Line-Based Face Recognition under Varying Pose

Olivier de Vel and Stefan Aeberhard

Abstract—Much research in human face recognition involves fronto-parallel face images, constrained rotations in and out of the plane, and operates under strict imaging conditions such as controlled illumination and limited facial expressions. Face recognition using multiple views in the viewing sphere is a more difficult task since face rotations out of the imaging plane can introduce occlusion of facial structures. In this paper, we propose a novel image-based face recognition algorithm that uses a set of random rectilinear line segments of 2D face image views as the underlying image representation, together with a nearest-neighbor classifier as the line matching scheme. The combination of 1D line segments exploits the inherent coherence in one or more 2D face image views in the viewing sphere. The algorithm achieves high generalization recognition rates for rotations both in and out of the plane, is robust to scaling, and is computationally efficient. Results show that the classification accuracy of the algorithm is superior compared with benchmark algorithms and is able to recognize test views in quasi-real-time.

Index Terms—Face recognition, line-based algorithm, classification accuracy, varying pose, real-time performance.

1 Introduction

Automated face recognition (AFR) has attracted much interest over the past few years. Such interest has been motivated by the growth in applications in many areas, including face identification in law enforcement and forensics, user authentication in building access or automatic transaction machines, indexing of, and searching for, faces in video databases, intelligent user interfaces, etc. AFR generally consists of different components, namely *face detection*, to determine the position and size of a human face in an image (see, for example, Sung and Poggio [18]), *face recognition* to compare an input face against models of faces that are stored in a database of known faces and indicating if a match is found, and *face verification* for the purpose of authentication and/or identification. In this paper, we study the face recognition and assume that the face location in the image is known.

Unfortunately, face recognition is difficult for a variety of reasons. First, different faces may appear very similar, thereby necessitating an exacting discriminant task. Second, different views of the same face may appear quite different due to imaging constraints, such as changes in illumination and variability in facial expressions, and due to the presence of accessories, such as glasses, beards, etc. Finally, when the face undergoes rotations out of the imaging plane, a large amount of detailed facial structure may be occluded. Therefore, in many implementations of face recognition algorithms, images are taken in a constrained environment with controlled illumination, minimal occlusions of facial structures, uncluttered background, and so on.

The most popular approaches in the face recognition literature are mainly identified by their differences in the input representation. Two major input representations are used, namely the geometric, feature-based approach and the example- or image-based approach. The matching procedure of input and model faces used in the majority of the geometric or image-based approaches utilizes fairly standard distance metrics like the Euclidean distance and correlation.

The feature-based technique extracts and normalizes a vector of geometric descriptors of biometric facial components such as the eyebrow thickness, nose anchor points, chin shape, zygomatic breadth, etc. The vector is then compared with, or matched against, the stored model face vectors. This approach, however, requires the solution of the correspondence problem, that is, the facial vector components must refer to the facial features in the image. Also, model face generation can be time consuming, particularly for large face databases, and the complexity of geometrical descriptors can be restrictive. Model generation and matching with non-fronto-parallel poses and varying illumination are more complex and can only be achieved at the expense of increased computation times (for example, Brunelli [4]). The technique was pioneered by Kanade [10] and, more recently, by workers such as Brunelli and Poggio [5].

The motivation for the image-based approach is its inherent simplicity compared with the feature-based approach, owing to the fact that it does not use any detailed biometric knowledge of the human face. Image-based techniques include any variation in face appearance due to changes in pose and lighting by simply storing many different 2D views of the face. These techniques use either the pixel-based bidimensional array representation of the entire face image or a set of transformed (e.g., gradient filtered) images or template subimages of facial features as the image representation. An image-based metric, such as correlation, is then used to match the resulting image with the set of model images. Two popular methods are used in the context of image-based face recognition techniques, namely template-based and neural networks. In the template-based approach, the face is represented as a set of templates of the major facial features which are then matched with the prototypical model face templates (see, for example, Baron [3]). Extensions to this technique include low dimensional coding to simplify the template representation and improve the performance of the template matching process (see, for example, the "eigenfaces" of Turk and Pentland [19]) or wavelets, stochastic modeling with Hidden Markov Models (HMMs) ([17]), and elastic face transforms to model the deformation of the face under a rotation in depth ([22]). Neural network-based image techniques use an input image representation that is the gray-level pixel-based image or transformed image which is used as an input to one of a variety of neural network architectures, including multilayer, radial basis functions and auto-associative networks (see, for example, Edelman et al. [9]).

Although geometric or image-based approaches are conceptually well-suited to face recognition, many of the techniques developed to date have been demonstrated on small, simplistic face databases with strict imaging constraints, requiring, in many cases, large processing times for training and/or recognition. In this paper, we propose a line-based face recognition technique under varying pose that is computationally efficient, has good recognition rates, handles face rotations both in and out of the imaging plane, and is robust to variations in scale. The image representation scheme used is a set of random one-dimensional rectilinear line segments of the gray-level face image and the matching scheme is an efficient nearest-neighbor classifier. The motivation for our line-based face recognition scheme is that, even though each line segment only predicts a correct face marginally better than random, the *combination* of line segments from a unique face image leads to a high probability of correct face classification. This argument follows the same line of reasoning as employed for image-based face recognition when compared with the geometric, feature-based approach. That is, we extend the idea that 3D object

