# Cost Effective Face Recognition Using a Web Cam

S. M. R. P. De Silva, H. M. N. Dilum Bandara, P. W. H. Dasun Weerasinghe

***Abstract*— This paper describes a facial recognition system using a Web cam. Typically Web cams do not provide enough resolution and quality required for facial recognition. Thereby it is impractical to use recognition algorithms that are based on facial features. Hence an approach that is also popular and well established known as the Eigenface method is employed. Eigenfaces approach can deal with the resolution and the quality of a Web cam without its accuracy being severely affected. The developed system provides satisfactory performance especially with regard to the total cost of ownership. The facial recognition system was developed as part of authors' final year project [1]. Thereby there are two versions of the facial recognition system, one modified to integrate with the final year project and the other a standalone facial recognition system.**

**Both versions have an immense value proposition. Differences between those two are also highlighted in this paper.**

***Index Terms*—Facial Recognition, Eigenfaces, PCA, BioAPI**

## I. INTRODUCTION

Face recognition uses the visible physical structure of an individual's face for recognition purposes. A typical face recognition system is explained below to illustrate the concept behind the rest of the section. (Re Fig.1)

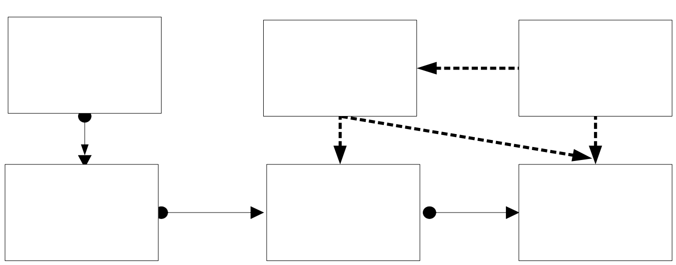

Fig. 1: Components of a typical Image processing system

*Image acquisition system*: This is the entry point to the face recognition process. The user presents an image to the system. An image capturing device is used to capture the image into digital format, like the web cam in authors' case.

The facial recognition system was developed as a part of the authors' final year project in partial fulfilment of the requirements of the degree of Bachelor of Science in Engineering, University of Moratuwa, Sri Lanka. Authors address: Department of Computer Science and Engineering, University of Moratuwa, Katubedda, Sri Lanka. E-mail: ravids@cse.mrt.ac.lk

*Pre-processing module:* Face images are enhanced to improve the recognition performance of the system. For example, noise (unwanted data present in the image due to environment or due to defects in the image acquisition system) is removed.

*Feature extraction module:* This module is responsible for composing a feature vector (a representation of the features in the face).

*Classification module:* Extracted features of the face image are compared with the ones stored in a face library (or face database). After 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 in order to achieve optimum recognition performance by making use of training sets.

*Face library or face database:* Face images are added to a library with their feature vectors for later comparisons.

## II. KEY FEATURES OF THE FACIAL RECOGNITION SYSTEM

It is cost effective. Images are captured through a low quality Web cam costing around 50 US$. The Web cam offers a resolution of $320\times240$ (without interpolation). Many popular face recognizing algorithms will prove inaccurate under these conditions. Hence the Eigenface approach, as explained later in the chapter is used. The Facial Recognition system supports identification and verification.

*Image Acquisition:* Microsoft DirectX is used to capture images through the Web cam. The UI (Re Fig.2) of the image capturing software was developed using Microsoft Foundation Classes (MFC). DirectX proved to be very powerful and useful than it was initially purported. After studying several samples shipped with DirectX, the authors' managed to implement the image capturing functionality reusing DirectX libraries and sample coding.

*Pre Processing:* The captured image was subjected to several image processing operations. Noise filtering was applied to remove any undue noise. Median Filtering was preferred over other filters, as it does not blur the image. For example the mean filter smoothes the image while removing noise. In object recognition, it is very important that images are sharp as much as possible.

Then the noise filtered image was sent through a background removal operation (i.e. remove the part of the image that does not include the object of interest, that is the face).

