Distortion Invariant Object Recognition in the Dynamic Link Architecture

Martin Lades, *Student Member, IEEE*, Jan C. Vorbrüggen, *Member, IEEE*, Joachim Buhmann, *Member, IEEE*, Jörg Lange, Christoph v.d. Malsburg, Rolf P. Würtz, Wolfgang Konen

***Abstract*—We present an object recognition system based on the Dynamic Link Architecture, which is an extension to classical Artificial Neural Networks. The Dynamic Link Architecture exploits correlations in the fine-scale temporal structure of cellular signals in order to group neurons dynamically into higher-order entities. These entities represent a very rich structure and can code for high level objects. In order to demonstrate the capabilities of the Dynamic Link Architecture we implemented a program that can recognize human faces and other objects from video images. Memorized objects are represented by sparse graphs, whose vertices are labeled by a multi-resolution description in terms of a local power spectrum, and whose edges are labeled by geometrical distance vectors. Object recognition can be formulated as elastic graph matching, which is performed here by stochastic optimization of a matching cost function. Our implementation on a transputer network successfully achieves recognition of human faces and office objects from gray level camera images. The performance of the program is evaluated by a statistical analysis of recognition results from a portrait gallery comprising images of 87 persons.**

***Index Terms*—Computer vision, distortion invariance, dynamic link architecture, elastic graph matching, object recognition, neural network, wavelet.**

I. Introduction

THIS paper describes an object recognition system based on a new neural information processing concept, the Dynamic Link Architecture (DLA) [1], [2], [3]. The DLA, first proposed in 1981, attempts to solve certain conceptual problems of conventional Artificial Neural Networks. One of the most prominent among these is the expression of syntactical relationships in neural networks. Various ambitious applications become accessible via the Dynamic Link Architecture, such as distortion invariant object recognition, sensory segmentation, and scene analysis. This might soon result in massively parallel and fault-tolerant technical applications as well as new insights into brain function.

The innovative idea behind the Dynamic Link Architecture is the use of synaptic plasticity already on the time scale of information processing and not only for memory acquisition. This enables it to instantly group sets of neurons into higher symbolic units. Conventional neural systems do not provide this ability to bind separate subsets of neurons, inevitably merging them into one structureless global assembly. Whereas conventional schemes are very successful with small, tightly defined problem spaces, they do not cope well with complex problems, especially when flexibility is required. The power of the Dynamic Link Architecture can best be demonstrated by applying it to a complex problem like position and distortion invariant object recognition, which is the subject of this paper.

To demonstrate the performance of our system we chose the problem of face discrimination, which is particularly demanding due to variations in perspective and facial expression. Our system is, however, in no way specialized to that application, as we demonstrate by letting it discriminate also between office items.

Manuscript received February 6, 1991: revised August 27, 1991 and July 16, 1992. This work was funded by grants from the German Federal Ministry of Science and Technology (ITR-8800-H1), from AFOSR (88-0274), by the Stimulus Programme of the European Community (BRAIN) and performed in part under the auspices of DOE by the Lawrence Livermore Laboratory (W-7405-Eng-48).

M. Lades, J. C. Vorbrüggen, R. P. Würtz, J. Lange, C. v.d. Malsburg, and W. Konen are with the Institut für Neuroinformatik, Ruhr-Universität Bochum, D–4630 Bochum 1, Federal Republic of Germany.

J. Buhmann is with the Institut für Informatik II, Rheinische Friedrich-Wilhelms Universität, D–5300 Bonn, Federal Republic of Germany.

IEEE Log Number 9205430

II. Invariant Object Recognition in the Dynamic Link Architecture

In this section we give a qualitative description of object recognition in the Dynamic Link Architecture, with special emphasis on a neural style of formulation. Our specific implementation in more conventional algorithmic fashion is described in Section III, which can be read independently. For a preliminary report see [4].

