BRIEF INTRODUCTION TO :

- **Pattern Recognition**
- **Face Recognition**
- **Face Recognition Using EigenFaces**

ILKER ATALAY - COMPUTER ENGINEER M.Sc

M.Sc Thesis: Face Recognition Using Eigenfaces
Istanbul Technical University – January 1996
E-mail: ilker@ilkeratalay.com

Pattern Recognition

Pattern recognition can be defined as the categorization of input data into identifiable classes via the extraction of significant features or attributes of the data from a background of irrelevant detail.

A **pattern class** is a category determined by some given common attributes or features. The features of a pattern class are the characterizing attributes common to all patterns belonging to that class. Such features are often referred to as intraset features. The features which represent the differences between pattern classes may be referred to as the interset features.

A **pattern** is the description of any member of a category representing a pattern class. For convenience, patterns are usually represented by a vector such as:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ \dots \\ x_n \end{bmatrix}$$

where each element x_j, represents a feature of that pattern. It is often useful to think of a pattern vector as a point in an n-dimensional Euclidean space.

Determination and then the discrimination of these pattern vectors form the two major problems in pattern recognition system design.

Generally, there exists little, if any, a priori knowledge about the patterns to be recognized. Under these circumstances pattern recognizing machines are best designed using a **training** or **learning** procedure. Arbitrary decision functions are initially assumed, and through a sequence of iterative training steps these decision functions are made to approach optimum or satisfactory forms.

Supervised pattern recognition is characterized by the fact that the correct classification of every training pattern is known. In the **unsupervised** case however, one is faced with the problem of actually learning the pattern classes present in the given data. This problem is also known as "**learning without a teacher**".

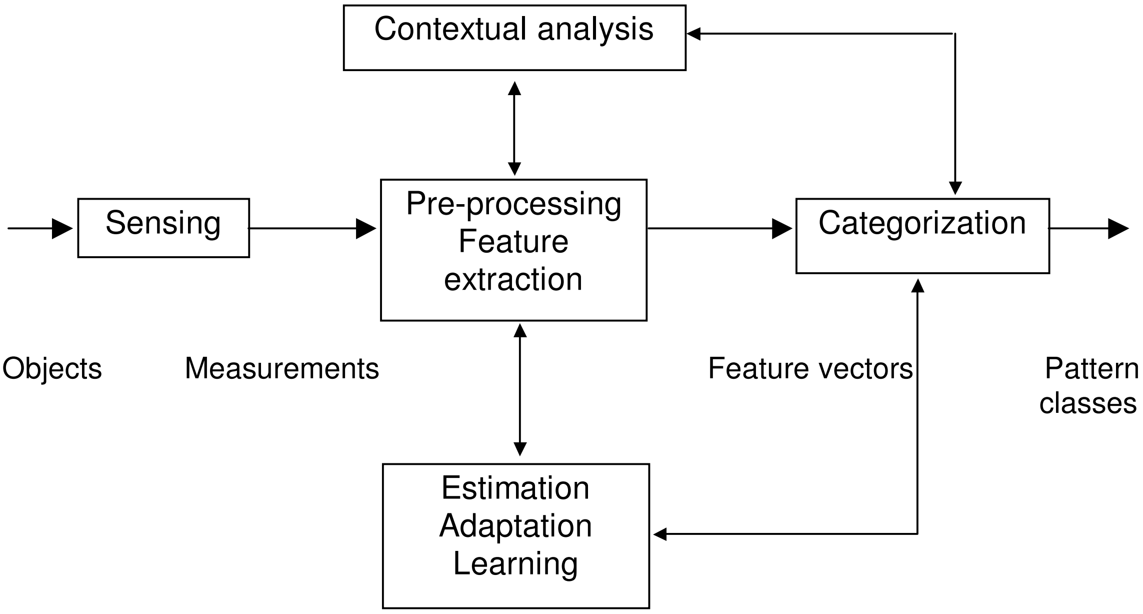


Figure 1. Functional block diagram of an adaptive pattern recognition system.

–

Face Recognition

Face recognition is a pattern recognition task performed specifically on faces. It can be described as classifying a face either "**known**" or "**unknown**", after comparing it with stored known individuals. It is desirable to have a system that has the ability of learning to recognize unknown faces.

Computational models of face recognition must address several difficult problems. This difficulty arises from the fact that faces must be represented in a way that best utilizes the available face information to distinguish a particular face from all other faces.

Faces pose a particularly difficult problem in this respect because all faces are similar to one another in that they contain the same set of features such as eyes, nose, mouth arranged in roughly the same manner.

