A Feature Based Approach to Face Recognition

B. S. Manjunath
Department of Electrical and Computer Engineering
University of California
Santa Barbara, CA 93106-9560

R. Chellappa
Department of Electrical Engineering
University of Maryland
College Park, MD 20742

C. von der Malsburg
Ruhr-Universität Bochum
Institut für Neuroinformatik
Germany

Abstract

Faces represent one of the most common visual patterns in our environment, and humans have a remarkable ability to recognize faces. Face recognition does not fit into the traditional approaches of model based recognition in vision. We present here a feature based approach to face recognition, where the features are derived from the intensity data without assuming any knowledge of the face structure. The feature extraction model is biologically motivated, and the locations of the features often correspond to salient facial features such as the eyes, nose, etc. Topological graphs are used to represent relations between features, and a simple deterministic graph matching scheme which exploits the basic structure is used to recognize familiar faces from a database. Each of the stages in the system can be fully implemented in parallel to achieve real time recognition.

1 Introduction

In recent years the problem of face recognition has attracted considerable attention. Human faces represent one of the most common patterns that our visual system encounters. They also provide a good example of a class of natural objects which do not lend themselves to simple geometrical interpretations. Current vision techniques for this problem range from simple template matching to sophisticated feature based systems. The method presented here differs significantly from other work in the following respects. Although feature based, the features are derived from the raw intensity data without making use of any prior knowledge. Thus, even though most of the time the feature locations correspond to salient features in the face such as eyes, nose, mouth etc., there is no internal representation for these high level features as such. Secondly, this scheme allows for a very simple representation for each feature node, with an order of magnitude savings in the memory requirements (see section (4)).

The face recognition scheme we propose consists of three main stages. In the first stage we derive a description of the intensity image in terms of features. Intensity based approaches such as template matching or eigenvalue analysis are sensitive to changes in intensity which might be caused by local distortions and changes in viewing angle as well as translation. A feature based approach, on the other hand, is less sensitive to such changes. In the second stage we construct a graph representation of the face, with the nodes in the graph representing feature information, and the links representing feature relations. In our current implementation these links represent Euclidian distances between the feature nodes. The final step, that of recognition, is formulated as an inexact graph matching problem. This involves matching the input graph (of a face which is to be recognized) with a stored database. The matching itself involves minimizing an appropriate cost function.

1.1 Previous Work

Human faces provide a very good example of a class of natural objects which do not lend themselves to simple geometrical representations, and yet the human visual system does an excellent job in efficiently recognizing these images. Considerable research has been done in developing algorithms to solve this problem.

0-8186-2855-3/92 $3.00 ©1992 IEEE

Kanade [1] describes one of the early systems built for this task. The system automatically localizes features such as corners of the eyes, nostrils, mouth etc. Then a set of sixteen facial parameters corresponding to these features is computed. A simple Euclidian distance measure is then used to compute the similarity between a test face and a stored face. The best case performance of the system was 15 correct identifications out of 20 test faces. The test data differed from the training data in that there was a period of one month between the acquisition of the samples; in both cases a full frontal view was used.

While [1] describes a top-down analysis of the problem, a completely data driven method is suggested in [2] by Turk and Pentland. Their system tracks a person's head and identifies the face by comparing its features with a known database. The basic idea is to find a lower dimensional feature space to represent the intensity data, and they make use of principal component analysis. Their database consists of images (of 16 persons) taken under different lighting conditions, sizes and orientation. They report classification accuracy of 96% over lighting variations, 85% over orientation variations and 64% over size variation. This approach to recognition is similar to many earlier attempts in transforming a 3-D recognition problem to a 2-D matching, without detecting any perceptually significant features.

The above two methods illustrate two diverse approaches to this problem, and a comprehensive discussion on various aspects of face recognition can be found in [3]. In a different context, and as an example to illustrate the principle of dynamic link architecture, Lades et al. [4] also provide results of their experiments on face recognition. In their case, the basic features are the Gabor coefficients obtained by convolving the image with a bank of Gabor filters at multiple scales and orientations. These features correspond to edges and lines in the image. The authors report an impressive 97% classification over a data set of about 80 faces, with the training and test sets differing in the orientations of faces. For other connectionist approaches see for example Kohonen [5], and Fuchs and Haken [6].

As mentioned in the previous section, the method presented here is based on low level features and does not make any explicit use of high level information. The next section describes the model for feature detection and localization. In Section 3 we discuss the use of topological graphs for representing face information and a simple graph matching algorithm for recognition. Section 4 provides experimental results.

