In *Intelligent Biometric Techniques in Fingerprint and Face Recognition*,
eds. L.C. Jain *et al.*, publ. CRC Press, ISBN 0-8493-2055-0, Chapter 11, pp. 355-396, (1999).

# Face Recognition by Elastic Bunch Graph Matching[*][†]

Laurenz Wiskott[1‡], Jean-Marc Fellous[2§],
Norbert Krüger[1¶], and Christoph von der Malsburg[1,2]

[1] Institute for Neural Computation
Ruhr-University Bochum
D-44780 Bochum, Germany

[2] Computer Science Department
University of Southern California
Los Angeles, CA 90089, USA

**Abstract**

We present a system for recognizing human faces from single images out of a large database containing one image per person. The task is difficult because of image variation in terms of position, size, expression, and pose. The system collapses most of this variance by extracting concise face descriptions in the form of *image graphs*. In these, fiducial points on the face (eyes, mouth, etc.) are described by sets of wavelet components (*jets*). Image graph extraction is based on a novel approach, the *bunch graph*, which is constructed from a small set of sample image graphs. Recognition is based on a straightforward comparison of image graphs. We report recognition experiments on the FERET database as well as the Bochum database, including recognition across pose.

## 1 Introduction

We set ourselves the task of recognizing persons from single images by reference to a gallery, which also contained only one image per person. Our problem was to address image variation due to differences in facial expression, head pose, position, and size (to name only the most important). Our task is thus a typical discrimination-in-the-presence-of-variance problem, where one has to try to collapse the variance and to emphasize discriminating features. This is generally only possible with the help of information about the structure of variations to be expected.

Classification systems differ vastly in terms of the nature and origin of their knowledge about image variations. Systems in Artificial Intelligence and Computer Vision often stress specific designer-provided structures, for instance explicit models of three-dimensional objects or of the image-generation process, whereas Neural Network models tend to stress absorption of structure from examples with the help of statistical estimation techniques. Both of these extremes are expensive in their own way and fall painfully

*Supported by grants from the German Federal Ministry for Science and Technology (413-5839-01 IN 101 B9) and from ARPA and the U.S. Army Research Lab (01/93/K-109).

†Portions reprinted, with permission, from IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7):775–779, July 1997. 

‡Current address: Institute for Advanced Studies, Wallotstrasse 19, D-14193 Berlin, Germany, wiskott@wiko-berlin.de.

§Current address: Computational Neurobiology Laboratory, The Salk Institute for Biological Studies, San Diego, CA 92186-5800, fellous@salk.edu.

¶Current address: Institute for Computer Science, Christian-Albrecht-University Kiel, Preusserstrasse 1-9, D-24105 Kiel, Germany, nkr@informatik.uni-kiel.de.

short of the ease with which natural systems pick up essential information from just a few examples. Part of the success of natural systems must be due to general properties and laws on how object images transform under natural conditions.

Our system has an important core of structure which reflects the fact that the images of coherent objects tend to translate, scale, rotate, and deform in the image plane. Our basic object representation is the labeled graph; edges are labeled with distance information and nodes are labeled with wavelet responses locally bundled in *jets*. Stored *model graphs* can be matched to new images to generate *image graphs*, which can then be incorporated into a gallery and become model graphs. Wavelets as we use them are robust to moderate lighting changes and small shifts and deformations. Model graphs can easily be translated, scaled, oriented, or deformed during the matching process, thus compensating for a large part of the variance of the images. Unfortunately, having only one image for each person in the galleries does not provide sufficient information to handle rotation in depth analogously. However, we present results on recognition across different poses.

This general structure is useful for handling any kind of coherent object and may be sufficient for discriminating between structurally different object types. However, for in-class discrimination of objects, of which face recognition is an example, it is necessary to have information specific to the structure common to all objects in the class. This is crucial for the extraction of those structural traits from the image which are important for discrimination ("to know where to look and what to pay attention to"). In our system, class-specific information has the form of *bunch graphs*, one for each pose, which are stacks of a moderate number (70 in our experiments) of different faces, jet-sampled in an appropriate set of fiducial points (placed over eyes, mouth, contour, etc.). Bunch graphs are treated as combinatorial entities in which, for each fiducial point, a jet from a different sample face can be selected, thus creating a highly adaptable model. This model is matched to new facial images to reliably find the fiducial points in the image. Jets at these points and their relative positions are extracted and are combined into an image graph, a representation of the face which has no remaining variation due to size, position (or in-plane orientation, not implemented here).

A bunch graph is created in two stages. Its qualitative structure as a graph (a set of nodes plus edges) as well as the assignment of corresponding labels (jets and distances) for one initial image is designer-provided, whereas the bulk of the bunch graph is extracted semi-automatically from sample images by matching the embryonic bunch graph to them, less and less often intervening to correct incorrectly identified fiducial points. Image graphs are rather robust to small in-depth rotations of the head. Larger rotation angles, i.e. different poses, are handled with the help of bunch graphs with a different graph structure and designer-provided correspondences between nodes in different poses.

After these preparations our system can extract from single images concise invariant face descriptions in the form of image graphs (called model graphs when in a gallery). They contain all information relevant for the face discrimination task. For the purpose of recognition, image graphs can be compared with model graphs at small computing cost by evaluating the mean jet similarity.

In summary, our system is based to a maximum on a general data structure — graphs labeled with wavelet responses — and general transformation properties. These are designer-provided, but due to their generality and simplicity the necessary effort is minimal. As described here our system makes use of hand-crafted object-specific graph structures and a moderately labor-intensive procedure to generate bunch-graphs. We plan to eliminate this need for human intervention and guess-work with the help of statistical estimation methods (cf. Maurer & von der Malsburg, 1996; Krüger et al., 1998). Our system comes close to the natural model by needing only a small number of examples to handle the complex task of face recognition.

This work has been described in short form in (Wiskott et al., 1997), from which portions have been adopted for this text. In the discussion we will compare our system to others and to our own previous work.

# 2 The System

## 2.1 Preprocessing with Gabor Wavelets

The representation of local features is based on the Gabor wavelet transform; see Figure 1. Gabor wavelets are biologically motivated convolution kernels in the shape of plane waves restricted by a Gaussian envelope function (Daugman, 1988). The set of convolution coefficients for kernels of different orientations and frequencies at one image pixel is called a jet. In this section we define jets, different similarity functions between jets, and our procedure for precise localization of jets in an image.

#### 2.1.1 Jets

A *jet* describes a small patch of grey values in an image $\mathcal{I}(\vec{x})$ around a given pixel $\vec{x} = (x, y)$. It is based on a wavelet transform, defined as a convolution

$$\mathcal{J}_j(\vec{x}) = \int \mathcal{I}(\vec{x}')\psi_j(\vec{x} - \vec{x}')d^2\vec{x}' \tag{1}$$

with a family of *Gabor kernels*

$$\psi_j(\vec{x}) = \frac{k_j^2}{\sigma^2} \exp\left(-\frac{k_j^2 x^2}{2\sigma^2}\right) \left[\exp(i\vec{k}_j\vec{x}) - \exp\left(-\frac{\sigma^2}{2}\right)\right] \tag{2}$$

in the shape of plane waves with wave vector $\vec{k}_j$, restricted by a Gaussian envelope function. We employ a discrete set of 5 different frequencies, index $\nu = 0, ..., 4$, and 8 orientations, index $\mu = 0, ..., 7$,

$$\vec{k}_j = \begin{pmatrix} k_{jx} \\ k_{jy} \end{pmatrix} = \begin{pmatrix} k_\nu \cos\varphi_\mu \\ k_\nu \sin\varphi_\mu \end{pmatrix}, \; k_\nu = 2^{-\frac{\nu+2}{2}}\pi, \; \varphi_\mu = \mu\frac{\pi}{8}, \tag{3}$$

with index $j = \mu + 8\nu$. This sampling evenly covers a band in frequency space. The width $\sigma/k$ of the Gaussian is controlled by the parameter $\sigma = 2\pi$. The second term in the bracket of Eq. (2) makes the kernels *DC-free*, i.e. the integral $\int \psi_j(\vec{x})d^2\vec{x}$ vanishes. This is known as a wavelet transform because the family of kernels is self-similar, all kernels being generated from one *mother wavelet* by dilation and rotation.

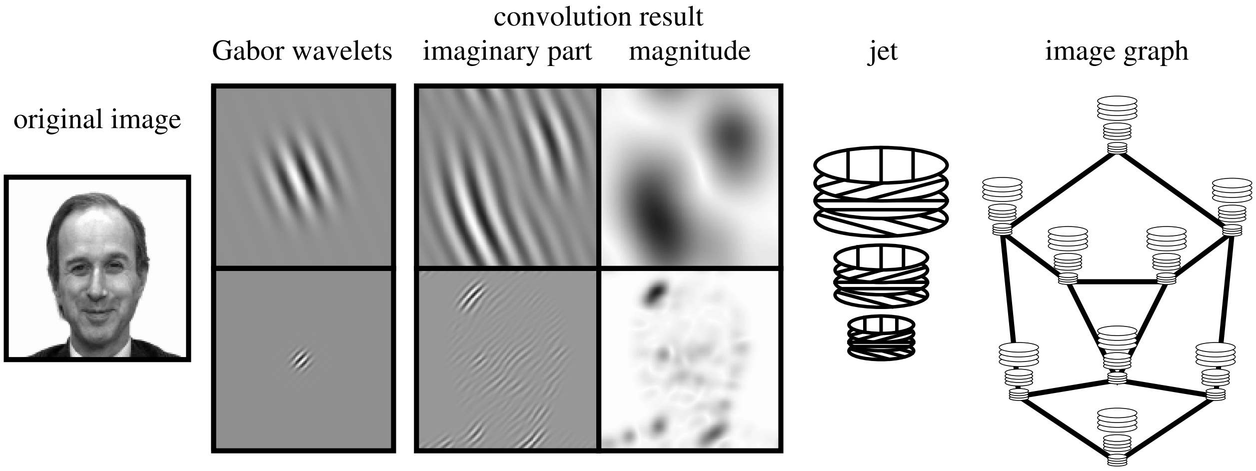


Figure 1: The graph representation of a face is based on the Gabor wavelet transform, a convolution with a set of wavelet kernels. These have the shape of plane waves restricted by a Gaussian envelope function. We compute 40 coefficients (5 frequencies × 8 orientations). Coefficient phase varies approximately with wavelet frequency (see imaginary part), magnitude varies slowly. The set of 40 coefficients obtained for one image point is referred to as a jet (for clarity, only 3 frequencies and 4 orientations are represented in the figure). A sparse collection of such jets together with some information about their relative location constitutes an image graph, used to represent an object, such as a face.