A. The Representation Domains

Although object representation and object recognition eventually will have to be implemented in a multi-level structure, we will restrict ourselves here for reasons of simplicity to the minimum of two levels—an image domain I and a model domain M (see Fig. 1). Biologically speaking, I may correspond to primary visual cortical areas, and M to inferotemporal cortex.

The image domain contains a two-dimensional array of nodes $\mathcal{A}_x^I = \{(x,\alpha) \,|\, \alpha = 1, \cdots, F\}$. Each node at position x consists of F different feature detector neurons (x,α), where the label α is used to distinguish different feature types. These types could simply be local light intensities, but it is desirable to have more complex types that are derived by some filter operation. The image domain I is coupled to a light sensor array (eye or camera). An input presented to that array leads to a specific activation $s_{x\alpha}^I$ of the feature neurons (x,α) in the image domain I. Thus each node $\mathcal{A}_x^I$ contains a set of

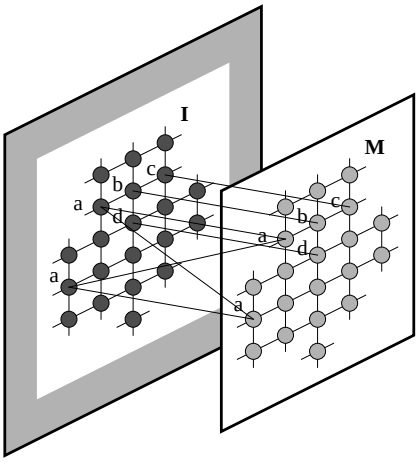

Fig. 1. Matching graphs in image domain and object domain. Within the object domain there is a subgraph M that is identical to a subgraph I in the image domain: I and M contain the same features $(a, b, c, \cdots)$ in the same arrangement. In both domains, neurons in neighboring nodes are connected. Connections between domains are feature-type preserving, but not position specific (for example, all a-detectors in I are connected to all a-detectors in M). Due to graph identity, all connections between corresponding points exist. Graph dynamics is a positive feedback loop: signal correlations favor activation of links, active links produce correlations. Locally, graph dynamics favors richly connected blocks of neurons—here, blocks are formed out of neighboring cells in I and neighboring cells in M, in corresponding positions. More globally, dynamic attractors are optimal combinations of local blocks—here, the graph composed of graph I, graph M and connections between corresponding nodes. Other objects in the model domain lose the competition; they have a different arrangement of feature types, and there is no complete system of one-to-one connections to I.

activity signals $J_x^I = \{s_{x\alpha}^I \,|\, \alpha = 1, \cdots, F\}$. We will use the term "jet" for the feature vectors J_x^I. All cells within I are connected excitatorily over short distances, including zero distance, irrespective of feature type. The connection in I between neuron (x, α) and neuron (y, β) will be denoted as $T_{x\alpha;y\beta}^I$.

With this given structure, images are represented in the image domain as attributed graphs. Their vertices are what we have called nodes. Attributes attached to the vertices are activity vectors of local feature detectors, which we called "jets." The links are the connections $T_{x\alpha;y\beta}^I$ for $x \neq y$. [1] A particular object is represented by that subgraph of the image domain which is affected by the object. For simplicity, we will continue to use the symbol I for object-representing subgraphs in the image domain.

The model domain is an assemblage of attributed graphs, all being idealized copies of subgraphs in the image domain. We will use the symbol M for individual model graphs, although it occasionally will also refer to the whole model domain.

There are excitatory connections $T_{x\alpha;y\beta}^{IM}$ between image domain and model domain. These connections are feature-preserving: Two neurons, one in the image domain, one in the model domain, have a connection between them if and only if they belong to corresponding feature types. There is no condition on the position within the image domain. Thus, a graph in I and an identical graph in M have a complete set of connections between corresponding neurons in corresponding nodes, irrespective of the position of I in the image domain.

Object identification in the structure just described can now be realized as a process of elastic graph matching. Two subproblems have to be solved: Identification of an appropriate subgraph I of the full image domain ("segmentation"), and identification of a matching subgraph M in the model domain.

