Face Recognition through Geometrical Features

R. Brunelli[1] *and T. Poggio*[2]

[1] Istituto per la Ricerca Scientifica e Tecnologica
I-38050 Povo, Trento, ITALY
[2] Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139, USA

Abstract. Several different techniques have been proposed for computer recognition of human faces. This paper presents the first results of an ongoing project to compare several recognition strategies on a common database.

A set of algorithms has been developed to assess the feasibility of recognition using a vector of geometrical features, such as nose width and length, mouth position and chin shape. The performance of a Nearest Neighbor classifier, with a suitably defined metric, is reported as a function of the number of classes to be discriminated (people to be recognized) and of the number of examples per class. Finally, performance of classification with rejection is investigated.

1 Introduction

The problem of face recognition, one of the most remarkable abilities of human vision, was considered in the early stages of computer vision and is now undergoing a revival. Different specific techniques were proposed or reproposed recently. Among those, one may cite neural nets [9], elastic template matching [5, 23], Karhunen-Loewe expansion [20], algebraic moments [11] and isodensity lines [16]. Typically, the relation of these techniques with standard approaches and their relative performance has not been characterized well or at all. Even absolute performance has been rarely measured with statistical significance on meaningful databases. Psycological studies of human face recognition suggest that virtually every type of available information is used [22]. Broadly speaking we can distinguish two ways [19] to get a one-to-one correspondence between the stimulus (face to be recognized) and the stored representation (face in the database):

Geometric, feature-based matching. A face can be recognized even when the details of the individual features (such as eyes, nose and mouth) are no longer resolved. The remaining information is, in a sense, purely geometrical and represents what is left at a very coarse resolution. The idea is to extract relative position and other parameters of distinctive features such as eyes, mouth, nose and chin [10, 14, 8, 2, 13]. This was the first approach towards an automated recognition of faces [13].

Template matching. In the simplest version of template matching, visual patterns, represented as bidimensional arrays of intensity values, are compared using a suitable metric (typically the euclidean distance) and a single template, representing the whole face, is used[3].

[3] There are of course several, more sophisticated ways of performing template matching. For instance, the array of grey levels may be suitably preprocessed before matching. Several full templates per each face may be used to account for the recognition from different viewpoints. Still another important variation is to use, even for a single viewpoint, multiple templates. A

In order to investigate the first of the above mentioned approaches we have developed a set of algorithms and tested it on a data base of 47 different people.

2 Experimental setup

The database we used for the comparison of the different strategies is composed of 188 images, four for each of 47 people. Of the four pictures available, the first two were taken in the same session (a time interval of a few minutes) while the other pictures were taken at intervals of some weeks (2 to 4). The pictures were acquired with a CCD camera at a resolution of 512 × 512 pixels as frontal views. The subjects were asked to look into the camera but no particular efforts were made to ensure perfectly frontal images. The illumination was partially controlled: the same powerful light was used but the environment where the pictures were acquired was exposed to sun light through windows. The pictures were taken randomly during the day time. The distance of the subject from the camera was fixed only approximately, so that scale variations of as much as 30 percent were possible.

3 Geometric, feature-based matching

As we have mentioned already, the very fact that face recognition is possible even at coarse resolution, when the single facial features are hardly resolved in detail, implies that the overall geometrical configuration of the face features is sufficient for discrimination. The overall configuration can be described by a vector of numerical data representing the position and size of the main facial features: eyes and eyebrows, nose and mouth. This information can be supplemented by the shape of the face outline. As put forward by Kaya and Kobayashi [14] the set of features should satisfy the following requisites:

- estimation must be as easy as possible;
- dependency on light conditions must be as small as possible;
- dependency on small changes of face expression must be small;
- information contents must be as high as possible.

The first three requirements are satisfied by the set of features we have adopted, while their information contents is characterized by the experiments described later.