A jet $\mathcal{J}$ is defined as the set $\{\mathcal{J}_j\}$ of 40 complex coefficients obtained for one image point. It can be written as

$$\mathcal{J}_j = a_j \exp(i\phi_j) \tag{4}$$

with magnitudes $a_j(\vec{x})$, which slowly vary with position, and phases $\phi_j(\vec{x})$, which rotate at a rate approximately determined by the spatial frequency or wave vector $\vec{k}_j$ of the kernels; see Figure 1.

Gabor wavelets were chosen for their robustness as a data format and for their biological relevance. Since they are DC-free, they provide robustness against varying brightness in the image. Robustness against varying contrast can be obtained by normalizing the jets. The limited localization in space and frequency yields a certain amount of robustness against translation, distortion, rotation, and scaling. Only the phase changes drastically with translation. This phase variation can be either ignored, or it can be used for estimating displacement, as will be shown later. A disadvantage of the large kernels is their sensitivity to background variations. It has been shown, however, that if the object contour is known the influence of the background can be suppressed (Pötzsch et al., 1996). Finally, the Gabor wavelets have a similar shape as the receptive fields of simple cells found in the visual cortex of vertebrate animals (Pollen & Ronner, 1981; Jones & Palmer, 1987; DeValois & DeValois, 1988) and can be statistically derived from images of natural scenes, at least qualitatively (Olshausen & Field, 1996; Bell & Sejnowski, 1997).

#### 2.1.2 Comparing Jets

Due to phase rotation, jets taken from image points only a few pixels apart from each other have very different coefficients, although representing almost the same local feature. This can cause severe problems for matching. We therefore either ignore the phase or compensate for its variation explicitly. The similarity function

$$\mathcal{S}_a(\mathcal{J}, \mathcal{J}') = \frac{\sum_j a_j a'_j}{\sqrt{\sum_j a_j^2 \sum_j a'^2_j}}, \tag{5}$$

already used in (Lades et al., 1993), ignores phase. With a jet $\mathcal{J}$ taken at a fixed image position and jets $\mathcal{J}' = \mathcal{J}'(\vec{x})$ taken at variable position $\vec{x}$, $\mathcal{S}_a(\mathcal{J}, \mathcal{J}'(\vec{x}))$ is a smooth function with local optima forming large attractor basins (see Figure 2a), leading to rapid and reliable convergence with simple search methods such as stochastic gradient descent.

Using phase has two potential advantages. Firstly, phase information is required to discriminate between patterns with similar magnitudes, should they occur, and secondly, since phase varies so quickly with location, it provides a means for accurate jet localization in an image. Assuming that two jets $\mathcal{J}$ and $\mathcal{J}'$ refer to object locations with small relative displacement $\vec{d}$, the phase shifts can be approximately compensated for by the terms $\vec{d}\vec{k}_j$, leading to a phase-sensitive similarity function

$$\mathcal{S}_\phi(\mathcal{J}, \mathcal{J}') = \frac{\sum_j a_j a'_j \cos(\phi_j - \phi'_j - \vec{d}\vec{k}_j)}{\sqrt{\sum_j a_j^2 \sum_j a'^2_j}}. \tag{6}$$

To compute it, the displacement $\vec{d}$ has to be estimated. This can be done by maximizing $\mathcal{S}_\phi$ in its Taylor expansion, as explained in the following section. It is actually a great advantage of this second similarity function that it yields this displacement information. Profiles of similarities and estimated displacements are shown in Figure 2.

#### 2.1.3 Displacement Estimation

To estimate the displacement vector $\vec{d} = (d_x, d_y)$, we have adopted a method used for disparity estimation (Fleet & Jepson, 1990; Theimer & Mallot, 1994). The idea is to maximize the similarity $\mathcal{S}_\phi$ in its Taylor expansion:

$$\mathcal{S}_\phi(\mathcal{J}, \mathcal{J}') \approx \frac{\sum_j a_j a'_j [1 - 0.5(\phi_j - \phi'_j - \vec{d}\vec{k}_j)^2]}{\sqrt{\sum_j a_j^2 \sum_j a'^2_j}}. \tag{7}$$

Setting $\frac{\partial}{\partial d_x}\mathcal{S}_\phi = \frac{\partial}{\partial d_y}\mathcal{S}_\phi = 0$ and solving for $\vec{d}$ leads to

$$\vec{d}(\mathcal{J}, \mathcal{J}') = \begin{pmatrix} d_x \\ d_y \end{pmatrix} = \frac{1}{\Gamma_{xx}\Gamma_{yy} - \Gamma_{xy}\Gamma_{yx}} \times \begin{pmatrix} \Gamma_{yy} & -\Gamma_{yx} \\ -\Gamma_{xy} & \Gamma_{xx} \end{pmatrix} \begin{pmatrix} \Phi_x \\ \Phi_y \end{pmatrix}, \tag{8}$$

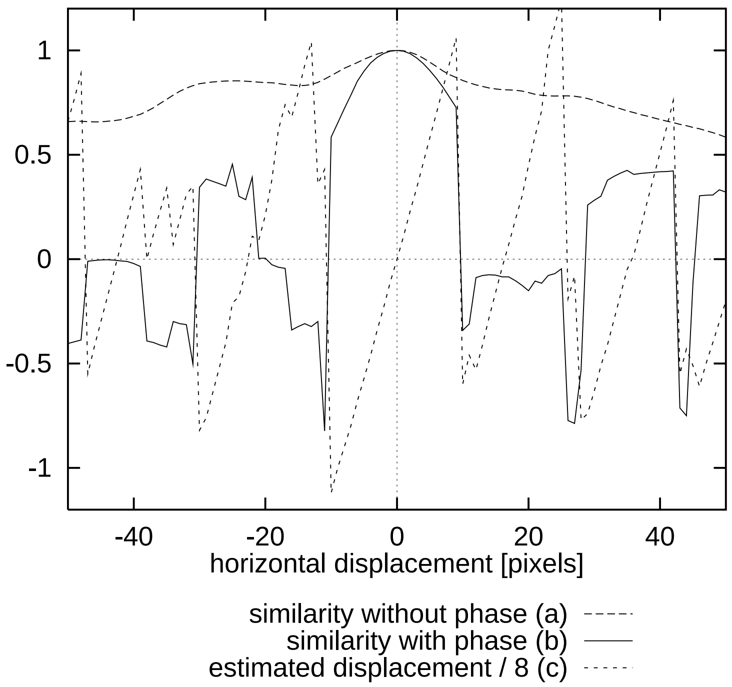


Figure 2: **a)** Similarity $\mathcal{S}_a(\mathcal{J}(\vec{x}_1), \mathcal{J}'(\vec{x}_0))$ with jet $\mathcal{J}'$ taken from the left eye of the face shown in Figure 1 and jet $\mathcal{J}$ taken from pixel positions of the same horizontal line, $\vec{x}_1 = \vec{x}_0 + (d_x, 0), d_x = -50, ..., 50$ (The image in Figure 1 has a width of 128 pixels). The similarity potential is smooth and has a large attractor basin. **b)** Similarity $\mathcal{S}_\phi(\mathcal{J}(\vec{x}_1), \mathcal{J}'(\vec{x}_0))$ and **c)** estimated displacement $\vec{d}(\mathcal{J}(\vec{x}_1), \mathcal{J}'(\vec{x}_0))$ for the same jets as in (a) (using focus 1). The similarity potential has many more local optima. The right eye is 24 pixels away from the left eye, generating a local maximum for both similarity functions close to $d_x = -24$. The estimated displacement is precise around the 0-position and rougher at other local optima, especially at the other eye. (The displacement values are divided by 8 to fit the ordinate range.)

if $\Gamma_{xx}\Gamma_{yy} - \Gamma_{xy}\Gamma_{yx} \neq 0$, with

$$\begin{aligned}
\Phi_x &= \sum\nolimits_j a_j a'_j k_{jx} (\phi_j - \phi'_j), \\
\Gamma_{xy} &= \sum\nolimits_j a_j a'_j k_{jx} k_{jy},
\end{aligned}$$

and $\Phi_y, \Gamma_{xx}, \Gamma_{yx}, \Gamma_{yy}$ defined correspondingly. In addition, the phase differences may have to be corrected by $\pm 2\pi$ to put them in the range of $\pm\pi$.

This equation yields a straightforward method for estimating the displacement or disparity between two jets taken from object locations close enough that their Gabor kernels are highly overlapping. Without further modifications, this equation can determine displacements up to half the wavelength of the highest frequency kernel, which would be two pixels for $k_0 = \pi/2$. The range can be increased by using low frequency kernels only. For the largest kernels, the estimated displacement may be 8 pixels. One can then proceed with the next higher frequency level and refine the result. When stepping to the next higher frequency, the phases of the higher frequency coefficients have to be corrected by multiples of $2\pi$ to match as closely as possible the expected phase differences inferred from the displacement estimated on the lower frequency level. This correction may lead to absolute phase differences larger than $\pi$. We refer to the number of frequency levels used for the first displacement estimation as *focus*. A focus of 1 indicates that only the lowest frequency level is used and that the estimated displacement may be up to 8 pixels. A focus of 5 indicates that all five levels are used, and the disparity may only be up to 2 pixels. In any case, all five levels are eventually used in the iterative refinement process described above.

If one has access to the whole image of jets, one can also work iteratively. Assume a jet $\mathcal{J}$ is to be accurately positioned in the neighborhood of point $\vec{x}_0$ in an image. Comparing $\mathcal{J}$ with the jet $\mathcal{J}_0 = \mathcal{J}(\vec{x}_0)$ yields an estimated displacement of $\vec{d}_0 = \vec{d}(\mathcal{J}, \mathcal{J}(\vec{x}_0))$. Then a jet $\mathcal{J}_1$ is taken from position $\vec{x}_1 = \vec{x}_0 + \vec{d}_0$ and the displacement is estimated again. But since the new location is closer to the correct position, the new displacement $\vec{d}_1$ will be smaller and can be estimated more accurately with a higher focus, converging eventually to subpixel accuracy. We have used this iterative scheme in the matching process described in Section 2.3.

