Afr. J. Comp. & ICT
Vol. 2, No. 2, 2009, pp.21 – 40
ISSN 2006 – 1781

FACE IMAGE PROCESSING, ANALYSIS AND RECOGNITION ALGORITHMS FOR ENHANCED OPTIMAL FACE RECOGNITION SYSTEMS DESIGN:A COMPARATIVE STUDY

Vincent AndrewAkpan*, *Member, IEEE*
Department of Electrical & Computer Engineering,
Aristotle University of Thessaloniki,
GR-54124 Thessaloniki,
Greece.
*Email (Corresponding author): avincent@ieee.org

&

Reginald R. A. O. Osakwe
Department of Physics,
The Federal University of Petroleum Resources,
P.M.B. 1221, Effurun, Delta State,
Nigeria.
Email: reginaldosakwe2003@yahoo.co.uk

Received 31 October, 2009

ABSTRACT

In this paper, the variational properties of face images are investigated with the objective of proposing suitable algorithms and technique for enhanced optimal face recognition system design. Ten different face image samples each of forty-eight different individuals taken under different light intensities were processed. The resulting four hundred and eighty face image samples were split into a training set which constitutes the database of known face images and a test set which constitutes unknown faces. The first face images of each of the forty-eight individuals were evaluated and analyzed using thirteen different mathematical algorithms available from MATLAB® image processing toolbox. The results of the analyses show significant differences for all the face image samples considered. Based on the results from the face image analyses; two algorithms: 1) the principal component analysis (PCA) and 2) the eigenfaces were proposed. The two algorithms were applied simultaneously for enhanced optimal face recognition, where a particular face must satisfy the two algorithms for recognition. The simulation results show that the proposed face image evaluation techniques as well as the proposed PCA and the eigenfaces algorithms recognizes a known face image or rejects an unknown face based on the database contents to a high degree of accuracy. The combination of the proposed face recognition strategy can be adapted for the design of on-line real-time embedded face recognition systems for public, private, business, commercial or industrial applications.

Keywords: Eigenfaces, face image evaluation, principal component analysis, variational properties.

I. Introduction

Faces and fingerprints recognitions have become an interesting field of research in recent times due to the increased identity fraud in our society reaching unprecedented proportions. To this end much emphasis and research have been placed on emerging algorithms and automated technologies for face and fingerprint identification and recognition. Ideally a face or fingerprint detection system should be able to take a new face or fingerprint and return a name identifying that person if the person exists in the database, or return an error of non-existence if there is no correspondence of such face or fingerprint. Statistically, faces and fingerprints can also be very similar. Using standard image sizes and the same initial conditions, a system can be built that looks at the statistical relationship of individual pixels; so that from signal processing point of view, the face and fingerprint recognition problem boils down essentially to the identification of an individual based on an array of pixel intensities and in addition to whatever information can be gleaned from the image to assign a name to the unknown set of pixel intensities. Characterizing the dependencies between pixel values becomes statistically a digital signal processing problem.

The theoretical foundations of digital signal processing were laid by Jean Baptiste Joseph Fourier [1]. Major theoretical developments in digital signal processing theory were made in the 1930s and 1940s by Nyquist and Shannon, among others (in the context of digital communication) and by the developers of the Z-transform (notably by Zadeh and Ragazzini in the West, and Tsypkin in the East) [2], [3]. The history of applied digital signal processing (at least in the electrical engineering world) began around the mid-1960s with the invention of the fast Fourier transform (FFT) [2], [4]. However, its rapid development started with the advent of microprocessors in the 1970s [5], [6]. In the present discussion, for space limitation and simplicity, we shall limit our investigations to face recognition since it might be possible to apply the proposed algorithms and techniques to fingerprints.

Several approaches to the overall face identification and recognition [7], [8], [9] problems have been devised over the years with considerable attention paid to methods by which effective face identification and recognition can be replicated. The intuitive way to do face recognition is to look at the major features of the face and compare them to the same features on other faces. Extensive research in face statistical data compression technique using principal component analysis (PCA) technique [10], face features comparisons and recognition based on the eigenfaces technique [11], and the direct face classification scheme using discrete cosine transform (DCT) coefficients [12], [13] have been reported. In [14], a comparative study of face recognition algorithms has been conducted and several issues such as: 1) preprocessing and cropping of the faces to equal and consistent sizes, database creation for the preprocessed images, and 2) a major problem in the amount of light intensity on the original face images have been reported. The comparison of the eigenfaces algorithm with the fisherfaces algorithm [15], [16] have also been reported; where it has been recommended that both algorithms can be used for accurate face recognition results except with higher computational load in the later which might limit its application in real-time systems. In addition to the above issues, the face data compression techniques has provided additional challenges to the face identification and recognition problem such as: 1) generating worse face images which are different from the original image [10], and 2) the difficulties associated with the restoration of the original face image leading to the inclusion of color information. This second problem has led to the new fractal image encoding research fields [17], [18], [19].

In general, the face identification and recognition problem becomes more complicated and computationally intensive as the number of the face images increases especially when used in real-time [20]. As reported in [11], a 90% success in face recognition is achieved using the eigenfaces algorithm and this algorithm has been widely used in research papers [20], [21], [22] but with greater light intensity on well-taken face images.

In the present study, we seek to investigate the variational properties of face images for enhanced optimal identification and recognition by evaluating the statistical data derived from a well-reduced form of each of the original face images under different angular positions and light intensities. Based on the results of the evaluation we propose two algorithms based on the: 1) PCA and 2) eigenfaces algorithms.

The rest of the paper is organized as follows: In Section II we present the material and methodology employed in this study. The evaluation algorithm, its analysis and results are also presented in this section. The proposed PCA and the eigenfaces algorithms are formulated in Section III. The applications of these two algorithms for enhanced face recognition together with their results are given in Section IV. A brief conclusion and recommendation are presented in Section V.

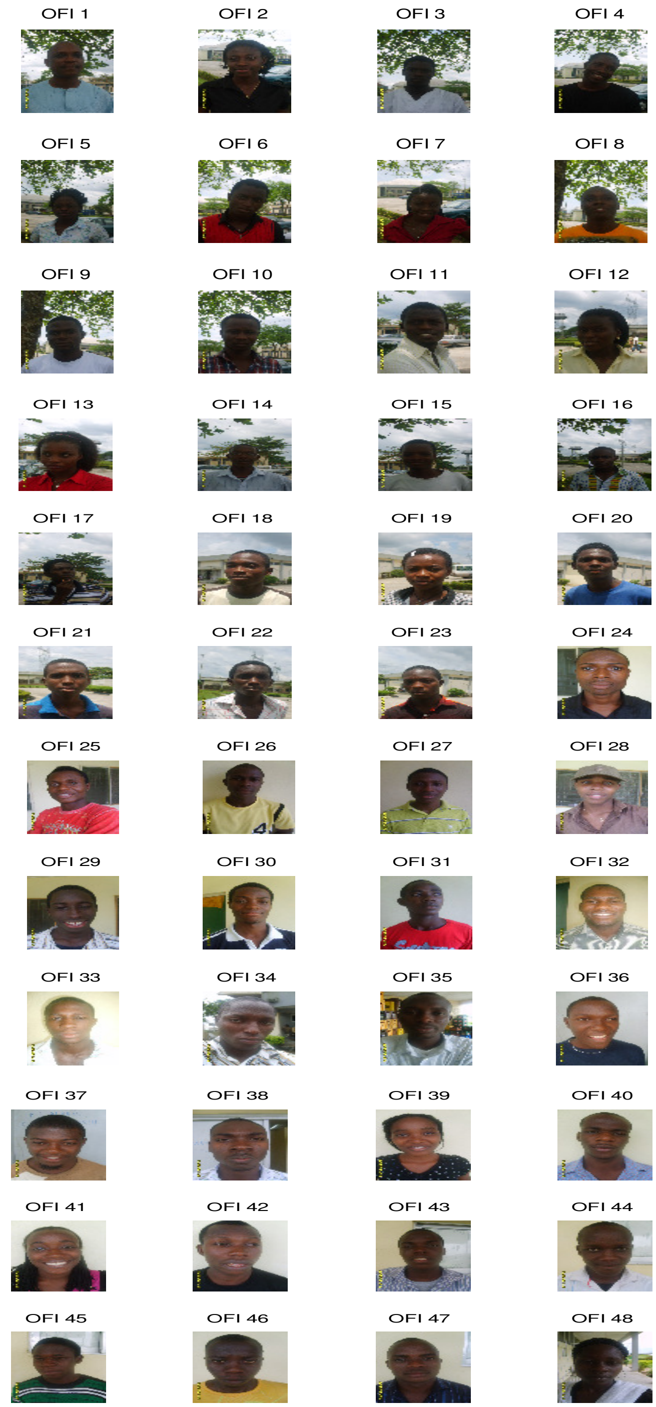


Fig. 1. Original face image (OFI) samples.

