ELSEVIER

Image and Vision Computing 15 (1997) 665–673

Determination of face position and pose with a learned representation based on labelled graphs[1]

N. Krüger[a,*], M. Pötzsch[a], C. von der Malsburg[a,b]

[a]Ruhr-Universität Bochum, Institut für Neuroinformatik, Universitätstrasse 150, D-44780 Bochum, Germany
[b]Department of Computer Science and Section for Neurobiology, University of Southern California, Los Angeles, CA 90089-2520, USA

Abstract

We present a new system for the automatic determination of the position, size and pose of the head of a human figure in a camera image. The system is an extension of the well-known face recognition system [15] to pose estimation. The pose estimation system is characterized by a certain reliability and speed. We improve this performance and speed with the help of statistical estimation methods. In order to make these applicable, we reduce the originally very high dimensionality of our system with the help of a number of *a priori* principles. We discuss a possible extension of the learning algorithm aiming an autonomous object recognition system at the end of the paper. © 1997 Elsevier Science B.V.

Keywords: Elastic graph matching; A priori constraints; Pose estimation; Face recognition

1. Introduction

In this paper we deal with two problems. Firstly, we describe a pose estimation algorithm based on Elastic Graph Matching (EGM) [11,15]. The algorithm is an extension of the face representation introduced in [15] to the problem of pose estimation, in [15] the poses of faces is assumed to be known. Secondly, we improve the performance and speed of the pose estimation algorithm by learning. This learning algorithm can be seen as an intermediate step towards the construction of an autonomous object recognition system that is based on examples and not on manually constructed world knowledge. At present, our mechanisms for comparing different poses are still in part constructed manually.

As basic local image features we use Gabor-based wavelets. As others before us, we treat the set of wavelets centered on one image point as a unit which we call a 'jet'. Like many other object recognition systems (e.g., [10,7]), ours is based on object models (or rather, models for two-dimensional aspects of objects as they appear in the image). In our hands. aspect models have the form of graphs, the nodes of which are labelled with jets or bunches of jets and the links of which are labelled with distance vectors between nodes (for examples see Fig. 1(b) and (c)). This representation of objects applied to faces combined with EGM allows us to determine the pose and position of faces.

Starting from this extension of the original face recognition system described in [15] to pose estimation we optimize the representation of faces by statistical methods. We want to achieve an optimal but sparse representation of faces of different sizes and poses. The free parameters we have to determine in our model are the nodes of the graphs and the jets included in the bunches used to represent a face. The computational time of the pose estimation system depends linearly on the number of nodes and the number of jets in a bunch. We have formulated three a priori principles. These principles guide the selection of nodes and jets included in a bunch and allow an effective search in the huge combinatorial space of subgrids and subsets of jet bunches. The principles refer to the selection of landmarks on a face, starting with an (as yet hand-crafted) comprehensive set of landmarks, and about the selection of jets from sample images, to form bunches of jets for the landmarks. Our principles are

P0. (Locality): Features (jets, nodes) referring to different landmarks are treated as independent.

[*] Corresponding author. Tel.: +49 234 700 7988; Email: nkrueger@neuroinformatik.ruhr-uni-bochum.de.

[1] This research is supported by EC grant ERBCHRX-CT-930097, an HFSPO grant and US Army Research Lab grant (01/93/K-0109).

PII S0262-8856(97)00012-7

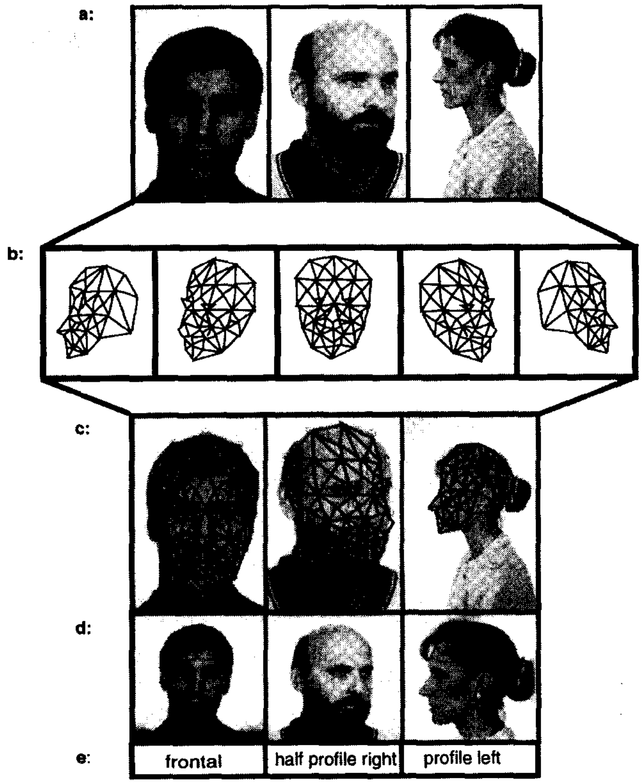

Fig. 1. The head finding and pose estimation system. (a) Input pictures with faces of different size and pose. (b) Graphs representing the five poses: profile left, half profile left, frontal, half profile right and profile right. (c) Input picture with the grid achieving the highest graph similarity S^G. (d) and (e) The output of the system. The face normalized according to scale and the pose estimated by the system.

P1. (Maximal Discrimination): Features varying little within classes (poses) and varying much between classes are preferred.

P2. (Minimal Redundancy): Features should be selected for minimal redundancy of information.

Principles P0, P1 and P2 are applied in the form of several special formalizations in our algorithm. We distinguish between derived principles and formalized principles. A derived principle is a conceptual adaptation of P1 or P2 to a specific task, a formalized principle is a particular mathematical expression into which a derived principle is cast. With this hierarchy of abstraction we would like to stress the general applicability of our a priori principles to representations based on labelled graphs, independent of the features used as labels and of details of the matching process.