- *O. de Vel is with the Information Technology Division, Defence Science and Technology Organisation, PO Box 1500, Salisbury 5108, Australia. E-mail: olivier.devel@dsto.defence.gov.au.*
- *S. Aeberhard is with the Department of Computer Science, James Cook University, Townsville, QLD 4811, Australia..*

Manuscript received 22 Apr. 1999; revised 14 July 1999.
Recommended for acceptance by D. Kriegman.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 107869.

recognition can effectively be undertaken using multiple redundant 2D views of an object to the idea that a combination of a set of randomly placed 1D line segments in a 2D object view exploits the coherence in that view. Therefore, the simpler rectilinear line segment primitive forms the basis for our image representation. We shall demonstrate that the use of simpler 1D line segments as the image representation model provides for better recognition rates compared with other face recognition techniques and executes in quasi-real-time. Related methods have been proposed by different authors (see, for example, [23], [11]). These methods have, however, used the length of a set of line segments (or "chords" in the authors' terminology) as a means of measuring the dissimilarity between shapes. In our method, we use the actual intensity values of line segments as the information contained in a given face. We present an overview of the performance of current face recognition systems in Section 2. We then describe our face recognition algorithm in Section 3 and present the face recognition experiments with results and comparisons with other benchmark algorithms in Sections 4 and 5.2, respectively. Finally, we conclude and give future work in Section 6.

2 Performance of Current Face Recognition Systems

We review the comparative performance of some face recognition systems that use the geometric or image-based approaches. A few authors report comparisons of two or more systems. In a later section, we will make a reference to the performance of some of these systems when we provide results for our algorithm (see Section 5.2). Achermann and Bunke [1] compare the eigenface classifier, a classifier based on hidden Markov models (HMM), and a profile classifier on a 30-person database with moderate pose variation among the 10 views per person. The eigenface classifier performed best (94.7 percent), followed by the HMM classifier (90.0 percent) and the profile based classifier (85.0 percent). Ranganath and Arun [15] compared radial basis functions and a nearest-neighbor classifier, using 1) an eigenface-based and 2) a wavelet-based image representation They observed that the radial basis function classifier performed better than the nearest-neighbor classifier and that the eigenface representation offered somewhat better discrimination than did the wavelet-based representation. Brunelli and Poggio [5] compared a feature based approach with a template based approach on a 47-person database of frontal views. The template-based technique achieved 100 percent correct classification, while the feature-based method achieved 90 percent, but was faster. Zhang et al. [24] compared three face recognition methods: the eigenface approach, a two-network-based connectionist approach, and a flexible template approach. The two networks are an auto-associative network for feature extraction and a classification network for recognition. With the template approach, Gabor filters were used to preprocess the image and elastic matching with an energy function was used for recognition. The tests were performed on four individual databases and on the combined set of 113 persons. The eigenface classifier was found to perform well for the individual databases where illumination conditions are constant, but performed badly (66 percent) on the combined database because of different light conditions among the databases. The flexible template approach performed well on all data, including the combined database (93 percent). The two neural networks did not perform well. A recent survey on face recognition in general was compiled by Chellappa et al. [6]; Valentin et al. [21] surveyed the use of connectionist models in particular.

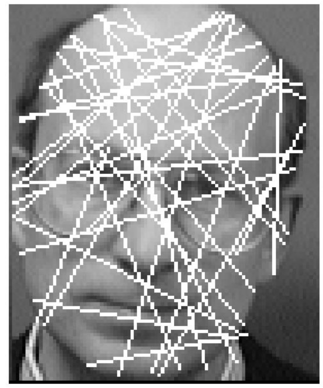

Fig. 1. Example set of random lines in a face view.

3 An Efficient Face Recognition Algorithm

Here, we briefly outline the face recognition algorithm which is based on a more general object recognition algorithm given in [7], [8]. We are interested in classifying K faces, F_k $(k = 1, \dots, K)$, given $V_k \in \mathbb{Z}_{\oplus}$ 2D image views of each unique face F_k, obtained by regular sampling in the viewing sphere. The aim is to recognize one of the K faces from one or more test image views.

A face image $\Im$ is modeled as a regular lattice of $w \times h$ pixels, with each pixel P having a depth equal to $\Im_P$ image planes. We first classify the pixels in $\Im$ into two classes, $C_{p,1}$ and $C_{p,2}$. Class $C_{p,1}$ consists of the background pixels in the face image $\Im$ and class $C_{p,2}$ consists of all those pixels that represent a face in $\Im$ such that $C_{p,1} \cap C_{p,2} = \phi$. We are interested in those pixels in $C_{p,2}$ with neighbors in $C_{p,1}$ and call the set of those face boundary pixels β.

Consider l pixel values extracted along a straight line or "chord" between two points in the image, comprising of $l \times \Im_P$ bits of data. The number of line pixels (or *line dimensionality*) is small enough for efficient classification but, of course, may not capture the information necessary for correct classification. However, with some reduced probability (larger than random), the line predicts the correct face class. The algorithm we propose is based on the observation that the classification of many such lines from a face image $\Im$ leads to an overall probability of correct classification (PCC) which approaches 1. This observation serves as the main motivation for the algorithm. An example set of face lines is shown in Fig. 1.