B. Links and Their Dynamics

What is the machinery necessary to perform the attributed graph matching? It must be based on a data format able to encode information on attributes and links in the image domain and to transport that information to the model domain. This has to be done without automatically sending information on position within the image domain, to achieve our goal of separating position information from relational information.

Here is a signal code which does just that. We assume that the actual output signal $\sigma_{x\alpha}^I(t)$ of neuron (x, α) in I fluctuates rapidly in time. The structure of the signal $\sigma_{x,\alpha}^I(t)$ is determined by three factors: the input image, random spontaneous excitation within neurons, and interaction with other cells of the same or neighboring nodes in the image domain, using connections $T_{x\alpha;y\beta}^I$. The actual attribute values can be read off the fluctuating signals as time averages, $s_{x\alpha}^I = \langle \sigma_{x\alpha}(t) \rangle_t$, where the average $\langle \cdot \rangle_t$ is taken over a time interval shorter than the presentation time of the image.

Binding between neurons is encoded in the form of temporal correlations $C_{x\alpha;y\beta}^I = \langle \sigma_{x\alpha}^I \sigma_{y\beta}^I \rangle_t$. These correlations are induced by the excitatory connections within I. The key feature introduced with such correlations is a higher-order neural coding scheme [5] as opposed to first-order coding schemes (mean firing rates) used in conventional neural networks.

Four types of bindings are relevant for the task of object recognition and object representation: i) binding all those nodes and cells together that belong to the same object (segmentation); ii) expressing neighborhood relationships within the image of the object (done with the help of connections $T_{x\alpha;x'\beta}^I$ with $x \neq x'$); iii) bundling individual feature cells within one node into a jet, thus avoiding conjunction errors between features present in different locations (done with the help of connections $T_{x\alpha;x\beta}^I$ within a node); iv) binding corresponding points in image graph and model graph to each other (for which the T^{IM} connections are important).

The basic mechanism of the Dynamic Link Architecture then is as follows. In addition to the connection parameter T_{ij} between two neurons[2] i and j, there is a dynamical variable J_{ij}. Only the J-variables play the role of synaptic weights for signal transmission. The T-parameters merely act to constrain the J-variables (for instance as $0 \leq J_{ij} \leq T_{ij}$). The T-parameters may be changed slowly by long-term synaptic plasticity. The connection weights J_{ij} themselves are subject to a process of rapid modification (taking place in fractions of a second). The weight J_{ij} is controlled (in a way similar to Hebbian plasticity) by the signal correlations $C_{ij} = \langle \sigma_i \sigma_j \rangle_t$ between the neurons i and j: negative C_{ij} lead to a decrease,

[1] An alternative (almost equivalent) view would consider individual feature detectors as nodes, their activities as attributes, and all T^I connections as links.

[2] The index i stands for both position index x and feature index α of neuron (x, α).

positive C_{ij} lead to an increase of J_{ij}. In the absence of any correlations between the two nodes, J_{ij} slowly returns to some resting value J_{ij}^o, a fixed fraction of T_{ij}.

Crucial for the Dynamic Link Architecture is a process of rapid network self-organization. This process is based on a positive feedback loop: A network with a given set of connectivity variables J_{ij} supports an activity process. This is characterized by strong correlations C_{ij} between strongly connected nodes i and j. In turn, strong correlations between nodes lead to strengthening of their connections, closing the loop. Certain constellations of links cooperate with each other in establishing correlations, and consequently in reinforcing themselves. In order to avoid certain instabilities, the positive feed-back loop must be complemented by a competition mechanism between the J_{ij} (which can be achieved in a crude but effective way by keeping the number of links into or out of a node constant).