Given a set of training images together with manually provided ground-truth as to the correct position of landmarks, each principle on its own would deterministically select a subset of landmarks and subsets of sample jets. However, these selections would differ for different principles and arbitration between them is required. This arbitration depends on relative weights given to the selection principles. We treat these relative weights and the number of features used during matching as free parameters and optimize them with respect to the overall performance of the system. These parameters constitute the total search space of our system of just five dimensions, a space we can afford to search exhaustively.

2. The algorithm for pose estimation

In this section we describe our representation of faces of different size and pose with bunch graphs and our matching algorithm based on this representation.

2.1. Representation of objects (faces)

As models for object aspects (i.e., for those regions of an image that are occupied by objects, in our case the faces and heads of people) our system employs labelled graphs. For estimating position, size and pose of a human head, the system uses a collection of such graphs (see Fig. 1(b)). The edges of graphs are labelled with distance vectors between node positions. Nodes are labelled with image information referring to landmarks, local areas on a head or face. such as the tip of the nose or the left eye. These labels are bunches of jets, each derived from the image of a different person, a bunch thus covering a variety of forms a single landmark may take. We call this structure a *bunch graph*, an idea first introduced in [15] (where, however, the term *general face knowledge* was used). The total model for heads we call a *collection of bunch graphs*.

Jets are derived from a set of linear filter operations in the form of convolutions of the image $I(\vec{x})$ with a set of Gabor wavelets[2] $\psi_{\vec{k}}$ (cf. [3]), whose wavelength and orientation are parameterized by $\vec{k}$. The $\psi_{\vec{k}}$ take the form of plane waves restricted by Gaussian envelopes function (see Fig. 2(a)). A jet is formed by the set of $5 \cdot 8$ complex values (corresponding to 5 frequencies and 8 orientations) rendered by all wavelets centered at a given position of the image (see Fig. 2(b)) [11]. Due to the spatial extent of the wavelets, jets describe a local area around their position. A bunch $\mathcal{B}$ of jets taken at the same landmark (that is, at corresponding positions) of different faces forms a generalized representation of this landmark.

We create the bunch graph for a given pose by placing an appropriate graph over about 80 images of faces in that pose and extracting the jets from the correctly positioned nodes in the sample images. Samples are shown in Fig. 1(a). In these images, pose-specific graphs were first positioned by our matching algorithm and individual nodes were then corrected manually if necessary, which was the case in approxi-

[2] We use the term 'Gabor wavelets' or 'set of Gabor wavelets' as a shorthand for 'a set of wavelets derived from a Gabor filter as mother wavelet'.

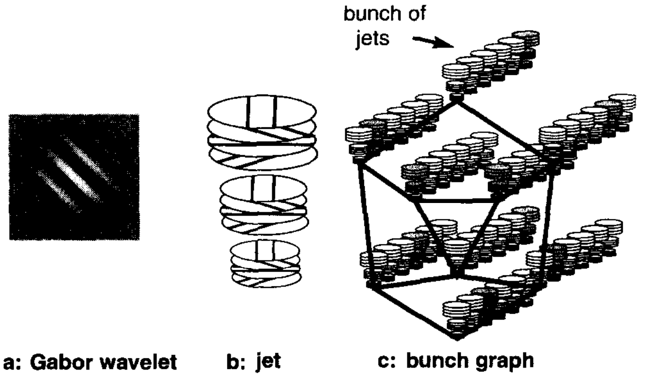

Fig. 2. Representation of heads at a certain pose and size. (a) The real part of a Gabor wavelet. (b) A jet calculated as a set of Gabor wavelets (the discs symbolize the different frequencies and directions of $\vec{k}$). (c) A bunch graph.

mately 10% of the pictures. Before the operation, images are scaled (and sampled) for constant head size. All jets for a given landmark are attached as a bunch to that node. For each landmark, node positions (measured relative to the center of gravity of their graph) are averaged, and the distance vectors between these average positions are stored as edge labels. One such bunch graph represents heads at a certain pose and size (see Fig. 2(c)). The collection of bunch graphs for five different poses p (frontal, left and right profile and half profile) and three different sizes s ($s \in$ {large, middle, small}) form our raw knowledge base, to be optimized by the statistical methods of Sections 3 and 4. Our notation for the various objects and collections of objects in our model domain is given in the upper part of Table 1.

2.2. The algorithm for pose estimation

A bunch graph is adapted to an image by Elastic Graph Matching (EGM) [11,15]. For this, a graph is compared node by node to jet information extracted at the current node position from the image and the total similarity is optimized by shifting. scaling and deforming the graph. The optimal similarity value for a graph gives the quality of its fit to the image. To estimate pose, the whole collection of bunch graphs is matched. The graph with the highest similarity determines the pose, but simultaneously it gives size and position of the face within the image, while the positions of its nodes identify landmarks.

