Face Recognition Using Principal Component Analysis (PCA)

ASRAF ALI Abdul Salam Rasmi

NAZIFI Nahid

Masters in Computer Vision

University of Burgundy

January 4, 2019

Supervisor: **Prof. SIDIBE Desire**

VIBOT, IUT Le Creusot, UMR6306 CNRS/uB

Contents

List of Figures

Chapter 1

Introduction

Face recognition (FR) is one of the most popular techniques used today for identification and verification due to it is efficiency. Face Recognition can be use for crowd surveillance, video content indexing, personal identification (example driver's license), entrance security, etc [1]. The basic idea behind any Face Recognition system is to extract the set of interesting and discriminate features from the face images with the aim of reducing the number of variables, which can easily be achieve by using principal component analysis (PCA). Principal Component Analysis is a way to identify and find the similar patterns from a given set of data. PCA has been implemented in various applications like face recognition, hand written text matching and in image compression and plays a vital role in image compression as we compress the data by reducing dimensions.

1.1 Objective

The main objective of this project is to apply mathematical concepts such as SVD and PCA in the real life problem solving and decision making through creating a simple face recognition system.

Chapter 2

Normalization

2.1 Procedure

Normalization is required in face recognition to acquire aligned face images to correct scaling, orientation and location variations. Normalization procedure is generally geometric transformation. The idea is to find the best transformation that maps the input image's feature points with a predefined feature points. This allows us to make all images with same size and similar orientations.

For every face image in our data set, we have a set of location for facial features F_i as well as a set of predefined locations for these features F_p in our matrix which is reduced into 64×64. So our F_i and F_p vectors are of same size. For instance, if we consider 5 facial features each with 2 positions (x, y), then both F_i and F_p will be a (5×2) Matrix and the algorithm works as follows[2].

1. The feature of the first image in training data set is loaded into $\bar{F}$

$$\bar{F} \longleftarrow F_1$$

2. For each iterations, find the best transformation, given by (A, b) that maps the features in $\bar{F}$ to those in F_p.

3. Apply this transformation onto $\bar{F}$ to get $\bar{F}^{'}$ and set $\bar{F} \leftarrow \bar{F}^{'}$.

4. For each image, I_i in the training set, find the best transformation (A, b) and apply (A, b) to F_i to get $F_i^{'}$.

5. At the end of the previous step the average of all the $F_i^{'}$ is taken and it will be set to $\bar{F}$.

$$\bar{F}_t \leftarrow \frac{1}{N}\sum_{i=1}^{N} F_i^{'}$$

6. Finally, we check for convergence (i.e) $|\bar{F}_t - \bar{F}_{t-1}| \leqslant \epsilon$ for a threshold 'ϵ'

2.2 Affine Transformation

Affine Transformation is the simplest Geometric Transformation in which the collinearity (i.e., all points lying on a line initially still lie on a line after transformation) and ratios of distances (e.g., the midpoint of a line segment remains the midpoint after transformation) are preserved [3]. Affine Transformation is a group of processes which include Reflection, Scale, Rotate and Shear. The MATLAB built-in function ***affine2d*** will compute all the necessary steps explained above based on the parameters of Transformation (A, b).

After computing the Affine Transformation we need to apply these transformation to the image set. The MATLAB built-in function ***imwarp*** will apply the Transformation to the input images and transforms it into (64×64) Normalized Images.

Chapter 3

Principal Components Analysis

3.1 What is PCA?

It is a mathematical procedure that uses an orthogonal transformation to convert a set of values of possibly correlated M variables to a set of K uncorrelated variables, called **Principle Components**. The number of principle components should be less than or equal to number of original values. The transformation is defined such that the 1^{st} principal component shows the most dominant feature or direction of data set; the next shows the next most dominant feature and so on. Dimensionality of original data sets reduced before calculation. First few principal components are selected and the rest are discarded [4].