## 2.2 Face Representation

### 2.2.1 Individual Faces

For faces, we have defined a set of *fiducial points*, e.g. the pupils, the corners of the mouth, the tip of the nose, the top and bottom of the ears, etc. A *labeled graph* $\mathcal{G}$ representing a face consists of $N$ nodes on these fiducial points at positions $\vec{x}_n, n = 1, ..., N$ and $E$ edges between them. The nodes are labeled with jets $\mathcal{J}_n$. The edges are labeled with distances $\Delta\vec{x}_e = \vec{x}_n - \vec{x}_{n'}, e = 1, ..., E$, where edge $e$ connects node $n'$ with $n$. Hence the edge labels are two-dimensional vectors. (When referring to the geometrical structure of a graph, unlabeled by jets, we call it a *grid*.) This face graph is *object-adapted*, since the nodes are selected from face-specific points (fiducial points); see Figure 4.

Graphs for different head pose differ in geometry and local features. Although the fiducial points refer to corresponding object locations, some may be occluded, and jets as well as distances vary due to rotation in depth. To be able to compare graphs for different poses, we have manually defined pointers to associate corresponding nodes in the different graphs.

### 2.2.2 Face Bunch Graphs

To find fiducial points in new faces, one needs a general representation rather than models of individual faces. This representation should cover a wide range of possible variations in the appearance of faces, such as differently shaped eyes, mouths, or noses, different types of beards, variations due to sex, age, race, etc. It is obvious that it would be too expensive to cover each feature combination by a separate graph. We instead combine a representative set of individual model graphs into a stack-like structure, called a *face bunch graph* (FBG); see Figure 3. Each model has the same grid structure and the nodes refer to identical fiducial points. A set of jets referring to one fiducial point is called a *bunch*. An eye bunch, for instance, may include jets from closed, open, female, and male eyes, etc., to cover these local variations. During the location of fiducial points in a face not seen before, the procedure described in the next section selects the best fitting jet, called the *local expert*, from the bunch dedicated to each fiducial point. Thus, the full combination of jets in the bunch graph is available, covering a much larger range of facial variation than represented in the constituting model graphs themselves. A similar data structure based on templates has been developed independently by Beymer (1994).

Assume for a particular pose that there are $M$ model graphs $\mathcal{G}^{\mathcal{B}m}$ $(m = 1, ..., M)$ of identical structure, taken from different model faces. The corresponding FBG $\mathcal{B}$ is then given the same structure, its nodes are labeled with bunches of jets $\mathcal{J}_n^{\mathcal{B}m}$ and its edges are labeled with the averaged distances $\Delta\vec{x}_e^{\mathcal{B}} = \sum_m \Delta\vec{x}_e^{\mathcal{B}m}/M$.

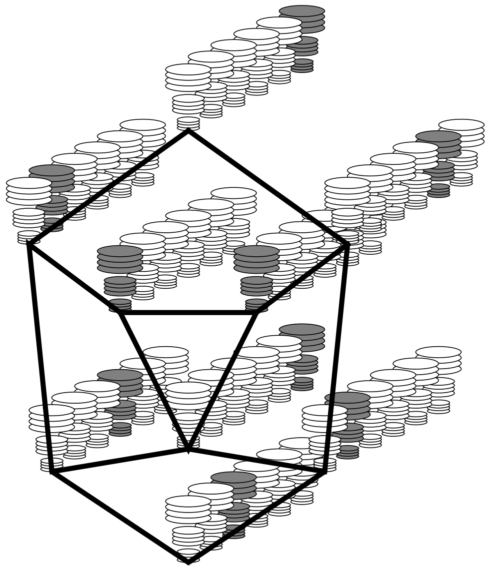


Figure 3: The Face Bunch Graph (FBG) serves as a representation of faces in general. It is designed to cover all possible variations in the appearance of faces. The FBG combines information from a number of face graphs. Its nodes are labeled with sets of jets, called bunches, and its edges are labeled with averages of distance vectors. During comparison to an image, the best fitting jet in each bunch, indicated by grey shading, is selected independently.

How large should an FBG be and which models should be included? This depends first of all on the variability of faces one wants to represent. If the faces are of many different races, facial expression, age, etc., the FBG must contain many different models to cope with this variability. The required FBG size also increases with the desired matching accuracy for finding the fiducial points in a new face. The accuracy can be estimated by matching the FBG to face images for which the fiducial points have been verified manually; cf. Section 3.2.3. FBG size does not depend on gallery size. In general, the models in the FBG should be as different as possible to reduce redundancy and maximize variability. Here we used FBGs with 30 models for the normalization stage and 70 models for the final graph extraction stage; cf. Section 2.3.4. These sizes seemed to give sufficient matching accuracy and reliability. We selected the models arbitrarily and did not optimize for maximal variability.

## 2.3 Generating Face Representations by Elastic Bunch Graph Matching

So far we have only described how individual faces and general knowledge about faces are represented by labeled graphs and the FBG, respectively. We are now going to explain how these graphs are generated. The simplest method is to do so manually. We have used this method to generate initial graphs for the system, one graph for each pose, together with pointers to indicate which pairs of nodes in graphs for different poses correspond to each other. Once the system has an FBG (possibly consisting of only one manually defined model), graphs for new images can be generated automatically by Elastic Bunch Graph Matching. Initially, when the FBG contains only a few faces, it is necessary to review and correct the resulting matches, but once the FBG is rich enough (approximately 70 graphs) one can rely on the matching and generate large galleries of model graphs automatically.

### 2.3.1 Manual Definition of Graphs

Manual definition of graphs is done in three steps. First, we mark a set of fiducial points for a given image. Most of these are positioned at well-defined features which are easy to locate, such as left and right pupil, the corners of the mouth, the tip of the nose, the top and bottom of the ears, the top of the head, and the tip of the chin. These points were selected to make manual positioning easy and reliable. Additional fiducial points are positioned at the center of gravity of certain easy-to-locate fiducial points. This allows automatic selection of fiducial points in regions where well-defined features are missing, e.g. at the cheeks or the forehead. Then, edges are drawn between fiducial points and edge labels are automatically computed as the differences between node positions. Finally, the Gabor wavelet transform provides the jets for the nodes.

In general, the set of fiducial points should cover the face evenly. But depending on the task, it may be appropriate to emphasize certain regions by additional nodes. For face finding, for example, we place more nodes on the outline, because with homogeneous background the contour is a good cue for finding faces. For face recognition, on the other hand, we place more nodes in the interior of the faces, because of its importance for recognition. A more systematic way of selecting nodes from a dense set is presented in (Krüger, 1997; Krüger et al., 1997). More nodes tend to yield better results, because more information is used, but this effect saturates if the nodes are too close and the corresponding Gabor coefficients become highly correlated due to overlap between the kernels. On the other hand, the computational effort increases linearly with the number of nodes. The optimal number of nodes will therefore be a compromise between recognition performance and speed.

### 2.3.2 The Graph Similarity Function

A key role in Elastic Bunch Graph Matching is played by a function evaluating the *graph similarity* between an image graph and the FBG of identical pose. It depends on the jet similarities and the distortion of the image grid relative to the FBG grid. For an image graph $\mathcal{G}^{\mathcal{I}}$ with nodes $n = 1, ..., N$ and edges $e = 1, ..., E$ and an FBG $\mathcal{B}$ with model graphs $m = 1, ..., M$ the similarity is defined as

$$\mathcal{S}_{\mathcal{B}}(\mathcal{G}^{\mathcal{I}}, \mathcal{B}) = \frac{1}{N} \sum_{n} \max_{m} \left( \mathcal{S}_{\phi}(\mathcal{J}_n^{\mathcal{I}}, \mathcal{J}_n^{\mathcal{B}m}) \right) - \frac{\lambda}{E} \sum_{e} \frac{(\Delta \vec{x}_e^{\mathcal{I}} - \Delta \vec{x}_e^{\mathcal{B}})^2}{(\Delta \vec{x}_e^{\mathcal{B}})^2}, \tag{9}$$

where $\lambda$ determines the relative importance of jets and metric structure. $\mathcal{J}_n$ are the jets at nodes $n$, and $\Delta \vec{x}_e$ are the distance vectors used as labels at edges $e$. Since the FBG provides several jets for each fiducial point, the best one is selected and used for comparison. These best fitting jets serve as *local experts* for the image face.

### 2.3.3 Matching Procedure

The goal of Elastic Bunch Graph Matching on a probe image is to find the fiducial points and thus to extract from the image a graph which maximizes the similarity with the FBG as defined in Eq. (9). In practice, one has to apply a heuristic algorithm to come close to the optimum within a reasonable time. We use a coarse to fine approach in which we introduce the degrees of freedom of the FBG progressively: translation, scale, aspect ratio, and finally local distortions. We similarly introduce phase information and increase the focus of displacement estimation: no phase, phase with focus 1, and then phase with focus 1 up to 5. The matching schedule described here assumes faces of known pose and approximately standard size, so that only one FBG is required. The more general case of varying size is sketched in the next section.

*Step 1: Find approximate face position.* Condense the FGB into an *average graph* by taking the average magnitudes of the jets in each bunch of the FBG (or, alternatively, select one arbitrary graph as a representative). Use this as a rigid model ($\lambda = \infty$) and evaluate its similarity at each location of a square lattice with a spacing of 4 pixels. At this step the similarity function $\mathcal{S}_a$ without phase is used instead of $\mathcal{S}_\phi$. Repeat the scanning around the best fitting position with a spacing of 1 pixel. The best fitting position finally serves as the starting point for the next step.

*Step 2:* *Refine position and size.* Now the FBG is used without averaging, varying it in position and size. Check the four different positions $(\pm 3, \pm 3)$ pixels displaced from the position found in Step 1, and at each position check two different sizes which have the same center position, a factor of 1.18 smaller or larger than the FBG average size. This is without effect on the metric similarity, since the vectors $\vec{x}_e^{\mathcal{B}}$ are transformed accordingly. We still keep $\lambda = \infty$. For each of these eight variations, the best fitting jet for each node is selected and its displacement according to Eq. (8) is computed. This is done with a focus of 1, i.e., the displacements may be of a magnitude up to eight pixels. The grids are then rescaled and repositioned to minimize the square sum over the displacements. Keep the best of the eight variations as the starting point for the next step.