Table 1
Denotation of the representation of faces and quantities concerning the matching process

Representation of faces	
$\mathcal{G}$	collection of bunch graphs
${}^{G\mathcal{G}}(p,s)$	bunch graph of pose p and size s
$\mathcal{B}^{\mathcal{G}}(p,s,k)$	bunch of jets at node k
$n^{\mathcal{G}}(p,s,k)$	number of jets in bunch at node k
$\mathcal{J}^{\mathcal{G}}(p,s,k,i) = B_i^{\mathcal{G}}(p,s,k)$	ith jet at node k

Matching process	
$\beta^{\mathcal{G}}(p,k)$	weight of node k in matching step m
$\mu^{\mathcal{G}}(m,p,k) \in \{0;1\}$	flag if node k is used in matching step m
$N^{\mathcal{G}}(m,p) := \Sigma_k \mu^{\mathcal{G}}(m,p,k)$	number of nodes used in matching step m
$p^{\rightarrow\,\mathcal{G}}(p,s,k)$	position of node k after matching process

To calculate graph similarities a similarity measure between two jets has to be defined. At different stages of the match procedure we use two different jet similarity measures, *magnitude-similarity* $S_{\text{mag}}(\mathcal{J},\mathcal{J}')$ [11] and *phase-similarity* $S_{\text{pha}}(\mathcal{J},\mathcal{J}')$ [15,16]. The phase similarity is more sensitive to spatial displacements and gives, in addition to the similarity value, an estimate of the displacement $d^{\rightarrow}$ which allows more precise matching of landmarks.

As a node is actually labelled with a bunch of jets, we define bunch similarities ($S^{\mathcal{B}}_{\text{mag}}(\mathcal{B},\mathcal{J})$) and ($S^{\mathcal{B}}_{\text{pha}}(\mathcal{B},\mathcal{J})$) to an image jet $\mathcal{J}$. These are

$$S^{\mathcal{B}}_{\text{mag}}(\mathcal{B},\mathcal{J}) = \max_i \{S_{\text{mag}}(\mathcal{B}_i,\mathcal{J})\}$$

and

$$S^{\mathcal{B}}_{\text{pha}}(\mathcal{B},\mathcal{J}) = \max_i \{S_{\text{pha}}(\mathcal{B}_i,\mathcal{J})\}.$$

During the matching process a bunch graph $G^{\mathcal{G}}$ is compared to a replica of itself that is positioned, scaled and distorted within the image. If the resulting graph is called G^I (*image graph*), if its image jets are called $\mathcal{J}^I$ and if the logical constants $\mu^{\mathcal{G}}\{m,p,k) \in \{0;1\}$ determine if the node k of pose p is actually used or not in the matching step m, the averaged node similarity between bunch graph and image graph is computed as

$$S^G(G^{\mathcal{G}}_{p,s}, G^I) = \frac{1}{N^{\mathcal{G}}_{m,p}} \cdot \sum_{k=1}^{N^{\mathcal{G}}_{m,p}} \mu^{\mathcal{G}}_{m,p,k} \cdot S^{\mathcal{B}}(\mathcal{B}^G_{p,s,k}, \mathcal{J}^I_k)$$

(where the arguments of all quantities are written as indices). The total similarity between graphs is calculated as the sum of two terms, the average of individual node similarities just computed, and a (negatively taken) measure for relative graph distortion, for details see [2,11].

The complete graph matching process used in this paper proceeds in four steps. The matching procedure is performed for the bunch graphs of all poses and sizes in the collection and the one with the best similarity in the final step is selected, its identity determining pose and size. The four matching steps are: (i) Rough location of the head in the image; (ii) Adaptation of scale and improvement of location; (iii) Independent scaling in x- and y-direction and further improvement of location; and (iv) Independent adaptation of node positions. In the first matching step we use the magnitude similarity and in steps (ii)–(iv) the phase similarity, utilizing the displacement estimation for fitting the grid to the correct location.

The system as described up to this point already determines position, size and pose of faces with fairly high reliability. We achieve a performance of 87.9% on a set of 413 pictures and the processing of one picture requires 112 s on a SPARC 20. In the following we optimize the speed and performance of the system described above by statistical methods. Instead of using all nodes and jets of our bunch graphs in all four matching steps, we drive towards a system which reduces matching time and improves overall performance by ignoring nodes and jets in our face representation in the first three matching steps. The selection of these is the subject of the next sections. It is important for the performance of the pose estimation system that the full grid is positioned after the final matching step. Since the fourth step is relatively fast, we are able to perform it on the full set of nodes and jets without significant increase of computational time.

The output of the pose estimation system can be used as input to the face recognition system described in [15], in which the pose of a face was assumed to be known.

3. Formalization of the principles of learning

We now proceed to formalize the principles discussed in the introduction. For each pose and for each matching step we would like to have sparse but efficient bunch graphs. The number of possible choices of bunch graph entries and subgrids is very large.[3] It seems to be impossible to do a learning in this large space, especially because the evaluation of the suitability of a specific selection of parameters takes a long time. By making use of principle P0 about locality and independence of nodes we reduce this enormous combinatoric space to a sum of small spaces, which can be further reduced to a five-parameters search by applying the properly formalized a priori principles P1 and P2.

In the following subsections we use P1 and P2 to define the derived and formalized principles for the task of finding suitable nodes and bunch graph entries. The derived principles can be divided into derived principles for choosing suitable jets in the bunch graph (DM1) and the derived principles for choosing a suitable subgrid (DG1, DG2, DG3 and DG4). The corresponding formalized principles are called FM1, FG1, FG2, FG3, and FG4. Expressed in our specific notation (Table 1) we have to formulate a learning scheme to determine the quantities $G^{\mathcal{G}}(p,s)$, $n^{\mathcal{G}}(p,s,k)$ and $\mu^{\mathcal{G}}(p,m,k)$.

Matching time depends linearly on the number of evaluations of our similarity functions,

$$\sum_{p,m,s,k} \mu^{\mathcal{G}}(p,m,k)\cdot n^{\mathcal{G}}(p,s,k). \qquad (1)$$

Reducing this quantity leads to a direct linear speed-up of matching. We achieve this by learning a representation which suppresses many jets or whole nodes according to their importance for the matching algorithm.