For any two points $B_1 \in \beta$ and $B_2 \in \beta$ in an image view V_k such that the Euclidean distance between B_1 and B_2 is greater than a minimum D_{min}, let $\mathbf{L}(B_1, B_2) \equiv (L^{(1)}, L^{(2)}, \dots L^{(l)})$ be a vector of length l, where l is the number of equi-spaced connected intensity values $L^{(q)} = P(\mathbf{L})_q$ (where $q = 1, 2, \dots, l$) along the image rectilinear segment from B_1 to B_2. We note that, in our algorithm, the points B_1 and B_2 need not necessarily belong to the set of face boundary pixels β. Indeed, rectilinear line segments may span any two pixels that are inside the face boundary, i.e., $B_1, B_2 \subseteq C_{p,2}$. The relative performance of the algorithm will depend on the coverage of the face by the set of line segments and maximum performance will generally be achieved when $B_1, B_2 \in \beta$. In this paper, we limit our discussion to the case $B_1, B_2 \in \beta$.

The line segment length l is a constant parameter determined a priori; larger values of l result in better classification rates at the expense of increased processing times. All lines are scaled to the value l by pixel interpolation. We call $\mathbf{L}(B_1, B_2)$ a lattice line, denoted by $\mathbf{L}$. The exact endpoints of L need not lie on a corner of the boundary pixels B_1 and B_2.

For each face class F_k in the training set of V_k image views, we randomly generate $N_k = V_k \times N_{V_k}$ lattice lines (N_{V_k} lines per image view per face class), $\mathbf{L}_{i,k} \equiv (L^{(1)}_{i,k}, L^{(2)}_{i,k}, \dots L^{(l)}_{i,k})$ for $i = 1, 2 \dots, N_k$.

Fig. 2. Example image views for one subject ("`s40`") from the ORL face database.

There are $M = \sum_{k=1}^{K} N_k$ such lattice lines for K face classes. The set of lattice lines for all K face classes is given by:

$$\Psi = \bigcup_{k=1}^{K} \bigcup_{i=1}^{N_k} \mathbf{L}_{i,k}.$$

We define the distance $D(\mathbf{L}_{r,s}, \mathbf{L}_{m,n})$ between two lattice lines $\mathbf{L}_{r,s}$ and $\mathbf{L}_{m,n}$ as

$$D(\mathbf{L}_{r,s}, \mathbf{L}_{m,n}) = \sum_{q=1}^{l} ((L_{r,s}^{(q)} - (L_{m,n}^{(q)} + \Delta))^2),$$

for $r, m = 1, 2 \ldots, N_k$ and $s, n = 1, 2 \ldots, K$, where $\Delta = \mu(\mathbf{L}_{r,s}) - \mu(\mathbf{L}_{m,n})$ and $\mu(\mathbf{L}_{r,s}) = \sum_l \mathbf{L}_{r,s}/l$. The value of Δ has the effect of shifting the two lines towards the same average value, making the distance measure invariant to illumination intensity.

Consider now a set of test lines sampled from one or more face views in the viewing sphere (for the same face subject). Given an unseen test lattice line $\mathbf{L}_j$ where, generally, $\mathbf{L}_j \notin \Psi$, we define $\mathbf{L}_{j,*}$ such that $D(\mathbf{L}_j, \mathbf{L}_{j,*})$ is a minimum, where $\mathbf{L}_{j,*} \in \Psi$. The Nearest-Neighbor Classifier (NNC) maps $\mathbf{L}_j$ to the class F_k to which $\mathbf{L}_{j,*}$ belongs. That is, $\mathrm{NNC}(\mathbf{L}_j) = F_k$. We write D_j for $D(\mathbf{L}_j, \mathbf{L}_{j,*})$. We note in passing that any classifier can be used to map $\mathbf{L}_j$ to the class F_k to which $\mathbf{L}_{j,*}$ belongs. We choose the nearest-neighbor classifier since it has a good performance over a range of problem domains [2].

We assume that there are N test lines $\mathbf{L}_j$ for a given face, where $j = 1, 2, \ldots N$ and, for each line, we have obtained an $\mathbf{L}_{j,*}$ and a D_j. Let $D_{\max} = k_1 \times \max_{1 \le j \le N}\{D_j\}$ (for some value k_1, where $0 < k_1 \le 1$) and $D_{\min} = \min_{1 \le j \le N}\{D_j\}$. We define the cumulative l_1-norm error statistic for line $\mathbf{L}_j$, $\mathrm{err}_j = (\sum_{q=1}^{l}(|L_{j,*}^{(q+1)} - L_{j,*}^{(q)}|)/(l-1)$ for $q = 1, 2, \ldots l-1$ and the maximum cumulative error statistic, $\mathrm{err}_{\max} = \max_{1 \le i \le N}\{\mathrm{err}_i\}$.

We define the measure of confidence that $\mathrm{NNC}(\mathbf{L}_j)$ is correct, conf_j:

$$\mathrm{conf}_j = \begin{cases} 0, & \text{if } D_j > D_{\max} \\ \left[\frac{(D_{\max} - D_j)}{(D_{\max} - D_{\min})} w_1\right]^{p_1} \left[(\frac{\mathrm{err}_j}{\mathrm{err}_{\max}}) w_2\right]^{p_2} & \text{otherwise.} \end{cases}$$

where p_1, p_2, w_1, and $w_2 \in \mathbb{R}_{\oplus}$. The variables p_1 and p_2 control the shape of the confidence function, whereas w_1 and w_2 are the weight magnitudes of the distance and cumulative error statistic components, respectively.