*Step 3:* *Refine size and find aspect ratio.* A similar relaxation process as described for Step 2 is applied, but relaxing the $x$- and $y$-dimensions independently. In addition, the focus is increased successively from 1 to 5.

*Step 4:* *Local distortion.* In a pseudo-random sequence the position of each individual image node is varied to further increase the similarity to the FBG. Now the metric similarity is taken into account by setting $\lambda = 2$ and using the vectors $\vec{x}_e^{\mathcal{B}}$ as obtained in Step 3. In this step only those positions are considered for which the estimated displacement vector is small ($d < 1$, see Eq. (8)). For this local distortion the focus again increases from 1 to 5.

The resulting graph is called the *image graph* and is stored as a representation of the individual face of the image.

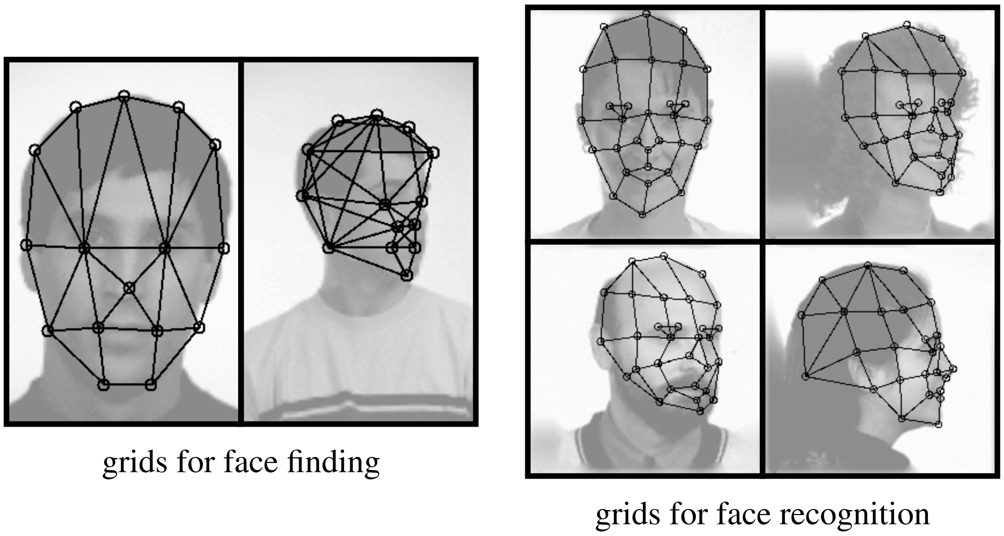


Figure 4: Object-adapted grids for different poses. The nodes are positioned automatically by elastic graph matching against the corresponding face bunch graphs. The two images on the left show originals with widely differing size and grids, as used for the normalization stage with many nodes on the outline for reliable face finding. The images on the right are already rescaled to normal size. Their grids have more nodes on the face, which is more appropriate for recognition (The grids used in Section 3.2 had about 14 additional nodes, which, for simplicity, are not shown here). One can see that, in general, the matching finds the fiducial points quite accurately. But mismatches occurred, for example, for the bearded man. The chin was not found accurately; the leftmost node and the node below it should be at the top and the bottom of the ear respectively.

### 2.3.4 Schedule of Graph Extraction

To minimize computing effort and to optimize reliability, we extract a face representation in two stages, each of which uses a matching procedure as described in the previous section. The first stage, called the *normalization stage* and described in greater detail in (Krüger et al., 1997), has the purpose of estimating the position and size of the face in the original image, so that the image can be scaled and cut to standard size. The second stage takes this image as input and extracts a precise image graph appropriate for face recognition purposes. The two stages differ in emphasis. The first one has to deal with greater uncertainty about size and position of the head and has to optimize the reliability with which it finds the face, but there is no need to find fiducial points with any precision or extract data important for face recognition. The second stage can start with little uncertainty about position and size of the head, but has to extract a detailed face graph with high precision.

In the experiments on the FERET database described below, original images had a format of 256×384 pixels, and the faces varied in size by a factor of three; see Figure 4. The poses were given and did not need to be determined. The normalization stage used three FBGs of appropriate pose which differed in face size. We somewhat arbitrarily picked approximately 30 images to form each FBG. More careful selection of images to cover a wider range of variations can only improve system performance. The grids used in the construction of the FBGs put little emphasis, i.e. few nodes, on the interior of the face and have fewer nodes than those used for the second stage; see Figure 4 for two examples. The smaller number of nodes speeds up the process of face finding. Using a matching scheme similar to the one described in Section 2.3.3, we match each of the three FBGs to the input image. We select the graph that matches best, cut a frame of appropriate size around it from the image and resize it to 128×128 pixels. The poses could be determined analogously (Krüger et al., 1997), but here the poses are assumed to be known. In our experiments, normalization took approximately 20 seconds on a SPARCstation 10-512 with a 50 MHz processor and identified face position and scale correctly in approximately 99% of the images.

The second stage uses the matching procedure exactly as described in Section 2.3.3, starting the match at standard size and position. The face bunch graphs used in this stage have more nodes, which we have placed in positions we believe are important for person identification, emphasizing the interior of the face. Each of the three principal poses (frontal, half-profile, and profile; left-facing poses are flipped to right-facing poses) is matched with a different grid structure and with a different FBG, formed by using 70 arbitrarily chosen images. This stage took approximately 10 seconds.

## 2.4 Recognition

After having extracted model graphs from the gallery images and image graphs from the probe images, recognition is possible with relatively little computational effort by comparing an image graph to all model graphs and picking the one with the highest similarity value. A comparison against a gallery of 250 individuals took slightly less than a second. The similarity function we use here for comparing graphs is an average over the similarities between pairs of corresponding jets. For image and model graphs referring to different pose, we compare jets according to the manually provided correspondences. If $\mathcal{G}^{\mathcal{I}}$ is the image graph, $\mathcal{G}^{\mathcal{M}}$ is the model graph, and node $n_{n'}$ in the model graph corresponds to node $n'$ in the image graph, we define *graph similarity* as:

$$\mathcal{S}_{\mathcal{G}}(\mathcal{G}^{\mathcal{I}}, \mathcal{G}^{\mathcal{M}}) = \frac{1}{N'} \sum_{n'} \mathcal{S}_a(\mathcal{J}^{\mathcal{I}}_{n'}, \mathcal{J}^{\mathcal{M}}_{n_{n'}}), \tag{10}$$

where the sum runs only over the $N'$ nodes in the image graph with a corresponding node in the model graph. We use the jet similarity function without phase here. It turned out to be more discriminative, possibly because it is more robust with respect to change in facial expression and other variations. Here we ignore the jet distortions created by rotation in depth, but will take up the subject in the discussion.

This graph similarity induces a ranking of the model graphs relative to an image graph. A person is recognized correctly if the correct model yields the highest graph similarity, i.e., if it is of rank one. A confidence criterion on how reliably a person is recognized can easily be derived from the statistics of the ranking (Lades et al., 1993). However, we have restricted our results to unqualified recognition rates, which already give an accurate impression of system performance.

# 3 Experiments

## 3.1 Databases

For the experiments we used image galleries taken from two different databases. Both of them explicitly distinguish different poses, and images are labeled with pose identity.

The first one is the ARPA/ARL FERET database provided by the US Army Research Laboratory. The poses are: frontal, quarter view, half-profile right or left (rotated by about 40-70°), and profile right or left; see Figure 5 for examples. We disregarded quarter view images, because there were only a few of them. For most faces there are two frontal views with different facial expression. Apart from a few exceptions, there are no disguises, variations in hair-style or in clothing. The background is always a homogeneous light or grey, except for smoothly varying shadows. The size of the faces varies by about a factor of three (but is constant for each individual, information which we could have used to improve recognition rates, but didn't). The format of the original images is $256 \times 384$ pixels.

The second database has been collected at the Institute for Neural Computation in Bochum and has been partly described in (Lades et al., 1993). The poses are: frontal, 11° and 22° rotated. The 11° and 22° angles have been estimated from the distance between the eyes (Thomas Maurer, personal communication, 1996). People were actually told to orient towards 15° and 30° marks on the wall, but these angles hold only for the gaze; the heads are usually rotated less. The 11° rotated faces are referred to as 15° rotated in (Lades et al., 1993). For all faces there are two frontal views, one neutral and one with a different facial expression. The latter includes a few cases where half of the face is occluded by hair or a hand. Two frontal views in this database generally differ more than those in the FERET database. All images were taken with the same set-up, so that faces varied in size only within the natural range. Our tests on this database allow direct comparison with the preceding system (Lades et al., 1993). The description of our algorithm in Section 2 referred to the FERET database. For the Bochum database we did not use the normalization stage, because faces varied only a little in size. For matching, the FBGs were comprised of all the available images except for the image onto which the FBG was matched.

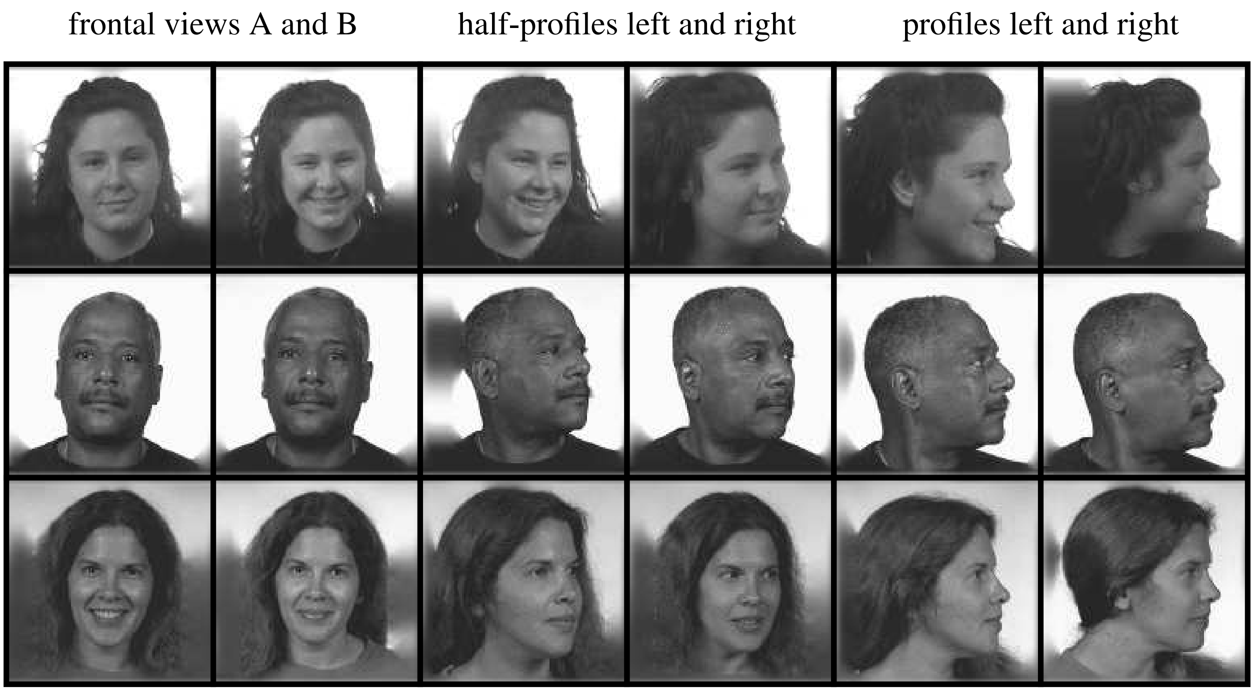


Figure 5: Sample faces from the ARPA/ARL FERET database: frontal views, half-profiles, and profiles. Pictures for left-facing poses are flipped around a vertical axis, and all images have been rescaled to standard size by our normalization stage (Section 2.3.4). Notice the large variation in the rotation angle of half-profiles and that some faces have no variation in facial expression in the two frontal views.