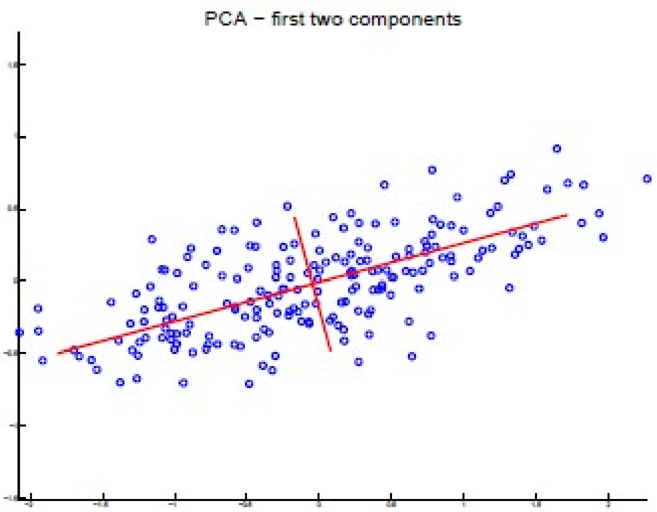

Figure 3.1: PCA - Principal Component Analysis

3.1.1 Geometric Rationale of PCA

Objects are represented as a cloud of n points in a multidimensional space with an axis for each of the p variables. The centroid of the points is defined by the mean of each variable. The variance, V_i of each variable is the average squared deviation of its n values around the mean of that variable.

$$V_i = \frac{1}{n-1}\sum_{m=1}^{n}(X_{im}-\bar{X}_i)^2$$

Degree to which variables are linearly correlated is represented by their co-variance, C_{ij}.

$$C_{ij} = \frac{1}{n-1}\sum_{m=1}^{n}(X_{im}-\bar{X}_i)(X_{jm}-\bar{X}_j)$$

3.2 Face recognition using PCA

Face recognition is a two way process in which first you try to detect the face and then recognize it. Consider, We have a new image of a person and we are trying to match it to other images present in our database. In this case, we are creating an intelligent system which is able to identify the incoming image. The basic idea is to train a model to be able to identify features of a face closest to the features of the faces in the database.

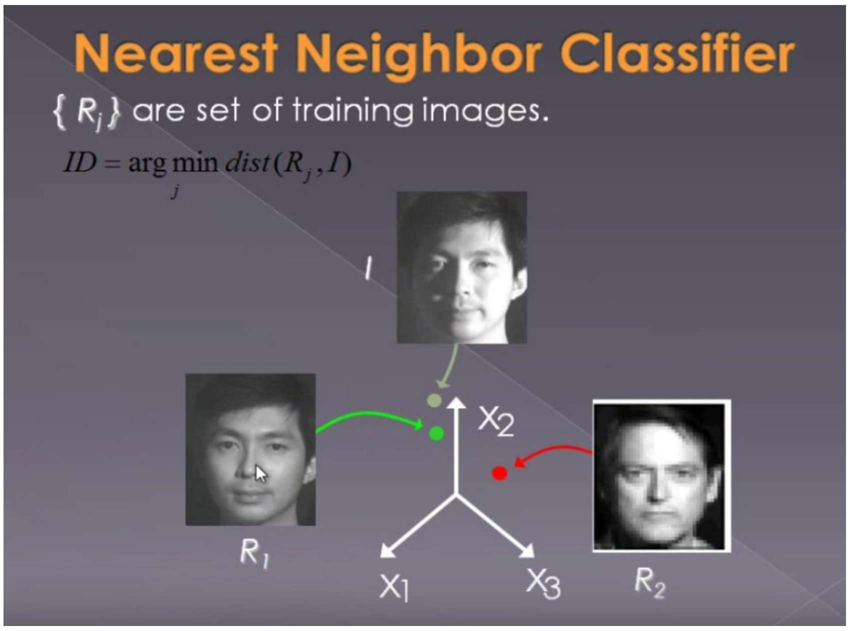
Nearest Neighbor Classifier
$\{R_j\}$ are set of training images.
$ID = \arg\min_j dist(R_j, I)$
I
R_1
R_2
X_1
X_2
X_3

Figure 3.2: Nearest Neighbor Classifier

The basic idea is to treat each pixel as an element of an 1-D vector. If we have an $N \times M$ matrix in which N is the number of images (individuals) and M is the number of pixels of each image. For a 64×64 image $M = 4096$ (the number of features). We write the intensity of each pixel for each image in one row and we have $N \times 4096$ Matrix. This will be so complicated in terms of computation and time consuming. By using PCA we can reduce the features from N to k for N images. k is a linear combination of M feature vectors. For example a_1 is a linear combination of $X_{11}, X_{12}, \ldots X1M$ and a_2 is a linear combination of $X_{21}, X_{22}, \ldots X2M$ with different weights assigned to it, where $a_1 \ldots a_k$ are the **Principal Components**. Principal Components are orthogonal to each other and correlation between them is zero. The criteria to select the k value in PCA should be $k \leq M$ ($k = M$ is the worst case)[2].