The positive feed-back loop leads to a run-away situation which can change the connectivity state J_{ij} profoundly and on a fast time scale, eventually stabilizing certain connectivity structures that maximize cooperation and minimize competition between links. Let us call such self-organized structures "connectivity patterns." Among connectivity patterns there is a number of useful structures. A large network can spontaneously decompose into smaller blocks—segments—, a process that is important for scene segmentation [6], [7], [8], [9], [10]). Another type of network pattern has the form of two-dimensional graphs with short-range connections, just as are needed for the representation of images. Finally, network self-organization can activate as connection patterns those composite graphs that are formed by linking all pairs of corresponding nodes in two identical graphs. This latter process is fundamental for the application dealt with here.

C. Object Recognition with Dynamic Links

Attributed graph matching has often been discussed and advocated in the context of object recognition. Our pattern recognition system is based on a special form of attributed graph matching which resembles elastic matching [11], [12]: an attributed graph in the model domain, encoding an object, is locally distorted to cope with deformations and changes in perspective. In the mathematical literature, the relation to be established in graph matching is precise isomorphism. This is too restrictive for the purposes of object recognition. Here, a somewhat less rigid notion of equivalence between graphs is more appropriate: Two graphs are "approximately identical" if there exists an approximate neighborhood-preserving and feature type-preserving mapping between almost all nodes of I and M.

We will give here a qualitative description of the particular process by which elastic graph matching takes place in the Dynamic Link Architecture. If there is a stored model graph M that is identical or approximately identical to a part I of the image domain, then the graph dynamics have to find and selectively activate the subgraph composed of I, M and the one-to-one connections between corresponding points in I and M. This task can be divided into three different aspects: i) group and selectively activate the nodes in the subgraph I of the image domain (segmentation), ii) identify and activate the nodes and links in the subgraph M in the model domain (retrieval of a connection pattern from an associative memory for connection patterns), iii) pare down the many-to-many connections between nodes with similar features in I and M to a consistent (i.e., topology-preserving) one-to-one mapping.

The first aspect amounts to figure-ground segmentation, which can be achieved in part without reference to the model domain, simply by binding nodes x, x' with similar feature vectors (jets) J_x^I and $J_{x'}^I$ together, such nodes being likely to lie within the same object. In such a way, nodes within parts of the image corresponding to one object tend to synchronize their activity, while nodes between different image segments tend to desynchronize and so break their dynamic links. This principle has been exploited and proved to be effective in a number of simulations [6]–[9].

The second of these processes, retrieval of a graph from an associative memory for graphs, has been demonstrated before [3], [13], [14]. The third process, paring down many-to-many connections between topological connectivity patterns to one-to-one connections, has been studied extensively, both in computer simulations [15], [16] and analytically [17]. Essential for this process are events with simultaneous activation of a block of neighboring cells in I and a block of neighboring cells in M, in corresponding positions. All these cells have potentially strong connections among themselves, favoring such double-blocks over other combinations of cells.

A system based on these principles will possess translational invariance, since, according to construction, the set of T^{IM}-connections between a graph I and a graph M is independent of the position of I in the image domain and depends only on the correspondence of feature types in I and M. More generally, further invariances like scale or rotational invariance can be implemented (for an algorithmic version see [18]) by extending the meaning of "feature correspondence" in the construction of the T^{IM}-connections: if, e.g., scale changes are to be accommodated, feature correspondence has to allow for connections between features encoding the same quality on different scales. These connections then allow for the matching of *similar* graphs, where the image is a scaled version of the model.

It should be emphasized here that the three processes i)–iii) described above cannot be carried out sequentially and have to happen in an interlaced fashion, each process needing the partial results of the others. The Dynamic Link Architecture has the potential to achieve this, because the different types of connections (image-image, image-model and model-model) are treated equivalently and can cooperate locally.

III. The Elastic Graph Matching Algorithm

We now come to a concrete implementation, which will specify all detail left open in the abstract description of the previous section. A literal, fully neural realization is not adapted to current computers, and our actual algorithm (see also [4]) is designed for efficient use of available arithmetic processors.

The success of our system rests to a large extent on the particular way in which we use graphs to represent form. Vertices are labeled with collections of features that describe the gray-level distribution locally with high precision and more globally with lower precision, providing for great robustness with respect to deformation. Edges are labeled with metric information on the relative position of vertices. During graph comparison, a parameter specifies the precision with which metric information is to be preserved.