## 3.2 Results

### 3.2.1 FERET Database

We used various model and probe galleries with faces of different pose. Each model gallery contained 250 faces with just one image per person. We relied on the explicitly labeled pose identity instead of using our own pose recognition capability. Recognition results are shown in Table 1 (from Wiskott et al., 1997).

| Model gallery | Probe images | First rank # | First rank % | First 10 ranks # | First 10 ranks % |
|---|---|---|---|---|---|
| 250 fa | 250 fb | 245 | 98 | 248 | 99 |
| 250 hr | 181 hl | 103 | 57 | 147 | 81 |
| 250 pr | 250 pl | 210 | 84 | 236 | 94 |
| 249 fa + 1 fb | 171 hl + 79 hr | 44 | 18 | 111 | 44 |
| 171 hl + 79 hr | 249 fa + 1 fb | 42 | 17 | 95 | 38 |
| 170 hl + 80 hr | 217 pl + 33 pr | 22 | 9 | 67 | 27 |
| 217 pl + 33 pr | 170 hl + 80 hr | 31 | 12 | 80 | 32 |

Table 1: Recognition results for cross-runs between different galleries (f: frontal views; a, b: expression a and b; h: half-profiles; p: profiles; l, r: left and right). Each gallery contained only one image per person; the different compositions in the four bottom rows are due to the fact that not all poses were available for all people. The table shows how often the correct model was identified as rank one and how often it was among the first 10 (4%).

The recognition rate is very high for frontal against frontal images (first row). This is mainly due to the fact that in this database two frontal views show little variation, and any face recognition system should perform well under these circumstances, cf. Table 3. See the results on the Bochum database for a more challenging test.

Before comparing left against right poses, we flipped all left pose images over. Since human heads are to some degree bilaterally symmetric and since our present system performs poorly on such large rotations in depth (see below), we proceeded under the expectation that it would be easier to deal with differences due to facial asymmetry than with differences caused by substantial head rotation. This assumption is borne out at least by the high recognition rate of 84% for right profile against left profile (third row). The sharply reduced recognition rate of 57% (second row) when comparing left and right half-profiles could be due to inherent facial asymmetry, but the more likely reason is the poor control in rotation angle in the database — inspection of images shows that right and left rotation angles differ by up to 30°, cf. Figure 5.

When comparing half-profiles with either frontal views or profiles, another reduction in recognition rate is observed (although even a correct recognition rate of 10% out of a gallery of 250 is still high above chance level, which would be 0.4%!). The results are asymmetrical, performance being better if frontal or profile images serve as model gallery than if half-profiles are used. This is due to the fact that both frontal and profile poses are much more standardized than half-profiles, for which the angle varies approximately between 40° and 70°. We interpret this as being due to the fact that similarity is more sensitive to depth-rotation than to inter-individual face differences. Thus, when comparing frontal probe images to a half-profile gallery, a 40° half-profile gallery image of a wrong person is often favored over the correct gallery image if in the latter the head is rotated by a larger angle. A large number of such false positives degrades the correct-recognition rate considerably. In these experiments we also flipped all left pose images over, so that, to a large extent, the recognition was not only done across pose but also across mirror reflection.

### 3.2.2 Bochum Database

We used 108 neutral frontal views as a model gallery and the other images as probe galleries. For comparison we also give recognition rates obtained with the preceding system (Lades et al., 1993). Recognition results are shown in Table 2 (from Wiskott et al., 1997).

On this database, recognition rates for frontal views are lower than on the FERET database. This is due to the fact that the frontal views in the Bochum gallery differ in facial expression more than those in

| Model gallery | Probe images | Preceding system First rank # | % | This system First rank # | % | First 4 ranks # | % |
|---|---|---|---|---|---|---|---|
| 108 fa | 108 fb | 99 | 92 | 98 | 91 | 102 | 94 |
| 108 fa | 108 11° | 105 | 97 | 101 | 94 | 105 | 97 |
| 108 fa | 108 22° | 92 | 85 | 95 | 88 | 103 | 95 |

Table 2: Recognition results for cross-runs between different galleries (f: frontal views; a: neutral; b: different facial expression; 11°, 22°: rotated faces). Each gallery contained only one image per person. For matching, only face bunch graphs of frontal pose and 22° pose were used. Matching on 11° pose images was done with the frontal pose face bunch graph. The table shows how often the correct model was identified as rank one and how often it was among the first 4 (4%). The results for the preceding system have been obtained with the original software of (Lades et al., 1993). All results are from (Wiskott et al., 1997).

the FERET database. This is consistent with the results for 11°-rotated but neutral faces, which are higher than those for the frontal views. This also indicates that the variation due to facial expression is relatively large.

Comparison with the preceding system (Lades et al., 1993) shows that both systems perform equally well on the Bochum gallery. In comparing the results, two differences in the algorithms should be taken into account; see Section 4.1.1 for a discussion of the algorithmic differences between the two systems. Firstly, the preceding system used 70 nodes while our system used only 30 nodes for the Bochum galleries. We have noticed that taking more nodes leads to higher recognition rates. Secondly, the preceding system did not scale model grids to compensate for size variations. Since all images were taken from the same distance, faces had natural size variation relative to the fixed grid and size could then implicitly contribute to recognition. Though our system did not resize the images themselves, the grids were averaged in size to generate the face bunch graph and matched to the size of each face individually with no cost to the similarity function (although, the jets were not scaled). Thus our system yields the same recognition performance with less information.

The preceding system (Lades et al., 1993) was implemented on a system with 23 transputers. The Gabor wavelet transform required less than 7 seconds. Comparing one image with a gallery of 87 models required about 25 seconds. On a SPARCstation 10-512, a single comparison of an image graph with a model graph would probably require about 0.2 seconds. The system presented here requires less than 30 seconds to extract the image graph once and can then compare with about 300 model graphs per second. Thus, there is a significant increase in speed of our system over the preceding system for large galleries.

### 3.2.3 Matching Accuracy

We have introduced phase information to improve matching accuracy. Thomas Maurer (personal communication, 1996) has tested the accuracy on the Bochum database by matching a face bunch graph onto images for which all fiducial points were controlled manually. He always left the person on the image out of the face bunch graph, so that no information about that particular person could be used for matching. He ran the same algorithm with phase information and without phase information, i.e. all phases set to zero. Matching accuracy was calculated as the mean Euclidean distance between matching positions and manually controlled reference positions. It was 1.6 with and 5.2 pixels without phase, and the histograms had their maximum at 1 and 4 pixels distance, respectively. The images had a size of 128×128 pixels. Notice that since the reference positions were set manually, one cannot expect a matching accuracy much better than one pixel. This is because manual positioning focuses more on local high frequency information, while the matching system takes into account low frequencies as well. In addition, the manual positioning may be inaccurate itself. One can get an impression of the matching accuracy of the preceding system from Figure 6 in (Lades et al., 1993). A typical effect without phase is that a node is positioned at the wrong side of an edge, e.g. fifth node from the top in the rightmost column.

To investigate the importance of matching accuracy for recognition performance, Maurer has performed three different cross-runs of the 22° probe images against the neutral frontal view gallery. In the first run he used manually controlled node positions for the probe images, in the second run positions were obtained by matching with phase information, and in the third run without phase information. The frontal gallery was always the same. Numbers of correctly recognized faces in these three cases were 96 (89%) for manual positioning, 95 (88%) with phase, and 72 (67%) without phase. This shows that the matching accuracy with phase is sufficient, while using no phase would cause a significant degradation in recognition performance.

Notice that the preceding system (Lades et al., 1993) achieves high recognition rates without using phase information for the matching. Two reasons for this may be the larger number of nodes and the advantage of using different grid sizes if faces are of different but reliable size, as discussed above. Another reason is that phase information becomes more important as more degrees of freedom are introduced. Apart from local distortions, the preceding system only varied the location of individual graphs while our system also varied grid size, aspect ratio, and the identity of the local experts during the matching. With these additional degrees of freedom, the matching is more likely to fail without phase information, while the preceding algorithm was still robust.

Another study on face recognition was also based on face bunch graphs (including the correct face) and Gabor jets, but the matching algorithm was much simpler and constrained to a sparse grid of points in the image (Wiskott, 1999). Matching accuracy was therefore limited by the spacing of the grid points, which was 8 pixels in images of 128×128 pixels. Tests on the same 108 images of the Bochum database as used here confirmed that matching with phase (and recognition without phase) yields the highest recognition rates. However, it was surprising that for frontal views with different mimic expression (fb) and for 11° rotated faces such a simple matching algorithm achieved recognition rates of 92% and 94%, respectively, which is comparable to the performance of our system. It was only for the 22° rotated faces that the more sophisticated method presented here performed significantly better (88%) than the simple matching algorithm (81%).