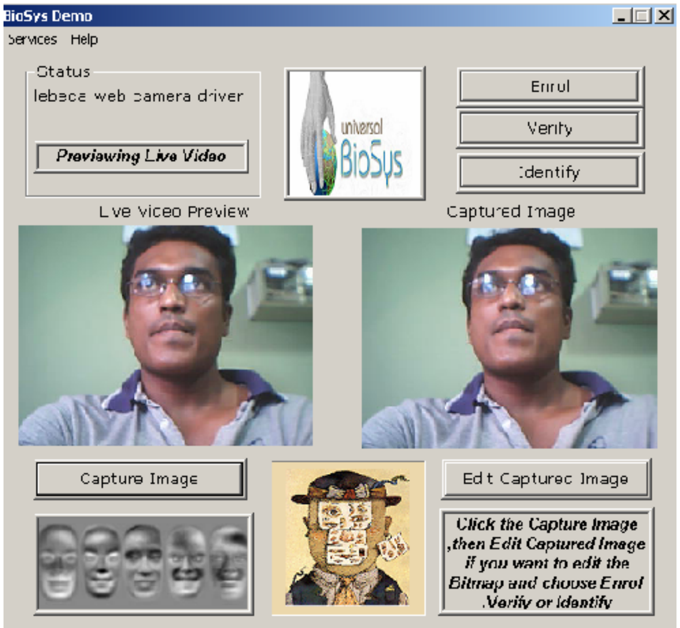


Fig. 2: GUI of the Facial Recognition System

*Eigen Face, Face Recognition:* This was the critical and complex area of the face recognizing system. The authors must thank Mr. Chathura de Silva for suggesting Eigenfaces for face recognition, when the authors were struggling with geometric feature based matching. To better understand the Eigen face concept, an insight into face recognition algorithms is given in the next few sub sections.

## III. Face Recognition Algorithms: BACKGROUND

Face recognition is a very complex activity in human brains without any solid explanation. Recognizing faces is different from finding faces. Face recognition is about describing who is the person, where as face detection is about locating a face in an input image or in a video sequence.

Many algorithms work as a two step process. First the image is projected in to a subspace (stored images make up the universal set spanning many sub spaces), and then a classification algorithm is used to identify the projected image in the context of the subspace.

Face recognition algorithms fall in to two categories. Principal Component Analysis (PCA) method is based on concepts in information theory. In this approach, only the most relevant information that best describes a face is derived from the entire face image. The second method is based on extracting feature vectors from the basic parts of a face such as eyes, nose, mouth, and chin. In this method, with the help of deformable templates and extensive mathematics, key information from the basic parts of a face is gathered and then converted into a feature vector.

*Dimensionality Reduction:* An image space has dimensions equal to the number of pixels in the image. Each dimension has values in the range of the pixels values. Thus, for example a grey scale image of size $( \mathrm{M} \times \mathrm{N} )$ has the dimension of $( \mathrm{M} \times \mathrm{N} )$. In the case of greyscale images, in each dimension the image could have a value in between 0 and 255 (0x00 to 0xFF). Computers need to store sufficient number of different views associated with each other object, for accurate recognition, and this requires a large memory space. Therefore it is required to reduce the storage per each stored item. This is what is meant by dimensionality reduction.

*Principal component analysis (PCA):* Central idea of PCA is to reduce the dimensionality of a data set, having a large number of interrelated variables, but retaining as much as possible of the variation present in the data set. Dimension reduction is achieved by transforming to a new set of variables (principal components, PCs), which are uncorrelated and ordered (so that the first few retain most of the variations). Principal component analysis, when applied to facial recognition seek a computational model that best describes a face, by extracting the most relevant information contained in that face.

## IV. Eigen Faces Algorithm

Eigenface approach is a variation of the PCA scheme. This approach decomposes face images into a set of characteristic feature images called Eigenfaces, which may be thought of as the principal components of the initial training set of face images. Recognition is performed by projecting a new image onto the subspace spanned by the Eigenfaces and then classifying the face by comparing its position in the face space with the positions of known individuals.