2 Feature Detection and Localization

The development of the feature detection model is motivated by the early processing stages in the visual cortex of mammals. The cells in the visual cortex can be classified into three broad functional categories: simple, complex and hypercomplex. Of particular interest here is the end-inhibition property exhibited by the hypercomplex cells. This property refers to the response of these cells to short lines and edges, line endings and sharp changes in curvature (e.g., corners). Since these correspond to some of the low level salient features in an image, these cells can be said to form in some sense a low level feature map of the intensity image. Although these cells were first discovered more than two decades back [7], it is only recently that researchers are beginning to gain some understanding of their role in visual perception. For example, von der Heydt and Peterhans [8] have provided conclusive experimental evidence that these cells play a very important role in perceiving illusory contours. A simple model incorporating their observations is developed in [9] for detecting texture boundaries and illusory contours.

The feature detection method presented here is based on a model of end-inhibition property, and it makes use of local scale interactions between simple oriented features. The scale interaction model was first suggested in [7] and more recently in [10]. The method described here consists of two basic steps: The first step is to extract oriented feature information at different scales. In the second step, interactions between these oriented features at different scales result in the end-inhibition effect.

Oriented feature information can be obtained by a Gabor wavelet transformation of the intensity image [9]. Gabor functions are Gaussians modulated by complex sinusoids. A wavelet transformation results in the decomposition of a signal in terms of basis functions, with all the basis functions obtained by simple dilations and translations of a basic wavelet. For the Gabor wavelet transformation, the basic function is of the form:

$$g_\lambda(x, y, \theta) = e^{-(\lambda^2 x'^2 + y'^2) + i\pi x'} \qquad (1)$$

$$x' = x\cos\theta + y\sin\theta$$

$$y' = -x\sin\theta + y\cos\theta$$

where λ is the spatial aspect ratio and θ is the preferred orientation. To simplify the notation, we drop the subscript λ and unless otherwise stated assume that $\lambda = 1$. For practical applications, discretization

of the parameters is necessary. The discretized parameters must cover the entire frequency spectrum of interest. Let the orientation range $[0, \pi]$ be discretized into N intervals and the scale parameter α be sampled exponentially as $\alpha^j, j \in \mathbf{Z}$. This results in the wavelet family

$$(g(\alpha^j(x - x_0, y - y_0), \theta_k)), \alpha \in \mathbf{R},\ j = \{0, -1, -2, ..\} \tag{2}$$

where $\theta_k = k\pi/N$. The Gabor wavelet transform is then defined by

$$W_j(x, y, \theta) = \int f(x_1, y_1)\, g^*(\alpha^j(x_1 - x, y_1 - y), \theta)\, dx_1 dy_1 \tag{3}$$

At each resolution in the representation hierarchy these wavelets localize the information content in both the frequency and spatial domains simultaneously. Any desired orientation selectivity can be obtained by controlling the parameter θ. The Gabor wavelet decomposition also has an important physical interpretation as to the type of features detected [9] and has been used in applications such as image coding [11, 12] and pattern recognition [4].

We now suggest a simple mechanism to model the behavior of end-inhibition. The hypercomplex cell receptive field must have inhibitory end zones along the preferred orientation. Such a profile can be generated either by modifying the profile of the simple cell itself or through interscale interactions, discussed below. The fact that both simple and complex cells often exhibit this end-stopping behavior further suggests that both these mechanisms are utilized in the visual cortex. If $Q_{ij}(x, y, \theta)$ denotes the output of the end-inhibited cell at position (x, y) receiving inputs from two frequency channels i and j $(\alpha^i > \alpha^j)$ with preferred orientation θ, then

$$Q_{ij}(x, y, \theta) = g(||W_i(x, y, \theta) - \gamma W_j(x, y, \theta)||) \tag{4}$$

where $\gamma = \alpha^{-2(i-j)}$ is the normalizing factor.

The next step is to localize the curvature changes signalled by these feature detectors. Locations (x, y) in the image which are identified as feature locations satisfy the following:

$$Q_{ij}(x, y) = \max_{(x', y') \in N_{xy}} Q_{ij}(x', y') \tag{5}$$

where

$$Q_{ij}(x', y') = \max_{\theta} Q_{ij}(x', y', \theta)$$

and $Q_{ij}(x', y', \theta)$ is given by (4). N_{xy} represents a local neighborhood of (x, y) within which the search is conducted.