# 4 Discussion

The system presented here is general and flexible. It is designed for an *in-class recognition* task, i.e. for recognizing members of a known class of objects. We have applied it to face recognition, but the system is in no way specialized in faces, and we assume that it can be directly applied to other in-class recognition tasks, such as recognizing individuals of a given animal species, given the same level of standardization of the images. In contrast to many neural network systems, no extensive training for new faces or new object classes is required. Only a moderate number of typical examples have to be inspected to build up a bunch graph, and individuals can then be recognized after storing a single image.

We tested the system with respect to rotation in depth and differences in facial expression. We did not investigate robustness to other variations, such as illumination changes or structured background. The performance is high on faces of the same pose. We also showed robustness against rotation in depth up to about 22°. For large rotation angles the performance degrades significantly.

## 4.1 Comparison to Other Systems

There is a considerable literature on face recognition, and many different techniques have been applied to the task; see (Samal & Iyengar, 1992; Valentin et al., 1994; Chellappa et al., 1995) for reviews. Here we relate our system to those of others in regard to conceptual and performance aspects.

### 4.1.1 Comparison to the Preceding System

We developed the system presented here based on (Lades et al., 1993), with several major modifications. We now utilize wavelet phase information for accurate node localization. Previously, node localization was rather imprecise. We have introduced the potential to specialize the system for specific object types and to handle different poses with the help of object-adapted grids. The face bunch graph is able to represent a wide variety of faces, which allows matching on face images of previously unseen individuals. These improvements make it possible to extract an image graph from a new face image in one matching process. Even if the person of the new image is not included in the FBG, the image graph reliably refers to the fiducial points. This considerably accelerates recognition from large databases, since for each probe image, correct node positions need to be searched only once instead of in each attempted match to a gallery image, as was previously necessary. The ability of the new system to refer to object-fixed fiducial points irrespective of pose represents an advantage in itself and is essential for some interesting graph operations; cf. Section 4.2. Computational efficiency, the ability to deal with different poses explicitly, and greater potential for further developments are the major advantages of the new system compared to the preceding one. We did not expect and experiments do not show an immediate improvement of recognition performance on faces of similar orientation.

#### 4.1.2 Recognizing Faces of the Same View

Some face recognition systems are based on user-defined face-specific features. Yuille (1991), for example, represented eyes by a circle within an almond-shape and defined an energy function to optimize a total of 9 model parameters for matching it to an image. The drawback of these systems is that the features as well as the procedures to extract them must be newly defined and programmed by the user for each object class, and the system has no means to adapt to samples for which the feature models fail. For example, the eye models mentioned above may fail for faces with sunglasses or have problems if the eyes are closed. In these cases the user has to design new features and new algorithms to extract them. With this type of approach, the system can never be weaned from designer intervention. Our system, in contrast, can be taught exceptional cases, such as sunglasses or beards, or entirely new object classes, by the presentation of examples and incorporation into bunch graphs.

An approach to face recognition also avoiding user-defined features is based on principal component analysis (PCA) (Sirovich & Kirby, 1987; Kirby & Sirovich, 1990; Turk & Pentland, 1991; O'Toole et al., 1993). In this approach, faces are first aligned with each other and then treated as high-dimensional pixel vectors from which eigenvectors, so-called eigenfaces, are computed, together with the corresponding eigenvalues. A probe face is decomposed with respect to these eigenvectors and is efficiently represented by a small number, say 30, of expansion coefficients. (The necessary image alignment can be done automatically within the PCA framework (Turk & Pentland, 1991; Moghaddam & Pentland, 1997)). PCA is optimal with respect to data compression and is successful for recognition purposes.

In its original, simple form, PCA treats an entire face as one vector, which causes two major problems. Firstly, because PCA is linear in the image space, it cannot deal well with variations in geometry. Consider, for example, two faces which have the mouth at a different height. Any linear combination of these two images can only generate a mouth at either height or two superimposed mouths but never a natural-looking mouth at an intermediate height. Linear combinations of images do not interpolate geometry. As a consequence, face images have to be aligned carefully before applying PCA or computing the expansion coefficients. Usually, face images are at least scaled, rotated, and shifted to align the eyes; sometimes also the aspect ratio is changed to also align the mouth. But then other facial features may still be misaligned. A solution to this problem is to factorize geometry and texture completely by warping face images to an average geometry. PCA is then applied to the warped face and to the geometrical features separately. Craw et al. (1995) have shown that this technique is advantageous for recognition. They used manually defined fiducial points for warping. Lanitis et al. (1995) apply a graph-matching algorithm similar to ours to find the fiducial points. Even more extreme in this sense are the systems (Beymer & Poggio, 1995; Vetter & Poggio, 1997; Vetter, 1998). They use an image-flow algorithm to match each pixel of a face image to a pixel in a different image. The warping is correspondingly accurate. These latter systems, however, require carefully taken images of high quality and are less robust against perturbations, such as occlusions or glasses.

A second problem of the original PCA approach is its sensitivity to occlusions or other localized perturbations, such as variations in hair style or facial hair. In a more localized feature representation, some regions can be explicitly treated as occluded, yielding good recognition results despite large occlusions (Wiskott & von der Malsburg, 1993). In the holistic representation of PCA, any local image perturbation will have an effect on all expansion coefficients and cannot be easily disregarded. One way this problem has been dealt with was by treating small image regions centered on fiducial points (eyes, nose, mouth) as additional pixel vectors from which to extract more features by PCA (Moghaddam & Pentland, 1997). A more systematic approach has been developed by Penev & Atick (1996). They explore spatial correlations within the set of eigenvectors found by PCA and generate a redundant set of localized kernels, one for each pixel location. Applying these kernels is called local feature analysis. A few of the local kernel responses are selected to generate a sparse representation.

The PCA approach with both extensions, a matching and warping stage at the beginning and PCA on localized regions or local feature analysis, becomes quite similar to our approach. A graph of fiducial points, labeled with example-derived feature vectors is matched to the image, and the optimally matching "grid" is used to extract a structural description from the image in the form of small sets of expansion coefficients in the PCA approach or jets in our approach. Recognition is then based on this. A remaining difference between the approaches lies in the nature of the underlying feature types: principal components statistically derived from a specific set of images of an object class, or Gabor-wavelets, which can be statistically derived from a more general set of natural images, at least qualitatively (Olshausen & Field, 1996; Bell & Sejnowski, 1997). It remains to be seen which of the approaches has the greater potential for development.

It is worth considering the system by Lanitis et al. (1995) in more detail because of its close relationship to our system. Both systems apply a graph-matching process for finding fiducial points and extract local features for recognition. The system by Lanitis et al. (1995) in addition warps the face to average geometry and applies PCA to it. There are two differences we want to point out. Firstly, the matching process differs in the way in which distortions are treated. Our system assumes a simple spring model, which introduces a large number of degrees of freedom and also includes distortions which are unrealistic, cf. mismatches in Figure 4. Lanitis et al. (1995), on the other hand, use PCA also to analyze distortion patterns of sample face images, the first eigenvectors providing a relatively small set of plausible geometrical distortions. Allowing the matching using only these distortions significantly reduces the number of degrees of freedom and leads to more reliable matchings in this respect. In addition, information about pose and shape can be inferred more easily and used for recognition. Secondly, the local features used in their system are relatively simple compared to our bunches of jets. The features are local grey value profiles along a line described by a few parameters. We assume that a combination of these two systems, using few geometrical distortion patterns and bunches of jets as local features, would improve matching performance compared to either system.

#### 4.1.3 Performance Comparison on the FERET Database

To obtain a meaningful performance comparison between different face recognition systems, the Army Research Laboratory has established a database of face images (ARPA/ARL FERET database) and compared our and several other systems in a blind test. Official results have been published in (Phillips & Rauss, 1997; Phillips et al., 1998). Here we summarize results which other groups have reported for their systems tested on the FERET database. The recognition rates are given in Table 3. It should be noted that we could not find results for the face recognition system developed by Atick et al. (1995) and Penev & Atick (1996). Their system performed well in one of the official tests (Phillips et al., 1998).

Gordon (1995) has developed a system which automatically selects regions around left eye, right eye, nose, and mouth for frontal views and a region covering the profile for profile views. The faces are then normalized for scale and rotation. The recognition is based on normalized cross-correlation of these five regions compared to reference models. Results are given for the fully automatic system, also for frontal views only, and for a system in which the normalization points, i.e. pupil centers, nose tip, and chin tip, are selected by hand. For the combined gallery (fa + pl), there is a great difference between the performance of the fully automatic system and that with manually located normalization points. This indicates that the automatic location of the normalization points is the main weakness of this system.

Gutta et al. (1995) have collected the images for the FERET database. They have tested the performance of a standard RBF (radial basis function) network and a system based on geometrical relationships between facial features, such as eyes, nose, mouth, etc. The performance of the latter was very low and is not summarized in Table 3.

Moghaddam & Pentland (1994) have presented results based on the PCA approach discussed in the previous section. A front-end system normalized the faces with respect to translation, scale, lighting, contrast, and slight rotations in the image plane. The face images were then decomposed with respect to the first eigenvectors, and the corresponding coefficients were used for face representation and comparison. The performance on frontal views, which are highly standardized, was high and comparable to that of our system, but the performance on half-profiles and profiles was relatively low. That indicates that the global PCA-approach is more sensitive to rotation in depth.

| Reference | Method | Model gallery | Probe images | First rank % |
|---|---|---|---|---|
| Gordon (1995) | normalized cross-correlation on different regions in a face | | | |
| | manually located normalization points | 202 fa + pl | 202 fb + pr | 96 |
| | fully automatic system | 194 fa + pl | 194 fb + pr | 72 |
| | | 194 fa | 194 fb | 62 |
| Gutta et al. (1995) | radial basis function network | | | |
| | on automatically segmented face images | 100 fa | 100 fb | 83 |
| Moghaddam & Pentland (1994) | principal component analysis on the whole face | | | |
| | fully automatic system | 150 fa | 150 fb | 99 |
| | | 150 hr | 150 hl | 38 |
| | | 150 pr | 150 pl | 32 |
| Phillips & Vardi (1995) | trained matching pursuit filters for different regions in a face | | | |
| | manually located feature points | 172 fa | 172 fb | 98 |
| | fully automatic system | 172 fa | 172 fb | 97 |
| Phillips (1996) | manually located feature points | 311 fa | 311 fb | 95 |
| | fully automatic system | 311 fa | 311 fb | 95 |
| Wiskott et al. | Gabor wavelets, labeled graphs and elastic bunch graph matching | | | |
| (1995) | fully automatic system | 300 fa | 300 fb | 97 |
| (1997) | fully automatic system | 250 fa | 250 fb | 98 |
| | | 250 hr | 181 hl | 57 |
| | | 250 pr | 250 pl | 84 |

Table 3: Methods and performances of the different systems discussed. For our system we repeat results from two different publications for comparison. For some systems it was not reported whether fa or fb was used for the model gallery; we consistently indicate the frontal model gallery by fa. When comparing the results, notice that the first rank recognition rates depend on gallery size. Only Moghaddam & Pentland (1994) have reported results on half-profiles and on profiles; none of the groups has reported results across different poses, such as half-profile probe images against profile gallery.

