}_b$ must equal zero. Therefore, the vectors $\mathbf{x}_a$ and $\mathbf{x}_b$ are orthogonal.

□

Proposition 2.3. *The eigenvalues of $A^T A$ are all non-negative and real.*

Proof. Let $\mathbf{x}_a$ be a normalized eigenvector of $A^T A$ and λ_a be the corresponding eigenvalue. Then,

$$\begin{aligned}
\|A\mathbf{x}_a\|^2 &= (A\mathbf{x}_a)^T(A\mathbf{x}_a) \\
&= \mathbf{x}_a^T A^T A \mathbf{x}_a \\
&= \mathbf{x}_a^T(\lambda_a \mathbf{x}_a) \\
&= \lambda_a(\mathbf{x}_a^T \mathbf{x}) \\
&= \lambda_a \|\mathbf{x}_a\| \\
&= \lambda_a(1) \\
\|A\mathbf{x}_a\|^2 &= \lambda_a.
\end{aligned}$$

Therefore, the eigenvalues (λ_a) that are produced by $A^T A$ are all non-negative and real since the length of a vector squared is always nonnegative and real.

□

Definition 2.4. The singular values of a matrix A are equal to the length of $A\mathbf{x}$, where $\mathbf{x}$ is an eigenvector of A^TA. The set of singular values is denoted by $\sigma_1, \ldots, \sigma_n$. From Proposition 2.3 we also know that the singular values are equivalent to the square root of the eigenvalues of A^TA.

For the remainder of this paper it will be assumed that the singular values are arranged such that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$. It will also be assumed that the corresponding eigenvalues are arranged in decreasing order as well, such that $\lambda_a = {\sigma_a}^2$.

Proposition 2.5. *If $A(n \times n)$ has n real eigenvalues then there exists a matrix U with orthonormal columns such that U^TAU is upper triangular.*

Proof. Since A has n real eigenvalues $(\lambda_1 \ldots \lambda_n)$ we know that each eigenspace has at least one corresponding eigenvector. Take any one of these normalized vectors and let it be $\mathbf{v}_1$ with corresponding eigenvalue λ_1. It is possible to use this vector to create an orthonormal basis for $\mathbb{R}^n$. To do this, create $n-1$ vectors that are linearly independent of $\mathbf{v}_1$ and use the Gram-Schmidt process to create an orthonormal basis for $\mathbb{R}^n$. Call this set of new vectors $\{\mathbf{w}_1 \ldots \mathbf{w}_{n-1}\}$, and use this set of vectors in tandem with the eigenvector of A to create the following $n \times n$ matrix.

$$U_1 = \left(\begin{array}{cccc} \\ \mathbf{v}_1 & \mathbf{w}_1 & \cdots & \mathbf{w}_{n-1} \\ \\ \end{array} \right)$$

For notation purposes it is also necessary to define the set of $\mathbf{w}$ vectors.

$$W = \left(\begin{array}{ccc} \\ \mathbf{w}_1 & \cdots & \mathbf{w}_{n-1} \\ \\ \end{array} \right)$$

The initial step in the formation of an upper triangular matrix is to multiply A and U_1 together to produce the matrix B, such that $B = U_1^T A U_1$.

$$\begin{aligned}
B &= \begin{pmatrix} \mathbf{v}_1^T \\ \mathbf{w}_1^T \\ \vdots \\ \mathbf{w}_{n-1}^T \end{pmatrix} \begin{pmatrix} A \end{pmatrix} \begin{pmatrix} \mathbf{v}_1 & \mathbf{w}_1 & \cdots & \mathbf{w}_{n-1} \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{v}_1^T \\ \mathbf{w}_1^T \\ \vdots \\ \mathbf{w}_{n-1}^T \end{pmatrix} \begin{pmatrix} A\mathbf{v}_1 & A\mathbf{w}_1 & \cdots & A\mathbf{w}_{n-1} \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{v}_1^T \\ \mathbf{w}_1^T \\ \vdots \\ \mathbf{w}_{n-1}^T \end{pmatrix} \begin{pmatrix} \lambda_1\mathbf{v}_1 & A\mathbf{w}_1 & \cdots & A\mathbf{w}_{n-1} \end{pmatrix} \\
&= \left(\begin{array}{c|ccc} \lambda_1\mathbf{v}_1 \cdot \mathbf{v}_1 & \mathbf{v}_1 \cdot (A\mathbf{w}_1) & \cdots & \mathbf{v}_1 \cdot (A\mathbf{w}_{n-1}) \\ \hline \lambda_1\mathbf{w}_1 \cdot \mathbf{v}_1 & & & \\ \vdots & & W^T A W & \\ \lambda_1\mathbf{w}_{n-1} \cdot \mathbf{v}_1 & & & \end{array}\right)
\end{aligned}$$

Since $\mathbf{v}_1$ is a normal vector and is orthogonal to the columns of W we can simplify this matrix to

$$B = \left(\begin{array}{c|c} \lambda_1 & \mathbf{v}_1^T A W \\ \hline & \\ \mathbf{0} & W^T A W \\ & \end{array}\right).$$

At this point it is important to note that the matrix B has the same eigenvalues as the original matrix A, this can be easily proven through the use of the characteristic polynomial.

$$\begin{aligned}
\mathrm{char}(U_1^T A U_1) &= \det(U_1^T A U_1 - \lambda I) \\
&= \det(U_1^T A U_1 - U_1^T \lambda U_1) \\
&= \det(U_1^T (A - \lambda I) U_1) \\
&= \det(U_1^T) \det(A - \lambda I) \det(U_1) \\
&= \det(U_1^T U_1) \det(A - \lambda I) \\
&= \det(I) \det(A - \lambda I) \\
&= \det(A - \lambda I) \\
&= \mathrm{char}(A)
\end{aligned}$$

Since B has the same eigenvalues as A $(\lambda_1 \dots \lambda_n)$ then the lower right partition of B, the $n-1 \times n-1$ matrix $W^T A W$, has $n-1$ eigenvalues $(\lambda_2 \dots \lambda_n)$. Now define $\mathbf{v}_2$ as being the normalized eigenvector of $W^T A W$, which corresponds to the eigenvalue λ_2. This vector is used to create an orthonormal basis for $\mathbb{R}^{n-1}$, $\{\mathbf{v}_2, \mathbf{y}_1, \dots, \mathbf{y}_{n-2}\}$. Use this set of vectors to create the matrix U_2.

$$U_2 = \left(\begin{array}{c|c}
1 & \mathbf{0} \\
\hline
\mathbf{0} & \mathbf{v}_2 \quad \mathbf{y}_1 \cdots \quad \mathbf{y}_{n-2}
\end{array}\right)$$

It will also become convenient later to define the $n-1 \times n-2$ matrix Y whose columns are the set $\{\mathbf{y}_1, \dots, \mathbf{y}_{n-2}\}$. Multiply U_2 and B in the same manner as previously conducted, which provided us with B.

$$
\begin{aligned}
U_2^T B U_2 &= \left(\begin{array}{c|c} 1 & \mathbf{0} \\ \hline \mathbf{0} & \begin{array}{c} \mathbf{v}_2^T \\ \mathbf{y}_1^T \\ \vdots \\ \mathbf{y}_{n-2}^T \end{array} \end{array}\right)
\left(\begin{array}{c|c} \lambda_1 & \mathbf{v}_1^T AW \\ \hline \mathbf{0} & W^T AW \end{array}\right)
\left(\begin{array}{c|c} 1 & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{v}_2 \quad \mathbf{y}_1 \cdots \quad \mathbf{y}_{n-2} \end{array}\right) \\
&= \left(\begin{array}{c|c} \lambda_1 & \mathbf{v}_1^T AW \\ \hline \mathbf{0} & \begin{bmatrix} \mathbf{v}_2^T \\ \mathbf{y}_1^T \\ \vdots \\ \mathbf{y}_{n-2}^T \end{bmatrix} W^T AW \end{array}\right)
\left(\begin{array}{c|c} 1 & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{v}_2 \quad \mathbf{y}_1 \cdots \quad \mathbf{y}_{n-2} \end{array}\right) \\
&= \left(\begin{array}{c|c} \lambda_1 & \mathbf{v}_1^T AW \\ \hline \mathbf{0} & \begin{array}{c} \lambda_2 \mathbf{v}_2^T \\ Y^T W^T AW \end{array} \end{array}\right)
\left(\begin{array}{c|c} 1 & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{v}_2 \quad \mathbf{y}_1 \cdots \quad \mathbf{y}_{n-2} \end{array}\right) \\
&= \left(\begin{array}{c|c} \lambda_1 & \mathbf{v}_1^T AW\mathbf{v}_2 \qquad \mathbf{v}_1^T AWY \\ \hline \mathbf{0} & \begin{array}{c|c} \lambda_2 & \lambda_2 \mathbf{v}_2^T Y \\ \hline Y^T \lambda_2 \mathbf{v}_2 & Y^T W^T AWY \end{array} \end{array}\right) \\
&= \left(\begin{array}{c|c} \lambda_1 & \mathbf{v}_1^T AW\mathbf{v}_2 \qquad \mathbf{v}_1^T AWY \\ \hline \mathbf{0} & \begin{array}{c|c} \lambda_2 & \lambda_2 \mathbf{v}_2^T Y \\ \hline \mathbf{0} & Y^T W^T AWY \end{array} \end{array}\right)
\end{aligned}
$$

The bottom right partition of this result $Y^T W^T AWY$ has eigenvalues $\lambda_3, \ldots, \lambda_n$

Using this knowledge, the same process can be repeated once again, using $\mathbf{v}_3$, the eigenvector of Y^TW^TAWY to create an orthonormal basis for $\mathbb{R}^{n-2}$. This matrix will be placed in the bottom right partition of an $n \times n$ matrix U_3, which will be filled in with 1's along the two blank diagonal entries and zeroes in all other blank entries. If this process is repeated n times, then the result will be in the form

$$U_n^T \dots U_1^T A U_1 \dots U_n.$$

Since all of the U_i matrices are orthonormal, then the results of the matrix multiplication $U_1 \dots U_n$ and $U_n^T \dots U_1^T$ are orthonormal. Furthermore, if we define $U = U_1 \dots U_n$, then $U^T = U_n^T \dots U_1^T$. Therefore, U is in the form that we desire, and the matrix multiplication U^TAU yields an upper triangular matrix with the eigenvalues of A in the diagonal entries. □

Proposition 2.6. *If A is symmetric then U^TAU is symmetric.*

Proof. To prove U^TAU is symmetric, we simply need to prove that $(U^TAU)^T = U^TAU$.

$$\begin{aligned} (U^TAU)^T &= U^TA^TU^{T^T} \\ &= U^TAU \end{aligned}$$

□

An easy corollary to the last two propositions is that any symmetric matrix A is diagnolizable, since a symmetric, upper triangular matrix will necessarily be diagonal.

In the next proposition we will use the preceding conclusions, which established a basis for $\mathbb{R}^n$ from the eigenvectors of A^TA, to derive a basis for the columnspace of A. Before we begin the next section, recall that the columnspace of an $p \times n$ matrix A (Col A) is defined as the subspace of $\mathbb{R}^p$ spanned by the columns of A. Similarly, the rowspace of A (Row A) is the subspace of $\mathbb{R}^n$ spanned by the rows of A.

Proposition 2.7. *Define $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ as an orthonormal basis for $\mathbb{R}^n$ consisting of eigenvectors of A^TA. These vectors also correspond to the eigenvalues of A^TA; $\lambda_1, \lambda_2, \ldots, \lambda_n$. Now, suppose that A^TA has r nonzero eigenvalues, then the set of vectors $\{A\mathbf{x}_1, A\mathbf{x}_2, \ldots, A\mathbf{x}_r\}$ is an orthogonal basis for the columnspace of A.*

Proof. For any $a \neq b$, $\mathbf{x}_a$ is orthogonal to $\mathbf{x}_b$.

$$(A\mathbf{x}_a) \cdot (A\mathbf{x}_b) = (A\mathbf{x}_a)^T(A\mathbf{x}_b) = \mathbf{x}_a^T A^T A\mathbf{x}_b = \mathbf{x}_a^T \lambda_b \mathbf{x}_b = \lambda_b \mathbf{x}_a \cdot \mathbf{x}_b = 0$$

Therefore, $\{A\mathbf{x}_1, A\mathbf{x}_2, \ldots, A\mathbf{x}_r\}$ is an orthogonal set. From Proposition 2.3 the eigenvalues of A^TA are equivalent to the lengths of the vectors squared. Since there are r nonzero eigenvalues, $A\mathbf{x}_a \neq 0$ if and only if $1 \leq a \leq r$, which means that $\{A\mathbf{x}_1, A\mathbf{x}_2, \ldots, A\mathbf{x}_r\}$ is a linearly independent set in Col A. Any vector $\mathbf{z}$ in Col A can be written as $\mathbf{z} = A\mathbf{v}$ and $\mathbf{v}$ can be written as a linear combination of $\mathbf{x}_1, \ldots \mathbf{x}_n$ with coefficients $c_1 \ldots c_n$.

$$\mathbf{z} = A\mathbf{v} = Ac_1\mathbf{x}_1 + \cdots + Ac_r\mathbf{x}_r + \cdots + Ac_n\mathbf{x}_n$$

$$= Ac_1\mathbf{x}_1 + \cdots + Ac_r\mathbf{x}_r + 0 + \cdots + 0$$

Therefore, $\mathbf{z}$ is in Span $\{A\mathbf{x}_1, A\mathbf{x}_2, \ldots, A\mathbf{x}_r\}$, which means that this set is a basis for Col A. □

Finally we are in a position to understand the reason why and how the Karhunen-Loéve method will work. In the interest of clarity, let's first consider a very simple example of what the following proposition is trying to establish. The goal of this example is to help us understand the steps involved. First, consider the matrix

$$Q = \begin{pmatrix} 2 & 3 & 5 & 6 \\ 5 & 7 & 8 & 10 \end{pmatrix}.$$

In this case, Col Q is simply $\mathbb{R}^2$, and the vectors that form the columns can actually be plotted, as in Figure 3.

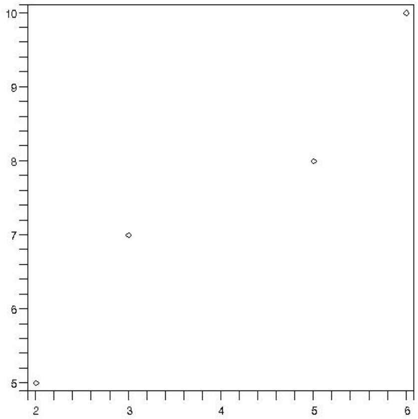

Figure 3: Vectors formed by the Columns of Q Plotted as Points in the Euclidean Plane

At this point, we can take the mean vector

$$M = \frac{1}{4}\begin{pmatrix} 2+3+5+6 \\ 5+7+8+10 \end{pmatrix} = \begin{pmatrix} 4 \\ 7.5 \end{pmatrix}$$

and subtract it from each of the columns of Q to obtain our new mean-subtracted matrix,

$$Qsub = \begin{pmatrix} -2 & -1 & 1 & 2 \\ -2.5 & -.5 & .5 & 2.5 \end{pmatrix}$$

the plot of which is now centered around the origin. Now we can calculate the small matrix

$$Qsub * Qsub^T = \begin{pmatrix} 10 & 11 \\ 11 & 13 \end{pmatrix}.$$

If we calculate the eigenvalues and corresponding eigenvectors, we get $\{22.6, .398\}$ for the eigenvalues and

$$\begin{pmatrix} .6576 \\ .7534 \end{pmatrix}, \begin{pmatrix} .7534 \\ -.6576 \end{pmatrix}$$

for the corresponding eigenvectors. If we view the graph of the space spanned by the first eigenvector corresponding to the larger eigenvalue, then we see it creates a nice"line of best fit" for the points in $Qsub$ (see Figure 4).