We now state the face recognition algorithm.

The Line-Based Face Recognition Algorithm:

To classify a face F_t for which we know its boundary pixel set β, we randomly select N lattice lines $\mathbf{L}_j$, $j = 1, 2, \ldots N$. For each face class $F_k = 1, 2, \ldots K$, define TC_k as $\mathrm{TC}_k = \sum_{j=1}^{N} \mathrm{conf}_j$, such that $\mathrm{NNC}(\mathbf{L}_j) = F_k$. We assign F_t to class F_g such that TC_g is maximum. That is,

$$\begin{array}{ll} \text{if} & \mathrm{TC}_g = \max_{1 \le k \le K}\{\mathrm{TC}_k\} \\ \text{then} & F_g \leftarrow F_t \quad \text{for} \quad F_g = 1, 2, \ldots K \end{array}$$

Because F_t is assigned to class F_g based on the combination of many assignments of individual lines, we may assess the likelihood that our decision is correct by the agreement within the line assignments. Specifically, we define the *confidence measure factor* as the ratio

$$CMF = [\mathrm{TC}_g - \mathrm{TC}_j^{(2)}]/\mathrm{TC}_j^{(2)},$$

where $\mathrm{TC}_j^{(2)}$ is the second largest compounded confidence measure that a class obtained. As our decision is based on the maximum score, the associated confidence CMF is proportional to the difference with the second largest score. The denominator normalizes CMF for different numbers of testing lines.

It is a considerable advantage if a classifier were to supply a confidence measure factor with its decision as the user is then given information about which assignments are more likely to be wrong so that extra caution can be exercised in those cases. Our

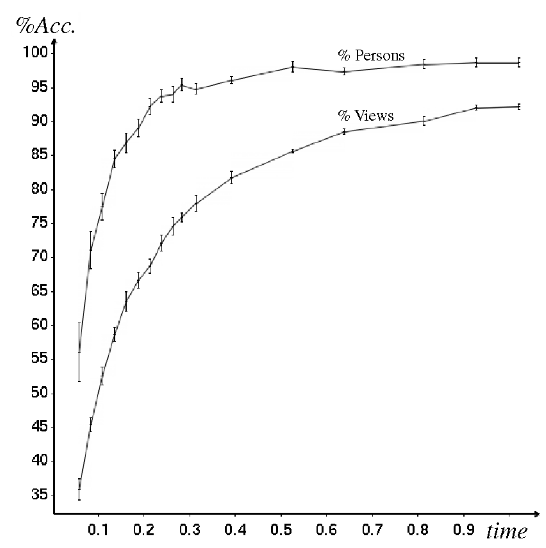

Fig. 3. PCC versus time for 17 values for the number of testing lines $N = 20, 30, \ldots, 110, 120, 150, 200, \ldots, 400$. Here, the number of training lines $N_k = 200$, the line dimensionality $l = 32$, and the confidence threshold $CMF_{min} = 0$.

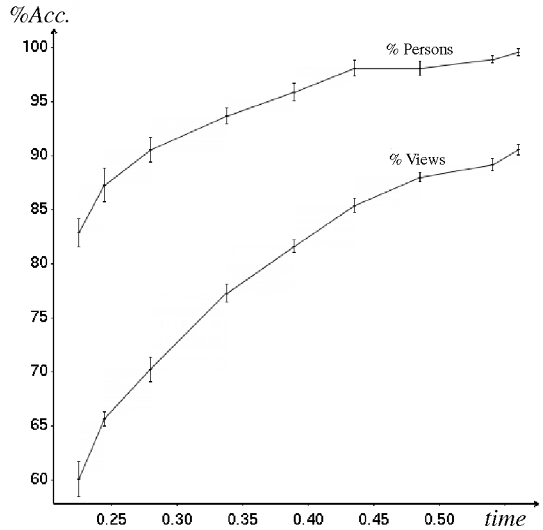

Fig. 4. PCC versus time for nine values for the number of training lines $N_k = 50, 70, 100, 150, 200, 250, 300, 250, 400$ with $N = 150$, $l = 32$, and $CMF_{min} = 0$.

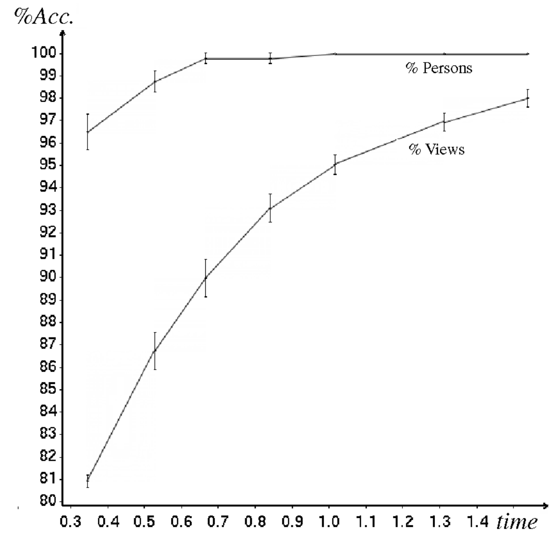

Fig. 6. PCC versus time for seven values of the minimum confidence measure $CMF_{min} = 0, 0.05, 0.10, 0.15, 0.20, 0.30, 0.40$ with $N_k = 300$, $N = 100$, and $l = 32$.