In a nutshell, our system works like this. A set of feature vectors over a *dense* grid of image points is formed, the feature vectors being based on Gabor-type wavelets. During storage, *sparse* model graphs are formed and are labeled with jets from a rectangular subgrid centered over the object to be stored. During recognition, matching takes place by the adaptive formation of a sparse image graph to best match a given model graph. The matching process is based throughout on one-to-one links between vertices in the model graph and the image graph. The process of image graph formation is controlled by a cost function which favors similarity of jets attached to corresponding vertices and which penalizes metric deformation. The process is repeated for every stored model graph, and the match with the lowest cost is identified as the model recognized.

A. Image Processing

Let $I(\vec{x})$ be the gray level distribution of the input image. Our preprocessing then starts with a linear filter operation, which can be written as a convolution of the image I with a family of kernels $\psi_{\vec{k}}$. The parameter $\vec{k}$ determines the wavelength and orientation of the kernel $\psi_{\vec{k}}$. The operator $\mathcal{W}$ symbolizes the convolution with all possible $\vec{k}$:

$$(\mathcal{W}I)\left(\vec{k},\vec{x_0}\right) := \int \psi_{\vec{k}}\left(\vec{x_0}-\vec{x}\right) I\left(\vec{x}\right) d^2x = \left(\psi_{\vec{k}} * I\right)\left(\vec{x_0}\right) . \tag{1}$$

We start with the definition of the kernels $\psi_{\vec{k}}$ in image coordinates. They take the form of a plane wave restricted by a Gaussian envelope function:

$$\psi_{\vec{k}}\left(\vec{x}\right) = \frac{\vec{k}^2}{\sigma^2} \exp\left(-\frac{\vec{k}^2\vec{x}^2}{2\sigma^2}\right)\left[\exp\left(i\vec{k}\vec{x}\right) - \exp\left(-\sigma^2/2\right)\right] . \tag{2}$$

The first term in the square brackets determines the oscillatory part of the kernel. The second term compensates for the dc-value of the kernel, to avoid unwanted dependence of the filter response on the absolute intensity of the image. For sufficiently high values of σ the effect of the dc-term becomes negligible. The complex valued $\psi_{\vec{k}}$ combine an even (cosine-type) and odd (sine-type) part (see Fig. 2).

The filter response of $\psi_{\vec{k}}$ in Fourier space is given by

$$\left(\mathcal{F}\psi_{\vec{k}}\right)\left(\vec{k_0}\right) = \exp\left(-\frac{\sigma^2\left(\vec{k_0}-\vec{k}\right)^2}{2\vec{k}^2}\right) - \exp\left(-\frac{\sigma^2\left(\vec{k_0}^2+\vec{k}^2\right)}{2\vec{k}^2}\right) \tag{3}$$