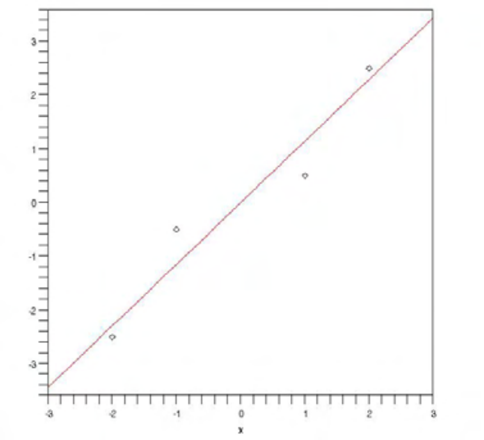

Figure 4: The Columns of $Qsub$ Plotted as Points with the One-Dimensional Subspace Spanned by the First Eigenvector of $Qsub * Qsub^T$

In fact, what Proposition 2.8 establishes is that this vector spans the best one-dimensional approximation of the data that we can find. For the purpose of maintaining consistency with the literature, we will define the matrix $Qsub * Qsub^T$ as the covariance matrix of $Qsub$, though $Qsub^T Qsub$ is also given the same title. Furthermore, in the proposition we contend that this method provides the "best" j-dimensional approximation of the rowspace of a matrix. While the word "best" may seem imprecise at first, it actually means that what we are trying to do is minimize the orthogonally projected distance. So in our previous example, if all of our four points were projected onto the one-dimensional subspace spanned by the vector $\begin{pmatrix} .6576 \\ .7534 \end{pmatrix}$, then the total of the distances that each point would have to travel to get to that line would be the smallest possible distance for any one-dimensional subspace.

Proposition 2.8. *Given any $p \times n$ matrix X, define its covariance matrix $C = \frac{1}{p}X^TX$. If the eigenvalues of C are $(\lambda_1, \ldots, \lambda_n)$, and the corresponding eigenvectors are $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$, such that each λ_i corresponds to $\mathbf{v}_i$. Then, the first j eigenvectors of C provide the best j-dimensional approximation of the rowspace of X iff $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$.*

Proof. Given a matrix X, we name the rows $\{\mathbf{x}_1, \ldots, \mathbf{x}_p\}$ and write

$$X = \begin{pmatrix} \\ \mathbf{x}_1 & \cdots & \mathbf{x}_p \\ \\ \end{pmatrix}^T .$$

Every $\mathbf{x}_i$ is a vector in $\mathbb{R}^n$. Additionally, define Φ as the matrix whose columns consist of the set of vectors $\{\phi_1, \ldots, \phi_n\}$ which form an orthonormal basis for $\mathbb{R}^n$. Therefore, any $\mathbf{x}_i$ is contained within the span of Φ and can be written as a linear combination of the columns:

$$\mathbf{x}_i = \sum_{k=1}^{n} \alpha_k \phi_k .$$

Since the $\phi' s$ form an orthonormal basis for $\mathbb{R}^n$, the coefficients α_k are easy to calculate;

$$\alpha_k = \mathbf{x}_i \cdot \phi_k = \langle \mathbf{x}_i, \phi_k \rangle.$$

Although the matrix Φ provides the full orthonormal basis for $\mathbb{R}^n$, in our application we will not be using the full basis. If we use, for example, only the first j vectors in Φ, then it is possible to split up the summation and write it as

$$\mathbf{x}_i = \sum_{k=1}^{j} \alpha_k \phi_k + \sum_{k=j+1}^{n} \alpha_k \phi_k.$$

When it is written this way, the first term $\left(\sum_{k=1}^{j} \alpha_k \phi_k\right)$ is the j-dimensional approximation of $\mathbf{x}_i$ and the second term can be thought of as the error. For notation purposes, define

$$\mathbf{x}_i^{(err)} = \sum_{k=j+1}^{n} \alpha_k \phi_k.$$

At this point we can also define the total error of the orthogonal projection of X onto the linear subspace of $\mathbb{R}^n$ spanned by the first j vectors of Φ.

$$\begin{aligned} \text{Error} &= \frac{1}{p} \sum_{t=1}^{p} \parallel \mathbf{x}_t^{(err)} \parallel^2 \\ &= \frac{1}{p} \sum_{t=1}^{p} \parallel \left(\sum_{k=j+1}^{n} \alpha_k \phi_k \right) \left(\sum_{k=j+1}^{n} \alpha_k \phi_k \right) \parallel \\ &= \frac{1}{p} \sum_{t=1}^{p} \parallel (\alpha_{j+1}\phi_{j+1} + \cdots + \alpha_n \phi_n)(\alpha_{j+1}\phi_{j+1} + \cdots + \alpha_n \phi_n) \parallel \end{aligned}$$

Since $\phi_m \cdot \phi_n = 0$ for all $m \neq n$ and $\phi_m \cdot \phi_m = 1$ for all m, the previous equation can be simplified.

$$
\begin{aligned}
\text{Error} &= \frac{1}{p}\sum_{t=1}^{p} \parallel \left(\alpha_{j+1}^2 + \cdots + \alpha_n^2\right) \parallel \\
&= \frac{1}{p}\sum_{t=1}^{p} \left(\sum_{k=j+1}^{n} \left(\alpha_k\right)^2\right) \\
&= \frac{1}{p}\sum_{t=1}^{p} \left(\sum_{k=j+1}^{n} \left(\mathbf{x}_t \cdot \phi_k\right)^2\right) \\
&= \frac{1}{p}\sum_{t=1}^{p} \left(\sum_{k=j+1}^{n} \left(\mathbf{x}_t \cdot \phi_k\right)\left(\mathbf{x}_t \cdot \phi_k\right)\right) \\
&= \frac{1}{p}\sum_{t=1}^{p} \left(\sum_{k=j+1}^{n} \left(\phi_k \cdot \mathbf{x}_t\right)\left(\mathbf{x}_t \cdot \phi_k\right)\right) \\
&= \frac{1}{p}\sum_{t=1}^{p} \left(\sum_{k=j+1}^{n} \phi_k^T \mathbf{x}_t \mathbf{x}_t^T \phi_k\right)
\end{aligned}
$$

Since both of the summation functions are finite sums, it is possible to commute the order.

$$
\begin{aligned}
\text{Error} &= \sum_{k=j+1}^{n} \left(\frac{1}{p}\sum_{t=1}^{p} \phi_k^T \mathbf{x}_t \mathbf{x}_t^T \phi_k\right) \\
&= \sum_{k=j+1}^{n} \left(\frac{1}{p}\left(\phi_k^T \mathbf{x}_1 \mathbf{x}_1^T \phi_k + \cdots + \phi_k^T \mathbf{x}_p \mathbf{x}_p^T \phi_k\right)\right) \\
&= \sum_{k=j+1}^{n} \left(\phi_k^T \left(\frac{1}{p}(\mathbf{x}_1 \mathbf{x}_1^T + \cdots + \mathbf{x}_p \mathbf{x}_p^T)\right) \phi_k\right) \\
&= \sum_{k=j+1}^{n} \langle \phi_k, \left(\frac{1}{p}\sum_{t=1}^{p} \mathbf{x}_t \mathbf{x}_t^T\right) \phi_k \rangle
\end{aligned}
$$

In order to proceed at this point, it is necessary to define some new notation. $\psi_t^{(a)}$ is equal to the a^{th} entry in the vector $\mathbf{x}_t$. Therefore,

$$\begin{aligned}
\sum_{t=1}^{p} \mathbf{x}_t \mathbf{x}_t^T &= \begin{pmatrix} \psi_1^{(1)}\psi_1^{(1)} & \cdots & \psi_1^{(1)}\psi_1^{(n)} \\ \vdots & \ddots & \vdots \\ \psi_1^{(n)}\psi_1^{(1)} & \cdots & \psi_1^{(n)}\psi_1^{(n)} \end{pmatrix} + \cdots + \begin{pmatrix} \psi_p^{(1)}\psi_p^{(1)} & \cdots & \psi_p^{(1)}\psi_p^{(n)} \\ \vdots & \ddots & \vdots \\ \psi_p^{(n)}\psi_p^{(1)} & \cdots & \psi_p^{(n)}\psi_p^{(n)} \end{pmatrix} \\
&= \begin{pmatrix} \psi_1^{(1)}\psi_1^{(1)}+\ldots+\psi_p^{(1)}\psi_p^{(1)} & \cdots & \psi_1^{(1)}\psi_1^{(n)}+\ldots+\psi_p^{(1)}\psi_p^{(n)} \\ \vdots & \ddots & \vdots \\ \psi_1^{(n)}\psi_1^{(1)}+\ldots+\psi_p^{(n)}\psi_p^{(1)} & \cdots & \psi_1^{(n)}\psi_1^{(n)}+\ldots+\psi_p^{(n)}\psi_p^{(n)} \end{pmatrix} \\
&= \begin{pmatrix} \\ \mathbf{x}_1 & \cdots & \mathbf{x}_p \\ \\ \end{pmatrix} \begin{pmatrix} \\ \mathbf{x}_1 & \cdots & \mathbf{x}_p \\ \\ \end{pmatrix}^T \\
&= X^T X.
\end{aligned}$$

Substituting this into the equation for the error yields

$$\begin{aligned}
\text{Error} &= \sum_{k=j+1}^{n} \langle \phi_k, \left(\frac{1}{p} \sum_{t=1}^{p} \mathbf{x}_t \mathbf{x}_t^T \right) \phi_k \rangle \\
&= \sum_{k=j+1}^{n} \langle \phi_k, \left(\frac{1}{p} X^T X \right) \phi_k \rangle \\
&= \sum_{k=j+1}^{n} \langle \phi_k, C\phi_k \rangle.
\end{aligned}$$

This provides a simple way to look at the error produced by the j-dimensional approximation of X. Now let's just assume for a moment that $j = 0$. In this case, the error can be expressed as

$$\text{Error} = \frac{1}{p} \sum_{t=1}^{p} \|\mathbf{x}_t^{(err)}\|^2$$

and $\mathbf{x}_t^{(err)}$ can be expressed as

$$\mathbf{x}_t^{(err)} = \sum_{k=1}^{n} \alpha_k \phi_k = \mathbf{x}_t.$$

Since $\mathbf{x}_t^{(err)} = \mathbf{x}_t$ when $j = 0$, we can substitute into the previous equation to obtain

$$\text{Error} = \frac{1}{p} \sum_{t=1}^{p} \|\mathbf{x}_t\|^2 = \text{Constant}.$$

This is the same as saying

$$\sum_{k=1}^{n} \langle \phi_k, C\phi_k \rangle$$

is also a constant and can be split up in the following manner:

$$\langle \phi_1, C\phi_1 \rangle + \sum_{k=2}^{n} \langle \phi_k, C\phi_k \rangle.$$

Notice now that the second term ($\sum_{k=2}^{n} \langle \phi_k, C\phi_k \rangle$) is equal to the error for the 1-dimensional approximation of X, so minimizing the error in this case is equivalent to maximizing

$$\langle \phi_1, C\phi_1 \rangle.$$

Recall that the set of eigenvectors of the covariance matrix form an orthogonal basis for $\mathbb{R}^n$ (which we will assume has been normalized). If we define V as

$$V = \begin{pmatrix} \\ \mathbf{v}_1 & \cdots & \mathbf{v}_n \\ \\ \end{pmatrix}$$

then any of the columns of Φ can be written as a linear combination of the columns of V. In other words

$$\phi_i = \gamma_1 \mathbf{v}_1 + \cdots + \gamma_n \mathbf{v}_n = V\Gamma$$

for

$$\Gamma = \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_n \end{pmatrix}.$$

Furthermore, if

$$\Lambda = \begin{pmatrix} \lambda_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \lambda_n \end{pmatrix}$$

$$CV = \begin{pmatrix} \\ \lambda_1 \mathbf{v}_1 & \cdots & \lambda_n \mathbf{v}_n \\ \\ \end{pmatrix} = V\Lambda.$$

Putting these two equalities together and applying it to our problem allows us to put the inner product in a different form:

$$\begin{aligned} \langle \phi_1, C\phi_1 \rangle &= \phi^T C \phi \\ &= \Gamma^T V^T C V \Gamma \\ &= \Gamma^T V^T V \Lambda \Gamma \\ &= \Gamma^T \Lambda \Gamma \\ &= \begin{pmatrix} \gamma_1 & \cdots & \gamma_n \end{pmatrix} \begin{pmatrix} \lambda_1 \gamma_1 \\ \vdots \\ \lambda_n \gamma_n \end{pmatrix} \\ &= \lambda_1 \gamma_1^2 + \cdots + \lambda_n \gamma_n^2. \end{aligned}$$

At this point, it is relatively easy to prove that

$$\langle \phi_1, C\phi_1 \rangle = \lambda_1 \gamma_1^2 + \cdots + \lambda_n \gamma_n^2 \leq \lambda_1.$$

First we show that, since ϕ_1 is a normal vector

$$\begin{aligned} \langle \phi_1, \phi_1 \rangle &= \|\phi_1\|^2 \\ &= \phi_1^T \phi_1 \\ &= \Gamma^T V^T V \Gamma \\ &= \Gamma^T \Gamma \\ &= \gamma_1^2 + \cdots + \gamma_n^2. \end{aligned}$$

Since this sum is equal to $\|\phi_1\|^2 = 1$, we can simply multiply it into our previous equation to get

$$\lambda_1 \gamma_1^2 + \cdots + \lambda_n \gamma_n^2 \leq \lambda_1(\gamma_1^2 + \cdots + \gamma_n^2).$$

Since the eigenvalues are organized such that $\lambda_1 \geq \ldots \lambda_n$, the first term on the left and right sides of the equation will be equal, and each of the subsequent terms on the left side of the equation will always be less than or equal to its corresponding term on the right side since all $\gamma_i^2 \geq 0$. This proves that

$$\langle \phi_1, C\phi_1 \rangle \leq \lambda_1,$$

and since we want to maximize the left side of the equation, then

$$\langle \phi_1, C\phi_1 \rangle = \lambda_1.$$

This result is easily obtained by setting ϕ_1 equal to $\mathbf{v}_1$. Finally we can conclude that the best 1-dimensional approximation of the rowspace of X is the eigenvector corresponding to the largest eigenvalue of the covariance matrix of X. This exact same argument can be applied to higher dimensional approximations, so that we can conclude that the best j-dimensional approximation of the rowspace of X is given by the first j sorted eigenvectors of the covariance matrix. □

We now have a good way to find the best basis for the rowspace of X, however, it requires calculating C, which is a $n \times n$ matrix. If n is significantly larger than p (as it will be in our application) and you still want to find the best j-dimensional approximation of the rowspace without calculating the large matrix C, it is possible to approach the problem from a slightly different angle. Consider the (not quite finished) SVD of the matrix X,

$$X = U\Sigma V^T,$$

where Σ is $k \times k$ and k is the rank of X. By definition, we know that V is calculated by taking the eigenvalues of the covariance matrix of X. Comparing this to Proposition 2.8, we see that this is also the best basis for the rowspace of X. In other words, the best approximation for the rowspace of X is given by the right singular vectors of X, which are the columns of V. Now consider;

$$X^T = V\Sigma U^T.$$

In this case, the covariance matrix of X^T is defined as $\frac{1}{n}XX^T$, which is the much smaller $p \times p$ matrix. Therefore, returning to the definition of the SVD, the columns of U are the eigenvectors of the covariance matrix of X^T, which also means that the best approximation for the rowspace of X^T is equal to the left singular vectors of X. Furthermore, since the rows of X^T are the same as the columns of X, we can conclude that the best basis for the columnspace of X is equal to the left singular vectors of X.

This is a very useful trick when you consider that the matrices U and V are related by the original equation

$$\begin{aligned} X &= U\Sigma V^T \\ XV &= U\Sigma \\ V\Sigma &= X^T U. \end{aligned}$$

We can now consider the case where the i^{th} column of U and V are given by u_i and v_i, respectively, and σ_i is the i^{th} diagonal entry of Σ. We can substitute these into the above equation to yield

$$\begin{aligned} v_i\sigma_i &= X^T u_i \\ v_i &= \frac{1}{\sigma_i} X^T u_i. \end{aligned}$$

This gives us an easy way to calculate the columns of V without having to calculate the large $n \times n$ matrix C. Thus, this provides for a much more computationally feasible method for this large $n \times n$ matrix.