The first attempts at automatic recognition of faces by using a vector of geometrical features are probably due to Kanade [13] in 1973. Using a robust feature detector (built from simple modules used within a backtracking strategy) a set of 16 features was computed. Analysis of the inter and intra class variances revealed some of the parameters to be ineffective, yielding a vector of reduced dimensionality (13). Kanade's system achieved a peak performance of 75% correct identification on a database of 20 different people using two images per person, one for reference and one for testing.

The computer procedures we implemented are loosely based on Kanade's work and will be detailed in the next sections. The database used is however more meaningful (in the sense of being greater) both in the number of classes to be recognized, and in the number of instances of the same person to be recognized.

face is stored then as a set of distinct(ive) smaller templates [1]. A rather different approach is based on the technique of elastic templates [6, 5, 23]

3.1 Normalization

One of the most critical point when using a vector of geometrical features is that of proper scale normalization. The extracted features must be somehow normalized in order to be independent of position, scale and rotation of the face in the image plane. Translation dependency can be eliminated once the origin of coordinates is set to a point which can be detected with good accuracy in each image. The approach we have followed achieves scale and rotation invariance by setting the interocular distance and the direction of the eye-to-eye axis. We will describe the steps of the normalization procedure in some detail since they are themselves of some interest.

The first step in our technique resembles that of Baron [1] and is based on template matching by means of a normalized cross-correlation coefficient, defined by :

$$C_N(\mathbf{y}) = \frac{< I_T T > - < I_T >< T >}{\sigma(I_T)\sigma(T)} \tag{1}$$

where I_T be the patch of image I which must be matched to T, $<>$ the average operator, $I_T T$ represent the pixel-by-pixel product, and σ the standard deviation over the area being matched. This normalization rescales the template and image energy distribution so that their average and variances match.

The eyes of one of the authors (without eyebrows) were used as a template to locate eyes on the image to be normalized. To cope with scale variations, a set of 5 eyes templates was used, obtained by scaling the original one (the set of scales used is 0.7, 0.85, 1, 1.15, 1.3 to account for the expected scale variation). Eyes position was then determined looking for the maximum absolute value of the normalized correlation values (one for each of the templates). To make correlation more robust against illumination gradients, each image was preprocessed by dividing each pixel by the average intensity on a suitably large neighborhood.

It is well known that correlation is computationally expensive. Additionally, eyes of different people can be markedly different. These difficulties can be significantly reduced by using hierarchical correlation (as proposed by Burt in [7]). Gaussian pyramids of the preprocessed image and templates are built. Correlation is done starting from the lowest resolution level, progressively reducing the area of computation from level to level by keeping only a progressively smaller area.

Once the eyes have been detected, scale is pre-adjusted using the ratio of the scale of the best responding template to the reference template. The position of the left and right eye is then refined using the same technique (with a left and a right eye template). The resulting normalization proved to be good. The procedure is also able to absorb a limited rotation in the image plane (up to 15 degrees). Once the eyes have been indipendently located, rotation can be fixed by imposing the direction of the eye-to-eye axis, which we assumed to be horizontal in the natural reference frame. The resolution of the normalized pictures used for the computation of the geometrical features was of 55 pixels of interocular distance.

3.2 Feature Extraction

Face recognition, while difficult, presents interesting constraints which can be exploited in the recovery of facial features. An important set of constraints derives from the fact that almost every face has two eyes, one nose, one mouth with a very similar layout. While this may make the task of face classification more difficult, it can ease the task of feature extraction: average anthropometric measures can be used to focus the search of a

particular facial feature and to validate results obtained through simple image processing techniques [3, 4].

