Informing Science Special Issue on Multimedia Informing Technologies – Part 2 Volume 3 No 1, 2000

An Introduction to Face Recognition Technology

Shang-Hung Lin, Ph.D.
IC Media Corporation
shanghung.lin@ic-media.com

Abstract

Recently face recognition is attracting much attention in the society of network multimedia information access. Areas such as network security, content indexing and retrieval, and video compression benefits from face recognition technology because "people" are the center of attention in a lot of video. Network access control via face recognition not only makes hackers virtually impossible to steal one's "password", but also increases the user-friendliness in human-computer interaction. Indexing and/or retrieving video data based on the appearances of particular persons will be useful for users such as news reporters, political scientists, and moviegoers. For the applications of videophone and teleconferencing, the assistance of face recognition also provides a more efficient coding scheme. In this paper, we give an introductory course of this new information processing technology. The paper shows the readers the generic framework for the face recognition system, and the variants that are frequently encountered by the face recognizer. Several famous face recognition algorithms, such as eigenfaces and neural networks, will also be explained.

Keywords: Face Recognition; Biometric Identification; Network Security and Surveillance; Neural Network; Karhonen-Loeve (KL) Transform; Image Processing

Introduction

In today's networked world, the need to maintain the security of information or physical property is becoming both increasingly important and increasingly difficult. From time to time we hear about the crimes of credit card fraud, computer break-in's by hackers, or security breaches in a company or government building. In the year 1998, sophisticated cyber crooks caused well over US $100 million in losses (Reuters, 1999). In most of these crimes, the criminals were taking advantage of a fundamental flaw in the conventional access control systems: the systems do not grant access by "who we are", but by "what we have", such as ID cards, keys, passwords, PIN numbers, or mother's maiden name. None of these means are really define us. Rather, they merely are means to authenticate us. It goes without saying that if someone steals, duplicates, or acquires these identity means, he or she will be able to access our data or our personal property any time they want. Recently, technology became available to allow verification of "true" individual identity. This technology is based in a field called "biometrics". Biometric access control are automated methods of verifying or recognizing the identity of a living person on the basis of some physiological characteristics, such as fingerprints or facial features, or some aspects of the person's behavior, like his/her handwriting style or keystroke patterns. Since biometric systems identify a person by biological characteristics, they are difficult to forge.

Among the various biometric ID methods, the *physiological* methods (fingerprint, face, DNA) are more stable than methods in behavioral category (keystroke, voice print). The reason is that physiological features are often non-alterable except by severe injury. The behavioral patterns, on the other hand, may fluctuate due to stress, fatigue, or illness. However, behavioral IDs have the advantage of being *non-intrusiveness*. People are more comfortable signing their names or speaking to a microphone than placing their eyes before a scanner or giving a drop of blood for DNA sequencing.

Face recognition is one of the few biometric methods that possess the merits of both high accuracy and low intrusiveness. It has the accuracy of a physiological approach without being intrusive. For this reason, since the early 70's (Kelly, 1970), face recognition has drawn the attention of researchers in fields from security, psychology, and image processing, to computer vision. Numerous algorithms have been proposed for face recognition; for detailed survey please see Chellappa (1995) and Zhang (1997).

While network security and access control are it most widely discussed applications, face recognition has also proven useful in other multimedia information processing areas. Chan *et al*. (1998) use face recognition techniques to browse video database to find out shots of particular people. Li *et al*. (1993) code the face images with a compact parameterized facial model for low-bandwidth communication applications such as videophone and teleconferencing.

Recently, as the technology has matured, commercial products (such as Miros' TrueFace (1999) and Visionics' FaceIt (1999)) have appeared on the market. Despite the commercial success of those face recognition products, a few research issues remain to be explored. In the next section, we will begin our study of face recognition by discussing several metrics to evaluate the recognition performance. Section 3 provides a framework for a generic face recognition algorithm. Then in Section 4 we discuss the various factors that affect the performance of the face recognition system. In section 5, we show the readers several famous face recognition examples, such as eigenface and neural network. Then finally a conclusion is given in section 6.

Performance Evaluation Metrics

The two standard biometric measures to indicate the identifying power are **False Rejection Rate** (FRR) and **False Acceptance Rate** (FAR). FRR (Type I Error) and FAR (Type II Error) are inversely proportional measurements; For example, if an ID system tunes its threshold value to reject all imposters (minimizing FAR), it may also improperly reject some authorized users (maximizing FRR). Therefore, ID system designers often provide a variable threshold setting for the customers to strike a balance. If a site requires near 100% rejection of imposters, authorized users will have to suffer, say, a 4% rejection rate.