Eigenface method is based on correlation thereby on the variation between a set of images. The face images when converted into vectors, will group at a certain location in the image space due to the similar structure, each having eye, nose and mouth in common and their relative position correlated. The Eigenface method seeks to find a lower dimensional space for the representation of the face images by eliminating the variance due to non-face images by focusing only on the variation between the face images.

Complex mathematics is involved in calculating Eigenfaces. Once the Eigenfaces are available, a template for each library member needs to be calculated. The theory behind template calculation is simple. Each library image is a linear combination of the principal components (i.e. Eigenfaces). Mathematically speaking each image $\mathrm{I}_i$ is a linear combination of the set of Eigen faces (K no of Eigenfaces) $v = [v_1, v_2, v_3, ......v_\mathrm{K}]$, and $[w_1, w_2, ..........w_\mathrm{K}]$ are the weights in the particular linear combination.

$$\mathrm{I}_i = w_1 v_1, w_2 v_2, ......, w_\mathrm{K} v_\mathrm{K} \qquad (1)$$

Weight vector associated with each library image uniquely identifies each library image.

Therefore each face is represented by two entries in the face library: One entry corresponds to the face image itself and the other corresponds to the weight vector associated with each face image.

Eigenfaces approach is very valuable in face recognition due to its simplicity, speed and learning capability.

## V. Eigen Face Calculation

An image can be thought as a vector in the image space through the concatenation of each column of the image one after the other. (Re Fig. 3)

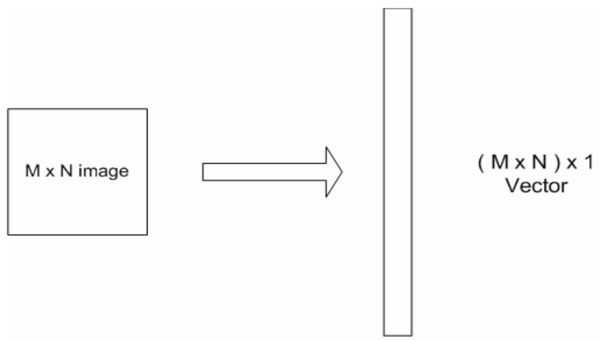


Fig. 3: Vector representation of an image

Let us denote an image by $\mathrm{I}_i$ which is a $(\mathrm{M} \times \mathrm{N}) \times 1$ vector.

And denote the set of images (K no of images) by $\mathrm{X} = [\mathrm{I}_1, \mathrm{I}_2, \mathrm{I}_3, ......\mathrm{I}_\mathrm{K}]$.

In the Eigen face method, features of studied images are obtained by looking for the maximum deviation of each image from the mean image. Therefore the mean image $\Psi$ is calculated as

$$\Psi = \frac{1}{\mathrm{K}} \sum_{i=1}^{i=\mathrm{K}} \mathrm{I}_i \qquad (2)$$

Then the deviation from the mean is calculated as $\Phi_i$ , which is the deviation of image $\mathrm{I}_i$ from the mean image.

$$\Phi_i = \mathrm{I}_i - \Psi \qquad (3)$$

Thus the set of mean subtracted images, $\mathrm{A} = [\Phi_1, \Phi_2, \Phi_3, ......\Phi_\mathrm{K}]$ are calculated.

This variance is obtained by calculating the Eigen vectors of the covariance matrix of the calculated mean subtracted images.

Covariance matrix $= \mathrm{AA}^\mathrm{T}$. But the resulting covariance matrix is huge for a considerable number of images, as explained below.

The dimension of $\mathrm{A} = (\mathrm{M} \times \mathrm{N}) \times \mathrm{K}$

And thereby dimension of $\mathrm{A}^\mathrm{T} = \mathrm{K} \times (\mathrm{M} \times \mathrm{N})$

Hence $\mathrm{AA}^\mathrm{T}$ will have dimension of $(\mathrm{M} \times \mathrm{N}) \times (\mathrm{M} \times \mathrm{N})$

Used Web cam provides a resolution of $320 \times 240$

Thus in authors case $\mathrm{M} = 320$ and $\mathrm{N} = 240$.