II. Face Image Processing, Evaluation and Analysis

A. *Materials and Methodology for Obtaining the Faces*

The material employed in this study were four hundred and eighty (480) joint photographic expert group (JPG) face image samples, shown in Fig. 1) taken with a Samsung® DIGIMAXS500/CYBER530 digital camera [23]. The 480 face image samples consist of 10 different face samples of 48 individuals (students) taken at The Federal University of Petroleum Resources, Effurun, Delta State, Nigeria. Out of the 480 face image samples, the first 75% (360 face images) shown in Fig. 1 were used to form the database of known faces images while the remaining 25% (120 face images) were reserved for investigating the performance and validation of the proposed face recognition algorithms. Due to space limitation, only the first out of the 10 face image samples for each of the 48 individuals are shown here. The proposed evaluation algorithm in this study are written, compiled and analyzed using MATLAB® 2009a from The MathWorks Inc. [24] software running on an Inter® Core™ Duo CPU E6750@2.67GHz computer on a Windows™ XP platform.

B. *Processing of the Face Images*

The processing of images is necessary to enhance fast and accurate image identification and recognition. Moreover, when the numbers face images are relatively large, each face image must be pre-processed by reducing its size and converting the image to a format requiring less memory storage capacity. In other to reduce the memory consumption for storing the face images, all the JPG face images were converted and resized to portable graymap (PGM) images of height 114 pixels and width 114 pixels, filtered to remove noise using an adaptive Wiener filter (AWF), and finally resized from their original height 640 pixels and width 480 pixels to 114 by 114 pixels respectively.

The PGM filtered and resized face images were then used for analysis in other to identify and recognized any face image within the database. The pre-processed face images (PPFI) used for analysis are the shown in Fig. 2 (note: rather than presenting all the 480 face image samples used in this work, only the first out of the ten face images for each of the 48 individual is shown in Fig. 2 for space economy). The pre-processing algorithm for the face images are given in number one through number six in the algorithm of TABLE I. In the following subsection we present the evaluation algorithms used in the analysis of each face image.

TABLE I

THE FACE IMAGE EVALUATION ALGORITHMS USING MATLAB®

```
% 1. Load the known face image from database
  (Only once for faster comparison)
  load image1_facedb  % Face database
  load image2_pca     % PCA data database

% 2. Incoming face image (INFI) for recognition.
  In this study, this image is randomly from the
  480 original face images (OFI) in terms of
  their number sequence between 1 and 480.
  src_face  =  input('Please  enter  any  number
  between 1 and 480 to recognize image =  ');

%  3.  Evaluate  the  input  face  image  identity
  number to check if it is within the database
  limit.  Otherwise  prompt  the  user  to  enter  a
  valid identity number.
  Image5_evalsrc;

% 4. Convert the JPG face images to PGM to reduce
  storage memory requirements and display result.
  i_face = imread(strcat('face',num2str(inface),
  ...'.JPG'));
  imwrite(i_face,'I_Fac.pgm');
  I_Fac = imread('I_Fac.pgm');
  figure(1), imshow(i_face); title('OFI ##');

% 5. Filtering and resizing the PGM face images.
  AWF_Face = wiener2(I_Fac,[3 3],0.8);
  rf_Face  = imresize(AWF_Face,([114 114);
  figure(2), imshow(rf_Face),
  title('PPFI ##')

%  6.  Reshape  the  filtered  image  matrices  to
  comply    with    MATLAB's    toolboxes    for
  analysis.
  [DIM, LEN] = size(rf_Face); nn=3;
  re_Face    = reshape(rf_Face,DIM*LEN/nn,nn);
  LEN_re     = length(re_Face);
  new_reFace = double(re_Face);

% 7. Analysis of the PPFI for each face image.
  STD_Face    = std(new_reFace,1,2);
  M_Face      = mean(new_reFace);
  S_Fcae      = sum(new_reFace);
  SVD_Face    = svd(new_reFace);
  DCT_Face    = dct2(new_reFace);
  VAR_face    = sqrt(sum(new_reFace.^2)/LEN_re);
  FFT_Face    = fftshift(new_reFace,1);
  Q_Face     = detrend(new_reFace,'linear','true');
  drf_Face    = double(rf_Face);
  [VEC,VAL]   = eig(drf_Face);
  CVM_Face    = cov(new_reFace);
  VCVM_Face   = diag(cov_face);
  SDCVM_Face = sqrt(diag(cov(new_reFace);
```

TABLE II

RE-ANALYZING THE ANALYTICAL RESULTS OF NUMBER 6 IN TABLE I FOR DISTINGUISHING PROPERTIES BETWEEN IMAGES

```
STD_anal   = mean(STD_Face);       % For Fig. 3(a)
M_anal     = mean(M_Face);         % For Fig. 3(b)
S_anal     = mean(S_Face);         % For Fig. 3(c)
SVD_anal   = mean(SVD_Face);       % For Fig. 3(d)
DCT_anal   = sum(mean(DCT_face));  % For Fig. 3(e)
VAR_anal   = mean(S_face);         % For Fig. 3(f)
FFT_anal   = mean(mean(FFT_Face));% For Fig. 4(a)
Q_anal     = sum(diag(Q_face));    % For Fig. 4(b)
EVEC_anal  = sum(diag(Vec_face));  % For Fig. 4(c)
VAL_anal   = sum(mean(Val_face));  % For Fig. 4(d)
CVM_anal   = mean(mean(CVM_Face));% For Fig. 4(e)
VCVM_anal  = mean(VCVM_Face);      % For Fig. 4(f)
SDCVM_anal = mean(SDCVM_Face);     % For Fig. 4(g)
```

Due to the fact that our ambition in the present investigation is high, we wish to stress here that the face images were not *cropped* so as to focus only on the foreface as can be seen in Fig. 1. This is to allow for the investigation and verification of the efficiency of the face identification and recognition strategies proposed in the present study; where a particular face sample can be taken amidst a crowd, criminal scenes or abnormal scenarios either with full light intensity and vice versa.

1) Evaluation of the Face Images using MATLAB®

Each of the face images were analyzed using thirteen different analysis algorithms available from MATLAB® in other to deduce the distinguishing and variational properties of each face image sample based on the results of their analysis. The thirteen variational properties investigated are summarized by the MATBLAB® algorithms listed in number seven of TABLE 1 namely: standard deviation (STD_Face), mean (M_Face), summation (S_Face), singular value decomposition (SVD_Face), discrete Fourier transform (DCT_Face), basic variance (VAR_Face), fast Fourier transform (FFT_Face), compute the residual by subtracting the column mean from each column of the face image matrix (Q_Face), eigenvectors (VEC) and

Fig. 2. The pre-processed and AWF face images (PPFI) samples.

eigenvalues (VAL), covariance matrix (CVM_Face), variance of the covariance

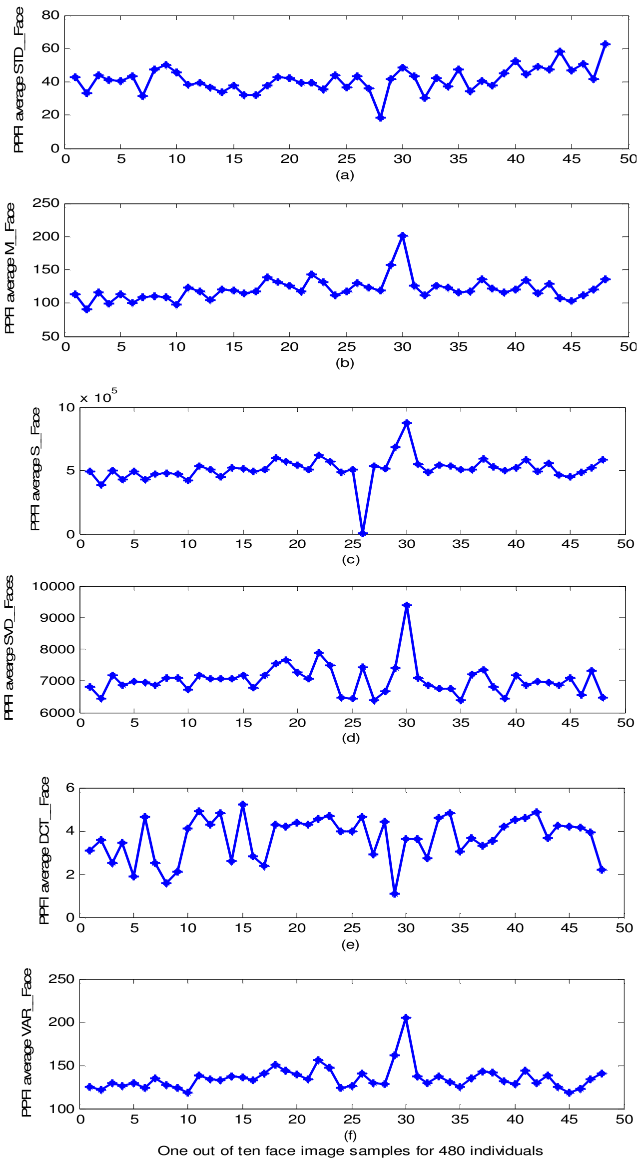


Fig. 3. The first part of the analytical results of the evaluation algorithm of TABLE II.