implementation makes use of the confidence measure factor by means of several decision stages. First, the number of testing lines is kept small, an initial decision is arrived at quickly, and the confidence measure factor is evaluated. Second, if the confidence measure factor is smaller than twice the minimum confidence measure factor threshold CMF_{min}, then the number of testing lines is doubled and a second decision is made at the cost of extra time. Finally, if the second confidence measure factor is smaller than CMF_{min}, the number of test lines is doubled again one last time. Thus, by specifying a larger value for CMF_{min}, the number of test lines will be increased and, hopefully, improve the rate of correct classification (see the section on experimental results for confirmation of this). However, by increasing the number of test lines, there will be a commensurate increase in the time required for classification. Therefore, depending on the application task at hand, the user can choose whether to seek a high classification rate at the expense of larger classification times or to achieve a lower classification rate with an accompanying reduction in classification times.

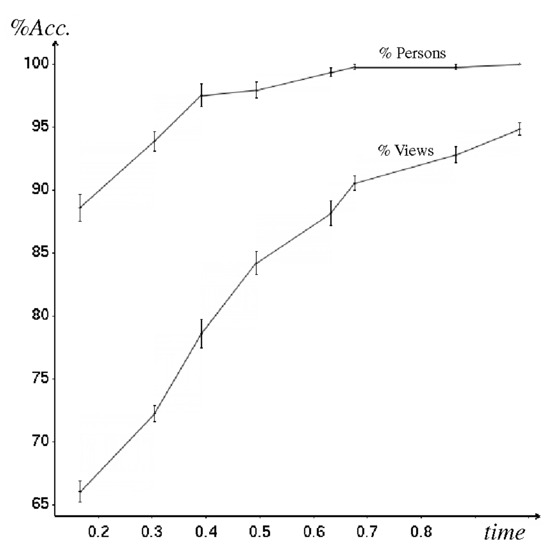

Fig. 5. PCC versus time for eight values of the minimum confidence measure $CMF_{min} = 0, 0.05, 0.10 \ldots, 0.30, 0.40$ with $N_k = 300$, $N = 50$, and $l = 32$.

The above algorithm is surprisingly simple and, as we shall demonstrate, is particularly effective in obtaining a high recognition rate performance, as well as achieving low computation times. Moreover, the algorithm has some inherent advantages. First, due to the randomized sampling of the image, the algorithm is robust to rotations of the face in the plane. Second, we reason that multiple views are even better suited to our 1D line-based algorithm and are better able to handle head rotations out of the plane than 2D view-based algorithms. Third, since the lines run from one face-boundary to another and have fixed dimensionality, the algorithm is also scale-invariant. Fourth, the choice of distance measure ensures that it is tolerant to changes in illumination intensity. Finally, because all lines are sampled from the entire head section of the image, the algorithm is also robust to changes in facial expressions and to the presence or absence of glasses or other accessories. Unfortunately, the current algorithm is not robust to changes in illumination direction, such as found in outdoor settings, or successful in cluttered scenes (such as in a video sequence).

4 Face Databases and Experimental Methodology

In order to evaluate the performance of the algorithm, we used two face databases, namely the University of Bern (UB) [20] and the Olivetti & Oracle Research Laboratory (ORL) [14] face databases. The UB face database contains 10 frontal face images for each of 30 persons acquired under controlled lighting conditions. The database is characterized by small changes in facial expressions and intermediate changes (± 30 degrees out of the plane) in head pose, with two images for each of the poses right, left, up, down, and straight. The ORL face database consists of 10 frontal face images for each of 40 persons (four female and 36 male subjects). There are intermediate changes in facial expression and unstructured intermediate changes (± 20 degrees) in head pose. Some people wear glasses in some images and the images were taken under different lighting conditions. Fig. 2 shows a typical set of 10 image views for one person in the ORL face database. In our experiments, we combined the two databases to form one larger one containing 700 face images of 70 people. The data set was used to assess the classification rate of our algorithm by cross-validation, using five images per person for training and the other five for testing.

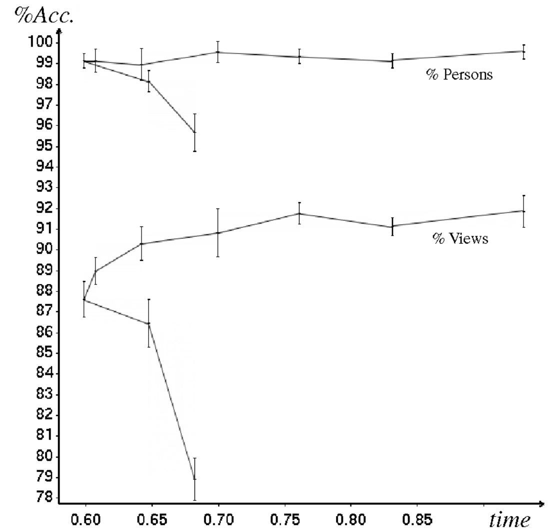

Fig. 7. PCC versus time for nine values of the line dimensionality $l = 8, 12, 16, 24, 32, 40, 48, 56, 64$ with $N_k = 300$, $N = 100$, and $CMF_{min} = 0.10$.

5 Experimental Results

Prior to evaluating the overall performance of the algorithm for the combined face database, various parameters were optimized. These included the number of training lines (N_k), the number of test lines (N), the line dimensionality (l), and the decision confidence measure factor (CMF).