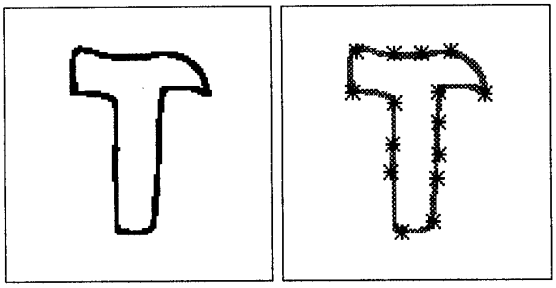

Figure 1: Salient features detected by the system. For the hand drawn hammer image, all the feature locations correspond to significant changes in curvature. The particular scale-pair used in this example is $i = 0$, $j = -6$, with $\alpha = \sqrt{2}$.

Figure 1 illustrates the observation that the feature locations correspond to points with significant curvature changes. All the corners in this hand-drawn hammer picture are located by the algorithm, although only one particular set of parameters is used for the scales. Figure 2 shows the location of features that are detected for a pair of face images. Information at these locations is used in the recognition process and will be discussed in detail in the following. In addition to this application, the feature detection scheme has been successfully used in motion tracking [13] where it is used to identify salient points in the image to be tracked, and in image registration [14]. The image registration application illustrates the robustness of this method in identifying consistent set of features irrespective of significant amounts of rotation, scaling and perspective distortion between pairs of images.

3 Representation and Recognition Using Graphs

The next step is to represent information about the face using the available information at the feature points. Topological graphs are used in our recognition scheme to represent relationships between features. For convenience the features detected in a given image are numbered as {1,2,...} (in any arbitrary, but consistent way). The nodes V_i in the graph correspond to the feature points, and are characterized by $\{S, \mathbf{q}\}$, where S represents information about the spatial location, and

$$\mathbf{q}_i = [Q_i(x, y, \theta_1), \ldots, Q_i(x, y, \theta_N)] \tag{6}$$

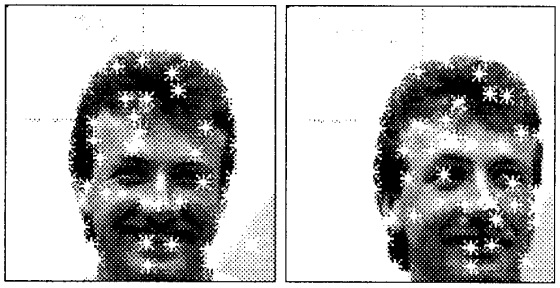

Figure 2: Feature locations marked for a pair of face images. The scales used in this case correspond to $i = -2, j = -5$ ($\alpha = \sqrt{2}$). Information at the feature locations is stored and used during the recognition process.

is the feature vector corresponding to the ith feature. Let N_i denote the set of neighbors of ith node. Directional edges connect the neighbors in the graph (i.e., the neighborhood is not symmetric). The neighborhood of a node is determined by taking into account both the maximum number of neighbors allowed as well as the distance between them. The Euclidian distance between two nodes V_i and V_j is denoted by d_{ij}.

To efficiently identify the input graph with a stored one (which is most similar to the input one based on certain criteria) is another important issue, and has received considerable attention recently. We describe below a very simple algorithm which only involves local search, is deterministic in nature and extremely fast. The algorithm, however, does not guarantee optimizing the criterion function. In spite of this the recognition rate is comparable to that of most face recognition schemes that we are aware of, demonstrating further the robustness of our feature extraction. Our implementation of the matching algorithm is given below:

In the following, subscripts i, j refer to nodes in the input graph $\mathcal{I}$, and i', j', m', n' correspond to nodes in the stored graph $\mathcal{O}$.

1. The input graph $\mathcal{I}$ is spatially aligned with the stored graph $\mathcal{O}$ by matching the centroids of the features $\{V_i\}$ and $\{V_i'\}$.

2. Let W_i be the spatial neighborhood for the ith feature in the input graph, over which a search is conducted to find the best matching feature node $V_{i'}$ in the stored graph, such that

$$S_{ii'} = 1 - \frac{\mathbf{q}_i \cdot \mathbf{q}_{i'}}{||\mathbf{q}_i||\,||\mathbf{q}_{i'}||} = \min_{m' \in W_i} S_{im'} \qquad (7)$$

3. After all the individual features are matched, total cost is computed by taking into account the topology of the matched graphs. Let the nodes i and j match i' and j' respectively, and further let $j \in N_i$ (i.e., V_j is a neighbor of V_i). Let $\rho_{ii'j'j} = \min\{d_{ij}/d_{i'j'}, d_{i'j'}/d_{ij}\}$. Then the topology cost for this particular pair of nodes is computed as

$$T_{ii'j'j} = 1 - \rho_{ii'j'j} \qquad (8)$$

Note that if the match is perfect, $d_{ij} = d_{i'j'}$ and $T_{ii'j'j} = 0$.