In Figure 2, the outline of a typical face recognition system is given. This outline heavily carries the characteristics of a typical pattern recognition system that was presented in Figure 1.

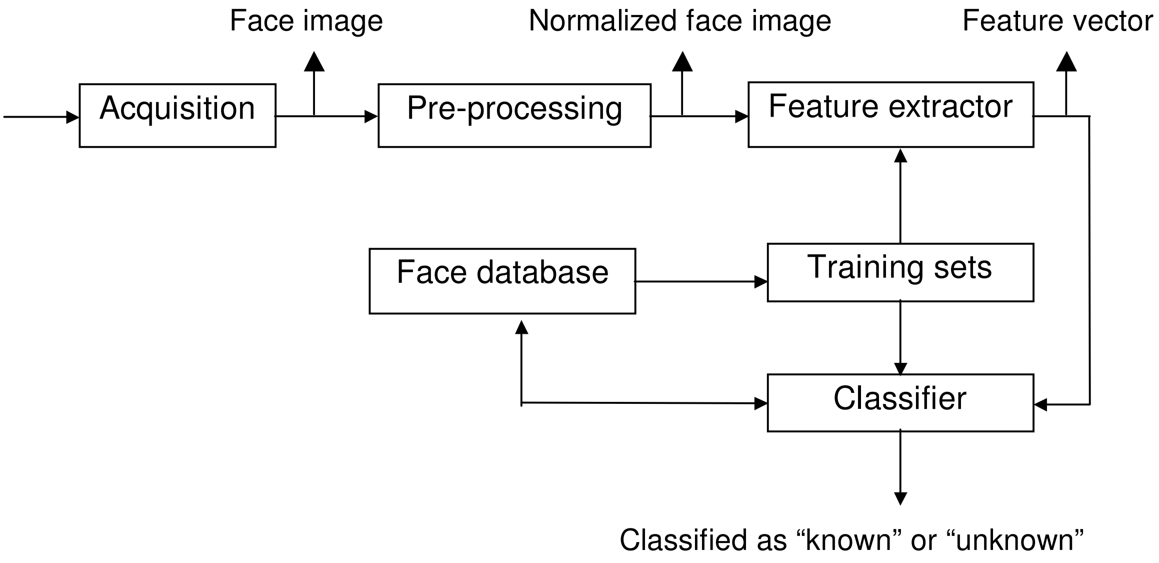


Figure 2. Outline of a typical face recognition system.

Acquisition module. This is the entry point of the face recognition process. It is the module where the face image under consideration is presented to the system. An acquisition module can request a face image from several different environments.

Pre-processing module. By means of early vision techniques, face images are normalized and if desired, they are enhanced to improve the recognition performance of the system. Some or all of the following pre-processing steps may be implemented in a face recognition system:

- Image size normalization
- Histogram equalization , illumination normalization
- Median filtering

- High-pass filtering
- Background removal
- Translational and rotational normalizations

Feature extraction module. After performing some pre-processing (if necessary), the normalized face image is presented to the feature extraction module inorder to find the key features that are going to be used for classification.

Classification module. In this module, with the help of a pattern classifier, extracted features of the face image is compared with the ones stored in a face library (or face database). After doing this comparison, face image is classified as either known or unknown.

Training set. Training sets are used during the "learning phase" of the face recognition process. The feature extraction, and the classification modules adjust their parameters inorder to achieve optimum recognition performance by making use of training sets.

Face library or face database. After being classified as "unknown", face images can be added to a library (or to a database) with their feature vectors for later comparisons. The classification module makes direct use of the face library.

Objectives of a Robust Face Recognition System

- **Scale invariance**
- **Shift invariance**
- **Illumination invariance**
- **Emotional expression and detail invariance**
- **Noise invariance**

A robust face recognition system should be capable of classifying a face image as "**known**" under even above conditions, if it has already been stored in the face database.