Now that we have examined the principle behind the partial SVD of the database we can begin to discuss the applications of the mathematical theory. While many of these propositions can be found in an excellent linear algebra textbook such as [15] and much is reprinted from the article by Giordano and Uhrig [7], the importance of discussing them here cannot be understated. Often, those wishing to use the applications that originate from mathematical principles gloss over the basic ideas behind the system, but this should not be performed. In order to understand the system as a whole and any inherent flaws, an examination of the mathematical ideas or computer science principles must be understood. With this understanding in mind we can now progress into our first application, human face recognition.

3

Face Recognition Programming Using MATLAB

In order to apply the mathematical processes to face recognition, a method to express the database of faces as a matrix is needed. The computer program MATLAB can be used to import images from a text file list of filenames and then convert each grayscale image into a $h \times w$ matrix, where w is the width (in pixels) of the image and h is the height. Once each image is imported and converted to the proper MATLAB format, it is then reshaped into a vector that is $(w * h) \times 1$, simply by placing the first column on the top and each successive column below its predecessor. Each vector now corresponds to a picture, and these vectors become the columns of a new matrix A, which is $(w * h) \times p$, where p is the total number of images used. MATLAB is then used to calculate the mean vector of the columns of A and subtracts this mean vector M from each of the columns in A to create the mean-subtracted matrix L. For a detailed explanation of this process, including the MATLAB code that we used, see Appendix A. It should be noted that Excel could be used to perform many of these calculations, yet the use of MATLAB simplifies much of the process. This simplification begins with the fact that images are essentially stored as matrices within the computer, and MATLAB is made to easily handle matrices. Many of the required calculations are built-in commands within the MATLAB environment. Yet, MATLAB is suited more for a programmer or a statistician, whereas Excel has wide familiarity within the entire scientific community. If Excel must be used, many of the built-in commands for MATLAB would have to be written as macros for Excel or found from a website accompanying [9].

We can now perform a simplified version of SVD calculations. The initial step is to calculate the covariance matrix $G = A^T A$. At this point the eigenvalues and their corresponding eigenvectors are calculated and sorted. They are sorted in descending order according to the size of the eigenvalues and the corresponding eigenvectors are placed in the same order. These eigenvectors become the columns of the $p \times p$ matrix V. In order to create a set of basis vectors for the columnspace of L the matrix U is created such that $U = L * V$. The span of the first column of U provides the best one-dimensional approximation of the column space of L. Similarly, the span of the first two columns provides the best two-dimensional approximation, and this line of thinking can be continued to obtain the best desired $j-\text{dimensional}$ approximation. It is then possible to calculate what percentage of the variance is used by the $j-\text{dimensional}$ approximation. This percentage (Per) is calculated using the sorted eigenvalues of the covariance matrix $\{\lambda_1 \ldots \lambda_n\}$ in the equation

$$Per = \frac{\sum\limits_{i=1}^{j} \lambda_i}{\sum\limits_{i=1}^{n} \lambda_i} * 100\%.$$

In order to test an image from outside the database, MATLAB imports and reshapes the image in the same manner as that used for the images of the database. The mean vector M is then subtracted from the test image vector (T) and the coefficients of the first j bases are calculated using the formula

$$C_t = \frac{U_t \cdot T}{U_t \cdot U_t}.$$

Where C_t is the t^{th} coefficient for T and U_t is the t^{th} column of U. From here, MATLAB simply calculates the Euclidean distance between the first j coefficients of T and the first j coefficients of all the pictures in the database. Once these distances are sorted (in ascending order) it allows for us to determine which image is the closest approximation of the test image. From that point onward, it is up to interpretation by human eyes to determine whether or not a match truly exists. The MATLAB code for testing an image from outside the database can be found in Appendix B.

3.1 First Database

The first database that we tested had 130 pictures of 65 different people. Each picture was 400×500 pixels and none of the pictures contained individuals wearing glasses. We chose to use images of people without their glasses to eliminate a "glasses" variable, which would potentially limit the accuracy of the system. We used 15 of the 130 basis vectors in the computations, which retained 81.229% of the variance. The computer took 90 minutes to load all 130 images into its memory and perform the required calculations. One output that is initially interesting to examine is the average face. This is simply the mean vector of the columns of A reshaped into a 400×500 picture. This yielded the image in Figure 5.

Another image that is output by MATLAB is informally called the eigenface. Mathematically, however, this image is one of the basis vectors that is reshaped into a 400×500 (the dimension of the image is contingent on the size of the original images in the database)

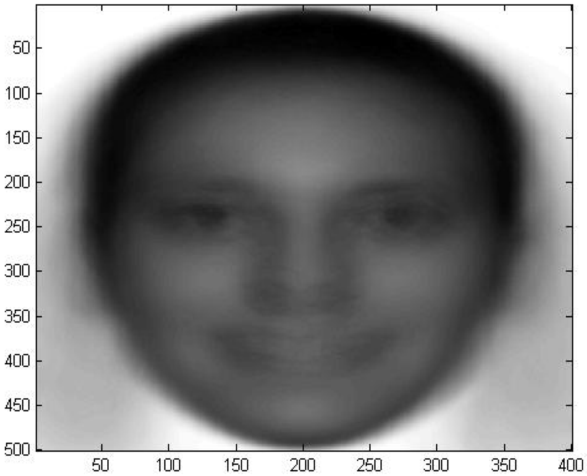

Figure 5: Average Face from Database 1

image. The following figure is the first basis vector (first eigenface) for the database.

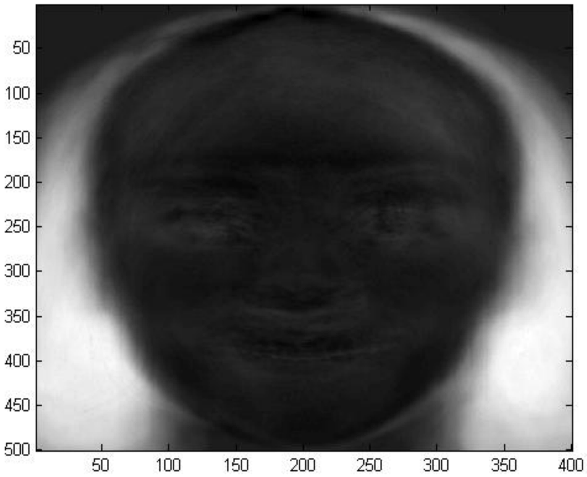

Figure 6: First Eigenface from Database 1

Each picture in the database, as has already been stated, has a coefficient that corresponds to this particular basis vector. Graphing the coefficients of this basis vector in terms of gender is very revealing to what facial feature this vector accentuates, as can be seen in Figure 7.

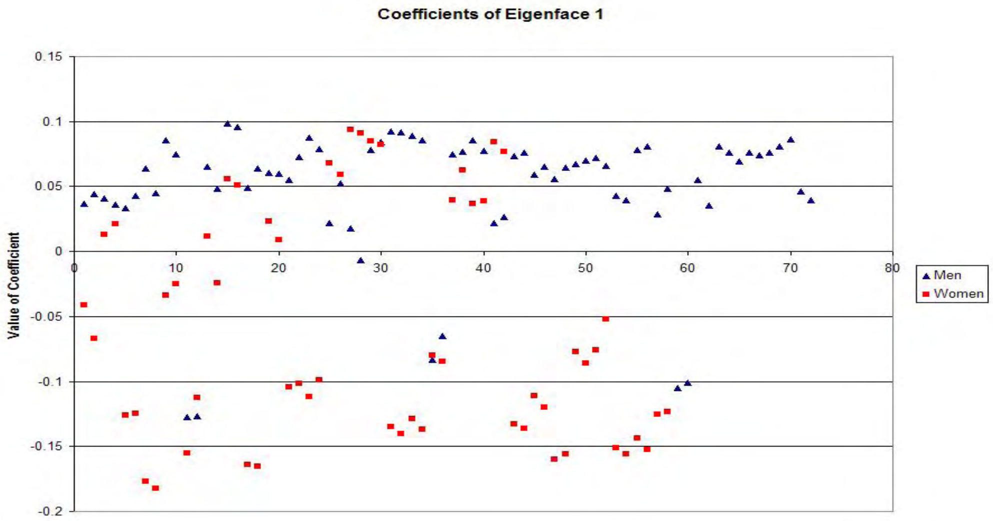

Figure 7: Coefficients of Eigenface 1 for each Face based on Gender

Most women have negative coefficients and the large majority of the males have positive coefficients. Furthermore, when we went back and examined the pictures of men with negative coefficients we noted that they all had long hair. Also, when the images of females were examined, those with a negative coefficient had long hair, while those with a more positive coefficient had shorter hair length. This fact is slightly troublesome because it indicates that the most important aspect of a person's face was the length/amount of hair, which is also the easiest feature of one's face to change. This fact could potentially be a weakness for the system since any major change in hair length or shape would likely cause failure in the system.

When examining the second eigenface it was less clear which features the system was identifying as important. We hypothesized that one of the possible features the eigenface may be identifying is the difference between a person's hair color and the darkness/lightness of a person's skin tone. We arrived at this conclusion by analysis similar to that used for the previous eigenface. We found it relatively difficult to determine which features were being distinguished in the eigenfaces that followed. Clearly, the program was differentiating between some features, but it was a failure of human perception to determine what those features were.

The next step was to take the first fifteen coefficients of each face and place them in a vector. The formula for the Euclidean length of a vector was used to determine the length of the vector. This allowed us to discover which face had the overall largest magnitude. Thus, the vector with the highest magnitude corresponded to the face that can be best approximated. Figure 8 shows the original face that could be best approximated (had the largest coefficient vector length) and the fifteen vector approximation of this face. This approximated image was obtained by taking the first fifteen vectors multiplying by their corresponding coefficients and adding the resulting vectors together. It is clear from these two images that using only 15 of the 130 basis vectors produces a readily recognizable approximation. In contrast, the face with the worst approximation is also shown in Figure 8.

It is now possible to look at the actual face recognition applications. As has already been stated, this was performed by taking the Euclidean distance between the coefficients of the orthogonal projection of the outside image with the coefficients of each image in the database. The image that had the smallest distance from the test image was identified by the program as being the closest match, and all the remaining images were ranked from closest to farthest away (in terms of Euclidean distance) from the test image. The results from this series of tests were very positive and indicated that the system worked well overall. Yet, it became clear very quickly that consistency in the orientation of the faces was needed so that proper recognition can occur. When all of the faces were facing forward, with minimal head tilt, the system identified the correct person as being the closest match 83.78% of the time. Furthermore, 97.30% of the trials identified at least one correct match in the top three closest matches. It is important to note that many of

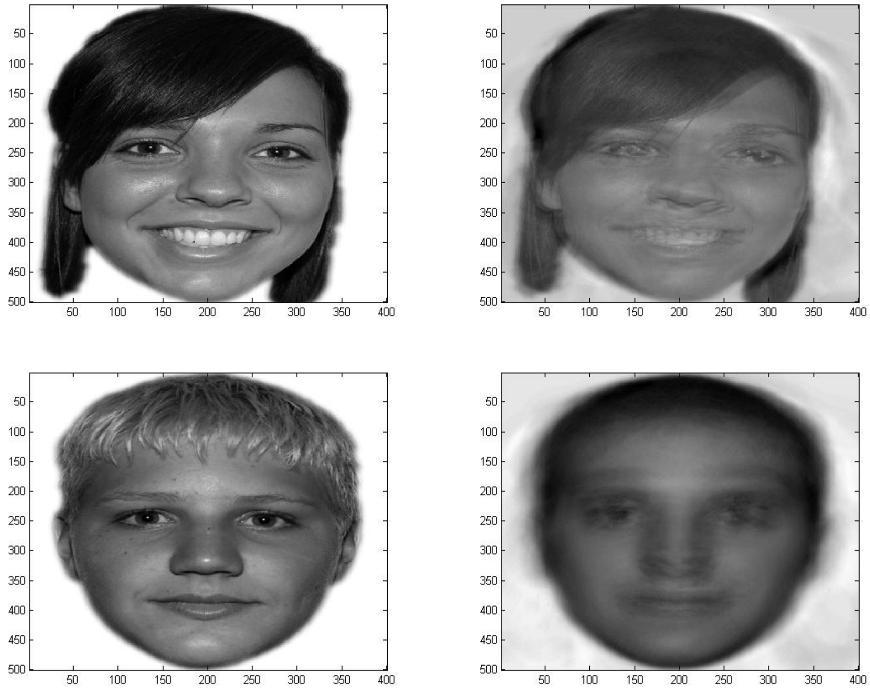

Figure 8: The top two are the original face and the fifteen vector approximation of the face that can be best approximated. Similarly, the bottom two are the original face and the fifteen vector approximation of the face that is most inaccurately approximated.

the test images that were used had individuals wearing glasses. We determined that the presence of glasses had little to no influence on the outcome of the results. On the other hand, facial expression did have an impact on the results. This was indicated by several trials where the person had a significantly contorted facial expression compared to the expression in the database. In these cases the system had a more difficult time identifying the proper faces as being a match. We also noticed that several of the failed trials most likely resulted from the tilt of the person's head. Thus, tilt and orientation of the person's face were the two most significant factors that negatively affected the outcome.

Since we noticed that the first eigenface identified long hair as being one of the most significant factors in making a correct match, we tested how well it would match a person when significant changes to their hair occurred. First, we used Photoshop to take one of the female subjects with black hair and change the color to blonde. Despite the color change, the system was still able to correctly identify her in the database (little to no influence on results occurred). Next, we took several of the pictures with long hair (people who had a large value for the first eigenface) and used Photoshop to remove a substantial portion of their hair. We found that the system was unable to recognize the correct person as the best match when large changes to hair occurred.

Another interesting test that we conducted involved placing a set of twins in the database. We tested two images of each person and in each case the system picked out both pictures in the database as being one of the top 3 matches. Interestingly enough, the other brother never appeared in the top three closest matches. This is especially important since even to the human eye, it is particularly difficult to differentiate between the two brothers as can be demonstrated by Figure 9.

3.2 Second Database

We thought at this point that it would be important to understand how well another database would work by using smaller image files (fewer pixel dimensions). The use of such downsized images is desirable because the amount of computer processing time greatly decreased with smaller image files. Thus, we used the same pictures, but reduced the size to 350×438 pixels. This decreased the processing time to approximately nine minutes, which is a tenth of the previous database's processing time. Again, we used 15 basis vectors, which retained 70.90% of the variance. As in the previous database, none of the subjects had glasses. However, we did use Photoshop to remove long hair around the neck and lower part of the face, which we hypothesized would reduce the overall amount of influence hair has in the system. Consequently, the average face for

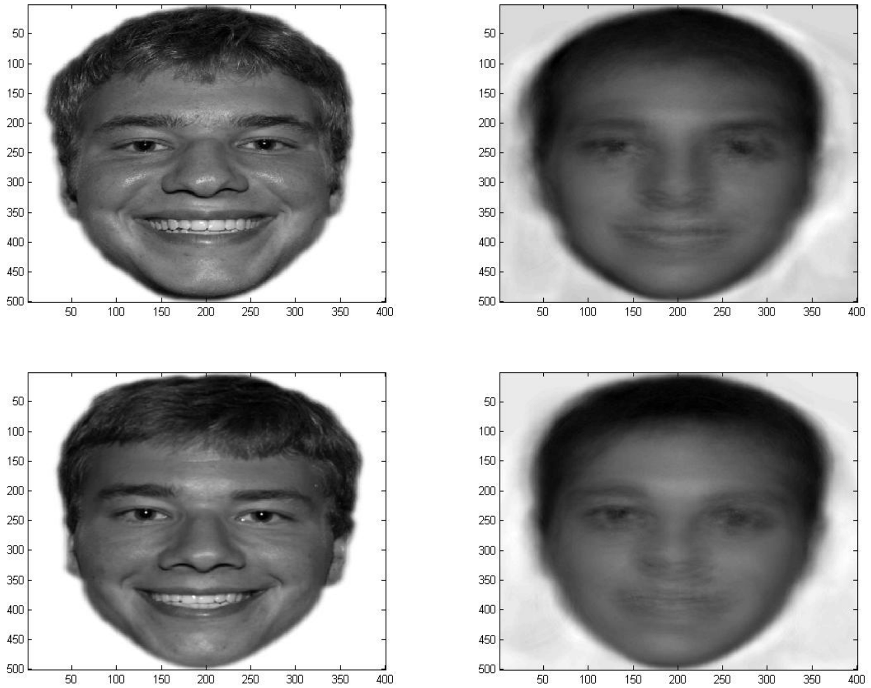

Figure 9: The top two images are the original face and the fifteen vector approximation of one of the brothers. The bottom two are the original face and the fifteen vector approximation of the other brother.

this database looked essentially the same, but the outline of hair around the lower half of the face is removed (see Figure 10). The lack of hair greatly influenced the outcome of the first eigenface, as can also be seen in Figure 10.