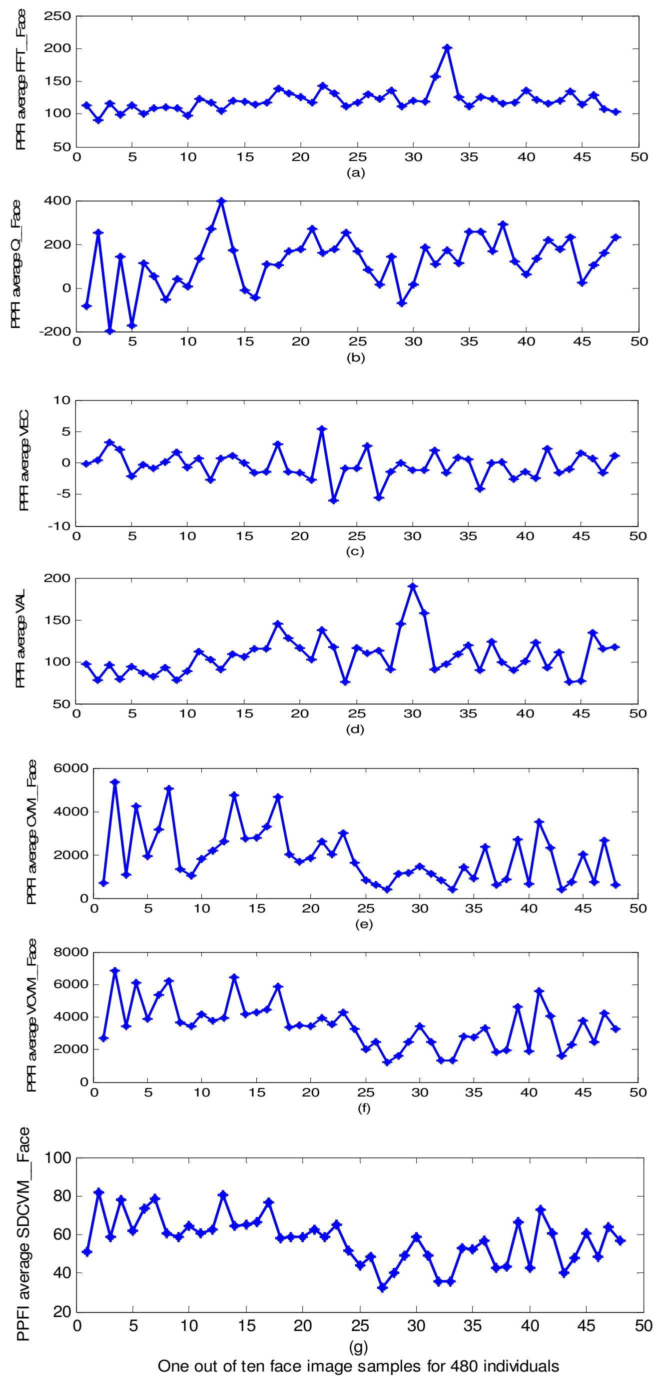


Fig. 4. The second part of the analytical results of the evaluation algorithm of TABLE II.

matrix (VCVM_Face), and STD of the covariance matrix (SDCVM_Face). These eleven parameters are selected because they form an integral part of the PCA, eigenfaces and eigenfisher algorithms for face recognition, image/signal processing and image data compression techniques.

C. Analysis of the Face Images

The analysis considered in the present study for evaluating all the thirteen parameters to investigate the variational properties of the face images are summarized in TABLE 2. The analyses were performed on the face images shown in Fig. 2. The results of these analyses are presented in Fig. 3 (a) – (f) and Fig. 4 (a) –

(g). The graphs of Fig. 3 and Fig. 4 show the variations in the properties for each of the face image samples evaluated in Section II-C. The DCT_Face, Q_Face, and VEC results of Fig. 3 (e), Fig. 4 (b) and Fig. 4(c) respectively show well distributed variations of the face images. Also the CVM, VCVM_Face and SDCVM of Fig. 4 (e), (f) and (g) respectively clearly distinguish each face image.

III. The Face Image Evaluation and Recognition Algorithms

A. *A. The PCA Algorithm Based on the SVD Algorithm*

Principal component analysis (PCA) is an orthogonal linear transformation tool that transforms data to a new coordinate system such that the greatest variance by any first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on [11], [25], [26]. PCA is theoretically the optimum transform for given data in least square terms and it involves a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible [25]. The PCA is the simplest of the true eigenvector-based multivariate analyses. Often, the PCA operation can be thought of as revealing the internal structure of the data in a way which best explains the variance in the data. Furthermore, PCA provides a strategy on how to reduce a complex data set to a lower dimension to reveal the sometimes hidden, simplified structure that often underlie it.

Two methods, with their merits and demerits, have been proposed in [25] for the evaluation of PCA using the: 1) covariance method and 2) correspondence analysis. Based on the results of the face evaluation recognition of TABLE I of Section II-C presented in Section II-D, we proposed a SVD algorithm which will utilize the properties of the covariance matrix to achieve faster and more concise computation of the principal components of the face images. In the following we present the SVD algorithm to facilitate the solution of the PCA algorithm. The mathematical intuition of the SVD is detailed in [26] and [27]. While the SVD formulated in [26] is directly related to PCA; the SVD formulation in [27] avoids PCA directly. The approach presented here combines ideas from [26] and [27].

1) *The formulation of the PCA Algorithm*

Given an arbitrary $m \times n$ matrix Φ, the objective of the PCA is to find some orthonormal matrix P by constructing another matrix $\Theta = P\Phi$ such that $C_E \equiv \frac{1}{n-1}\Theta\Theta^T$ is diagonalized. The rows of P are the principal components of Φ; where *m* is the number of measurement types, *n* is the number of samples (measurement trials), and is the C_E covariance matrix (an explanation of C_E is provided in *Corollary 1* below). Thus, by rewriting C_E as

$$\left.\begin{aligned} C_E &= \frac{1}{n-1}\Theta\Theta^T = \frac{1}{n-1}(P\Phi)(P\Phi)^T = \frac{1}{n-1}P\Phi\Phi^T P^T \\ &= \frac{1}{n-1}P(\Phi\Phi^T)P^T = \frac{1}{n-1}PAP^T \end{aligned}\right\} \quad (1)$$

where $A \equiv \Phi\Phi^T$ and by APPENDIX II, *A* is a symmetric matrix.

Then, by the theorems of APPENDICES III AND IV, we recognize that the symmetric matrix *A* is diagonalized by an orthonormal matrix of its eigenvectors. According to the theorem of APPENDIX IV, we can express *A* as:

$$A = EDE^T \quad (2)$$

where D is a diagonal matrix and E is a matrix of eigenvectors of A arranged as columns. The matrix A has $r \le m$ orthonormal eigenvectors where r is the rank of the matrix A. the rank of A is less than m when A is *degenerate* or all data occupy a sub-space of dimension $r \le m$. The issue of degeneracy is handled here to maintain the orthogonality, we select an $(m-r)$ additional orthonormal vectors to "fill up" the matrix E. we note that these additional vectors do not effect the final solution because the variances (see *Corollary 1*) associated with these directions are zero.

To obtain the principal components (*PCs*) of Φ, we select the matrix P in (1) as a matrix where each p_i is an eigenvector of $A = \Phi\Phi^T$. By selecting and substituting $P = E^T$ into (2), we obtain:

$$A = P^T DP \quad (3)$$

Using (3) and the theorem of APPENDIX I (i.e. $P^{-1} = P^T$), we recomputed (1) as follows:

$$\left.\begin{aligned} C_E &= \frac{1}{n-1} PAP^T = \\ &\frac{1}{n-1} P(P^T DP)P^T = \frac{1}{n-1}(PP^T)D(PP^T) \\ &= \frac{1}{n-1}(PP^{-1})D(PP^{-1}) = \frac{1}{n-1} D \end{aligned}\right\} \quad (4)$$

As it is evident in (4), the choice of P diagonalizes C_E as we desired. Finally, we summarize the PCA procedure as follows: 1) the PCs of Φ are the eigenvectors of $\Phi\Phi^T$ or the rows p_i of P; and 2) the i^{th} diagonal of C_E is the variance of Φ along p_i.