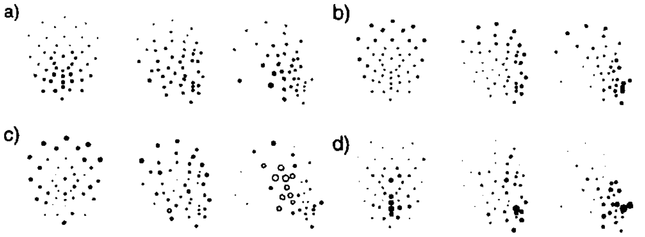

Fig. 3. (a) $n^{\mathcal{G}}(p,\text{large},k)$, i.e., the number of entries in the clustered bunch graph corresponding to the largest size. There are significant differences in the number of jets needed to cover the different landmarks; e.g., for the frontal pose the mouth region has many more entries than the outline of the face or the eye region. (b) FG1$(p,1,k)$, i.e., the difference of the mean of the distribution for the magnitude similarity of correctly positioned nodes to incorrectly positioned nodes in the first matching step. For all poses the occluding boundary at the top of the head is a good feature according to FG1. (c) FG3$(p,3,k)$, estimation of the displacement in the third scan step. It is more reliable near the occluding boundary and the mouth region than in the eye regions (unfilled circles mark negative values – cases where the estimated displacement worsens rather than betters reliability). (d) The learned weights for the pose estimation task for frontal (left), half profile (middle) and profile (right) views. The tip of the nose is very significant for the recognition of the frontal and half profile poses, the lips are very significant for the discrimination of frontals and half profiles. The top of the head and the eyes are not very significant for the task of pose discrimination.

For the final decision of the pose estimation algorithm, we extend the total similarity described in Section 2 by a weighted average of the local similarities which takes the importance of the different landmarks for the pose estimation problem into account (see Fig. 3(d)). These weights are learned by an algorithm introduced in [8] which also makes use of derived principles based on P1.

3.1. Selection of jets

In our bunch graph approach. to represent the shape of a landmark we store $n^{\mathcal{G}}(p,s,k)$ jets, extracted from as many images of pose p and size s, at the kth landmark. In the past [15] we based the selection of suitable sample images on intuitive criteria such as balancing the data base in terms of gender or race, hoping to cover the space of eye jets, nose jets, etc., appropriately while avoiding redundancy. The same number $n^{\mathcal{G}}(p,s,k) = n^{\mathcal{G}}(p)$ of jets was used for all nodes. Here we define a *more systematic* and clustering-like selection of jets. Assuming a landmark is already represented by a number of jets, as a simple application of principle P2 we can formulate

DM1. A new jet should be added only if it is not similar to an existing one in the representation.

[3] E.g. the number of possible subgrids is 2^N with $N = \sum_{p,m} N^{\mathcal{G}}(p,m)$, in our case this would be approximately $2^{(3\cdot 3\cdot 45)} = 2^{405}$.

We implement this principle with the help of a very simple clustering algorithm. Before including a new jet we check whether there is one already in the same bunch that is close to this entry according to our similarity function S_{pha}. If that is the case we erase one of the two entries from our bunch graph. More formally the algorithm can be described in an iterative way as the formal principle

FM1. For $i > 1$ we assume a bunch and $\mathcal{B}^{\mathcal{G}'}$ is already defined (for $i = 1$ we take the jet of the first entry $\mathcal{J}^{\mathcal{G}}(p,s,k,1)$). We take $\mathcal{J}^{\mathcal{G}}(p,s,k,i)$, $i > 1$. Then we calculate for each k the similarity $S^{B}_{pha}(\mathcal{B}^{\mathcal{G}}{}'(p,s,k), \mathcal{J}^{\mathcal{G}}(p,s,k,i))$ of $J^{G}(p,s,k,i)$ to the bunch $\mathcal{B}^{\mathcal{G}}{}'(p,s,k)$ we have selected so far and we add $J^{G}(p,s,k,i)$ to the bunch graph $\mathcal{B}^{\mathcal{G}}{}'(p,s,k)$ if $S^{B}_{pha}(\mathcal{B}^{\mathcal{G}}{}'(p,s,k), \mathcal{J}^{\mathcal{G}}(p,s,k,i))$ is smaller than α_0 or we skip $\mathcal{J}^{\mathcal{G}}(p,s,k,i)$ if the similarity to the bunch graph is higher than α_0. It is obvious that the size of the clustered bunch graph depends on α_0. In case of a smaller α_0 we have a coarser representation than compared to a larger α_0. The α_0 is controlled at a more global stage of the algorithm (see Section 4).

Fig. 3(a) shows the number of entries in the bunch graph after clustering for each pose for the largest size. It is obvious that certain nodes need fewer entries than others to cover their landmark. This fact is especially important regarding DG2.

Comparing our clustering algorithm to Principle Component Analysis (PCA), we remark that the samples of a landmark may refer to *separated* locations in the sample space. For instance, it is questionable that there exists a continuous transformation between eyes with and without glasses, but it is more likely that these two types of samples occur as separated clusters in the space of eye jets. Here we want to reduce the amount of data needed to represent the jet-space spanned by a certain landmark. If principle components of this space are analyzed and the subset spanned by the first few components is used to describe a landmark 'empty' regions between clusters would be part of the representation of faces. Therefore we assume that for this problem a clustering-like selection of jets is more appropriate than PCA. For a more detailed discussion of the problems and restriction of PCA we refer to [4].