In this eigenface, there appears to be a contrast between a person's facial color and the color of their hair, as opposed to the first eigenface from the previous database, which focused on whether or not a person had long hair. It became difficult to determine which aspects of the faces were being differentiated in the subsequent eigenfaces. Yet, the second eigenface seemed to be detecting differences in skin tone, and the third eigenface appeared to be looking at how much hair was at the top of a person's head.

We noticed, again, that this system was very sensitive to the rotation of a person's head. Consequently, we decided to test how much tilt could be present before the system

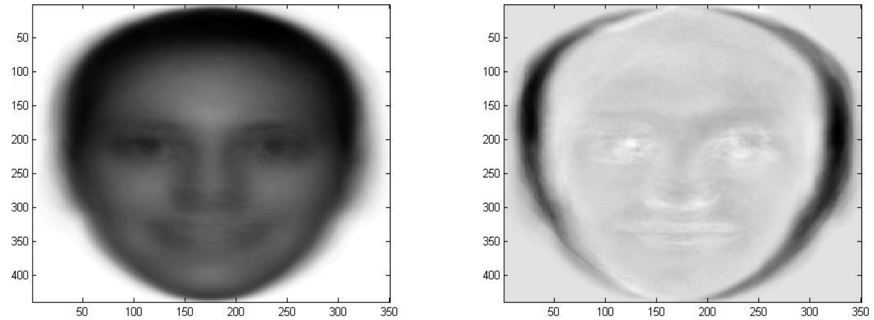

Figure 10: The image on the left is the average face from the second database, and the image on the right is the first eigenface from this database.

started to fail. We began with a different picture of a person who was already in the database and rotated it 25°. When this rotation was used the correct face was the 51^{st} closest match. We then made the rotation progressively closer to center in 5° increments. When the face was 20° from center the closest matching face was number 34. When there was a 15° offset from center, the closest match appeared in the 11^{th} position. When the rotation was 10°, the fifth match was correct. Once the rotation was both 5° and straight up and down, the first match was the correct face.

When we tested images from outside the database we also removed the long hair using the same method to create the database. This experiment yielded a 70.00% success rate. There are several factors that contributed to the decrease in success. The most likely scenario is that the removal of the hair in Photoshop was not performed in a completely controlled/automated manner. Instead the human eye was used to judge how much excess hair needed to be removed. Another possible factor for the decreased success rate originates from the fact that smaller image files were used. These smaller images resulted in less of the variance being retained (compared to the first database) even though the same number of basis vectors were used as in the first database. In order to correctly determine which factor was most influential we needed to run more tests, which will be discussed shortly. We also tested images with the long hair left in the picture and the success rate was 30.00%.

The next test we ran was with people who already had short hair. This produced a 93.75% success rate, in that at least one of the correct images from the database was chosen as one of the top three matches. Furthermore, 83.38% were selected as the number one closest match. These results indicate that in the previous consideration of whether or not human error or the decreased variance contributed most to the smaller success rate, human error was the largest contributing factor. We drew this conclusion on the indication that in the test where human error was not a factor the success rate only decreased slightly from that of the first database.

3.3 Conclusion

The technique is extremely successful and requires a minimal amount of data input. Although it is extremely important for all of the faces to be oriented in the same manner once this is achieved, the technique works very well. Thus, if the images are taken in a very controlled way, such as is done with police mug shots, the database would prove to be extremely useful. Although we decreased the size of the images in the second database by over 23% and the processing time by approximately 90% we were still able to obtain a success rate comparable to that of the first database.

Further research in this area could focus on testing a larger number of variables in the pictures. Another great consideration would be further analysis regarding the most effective method to use for the preprocessing of images that comprise the database. We would also like to pursue greater automation of the entire system so that more tests with a greater number of images can be performed in an efficient manner. Yet, this makes us wonder if variables such as lighting, orientation of the head, or hair are not a factor of how well the system can work.

This will now bring us to the next recognition database involving infrared (IR) spectroscopic data. Since the data obtained from the IR spectrophotometer will not have the variations that are inherent to image files it seems likely that it would work much more reliably as a whole. Yet, before we can examine this database we must first understand in greater detail the principles of chemometrics and how IR spectroscopy functions. Since Fourier transform IR (FTIR) is one of the most common forms of IR spectroscopy, it will again be important to understand the mathematics behind this system to some extent in order to discover any potential problem areas that could arise when the data output by the instrument is placed within the recognition program.

4

Introduction to Infrared Spectroscopy

4.1 Theory of Infrared Absorption Spectroscopy

The very first infrared spectrum was obtained in 1840 by Sir John F. W. Herschel, who was also the son of Sir William Herschel, the original discoverer of infrared radiation [18, p. 7, 80]. After years of study, it was determined that the infrared region of the electromagnetic spectrum ranges in terms of wavenumbers ($\tilde{\nu}$) from 12,800 to 10 cm^{-1}, or in terms of wavelengths from 0.78$\times 10^{-6}$ to 1$\times 10^{-3}$ meters [17, p. 380]. We should note at this point that other sources often call wavenumbers *frequencies* (ν), where $\nu = \frac{c}{\lambda}$, λ is the wavelength and c is the speed of light) [22]. The infrared portion of the spectrum is divided into near, middle, and far regions[18, p. 80]. Of these aforementioned regions, the middle infrared spectral region ranging from 4000 to 200 cm^{-1} is the most commonly used for the purposes of IR spectroscopy [17]. This region is useful for the elucidation of chemical structure, particularly the presence of certain functional groups within a molecule [19, p. 559]. Essentially, a functional group is the atom or group of atoms in a molecule that are responsible for the characteristic reactions of a compound [34]. These functional groups appear in specific regions of an infrared spectrum. Table 1 details the infrared absorption regions of some common functional groups.

Before examining the instrument used to produce the IR spectra of compounds, it is important to understand the underlying principles of this method. Figure 11 shows a

Functional Group	Wavenumbers (cm^{-1})
alcohols	3200-3600
alcohol of a carboxylic acid	2500-3600
carboxylic acids	2500-3600
secondary amines	3350-3500
C=C	1620-1680
carbonyl	1710-1750
ether	1025-1200
alkanes	2850-2950
monosubstituted benzene	730-770 and 690-710
esters	1730-1750
amides	1680-1700

Table 1: Infrared absorption region of common functional groups in terms of wavenumbers cm^{-1}. This table has been adapted from information found in [19, p. 561].

typical infrared spectra of polystyrene with $\tilde{\nu}$ on the x-axis and % transmittance on the y-axis. Figure 12 shows the structure of this polymer.

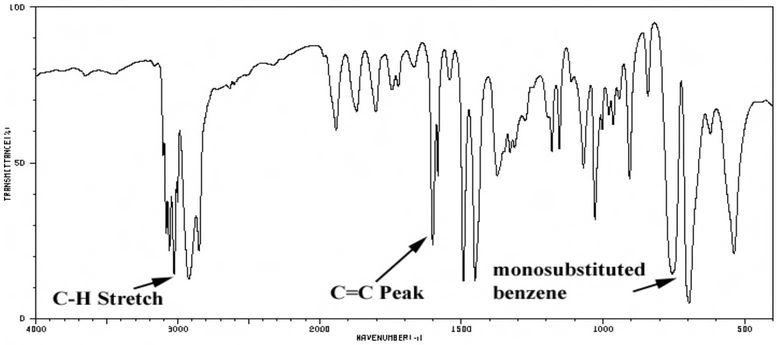

Figure 11: Infrared absorption spectrum of polystyrene, note the use of percent transmittance rather than absorption (as in other spectroscopic applications) on the y-axis [20]. Note the labeled regions of the spectrum correspond to several of the important functional groups found within polystyrene. The monosubstituted benzene portion of the spectrum indicates that benzene has only one substitutent on the ring and this occurs in the "fingerprint" region of the spectrum.

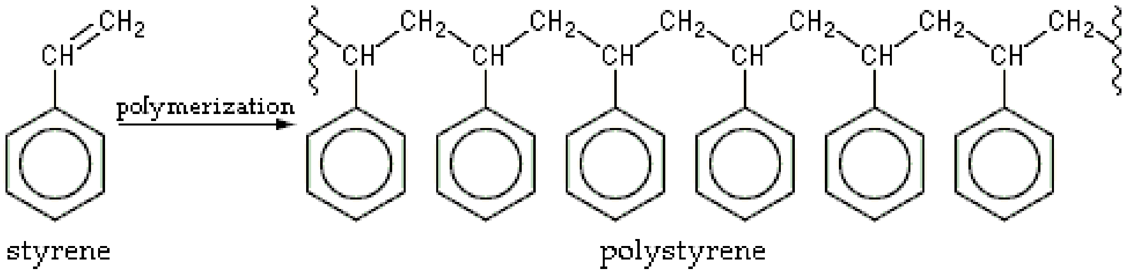

Figure 12: Structure of the monomer styrene and the polymerized form polystyrene, whose spectra is shown in Figure 11.

4.1.1 Causes of Absorption in the Infrared Region

In order for a spectrum such as the one above to be produced, there is a necessary dipole change that must occur within the molecule. We should note that a dipole is essentially a pair of separated opposite electric charges. In a polyatomic molecule, the dipole of the entire molecule is the vector sum of the dipole found within individual bonds [34]. This does not mean that only molecules with permanent dipoles will produce an IR spectra. Rather, this change in dipole can originate from the molecular vibrational or rotational motion of the molecule [21, p.684]. To this end, we can examine a molecule of carbon dioxide. The structure of carbon dioxide is given in Figure 13. This molecule is perfectly symmetric, and consequently has no overall dipole moment. Yet, bending vibrations and antisymmetric vibrations present within the molecule can give rise to an oscillating dipole moment, and consequently produce an infrared spectrum [21, p. 684]. Figure 14 shows several of the motions that can occur within a molecule to produce changes in the dipole moment. Due to these changes in dipole moment within the molecule, IR spectra are often

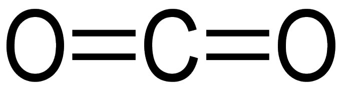

Figure 13: Structure of carbon dioxide.

referred to as *vibrational-rotational spectra* [22]. It is important to note that homonuclear species such as O_2, N_2, and H_2 do not see a net change in dipole, even when vibrations and rotations within the molecule are considered. Consequently, these molecules do not absorb in the infrared region. Such molecules are more the exception, since the vast majority of molecular species do absorb within the infrared range [22, p. 52-53].

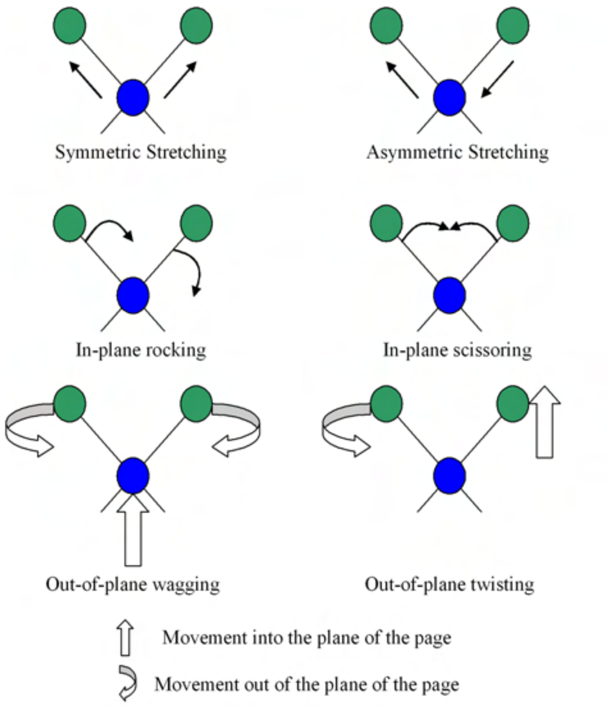

Figure 14: Types of molecular vibrations that can occur within a molecule.

4.1.2 A Model for Understanding Infrared Vibrations

Since we now understand the importance of a change in dipole moment for the production of an IR spectrum, we can further examine the molecular vibrations of a molecule. The simplest method by which to conduct such an examination is to reduce the molecule to a two-body problem and imagine atoms as balls and the bonds connecting the atoms as massless springs. This model, which is shown in Figure 15, is the ball-and-spring model for such a two-body system, which could be either a diatomic molecule, or two bonded atoms in a larger molecule. In this case, we have assumed that the mass of atom **A** is m_1 and the mass of atom **B** is m_2, and the "spring" has a restoring force of F, which is proportional to the displacement x of the atoms from the equilibrium position [23, p. 27]. Thus, F is given by Hooke's law, which is

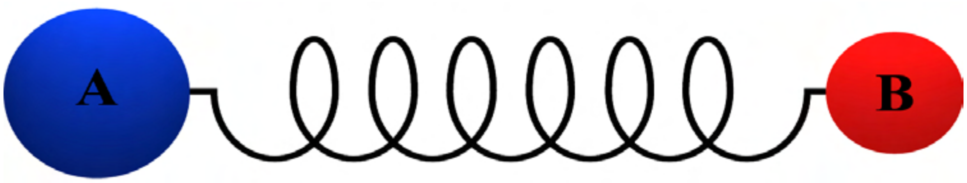

Figure 15: The ball and spring model for a diatomic molecule consisting of atoms **A** and **B**. The spring has a force constant of k.

$$F = -kx$$

where k is the force constant of the spring in $\frac{\mathrm{N}}{\mathrm{m}}$. The value produced essentially tells us the overall strength of the bond between the two atoms. Finding the value of ν_c (the classical vibrational frequency) is relatively easy through the use of the classical mechanics principle of the harmonic oscillator, which is given by

$$\nu_c = \frac{1}{2\pi}\sqrt{\frac{k}{m}} = \frac{1}{2\pi}\sqrt{\frac{k(m_1 + m_2)}{m_1 m_2}}$$

where m is the reduced mass of the two atoms in kg. Consequently, only the force constant and the reduced mass of the bonded atoms have any influence on the resulting frequency. Therefore, this method can be used to aid in determining where stretching bands of similar two-body systems might occur within the spectrum. For instance, it would be possible to compare a C-H bond with a C-D bond (where D stands for deuterium, which is heavy hydrogen and is represented by ^{2}H) since these two diatomic systems have nearly identical force constants. In this case, the value for the reduced mass in the equation to find frequency changes from $\sqrt{\frac{12+1}{12\times1}}$ to $\sqrt{\frac{12+2}{12\times2}}$ for C-H and C-D, respectively.

While the equations from classical mechanics that have been used so far in our discussion of IR spectroscopy have been helpful in understanding the concept as a whole, they do not completely describe behavior that occurs on the atomic level. Thus, we must consider how we can treat these molecular vibrations in a quantum mechanical sense. This consideration still requires the use of the harmonic oscillator previously described in order to arrive at the wave equation of quantum mechanics [17, p. 385]. If we look at the solutions to these wave equations, the potential energies produced have the general form

$$E = \left(v + \frac{1}{2}\right)\frac{h}{2\pi}\sqrt{\frac{k(m_1 + m_2)}{m_1 m_2}}$$

where h is Planck's constant, and v is the vibrational quantum number, which has integer values (0, 1, 2, 3, . . .) [17, p. 385]. This means that only discrete energies can be found, whereas in a classical treatment such restrictions do not exist, and the resulting energy values can be prodcued over a wide range. When we combine $\nu_c = \frac{1}{2\pi}\sqrt{\frac{k(m_1+m_2)}{m_1 m_2}}$ with the previous equation, we obtain

$$E = h\nu_c\left(v + \frac{1}{2}\right)$$

where ν_c is once again the classical vibrational frequency [24, p. 13].