Thus, computing the *PCs* of Φ, reduces to 1) subtracting off the mean of each measurement type; and 2) computing the eigenvectors of $\Phi\Phi^T$.

2) The formulation of the SVD Algorithm

Here, let Φ be an arbitrary $n \times m$ matrix and $\Phi^T\Phi$ be a rank *r*, square and symmetric $m \times m$ matrix. Also let $\{\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_r\}$ be the set of *orthonormal* $m \times 1$ eigenvectors (see APPENDIX IV second part for proof of orthonormality) with associated eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_r\}$ for the symmetric matrix $\Phi^T\Phi$ such that $(\Phi^T\Phi)\hat{\alpha}_i = \lambda_i \hat{\alpha}_i$; $\varsigma_i \equiv \sqrt{\lambda_i}$ are positive real and termed the singular values; and $\{\hat{\beta}_1, \hat{\beta}_2, \ldots \hat{\beta}_r\}$ be the set of $n \times 1$ vectors defined by:

$$\hat{\beta}_i \equiv (1/\varsigma_i)\Phi\hat{\alpha}_i \quad (5)$$

where i and j are the length of *m*. Before stating the last two additional properties required for deriving the SVD algorithm, it is necessary to first consider the Theorem below for the orthonormality and orthogonality properties of Φ.

Theorem 1:

For any arbitrary $m \times n$ matrix Φ, the symmetric matrix $\Phi^T\Phi$ has a set of orthonormal eigenvectors of $\{\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_r\}$ and a set of associated eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_r\}$ (see APPENDIX IV). The set of vectors $\{\Phi\hat{\alpha}_1, \Phi\hat{\alpha}_2, \ldots, \Phi\hat{\alpha}_r\}$ then form an orthogonal basis (see APPENDIX I for proof), where each vector $\Phi\hat{\alpha}_i$ is of length $\sqrt{\lambda_i}$. Based on the dot product of any two vectors [26], we have the following:

$$(\Phi\hat{\alpha}_i)\cdot(\Phi\hat{\alpha}_j) = (\Phi\hat{\alpha}_i)^T(\Phi\hat{\alpha}_j) = \hat{\alpha}_i^T\Phi^T\Phi\hat{\alpha}_j$$
$$= \hat{\alpha}_i^T(\lambda_j\hat{\alpha}_j) = \lambda_i\hat{\alpha}_i \cdot \hat{\alpha}_j$$

So that $(\Phi\hat{\alpha}_i)\cdot(\Phi\hat{\alpha}_j) = \lambda_j\delta_{ij}$ (6)

where j is the length of *n*.

Equation (6) arises due to the fact that the set of eigenvectors of Φ is orthogonal resulting in the Kronecker delta δ_{ij}. Alternatively, (6) can be expressed from [26] as:

$$(\Phi\hat{\alpha}_i)\cdot(\Phi\hat{\alpha}_j)=\begin{cases}\lambda_j & i=j\\ 0 & i\neq j\end{cases} \quad (7)$$

which states that any two vectors in the set are orthogonal. The second property arises from (6) by realizing that the length squared of each vector can be defined as [26]:

$$\|\Phi\hat{\alpha}_i\|^2=(\Phi\hat{\alpha}_i)\cdot(\Phi\hat{\alpha}_i)=\lambda_i \quad (8)$$

With Theorem 1, we can define the following two properties (see also APPENDIX I for proof of orthogonality):

$$\hat{\alpha}_i\cdot\hat{\alpha}_j=\begin{cases}1 & if \quad i=j\\ 0 & otherwise\end{cases} \quad (9)$$

$$\|\Phi\hat{\alpha}_i\|=\varsigma_i \quad (10)$$

Finally, we define the "value" version of SVD as a restatement of (5) which can be expressed as:

$$\Phi\hat{\alpha}_i=\varsigma_i\hat{\beta}_i \quad (11)$$

Equation (11) implies that Φ multiplied by an eigenvector $\hat{\alpha}_i$ of $\Phi^T\Phi$ is equal to a positive scalar ς_i times another vector $\hat{\beta}_i$. The set of eigenvectors $\{\hat{\alpha}_1,\hat{\alpha}_2,\ldots,\hat{\alpha}_r\}$ and the set of vectors $\{\hat{\beta}_1,\hat{\beta}_2,\ldots\hat{\beta}_r\}$ can both be orthonormal sets or bases in *r*-dimensional space.

To obtain the matrix form of the SVD, we begin by constructing the following three new matrices: *Ψ*, *A* and *B* as:

$$\Psi=\begin{bmatrix}\varsigma_1 & 0 & 0 & 0 & 0 & 0\\ 0 & \ddots & 0 & 0 & 0 & 0\\ 0 & 0 & \varsigma_r & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & \ddots & 0\\ 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix} \quad (12)$$

$$\left.\begin{aligned}\mathrm{A}&=[\hat{\alpha}_1 \quad \hat{\alpha}_2\ldots\hat{\alpha}_m]\\ B&=[\hat{\beta}_1 \quad \hat{\beta}_2\ldots\hat{\beta}_n]\end{aligned}\right\} \quad (13)$$

where Ψ is a $n\times m$ diagonal matrix with a few non-zero values $\varsigma_1,\varsigma_2,\ldots,\varsigma_r$ such that $\varsigma_1\geq\varsigma_2\geq\ldots\geq\varsigma_r$ are the rank-ordered set of singular values; and matrices *A* and *B* are $m\times m$ and $n\times n$ respectively and the corresponding vectors are indexed in the same rank order. To handle the issue of degeneracy in A and B, we append an additional $(m-r)$ and $(n-r)$ orthonormal vectors to "fill up" the matrices for *A* and *B* respectively.

The mathematical intuition behind the construction of the matrix forms of (12) and (13) is to express all the *n* "value" equations of (11) into a single invertible form; so that the matrix multiplication for all the vectors can be accomplished in a single *n* iteration.

Each pair of associated vectors $\hat{\alpha}_i$ and $\hat{\beta}_i$ in (13) are stacked in the i^{th} columns along their respective matrices. The corresponding singular value ς_i is placed along the diagonal (i.e. the ii^{th} position) of Ψ in (12). So that (11) degenerates into the following form expressible as:

$$\Phi\mathrm{A}=\mathrm{B}\Psi \quad (14)$$

where each column of *A* and *B* in (13) performs the "value" version of the decomposition using (11) subject to (10). Since A is orthogonal, multiplying both sides of (10) by $\mathrm{A}^{-1}=\mathrm{A}^T$ leads to the final form of the decomposition which can be expressed as:

$$\Phi=B\Psi A^T \quad (15)$$

From (11), the SVD technique implies that: 1) any arbitrary matrix Φ can be converted to an orthogonal matrix B, a diagonal matrix Ψ and another orthogonal matrix A; 2) the columns of matrix *A* contain the eigenvectors of $\Phi^T\Phi$.

In the following discussion, we consider the relationship between PCA and SVD; and then we present a summary of the procedure for computing the *PCs* of Φ using SVD.

3) The Principal Components of the SVD Algorithm

Considering the original matrix $m\times n$ data matrix Φ such that we can define a new $n\times m$ matrix Θ given below as:

$$\Theta \equiv \frac{1}{\sqrt{n-1}}\Phi^T \qquad (16)$$

where each column of Θ has zero mean. To obtain the PCA using the SVD described above based on (15), it is necessary to first express (16) as a symmetric matrix $\Theta^T\Theta$ which we express below as:

$$\Theta^T\Theta = \left(\frac{1}{\sqrt{n-1}}\Phi^T\right)^T\left(\frac{1}{\sqrt{n-1}}\Phi^T\right) = \frac{1}{n-1}\Phi^{TT}\Phi^T$$

So that $\Theta^T\Theta = \frac{1}{n-1}\Phi\Phi^T = \Gamma_X$ (17)

where Γ_X (similar to C_E in (1) and (4)) is the covariance matrix of Φ. From (15) and (17), it can be seen that $A = \Phi^T$, $B = 1/(n-1)$ and $\Psi = \Phi$. So that by computing the SVD of Θ, the columns of matrix *A*, as in (13), will give the eigenvectors of (17) that are the *PCs* of Φ presented following the discussion of the corollary below.