5.1 Evaluation of Parameters

In order to investigate the effect of various parameter settings on classification time and correctness, we ran four experiments, varying one parameter at a time. In each experiment, the parameters were resampled over the combined face database. For example, for evaluating the probability of correct classification (PCC) for different numbers of test lines, each set of test lines was obtained by resampling the combined face database.

The results are presented in Figs. 3, 4, 5, 6, and 7. The vertical axis shows the classification accuracy (probability of correct classification, PCC) as a percentage, the horizontal axis represents the computation time in seconds, per view. Each figure shows two lines, the upper line indicates the percentage of correctly classified persons based on the *majority* of the view classifications (i.e., a majority vote of the classification results of the five views in the viewing sphere), the lower line shows the percentage of correctly classified *individual* views. Each point on both lines corresponds to one of the parameter settings (see the figure caption for their values). Each graph also shows the standard deviations obtained for the 10 repetitions undertaken for each experiment. For each repetition, the training and test lines were resampled and the PCC recorded.

In the first experiment, the number of test lines N was varied from 20 to 400 (17 values in total). The number of training lines, N_k, was set to 200, the line dimensionality was set to $l = 32$, and the minimum confidence factor was set to zero. This corresponds to a ratio in the number of training pixels to image size equal to 0.078, equivalent to an effective dimensionality reduction for model storage by a factor equal to 12.8. Fig. 3 shows the results. As expected, both the computation time and the classification accuracy increase almost monotonically with the number of test lines. The increase in time is approximately linear, while the accuracy first increases rapidly and then levels out. A distinctive "knee" in the curve of the percentage of correctly classified persons occurs at about $N = 100$ test lines with a classification accuracy of approximately 95 percent.

In the second experiment, the number of training lines was varied from 50 to 400 (nine values in total). The number of testing lines was fixed at 150, the line dimensionality was set to 32, and the minimum confidence parameter was set to 0. Fig. 4 shows the results.

Again, as expected, both the time and the classification accuracy increase with increased number of training lines. However, the shapes of the curves are flatter than those for the test lines (Fig. 3), i.e., there is no distinctive "knee" where the curve flattens out. This is as expected, as increasing the number of training lines increases the PCC of a test line, while increasing the number of test lines only decreases the variance in the procedure that forms a decision from the classifications of all test lines. Once that variance is reduced significantly, the inaccuracy due to a finite number of training views dominates and the curve flattens out.

In the third experiment, the minimum confidence measure factor value was varied from 0.0 to 0.4 (eight values in total). The number of training lines was set to 300, the initial number of testing lines set to 50, and the line dimensionality set to 32. The results are shown in Fig. 5. The experiment was repeated with the number of initial testing lines set to 100 (see Fig. 6). Both Fig. 5 and Fig. 6 look very similar, with the larger values of the minimum confidence factor resulting in larger computation times. However, an improved accuracy for the case when the initial number of test lines is doubled is observed. On the other hand, both figures converge to a similar classification accuracy. For example, for a time equal to 0.7 sec, both cases achieve near 100 percent accuracy for the people and greater than 90 percent accuracy for the views. This means we can decrease the number of initial testing lines without ill-effect so long we increase the minimum confidence measure factor accordingly and vice versa.

In the last experiment, the line dimensionality l was varied from 8 to 64 (nine values in total). The number of training lines was set

TABLE 1
Face and View Recognition Rates for Random and Selected Sampling of Training Poses for the Combined Face-Database

Training Procedure	*Classification Accuracy (%)* *Face Recognition*	*View Recognition*	*Training Time per view (sec)*	*Test Time per view (sec)*
Random	98.0 (max=100.0, min=94.3)	80.8 (max=87.1, min=76.3)	0.8 max	0.79 max
Selected	100.0	88.6 (max=89.3, min=87.4)	0.8 max	0.69 max
Selected	100.0	99.8 (max=100.0, min = 99.7)	1.6 max	5.0 max

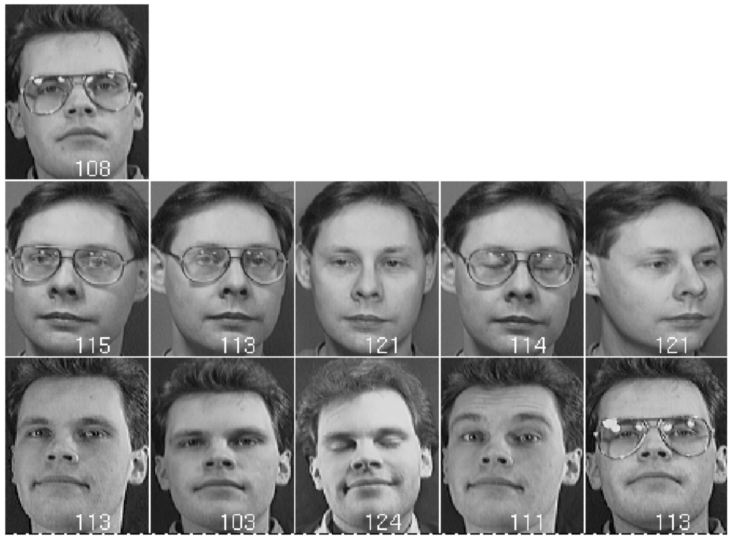

Fig. 8. Example misclassification of a test view for five training views of two people from the ORL database (see text for explanation).