We then make the assumption that there is a specific requirement on the energy of the radiation that causes transitions from one quantum mechanical level to the next. Specifically, this means that the value for the energy of the radiation, in this case infrared, must exactly match the difference in energy levels of the molecule to be excited (ΔE).

Thus, if we reexamine $E = h\nu_c\left(v + \frac{1}{2}\right)$ for the ground state (when v is 0) we obtain

$$E_0 = \left(\frac{1}{2}\right) h\nu_c.$$

Similarly, if excitation occurs to the first excited state, where $v = 1$, the energy for this level would be

$$E_1 = \left(\frac{3}{2}\right) h\nu_c$$

meaning that excitation from the ground to the first excited state would require an input of energy equal to

$$\Delta E = \left(\frac{3}{2}h\nu_c - \frac{1}{2}h\nu_c\right) = h\nu_c.$$

The frequency of radiation that will bring about this excitation is actually identical to the classical vibrational frequency of the bond ν_c [17, p. 386]. Thus,

$$E_{\mathit{radiation}} = h\nu = \Delta E = h\nu_c.$$

Expressing this in wavenumbers then means that we can directly calculate $\tilde{\nu}$ by using

$$\tilde{\nu} = \frac{1}{2\pi c}\sqrt{\frac{k(m_1 + m_2)}{m_1 m_2}}$$

meaning that a better interpretation of the force constant can be obtained.

This comparison of force constants might not be as straightforward as comparing the reduced masses of a deuterated compound with its hydrogen counterpart, as previously shown, but it is important to understand the relative strength of different bonds. Thus, if we wanted to examine the difference between a single and a double bond between two carbon atoms, C-C and C=C, we would immediately know that the reduced mass of the system remains unchanged. Therefore, only the force constant changes between the single and the double bond. Typically, values for these force constants lie in the range between 3×10^2 and 8×10^2 N/m for a single bond [17, p. 386]. The force constant for a double bond, in our example C=C, would be about twice that of a single bond. Similarly, a triple bond would have a value of approximately three times that of a single bond since it is approximately three times as strong as a single bond [17, p. 386].

4.2 Instrumentation for Infrared Spectroscopy

4.2.1 Basic Principles of Infrared Spectroscopy

Before examining the functioning of the actual IR spectrometer, it is important for us to understand the general principle of the method as a whole. To this end, Figure 16 shows the basic idea behind the functioning of an IR spectrometer. The radiation travels on a path from the source, which will be discussed shortly, intersects with the sample, and eventually comes into contact with the detector. The detector then converts its readings into an electronic output, which allows for the formation of a digital IR spectrum. In the following sections, the different possible sources, general path of radiation within an IR

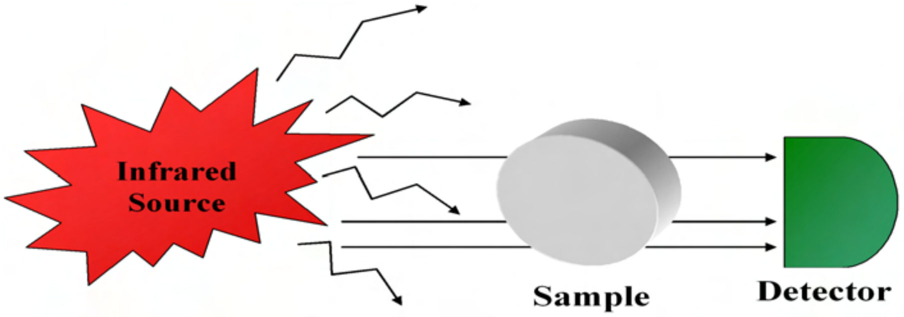

Figure 16: This block diagram of an IR spectrometer shows IR radiation coming from the source. This radiation comes into contact with the sample, where absorption of the radiation by the molecule occurs. Upon exiting the sample, the radiation intersects with the detector, which allows for the spectrum to be produced.

spectrometer, and Fourier Transform infrared spectroscopy will be discussed in order to aid in your understanding of the pattern recognition system that will follow in Chapter 5.

4.2.2 Infrared Sources

In order to produce the necessary infrared radiation for excitation, a source is required. There are several specific types of sources, which when heated, produce radiation of the desired intensity [25, p. 11]. Typically, these sources are heated by electrical means to temperatures between 1500 and 2200 K, which causes the source to approximate a continuum of radiation similar to that of a blackbody [17, p. 389].

A Nernst Glower filament within a spectrophotometer is essentially similar in principle to that of a common household light bulb. The filament is drawn out into a cylinder with a diameter between 1 to 2 mm that is high resistance across the length of the wire [17, p. 389],[25, p. 11]. The composition of the Glower is mainly powdered, sintered oxides of zirconium, thorium, and cerium held together through the use of binding materials [25, p. 11]. When an electrical current is applied across the filament, the temperature rises causing the emission of radiation. It should be noted that the hotter the filament gets, the less electrical resistance is present within the filament. Therefore, the circuit that provides the filament with electricity must be designed in such a way that the current can be limited, otherwise the filament would burn out extremely quickly [17, p. 389].

Another extremely common source used for infrared radiation is a Globar. A Globar is essentially a rod of silicon carbide that is heated to between 1300 to 1500 K [26, p. 75]. Again, the spectral distribution of the Globar approaches that of a theoretical blackbody. The main advantage that a Globar has over the Nernst Glower is the greater output in energy seen below the 5 μm range [17, p. 389]. In addition to the Nernst Glower and the Globar, an incandescent wire source, mercury arc, tungsten filament lamp, and carbon dioxide laser source can all be used to produce infrared radiation [17, p. 390]. Each type of source has its own benefits, but the Nernst Glower and the Globar are the most common sources in the average infrared spectrophotometer [22, p. 52].

4.2.3 Infrared Instruments

There are multiple types of infrared spectrophotometers available including: (a) dispersive grating spectrophotometers that are generally used for qualitative analysis; (b) multiplex instruments that use the methods of Fourier transform; and (c) nondispersive photometers that are used for quantitative work in the determination of organic species present within normal atmospheric conditions [17, p. 392]. Even more recently two-dimensional infrared (2D IR) spectroscopy has become a quite common instrumental technique [27]. The most common technique, however, is still the use of Fourier transform (discussed shortly) spectrometers due to their ability to observe the entire spectrum at once [28, p. 481]. Consequently, this will be the only method discussed within the scope of this analysis.

The vast majority of modern FTIR instruments have their basis in a Michelson interferometer. This type of device modulates optical radiation and produces an interferogram by scanning the difference in the optical path lengths of two beams [24, p. 66]. As can be seen in Figure 17 the interferometer consists of a fixed mirror, a moveable mirror, and a beam splitter [26, p. 80].

The beam splitting mirror divides the source wavelength into two essentially equally powerful beams. These beams are then recombined in such a way that variations in the intensity of the combined beam are measured as a function of the differences in the path lengths of the two beams (with the speed of the moveable mirror remaining constant) [17, p. 186]. Thus, when the two mirrors are equidistant from the beam-splitting mirror, the amount of radiation allowed to pass is at its maximum value. A single scan along the entire

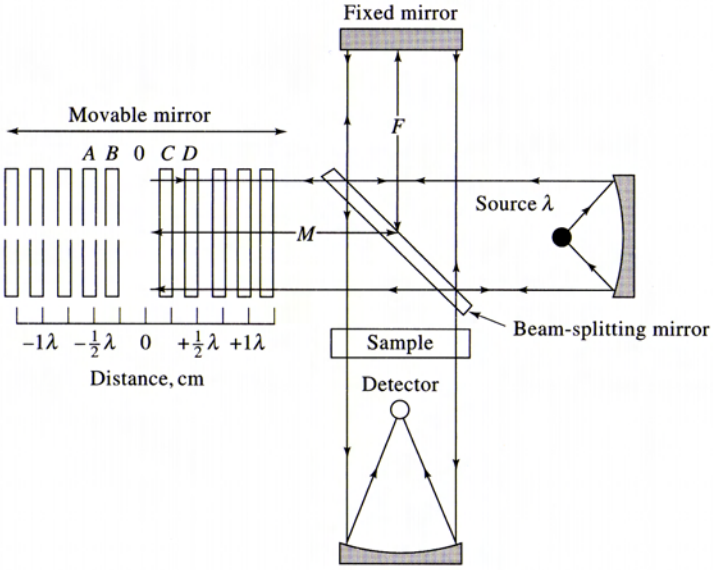

Figure 17: A schematic diagram of the Michelson interferometer, shown here with a monochromatic source. The source produces a wavelength λ that is divided into two beams. These beams interact with both the fixed mirror and the moveable mirror, interacting with the sample, and eventually recombining into one beam that intersects with the detector. Note the moveable mirror travels in distances that are determined by the wavelength of the source [17, p. 186].

length of the moveable mirror's plane allows for a complete single-beam spectrum to be produced, and the greater the length of travel the mirror is allowed, the greater the overall resolution of the produced spectrum. It should be noted, however, that numerous scans (dependent upon the application and the overall concentration of the sample) are collected and then signal averaged by the computer [26, p. 82]. There is much more that can be discussed regarding the actual functioning of this interferometer but such a discussion would lead us away from a more generalized understanding of the instrumentation. For more information regarding this and other types of interferometers see [17] and [26].

With a general understanding of the Michelson interferometer, it is possible to discuss how the radiation produced by the Nerst filament or the Globar is used to gather IR spectroscopic data. To begin, let us examine Figures 18 and 19, which are two schematic diagrams of Fourier transform infrared spectrophotometers. Figure 18 is shown since the

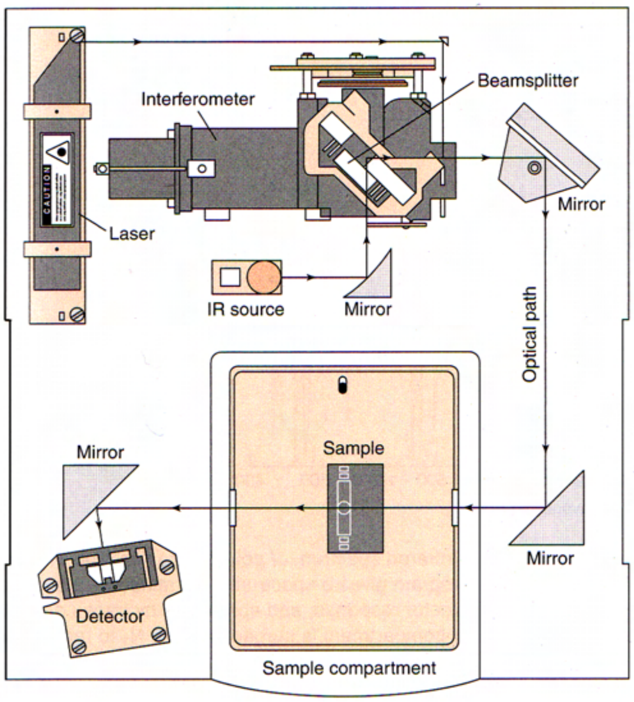

Figure 18: A typical layout of a FTIR spectrometer [28, p. 485]

path of radiation from the source can be readily followed to the sample and eventually the detector. Figure 19 shows the same path, but in a manner that details to a greater extent the actual instrument used in the data collection phase for the formation of the FTIR database. Also, this figure indicates in greater detail the number of mirrors required, and their relative positions to one another, which is what can be seen within the actual instrument. The first item to note on each diagram is the presence of the laser, generally a HeNe source. This laser serves not as the source of the IR radiation, but rather as both a visible means with which to align your sample with the path of the infrared radiation, and as an experimental control for obtaining the infrared interferogram [28, p. 487]. Figure 19 also shows a dessicant box and a shield surrounding the IR source. Both of these have importance within the overall instrumentation. The dessicant box serves to remove excess

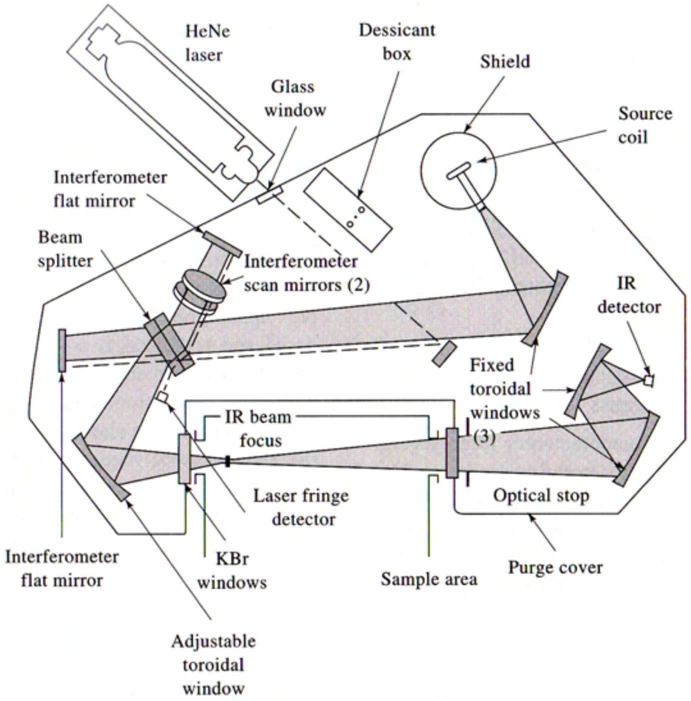

Figure 19: Schematic diagram for a single-beam FTIR spectrometer [17, p. 395]

water vapor from within the inner workings of the spectrometer. Water absorbs within the infrared region, and the presence of such vapor along the path of the radiation would cause erroneous readings to result. Similarly, the shield around the IR source is important in order to direct the radiation along the desired path.

In each case, after leaving the actual source, the IR radiation strikes an initial mirror that directs the radiation toward the Michelson interferometer and eventually to the beamsplitter. The beamsplitter itself is composed of transparent materials with refractive indices such that approximately 50% of the radiation from the source is reflected and 50% is transmitted [17, p. 395]. Usually, this is accomplished through the use of Mylar that is placed between two solid plates that have a low refractive index such as germanium or silicon deposited on cesium iodide or bromide, sodium chloride, or potassium bromide [17, p. 396]. The radiation then leaves the interferometer and comes into contact with another mirror that directs and focuses the radiation on a particular area of the sample cell. The sample is held in place by a cell that is made of an infrared transmitting material. Such materials include glass, quartz, sodium chloride, potassium bromide, and other salts each having their own transmission limits [26, p. 83]. Unless you are conducting an experiment involving thin films either KBr or NaCl is the material of choice. Upon exiting the sample, the radiation passes through another series of mirrors, where the two beams are recombined into one before coming into contact with the detector.

This process is repeated 32 times per sample, which produces approximately 934 data points between 4000 cm^{-1} and 400 cm^{-1} (of course this value varies based on the resolution setting of the spectrometer). A background spectrum is required for subtraction so that any atmospheric contaminants can be removed from the final recording. Once the background is subtracted from the interferogram produced by the sample, the 32 scans are then averaged by the root-mean-square (rms) noise equation

$$\text{rms noise} = \sqrt{\frac{\Sigma_i (A_i - \overline{A})^2}{n}}.$$

In this equation, A_i is the measured signal for the ith data point, $\overline{A}$ is the average signal, and n is the total number of data points (in our case 934) [28, p. 488]. This aids in reducing the overall random electrical noise experienced during data acquisition. Keep in mind, however, that an averaging of n spectra only improves the signal-to-noise ratio by a factor of $\sqrt{n}$. Thus, if the overall signal-to-noise ratio is to be improved by a factor of 5, 25 additional scans would be required. Consequently, for weak samples it would not be unheard of to have the number of scans be on the order of 10^4 or 10^5 [28, p. 488]. Of course, since the output data is digital, the process of averaging is extremely efficient. Yet, this still leaves us with an interferogram and not an actual spectra. The next section will discuss how this interferogram is transformed into a viable spectra such as that seen in Figure 11.