3.2. Principles for selecting nodes

In this subsection we define a learning algorithm for a suitable subgraph which only contains the nodes important for head finding and pose identification. The bunch of jets that we are using to describe a landmark can be seen as a composite feature which during the match is compared to the jet centered at a given pixel to decide whether the given landmark is present or not. These local comparisons, in the form of similarity values, are averaged over the nodes of the graph as a basis for the decision whether there is a head in a specific pose and size is found at the actual position in the image or not. Furthermore, each bunch gives an estimate of the displacement to the correct location. Bearing these remarks in mind we can formulate derived principles for the usefulness of a certain landmark of the grid to represent a face. A certain node is

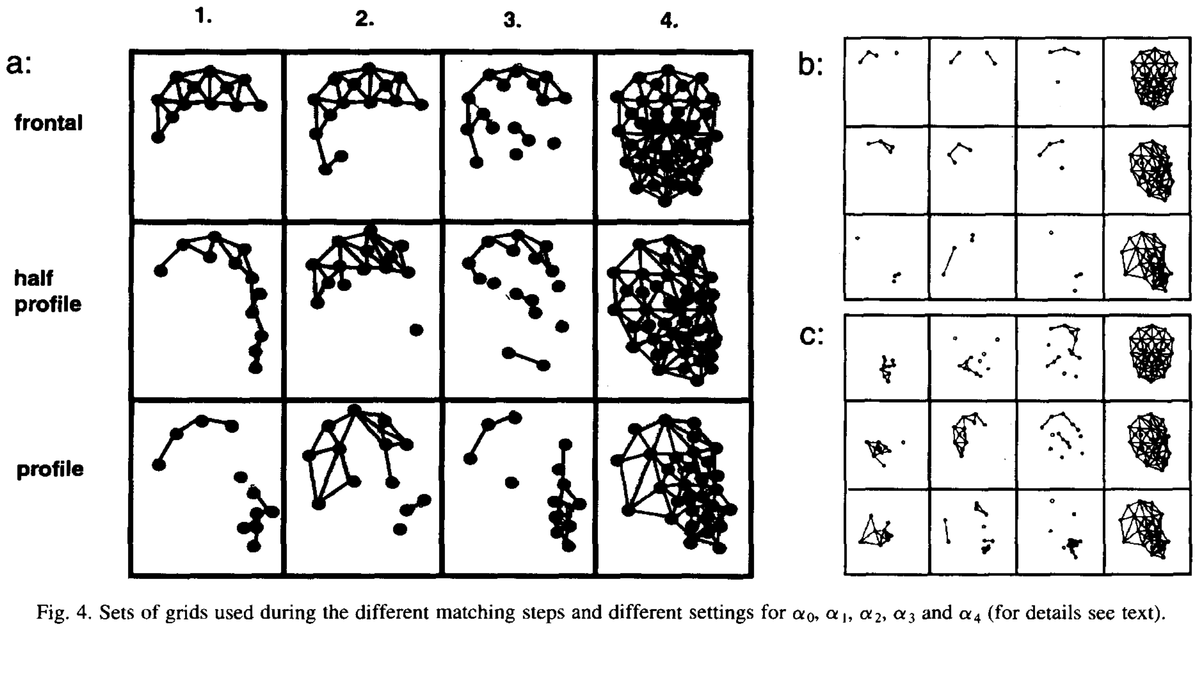

Fig. 4. Sets of grids used during the different matching steps and different settings for α_0, α_1, α_2, α_3 and α_4 (for details see text).

useful if:

DG1. It provides high values at the correct position and low values at incorrect positions.

DG2. The number of jets needed to cover all variations of a local region is low.

DG3. The estimate of displacements of the node relative to the correct position is mostly correct.

DG4. Nearby in the graph there is no other node that is covering parts of the information of this landmark.

DG1 is a simple application of P1, DG2 and DG4 are applications of P2. DG2 is specifically important for speed. DG3 is the only principle which is specific for our system. It simply gives an additional hint for the suitability of a feature and could easily be excluded from the algorithm.

In the following the derived principles DG1, DG2, DG3 and DG4 are formalized. By using these formalized principles we define an expression FG1234(p,m,k,α_1,α_2) which estimates the importance of nodes for the different poses and matching steps (The notation FG1234 stresses that DG1, DG2 DG3, and DG4 were used in its formulation). The α_1, α_2 are parameters which are adapted at a global stage of the algorithm (Section 4). They control the relative weight of the clues DG1–DG3. Fig. 4(a) shows one set of the learned subgrids used in the different matching steps. The reader who is not interested in details of the formalization may skip to Section 4.

3.2.1. Discrimination

Assuming $\mathcal{J}^l(x^{\rightarrow})$ is a jet extracted from the image at or near the kth landmark. We then can take the similarity $S^{\mathcal{B}}(\mathcal{B}^{\mathcal{G}}(p,s,k),\mathcal{J}^l(x^{\rightarrow}))$ as evidence for whether $\mathcal{J}^l(x^{\rightarrow})$ does or does not represent the kth landmark. We can compare this to the ground-truth, in the following called T_0, given for those pictures. More formally we define the FG1 as

$$\mathrm{FG1}(p,m,k) := \langle \max_s (S^{\mathcal{B}}(\mathcal{B}^{\mathcal{G}}(p,s,k),J^c))\rangle_{J^c}$$
$$-\langle \max_s ((S^{\mathcal{B}}(\mathcal{B}^{\mathcal{G}}(p,s,k),J^w))\rangle_{J^w},$$

with $\langle\rangle_{J^c}$, and $\langle\rangle_{J^w}$ as the mean over correctly and incorrectly positioned jets, respectively. The dependence on m derives from that of $S^{\mathcal{B}}$. We further mark that the FG1(p,m,k) are all positive, i.e., each landmark is useful for the segmentation according to DG1.

The FG1(p,1,k) values for the different poses p for the first matching step are displayed in Fig. 3(b). For this step the magnitude similarity S_{mag} is used and the bunch graph is replaced by one containing only one average jet per node. There are significant differences in the quality of a node according to DG1. For example, the nodes at the outline are much more suitable as detectors of the corresponding landmarks compared to the inner nodes (a difference that may vanish when working with non-homogenous background).