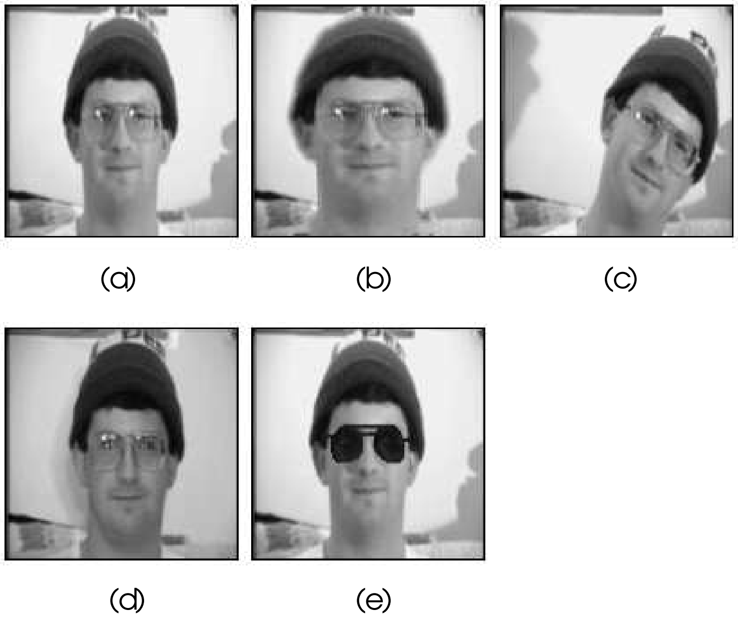


Figure 3. (a) Original face image. (b) Scale variance. (c) Orientation variance. (d) Illumination variance. (e) Presence of details.

Two Major Approaches to Face Recognition

- **Feature Based Face Recognition** which is based on the extraction of the properties of individual organs located on a face such as eyes, nose and mouth, as well as their relationships with each other. Effective features that can be used in feature based face recognition can be classified as follows:

    - **First-order features values.** Discrete features such as eyes, eyebrows, mouth, chin, and nose, which have been found to be important in face identification and are specified without reference to other facial features, are called first-order features.

- **Second-order features values.** Another configural set of features which characterize the spatial relationships between the positions of the first-order features and information about the shape of the face are called second-order features.

Measurement	Facial Location
Area, angle	left eyebrow right eyebrow left eye right eye mouth face
Distance	length of left eyebrow length of right eyebrow length of left eye length of right eye length of mouth length of face height of face

Table 1. First-order features

Measurement	Facial Location
Distance	left eyebrow <-> right eyebrow left eye <-> right eye left eyebrow <-> left eye right eyebrow <-> right eye left eyebrow <-> mouth right eyebrow <-> mouth left eye <-> mouth right eye <-> mouth eyebrow <-> side of face eye <-> side of face mouth <-> side of face mouth <-> lower part of face
Angle	left eyebrow - left eye - left eyebrow right eyebrow - right eye - right eyebrow left eye - left eyebrow - left eye right eye - right eyebrow - right eye left eyebrow - mouth - right eyebrow left eye - mouth - right eye left eyebrow - left eye - mouth right eyebrow - right eye - mouth

Table 2. Second-order features

For different facial contours, different models should be utilized to extract them from the original portrait. Because the shapes of eyes and mouth are similar to some geometric figures, they can be extracted in terms of the **deformable template model**.

The other facial features such as eyebrows, nose and face are so variable that they have to be extracted by the **active contour model**. These two models can be illustrated in the following:

- **Deformable template model.** The deformable templates are specified by a set of parameters which uses a priori knowledge about the expected shape of the features to guide the contour deformation process. The templates are flexible enough to change their size and other parameter values, so as to match themselves to the data. The final values of these parameters can be used to describe the features. This method works well regardless of variations in scale, tilt, and rotations.

- **Active contour model (Snake).** The active contour or snake is an energy minimizing spline guided by external constraint forces and influenced by image forces that pull it toward features such as lines and edges. Snakes lock onto nearby edges, localizing them accurately. Because the snake is an energy minimizing spline, energy functions whose local minima comprise the set of alternative solutions to higher level processes should be designed. Selection of an answer from this set is accomplished by the addition of energy terms that push the model toward the desired solution.

  Both of these models use complex geometry and require the minimization of some energy functions

- **Principal component analysis,** based on information theory concepts, seek a computational model that best describes a face, by extracting the most relevant information contained in that face without dealing with the individual properties of facial organs such as eyes or mouth.

Face Recognition Using Eigenfaces

Eigenfaces approach is a principal component analysis method, in which a small set of characteristic pictures are used to describe the variation between face images. Goal is to find out the eigenvectors (eigenfaces) of the covariance matrix of the distribution, spanned by a training set of face images. Later, every face image is represented by a linear combination of these eigenvectors. Evaluation of these eigenvectors are quite difficult for typical image sizes but, an approximation can be made. Recognition is performed by projecting a new image into the subspace spanned by the eigenfaces and then classifying the face by comparing its position in face space with the positions of known individuals.

Calculating Eigenfaces

Let a face image I(x,y) be a two-dimensional N x N array of 8-bit intensity values. An image may also be considered as a vector of dimension N^2, so that a typical image of size 256 x 256 becomes a vector of dimension 65,536, or equivalently a point in 65,536-dimensional space. An ensemble of images, then, maps to a collection of points in this huge space.