Corollary 1:

Consider two sets of measurements with zero means which can be expressed as [26]:

$$\mathrm{A} = [\alpha_1, \alpha_2, \ldots, \alpha_n] \text{ and } \mathrm{B} = [\beta_1, \beta_2, \ldots, \beta_n] \qquad (18)$$

where the *n* denotes the numbers of trial measurements. The variance ς_A^2 and ς_B^2 of *A* and *B* can be defined separately as:

$$\varsigma_A^2 = (1/n)\sum\nolimits_{i=1}^{n}\alpha_i^2 \quad and \quad \varsigma_B^2 = (1/n)\sum\nolimits_{i=1}^{n}\beta_i^2 \qquad (19)$$

The covariance between A and B can be generalized as [26]:

$$\varsigma_{AB}^2 = \frac{1}{n}\sum\nolimits_{i=1}^{n}\alpha_i\beta_i$$

The covariance measures the degree of the linear relationship between the two variables A and B. A large positive value indicates positively correlated data while a large negative value denotes negatively correlated data. The absolute magnitude of the covariance measures the degree of redundancy. Additionally properties of ς_{AB}^2 are: 1) $\varsigma_{AB} = 0$, if and only if A and B are uncorrelated; and 2) $\varsigma_{AB}^2 = \varsigma_A^2$, if $A = B$.

By converting *A* and *B* into corresponding row vectors as $\phi_1 = [\alpha_1, \alpha_2, \ldots, \alpha_n]$ and $\phi_2 = [\beta_1, \beta_2, \ldots, \beta_n]$ respectively; the covariance for the unbiased estimate can be expressed as the following dot product matrix computation:

$$\varsigma_{\phi_1\phi_2}^2 = \frac{1}{n-1}\phi_1\phi_2^T \qquad (20)$$

We collect the terms ϕ_1 and ϕ_2 in (20) with additional indexed row vectors $\phi_3, \phi_4, \ldots, \phi_m$ appended to form a new $m \times n$ matrix Φ_τ expressed as:

$$\Phi_\tau = [\phi_1, \phi_2, \ldots, \phi_m]^T$$

so that each row of Φ_τ corresponds to all i^{th} measurements of a particular type while each column corresponds to a set of j^{th} measurements from one particular trial condition. Thus, the covariance matrix Γ_{Φ_τ} can be defined as:

$$(\Gamma_{\Phi_\tau})_{i,j} = (1/n)\Phi_\tau\Phi_\tau^T \qquad (21)$$

It should be note that the ij^{th} element of (21) is the dot product between the vectors of the i^{th} measurement type with the vector of the j^{th} measurement type base on (20). We here some summarize the properties of Γ_{Φ_τ} in (21) as follows: 1) Γ_{Φ_τ} is a square symmetric $m \times n$ matrix; 2) the diagonal terms of Γ_{Φ_τ} are the variance of particular measurement types; and 3) the off-diagonal terms of Γ_{Φ_τ} are the covariance between measurement types. Moreover, Γ_{Φ_τ} captures the correlations between all possible pairs of measurements where the correlation values reflect the noise and redundancy in the measurements data types.

As in (17), $\Theta^T\Theta$ equals the covariance matrix of Φ and the principal components of Φ are the eigenvectors of Γ_X. The computation of the SVD of Θ results in the eigenvectors of $\Theta^T\Theta = \Gamma_X$ contained in the columns of matrix A. This implies that: 1) the columns of A are the principal components of Φ; 2) matrix A spans the row space of $\Theta \equiv (1/\sqrt{n-1})\Phi^T$ and A must also span the column space of $\Theta \equiv (1/\sqrt{n-1})\Phi^T$; 3) by symmetry the columns of B produced by the SVD of $(1/\sqrt{n-1})\Phi^T$ is also the principal components.

Finally, we summarize the SVD procedure for obtaining the *PCs* of Φ as follows: 1) organize the $m \times n$ data of matrix Φ, where m is the number of measurements types and n is the number of samples (measurement trial types); 2) subtract off the mean for each m measurement types; and 3) compute the SVD or the eigenvectors of the covariance matrix Γ_X.

B. *The Formulation of the Eigenfaces Algorithm*

Eigenfaces are basically basis vectors for real faces and it is directly related to the Fourier analysis which reveals that the sum of weighted sinusoids at different frequencies can be recomposed back to the original signal. This implies that the sum of weighted eigenfaces can seamlessly reconstruct a specific face image.

To formulate the eigenfaces algorithm, we convert each face image to a vector Γ_n of length N = image width (114 pixels) multiplied by image height (114 pixels) which forms the measurement data. To increase the information available for the known face images in the database, we use 10 samples of each of the 48 individuals as described in Section II-A above. Here we denote the face image samples as "face space" of dimension *N*, so that the average face image in the face space can be computed as follows:

$$\varphi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n \qquad (22)$$

where *M* is the number of face images in the face space.

The covariance matrix discussed in Section III-B is denoted here for the eigenfaces algorithm as C_E of dimension *N* (number of pixels in the face images) which is expressed here from [11] as:

$$C_E = \frac{1}{M}\sum_{\eta=1}^{M}\Phi_\eta\Phi_\eta^T$$

$$= \frac{1}{M}\sum_{\eta=1}^{M}\begin{pmatrix} \mathrm{var}(p_1) & \cdots & \mathrm{cov}(p_1, p_N) \\ \vdots & \ddots & \vdots \\ \mathrm{cov}(p_N, p_1) & \cdots & \mathrm{var}(p_N) \end{pmatrix}_\eta$$

So that $C_E = AA^T$ (23)

where var and cov are the variance and covariance of their respective arguments (see Corollary 1 for details on variance and covariance); $\Phi_\eta = \Gamma_\eta - \varphi_\eta$ is the difference between each face image under consideration and the average of the face images in the database; $A = [\Phi_1, \Phi_2, \ldots, \Phi_M]$; and p_η is the pixel of face η. Thus the eigenfaces algorithm reduces to the evaluation of the eigenvalues of C_E, which can be computationally intensive due to the large value of *N*.

By using the data transformation properties provided by the PCA just described above, the *N*-dimensional covariance matrix (23) can be reduced to an *M*-dimensional matrix corresponding to the number of face images in the database. By constructing a new $M \times M$ matrix *L* as follows:

$$L = A^T A \qquad (24)$$

According to the PCA [26], since we have only *M* images; there will be only *M* non-trivial eigenvectors.

Theorem 2:

Suppose that we can define the right hand side of (24) as:

$A^T A v_i = \mu_i v_i$

where μ_i is scalar and v_i is an eigenvector of *L*. By multiplying both sides of $A^T A v_i = \mu_i v_i$ by A, we obtain:

$A A^T A v_i = \mu_i A v_i$

where $A v_i$ are the eigenvalues of the covariance matrix $A^T A$ (where $A^T A = A A^T$ from theorem of APPENDIX II).

From Theorem 2, it is easy to see that $A v_i$ are eigenvalues of C_E by (23). So that we can use the *M* eigenvectors of L in (24) to form the *M* eigenvectors u of C_E as follows:

$$u = \sum_{k=1}^{M} v_k \Phi_k \qquad (25)$$

Thus, u forms the eigenfaces basis which is a linearly independent basis set for the face space. It is evident that only $M - k$ eigenfaces are needed to produce the complete basis for the face space to achieve a descent reconstruction of the original face image using on few eigenfaces (M'); where *k* is the number of unique individuals in the database of known face images, and M' corresponds to the vectors with the highest eigenvalues within the face space. These eigenfaces also represent the *PCs* of the face image samples.

Corollary 2:

Consider a set of vectors representing the weight (w_p) and height (h_p) components of an individual which can be expressed as the dot product of two vectors as follows [11]:

$$\left.\begin{aligned} w_p &= individual \cdot \overline{weight} \\ h_p &= individual \cdot \overline{height} \end{aligned}\right\} \qquad (26)$$

Projecting a given individual onto these vectors should yield the individual's weight and height components in other to obtain the closest match between the individual and the set of individuals available in the database of the face images.

To accomplish the face recognition task, we begin by subtracting and projecting the mean of all the face images in the database onto the face space. According to *Corollary 2*, this corresponds to the dot product of each face image with one of the eigenfaces; which can be combined as matrices to obtain the weight matrix W as follows [11]:

$$W = \begin{pmatrix} w_{1,1} & \cdots & w_{1,\eta} \\ \vdots & \ddots & \vdots \\ w_{M',1} & \cdots & w_{M',\eta} \end{pmatrix} \qquad (27)$$

where $w_k = \mu_k(\Gamma_{new} - \varphi_\eta)$ and $k = 1, 2, \ldots, M'$.