3.2.2. Jet selection

The formalization of DG2 is based on the clustering algorithm described in Section 3.1 and a count of the number of entries $n^{\mathcal{G}}(p,s,k)$ for each node after clustering. The value to be optimized is inversely proportional to the number of jets in a node:

$$\mathrm{FG2}(p,k) := \left(\frac{\sum_s n^{\mathcal{G}}(p,s,k)}{\sum_s \max_k\ (n^{\mathcal{G}}(p,s,k))}\right)^{-1}.$$

3.2.3. Estimate of displacement

Our similarity function S_{pha} gives us an estimate of the displacement $d^{\rightarrow}$. A plausible quality measure FG3(p,m,k) for node (p,m,k) can be derived from the improvement of the distance to the correct position by the displacement.[4] Fig. 3(c)) shows the FG3(p,3,k).

3.2.4. Reduction of mutual node dependence

As the receptive fields of jets extracted at neighbouring pixel positions are overlapping their responses are not independent. There might be a particular node which is important according to the principles DG1, DG2 and DG3 but the features of this node are already covered by a node nearby. In that case it might be better to include a node which is less significant according to these principles but which represents novel features. We define a value FG4(p,k,l) which measures the dependence of two nodes k,l in an average grid representing pose p. Furthermore we define a measure FG123(p,m,k,α_1,α_2) for the importance of the nodes according to the criteria DG1, DG2 and DG3 only. The two measures FG4(p,k,l) and FG123(p,m,k,α_1, α_2) are combined to a measure FG1234(p,m,k,α_1,α_2) for the importance of the nodes after taking their

[4] Let $x^{\rightarrow}$ be an arbitrary position near the correct position $p^{\rightarrow T_0}(p,s,k,i)$. Then we define

$$id(\vec{x},\vec{p}\,T_0(p,s,k,i)) = \|\vec{x}-\vec{d}-\vec{p}\,T_0(p,s,k,i)\| - \|\vec{x}-\vec{p}\,T_0(p,s,k,i)\|,$$

which gives the improvement achieved by using the estimated displacement. As a measure for the quality of the displacement estimation we choose the mean of the *id* over the jets in the bunch and over the positions actually tested during the scan process (which axe already near the correct point),

$$\mathrm{FG3}(p,m,k) := \langle id(\vec{x},\vec{p}\,T_0(p,m,k,i))\rangle_{(x,i)}.$$

dependencies into account.[5] The parameters α_1, α_2 are mainly responsible for the weighting of the different clues DG1, DG2 and DG3.

4. Overall control

Section 3 gives a scheme which determines significant nodes depending on the parameters α_1,α_2. The number of nodes has to he fixed before this scheme can be applied. We assume a linear dependence for the number of nodes in the different matching steps,

$$N^{\mathcal{G}}(p,m) = \alpha_3 + \alpha_4 \cdot m, \quad m\ :\ 0,1,2. \tag{5}$$

In the last step the complete grid is used (see Fig. 4).

Now we are able to control the complete representation by the parameters $\alpha_0,\ldots,\alpha_4$. We thus have been able to reduce the dimension of our problem to five parameters by using our a priori constraints. The optimal parameters are now learned by introducing an evaluation function $Q(\alpha_0,\ldots,\alpha_4)$ in which two goals are formalized, the system should to reliable and efficient. We measure the performance $P(\alpha_0,\ldots,\alpha_4)$ by simply counting the number of correctly recognized poses, and measure the speed $S(\alpha_0,\ldots,\alpha_4)$ of the system by the number of similarity function evaluations (see eqn (1)). The relative importance γ of both criteria may depend on the actual task, and we combine P and S to Q accordingly:

$$Q(\alpha_0,\ldots,\alpha_4) := \gamma \cdot P(\alpha_0,\ldots,\alpha_4) + (1-\gamma) \cdot S(\alpha_0,\ldots,\alpha_4), \quad 0 \leq \gamma \leq 1. \tag{6}$$

This function is now optimized by a simple searching the full parameter space for local optima.

5. Results and discussion

We tested our head finding and pose estimation algorithm on a set containing 413 pictures, of which 180 are frontals, 108 halfs (46 half left and 61 half right) and 126 profiles (26 left and 100 right). The size varies in the range of 3 octaves and the poses vary in a range of 180 degrees. This test set is completely independent of the data used for generation of the bunch graphs or our learning algorithm.

Table 2 gives the performance and CPU-time for the different settings of the pose estimation algorithm. The first row shows the results on the test set with the full setting, i.e. the performance of the algorithm with all nodes and jets and without a pose specific weight matrix. The other rows show the performance and time requirement for the learned settings. We can increase performance of recognition from 87.9% to 92% and speed up processing by more than a factor of six over that of the 'full setting' (row 2). Fig. 4(a) shows the grids for the different poses and scan steps corresponding to the parameter in row 2. The 5th row shows the results for a learned representation under the constraint that only 4 nodes are used in each of the first three matching steps. Here we still got a slightly better performance compared with the full setting and we gain a speed-up of a factor of 9.7. Fig. 4(b) shows the corresponding grids. The 6th row gives the results for a bad parameter choice with negative α_1 and α_2. The performance decreases significantly, mainly because the first matching step often fails. Fig. 4(c) shows the corresponding grids. As a qualitative description of the influence of the parameters α_0–α_4 on the performance of the system, we find that for clustering a threshold $\alpha_0 = 0.9$ is sufficient. A smaller α_0 only leads to an increase of matching time (caused by the increase of the number of jets in the bunches) without improving the performance. The performance does not change significantly under variations of α_1 and α_2, as long as both values are larger than 0. This can be explained by the fact that FG1, FG2 and FG3 are not independent variables but have high correlation (see also Fig. 3(a)–(c)). Furthermore, it turned out that approximately twelve properly selected nodes (controlled by α_3 and α_4) for the first three scan steps are sufficient for a good matching and an increase of the number of nodes only leads to an increase of CPU-time. Concerning the assessment of the recognition rate of 92% we remark that our algorithm has to simultaneously solve three problems, head finding, scale normalization and pose estimation.