Phillips & Vardi (1995) and Phillips (1996) have trained two sets of matching pursuit filters for the tasks of face location and identification. The filters focus on different regions: the interior of the face, the eyes, and the nose for location; tip of the nose, bridge of the nose, left eye, right eye, and interior of the face for identification. The performance is high and comparable to that of our system. The small performance difference between the fully automatic system and the identification module indicates that the location module works reliably.

For none of the systems were results across different poses reported. In the next section we will therefore summarize systems which have been tested for rotation in depth on different databases.

### 4.1.4 Recognizing Faces Rotated in Depth

While there is a considerable literature on face recognition in the same pose, there are few systems which deal with large rotation in depth. It is difficult to compare these systems in terms of performance, because they have been tested on different galleries. Furthermore, the recognition rates are a result of complete systems and do not necessarily reflect the usefulness of a particular method to compensate for rotation in depth. However, we think it may still be useful to give an overview and to briefly discuss the different approaches. Results are given in Table 4.

| Reference | Database and Method | Model gallery # | Model gallery angle(s) | Probe images # | Probe images angle(s) | First rank % |
|---|---|---|---|---|---|---|
| Moghaddam & Pentland (1997) | PCA approach separately for different views, no specific transformation | | | | | |
| | interpolation performance ±23° | 21 | ±90°, ±45°, 0° | 21 | ±68°, ±23° | 90 |
| | extrapolation performance ±23° | 21 | e.g. −90°, ..., +45° | 21 | e.g. +68° | 83 |
| | extrapolation performance ±45° | 21 | e.g. −90°, ..., +45° | 21 | e.g. +90° | 50 |
| Wiskott et al. (1997) | Gabor jets, no specific transformation | | | | | |
| | Bochum database | 108 | 0° | 108 | 11° | 94 |
| | | 108 | 0° | 108 | 22° | 88 |
| | FERET database | 250 | 0° | 250 | 45° | 18 |
| | | 250 | 45° | 250 | 0° | 17 |
| | | 250 | 45° | 250 | 90° | 9 |
| | | 250 | 90° | 250 | 45° | 12 |
| Maurer & von der Malsburg (1995) | Gabor jets, learned normal vectors for geometrical rotation transformation | | | | | |
| | Bochum database, no transformation | 110 | 0° | 110 | 22° | 88 |
| | transforming 22° to 0° | 110 | 0° | 110 | 22° | 96 |
| | FERET database, no transformation | 90 | 0° | 90 | 45° | 36 |
| | transforming 45° to 0° | 90 | 0° | 90 | 45° | 50 |
| | transforming 0° to 45° | 90 | 0° | 90 | 45° | 53 |
| Beymer & Poggio (1995) | well-controlled gallery, little hair information | | | | | |
| | warping between different views | 62 | (±)20° | 620 | range ±40° | 82 |
| | linear decomposition, no shape info. | 62 | (±)20° | 620 | range ±40° | 70 |
| Vetter (1998) | images rendered from 3D face data, no hair information | | | | | |
| | mapping onto 3D-model | 100 | 0° | 100 | 24° | 100 |
| | linear decomposition and synthesis | 100 | 0° | 100 | 24° | 100 |
| | linear decomposition on four subregions | 100 | 0° | 100 | 24° | 100 |

Table 4: Methods and performances of the different systems discussed. Our results are repeated for comparison. When comparing the results, notice that the first rank recognition rates depend on gallery size as well as on the quality of the databases.

The system by Moghaddam & Pentland (1997) simply applies several recognition subsystems in parallel, each of which is specialized to one view and is based on the PCA approach described above for recognition of the same views. The subsystem which is specialized for a view closest to the view of the probe image is usually best suited to explain the image data in terms of its eigenvectors. It therefore can be selected automatically to perform the recognition. This system has been tested on galleries of 21 persons in different views. The results listed in Table 4 are averages over several different combinations of training and testing views. The recognition rates are an example of how well a system can perform if it does not compensate for effects of rotation in depth but relies only on the robustness of the subsystem which is closest to the view of the probe image. Our basic system compensates for rotation in depth only in that matching is done with a bunch graph of the new view and correspondences are defined between fiducial points of the new view and fiducial points of the standard view for which the model graphs are available. Thus, corresponding jets are compared across different views, but the jets are not modified in any way to compensate for the effects of rotation in depth.

There are at least three different approaches to compensating for the effects of rotation in depth more explicitly: transforming feature vectors, warping images of faces, and linear decomposition and synthesis of faces in different views. Let us first consider transforming feature vectors. As an extension to our system, Maurer & von der Malsburg (1995) have applied linear vector transformations to the jets to compensate for the effect of rotation in depth. The assumption was that faces can be locally treated as plane surfaces and that the texture transforms accordingly. Since the total rotation of the faces is known, only the normal of the surface at each node has to be estimated, which is done on a training set of faces available in both views. This results in a significant improvement. Notice that transformations of feature vectors can only be an approximation to the true transformations of images. This is due to the fixed and limited support of the kernels which are used to extract the features. For instance, a circular region on a plane becomes an ellipse if the plane is tilted. Feature vectors based on kernels with circular support can only represent a circular region. If this circular region needs to be transformed into an elliptic region, some information is lost or incorrect information is added to obtain a circular region again. An advantage of this method is that transformations can be performed without reference to the original image.

More accurate results can be obtained if the grey-value distributions of faces are warped from one view to another view directly on a pixel level. Vetter (1998) has done this by means of a 3D-model onto which the texture of a face is projected and from which it is then back-projected onto the image plane in a different view. Beymer & Poggio (1995) have used a warping transformation derived from sequences of rotating sample faces. Another interesting approach is the concept of linear object classes (Vetter & Poggio, 1997). It is assumed that objects in one view can be linearly decomposed with respect to images of a set of prototype objects of the same view. When images of these prototypes are available in another view, the object can be linearly synthesized in that view with the same coefficients as used for the decomposition. Vetter (1998) tested this method on images rendered from 3D face data. Shape and texture were processed separately, the texture being processed as a whole or broken down into four local regions. Recognition was based on a simple similarity measure, e.g. Euclidean distance, applied directly to the image grey values. The recognition rates of 100% for this and the warping method described above are remarkable. Although, it has to be taken into account that the images were rendered from 3D face representations and that the galleries were correspondingly perfect. Beymer & Poggio (1995) also used this method, but they did the decomposition with respect to eigenfaces and did not use shape information. Their model gallery included 20° rotated faces plus the mirror-reflected images, and they tested on probe faces randomly drawn from a range of approximately ±40° rotation angle. They also considered rotation around horizontal axes. This system as well as the one by Vetter (1998) used an image-flow algorithm to find correspondences between different faces. We have tested a similar method for our system on the FERET database. For a half-profile face image we used a half-profile face bunch graph to generate a phantom face (Wiskott, 1997), which was then transformed into a frontal pose by using corresponding jets of the same fiducial points and individuals from a frontal face bunch graph. The idea was that if, for instance, the noses of two persons look similar in one pose, they would look similar in another pose as well. The results were disappointing and are not reported in Table 4. It is surprising that these three systems, which are based on similar ideas, perform so differently. A possible reason might be the different quality of the databases, which was perfect for the system by Vetter (1998) and worst for ours.

Each of these three approaches (transforming feature vectors, warping images of faces, and the concept of linear object classes) has its own advantages and drawbacks, and none is clearly superior to the others. Warping can potentially deal well with new types of faces not seen before, e.g. of new race or age, but it cannot be applied to transform between other variations, e.g. in illumination. Linear decomposition, on the other hand, can be applied to different types of variations, but it does probably not extrapolate well to new types of faces. Transforming features may deal well with different kinds of variations as well as new types of faces, but in its current formulation it is limited because of the fixed and limited kernels. The decision for one of the approaches will also depend on how well it integrates into a particular recognition system.

In the following section we discuss some methods which can potentially show or have been shown to further improve the performance of our system; see also (Okada et al., 1998).

### 4.2 Further Developments

The current system can be improved in many respects. In Section 4.1 we already argued that the simple spring model used here for the grid has too many degrees of freedom, which could be considerably reduced by using only a small number of typical distortions found by PCA on manually controlled grids (Lanitis et al., 1995). This would probably improve matching accuracy further and would provide more precise geometrical information which could be used to increase recognition performance. However, the matching precision achieved in our system, as compared to the preceding version (Lades et al., 1993), is already sufficient to apply specific methods which require reliable fiducial points, for instance when the issue is learning about local object properties. One such local property is differential degree of robustness against disturbances. In an extension of the basic system presented here, Krüger (1997) and Krüger et al. (1997) have developed a method for learning weights emphasizing the more discriminative nodes. On model galleries of size 130–150 and probe images of different pose, the first rank recognition rates have been improved by an average of 6%, from 25% without to 31% with weights. As mentioned in Section 4.1.4, another extension of our system also requiring reliable fiducial points has been developed by Maurer & von der Malsburg (1995) to compensate for rotation in depth.

In (Wiskott, 1997) the bunch graph technique has been used to fairly reliably determine facial attributes from single images, such as sex or the presence of glasses or a beard. If this technique was developed to extract independent and stable personal attributes, such as age, race, or sex, recognition from large databases could be improved and speeded up considerably by preselecting corresponding sectors of the database.

We did some preliminary experiments on images with structured background and got encouraging results. In this case more nodes in the interior of the faces were used. However, a more principled method should be employed. One could use only Gabor kernels lying within the face (Würtz, 1997) or, alternatively, transform jets from contour nodes such that the influence of background structure is suppressed (Pötzsch et al., 1996). Robustness with respect to illumination variations also has to be investigated.