Images of faces, being similar in overall configuration, will not be randomly distributed in this huge image space and thus can be described by a relatively low dimensional subspace. The main idea of the principal component analysis (or Karhunen-Loeve expansion) is to find the vectors that best account for the distribution of face images within the entire image space.

These vectors define the subspace of face images, which we call "face space". Each vector is of length N^2, describes an N x N image, and is a linear combination of the original face images.

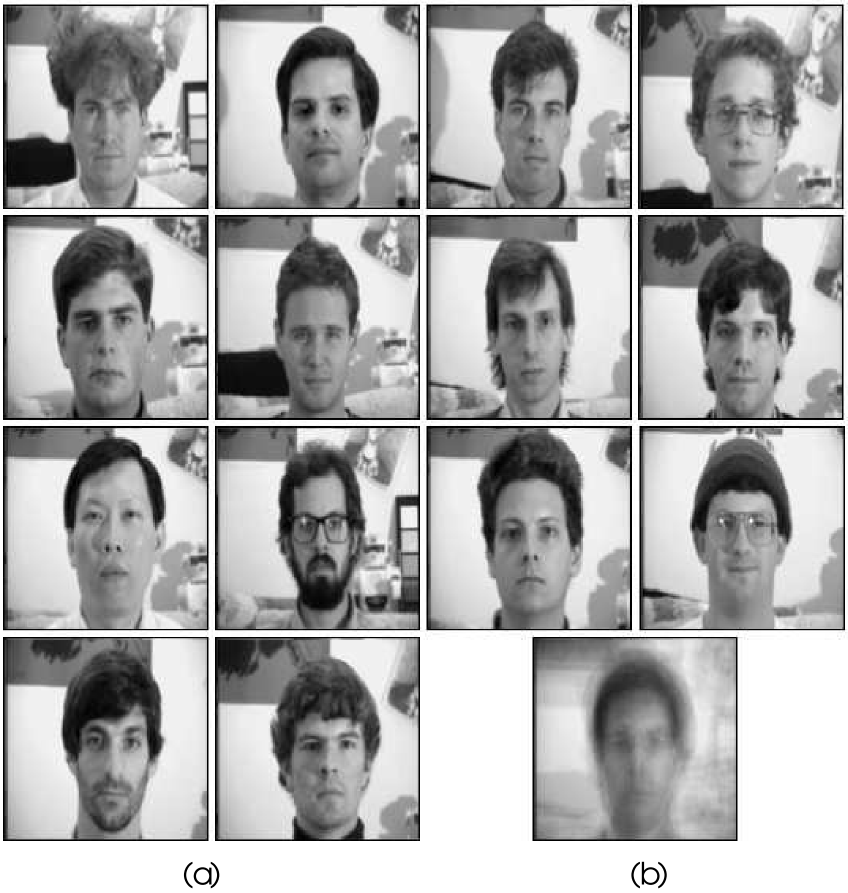


Figure 4. (a) Sample, training set face images. (b) Average face image of the training set.

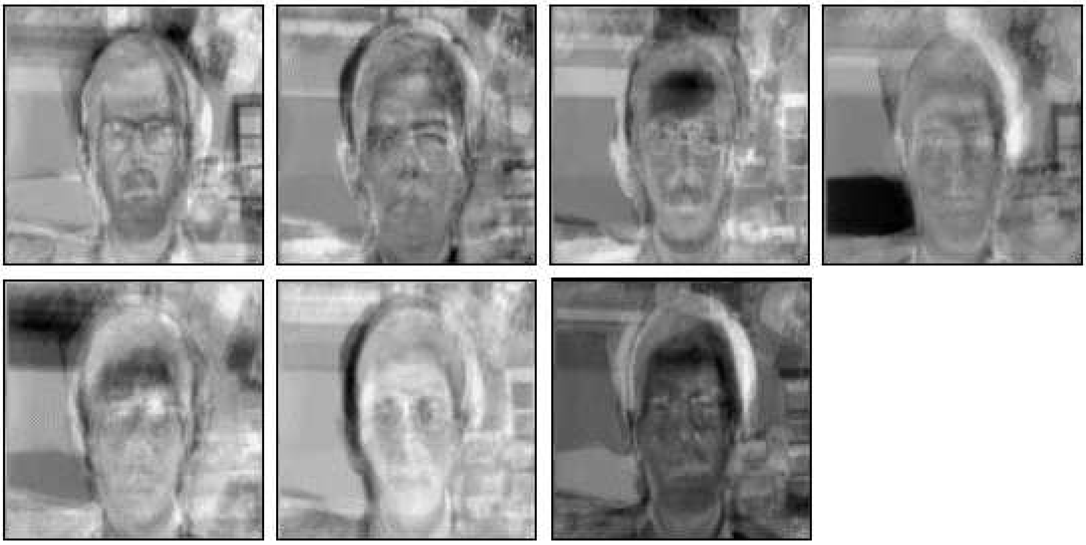


Figure 5. 7 eigenfaces with highest eigenvalues, that were calculated from the sample training set, given in Figure 4.