In this way, the face image of the same individual can be mapped fairly close to one another in the face space; so that the face recognition problem further reduces to finding the minimum Euclidean distance ε_k between a face image point Ω_{new} and the database point Ω_k which can be expressed as follows (by noting from (27) that $\Omega = [w_1, w_2, \ldots, w_{M'}]^T$):

$$\varepsilon_k = \sqrt{\left\|\Omega_{new} - \Omega_k\right\|^2} \qquad (28)$$

The combination of the distribution in (28) and the PCA allows for dimensional reduction; where only the first several eigenfaces represent the majority information in the database and the computational complexities becomes significantly reduced.

As pointed out in [11], further verification on the decision to accept or reject the recognized face image can be achieved by considering the face space as an *N*-dimensional sphere encompassing all the weight vectors in the entire database; so that an approximate radius of the space should be half the distance between the furthest points in the sphere according to the following expression derived from (28):

$$\theta_{threshold} = (1/2)\max(\varepsilon_k) \quad (29)$$

To justify whether a new face lies within this radius, it is necessary to compute the reconstruction error ε between the face and its reconstruction using M eigenfaces as follows:

$$\varepsilon_{recon} = \sqrt{\left\|\Phi_{face} - \Phi_{recon}\right\|^2} \quad (30)$$

where $\Phi_{recon} = \sum_{\eta=1}^{M} w_\eta \mu_\eta$ is the reconstructed face image. If the face image projects fairly well onto the face space and follows the face image distribution, then the error should be small, i.e. $\varepsilon_{recon} < \theta_{threshold}$ and can be compared with the $\theta_{threshold}$ of the face images in the database. However, a non face image will always lie outside the radius of the face space (i.e., $\varepsilon_{recon} > \theta_{threshold}$) and the algorithm should terminate as there is no point for comparing $\theta_{threshold}$ with the face images in the database.

Finally, the implementation of our eigenfaces algorithm based on an averaging technique where all weight vectors of a like individual within the database are averaged together. This creates a "face class" where an even smaller weight matrix represents the general faces of the entire system. When a new face image is introduced for recognition, the algorithm creates its weight vector by projecting the introduced face image (*INFI*) onto the face space. The *INFI* is then matched to the face class that minimizes the Euclidean distance based on the number of signature (num_{sign}) required to recognize the *INFI*.

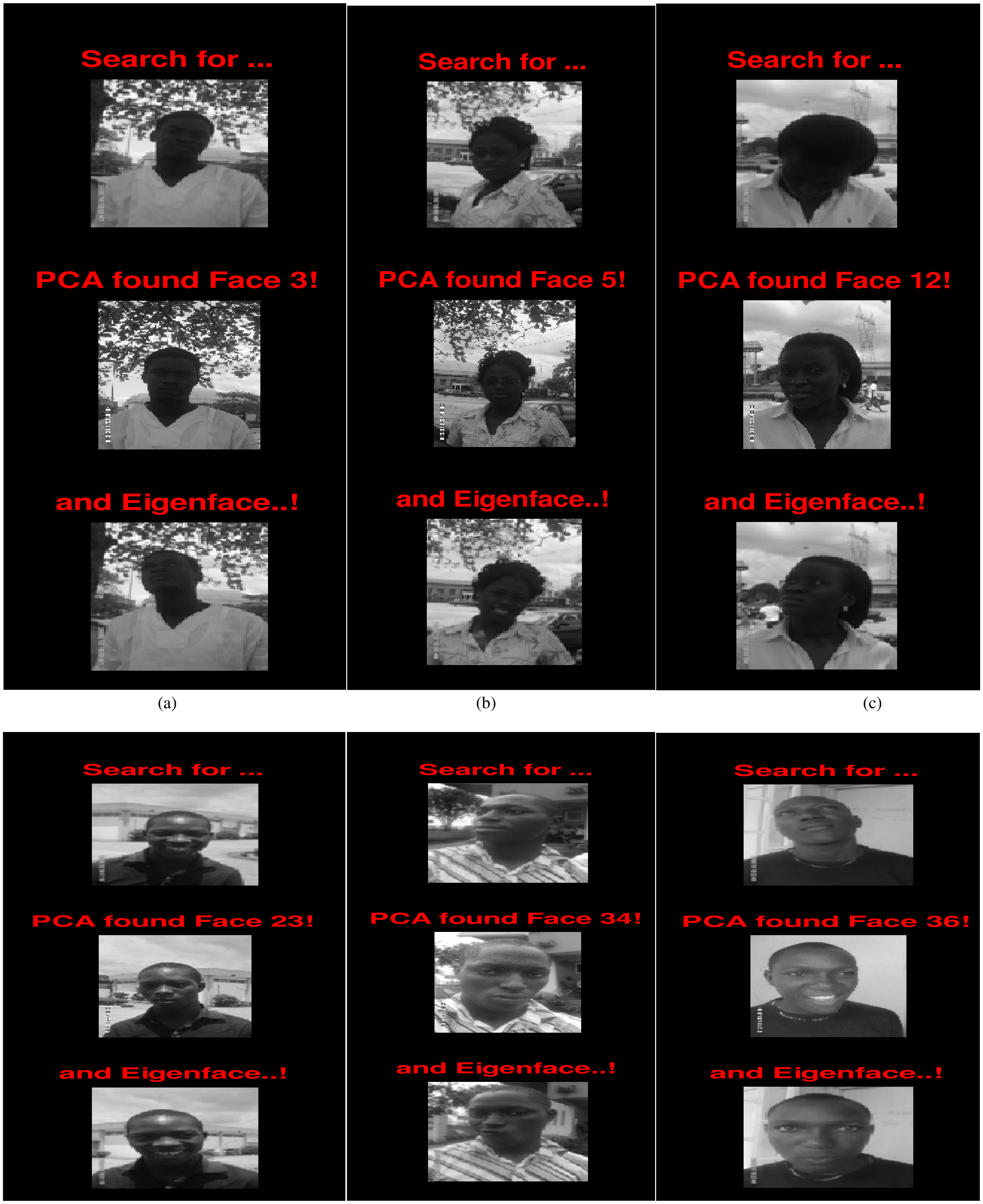
For the present study, our database has a total of three hundred and sixty face images, composed of thirty-six individuals with ten images each. The averaging technique thus yields a weight matrix with thirty-six vectors (thirty-six distinct face classes). Here we incorporate a face image counter (fim_{count}) which increments if the image matches correctly its own face class or does not change if the minimum distance matches to a face class of another person.

IV. Implementation of the PCA and Eigenfaces Algorithms for INFI Recognition

The PCA and the face recognition algorithms in the present study is implemented in such a way that the user of the algorithm is prompted for the *INFI* identify as a numerical number between 1 and 480 which corresponds to the number of face images in the database. If any numerical value between 1 and 360 is supplied, the PCA and the face recognition algorithms are activated.

Initially, the principal components (*PCs*) of all the known 360 face images in the database were computed and stored in a database as *image2_pca* (see TABLE I). The *PC* of the *INFI* is then computed and compared with the 360 known face image *PCs* in the database. If the *PC* of the *INFI* (shown in the upper segment of Fig. 5 (a), (b), (c), (d), (e) and (f)) corresponds to any *PC* in the *PC* database; the corresponding face image and name of the *INFI* are displayed as shown in the middle segment of Fig. 5 (a), (b), (c), (d), (e) and (f). However, if there is no corresponding *PC* for the *INFI* in the *PC* database, the comments shown in the middle part of Fig. 6 (a) and (b) are displayed (instead of Fig. 5) indicating non-existence of such face image in the database.