$$\begin{bmatrix} X_{11} & X_{12} & \ldots & X1M \\ X_{21} & X_{22} & \ldots & X2M \\ \vdots & \vdots & \ddots & \vdots \\ X_{N1} & X_{N2} & \ldots & XNM \end{bmatrix} \Longrightarrow \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1M} \\ a_{21} & a_{22} & \ldots & a_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \ldots & a_{NM} \end{bmatrix} \Longrightarrow \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix}$$

To perform PCA, we have to compute the co-variance of the the $N \times M$ matrix. So, for each columns of the matrix we find the mean and subtract the mean from each element of the column vector. Then the Covariance Matrix is computed as,

$$\sum = \frac{1}{p-1} D^T D$$

where D is the $N \times M$ Matrix. Next step is to calculate the eigen vectors V from the covariance matrix corresponding to the eigen values λ. The resulting Eigen vector Matrix is of $M \times N$. We reduce the dimensionality of this Matrix by keeping the n eigen vectors corresponding to the n largest eigen values.

3.2.1 Computing Eigenfaces

We can convert the Principal Components to images by reversing the concatenation process from $1 \times M$ to $\sqrt{M} \times \sqrt{M}$ (for a square image). This images are called Eigen faces as they are similar to the input images[2].

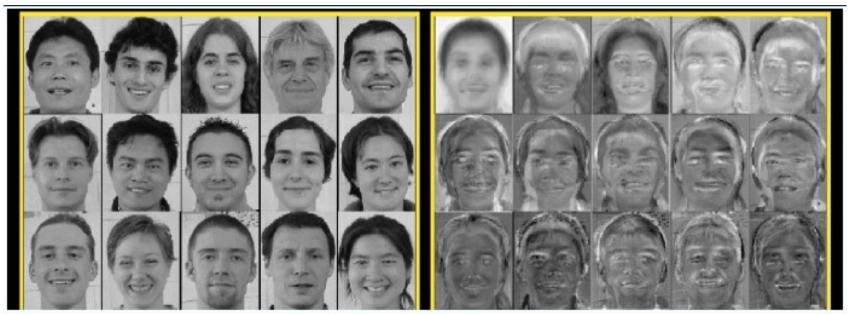

Figure 3.3: Examples for Eigen faces

3.3 Modeling a Recognition System

3.3.1 For Training

- Pre-process (Normalization) the Images of a database. The size of all images of the database should be same and should primarily have the facial features present in it.
- Make all $n \times m$ Images into $1 \times d$ Image vectors where $d = mn$ and store it as matrix.
- Take the mean of all columns of the matrix and remove the mean prior from the matrix (Subtract the mean with all values of matrix).
- Calculate the eigen vectors by computing the covariance matrix.
- Select k best eigen vectors (principal components) to represent the whole training set.
- Get weight vector for each image in the database.

3.3.2 For Testing

- Input image of an unknown person.
- Normalize this image.
- Convert the input as image vector.
- Project the normalized image vector onto the eigen space (PCA space).
- Find weight vector of the input image.
- Calculate the euclidean distance between the input weight vector and all weight vectors of training set.

- If distance is less than the threshold (found by hit and trial) show result, else it's a new image.

Chapter 4

Results

4.1 Normalization

The Results of Normalization has high impact in Face Recognition (using PCA). Since Normalization deals with Geometric Transformation of the Image (in our case Affine Transformation), it is necessary to get precise features to increase the accuracy of Normalization.

One of the main factor which affects the Normalization efficiency is the *Distance between the Camera and the Face.* In the given set of Input Faces and Features, some faces are zoomed in and some are zoomed out. This results in loss of important features in some faces. The results of Normalization is shown in the figure below.