An ID system with *both* low FAR and FRR is considered having good discriminating power. ID system developers always make effort to reduce the area under the FAR-FRR curve (a practical indicator of this integral is the value where FAR equals to FRR. Usually the lower the equal value is, the better identifying power the system possesses). Developers need to select a good sensor device, a proper feature extraction scheme, and/or a powerful pattern matcher to achieve this task.

Generic Framework

In most cases, a face recognition algorithm can be divided into the following functional modules: **a face image detector** finds the locations of human faces from a normal picture against simple or complex background, and a **face recognizer** determines who this person is. Both the face detector and the face recognizer follow the same framework; they both have a **feature extractor** that transforms the pixels of the facial image into a useful vector representation, and a **pattern recognizer** that searches the database to find the best match to the incoming face image. The difference between the two is the following; in the face detection scenario, the pattern recognizer categorizes he incoming feature vector to one of the two image classes: "face" images and "non-face images. In the face recognition scenario, on the other hand, the recognizer classifies the feature vector (assuming it is from a "face" image) as "Smith's face", "Jane's face", or some other person's face that is already registered in the database. Figure 1 de-

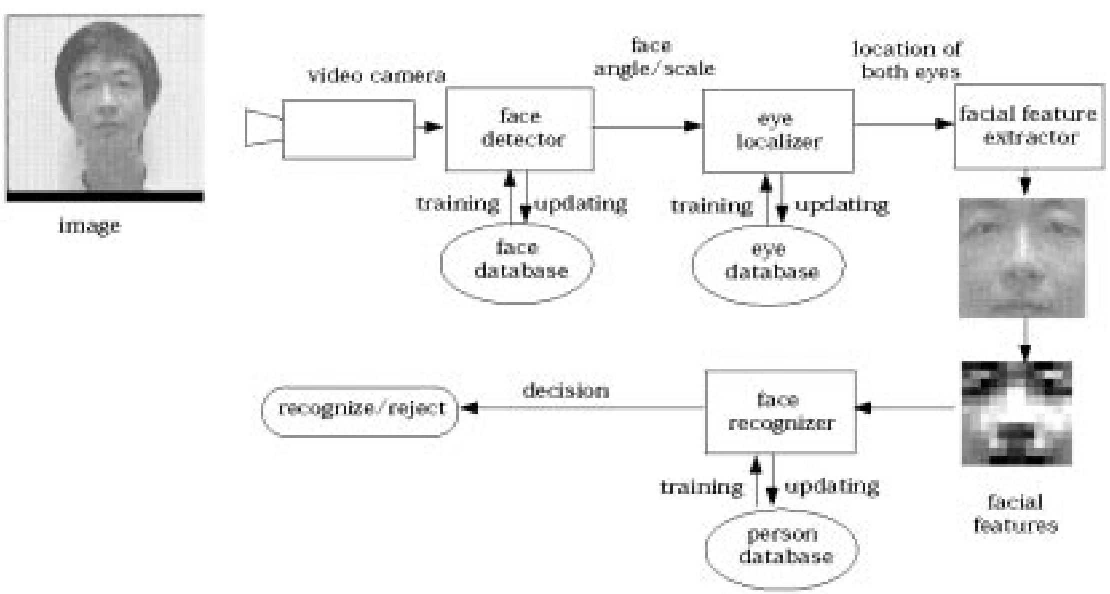

Figure 1. A framework for face recognition system.

picts one example of the face recognition system. Notice that there is an additional "eye localizer" in this system. Due to all kinds of the variations that may occur in the image (as discussed below), the face detector usually produces only an "approximate" location of the face. In order to find the "exact" location of the face for recognition, face recognition system designers often use the locations of both eyes for assistance. In this system, all the three functional modules (face detector, eye localizer, and face recognizer) follow the "feature extractor + pattern recognizer" schematics.

Feature Extraction

We start from modeling the image of a face as a two dimensional array of numbers, i.e., pixel values. It can be written as $X = \{x_i, i \in S\}$ where **S** is a square lattice. Sometimes it is more convenient to express **X** as a one dimensional column vector of concatenated row of pixels, $X = [x_1, x_2, \ldots, x_N]^T$ where **N** is the total number of pixels in the image. For a quarter VGA image (320x240), N is as large as 76,800. Such a high dimensional feature is usually inefficient and also lacks of discriminating power. Therefore, we need to transform X into a *feature vector*` $\mathrm{f}(X) = [f_1(X), f_2(X), \ldots, f_M(X)]^T$ where *f1(x), f2(x),...* are linear or nonlinear functionals. To increase the efficiency of the new representation, generally **M** is much smaller than **N**.