A very useful technique for the extraction of facial features is that of integral projections. Let $\mathcal{I}(x, y)$ be our image. The vertical integral projection of $\mathcal{I}(x, y)$ in the $[x_1, x_2] \times [y_1, y_2]$ domain is defined as:

$$V(x) = \sum_{y=y_1}^{y_2} \mathcal{I}(x, y) \tag{2}$$

The horizontal integral projection is similarly defined as:

$$H(y) = \sum_{x=x_1}^{x_2} \mathcal{I}(x, y) \tag{3}$$

This technique was succesfully used by Takeo Kanade in his pioneering work [13] on recognition of human faces. Projections can be extremely effective in determining the position of features provided the window on which they act is suitably located to avoid misleading interferences. In the original work of Kanade the projection analysis was performed on a binary picture obtained by applying a laplacian operator (a discretization of $\partial_{xx} I + \partial_{yy} I$) on the grey-level picture and by thresholding the result at a proper level. The use of a laplacian operator, however, does not provide information on edge (that is gradient) directions. We have chosen therefore to perform edge projection analysis by partitioning the edge map in terms of edge directions. There are two main directions in our constrained face pictures: horizontal and vertical[4].

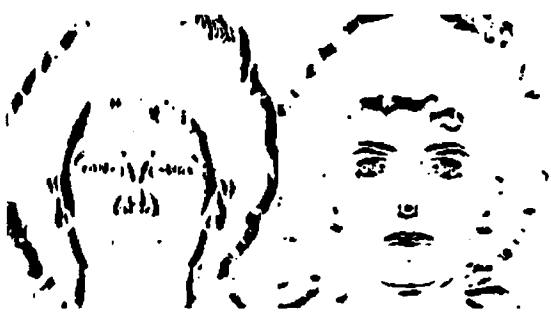

Fig. 1. Horizontal and vertical edge dominance maps

Horizontal gradients are useful to detect the left and right boundaries of face and nose, while vertical gradients are useful to detect the head top, eyes, nose base and mouth.

Once eyes have been located using template matching, the search for the other features can take advantage of the knowledge of their average layout.

Mouth and nose are located using similar strategies. The vertical position is guessed using anthropometric standards. A first, refined estimate of their real position is obtained

[4] A pixel is considered to be in the vertical edge map if the magnitude of the vertical component of the gradient at that pixel is greater than the horizontal one. The gradient is computed using a gaussian regularization of the image. Only points where the gradient intensity is above an automatically selected threshold are considered [21, 3].

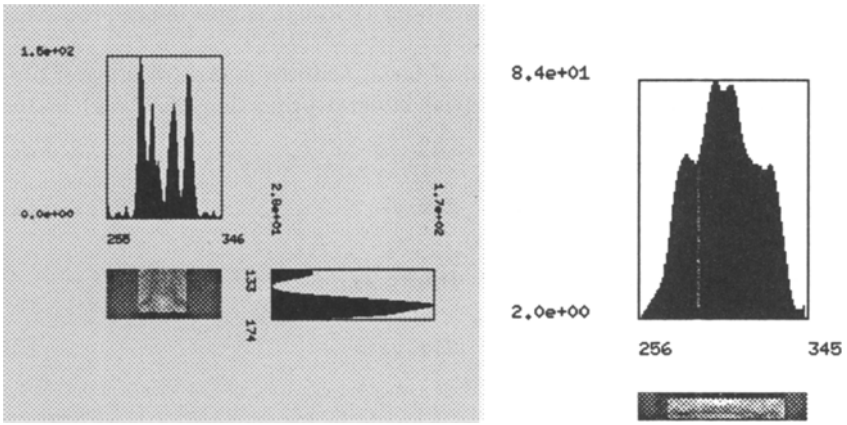

Fig. 2. LEFT: Horizontal and vertical nose restriction. RIGHT: Horizontal mouth restriction

looking for peaks of the horizontal projection of the vertical gradient for the nose, and for valleys of the horizontal projection of the intensity for the mouth (the line between the lips is the darkest structure in the area, due to its configuration). The peaks (and valleys) are then rated using their prominence and distance from the expected location (height and depth are weighted by a gaussian factor). The ones with the highest rating are taken to be the vertical position of nose and mouth. Having established the vertical position, search is limited to smaller windows.

The nose is delimited horizontally searching for peaks (in the vertical projection of horizontal edge map) whose height is above the average value in the searched window. The nose boundaries are estimated from the leftmost and rightmost peaks. Mouth height is computed using the same technique but applied to the vertical gradient component. The use of directional information is quite effective at this stage, cleaning much of the noise which would otherwise impair the feature extraction process. Mouth width is finally computed thresholding the vertical projection of the horizontal edge map at the average value (see Fig. 2).