(a) Good Normalization **(b)** Bad Normalization

Figure 4.1: Output of Normalization

4.2 Face Recognition

The main factor affecting PCA - Face Recognition is Normalization.In order to achieve better results in Face Recognition we took the sets of Faces which are closely identical in terms of distance (as discussed above) and Normalized each set separately. The Recognition Accuracy is calculated only for the First Match. We achieved better accuracy in PCA and the results are shown below.

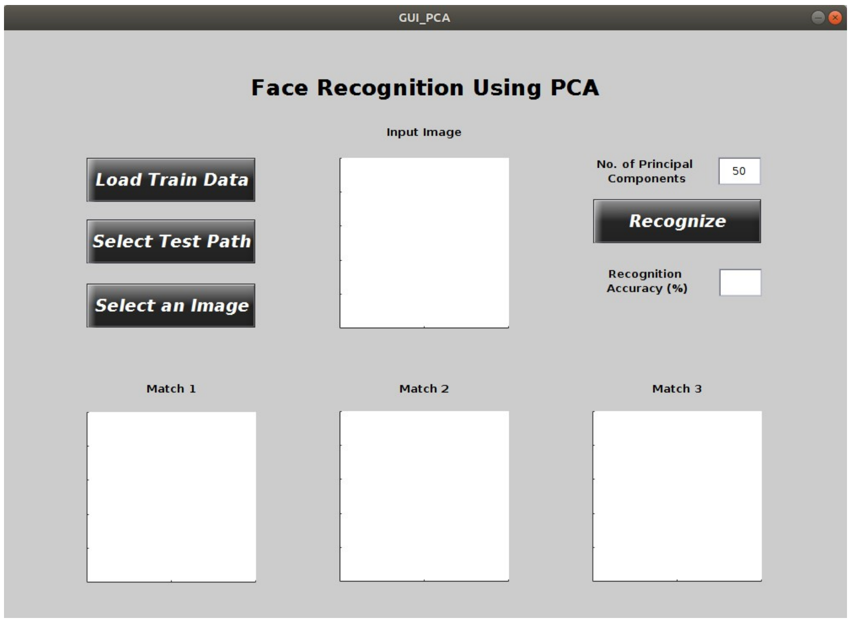

Figure 4.2: GUI Layout

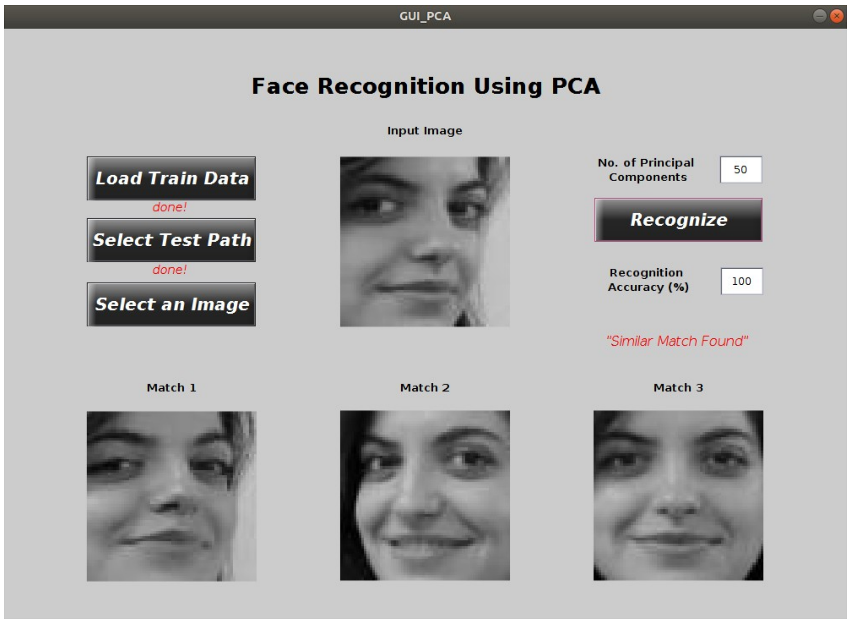

Figure 4.3: Recognition Accuracy 100%

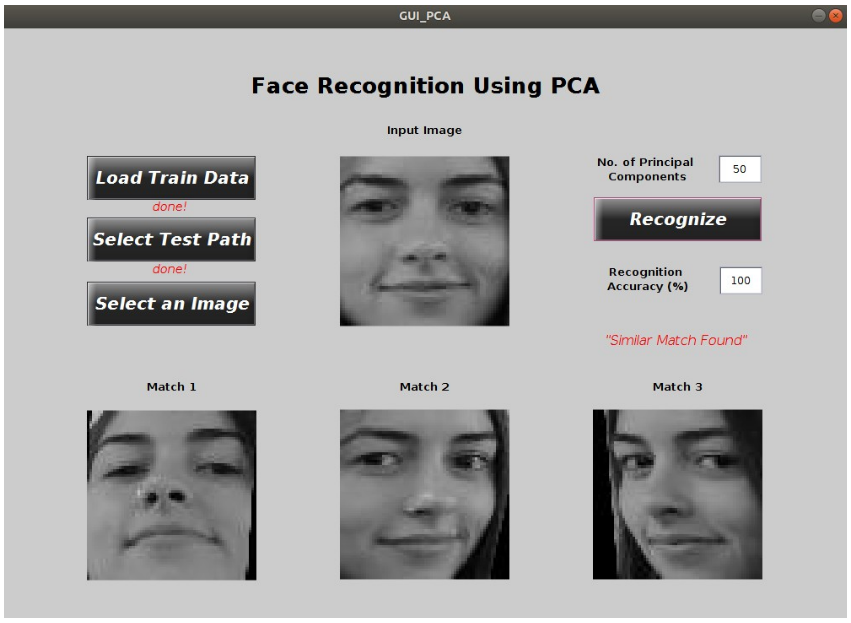

Figure 4.4: Recognition Accuracy 100%

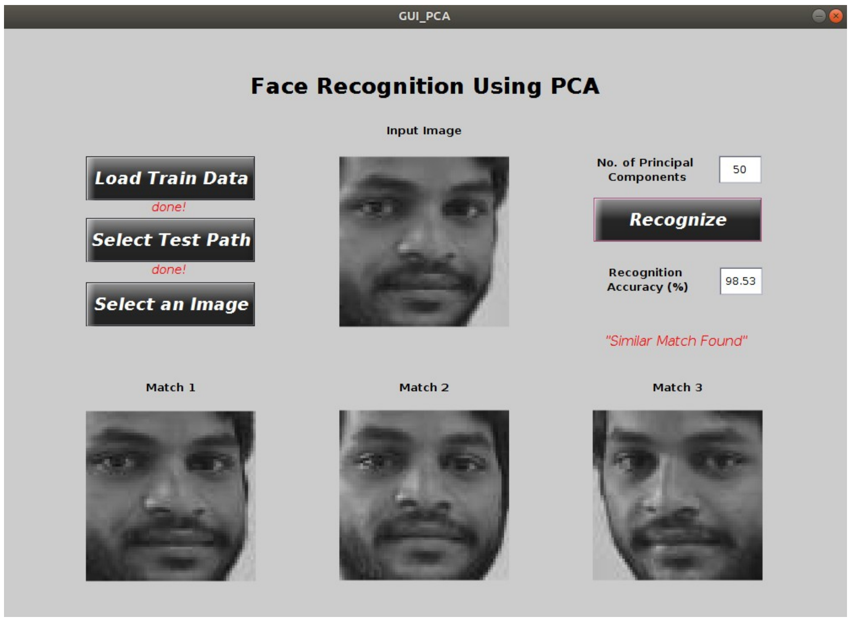

Figure 4.5: Recognition Accuracy 98.53%