Therefore dimension of $\mathrm{AA}^\mathrm{T} = 76800 \times 76800$

It will be computationally intensive to find the Eigen values of this huge matrix. But fortunately through the Eigen values of $\mathrm{A}^\mathrm{T}\mathrm{A}$ , Eigen values of $\mathrm{AA}^\mathrm{T}$ can be calculated, without loss of accuracy. There is an elegant proof to support it [1]. Since $\mathrm{A}^\mathrm{T}\mathrm{A}$ is of dimension $\mathrm{K} \times \mathrm{K}$ , and typically the image base will be very much less than 76800 (i.e. typically K<<76800), a huge amount of computation can be reduced.

The eigenvectors are produced in such a way that the first vector captures most of the variance in the images, the second captures most of the variance once the first has been removed, and so on.

Once Eigen vectors of $\mathrm{AA}^\mathrm{T}$ are calculated, each of these vectors will be an Eigenface.

The word Eigenface comes from the fact that they are generated from Eigen vectors and have a face like appearance.

Let $\nu = [\nu_1, \nu_2, \nu_3, ......\nu_\mathrm{K}]$ be the set of Eigen faces.

Out of these, Eigenfaces that have the largest Eigen values account for the most variance within the set of library face images. Therefore only those best Eigenfaces needs to be used, without a significant loss of accuracy. The Eigen values are ordered in descending order. Thereafter through experimental procedures, one can determine the number of Eigen values (vectors), to be considered without significant loss of accuracy in the context of that application. The best **E** Eigenfaces span an **E**-dimensional subspace, which is referred as the "face space" of all possible images.

This set of best vectors (i.e. best Eigenfaces) will be saved to the hard disk, for encoding of the library images, as explained in the next section.

Note that as new faces are experienced, the Eigenfaces needs to be recalculated.

## VI. Encoding of Faces

The concept behind encoding of library images is that each individual face can be represented exactly in terms of a linear combination of the Eigenfaces.

Therefore the next step is to project each image (mean subtracted image) into the Eigen vector space.

The projection is identified by a set of coefficients referred as weights and this set of coefficients is the encoding.

Let Weights of the $\mathrm{I}_i$ (actually the projection of $\Phi_i$) be

$$W_i = \nu^\mathrm{T} \Phi_i \qquad (4)$$

Therefore the weights matrix $W$ , which is the matrix of weight vectors, is given by

$$W = \nu^\mathrm{T} \mathrm{A} \qquad (5)$$

## VII. Recognition and Learning Phase

After the construction of weight vectors of face library members, now the system is ready to perform the recognition process. In the Eigenfaces approach, recognition is very obvious and simple. The mean image is subtracted from the test image and projected into the Eigenface vector space. After obtaining the weight vector, it is compared with the weight vector of every face library member within a user defined "threshold". The method of comparison the authors have used is the Euclidian distance. That is, for one stored weight vector, the difference between each weight and the corresponding test user weight is calculated then squared and squared values are summed to obtain the Euclidian distance.

## VIII. SERVICES OF THE FACIAL RECOGNITION SYSTEM

There are two versions of the facial recognition system. One is a stand-alone system. Other one was developed with the aim of integrating with another component in authors' final year project. Second version is BioAPI compliant (BioAPI is the Defacto standard for biometrics) and demonstrates the recommended method of adhering to the BioAPI standard. BioAPI recommends recognition algorithms to be present in a centralized server when it is used in a network environment and where the image capturing devices are attached to the network. Hence in that version the image capturing device and the location at which recognition is performed is different. The usage of that version is also described as it may have a huge potential in the future with establishment and wide spread of BioAPI.

*1) Enrolment*

Enrolment service of the Eigenfaces approach is quite different to geometric feature based methods. In those methods, at each enrolment, geometric features of the new image are examined and a template is saved.

Thereby each image is independent. But in the Eigenface method, at each enrolment a new set of Eigen faces needs to be built, as explained earlier. The new Eigenfaces are built from the existing images with the inclusion of the new image. Then the weight vectors of each and every image will be recalculated. Thereby enrolment takes more time compared to other methods.