Eyebrows position and thickness can be found through a similar analysis. The search is once again limited to a focussed window, just above the eyes, and the eyebrows are found using the vertical gradient map. Our eyebrows detector looks for pairs of peaks of gradient intensity with opposite direction. Pairs from one eye are compared to those of the other one: the most similar pair (in term of the distance from the eye center and thickness) is selected as the correct one.

We used a different approach for the detection of the face outline. Again we have attempted to exploit the natural constraints of faces. As the face outline is essentially elliptical, dynamic programming has been used to follow the outline on a gradient intensity map of an elliptical projection of the face image. The reason for using an elliptical coordinate system is that a typical face outline is approximately represented by a line. The computation of the cost function to be minimized (deviation from the assumed shape, an ellipse represented as a line) is simplified, resulting in a serial dynamic problem which can be efficiently solved [4].

In summary, the resulting set of 22 geometrical features that are extracted automatically in our system and that are used for recognition (see Fig. 3), is the following:

- eyebrows thickness and vertical position at the eye center position;

Fig. 3. Geometrical features (black) used in the face recognition experiments

- nose vertical position and width;
- mouth vertical position, width and height;
- eleven radii describing the chin shape;
- bigonial breadth;
- zygomatic breadth.

3.3 Recognition Performance

Detection of the features listed above associates to each face a twentytwo-dimensional numerical vector. Recognition is then performed with a Nearest Neighbor classifier, with a suitably defined metric. Our main experiment aim to characterize the performance of the feature-based technique as a function of the number of classes to be discriminated. Other experiments try to assess performance when the possibility of rejection is introduced. In all of the recognition experiments the learning set had an empty intersection with the testing set.

The first observation is that the vectors of geometrical features extracted by our system have low stability, i.e. the intra-class variance of the different features is of the same order of magnitude of the inter-class variance (from three to two times smaller). This is reflected by the superior performance we have been able to achieve using the centroid of the available examples (either 1 or 2 or 3) to model the frontal view of each individual (see Fig. 4).

An important step in the use of metric classification using a Nearest Neighbor classifier is the choice of the metric which must take into account both the interclass variance and the reliability of the extracted data. Knowledge of the feature detectors and of the face configuration allows us to establish, heuristically, different weights (reliabilities) for the single features. Let $\{x_i\}$ be the feature vector, $\{\sigma_i\}$ be the inter class dispersion vector and $\{w_i\}$ the weight (reliability) vector. The distance of two feature vectors $\{x_i\}$ $\{x'_i\}$ is then expressed as:

$$\Delta^{\alpha}(x, x') = \sum_{i=1}^{n} w_i \left(\frac{|x_i - x'_i|}{\sigma_i} \right)^{\alpha} \tag{4}$$

A useful data on the robustness of the classification is given by an estimate of the class separation. This can be done using the so called MIN/MAX ratio [17, 18], hereafter R_{mM}, which is defined as the minimum distance on a wrong correspondence over the distance from the correct correspondence. The performance of the classifier at different values of α has also been investigated. The value of α giving the best performance is $\alpha = 1.2$, while the *robustness* of the classification decreases with increasing α. This result, if generally true, may be extremely interesting for hardware implementations, since absolute values are much easier to compute in silicon than squares. The underlying reason for the good performance of α values close to 1 is probably related to properties of robust statistics [12] [5]. Once the metric has been set, the dependency of the performance on the number of classes can be investigated. To obtain these data, a number of recognition experiments have been conducted on randomly chosen subsets of classes at the different required cardinalities. The average values on round robin rotation experiments on the available sets are reported. The plots in Fig. 4 report both recognition performance and the R_{mM} ratio. As expected, both data exhibits a monotonically decreasing trend for increasing cardinality.