Fig. 9. Example misclassification of a test view for five training views of two people from the UB database (see text for explanation).

to 300, the initial number of testing lines was set to 100 and the minimum confidence value was set to 0.1. The results are shown in Fig. 7. The classification accuracy first increases rapidly, then stabilizes, with the change occurring between $l = 16$ and 24. The timings first decrease, then increase. This is because the confidence value is not set to zero and small dimensionalities triggered frequent increases in the number of test lines. However, it is perhaps surprising that, for $l = 16$, the classification rate for persons is already near 100 percent.

The results indicate that, despite very low PCC of individual lines (we observed values around 0.1), combining a large number of such classification results in an overall PCC which is much higher. In fact, the graphs indicate that, as the number of training lines goes to infinity, the PCC approaches 1.0. In the following section, we present results for the combined face database using optimal parameter values. The number of training and test lines per face view were chosen to be equal to 200 and 80, respectively, a dimensionality of 32 and minimum confidence measure factor equal to 0.4 were selected.

5.2 Face and View Recognition Performance Results

We ran two sets of experiments using the optimal parameter values. In the first set, we selected the training set by inspection in order to provide a good cover of the varying head positions and facial expressions. In the second set of experiments, we randomly selected the image views, repeating the process three times. We expect to obtain an inferior recognition rate for the random sampling of the training set compared with the selected sampling. Most real-world applications would allow selecting good training views, as with our first approach. However, in some applications such as video sequences, random sampling is more realistic. We also evaluated the algorithm for both face recognition and view recognition. That is, in the former case, the algorithm is presented with all the test image views for a given face, whereas, in the latter case, the algorithm is presented with just a single test view (as would be found in, for example, an access control situation). Again, we expect lower recognition performance results for the view recognition as compared with the face recognition.

Table 1 shows the results obtained for both the regular (selected) and random set of training and test views and, for both face and view recognition, for the combined face database. Also

TABLE 2
Comparative Recognition Rates for ORL and Bern Face Databases

Authors	*Classifier Method*	*ORL*	*Bern*
Samaria [16]	HMM	95.0	–
Zhang *et al* [24]	Eigenface	80.0	87.0
Zhang *et al* [24]	Elastic Matching	80.0	93.0
Lin *et al* [13]	Neural network	96.0	–
Lawrence *et al* [12]	Neural network	96.2	–
Lawrence *et al* [12]	Eigenface	89.5	–
Achermann *et al* [1]	HMM	–	90.0
Achermann *et al* [1]	Eigenface	–	94.7
Achermann *et al* [1]	Combination	–	99.7
de Vel and Aeberhard	Line segments	99.7 min. 100.0 max.	

shown are the maximum and minimum recognition rates obtained in all experiments.

With selective (nonrandom) view cover, we found that 100 percent of people are correctly classified if approximately 0.7 sec or more is spent per view to form the decision. Further experiments have shown that we can reduce the test times at the expense of a reduced recognition rate—approximately 85 percent of people were correctly classified if a maximum of 0.1 sec was allowed. On the other hand, a significant improvement in the view recognition rate can be achieved if a higher test time is allowed—nearly 100 percent of all views are correctly classified if up to 5 sec is used for testing. For random sampling, we observe slightly reduced recognition rates and fractionally longer test times (per view). The results confirm that a relatively small number of views is sufficient for good view-based classification performance if the test views are sufficiently covered by the training views.

Figs. 8 and 9 show examples of a face view instance that was misclassified by the algorithm in one of the experiments. The second and third rows of faces are training examples of two people (one person for the second row and one for the third row). The first row shows the test image view corresponding to the person in the third row that was misclassified as the person in the second row. As can be seen, it is not always straightforward to discriminate between the two pople in some poses.

We also tested our method on the individual databases. Our method achieved 100 percent correct recognition of views on the ORL database using an average of 3.9 sec per view for testing and 100 percent recognition of views on the Bern database using, on average, 1.5 sec per view. More computation time is spent classifying views from the ORL database because it contains many more views which are difficult to recognize. The parameter settings for these results were: 500 training lines, 120 initial testing lines, the line dimensionality was 24, and the minimum confidence factor was set to 0.5. For comparison, we include some benchmark results obtained by other workers on the same face databases (see Section2). Samaria [16] used the HMM implementation, Zhang et al. [24] implemented both the eigenface and elastic matching algorithms, Lin et al. [13] use intensity and edge information with a neural network, Lawrence et al. [12] tested local image sampling with two neural networks as well as the eigenface classifier, and Achermann and Bunke [1] used a combination of eigenface, HMM and profile classifiers. Their results are summarized in Table 2. Some of the benchmark results obtained from the author references did not clearly state how each test view was chosen (e.g., whether or not the views were selected randomly or uniformly from the set of poses). We state our results for the line-based algorithm for the combined face database in the last row of the table. We include results for both the minimum and maximum view recognition rates obtained for a maximum computation time equal to 5 sec per test view.

As can be observed, our worst-case result on the combined databases is no worse than the best result of any of the benchmarks on the individual databases—even when compared with combined classifiers used in Achermann and Bunke's experiments. Our best-case result gives a zero error recognition rate, a result not achieved by any of the other classifier methods. Furthermore, none of the benchmarks give results for the execution times (which, we suspect, are larger than our results for comparable classification accuracies). Our test time results are quite adequate for real-time applications such as security access and video sequence tracking.