The process is slightly different and complex in the BioAPI compliant system. The biggest issue was that the BioAPI expects a single template as the result of enrolment. But in the Eigenfaces approach the result of an enrolment is mainly the set of Eigenfaces and the weight vectors. Since only one template can be returned, there was no other option other than to save the captured image as the template (theoretically the template that is saved for each user in the database is a set of double coefficients). Thereby only the captured images are saved and at each verification or identification the Eigenfaces needs to be calculated, and so do the weight vectors.

*2) Identification*

In recognition systems the input to the identification function is a set of stored images plus the image of the test user. The system tries to identify the best match for the test image from the stored images.

The logic of identification is very clear with traditional geometric based approaches. Simply calculate the template of the test image and compare with stored templates. And to create templates of test users, the server does not need any other image.

Eigenface approach gives a performance boost in this aspect.

At enrolment the weight vector of each existing user was calculated and saved. These weight vectors are the templates of those users. Therefore what is required is to calculate the weight vector of the potential user and compare with the existing, saved weight vectors. Thereby theoretically the weight vectors of the existing users and the potential image need to be inputted to the identification function. Furthermore to calculate the weight vectors of the potential user the saved Eigenfaces are required. Hence the set of Eigenfaces is also an input to identification.

Then the test image is classified using the Euclidean distance between its weight vector and the weight vectors of the stored images (the stored image with the least Euclidian distance is identified). Verification is very much similar, where only one stored image is inputted. As the identification process primarily deals with a small number of double coefficients, it is very much faster compared to feature extraction methods.

In the BioAPI complaint system, as the weight vectors of the existing users are not saved, weight vectors of those also need to be calculated, thereby severely affecting performance of the identification process.

The verification process follows the same guidelines with less complexity.

## IX. PERFORMANCE OPTIMIZATION

The development of the facial recognition system involved lot of matrix manipulations and computations. Therefore it was necessary to use libraries to optimize those computations. The obvious choice was OpenCV [4] the well established Open Source library of Intel Performance Libraries. The Intel Performance Libraries are a set of optimized C++ libraries providing simpler algorithmic development for scientific and mathematical applications, similar to the Matlab platform, but with the advantage of fully compiled (and speedy) C code.

## X. EXPERIMENTAL RESULTS

Several tests were carried out on the stand-alone facial recognition system to evaluate accuracy and the speed of operation. Training data consisted of 50 images (Re Fig.4). There were 45 images from 15 individuals with 3 images per each. The other 5 were from 5 other individuals who closely resembled some of the 15 members. Frontal illumination was applied during the image acquiring process using the Web cam.

Fig. 4: Image Database

As test data, an image not contained in the above image database of each member was used. Likewise 15 images were used as test data. Each test image was used to test both identification and verification.

In the Eigenface approach it is possible to use a subset (prominent) of the generated Eigenfaces without a significant loss of accuracy leading to faster identification and verification.

As a low quality Web cam is employed in this situation it was decided to employ all the Eigenfaces ignoring the impact on the speed of operation. The figure below (Re Fig.5) illustrates some of the Eigenfaces generated for the above image database.

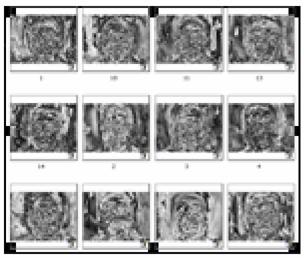

Fig. 5: Sample Eigenfaces

*1) Accuracy*

In About 85% of the test cases, system was able to identify the best match in the identification process.

The accuracy of verification varied a great deal with the False Acceptance threshold. Even with a threshold of 10% the system was able to verify the users without any false rejections. With the increase in threshold it gave several false acceptances. The level of threshold that started to give away false acceptances for this image database was 30%.

*2) Speed of operation*

The tests were carried out on an Intel Pentium IV computer. As the intention of speed of operation tests was to compare enolment, verification, and identification the configuration of the testing machine can be ignored.