4.3 Fourier Transform in Infrared Spectroscopy

As we previously discussed, it is important to turn the interferogram into a viable spectrum through the use of Fourier transform. Recall from elementary calculus that both differentiation and integration are considered to be transforms. That is, each of these methods transforms one function *into* another function [29, p. 278]. For instance, if we examine the function $f(x) = x^4$ we know that this can be *transformed* into both a cubic polynomial or a family of fifth degree polynomials by integration. Thus, $\frac{d}{dx}x^4 = 4x^3$ and $\int x^4 dx = \frac{x^5}{5} + c$. With this idea in mind, it is possible to further explore how we are able to analogously obtain a spectrum from the Fourier transform of the interferogram.

Mathematically, a Fourier series, which takes its name from the French mathematician J. B. J. Fourier, of a function f defined on an interval $(-p, p)$ is given by [30, p. 643]

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi}{p} x + b_n \sin \frac{n\pi}{p} x \right)$$

where

$$a_0 = \frac{1}{p} \int_{-p}^{p} f(x) dx$$

$$a_n = \frac{1}{p} \int_{-p}^{p} f(x) \cos \frac{n\pi}{p} x dx$$

$$b_n = \frac{1}{p} \int_{-p}^{p} f(x) \sin \frac{n\pi}{p} x dx.$$

This essentially tells us that the original function f can be represented through a *transformation* into a function composed of the summation of sines and cosines. Thus, if the interferogram resembles Figure 20, it is possible to transform this into a series of trigonometric functions. Note that in the image below, the x-axis retardation value (δ) is obtained from the difference in pathlength between the two waves produced in the Michelson interferometer. Figure 20 is far from a simple example, but if we consider the interferogram produced by a monochromatic source of radiation, then the resulting plot is equivalent to a cosine equation resembling

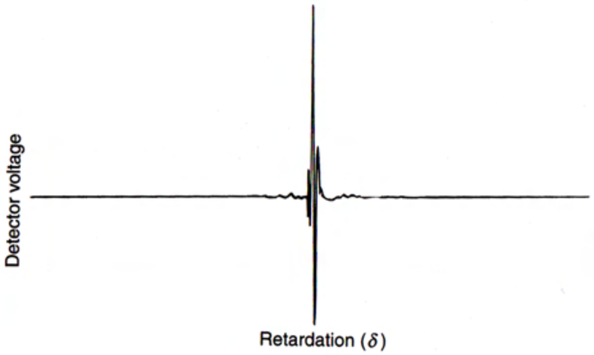

Figure 20: Sample interferogram. Note that constructive interference occurs whenever the retardation, δ, is an integral multiple of wavelength λ of the light. Therefore, as the mirror moves the formation of constructive and destructive phases occurs creating both local and global maxima and minima within the interferogram. [28, p. 486].

$$I(\delta) = B(\tilde{\nu}) \cos\left(\frac{2\pi\delta}{\lambda}\right) = B(\tilde{\nu}) \cos(2\pi\tilde{\nu}\delta)$$

where $I(\delta)$ is the intensity of light reaching the detector and $\tilde{\nu}$ is the wavenumber of the light [28, p. 482]. Thus, it becomes apparent that the intensity, I, of the radiation is dependent upon the retardation term. Yet, this only accounts for monochromatic radiation. The concept can be expanded in a similar fashion if there are two wavelengths of light present to produce

$$I(\delta) = B_1 \cos(2\pi\tilde{\nu}_1\delta) + B_2 \cos(2\pi\tilde{\nu}_2\delta).$$

Generalization of this idea for a continuum source such as that found within an infrared spectrometer means that an infinite number of cosine terms must be used, producing

$$I(\delta) = \int_{-\infty}^{+\infty} B(\tilde{\nu}) \cos(2\pi\tilde{\nu}\delta \; d\tilde{\nu}).$$

This would represent the interferogram as in Figure 20, and taking the Fourier transform of the previous integral provides us with

$$B(\tilde{\nu}) = \int_{-\infty}^{+\infty} I(\delta) \cos(2\pi\tilde{\nu} \; \delta d\delta).$$

Of course, a true Fourier transform would require both a real (the cosine portion) and imaginary (sine) components, but for the purposes of spectroscopy the imaginary portion of the integral can be ignored and it is sufficient to only manipulate the real functions [17, p. 189].

The careful reader would note that the Fourier transform equation as written could not be directly used because this assumes that the source contains radiation from anywhere between zero to infinite wavenumbers ($\tilde{\nu}$). Similarly, the drive mechanism of the moveable mirror would have to be infinitely long in order for the integration to occur from positive infinity to negative infinity. Also, since the output from the spectrometer is digital it cannot be taken in infinitessimally small intervals [17, p. 189]. These facts do not actually have any influence on the mathematics behind the construction of the final output spectra, but they do limit the overall resolution of the instrument. If data could be collected in a manner that would be conducive to the use of the full Fourier transform equation we would have infinitely resolved information. Yet, this cannot practically be the case since we are limited by the computer technology currently available. Even with such limitations, IR spectroscopy is an inherently powerful technique. We will next explore how data obtained from the use of this instrumentation can be manipulated in order to build a database of chemical compounds, similar to the database of faces discussed in Chapter 3.

5

Infrared Spectra Recognition Using MATLAB

Using the mathematical principles discussed in Chapter 2 it is possible to form a database of spectral information obtained through the use of an infrared spectrophotometer. Overall, the process is similar to what has been discussed previously, but the actual technique is quite simplified since we do not have to concern ourselves with converting between an $h \times w$ matrix and a $(w * h) \times 1$ vector. This simplification originates from the fact that the data obtained from the spectrometer is already in vector form (wavenumbers versus percent transmittance). Thus, since the data needed for analysis is already contained in vector form, the resulting SVD calculations follow the same logic that was discussed in Chapter 3. The code for the formation of the database can be found in Appendix C and the code used for testing a spectra from outside the database can be found in Appendix D.

In order to create the database, the FTIR of various chemical compounds had to be obtained. As we saw with the previous database, the larger the amount of information that composes the database the better the ability of the system to perform identification. Therefore, we want to maximize the stretching patterns seen within the various spectra in order to increase the effectiveness of the system as a whole. Reference [31] discusses a similar need to maximize a database of chemical compounds for the purposes of Mass Spectra identification. While the applications are different, the underlying principle remains the same. Table 2 details the compounds that were used in the formation of each database. Ideally, however, a database of approximately 50 compounds or more would serve our purposes much better, but using a smaller set works well to perform a proof of concept. The infrared spectra of the various compounds in Table 2 that comprise the database were collected on a Nicolet 210 FTIR. Spectra of the same compounds were also collected on a Nicolet 510P FTIR in order to determine whether it would be possible for the system to work from instrument to instrument. All spectra were collected using a neat method since all compounds tested exist in liquid form. NaCl windows were cleaned using dicholormethane, and a small amount of solution was placed between the two windows. The spectrophotometer was set to scan a range from 4000 cm^{-1} to 400 cm^{-1} at a resolution setting of 8. It should be noted that the spectrophotometer used for

Compound	Functional Group of Interest
2-methoxyethyl ether	Ether
2-octanol	Alcohol and Alkane Chain
acetone	Carbonyl
aniline	Phenyl ring and Amine
benzaldehyde	Carbonyl and Phenyl ring
benzene	Aromatic ring
cyclohexanone	Carbonyl and Cyclic ring
cyclopentanone	Carbonyl and Cyclic ring
decane	Alkane
ethanol	Alcohol
ethyl acetate	Ester
methanol	Alcohol
p-anisaldehyde	Carbonyl, Phenyl ring, and Ether
tertiary butanol	Alcohol and Tertiary Butyl
toluene	Phenyl ring and Benzylic carbon

Table 2: Details regarding the chemical compounds used to form the database and the functional group of interest on each molecule.

the formation of the database failed completely shortly after data collection began due to a hardware malfunction. Some "bad scans" were noted during data collection, but it was determined that the spectra collected were still valid for our purposes.

5.1 First Spectral Database

The first database we formed contained 27 spectra of 15 different compounds. During data collection several spectra of the same compound were collected. Since many of the compounds in the database volatilize easily this allowed us to obtain spectra with very intense peaks, and ones where many of the important peaks had weakened to a great extent. We decided to use 5 of the 27 basis vectors, which retained 91.185% of the variance. Recall that the percentage of the variance retained is calculated by

$$Per = \frac{\displaystyle\sum_{i=1}^{j} \lambda_i}{\displaystyle\sum_{i=1}^{n} \lambda_i} * 100\%.$$

The computer took less than one second to load the spectra and perform the necessary calculations to construct the database. As with face recognition, it is possible to examine the average spectra of all compounds tested. The construction of this image was accomplished by taking the mean of all vectors in the matrix B, which is $h \times i$ where h is the height of the data vectors, and i is the number of spectra in the database. This mean vector was then plotted against wavenumbers, which produced the image in Figure 21. It can be noted that from this average spectrum little information about functional groups in the database can be obtained between approximately 2750 cm^{-1} and 2000 cm^{-1}. Also, alcohol functional groups, which normally occur between 3600 cm^{-1} and 3000 cm^{-1}, are prominent within this average spectrum [19, p. 561]. Still we should observe that this

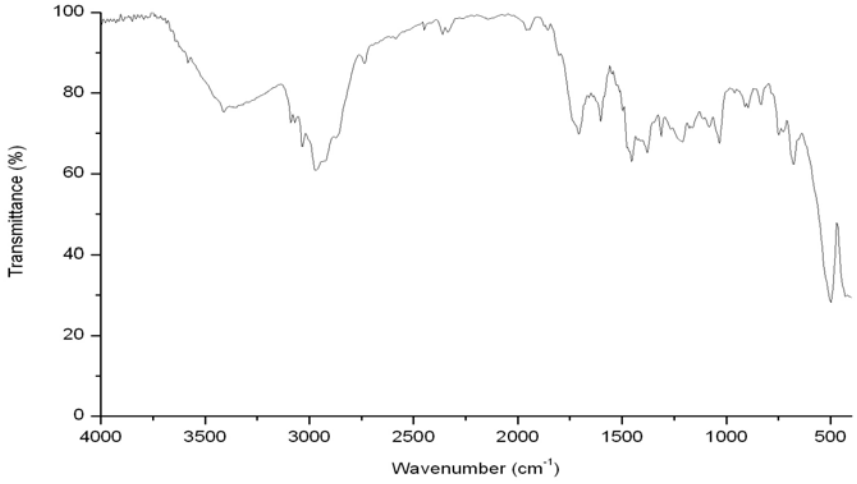

Figure 21: Average Spectrum from the First Database

average image holds a striking resemblance to an infrared spectrum, just as the average face in Chapter 3 resembled a human face.

Similarly, another image that can be examined is the first basis vector from the computation. We will informally call such an image an eigenspectrum. Figure 22 shows this first basis vector. It is not readily apparent which features of the spectra are being distinguished as it was for face recognition. One likely reason for such difficulties in determination might

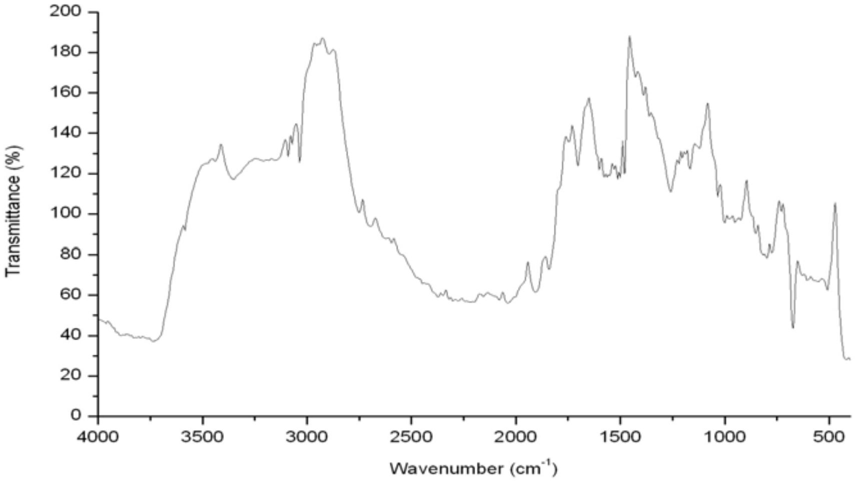

Figure 22: First Eigenspectrum from Database 1

stem from how humans interpret information. The spectra produced does not give clear areas of differences like the eigenfaces do since basis vectors from a facial recognition database can readily display contrast between areas of differentiation. One likely hypothesis is that this first basis vector is classifying compounds based on stretches that occur between 2750 cm^{-1} and 2000 cm^{-1}. Most of the spectra in the database see almost no peaks occurring within this range. There are, however, several spectra that show weak peaks within the given range. Just as hair was a major determining factor within the first facial recognition database, areas where only some compounds exhibit vibrational modes aid the system in beginning the classification process. We should note that due to the smaller sample size of the database the overall implications that the eigenspectra hold are somewhat diminished. A large database would have to be constructed in order to fully interpret what information can be gleaned from the eigenspectra.

Overall, this database served the purpose of proving the concept that the same mathematical basis for face recognition could be applied in other settings to perform similar tasks. The system was able to identify, with 100% accuracy, test spectra that already existed in the database. It is not until the second database was constructed that different spectra of compounds in the database were tested, as well as the ability of the system to correctly identify compounds from one spectrometer based on a database of spectra taken on an different model of FTIR.

5.2 Second Spectral Database

In order to properly test the system repeat spectra were removed from the database composition and only one spectrum from each compound listed in Table 2 were used to build this second database. Thus, this database consisted of 15 total spectra and we used 5 of the basis vectors, which retained 89.5815% of the variance within the database. Overall, this elimination of duplicate spectra did not greatly alter the average spectrum, which can be seen in Figure 23. This minimal alteration indicates that the various spectra collected for each compound have peaks within essentially the same region between subsequent measurements of the same compound.

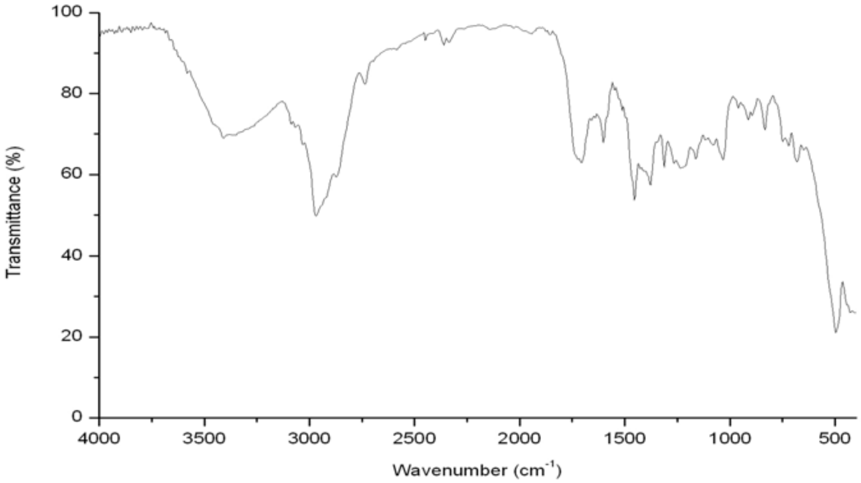

Figure 23: Average Spectrum from the Second Database

The eigenspectra from this database are also interesting to examine. Figure 24 shows the first and second eigenspectra produced by this database. The first basis vector (eigenspectrum) appears to resemble a generic infrared spectrum to a greater degree than what we have previously encountered. Interestingly, the entire graph occurs in a negative region. Since the collected spectra that form the database were not normalized, this appearance of an almost "negative" eigenspectrum indicates that the most important factor in this database is examining whether the data input falls within a range that exceeds the normal 0 to 100% transmittance. We then hypothesized that the second eigenspectrum is examining peaks that occur in both the alcohol and carbonyl stretches. This hypothesis would make sense when coupled with the fact that such peaks are always strikingly intense and play a predominant role in human identification of spectra since such peaks are almost always readily indicative of these functional groups.