A possible way to enhance the robustness of classification is the introduction of a rejection threshold. The classifier can then suspend classification if the input is not sufficiently similar to any of the available models. Rejection could trigger the action of a different classifier or the use of a different recognition strategy (such as voice identification). Rejection can be introduced, in a metric classifier, by means of a rejection threshold: if the distance of a given input vector from all of the stored models exceeds the rejection threshold the vector is *rejected*. A possible figure of merit of a classifier with rejection is given by the recognition performance with no errors (vectors are either correctly recognized or rejected). The average performance of our classifier as a function of the rejection threshold is given in Fig. 5. [6]

4 Conclusion

A set of algorithms has been developed to assess the feasibility of recognition using a vector of geometrical features, such as nose width and length, mouth position and chin shape. The advantages of this strategy over techniques based on template matching are essentially:

- compact representation (as low as 22 bytes in the reported experiments);
- high matching speed.

The dependency of recognition performance using a Nearest Neighbor classifier has been reported for several parameters such as:

- number of classes to be discriminated (i.e. people to be recognized);
- number of examples per class;
- rejection threshold.

[5] We could have chosen other classifiers instead of Nearest Neighbor. The HyperBF classifier, used in previous experiments of 3D object recognition, allows the automatic choice of the appropriate metric, which is still, however, a weighted euclidean metric.

[6] Experiments by Lee on a OCR problem [15] suggest that a HyperBF classifier would be significantly better than a NN classifier in the presence of rejection thresholds.

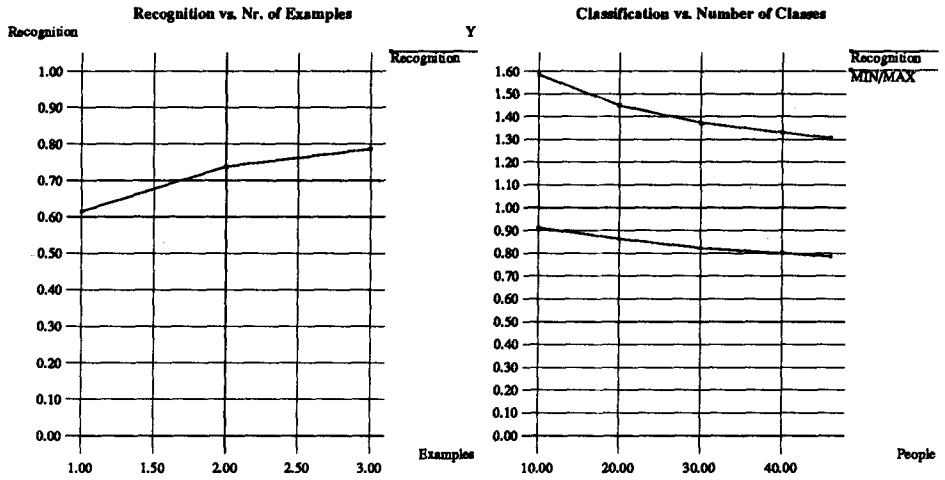

Fig. 4. LEFT: Performance as a function of the number of examples. RIGHT: Recognition performance and MIN/MAX ratio as a function of the number of classes to be discriminated

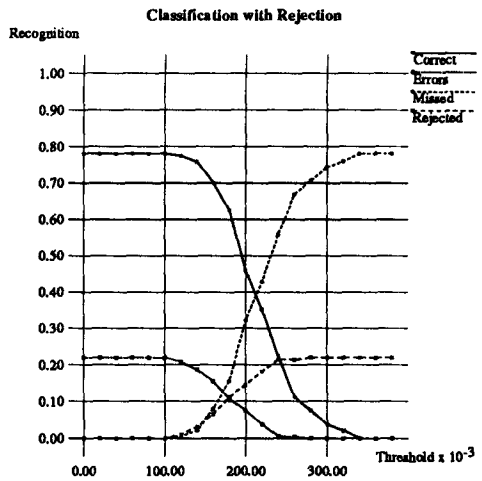

Fig. 5. Analysis of the classifier as a function of the rejection threshold

The attained performance suggests that recognition by means of a vector of geometrical features can be useful for small databases or as a screening step for more complex recognition strategies.