Let the training set of face images be $\Gamma_1, \Gamma_2, \ldots, \Gamma_M$ then the average of the set is defined by

$$\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n$$

Each face differs from the average by the vector

$$\Phi_i = \Gamma_i - \Psi$$

This set of very large vectors is then subject to principal component analysis, which seeks a set of M orthonormal vectors, u_n, which best describes the distribution of the data. The kth vector, u_k, is chosen such that

$$\lambda_k = \frac{1}{M}\sum_{n=1}^{M}(u_k^T \Phi_n)^2$$

is a maximum, subject to

$$u_l^T u_k = \delta_{lk} = \begin{Bmatrix} 1, & \text{if l=k} \\ 0, & \text{otherwise} \end{Bmatrix}$$

The vectors u_k and scalars λ_k are the eigenvectors and eigenvalues, respectively of the covariance matrix

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n \Phi_n^T = AA^T$$

where the matrix $A = [\Phi_1 \Phi_2 \ldots \Phi_M]$ The covariance matrix C, however is $N^2 x N^2$ real symmetric matrix, and determining the N^2 eigenvectors and eigenvalues is an intractable task for typical image sizes. We need a computationally feasible method to find these eigenvectors.

If the number of data points in the image space is less than the dimension of the space (M < N^2), there will be only M-1, rather than N^2, meaningful eigenvectors. The remaining eigenvectors will have associated eigenvalues of zero. We can solve for the N^2 dimensional eigenvectors in this case by first solving the eigenvectors of an M x M matrix such as solving 16 x 16 matrix rather than a 16,384 x 16,384 matrix and then, taking appropriate linear combinations of the face images Φ_i.

Consider the eigenvectors v_i of $A^T A$ such that

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

Premultiplying both sides by A, we have

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

from which we see that Av_i are the eigenvectors of $C = AA^T$.

Following these analysis, we construct the M x M matrix $L = A^T A$, where $L_{mn} = \Phi_m^T \Phi_n$, and find the M eigenvectors, v_l, of L. These vectors determine linear combinations of the M training set face images to form the eigenfaces u_l.

$$u_l = \sum_{k=1}^{M} v_{lk} \Phi_k , \qquad l = 1,\ldots,M$$

Using Eigenfaces to Classify a Face Image

A new face image (Γ) is transformed into its eigenface components (projected onto "face space") by a simple operation,

$$w_k = u_k^T (\Gamma - \Psi)$$

for k = 1,...,M. This describes a set of point by point image multiplications and summations, operations performed at approximately frame rate on current image processing hardware, with a computational complexity of O(N^4).

The weights form a feature vector,

$$\Omega^T = [w_1 w_2 \ldots w_{M'}]$$

that describes the contribution of each eigenface in representing the input face image, treating the eigenfaces as a basis set for face images. The feature vector is then used in a standard pattern recognition algorithm to find which of a number of predefined face classes, if any, best describes the face.

$$\frac{\|\Omega - \Omega_k\|}{\|\Omega_k\|} \le \varepsilon_k$$

Rebuilding a Face Image with Eigenfaces

A face image can be approximately reconstructed (rebuilt) by using its feature vector and the eigenfaces as

$$\Gamma' = \Psi + \Phi_f$$

where

$$\Phi_f = \sum_{i=1}^{M'} w_i u_i$$

is the projected image.

We see that the face image under consideration is rebuilt just by adding each eigenface with a contribution of w_i to the average of the training set images. The degree of the fit or the "rebuild error ratio" can be expressed by means of the Euclidean distance between the original and the reconstructed face image as

$$\text{Rebuild error ratio} = \frac{\|\Gamma' - \Gamma\|}{\|\Gamma\|}$$

It has been observed that, rebuild error ratio increases as the training set members differ heavily from each other. This is due to the addition of the average face image. When the members differ from each other (especially in image background) the average face image becomes more messy and this increases the rebuild error ratio.

There are four possibilities for an input image and its pattern vector:

1. Near face space and near a face class,
2. Near face space but not near a known face class,
3. Distant from face space and near a face class,
4. Distant from face space and not near a known face class.