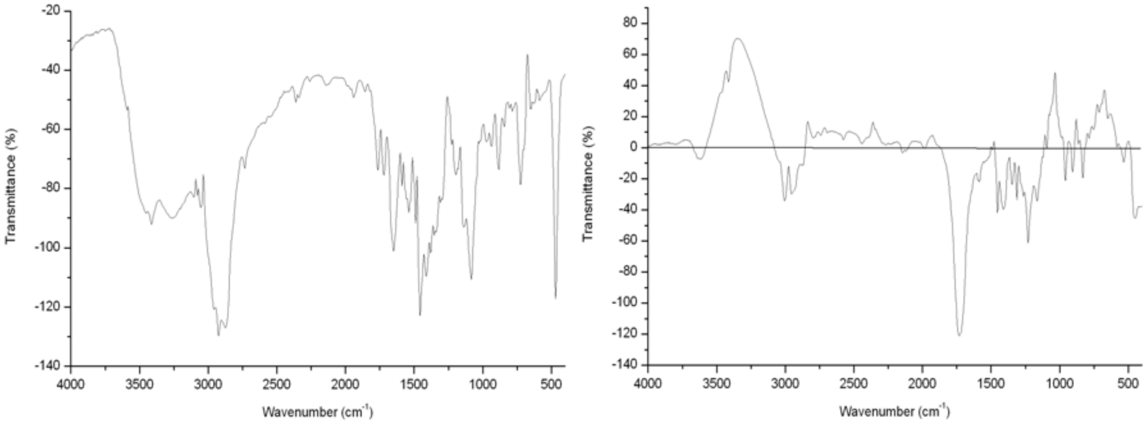

Figure 24: The graph on the left shows the first eigenspectrum from this Second Database. Of note in this graph should be the Percent Transmittance scale since the entire spectrum occurs below zero. The graph on the right shows the second eigenspectrum from this second database. This spectrum shows regions of importance where alcohols and carbonyl functional groups normally occur.

The first test that was performed with this database involved using spectra of the same compounds that were obtained on the same instrument used in the creation of the database. Thus, the extra spectra that were removed from the first database were used as spectral images from outside the database. Each spectra was tested, and the ranking of the correct spectrum by the database was noted. Overall, this led to a 58.33% success rate, where the correct spectrum was placed as one of the top three closest matches and 41.66% of the time the first choice was the correct choice. Yet, when conducting these tests, it was noted that 2-octanol, benzaldehyde, cyclohexanone and ethanol all produced rankings that were quite high, which is far from ideal since we would want the ranking of the spectra to be as low as possible. Upon further examination of these graphs, we noticed that the tested spectra and the spectra of these compounds did not correlate well with one another, even to the human eye. Figure 25 shows the spectra in the database compared with the spectra that were used to test the system. It became clear after examining these spectra side by side why the system was not able to correctly identify these spectra since

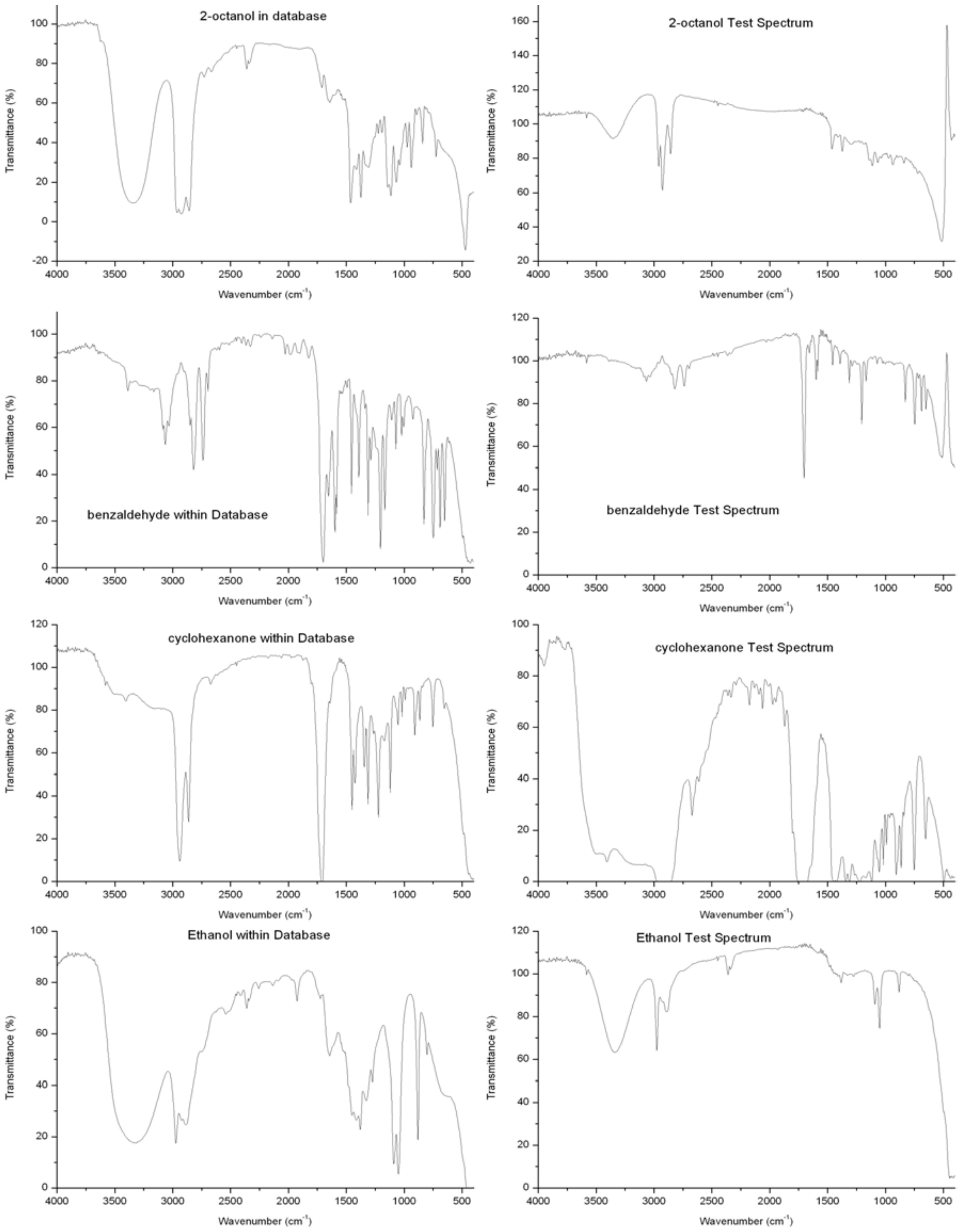

Figure 25: The graphs on the left show the spectra that was placed in the database and the graphs on the right show the spectra of the same compound that was used to test the database. In the cases of 2-octanol, benzaldehyde, and ethanol the spectra on the right were deliberately obtained after allowing the compound to volatilize in the IR cell. For cyclohexanone, the IR cell was loaded with excess liquid in order to obtain broadened peaks.

even the human eye has difficulty seeing similarities between the spectra on the left and the spectra on the right of Figure 25. If these results are removed from consideration the overall accuracy of the system to correctly identify the spectrum in the top three choices increases to 87.50% and 62.50% of the time the computer identifies the correct spectrum as the closest match.

The next test involved obtaining spectra (of the compounds that are within the database) on another fourier transform spectrometer. This was an important variable to test since a pattern recognition system that could not correctly identify spectra taken on another instrument would not prove to be a viable system. The same methodology already discussed for the preparation of the cells was used and measurements were obtained on a Nicolet 510P FTIR. The resolution of the original measurements that comprise the database was matched on this second FTIR and nine of the original 15 compounds were measured. Overall, we were able to determine that the system correctly identified the spectra as one of the top three closest matches with 64.71% accuracy. It was noted, however, that the samples for toluene used to conduct these tests did not closely correlate with the toluene spectra found within the database. This resulted in the system having difficulty correctly identifying this spectra. Thus, with these erroneous tests removed from consideration we found that the system had an accuracy of 71.43% of the correct match being in the top three and 50.00% of the time the closest match was the correct match.

One interesting result that was discovered during testing was the influence of using normalized versus non-normalized data. When a spectrum is collected it can often exceed 100% transmittance on the y-axis. In the construction of the database all of the spectra did not greatly exceed 100% transmittance. When testing, the data on the Nicolet 510P spectra that would go beyond 100% would be saved both in a normalized and non-normalized (raw) form. It was noted that most of the time the non-normalized data would result in the correct identification being the first or second choice, while the normalized

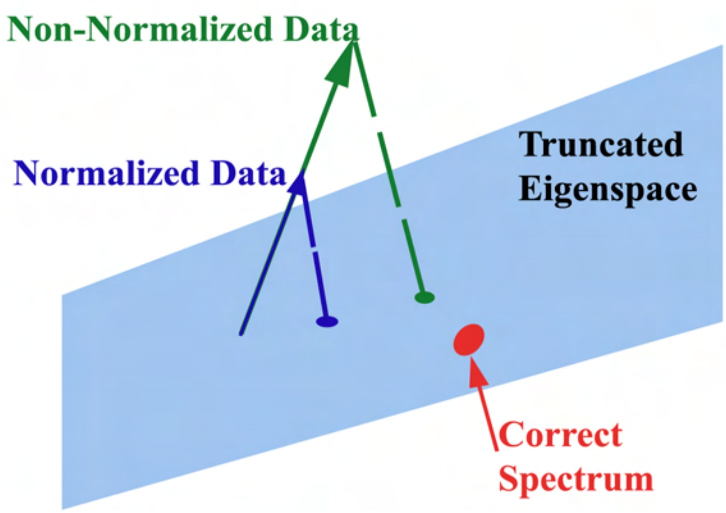

Figure 26: This image represents the projection of the non-normalized and normalized spectra onto the subspace spanned by the eigenspace. Essentially, the projection of each data set intersects the truncated eigenspace at slightly different values due to differences in normalization. Then, since the Euclidean distance is measured between these projections and the spectra already in the database, it is possible for the correct spectrum to be closer to one rather than the other. The results indicate that this reasoning is likely correct since we are not able to form a generalized trend for whether normalized or non-normalized spectra work better in the database.

data would be correctly identified between the fourth and seventh positions. Yet, there was one slight discrepancy noted when the 2-octanol spectrum was used. In this instance the non-normalized data was correctly identified in the second position, while the normalized data was identified correctly as the first choice. This discrepancy likely originates from the projection of the image outside the database onto the subspace spanned by the eigenspace. This is demonstrated in Figure 26. Thus, in future studies and uses of this method, consistency in the use of normalized or non-normalized spectra must be achieved.

At this point, we should note that while the accuracy ratings are not 90% or above, the speed with which the process occurs makes the system beneficial for use within the laboratory. Normal spectral identification without the use of a computer involves analyzing the peaks that are found within the spectrum and determining which functional groups correspond to the identified peaks. Once that is complete then other spectroscopic information such as nuclear magnetic resonance (NMR) or mass spectrometry are used to further elucidate the identity of the molecule. Yet, the system we have tested thus far can build the spectral database of compounds in **less than one second** and test a compound in approximately 0.01 seconds. Thus, while the match might not always be in the top three, this information can be used as a starting point for elucidation of structure since the system does appear to often choose other molecules within the database that have similar functional groups.

5.3 Third Spectral Database

The final database was composed in order to determine how much of an influence the number of basis vectors used in the composition of the system had on the accuracy of its results. Thus, we chose to use 10 basis vectors in this database, which retained 97.8694% of the variance. We would then expect, at the very least, a slight increase in the accuracy of the system. The average spectrum produced by MATLAB did not have any noticeable changes from Figure 23. Also, the first few eigenspectra were identical to those seen in Figure 24.

The same tests performed in Section 5.2 were conducted on this new database. The percentage of correct identification remained the same for spectra obtained on the same instrument used to construct the database, 87.50%. The overall percentage for choosing the correct match in the top three choices increased slightly to 78.57% (with the exclusion of the erroneous toluene data) and 57.14% of the time the correct spectrum was the first choice. Thus, while it is difficult to draw conclusions because of the small size of the database we can say that at this point that the first five basis vectors do contain the most information about the spectra within the database. It does appear that a simple increase in the number of basis vectors used is not necessarily indicative of an increase in overall accuracy. Still the fact that both trials produced results in approximately the 80% range indicates that the system should be examined further in the future with a larger database to determine its overall viability as an everyday laboratory tool.

6

Conclusion

We now have a better overall understanding of what pattern recognition is, how it can be implemented mathematically, and the types of applications where such methods might prove useful. The two applications discussed here are far from the only possible uses for this pattern recognition system. The same principles could be applied to identify animals, fingerprints, other types of spectroscopic data, or almost anything we could think of involving a pattern that humans might find tedious to analyze.

While we have shown that the system works exceptionally well with face recognition and infrared spectroscopy pattern recognition, there is still room for improvement. Mainly, one area that could be improved with both applications is that of automation. Currently, there still remains some work to be done by humans in order to make the program function. Ideally, an exceptionally viable program would require little input on the part of the user and could be able to rebuild databases as new information became available on the fly. Such automation would also lead to the ability of applying this system to recognizing faces within moving images, or having the computer immediately output the identity of a compound once it has finished collecting data from the FTIR. In particular, with regard to the system used for face recognition, it would be wonderful to further examine how the problems of head tilt, lighting, and changes to hair could be readily resolved within the mathematical basis for the system. Also, as with any system it still remains to be seen how a database of 200, 500, or 1000 faces would influence not only the time it takes for recognition to occur, but the overall accuracy of the system. In theory, the system should be more accurate since it would have a greater amount of information from which to draw.

Many of the key points to examine in the face recognition database could also use further scrutiny regarding the spectroscopic identification. One of the main problems with the database previously discussed is its size. This small size originated from equipment failure over the course of experimentation. Yet, the results do provide the proof of concept for this application. There are already commercially available applications that perform such recognition of spectra such as [32]. How such systems function are not readily revealed due to patents, and programs such as these can cost in excess of $2000 for a single database of compounds and the software. It would be interesting to examine in future studies the accuracy of these systems in determining not only the correct identity of a compound, but also whether it works for mixtures of compounds or non-neat methods (since non-liquid compounds must be treated in a slightly different manner than liquids for infrared spectroscopy).

As technology continues to progress the ability of computers to accurately and quickly perform pattern recognition will only increase. Thus, it is possible to foresee that construction of large databases to perform these recognition tasks will become much more commonplace in the future. Remember, it was not so long ago that only extraordinarily expensive computers could perform the tasks described here. In fact, all calculations were performed on desktop computers using a standard amount of RAM and a processor greater than 2GHz. Yet, we must remember the power of our own minds. While we have shown the ability of a computer to perform tasks that humans do without thinking, what we have actually shown is the ingenuity we inherently possess within our own minds to solve problems and better understand the world around us. We must never forget that, as Marcus Aurelius put it, "nothing has such power to broaden the mind as the ability to investigate systematically and truly all that comes under thy observation in life" [33]. Such observation cannot be performed by a computer. While our system shows the power of a computer there are always limitations and tasks that the human mind can perform with much greater ease. So, the next time you look at a person's face and recognize them, examine an infrared spectra, or simply see a pattern within an object, remember how easily you complete the task and how much effort it would be to make a computer carry out the exact same charge.

Appendices

Appendix A

Code for Face Recognition Database Formation

The following code was used to load all of the images in the database from a textfile. This textfile was generated through the use of command line within Windows. For more information regarding how to change directories and produce a bare format filelist see [16].