These data are the first results of a project which will compare several techniques for automated recognition on a common database, thereby providing quantitative information on the performance of different recognition strategies.

Acknowledgements

The authors thanks Dr. L. Stringa for helpful suggestions and stimulating discussions. One of the authors (R.B) thanks Dr. M. Dallaserra for providing the image data base. Thanks are also due to Dr. C. Furlanello for comments on an earlier draft of this paper.

References

1. R. J. Baron. Mechanisms of human facial recognition. *International Journal of Man Machine Studies*, 15:137–178, 1981.
2. W. W. Bledsoe. Man-machine facial recognition. Technical Report Rep. PRI:22, Panoramic Research Inc, Paolo Alto, Cal., 1966.
3. R. Brunelli. Edge projections for facial feature extraction. Technical Report 9009-12, I.R.S.T, 1990.
4. R. Brunelli. Face recognition: Dynamic programming for the detection of face outline. Technical Report 9104-06, I.R.S.T, 1991.
5. J. Buhmann, J. Lange, and C. von der Malsburg. Distortion invariant object recognition by matching hierarchically labeled graphs. In *Proceedings of IJCNN'89*, pages 151–159, 1989.
6. D. J. Burr. Elastic matching of line drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(6):708–713, 1981.
7. P. J. Burt. Smart sensing within a pyramid vision machine. *Proceedings of the IEEE*, 76(8):1006–1015, 1988.
8. H. Chan and W. W. Bledsoe. A man-machine facial recognition system: some preliminary results. Technical report, Panoramic Research Inc., Cal, 1965.
9. G. Cottrell and M. Fleming. Face recognition using unsupervised feature extraction. In *Proceedings of the International Neural Network Conference*, 1990.
10. A. J. Goldstein, L. D. Harmon, and A. B. Lesk. Identification of human faces. In *Proc. IEEE, Vol. 59*, page 748, 1971.
11. Zi-Quan Hong. Algebraic feature extraction of image for recognition. *Pattern Recognition*, 24(3):211–219, 1991.
12. P. J. Huber. *Robust Statistics*. Wiley, 1981.
13. T. Kanade. Picture processing by computer complex and recognition of human faces. Technical report, Kyoto University, Dept. of Information Science, 1973.
14. Y. Kaya and K. Kobayashi. A basic study on human face recognition. In S. Watanabe, editor, *Frontiers of Pattern Recognition*, page 265. 1972.
15. Y. Lee. Handwritten digit recognition using k nearest-neighbor, radial basis functions and backpropagation neural networks. *Neural Computation*, 3(3), 1991.
16. O. Nakamura, S. Mathur, and T. Minami. Identification of human faces based on isodensity maps. *Pattern Recognition*, 24(3):263–272, 1991.
17. T. Poggio and S. Edelman. A network that learns to recognize three-dimensional objects. *Nature*, 343(6225):1–3, 1990.
18. T. Poggio and F. Girosi. A theory of networks for approximation and learning. Technical Report A.I. Memo No. 1140, Massachusetts Institute of Technology, 1989.
19. J. Sergent. Structural processing of faces. In A.W. Young and H.D. Ellis, editors, *Handbook of Research on Face Processing*. North-Holland, Amsterdam, 1989.
20. M. Turk and A. Pentland. Eigenfaces for recognition. Technical Report 154, MIT Media Lab Vision and Modeling Group, 1990.
21. H. Voorhees. Finding texture boundaries in images. Technical Report AI-TR 968, M.I.T. Artificial Intelligence Laboratory, 1987.
22. A. W. Young and H. D. Ellis, editors. *Handbook of Research on Face Processing*. NORTH-HOLLAND, 1989.
23. Alan L. Yuille. Deformable templates for face recognition. *Journal of Cognitive Neuroscience*, 3(1):59–70, 1991.

This article was processed using the LaTeX macro package with ECCV92 style