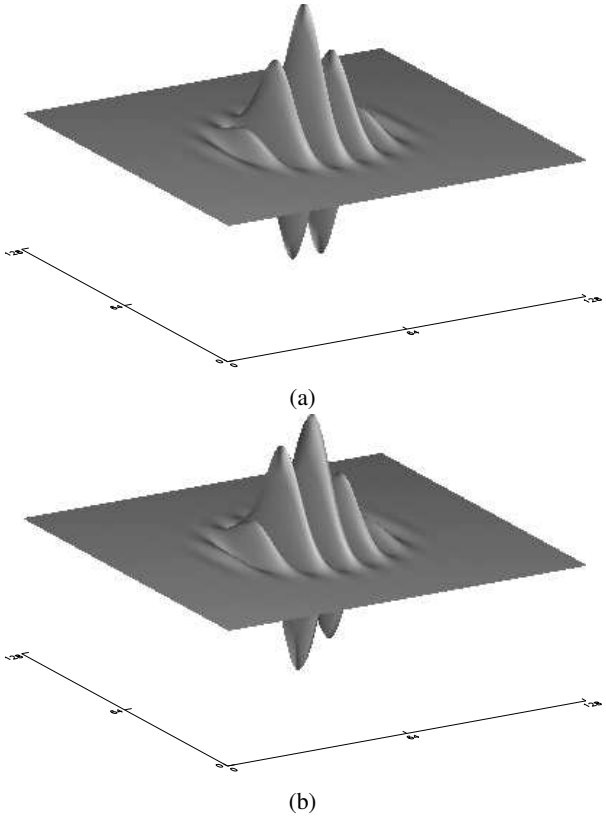

Fig. 2. The shape of a wavelet. (a) The real part (cosine phase) for $|\vec{k}| = 0.72$, $\phi = 45°$. (b) The imaginary part (sine phase). All kernels have the same shape except for size and orientation.

where $\mathcal{F}$ denotes the Fourier transform. The first Gaussian centered at the characteristic frequency $\vec{k}$ provides a bandpass filter. The second exponential removes the dc-component of $\psi_{\vec{k}}$. Equation (3) does not normalize the energy picked up by a kernel in the convolution. Consequently, this energy will be proportional to $|\vec{k}|^2$. D. Field [19] noted that the power spectrum of "natural images" decreases like $1/|\vec{k}|^2$. The energy in the resulting components of our image transform should therefore be roughly independent of $|\vec{k}|$, an assumption which we have confirmed for our images. This property is important for matching and had been enforced rigidly in earlier versions of our system [4].

The $\psi_{\vec{k}}$ form a family that is self-similar under the application of the group of translations, rotations, and scalings. The family is also known as "Gabor-based wavelets". The wavelets are parameterized by the wave vector $\vec{k}$, which controls the width of the Gaussian window and the wavelength and orientation of the oscillatory part. The parameter σ determines the ratio of window width to wavelength, i.e., the number of oscillations under the envelope function.

The Gabor-based wavelets seem to be a good approximation to the sensitivity profiles of neurons found in visual cortex of higher vertebrates (see, e.g., [20]). There is evidence that those cells tend to come in pairs with even and odd symmetry (see [21] and references from [19]), similar to the real and imaginary parts of (2). In the choice of the actual values of relative bandwidth σ and the sampling density of $\vec{k}$, which

are closely related, we diverged, however, somewhat from biological findings in order to achieve better recognition (see Section III-H). Research on this point is ongoing. The convolution (1) is evaluated on a sampling grid of both the spatial domain ($\vec{x}_0$) and the frequency domain ($\vec{k}$), as discussed in the next section and Section III-H.

B. Vertex Labels

In order to derive suitable vertex labels for our graphs from the wavelet transform we had to overcome the following problem: it is known that sharp edges are especially important locations for object recognition. The Gabor-based wavelets respond strongly to edges if the direction is perpendicular to their wavevector $\vec{k}$. But when hitting an edge, the real and the imaginary parts of $\mathcal{W}I$ oscillate with the characteristic frequency instead of providing a smooth peak for matching. To remedy this we abandoned the linearity of our transform and used the magnitude, i.e., the absolute value of the complex response. The magnitude provides a monotonic measure of image properties, i.e., "there is an edge present at position x." Following Fourier terminology we call this quantity the "root of a local power spectrum." It is a positive-valued real function of $\vec{k}$ attached to every point of the image domain. To generate a local description of an image we sample $\mathcal{W}$ at five logarithmically spaced frequency levels (see Fig. 3) and eight orientations indexed by $\nu \in \{0, \cdots, 4\}$ and $\mu \in \{0, \cdots, 7\}$:

$$\vec{k}_{\nu\mu} = k_\nu \begin{pmatrix} \cos\phi_\mu \\ \sin\phi_\mu \end{pmatrix} \text{ with } k_\nu = k_{max}/f^\nu,\ \phi_\mu = \frac{\pi\mu}{8} \quad (4)$$

where f is the spacing factor between kernels in the frequency domain. We have investigated values of $f = 2$ ($k_{max} = \pi/2$) and $f = \sqrt{2}$ (with various values for k_{max}, see Section III-H).