Pattern Recognition

Due to variants such as viewing angles, illumination, facial expression and so on, the facial feature vector obtained from previous equations can have random variations and therefore it is better modeled as a random vector. If the incoming person is equally likely to be any person in the database (equal *a priori* probability), then according to Bayes decision theory, the minimum recognition error rate can be achieved if the recognition is following the *maximum-likelihood* (ML) criterion. That is, suppose Y = f(X) is the feature vector and suppose that there are K persons in the database, the identity of the incoming person is assigned by $k_0 = \arg\min_{1 \le k \le K} \log p(Y \mid k)$ where p(Y|k) is the likelihood density of Y conditioning on its being the kth person.

If we assume the variations in the facial feature vector are caused by zero-mean, additive white Gaussian noise (AWGN), then the ML matching becomes the common *minimum distance* matching. That is, the identity of the incoming person is k if the Euclidean distance between the feature vector of the incoming person and the mean vector of the kth person is the smallest among all people in the database. Unfortunately, in the real world situation the variations of facial images are much more complex than AWGN. In the following section we will describe various kinds of variants that may appear in the face recognition problem.

Variations in Facial Images

Face recognition is one of the most difficult problems in the research area of image recognition. A human face is not only a 3-D object, it is also a non-rigid body. Moreover, facial images are often taken under natural environment. That is, the image background could be very complex and the illumination condition could be drastic. Figure 2 is an example of an image with a complex background.

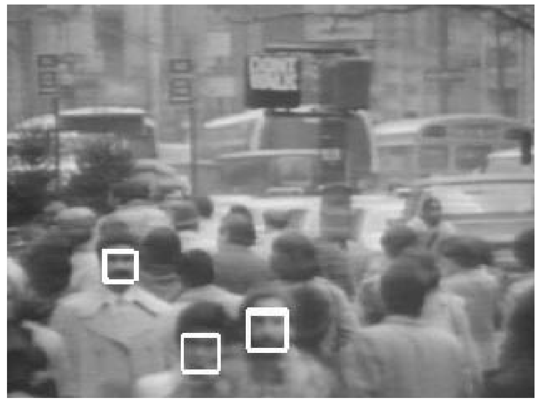

Figure 2. The face detector found three faces from a complex background.

The variations in facial images could be categorized as follows:

- Camera distortion and noise
- Complex background
- Illumination
- Translation, rotation, scaling, and occlusion
- Facial expression
- Makeup and hair style

Camera distortion and noise are standard variations in image recognition problems. Previous researchers have developed numerous tools to increase the signal-to-noise ratio. To deal with complex image background, the recognizer requires a good face detector to isolate the real faces from other parts of the image. Illumination is often a major factor in the obstruction of the recognition process. To alleviate the influence of the illumination effect, people may take conventional image enhancement techniques (dynamic thresholding, histogram equalization), or train a neural network for feature extraction (Brunelli, 1993)(Lin, 1997). Another approach to reduce the illumination effect is using the eigenface method. As will be mentioned in the later section, eigenface algorithm reduces the high dimensional feature space into a low dimensional subspace where most of the energy resides (i.e., eigenspace). According to literature (Pentland, 1994), one or a few **eigen-**

faces (terminology for the eigenvectors in the eigenface algorithm) could be used to represent the "illumination effect" on facial images. Therefore putting lower weighting on those eigenfaces when doing the recognition reduces the illumination effect. Yet another remedy for illumination variation is using the **Fisherface** algorithm. Fisherface algorithm is a refinement of the eigenface algorithm. It further reduces the eigenspace by the Fisher's linear discriminant (FLD). FLD selects the subspace in such a way that the ratio of the between-class scatter and the within-class scatter is maximized. It is reported that the fisherface algorithm outperforms the eigenface algorithm on the facial database with wide variation in lighting condition (Belhumeur, 1997). (The detail of the Fisherface algorithm will not be covered in this paper. Interested readers please refer to (Belhumeur, 1997).)