In the first case, an individual is recognized and identified. In the second case, an unknown individual is presented. The last two cases indicate that the image is not a face image. Case three typically shows up as a false classification. It is possible to avoid this false classification in this system as

$$\frac{\left\| \Phi - \Phi_f \right\|}{\left\| \Phi_f \right\|} \leq \phi_k$$

where ϕ_k is a user defined threshold for the faceness of the input face images belonging to kth face class.

Eigenface Recognition Summary

- Form a face library that consists of the face images of known individuals.
- Choose a training set that includes a number of images (M) for each person with some variation in expression and in the lighting.
- Calculate the M x M matrix L, find its eigenvectors and eigenvalues, and choose the M' eigenvectors with the highest associated eigenvalues.
- Combine the normalized training set of images to produce M' eigenfaces. Store these eigenfaces for later use.
- For each member in the face library, compute and store a feature vector.
- Choose a threshold ε that defines the maximum allowable distance from any face class. Optionally choose a threshold ϕ that defines the maximum allowable distance from face space.
- For each new face image to be identified, calculate its feature vector and compare it with the stored feature vectors of the face library members. If the comparison satisfies the threshold for at least one member, then classify this face image as "known", otherwise a miss has occured and classify it as "unknown" and add this member to the face library with its feature vector.

Eigenfaces vs Feature Based Face Recognition

Speed and simplicity. Eigenfaces approach is superior in its near real time speed and reasonably simple implementation, where as feature based face recognition involves complex computations such as deformable templates and active contour models.

Learning capability. Feature based face recognition systems are generally trained to optimize their parameters in a supervised manner. In the eigenfaces approach, training is done in an unsupervised manner. User selects a training set that represents the rest of the face images. Eigenfaces are obtained from the training set members and feature vectors are formed.

Face background. Eigenfaces approach is very sensitive to face background, in case feature vectors are obtained by image additions and multiplications. Feature based face recognition algorithms are less sensitive to face background due to the localization of facial contours by deformable templates.

Scale and orientation. In the eigenfaces approach, recognition performance decreases quickly as head size or orientation is misjudged. Inorder to overcome this problem, multiscale eigenfaces can be used or the head can be removed from the rest of the image, and then scaled or rotated to meet the specifications of the eigenfaces. Again, feature based face recognition algorithms can score better in this comparison because they find facial features by using deformable templates and active contour models that are less sensitive to scale and orientation.

Presence of small details. Feature based face recognition algorithms can suffer when some details are present on the face image such as dark glasses or beards. For a feature based face recognition system, it is quite impossible to extract the features that are related to the eyes when dark glasses is present on the face. Eigenfaces approach excels in this aspect of face recognition. Small changes in face images such as glasses, beards or mustaches does not cause a decrease in the face recognition performance because the information that is present in the rest of the face image makes it enough to be classified correctly.