4. The total cost for matching input graph $\mathcal{I}$ to a stored graph $\mathcal{O}$ is then given by

$$C_1(\mathcal{I}, \mathcal{O}) = \sum_i S_{ii'} + \lambda_t \sum_i \sum_{j \in N_i} T_{ii'j'j} \qquad (9)$$

where λ_t is a scaling parameter which controls the relative importance of the two cost functions.

5. The total cost is then scaled appropriately to reflect the the difference in the number of features between the input and stored graphs. If $n_{\mathcal{I}}, n_{\mathcal{O}}$ denote the number of feature nodes in the input and stored graphs respectively, then the scaling factor $s_f = \max\{n_{\mathcal{I}}/n_{\mathcal{O}}, n_{\mathcal{O}}/n_{\mathcal{I}}\}$, and the scaled total cost $C(\mathcal{I}, \mathcal{O}) = s_f C_1(\mathcal{I}, \mathcal{O})$.

6. The best candidate match $\mathcal{O}^*$ then satisfies

$$C(\mathcal{I}, \mathcal{O}^*) = \min_{\mathcal{O}'} C(\mathcal{I}, \mathcal{O}') \qquad (10)$$

Note that the above algorithm does not take into account the topology cost during the matching process. The topology cost is computed only after the features are matched. The advantage is that there are no iterations, and no stochastic elements involved in the search, resulting in a very fast algorithm for matching.

4 Experimental Results and Discussion

We have implemented a simple face recognition system based on the above principles. The input is a 128 x 128 image, having very little background noise. In our current implementation, the feature responses

are computed at only one scale, corresponding to the scale parameters $i = -2, j = -5$ in (4). Typical numbers of feature points detected in a face image using (5) vary from 35 to 50. The number of discrete orientations used was $N = 4$ (in (6)), corresponding to $\theta = \{0, 45, 90, 135\}$. One byte of information is stored for each of the components in the feature vector, or approximately 200 bytes of information per face. This constitutes an order of magnitude savings in memory, from the 16K raw intensity data.

The database we have used has face images of 86 persons, with two to four images per person, taken with different facial expressions and/or orientations. Often there is a small amount of translation and scaling as well. There are a total of 306 such face images in the current database. Figure 3 shows some of the images in the database. For each face image, the stored information corresponds to the feature graph $\{S, \mathbf{q}\}$ discussed in the previous section. The neighborhood set N_i of a feature node i consists of its five nearest neighbors. Note that this set is not necessarily symmetric.

The performance of the system is evaluated as follows: For each entry of a face image in the database, the cost of associating another entry in the database is computed according to (9). The parameter λ_t in (9) is set to 0.2, so as to have equal contributions to the total cost from the similarity measure and the topological cost (as the summation over j is over the neighbors, which in our case total five). These costs are then sorted and the best match is the one having the minimum associated cost as in (10). Note that in doing this self-matches (which obviously result in zero total cost) are ignored. The recognition accuracy in terms of the best match corresponding to the right person was 86%, and in 94% of the cases the correct person's face was in the top three candidate matches. The graph matching steps 1 through 5 discussed in Section 3 typically take less than 0.5 seconds for each graph (on a SUN-Sparc workstation). Some results of successful matches as well as failures are shown in Figures 4 and 5.

In a typical application of this system, one can store 10 to 20 images of each person's face, taken from different angles, with different facial expressions. Any incoming face image can then be matched to this set of images, and a threshold can be associated with the matching cost to either accept a match or to reject. Due to the nature of representation used, the associated memory requirements are minimal. The entire matching process can be implemented on a parallel hardware or connectionist network for real time applications. Among the issues to be addressed for future work are the scale invariance and use of high level feature information.