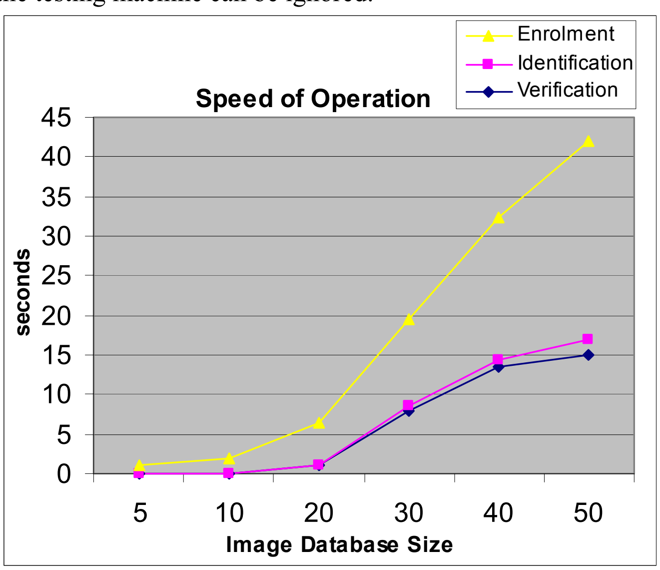


Fig. 6: Speed of Operation Comparison Graph

Above result (Re Fig.6) can be explained as follows. At enrolment all the Eigenfaces and weights needs to be recalculated for all the images. Due to that and the fact that Eigenfaces equal to the number of stored images are saved, enrolment takes a significant time to complete with the increase in the image database size. Verification and identification are not severely affected by the size of the image database as their main computations deal with double coefficients. Ideally verification should have taken less time compared to the identification process. At present weight vectors of both the input image and the purported user are calculated, but the weigh vector of the purported user could be read instead of recalculating. The authors hope to modify that as well as many other areas that would improve the performance.

## XI. Limitations

The Eigenface approach has several drawbacks thereby imposing certain limits on the developed facial recognition system. Eigenface approach is susceptible to changes in faces, which is typical with age. Accuracy of any face recognition algorithm fails with changes in the stored images, but Eigen face method is affected severely. At each enrolment, all the stored templates are recalculated, which is another drawback, but verification and identification works on simple and small templates, enjoying a significant speed factor. As enrolment is the less frequent case, it is not that much of an issue. Furthermore Eigenface approach is dependent on face orientation, lighting and other environmental conditions more than feature extraction methods. Hence it is recommended to maintain those factors consistently for better accuracy.

## XII. Conclusion

Facial recognition with a Web cam is an immense challenge, as Web cams do not offer the resolution and the quality required for recognition purposes. Thereby the authors used the Eigenface approach, which can work with the resolution and the quality a Web cam offers, but this approach has certain drawbacks. High susceptibility to changes in faces is the main problem. This system incorporates a pre-processing stage to improve the recognition. The pre-processing stage can be further enhanced; finding the threshold to remove the background using histogram analysis is one such possibility. At present the threshold is hard coded, based on the experiments results with the image database. Though there are future areas of improvement, the facial recognition system is a fully functional product with satisfactory performance. The authors are proud to be able to develop the Facial Recognition system with a comparatively low cost of ownership

## Acknowledgment

Over the last six months many people have contributed to this project. In particular, authors thank Mr. Chathura de Silva who was the project supervisor as well as the final year project coordinator. Authors also thank Dr. Sanath Jayasena the Head, Department of Computer Science and Engineering and other staff members. Finally many thanks goes to authors own batch mates, system support team, non-academic staff of the department who helped in numerous ways and each and very individual that authors have not mentioned above, who gave the slightest support through the lifetime of the project.

## References

[1] Ilker Atalay, "Face Recognition using Eigenfaces", January, 1996 Available: http://www.ilkeratalay.com/facerecognition.php.

[2] Universal BioSys official web site (Authors' project website) www.biosys.net.tc

[3] The BioAPI Consortium, "BioAPI Specification Version 1.1", 16th March 2004. Available: http://www.bioapi.org

[4] OpenCV Available: http://www.intel.com/research/mrl/research/opencv Dowloads at (Version Beta 3.1 was released Feb. 2003) http://sourceforge.net/projects/opencvlibrary