The manual definition of appropriate grid structures and the semi-autonomous process of bunch graph acquisition will have to be replaced by a fully autonomous process. Automatic reduction of an FBG to its essential jets in the bunches has been demonstrated in (Krüger et al., 1997). The creation of a new bunch graph is most easily based on image sequences, which contain many cues for grouping, segmentation, and detecting correspondences. This has been demonstrated for individual graphs in (Maurer & von der Malsburg, 1996; Krüger et al., 1998).

In the current system, one recognition against a gallery of several hundred models takes approximately 30 seconds on a SPARCstation 10-512. This is too slow for most applications. However, there are many possibilities to optimize the system with respect to speed: reducing the number of Gabor-kernels, reducing the number of steps in the matching process, using fewer nodes, etc. The system described in (Lades et al., 1993), which is computationally even more expensive, has been optimized in this way and has been successfully turned into a commercial product for access control (Konen & Schulze-Krüger, 1995), which runs on a standard PC. Our group is also currently working on a real-time face-tracking system based on the matching process presented here. For this project a high performance parallel processor system will be employed.

## Acknowledgements

We wish to thank Irving Biederman, Ladan Shams, Michael Lyons, and Thomas Maurer for very fruitful discussions and their help in evaluating the performance of the system on the FERET database. Many thanks go to Thomas Maurer also for reviewing and optimizing the code and performing tests on the Bochum gallery. Jan Vorbrüggen performed the tests for the preceding system, which we used for comparison. We acknowledge helpful comments on the manuscript by Jonathon Phillips and Marni Stewart Bartlett. For the experiments we have used the FERET database of facial images collected under the ARPA/ARL FERET program and the Bochum gallery collected at the Institute for Neural Computation, Ruhr-University Bochum.

## References

Atick, J. J., Griffin, P., and Redlich, A. N. (1995). Face-recognition from live video for real-world applications — now. *Advanced Imaging*, 10(5):58–62. 16

Bell, A. J. and Sejnowski, T. J. (1997). The independent components of natural scenes are edge filters. *Vision Research*, 37(23):3327–3338. 4, 16

Beymer, D. (1994). Face recognition under varying pose. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 756–761, Seattle, WA. 7

Beymer, D. and Poggio, T. (1995). Face recognition from one example view. In *Proc. 5th Int'l Conf. on Computer Vision*, pages 500–507, Cambridge, MA. IEEE Comput. Soc. Press. 15, 18, 19

Chellappa, R., Wilson, C. L., and Sirohey, S. (1995). Human and machine recognition of faces: A survey. *Proc. of the IEEE*, 83(5):705–740. 14

Craw, I., Costen, N., Kato, T., Robertson, G., and Akamatsu, S. (1995). Automatic face recognition: Combining configuration and texture. In Bichsel, M., editor, *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition, IWAFGR'95, Zurich*, pages 53–58. MultiMedia Laboratory, University of Zurich. 15

Daugman, J. G. (1988). Complete discrete 2-D Gabor transform by neural networks for image analysis and compression. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 36(7):1169–1179. 2

DeValois, R. L. and DeValois, K. K. (1988). *Spatial Vision*. Oxford Press. 4

Fleet, D. J. and Jepson, A. D. (1990). Computation of component image velocity from local phase information. *Int'l J. of Computer Vision*, 5(1):77–104. 4

Gordon, G. G. (1995). Face recognition from frontal and profile views. In Bichsel, M., editor, *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition, IWAFGR'95, Zurich*, pages 47–52. MultiMedia Laboratory, University of Zurich. 16, 17

Gutta, S., Huang, J., Singh, D., Shah, I., Takacs, B., and Wechsler, H. (1995). Benchmark studies on face recognition. In Bichsel, M., editor, *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition, IWAFGR'95, Zurich*, pages 227–231. MultiMedia Laboratory, University of Zurich. 16, 17

Jones, J. P. and Palmer, L. A. (1987). An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. *J. of Neurophysiology*, 58:1233–1258. 4

Kirby, M. and Sirovich, L. (1990). Application of the Karhunen-Loève procedure for the characterization of human faces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(1):103–108. 15

Konen, W. and Schulze-Krüger, E. (1995). ZN-Face: A system for access control using automated face recognition. In Bichsel, M., editor, *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition, IWAFGR'95, Zurich*, pages 18–23. MultiMedia Laboratory, University of Zurich. 20

Krüger, N. (1997). An algorithm for the learning of weights in discrimination functions using a priori constraints. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):764–768. 8, 20

Krüger, N., Maël, E., Pagel, M., and von der Malsburg, C. (1998). Autonomous learning of object representations utilizing self-controlled movements. In *Proc. of Neural Networks in Applications, NN'98, Magdeburg, Germany*, pages 25–29. 2, 20

Krüger, N., Pötzsch, M., and von der Malsburg, C. (1997). Determination of face position and pose with a learned representation based on labelled graphs. *Image and Vision Computing*, 15:665–673. 8, 10, 20

Lades, M., Vorbrüggen, J. C., Buhmann, J., Lange, J., von der Malsburg, C., Würtz, R. P., and Konen, W. (1993). Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans. on Computers*, 42(3):300–311. 4, 10, 11, 12, 13, 14, 20

Lanitis, A., Taylor, C. J., and Cootes, T. F. (1995). An automatic face identification system using flexible appearance models. *Image and Vision Computing*, 13(5):393–401. 15, 16, 20

Maurer, T. and von der Malsburg, C. (1995). Linear feature transformations to recognize faces rotated in depth. In *Proc. Int'l Conf. on Artificial Neural Networks, ICANN'95, Paris*, pages 353–358, Paris. EC2 & Cie. 18, 19, 20

Maurer, T. and von der Malsburg, C. (1996). Tracking and learning graphs and pose on image sequences of faces. In *Proc. 2nd Int'l Conf. on Automatic Face- and Gesture-Recognition*, pages 176–181, Los Alamitos, CA. IEEE Comp. Soc. Press. 2, 20

Moghaddam, B. and Pentland, A. (1994). Face recognition using view-based and modular eigenspaces. In *Proc. SPIE Conf. on Automatic Systems for the Identification and Inspection of Humans*, volume SPIE 2277, pages 12–21. 16, 17

Moghaddam, B. and Pentland, A. P. (1997). Probabilistic visual learning for object representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):696–710. 15, 18

Okada, K., Steffens, J., Maurer, T., Hong, H., Elagin, E., Neven, H., and von der Malsburg, C. (1998). The Bochum/USC face recognition system and how it fared in the FERET phase III test. In Wechsler, H. et al., editors, *Face Recognition: From Theory to Applications*. Springer-Verlag. 20

Olshausen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609. 4, 16

O'Toole, A. J., Abdi, H., Deffenbacher, K. A., and Valentin, D. (1993). Low-dimensional representation of faces in higher dimensions of the face space. *J. of the Optical Society of America A*, 10(3):405–411. 15

Penev, P. S. and Atick, J. J. (1996). Local feature analysis: A general statistical theory for object representation. *Network: Computation in Neural Systems*, 7(3):477–500. 15, 16

Phillips, P. J. (1996). *Representation and Registration in Face Recognition and Medical Imaging*. PhD thesis, RUTCOR, Rutgers University. 17, 18

Phillips, P. J. and Rauss, P. J. (1997). Face recognition technology (FERET program). In *Proc. Office of National Drug Control Policy*. (in press). 16

Phillips, P. J. and Vardi, Y. (1995). Data driven methods in face recognition. In Bichsel, M., editor, *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition, IWAFGR'95, Zurich*, pages 65–70. MultiMedia Laboratory, University of Zurich. 16, 17

Phillips, P. J., Wechsler, H., Huang, J., and Rauss, P. (1998). The FERET database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing*, 16(5):295–306. 16

Pollen, D. A. and Ronner, S. F. (1981). Phase relationship between adjacent simple cells in the visual cortex. *Science*, 212:1409–1411. 4

PÖTZSCH, M., KRÜGER, N., AND VON DER MALSBURG, C. (1996). Improving object recognition by transforming Gabor filter responses. *Network: Computation in Neural Systems*, 7(2):341–347. 4, 20

SAMAL, A. AND IYENGAR, P. A. (1992). Automatic recognition and analysis of human faces and facial expressions: A survey. *Pattern Recognition*, 25(1):65–77. 14

SIROVICH, L. AND KIRBY, M. (1987). Low-dimensional procedure for the characterization of human faces. *J. of the Optical Society of America A*, 4(3):519–524. 15

THEIMER, W. M. AND MALLOT, H. A. (1994). Phase-based binocular vergence control and depth reconstruction using active vision. *CVGIP: Image Understanding*, 60(3):343–358. 4

TURK, M. AND PENTLAND, A. (1991). Eigenfaces for recognition. *J. of Cognitive Neuroscience*, 3(1):71–86. 15

VALENTIN, D., ABDI, H., O'TOOLE, A. J., AND COTTRELL, G. W. (1994). Connectionist models of face processing: A survey. *Pattern Recognition*, 27(9):1209–1230. 14

VETTER, T. (1998). Synthesis of novel views from a single face image. *Int'l J. of Computer Vision*, 28(2):103–116. 15, 18, 19

VETTER, T. AND POGGIO, T. (1997). Linear object classes and image synthesis from a single example image. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):733–741. 15, 19

WISKOTT, L. (1997). Phantom faces for face analysis. *Pattern Recognition*, 30(6):837–846. 19, 20

WISKOTT, L. (1999). The role of topographical constraints in face recognition. *Pattern Recognition Letters*, 20(1):89–96. 14

WISKOTT, L., FELLOUS, J.-M., KRÜGER, N., AND VON DER MALSBURG, C. (1995). Face recognition and gender determination. In BICHSEL, M., editor, *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition, IWAFGR'95, Zurich*, pages 92–97. MultiMedia Laboratory, University of Zurich.

WISKOTT, L., FELLOUS, J.-M., KRÜGER, N., AND VON DER MALSBURG, C. (1997). Face recognition by elastic bunch graph matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):775–779. 2, 12, 13, 18

WISKOTT, L. AND VON DER MALSBURG, C. (1993). A neural system for the recognition of partially occluded objects in cluttered scenes. *Int'l J. of Pattern Recognition and Artificial Intelligence*, 7(4):935–948. 15

WÜRTZ, R. P. (1997). Object recognition robust under translations, deformations, and changes in background. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):769–775. 20

YUILLE, A. L. (1991). Deformable templates for face recognition. *J. of Cognitive Neuroscience*, 3(1):59–70. 15