The algorithm as described in Section 2 already works

[5] We formalize FG1234(p,m,k,α_1,α_2) in our algorithm as follows: We assume that the number of nodes $N^{\mathcal{G}}(p,m)$ to be chosen in a particular matching step is already given (the number of nodes is determined at a more global stage of the algorithm, see Section 4). Furthermore, we define a measure FG123(p,k,α_1,α_2) for the importance of the different nodes without taking their dependency into account by the weighted sum of the measures estimating the quality of landmarks according to DG1, DG2 and DG3,

$$\mathrm{FG123}(p,m,k,\alpha_1,\alpha_2) = \alpha_1 \cdot \mathrm{FG1}(p,m,k) + \alpha_2 \cdot \mathrm{FG2}(p,m,k) + (1-\alpha_1-\alpha_2) \cdot \mathrm{FG3}(p,m,k).$$

The mutual dependence of the two nodes k and l for pose p is defined as

$$\mathrm{FG4}(p,k,l) := g(\vec{p}\, T_0(p,k) - \vec{p}\, T_0(p,l)), \tag{2}$$

with

$$g(\vec{x}) := \max(1 - \frac{|\vec{x}|}{10}, 0) \tag{3}$$

and with $p^{\rightarrow T_0}(p,k)$ as pixel position of the kth node of an average graph for pose p.

Now we define our subgraph iteratively. First we take the best node k_1 according to FG123(p,m,k,α_1,α_2) as the first node of the subgraph. Then we take the second-important node k_2 according to FG123(p,m,k,α_1,α_2) and we calculate

$$\mathrm{FG1234}(p,m,k_2,\alpha_1,\alpha_2) := \mathrm{FG123}(p,m,k_2,\alpha_1,\alpha_2) \cdot \sum_{l \in \mathrm{grid}} (1 - \mathrm{FG4}(p,m,k_2,l). \tag{4}$$

If $\mathrm{FG1234}(p,m,k_2,\alpha_1,\alpha_2)$ is larger than all FG123(p,m,k,α_1,α_2) with $k \neq k_1$ and $k \neq k_2$ we include k_2 into the grid, otherwise we skip k_2. We continue this scheme until we have found $N^{\mathcal{G}}(p,m)$ nodes.

Table 2
Performance and CPU-time depending on the parameter $\alpha_0,\ldots,\alpha_4$ (for details see Section 5)

Results							
	Parameters					Performance on test set	CPU-time per picture
	α_0	α_1	α_2	α_3	α_4		
1	full setting					87.9%	108.8 + 4 s
2	0.9	0.1	0.1	10	2	92.0%	14.8 + 4 s
3	0.9	0.5	0.2	6	15	91.3%	17.3 + 4 s
4	0.9	0.8	0.1	4	10	90.8%	18.6 + 4 s
5	0.9	0.1	0.1	4	0	88.3%	7.5 + 4 s
6	0.9	− 0.3	− 0.3	10	2	80.2%	16.1 + 4 s

with high reliability. By applying our learning algorithm described in Section 3 we can achieve further gains in performance and a significant speed-up. The improvement of performance can be explained by the introduction of weighting of the nodes for the classification of poses. Furthermore, the accuracy of matching is improved by the fact that our learning algorithm allows us to use only important nodes for the head finding problem. The improvement of speed is caused from the sparseness of the learned representation.

6. Comparison to other pose estimation systems

We improved the representation of faces described in [15] by introducing our learning algorithm and we applied this representation to the problem of head finding, size normalization and pose estimation over a wide range of sizes (three octaves) and poses (180 degrees rotation in depth). In [15] the pose of a face is assumed to be known. Here we made this system able to deal with arbitrary poses.

A system which can determine an arbitrary pose more precisely in the context of image sequences within the very same framework of EGM and jets is described in [12], where in contrast to our approach the faces have to be normalized according to size and the initial pose in the sequence has to be known. Therefore the pose estimation system described here can be used as an initial step of the system in [12]. In [12] head image sequences starting with the frontal face view the landmarks are placed automatically in the start frame similar to the algorithm described here and then tracked from frame to frame. The pose is determined simultaneously by integrating the motion of the landmarks. Doing so with different subjects leads to a bunch of face graphs of all possible poses, which can be sorted to build the bunch graphs for new pose classes. In this way our system can be extended to new poses automatically.

In [13] PCA is used to discriminate poses. This system uses local Eigen-features. Manually selected landmarks, such as eyes and mouth, are used to apply a local PCA. In comparison to [13] we are able to learn the important landmarks for a certain task from a large set of landmarks. As for the landmarks themselves there are two main differences: Firstly, the linear transformation based on Gabor filters as basis feature is more biologically plausible than those filters created by a PCA [6]. Secondly, PCA is exclusively data-driven, i.e., applied to instances of a certain landmark it leads to a special set of filters which is independent of the task to be solved (e.g., head finding and pose discrimination). Our system has a task-driven component because a subset of those jets (each of which represents an instance of the landmark considered) which solves a certain task best is applied. Besides, a quantitative comparison is hardly possible, because the CPU-time needed. the variability of sizes and poses considered, and the number of poses being separated in [13] remain unclear.