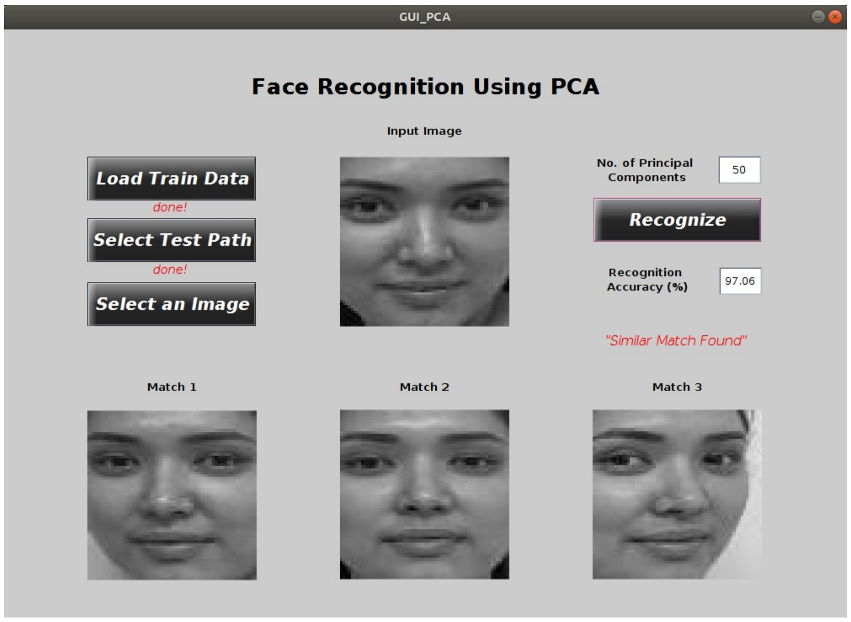

Figure 4.6: Recognition Accuracy 97.06%

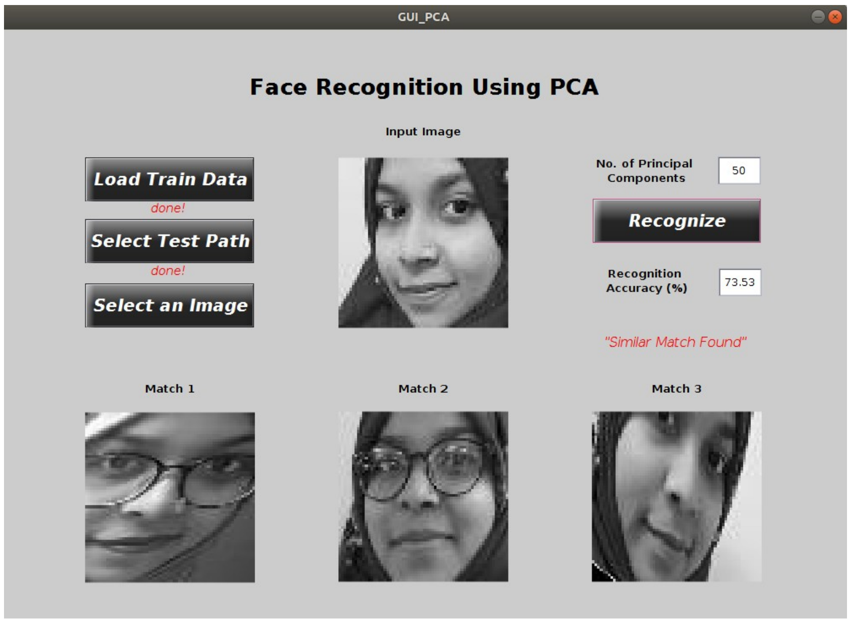

Figure 4.7: Recognition Accuracy 73.53%

Chapter 5

Conclusion

5.1 Conclusion

The main objective of this project was to understand the application Mathematical concepts in Real life. The quality of images to be processed is the main factor in determining the performance of PCA based Face Recognition System. Because of lack of similarity in the Face data, Normalization for entire Face data has less precision. Higher the Precision of Normalization, higher the Accuracy of Face Recognition.

5.2 What we learned...?

1. We learned in detail and understood how to implement the concepts of PCA and SVD Face Recognition Techniques.
2. The techniques of Geometric Transformations especially Affine Transformation is clear and applicable in further aspects.
3. We learned MATLAB-gui and some in-built functions like 'pinv' (to find the sudo-inverse), 'affine2d' (to compute affine parameters), 'imwarp' (to apply transformation to the images), etc.
4. By now we are good in using MATLAB (for programming) and Latex Editor (for documentation).

Bibliography

[1] M. Kaur and R. Vashisht, "Comparative Study of Facial Expression Recognition Techniques," in *International Journal of Computer Applications*, January 2011.

[2] D. Sidibe, "Applied mathematics," in *Lecture Note*, 2018.

[3] Y. Wang, "Geometric transformations: Warping, registration, morphing," in *Lecture Note : Polytechnic University, Brooklyn, NY*, 2009.

[4] K. P. F.R.S., "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.