Similarly, the 360 face image sample were converted to *M*-dimensional matrices and stored in the database of known face images as *image1_facedb* (see TABLE II). The

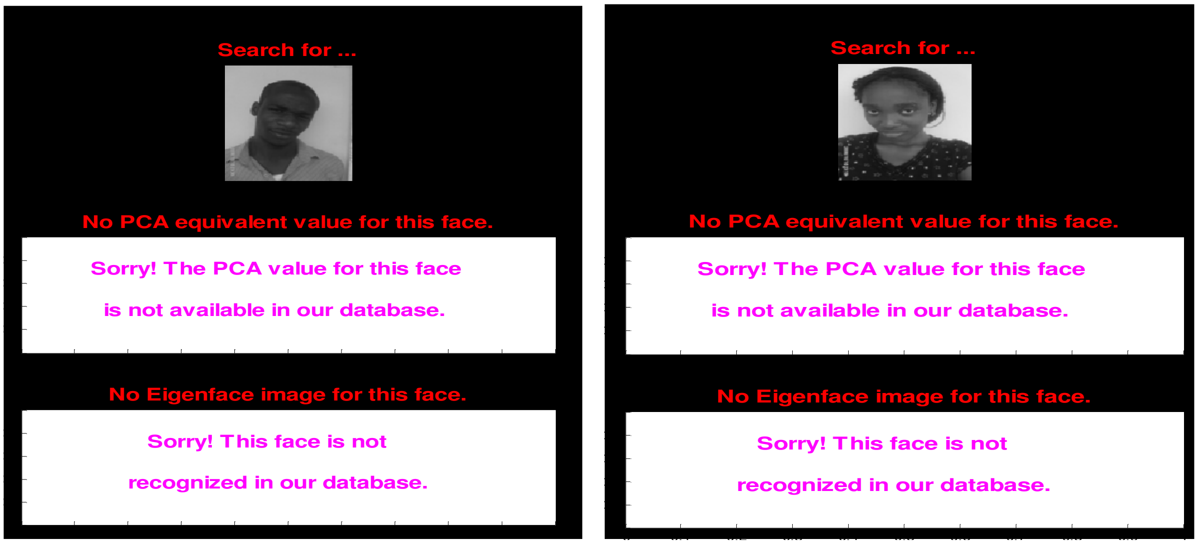


(a) (b) (c)

(d) (e) (f)

Fig. 5. The results obtained when the incoming face image (INFI) with known identity is recognized by both the PCA and the eigenfaces algorithms with their respective names.

eigenfaces algorithm is then evaluated for the *INFI* with $num_{sign} = 30$, projected and compared to the face space. If $\varepsilon_{recon} < \theta_{threshold}$, the *INFI* is compare to the available face images in the database in terms of the projected matrices on



(a) (b)

Fig. 6. The result obtained when the incoming face image (INFI) with known identity is not part of the known face images in the database.

Fig. 7. The result for invalid face image identity.

the face space; and the first $fim_{count} = 10$ highest eigenvalues that matches the *INFI* is selected to represent the *INFI* as shown in the lower segment of Fig. 5 (a), (b), (c), (d), (e) and (f). On the other hand if $\varepsilon_{recon} > \theta_{threshold}$, it implies that the *INFI* is not available and there should be no need for comparison with face image samples in the database; and consequently the comments in the lower part of Fig. 6 (a) and (b) are displayed.

The comments shown in the middle and lower segments were obtained for *INFI* with numerical values ranging between 361 and 480 which correspond to the 120 test face images (i.e., the last 25 % of the 480 face image samples) However, when numerical values outside the ranges 1 to 480 are used, the proposed algorithm responds with the comments shown in Fig. 7 and terminates automatically.

Because the PCA and the image processing algorithms are computationally intensive, the pre-computed values of the *PCs* and pre-processed face images *(PPFIs)* are loaded only once; so that only the values of *PC* and pre-processing of the *INFI* is computed at all future instances. In this way, the comparison, identification and recognition process is enhanced in terms of speed; which suggest a possibility for real-time online implementation of our proposed technique. *Theore*

V. CONCLUSION AND RECOMMENDATION

The variational properties of face images have been investigated based on numerical analysis. On the basis of the analysis, two algorithms: the principal component analysis and the eigenfaces algorithms have been proposed for efficient and enhanced optimal face recognition. These algorithms have been tested and validated with 360 known image face samples stored in a database and 120 unknown face image samples that were *Theore* not part of the database of known face images.

The performance of the proposed algorithms on 480 face image samples taken under varying angular positions and mostly under low light intensity conditions demonstrates the efficiency and adaptability of the proposed algorithms in critical and unprecedented situations. The implementation technique demonstrated in this work based on face image identity (i.e. the prompted numerical values) could be likened to bank account, identity card or passport numbers as a means of personal identification to checkmate crimes and fraudulent activities. However, a valid identity number only does not guarantee recognition if the face is not recognized.

The speed and accuracy of the proposed algorithms, based on the implementation strategy, show that it could be adapted for on-line face analysis and recognition in real-time. Further work could be on the parallel implementation of the proposed algorithms on real-time embedded reconfigurable computing machines such as field programmable gate arrays (FPGAs) or complex programmable logic devices (CPLDs) with real-time camera interfaced for on-line face analysis and recognition systems design.

APPENDIX I

Theorem: The inverse of an orthogonal matrix is its transpose

Proof:

Let $\mathbf{A} = [\mathbf{a}_1 \ \ \mathbf{a}_2 \ldots \mathbf{a}_n]$ be an $m \times n$ orthogonal matrix, where $\mathbf{a}_i$ is the i^{th} column vector. The ij^{th} element of $\mathbf{A}^T\mathbf{A}$ is expressed as

$$(\mathbf{A}^T\mathbf{A})_{ij} = \mathbf{a}_i{}^T\mathbf{a}_j = \begin{cases} 1 & if \quad i = j \\ 0 & otherwise \end{cases}$$

Therefore, because $\mathbf{A}^T\mathbf{A} = \mathbf{I}$, where $\mathbf{I}$ is an identity matrix; it follows that $\mathbf{A}^{-1} = \mathbf{A}^T$.

APPENDIX II

Theorem: For any matrix $\mathbf{A}$, $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}\mathbf{A}^T$ are symmetric.

Proof:

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

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

The equality of quantities in the above two expressions with their transpose completes this proof.

APPENDIX III

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

Proof:

If $\mathbf{A}$ is orthogonally diagonalizable, then $\mathbf{A}$ is symmetric. By hypothesis, orthogonally diagonalizable implies that there exist some matrix $\mathbf{E}$ such that $\mathbf{A} = \mathbf{EDE}^T$, where $\mathbf{D}$ is a diagonal matrix and $\mathbf{E}$ is some special matrix which diagonalizes $\mathbf{A}$. So that $\mathbf{A}^T$ can be computes as follows:

$$\mathbf{A}^T = (\mathbf{EDE}^T)^T = \mathbf{E}^{TT}\mathbf{D}^T\mathbf{E}^T = \mathbf{EDE}^T = \mathbf{A}$$

It is evident that if $\mathbf{A}$ is orthogonally diagonalizable, it must also be symmetric.

APPENDIX IV

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

Proof:

Let $\mathbf{A}$ be a square $n \times n$ matrix with associated eigenvectors $\{\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_n}\}$. Let $\mathbf{E} = \{\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_n}\}$ where the i^{th} column of $\mathbf{E}$ is the eigenvector $\mathbf{e_i}$. This theorem asserts that there exists a diagonal matrix D where $\mathbf{A} = \mathbf{EDE^T}$.

This theorem is an extension of APPENDIX III. It provides a prescription of how to find the matrix $\mathbf{E}$, the "diagonalizer" for a symmetric matrix. This theorem emphasizes that the diagonalizer is in fact a matrix of the original matrix's eigenvectors. The proof is in two parts.

For the first part, let **A** be some matrix, not necessary symmetric, having independent eigenvectors (i.e. no degeneracy). Furthermore, let $\mathbf{E} = \{\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_n}\}$ be the matrix of eigenvectors placed in the columns. Let **D** be a diagonal matrix where the i^{th} eigenvalue is placed in the ii^{th} position. The proof is to show that $\mathbf{AE} = \mathbf{ED}$.

$$\mathbf{AE} = [\mathbf{Ae_1}, \mathbf{Ae_2}, \ldots, \mathbf{Ae_n}]$$
$$\mathbf{ED} = [\lambda_1\mathbf{e_1}, \lambda_2\mathbf{e_2}, \ldots, \lambda_n\mathbf{e_n}]$$

Evidently, if $\mathbf{AE} = \mathbf{ED}$, then $\mathbf{Ae_i} = \lambda_i\mathbf{e_i}, \ \forall i$. This equation is the definition of the eigenvalue equation. Therefore, it must be that $\mathbf{AE} = \mathbf{ED}$. Using the result of APPENDIX III, it is easy to see that $\mathbf{A} = (\mathbf{EDE})^{-1}$.

The second part, let λ_1 and λ_2 be distinct eigenvalues for eigenvectors $\mathbf{e_1}$ and $\mathbf{e_2}$; so that

$$\lambda_1\mathbf{e_1} \cdot \mathbf{e_2} = (\lambda_1\mathbf{e_1})^T\mathbf{e_2} = (\mathbf{Ae_1})^T\mathbf{e_2} = \mathbf{e_1}^{\mathbf{T}}\mathbf{A}^{\mathbf{T}}\mathbf{e_2}$$
$$= \mathbf{e_1}^{\mathbf{T}}\mathbf{Ae_2} = \mathbf{e_1}^{\mathbf{T}}(\lambda_2\mathbf{e_2})$$
$$\therefore \lambda_1\mathbf{e_1} \cdot \mathbf{e_2} = \lambda_2\mathbf{e_1} \cdot \mathbf{e_2}$$

Equating the last equation gives $(\lambda_1 - \lambda_2)\mathbf{e_1} \cdot \mathbf{e_2} = \mathbf{0}$. Since we have conjectured that the eigenvalues are in fact unique, it must be the case that $\mathbf{e_1} \cdot \mathbf{e_2} = \mathbf{0}$. Therefore, the eigenvectors of a symmetric matrix are orthogonal.