The pose estimation system [10] determines the pose in a range of ± 25 degrees differing from frontal pose with slight variations in size. It is based on a manually defined grey-level model of a frontal face in which rotation is parameterized. An adaptation of the model to the input face is controlled by the parameters of the model from which the rotation angle can he calculated after matching. In comparison to this system our representation is partially learned and deals with a much higher variability of the sizes and poses.

The pose estimation system in [1] also deals with poses with a variation of ± 25 degrees differing from frontal faces without significant size differences. The face representation is based on a large amount of local templates based on the pixel values in local areas. The position of the local templates (eyes and nose) are defined by hand. Matching is accelerated by a coarse to fine strategy. But still the pose estimation in [1] is much slower and needs 10–15 min on a Sparc 2.

7. Outlook

Currently we are working on a real-time face-spotting system based on the algorithm described here. This system shall extract a size-normalized frontal face from a sequence of images of a person moving towards a camera.

We readily concede that at the present stage our object representation is still very dependent on manual construction of the initial pose graphs and on manual provision of ground truth in the form of correct landmark positioning in training images. Our eventual goal must be to reduce the importance of these knowledge sources and to achieve a system that learns autonomously, which can then recognize and track a wide range of objects in complex scenes. Based on the idea of reduction of dimensionality by general a priori principles (as described in this paper) and the concept of sparse coding (discussed in [4]) we are currently working on features which are more flexible and able to express the 'essence' of landmarks, representing for instance, an eye brow as a horizontal line which is slightly bending downwards [9]. Such a representation will reduce memory requirements compared to our bunch-of-jets approach, may increase speed of matching, and could be more reliable, as the system could more narrowly focus on essential features. We believe that with a more extensive application of the concepts introduced in this paper, a system could autonomously learn the necessary representations and mechanisms to deal with arbitrary objects in complex scenes. In [9] we already made one more step towards this demanding goal.

8. For further reading

[5,14]

Acknowledgements

We wish to thank Laurenz Wiskott, Michael Lyons, Thomas Maurer, Ladan Shams, Jan Vorbrüggen and Gabi Peters for fruitful discussions. We thank Michael Rinne for his great help during the implementation of the algorithms. For the experiments we have used the FERET database of facial images collected under the ARPA/ARL FERET program.

References

[1] D.J. Beymer, Face recognition under varying pose, Proc. Int. Workshop on Automatic Face and Gesture recognition, Zürich, 1995, Editor M. Bichsel. Published by the Multimedia Laboratory.

[2] J. Buhmann, M. Lades, C v.d. Malsburg, Size and distortion invariant object recognition by hierarchical graph matching, Proc. IJCNN Int. Joint Conf. on Neural Networks, San Diego 1990. Published by IEEE Neural Network Council, pp. 411–416.

[3] J.D. Daugman, Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression, IEEE Trans. Acoustics, Speech, and Signal Processing 36 (1988) 1169–1179.

[4] D. Field, What is the goal of sensory coding?, Neural Computation 6(4) (1994) 561–601.

[5] S.L.S. Jacoby, J.S. Kowalik, J.T. Pizzo, Iterative Methods for Nonlinear Optimization Problems, Prentice-Hall, Englewood Cliffs, NJ, 1972.

[6] J.P. Jones, L.A. Palmer, An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex, J. Neurophysiology 58 (6) (1987) 1233–1258.

[7] D. Koller, K. Daniilidis, H.H. Nagel, Model-based tracking in monocular image sequences of road traffic scenes, Int. J. Computer Vision 10 (3) (1993) 257–281.

[8] N. Krüger, Learning weights in discrimination functions using a priori constraints, DAGM Editors: G. Sagerer, S. Posch, F. Kummart, Mustererkennung 1995. Springer Verlag, pp. 110–117 (See also WWW: http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/PUBLIST/1995/html/pub95.html.).

[9] N. Krüger, G. Peters, Object recognition with banana wavelets, Proceedings of the European Symposium on Artificial Neural Networks, Brügge 1997, Published by D.Facto, Brüssels, Editor M. Verleysen, pp. 61–66.

[10] A. Lanitis, C.J. Taylor, T.F. Cootes, T. Ahmed, Automatic interpretation of human faces and hand gestures using flexible models, Proc. Int. Workshop on Automatic Face and Gesture recognition, Zürich, 1995, Editor M. Bichsel. Published by the Multimedia Laboratory.

[11] M. Lades, J.C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R.P. Würtz, W. Konen, Distortion invariant object recognition in the dynamic link architecture, IEEE Trans. Computers 42 (3) (1993) 300–311.

[12] T. Maurer, C. von der Malsburg, Tracking and learning graphs and pose on image sequences of faces, Proc. 2nd Int. Conf. on Automatic Face and Gesture Recognition, Killington, Vermont 1996, IEEE Computer Society Press.

[13] A. Pentland, B. Moghaddam, T. Starner, View-based and modular eigenspaces for face recognition, IEEE Conf. Computer Vision and Pattern Recognition, Seattle 1994.

[14] C. von der Malsburg, The correlation theory of brain function, Internal Report 81-2, MPI Bio-physikalische Chemie, Göttingen, 1981. Also in: E. Domany, J.L. van Hemmen, K. Schulten (Eds.), Models of Neural Networks II, Springer, Berlin, 1994, pp. 95–119.

[15] L. Wiskott, J.-M. Fellous, N. Krüger. C. von der Malsburg, Face recognition and gender determination, Proc. Int. Workshop on Automatic Face and Gesture recognition, Zürich, 1995. Editor M. Bichsel. Published by the Multimedia Laboratory.

[16] L. Wiskott, Labeled Graphs and Dynamic Link Matching for Face Recognition and Scene Analysis, Verlag Harri Deutsch, Reihe Physik, vol. 53, 1995.