Translation, scaling, and rotational variations should also be dealt with in the face detection phase. Among the three variations, translation is the easiest one to solve. A simple windowing approach can do the job. Scaling problem (different face sizes) is also easy to solve if we create an image pyramid to represent the input image (image pyramid is a collection of the same image with different resolutions). Rotation along the Z axis (the axis that is perpendicular to the image plane) is harder. A brute force solution is time-consuming. Rowley *et al.* (Rowley, 1998) build a “router network” in front of their face detector to determine the Z-axis rotation angle of the image patch. The image patch will be rotated back to upright position based on the output of the router network. The hardest problem is to handle the rotation along the X axis, Y axis, or the combination of the two. These kinds of rotation may cause the occlusion of the face. An occluded face is usually not suitable for recognition. Therefore, recognition system designers often bypass this problem by using the face detector that selects only the frontal facial images from the video stream (Lin, 1997). Figure 3 shows a test result of a face detector who is designed to recognize only the frontal faces.

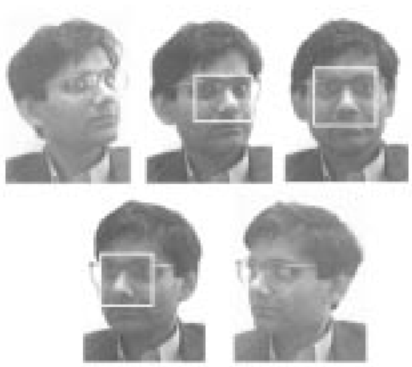

Figure 3. A face detector who detects frontal faces only.

Facial expression and hairstyle changes are yet other two important variations. A smiling face and a frowning face are considered as two totally different image templates in rigid-body sense. In order to overcome this "surface deformity", *elastic matching* algorithm trains a 2D mesh-like neural network to model the face surface. If the mesh (the deformable template) is successfully trained, it is possible to "correct" the expression changes when doing the recognition (Lades, 1993).

Another way to deal with facial expression changes is, instead of using the whole facial area to perform recognition task, using only the "significant facial region". The significant facial region is a square area close to the center of the human face. It contains both eyes and the nose, but excludes the mouth and ears. Study shows that facial expressions and hairstyle changes have less influence on the significant facial region, and yet the face is still recognizable by viewing only the significant facial region (Lin, 1997). Figure 4 shows the significant facial region.

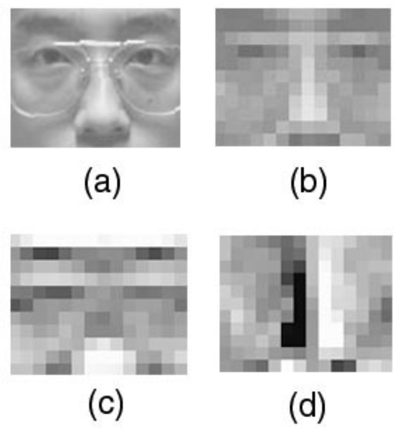

Figure 4. (a) Significant facial region. (b) The downsampled image of the region. (c) X-directional gradient image map. (d) Y-directional gradient image map.

Facial makeup is less influential than other facial variations unless it is a theatrical cosmetic. Usually the face recognition system requires a certain amount of user's cooperation on this problem; that is, if you are an enrolled member in the database, it is better you do not wear a rubber mask when you are requesting for admission. Figure 5 is an example of how facial makeup affects the performance of the face detection.

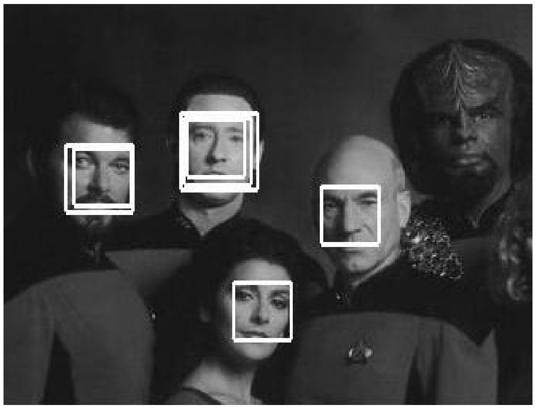

Figure 5. An example of face detection. Notice that the Klingon's face (rightmost person) is not detected.