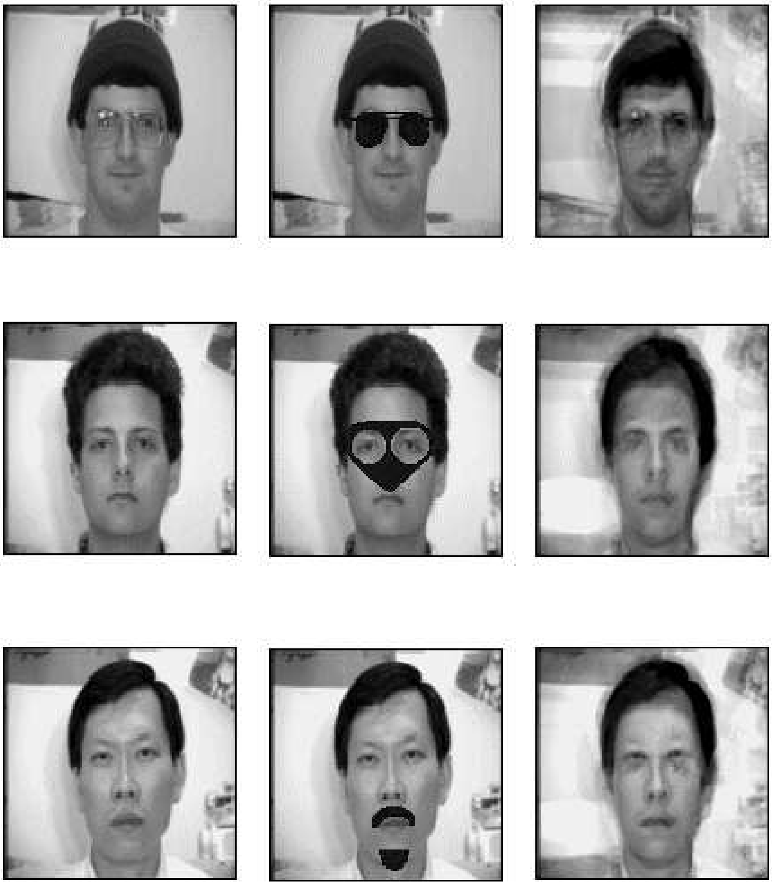
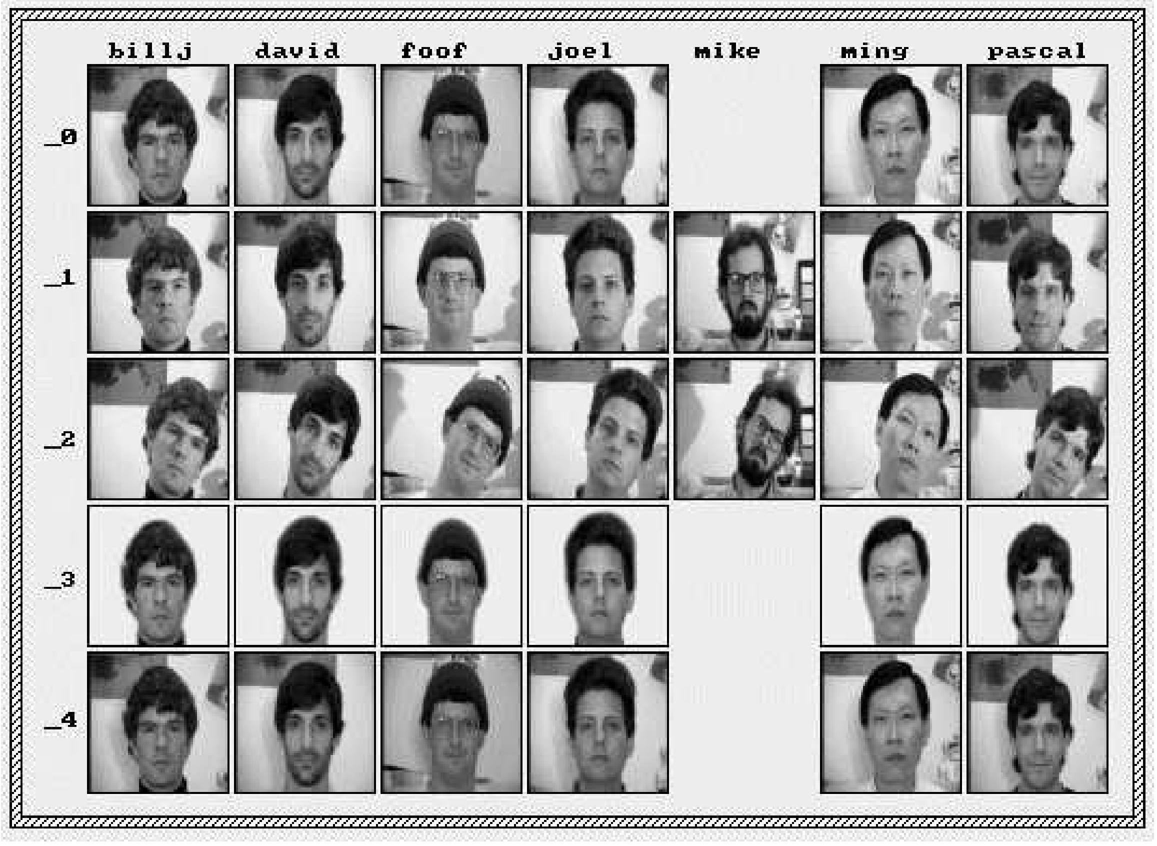
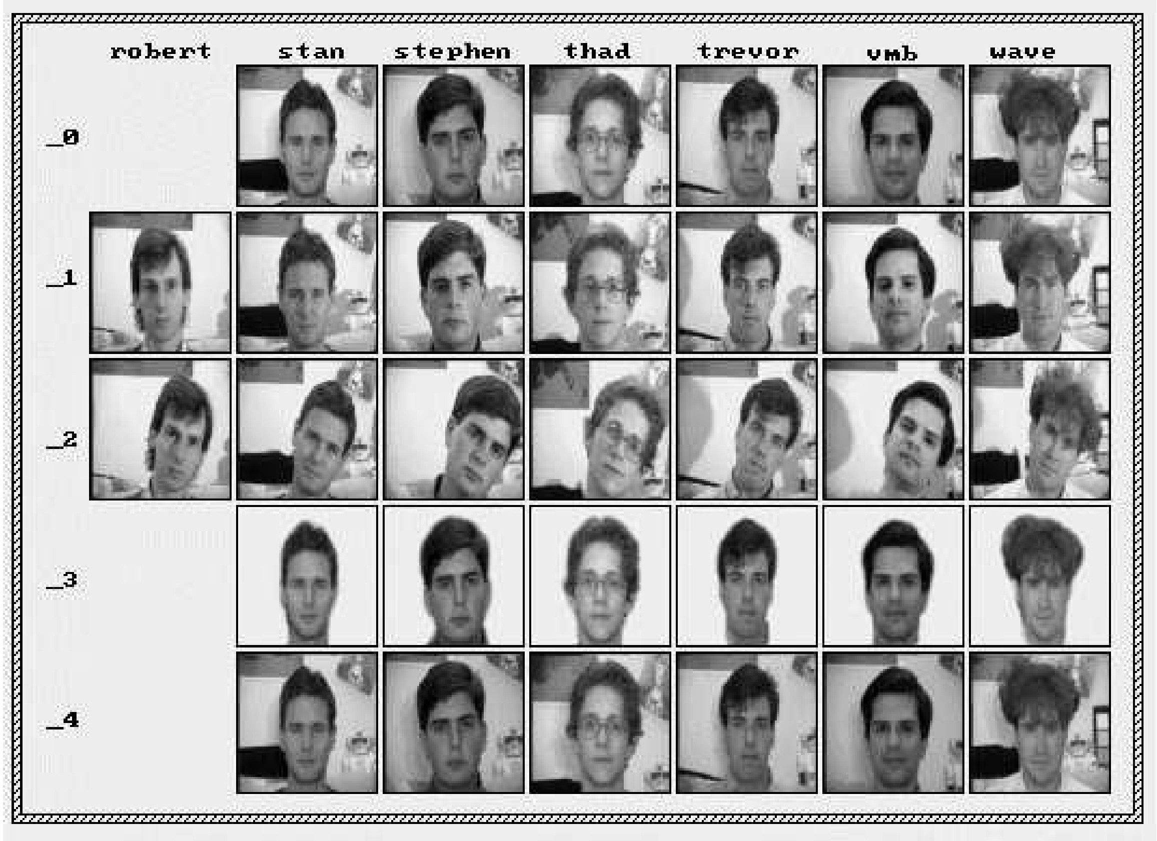