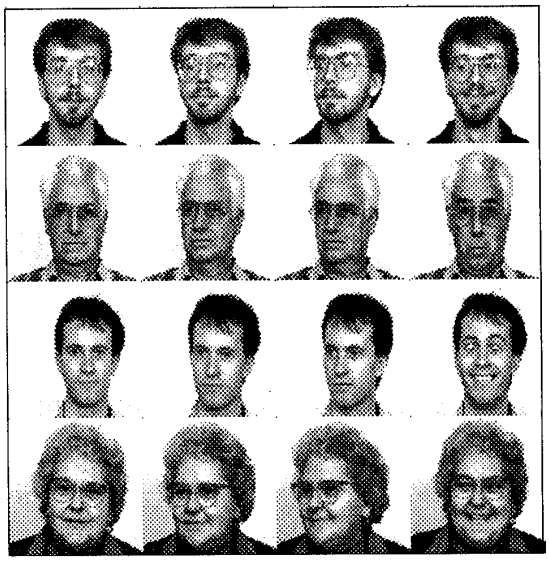

Figure 3: Some of the face images in the database.

Acknowledgements

This work was carried out in part at the Signal and Image Processing Institute, University of Southern California. This research was partially supported by grants DARPA No. 6989 and DACA 76-89-C-0019.

References

[1] T. Kanade, *Picture processing system by computer complex and recognition of human faces.* PhD thesis, Kyoto University, Department of Information Science, November 1973.

[2] M. Turk and A. Pentland, "Face recognition using eigenfaces," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (Maui, Hawaii), pp. 586–591, June 1991.

[3] V. Bruce and M. Burton, "Computer recognition of faces," in *Handbook of Research on Face Processing* (A. W. Young and H. D. Ellis, eds.),

Figure 4: Examples of successful matches. The left image of each pair is the input image to the system, and the right image is the best match found.

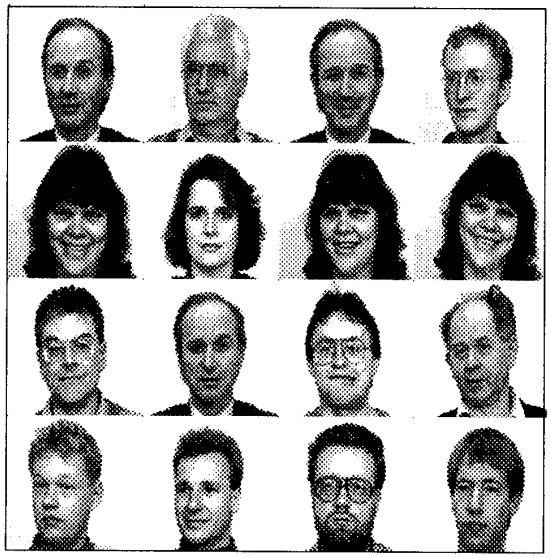

Figure 5: Examples of failures. The first image in each row is the input image, and the following three images are the top three matches found. Note that in the first two rows the correct match is among the three best matches.

pp. 487–506, Elsevier Science Publishers B.V. (North Holland), 1989.

[4] M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Würtz, and W. Konen, "Distortion invariant object recognition in the dynamic link architecture," 1991. preprint.

[5] T. Kohonen, *Self-Organization and Associative Memory*. New York: Springer-Verlag, 1989.

[6] A. Fuchs and H. Haken, "Pattern recognition and associative memory as dynamical processes in a synergetic system ii," *Biological Cybernetics*, vol. 60, pp. 107–109, 1988.

[7] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture in two nonstriate visual areas(18 and 19) of the cat," *Journal of Neurophysiology*, vol. 28, pp. 229–289, March 1965.

[8] R. von der Heydt and E. Peterhans, "Mechanisms of contour perception in monkey visual cortex. i. lines of pattern discontinuity," *Journal of Neuroscience*, vol. 9, pp. 1731–1748, May 1989.

[9] B. S. Manjunath and R. Chellappa, "A computational model for boundary detection," in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (Maui, Hawaii), pp. 358–363, June 1991.

[10] A. Dobbins, S. W. Zucker, and M. S. Cynader, "Endstopped neurons in the visual cortex as a substrate for calculating curvature," *Nature*, vol. 329, pp. 438–441, October 1987.

[11] M. Porat and Y. A. Zeevi, "The generalized gabor scheme of image representation in biological and machine vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-10, pp. 452–468, July 1988.

[12] J. G. Daugman, "Relaxation neural network for non-orthogonal image transforms," in *Proc. Int. Conf. on Neural Networks*, vol. 1, (San Diego, CA), pp. 547–560, June 1988.

[13] S. Chandrashekhar and R. Chellappa, "Passive navigation in a partially known environment," in *IEEE Workshop on Visual Motion*, (Princeton, NJ), pp. 2–7, October 1991.

[14] Q. Zheng, R. Chellappa, and B. S. Manjunath, "Balloon motion estimation using two frames," in *Proc. 25th Asilomar Conf. on Systems and Signals*, November 1991. (invited paper).