Face Recognition Algorithms

In the previous section we have shown that the task of face recognition encounters complex variations. In order to cope with such complication and find out the true invariant for recognition, researchers have developed various recognition algorithms. In this section, we will describe two representative ones. The **eigenface** approach applies the Karhonen-Loeve (KL) transform for feature extraction. It greatly reduces the facial feature dimension and yet maintains reasonable discriminating power. The **neural network** approach, though some variants of the algorithm work on feature extraction as well, mainly provides sophisticated modeling scheme for estimating likelihood densities in the pattern recognition phase.

Eigenface

As mentioned, one of the goals that the feature extraction routine wishes to achieve is to increase the efficiency. One simple way to achieve this goal is using alternative orthonormal bases other than the natural bases. One such basis is the Karhonen-Loeve (KL). KL bases are formed by the eigenvectors of the covariance matrix of the face vector X. In the high dimensional "face" space, only the first few eigenvalues have large values. In other words, energy mainly locates in the subspace constituted by the first few eigenvectors. Therefore, a great compression can be achieved by letting those eigenvectors with large eigenvalues to represent the face vector X,

$X \cong \sum_{i=1}^{M} \hat{x}_i u_i$ where u is the eigenvector and M is usually much smaller than original vector dimension N. Since the eigenvectors associated with the first few eigenvalues look like face images, KL bases are also referred to as eigenfaces.

The eigenface representation is well known in statistics literature as the principal component analysis. It is optimal in the sense of efficiency: for any given M < N, the KL representation has the minimum mean square error among all possible approximations of X that uses M orthonormal vectors. However, it does not mean that the KL representation is optimal in the sense of discriminating power, which relies more on the *separation* between different faces rather than the spread of all faces.

Pentland's Photobook is one implementation of the eigenface algorithm. It compresses a facial image with 128x128 pixels (16,384 pixels) into a vector with only 40 eigenfaces (80 bytes). It recognizes 95% of the 200 faces chosen from a large database with 7562 facial images (3000 different persons) (Pentland, 1994).

Neural Network

In principle, the popular back-propagation neural network may be trained to recognize face images directly. For even an image with moderate size, however, the network can be very complex and therefore difficult to train. For example, if the image is 128x128 pixels, the number of inputs of the network would be 16,384. To reduce complexity, neural network is often applied to the pattern recognition phase rather than to the feature extraction phase. Sung and Poggio's face detection algorithm (Sung, 1995) down-samples a face image into a 19x19 facial feature vector before they apply the *elliptical k-mean* clustering to model the distributions of the "face samples" and the "non-face samples". Rowley *et al.* (Rowley, 1998) also reduce the dimension of the facial image to 20x20 by downsampling before the facial image is fed into their multi-layer neural network face detector.

One example of the neural classifier is the Probabilistic Decision-based Neural Network (PDNN) (Lin, 1997). PDNN does not have the fully connected network topology. Instead, it divides the network into K subnets. Each subnet is dedicated to recognize one person in the database. PDNN uses the Gaussian activation function for its neurons, and the output of each "face subnet" is the weighted summation of the neuron outputs. In other words, the face subnet estimates the likelihood density using the popular **mixture-of-Gaussian** model. Compared to the AWGN scheme, mixture of Gaussian provides a much more flexible and complex model for approximating the true likelihood densities in the face space.

The learning scheme of the PDNN consists of two phases. In the first phase, each subnet is trained by its own face images (no cross training). In this phase, the weights and bias are trained by the *Expectation-Maximization* (EM) algorithm (Dempster, 1976). The EM has been proven to be an efficient algorithm for ML estimation. In terms of system implementation, the advantage of the EM algorithm is that it does not require the learning rate parameter. The learning rate parameter in the conventional neural network training scheme controls the speed of adjusting the network weights. The learning rate is often a sensitive parameter; an improper selection may cause the whole network fail to converge.

The second phase of the PDNN learning is called the *decision-based learning*. In this phase, the subnet parameters may be trained by some particular samples from other face classes. The decision-based learning scheme does not use all the training samples for the training. Only those who are mis-classified are used. If a sample is mis-classified to the wrong subnet, the rightful subnet will tune its parameters so that its "territory"(decision region) can be moved closer to the mis-classified sample. This learning process is also known as the **Reinforced Learning**. In the meantime, the subnet that wrongfully claims the identity of the questionable sample will try to move itself away from the sample. This is called the **Anti-reinforced Learning**.