6 Conclusions

We have described a computationally efficient view-based face recognition algorithm using random rectilinear line segments of face images. The algorithm is robust to rotations in, and out of, the plane, robust to variations in scale, and is robust to changes in illumination intensity and to changes in facial expressions. Experiments have demonstrated that the algorithm is superior compared with available benchmark algorithms and is able to recognize test views in quasi real-time. The relatively good performance of the algorithm is due to the fact that a combination of 1D line segments effectively exploits the inherent coherence in a 2D face image view. Even though each line segment only predicts a correct face marginally better than random, the combination of line segments leads to a high probability of correct face classification.

The main drawback of our technique lies in the assumption that the face detection has been undertaken prior to the application of the line-based algorithm and that the face boundaries are available. If the boundaries are largely occluded or indistinguishable from the background, then the performance of the current algorithm will be reduced. We are currently investigating modifications to the algorithm that will account for the absence of face boundaries.

References

[1] B. Achermann and H. Bunke, "Combination of Face Classifiers for Person Identification," Technical Report IAM–96–002, Institut für Informatik und Angewandte Mathematik, Universität Bern, 1996.

[2] S. Aeberhard, D. Coomans, and O. de Vel, "Comparative Analysis of Pattern Classifiers in a High Dimensional Setting," *Pattern Recognition,* vol. 24, pp. 1,065-1,077, 1994.

[3] R. Baron, "Mechanisms of Human Facial Recognition," *Int'l J. Man Machine Studies,* vol. 15, pp. 137-178, 1981.

[4] R. Brunelli, "Estimation of Pose and Illuminant Direction for Face Processing," *Image and Vision Computing,* vol. 15, pp. 741-748, 1997.

[5] R. Brunelli and T. Poggio, "Face Recognition: Features versus Templates," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, pp. 1,042-1,052, 1993.

[6] R. Chellappa, C. Wilson, and S. Sirohey, "Human and Maching Recognition of Faces: A Survey," *Proc. IEEE,* vol. 83, pp. 705-740, 1995.

[7] O. de Vel and S. Aeberhard, "View-Based Object Recognition Using Image Lines," Technical Report 97–09, Dept. of Computer Science, James Cook Univ., Australia, 1997.

[8] O. de Vel and S. Aeberhard, "Object Recognition Using Random Image-Lines," *Image and Vision Computing,* 1999 (to appear).

[9] S. Edelman, D. Reisfield, and Y. Yeshurun, "Learning to Recognize Faces from Examples," *Lecture Notes in Computer Science,* vol. 588, 787-791, 1992.

[10] T. Kanade, "Picture Processing by Computer Complex and Recognition of Human Faces," technical report, Dept of Information Sciences, Kyoto Univ., 1973.

[11] D. Lavine, B. Lambird, and L. Kamal, "Recognition of Spatial Point Patterns," *Proc. IEEE Conf. Pattern Recognition and Image Processing,* pp. 49-53, 1981.

[12] S. Lawrence, C. Giles, A. Tsoi, and A. Back, "Face Recognition: A Convolutional Neural Network Approach," *IEEE Trans. Neural Networks,* vol. 8, pp. 98-113, 1997.

[13] S. Lin, S. Kung, and L. Lin, "Face Recognition/Detection by Probabilistic Decision-Based Neural Network," *IEEE Trans. Neural Networks,* vol. 8, pp. 114-132, 1997.

[14] Olivetti & Oracle Research Laboratory, *The Olivetti & Oracle Research Laboratory Face Database of Faces,* http://www.cam-orl.co.uk/facedatabase.html.

[15] S. Ranganath and K. Arun, "Face Recognition Using Transform Features and Neural Networks," *Pattern Recognition,* vol. 30, pp. 1,615-1,622, 1997.

[16] F. Samaria, "Face Recognition Using Hidden Markov Models," PhD thesis, Univ. of Cambridge, Cambridge, U.K., 1994.

[17] F. Samaria and A. Harter, "Parametrisation of a Stochastic Model for Human Face Identification," *Proc. Second IEEE Workshop Applications of Computer Vision,* Sarasota, Fla., Dec. 1994.

[18] K. Sung and T. Poggio, "Example-Based Learning for View-Based Human Face Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, pp. 39-51, 1998.

[19] M. Turk and A. Pentland, "Eigenfaces for Recognition," *J. Cognitive Neuroscience,* vol. 3, pp. 71-86, 1991.

[20] University of Bern, Bern, Switzerland, *University of Bern Face Database,* ftp://iamftp.unibe.ch/pub/Images/FaceImages/.

[21] D. Valentin, H. Abdi, A. O'Tolle, and G. Cottrell, "Connectionist Models of Face Processing: A Survey," *Pattern Recognition,* vol. 27, pp. 1,209-1,230, 1994.

[22] L. Wiskott, J. Fellous, N. Kruger, and C. von der Malsburg, "Face Recognition by Elastic Bunch Graph Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, pp. 775-?, 1997.

[23] Z. You and A. Jain, "Performance Evaluation of Shape Matching via Chord Length Distribution," *Computer Vision, Graphics, and Image Processing,* vol. 28, pp. 185-198, 1984.

[24] J. Zhang, Y. Yan, and M. Lades, "Face Recognition: Eigenface, Elastic Matching, and Neural Nets," *Proc. IEEE,* vol. 85, pp. 1,423-1,435, 1997.