In general this MATLAB function loads all of the images, converts them to the proper ("double") format and reshapes them all into vectors. Then, the function performs all of the computations on the database which are necessary to create the best basis as described by Proposition 2.8. Please, see our explanations throughout the code (denoted with the % symbol). Before considering the code, however, we must note it is necessary to define two other functions which were used within this image loading function. The first is defined as "linecount" and its purpose is to count the number of lines within the filelist to determine how many images we will be loading. The code is as follows.

```
function lc=linecount(filename)

fid=fopen(filename,'r'); if fid < 0
    lc=0;
else
```

```
    lc=0;
    while 1
        ln=fgetl(fid);
        if ~isstr(ln) break; end;
        lc=lc+1;
    end;
    fclose(fid);
end;
```

The second function which is needed to execute the “load images” function is known as “sorteig”. This function simply sorts the eigenvalues of our covariance matrix in descending order and then sorts the corresponding eigenvectors in the same order.

```
function[V1,L1]=sorteig(V,L)
lambda=diag(L);
n=length(lambda);
[val,idx]=sort(lambda);
val=val(n:-1:1);
idx=idx(n:-1:1);
L1=diag(val);
V1=V(:,idx);
```

Now, with all the previous m-files loaded into the work folder of MATLAB we can define our “load images” function. We should note the syntax that MATLAB uses for its coding. The initial line tells the program that we are defining a function. The information in square brackets on the left side of the equal sign indicates variables that will be output after the code has been allowed to run. The “load_images” tells MATLAB the name of the function, and the variables within parenthesis are what the program needs in order to run properly. In this case that would be a filelist with a .txt extension, such as “filelist.txt” and b determines the desired number of basis vectors used in the computations. Finally, in order to run the code from the command screen we would have to type

```
>>[Images, w, h, Meanimage, Meanface, Meansub, V, S, Basis, SmallBasis,
SmallV, Percent]=load_images(’filelist.txt’,5)
```

making sure to note the use of single quotes around the filename of the filelist. In this case, we would be using five basis vectors for analysis. The code that MATLAB would follow is now detailed below.

```
function [Images, w, h, Meanimage, Meanface, Meansub, V, S, Basis,
SmallBasis, SmallV, Percent] =load_images(filelist,b)
numimgs=linecount(filelist);
fid=fopen(filelist,’r’);

for i=1:numimgs
  imgname=fgetl(fid);
  if ~isstr(imgname)
      break;
  end;

  fprintf(1, ’loading JPG file %s\n’, imgname);

  Img=imread(imgname);

  Img=double(Img)+1;

  if i==1
      [w,h]=size(Img);
  end;

  Images(1:w*h,i)=reshape(Img,w*h,1);
   end;
  fclose(fid);

  fprintf(1, ’Read %d images.\n’, numimgs);
%All of these lines up to now have simply loaded the images,
%converted them to the proper format, reshaped them into
%vectors and printed out a running list as the computer loaded each file.

  Meanimage=mean(Images’);
  Meanface=reshape(Meanimage, w, h);
  imagesc(Meanface); figure(gcf);
  colormap(gray);
%Displays the average face.

  Meanimage=Meanimage’;

  Meansub=Images-repmat(Meanimage, 1, numimgs);

  Covar=Meansub’*Meansub; %Calculates the covariance matrix.

  [V, S]=eig(Covar);
  [V, S]=sorteig(V,S);

  Basis=Meansub*V;
%Calculates basis from eigenvectors.

  for g=1:b
      SmallBasis(:,g)=Basis(:,g);
      SmallV(:,g)=V(:,g);
  end

  for t=1:b
      Per=sum(diag(S(1:t,1:t)));
      Percent=(Per/(sum(diag(S))))*100;
%This calculates the percent of the variance retained by
%our b basis vectors.
  end
```

Appendix B

Code for Testing Facial Image from Outside the Database

This code begins in basically the same way as the database formation by loading the outside image into the MATLAB workspace. The test image is then mean subtracted and projected onto the subspace spanned by our chosen number of basis vectors. The distance from each of the images in the database is then calculated. These vectors, which are the database images, are then sorted from smallest to greatest distance. The final aspect of the code displays the test image alongside the top three closest matches from within the database.

```
function [RTest, Coeff, Dist, B, Index, best1, best2,
best3]=outside_image(testimage,SmallBasis, Meanimage, SmallV, Image,
k)
numimgs=linecount(testimage); fid=fopen(testimage,'r');

for i=1:numimgs
  imgname=fgetl(fid);
  if ~isstr(imgname)
      break;
  end;

  Img=imread(imgname);

  Img=double(Img)+1;

  if i==1
```

```
      [w,h]=size(Img);
  end;

  RTest(1:w*h,i)=reshape(Img,w*h,1);
   end;
  fclose(fid);

  RTest=RTest-Meanimage; %Mean-subtract the test image

  for i=1:k
     Coeff(1,i)=(dot(SmallBasis(:,i),RTest))/(dot(SmallBasis(:,i)
       ,SmallBasis(:,i)));
  end
%Projects the test image onto the space spanned by the chosen basis vectors.

  [r, c]=size(SmallV);

  for i=1:r
      Dist(i,1)=norm(SmallV(i,:)-Coeff);
  end
%Calculates the distance of this projection from all the database images.

  [B,Index]=sortrows(Dist);
%Sorts the database images from least to greatest
%distance from the test image.

  m1=Index(1,1);
  m2=Index(2,1);
  m3=Index(3,1);

  best1=Image(:,m1);
  best2=Image(:,m2);
  best3=Image(:,m3);

  best1=reshape(best1, w, h);
  best2=reshape(best2, w, h);
  best3=reshape(best3, w, h);

  figure(1);
  imagesc(Img);
  colormap(gray);
  figure(2)
  imagesc(best1);
  colormap(gray);

  figure(3);
  imagesc(best2);
  colormap(gray);

  figure(4);
  imagesc(best3);
  colormap(gray);
%Displays the test image and three closest matches from the database.
```

Appendix C

Code for Infrared Spectra Database Formation

The following code was used to load all of the IR spectra into the computer's memory and form the database. A textfile was again used to tell MATLAB which spectra to load since multiple databases were formed as discussed in Chapter 5. Again, as in Appendix A the instructions for generating a filelist from within Windows can be found in [16].

While the code remains essentially the same, there are several significant differences. For instance, the problem encountered when forming the database for face recognition required the conversion of images into the "double" format. In this case, the data was taken from the spectrometer as a **.csv** or comma separated value file. Therefore, when this data was imported into MATLAB through the use of the "csvread" command, it was already in the proper format for mathematical analysis. Note that the two functions "linecount" and "sorteig" were required for the functioning of the system, just as was shown in Appendix A.

```
function [Spectra, w, h, MeanSpectra, Meansub, V, S, Basis, SmallBasis,
SmallV,Percent]=load_ir(filelist,b)

numir=linecountir(filelist);
fid=fopen(filelist,'r');
  %This opens the file input/output commands within Matlab.
  %The 'r' tells Matlab it has only read permission for the
  %filelist in question.
```

```
for i=1:numir
  irname=fgetl(fid);
  if ~isstr(irname)
      break;
  end;

  fprintf(1, 'loading IR file %s\n', irname);
  %Tells the user which IR spectra file is currently being loaded
  %into the system's memory.

  Spec=csvread(irname);
  %Uses the built in csvread command to convert the commaseparated value
  %files into matricies within Matlab
  if i==1
      [w,h]=size(Spec);
  end;

  Spectra(1:w,i)=Spec(:,2);
  %Since the first column of all the spectra (the values for wavenumbers)
  %are the same this extracts only the second column, which contains
  %information regarding percent transmittance of the compound.
   end;
  fclose(fid);

  fprintf(1, 'Read %d IR.\n', numir);

  MeanSpectra=mean(Spectra');

  MeanSpectra=MeanSpectra';

  Meansub=Spectra-repmat(MeanSpectra, 1, numir);

  Covar=Meansub'*Meansub;
  %Calculates the covariance matrix.

  [V, S]=eig(Covar);
  [V, S]=sorteig(V,S);
  %Uses the sorteig command to sort the eigenvalues.

  Basis=Meansub*V;

  for g=1:b
```

```
      SmallBasis(:,g)=Basis(:,g);
      SmallV(:,g)=V(:,g);
  end

  for t=1:b
      Per=sum(diag(S(1:t,1:t)));
      Percent=(Per/(sum(diag(S))))*100;
%Calculates the percent of the variance retained through the
%use of the b basis vectors.
  end
```

Appendix D

Code for Testing Infrared Spectra from Outside the Spectral Database

Again, the code begins by loading the spectra from outside the database into the MATLAB workspace. This is accomplished by the “csvread” command once again. Again, the spectra to be tested is mean subtracted and projected onto the subspace spanned by the chosen number of basis vectors. The Euclidean distance is then found between the tested spectra and those within the database. These vectors are then sorted from closest (most probable match to a spectra in the database) to most distant (least likely match with the spectra in the database). Since the vector containing the wavenumber values has been removed for the purpose of calculation MATLAB does not readily show the top three matches from the database, but an examination of the Index variable tells us which spectra the test image corresponds to in order of probability. If the spectra are desired to be viewed, a scientific graphing spreadsheet such as Excel or OriginPro can be used to produce the spectra.

```
function [RTest, Coeff, Dist, B, Index]=outside_irpaper(testir,
SmallBasis, MeanSpectra, SmallV, k)
numir=linecount(testir);
%The linecount function here will have to only count one line,
%but it makes it easier to switch which file to test.
fid=fopen(testir,'r');
```

```
for i=1:numir
  irname=fgetl(fid);
  if ~isstr(irname)
      break;
  end;

  Spec=csvread(irname);

  if i==1
      [w,h]=size(Spec);
  end;

  RTest(1:w,i)=Spec(:,2);
   end;
  fclose(fid);

  RTest=RTest-MeanSpectra;
  %Produces the mean subtraced test spectrum.

  for i=1:k
     Coeff(1,i)=(dot(SmallBasis(:,i),RTest))
     /(dot(SmallBasis(:,i),SmallBasis(:,i)));
  end

  [r, c]=size(SmallV);

  for i=1:r
      Dist(i,1)=norm(SmallV(i,:)-Coeff);
  end

  [B,Index]=sortrows(Dist);
%Produces the sorted rows of the measured distances so that the
%index matrix reflects the overall ranking of spectra within the database.
```

Bibliography

[1] F.A. Farris and N.K. Rossing. "Woven Rope Friezes." *Mathematics Magazine.* Vol. 72, No. 1. February 1999, pp. 32-39. [cited at p. iii, 1]

[2] W. Andrew. *Statistical Pattern Recognition.* West Sussex: John Wiley & Sons, 2002. [cited at p. 2, 3]

[3] S. Theodoridis and K. Koutroumbas. *Pattern Recognition.* 2^{nd} ed. Amsterdam: Academic Press, 2003. [cited at p. 3, 4]

[4] B. Farhang-Boroujeny. *Adaptive Filters: Theory and Applications.* Chichester: John Wiley & Sons, 1998. [cited at p. 3]

[5] A.J. Myles. *An Investigation of Chemical Pattern Recognition Methodologies.* Diss. U of Delaware, 2005. Ann Arbor: UMI, 2005. 3169527. [cited at p. 4]

[6] F. Chau, Y. Liang, J. Gao, and X. Shao. *Chemometrics: From Basics to Wavelet Transform.* Hoboken, New Jersey: Wiley-Interscience, 2004. [cited at p. 5]

[7] A. Giordano and M. Uhrig. "Human Face Recognition Technology Using the Karhunen-Loéve Expansion Technique." *Rose-Hulman Undergraduate Mathematics Journal.* Vol. 7, No. 1. 2006. <http://www.rose-hulman.edu/mathjournal/archives/2006/vol7-n1/paper11/v7n1-11pd.pdf>. [cited at p. 25]

[8] R. Chellappa, C.L. Wilson, and S. Sirohey. "Human and Machine Recognition of Faces: A Survey." *Proceedings of the IEEE.* Vol. 83, No. 5. May 1995, pp. 709-40. [cited at p. 4]

[9] R.G. Brereton. *Chemometrics: Data Analysis for the Laboratory and Chemical Plant.* West Sussex: John Wiley & Sons, 2003. [cited at p. 28]

[10] L. Sirovich and M. Kirby. "Low dimensional procedure for the characterization of human faces." *Journal of the Optical Society of America A, Optics and Image Science.* Vol. 4, No. 3. March 1987, pp. 519-24. [cited at p. 4]

[11] M.A. Turk and A.P. Pentland. "Face Recognition Using Eigenfaces." *International Conference on Pattern Recognition.* 1991, pp. 586-91. [cited at p. 4]

[12] M. Turk and A. Pentland. “Eigenfaces for Recognition.” *Journal of Cognitive Neuroscience.* Vol. 3, No. 1. 1991, pp. 71-86. [cited at p. 4]

[13] D. Hundley. *Chapter 6: The Best Basis.* 2004. Whitman College. 20 June 2005. <http://marcus.whitman.edu/∼hundledr/courses/M350/Ch5-Ch7.ps>. [cited at p. 4]

[14] R.A. Horn and C.R. Johnson. *Matrix Analysis*. New York: Cambridge University Press, 1992. [cited at p. 6]

[15] D.C. Lay. *Linear Algebra And Its Applications*. 3rd ed. Boston: Addison Wesley, 2003. [cited at p. 6, 25]

[16] S. Dutch. “Capturing File Lists.” 28 March 2002. University of Wisconsin-Green Bay. 5 July 2005. <http://www.uwgb.edu/dutchs/CompTips/FileLists.HTM>. [cited at p. 73, 80]

[17] D.A. Skoog, F.J. Holler, and T.A. Nieman. *Principles of Instrumental Analysis*. 5th ed. Philadelphia: Thomson Learning, 1998. [cited at p. iv, v, 39, 44, 45, 46, 47, 48, 49, 51, 55]

[18] I. Simon. *Infrared Radiation*. Princeton, New Jersey: D. Van Nostrand Company, Inc., 1966. [cited at p. 39]

[19] F.A. Carey. *Organic Chemistry*. 5th ed. New York: McGraw-Hill, 2003. [cited at p. vii, 39, 40, 59]

[20] Spectral Database for Organic Compounds (SDBS) 17 February 2007. <www.aist.go.jp/RIODB/SDBS/menu-e.html>. [cited at p. iv, 40]

[21] K.J. Laidler, J.H. Meiser, and B.C. Sanctuary. *Physical Chemistry*. 4th ed. Boston: Houghton Mifflin, 2003. [cited at p. 41]

[22] L.D.S. Yadav. *Organic Spectroscopy*. New Delhi, India: Kluwer Academic Publishers, 2005. [cited at p. 39, 42, 48]

[23] S. Wartewig. *IR and Raman Spectroscopy: Fundamental Processing*. Germany: Wiley-VCH GmbH & Co. KGaA, 2003. [cited at p. 43]

[24] H.W. Siesler, Y. Ozaki, S. Kawata, H.M. Heise, eds. *Near-Infrared Spectroscopy: Principles, Instruments, Applications*. Weinheim, Germany: Wiley-VCH, 2002. [cited at p. 44, 48]

[25] A.D. Cross. *An Introduction to Practical Infra-Red Spectroscopy*. 2nd ed. Norwich, England: Butterworth & Co., 1967. [cited at p. 47]

[26] N.B. Colthup, L.H. Daly, and S.E. Wiberley. *Introduction to Infrared and Raman Spectroscopy*. 3rd ed. Boston: Academic Press, 1990. [cited at p. 47, 48, 49, 52]

[27] J.K. Gillie, J. Hochlowski, and G.A. Arbuckle-Keil. "Infrared Spectroscopy." *Analytical Chemistry*. Vol. 72, No. 12. 2000, pp. 71R-79R. [cited at p. 48]

[28] D.C. Harris. *Quantitative Chemical Analysis*. 6th ed. New York: W. H. Freeman and Company, 2003. [cited at p. iv, v, 48, 50, 52, 54]

[29] D.G. Zill and M.R. Cullen. *Differential Equations with Boundary-Value Problems*. 6th ed. Belmont, CA: Thomson Learning, 2005. [cited at p. 53]

[30] W. Fulks. *Advanced Calculus: An Introduction to Analysis*. 3rd ed. New York: John Wiley & Sons, 1978. [cited at p. 53]

[31] T.J. Stonham and M.A. Shaw. "Automatic Classification of Mass Spectra by Means of Digital Learning Nets - Existence of Characteristic Features of Chemical Class in Mass Spectra." *Pattern Recognition*. Vol. 7. 1975, pp. 235-241. [cited at p. 57]

[32] "FDM Library Search." *FDM Reference Spectra Databases* 2 March 2007. 25 March 2007 <http://www.fdmspectra.com/FDM_Library_Search.htm>. [cited at p. 70]

[33] J. Bartlett. *Familiar Quotations*. 10th ed. *Bartleby.com*. 27 March 2007 <http://www.bartleby.com/100/718.17.html>. [cited at p. 71]

[34] J. Daintith, Ed. *A Dictionary of Chemistry*. 4th ed. Oxford: Oxford University Press, 2000. [cited at p. 39, 41]