Following our original postulate that **A** is a symmetric matrix and by the second part of the proof, it can be seen that the eigenvectors of **A** are al orthonormal (we choose the eigenvectors to be normalized). This means that **E** is an orthogonal matrix, and by APPENDIX I, $\mathbf{E}^T = \mathbf{E}^{-1}$; so that we can express the final result as $\mathbf{A} = (\mathbf{EDE})^{-T}$. Thus, a symmetric matrix is diagonalized by a matrix of its eigenvectors.

In the first part, we see that any matrix can be orthogonally diagonalized if and only if the matrix's eigenvectors are linearly independent. In the second part, it has been seen that a symmetric matrix has the special property that all of its eigenvectors are not just linearly dependent but also orthogonal; and thus, completing the proof.

ACKNOWLEDGMENT

The authors wish to thank the students of The Federal University of Petroleum Resources, Effurun, Delta State, Nigeria, whose photographs form the database, for accepting to be part of this work.

REFERENCES

[1] S. Haykin and B. Veen. *Signals and Systems*. U.S.A.: John Wiley & Sons, Inc, 1999.

[2] K. A. Jain. *Fundamentals of Digital Image Processing*. U. S. A.: Prentice–Hall International Inc, 1998.

[3] B. Porat. *A Course in Digital Signal Processing*. U. S. A.: John Wiley & Sons, Inc., 1997.

[4] C. B. Rorabaugh. *DSP Primer: In-dept Coverage for Singular Result.*, New York. U.S.A.: McGraw-Hill, New York, 1999.

[5] K. Rumiana, B. Ani, B. Ditchko and G. Veselin. "Basic Components in Optical Character Recognition (OCR) Systems. Experimental Analysis on old Bulgarian Character Recognition," *Academic Open Internet Journal*, vol. 5, 2000. [Online]. Available: www.acadjournal.com.

[6] Source Signal Imaging Inc. (2008). EMSE Suite Module Software Specifications: Image Processor & Image Processing Toolkit. [Online]. Available:http://www.sourcesignal.com/imageproc.html

[7] J. Wu, W. A. P. Smith and E. R. Hancock (2009), "Facial gender classification using shape from shading," *Image and Vision Computing*, doi: 10.1016/j.imavis.2009.09.003. (Article in Press).

[8] G. C. Littlewort, M. S. Bartlett and K. Lee, "Automatic coding of facial expressions displayed during posed and genuine pain," *Image and Vision Computing*, vol. 27, pp. 1797 – 1803, 2009.

[9] S. Lucey, Y. Wang, M. Cox, M. Sridharan and J. F. Cohn, "Efficient constrained local model fitting for non-rigid face alignment," *Image and Vision Computing*, vol. 27, pp. 1804 – 1813, 2009.

[10] L. I. Smith. (Feb. 26, 2002). A Tutorial on Principal Component Analysis. *Technical Report*, [Online]. Available: http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

[11] M. Robinson, M. Escarra, J. Krueger and D. Kochelek (2009). Face Recognition using Eigenfaces. *Course Number: ELEC 301*, CONNEXIONS, Rice University, Houston, Texas. [Online]. Available: http://cnx.org/content/col10254/1.2/

[12] D. J. Duh, J. H. Jeng and S. Y. Chen, "DCT based simple classification scheme for fractal image compression," *Image and Vision Computing*, vol. 23, pp. 1115 – 1121, 2005.

[13] N. Chen, K. Chung and J. Hung, "Novel fractal encoding algorithm using normalized one-norm and kick-out condition," *Image and Vision Computing,* (2009), doi:10.1016/j.imavis.2009.08.007. (Article in Press).

[14] P. K. Yalamanchili and B. D. Paladugu. (2007). *Comparative Study of Face Recognition Algorithms*. Final Project, ECE847: Digital Image Processing, Department of Electrical Engineering, Clemson University, South Carolina, U. S. A. [Online]. Available: http://www.ces.clemson.edu/~stb/ece847/fall2007/projects/face%20recognition.pdf

[15] P. N. Belhunmeur, J. P. Hespanha and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE Trans. Pattern Annalysis and Machine Intelligence,* vol. 19, no. 7, pp. 711-720, 1997.

[16] J. Choi, S. Lee, C. Lee and J. Yi, A real-time face recognition system using multiple mean faces and dual mode fisherfaces, *IEEE International Symposium on Industrial Electronics (ISIE 2001)*, pp. 1686 – 1689, 2001.

[17] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Trans. Image Process*, vol. 1, no. 1, pp. 18 – 30, 1992.

[18] T. K. Truong, J. H. Jeng, I. S. Reed, P. C. Lee and A. Q. Li, "A fast encoding algorithm for fractal image compression using the DCT inner product," *IEEE Trans. Image Process*, vol. 4, no. 4, pp. 529 – 535, 2000.

[19] R. Distasi, M. Nappi and D. Riccio, "A range/domain approximation error-based approach for fractal image compression," *IEEE Trans. Image Process*, vol. 15, no.1, pp. 89 – 97, 2006.

[20] H. Bae and S. Kim, "Real-time face detection and recognition using hybrid-information extracted from face space and facial features," Image and Vision Computing, vol. 23, pp. 1181 – 1191, 2005.

[21] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neurosicence*, vol. 3, no. 1, pp. 71 – 86, 1991.

[22] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 586 – 591, 1991.
[23] Samsun Opto-Electronics Co. Ltd. www.samsungcamera.com
[24] The MathWorks Inc., MATLAB 2009a, http://www.mathworks.com, 3 Maple Drive, California, U. S. A.
[25] Wikipedia, the free encyclopedia (Oct. 13, 2009). Principal component analysis, [Online]. Available: http://en.wikipedia.org/wiki/Principal_component_analysis
[26] D. Lay, *Linear Algebra and It's Applications*. New York: Adison-Wesley, 2000.
[27] T. Will, Introduction to the Singular Value Decomposition (1999), www.davidson.edu/academic/math/will/svd/index.html.

Vincent Andrew Akpan (M'08) received his B.Sc. in Physics from The Delta State University (DELSU), Abraka, Nigeria in 1997 and his M.Tech. in Electronic Measurement & Instrumentation from The Federal University of Technology (FUTA), Akure, Nigeria in 2003.

Between 1998 and 2004, he was a Graduate Assistant with the Department of Physics, DELSU. Since 2004 he has been an Assistant Lecturer with the Department of Physics Electronics, FUTA. He is currently on a PhD leave at the Aristotle University of Thessaloniki, Greece. He is the co-authored of a book; and has authored or co-authored more than 20 articles in refereed conference proceedings and journals as well as 7 advanced technical reports (on-line). He is interested in learning about the application of artificial intelligence to totally-integrated industrial network control systems and signal processing & image recognition systems design.

Mr. Akpan is a member of The IEEE, USA; The IET, UK; and The IOP, UK. He is one of the two Nigerian recipients of the 2005/2006 Bilateral Educational Agreement scholarship between The Federal Government of Nigeria and The Greek State Scholarships Foundation (IKY) for a Ph.D programme tenable in Greece.

Reginald A. O. Osakwe received his B.Sc. degree in Physics from the University of Lagos, Nigeria in 1979; his M.Sc. in solid state Physics from the Purdue University, West Lafayette, Indiana, U.S.A. in 1984; and his Ph.D in electronics from the University of Ibadan, Nigeria in 1995.

Between 1991 and 2007, he was a lecturer with the Department of Physics, Delta State University, Abraka, Nigeria. Since 2007 he has been with the Department of Physics, Federal University of Petroleum Resources, Effurun, Delta State, Nigeria where he is currently a Senior Lecturer. During the periods 1998 – 2009 he has been a visiting lecturer to the Shell Intensive Training Programme (SITP), Warri, Delta State under the Shell Petroleum Development Company (SPDC) of Nigeria. His research interests are on microprocessor architecture, embedded computer systems, chaotic dynamics, and image and signal processing systems.

Dr. Osakwe is a member of the Nigeria Institute of Physics (NIP). He was a recipient of The Federal Government of Nigeria Scholarship to study in the U.S.A. leading to the award of an M.Sc.