Lin (1997) reports a performance comparison of various face recognition algorithms (see Table 1). This comparison adopted the public Olivetti facial database. There are 40 different persons in the database and 10 images per person. There are variations in facial expression (open/close eyes, smiling/frowning), facial details (with or without glasses), scale (up to 10%) and orientation (up to 20 degrees). Five face recognition algorithms were compared in this experiment. PDNN recognizer achieved 4% error rate. It outperformed the eigenface-based recognizer, whose error rate is 10%. The other three algorithms are SOM+CN (self-organized map with convolutional neural network), HMM (Hidden Markov Model), and Pseudo 2D-HMM.

System	Error Rate	Classification Time	Training Time
PDNN	4%	< 0.1 sec	20 min
SOM+CN	3.8%	<0.5 sec	4 hr
Pseudo 2D-HMM	5%	240 sec	N/A
Eigenface	10%	N/A	N/A
HMM	13%	N/A	N/A

Table 1. Performance of different face recognizer on the Olivetti database. This table is adapted from Lin (1997).

There are many other face recognition algorithms that are not discussed in this paper, such as elastic matching, HMM and convolutional neural network. Readers who are interested are encouraged to go to (Zhang, 1997) and (Chellappa, 1995) for more thorough survey.

Conclusion

Face recognition is a both challenging and important recognition technique. Among all the biometric techniques, face recognition approach possesses one great advantage, which is its user-friendliness (or non-intrusiveness). In this paper, we have given an introductory survey for the face recognition technology. We have covered issues such as the generic framework for face recognition, factors that may affect the performance of the recognizer, and several state-of-the-art face recognition algorithms. We hope this paper can provide the readers a better understanding about face recognition, and we encourage the readers who are interested in this topic to go to the references for more detailed study.

References

Belhumeur, P., hespanha, J., and Kriegman, D., "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 19, no. 7, pp. 711-720, July 1997.

Brunelli, R., and Poggio, T., “Face Recognition: Features versus Templates”, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp.1042-1052, 1993.

Chan, Y., Lin, S.-H., and Kung, S.Y., "Video Indexing and Retrieval", *Multimedia Technology for Applications*, Sheu and Ismail editors. IEEE Press, 1998.

Chellappa, R., Wilson, C.L., and Sirohey, S., “Human and Machinr Recognition of Faces: A Survey”, *Proc. IEEE*, vol.83, pp.705-741, May 1995.

Dempster, A.P., Laird, N.M., and Rubin, D.B., “maximum Likelihood from Incomplete Data via the EM Algorithm”, *Journal of Royal Statistics Society*, B39, pp.1-38, 1976.

Kelly, M.D., “Visual Identification of People by Computer”, *Stanford AI Project*, Stanford, CA, Technical Report. AI-130, 1970.

Lades, M., *et al*, “Distortion Invariant Object Recognition in the Dynamic Link Architecture”, IEEE Trans. Computers, vol. 42, No. 3, pp.300-311, 1993.

Li, H., Roivainen, P., and Forchheimer, R., "3-D Motion Estimation in Model-Based Facial Image Coding", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 6, pp.545-555, 1993.

Lin, S.-H., Kung, S.Y., and Lin, L.-J., “Face Recognition/Detection by Probabilistic Decision-Based Neural Network”, *IEEE Trans. Neural Networks*, vol. 8, No. 1, pp.114-132, Jan. 1997.

Miros, “TrueFace Stands Alone on NT”, http://www.miros.com/TrueFace_NT_Standalone_1_99.htm (March 10, 1999).

Pentland, A.P., Moghaddam, B., Starner, T., and Turk, M.A., “View-based and Modular Eigenspaces for Face Recognition”, *Proc. IEEE*

Computer Society Conf. Computer Vision and Pattern Recognition, pp.84-91, Seattle, WA, 1994.

Rowley, H., Baluja, S., and Kanade, T., "Rotation Invariant Neural Network-Based Face Detection", *Computer Vision and Pattern Recognition*, pp. 38-44, 1998.

Reuters News, "Computer Security Threat On Rise – U.S. Survey", March 7, 1999.

Sung, K. and Poggio, T., "Learning a Distribution-Based Face Model for Human Face Detection", Neural Networks for Signal Processing V, pp. 398-406, Cambridge, Mass. 1995.

Visionics, "Visionics FaceIt is First Face Recognition Software to be used in a CCTW Control Room Application", http://www.visionics.com/live/frameset.html (March 10. 1999).

Zhang, J., Yan, Y., and Lades, M., "Face Recognition: Eigenfaces, Elastic Matching, and Neural Nets", *Proc. IEEE*, vol.85, no.9, pp.1423-1435, 1997.