Figure 6. Original, manually decorated and rebuilt versions of sample face images.

Experimental Results

The proposed face recognition system was tested with face images obtained from the MIT face database. 14 individuals were captured under different lighting conditions and head orientations yielding a total number of 70 face images.


billj
david
foof
joel
mike
ming
pascal
_0
_1
_2
_3
_4


robert
stan
stephen
thad
trevor
vmb
wave
_0
_1
_2
_3
_4

Results Drawn from the Experiments

1. **Rebuild error ratio decreases** when the number of eigenfaces involved in recognition increases.

2. **Rebuild error ratio decreases** when face background is removed.

3. **Misclassification rate** is directly proportional with threshold value, where as miss rate is inversely proportional with threshold value.

4. **Misclassification rate** is inversely proportional with the number of eigenfaces, where as miss rate is directly proportional with the number of eigenfaces.

5. **The effect of illumination:** Pixel values of original face images were decreased 10% for this experiment. 92% correct classification was achieved within 45% threshold.

6. **The effect of light source position:** Original light source was moved 45 degrees to the left. 58% correct classification rate was achieved within 45% threshold. 17% of face images were misclassified and 25% of them were classified as “unknown”.

7. **The effect of head orientation:** Original head positions were rotated 45 degrees to the left. A correct classification rate of 14% was achieved yielding an “unknown” rate of 86% within 45% threshold.

8. **Presence of details:** Manually decorated face images were classified 100% correctly within 30% threshold.

9. **Recognition speed:** Once a training set is built, recognitions are performed in near real time.

Number of Eigenfaces	Update time (s) for 58 members
5	50
7	61
9	73
11	84

These experiments were performed on “FACE-PRO”, an MS-DOS based face recognition software that was written to demonstrate the viability of the eigenfaces approach.

Conclusions

Eigenfaces approach excels in its speed and simplicity and delivers good recognition performances under controlled conditions. Unlike feature based recognition, complex geometry is not involved. Training is done in an unsupervised manner. Experimental results show that, eigenfaces approach is very sensitive to face background and head orientations. Illumination and presence of details are reasonably simple problems for the proposed face recognition system.

Directions for Future Work

- Implementation of a background removal algorithm.
- Recognition from multiple views involving neural networks.
- Scanner and camera support.
- Migration to client/server architecture.