The magnitudes of $(\mathcal{W}I)(\vec{k}_{\nu\mu}, \vec{x}_0)$ form a feature vector located at $\vec{x}_o$, which will be referred to as a "jet":

$$\mathcal{J}_{\nu\mu}(\vec{x}_0) := \left|(\mathcal{W}I)\left(\vec{k}_{\nu\mu}, \vec{x}_0\right)\right| . \quad (5)$$

As part of elastic graph matching, the similarity of pairs of vertex labels has to be evaluated. After some experiments, we settled for the normed dot product of jets as our similarity function. For jets $\mathcal{J}^I$ and $\mathcal{J}^M$ in image domain and model domain, respectively, it is defined as

$$\mathcal{S}_v\left(\mathcal{J}^I, \mathcal{J}^M\right) := \frac{\mathcal{J}^I \cdot \mathcal{J}^M}{\|\mathcal{J}^I\|\ \|\mathcal{J}^M\|} . \quad (6)$$

Being invariant to changes in jet length, $\mathcal{S}_v$ proved to be rather robust with respect to the global changes in contrast induced by varying illumination.

C. Edge Labels

Edge labels encode information on relative position. In the fully neural version of our system, described in Section II, edges were just required to encode neighborhood relationships. During the matching process, the preservation of topology between image graph and model graph was imposed by the constraint of neighboring vertices matching to neighboring vertices, neighborhood being encoded by simultaneous activity. In a digital computer, we can afford to pass metric edge information between the graphs to be compared. We have labeled edges between vertices $\vec{x}_i$ and $\vec{x}_j$ with the Euclidean distance vector:

$$\vec{\Delta}_{ij} := \vec{x}_j - \vec{x}_i, \qquad (i,j) \in E , \quad (7)$$

where E is the set of edges in the image or model graph. The edge labels of the image graph are compared to the corresponding ones in the model graph by a quadratic comparison function

$$\mathcal{S}_e(\vec{\Delta}^I_{ij}, \vec{\Delta}^M_{ij}) := (\vec{\Delta}^I_{ij} - \vec{\Delta}^M_{ij})^2 . \quad (8)$$

In our simulations we have tried different edge sets E, in particular the set $E_{complete}$ of all possible connections and the set E_{nn} containing the (four) next neighbors of each node. Most of our results have been obtained with the latter, which is better suited for handling local distortions.

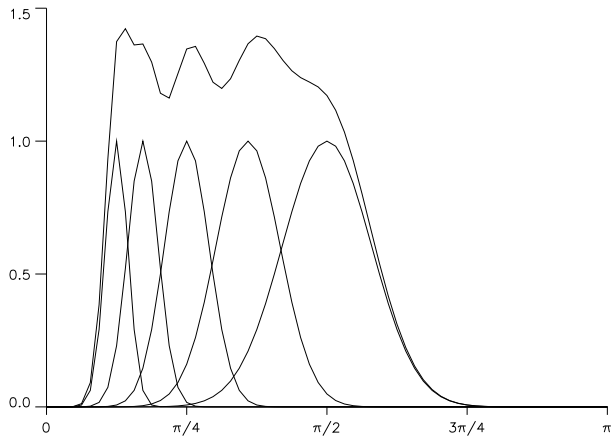

Fig. 3. Plot of the sum of the Fourier transforms of all kernels, section through the line $\vec{k} = (0, k)$. Note that inside the critical frequency range between $\pi/16$ and $\pi/2$ the variations are small compared to the height of the plateau. This indicates that a bandpass-filtered version of the image could be reconstructed from the (linear) transform with good accuracy.

D. Elastic Graph Matching

Elastic matching of a model graph M to a variable graph I in the image domain amounts to a search for a set $\{x^I_i\}$ of vertex positions which simultaneously optimizes the matching of vertex labels and of edge labels. We evaluate the quality of a match according to the cost function

$$\begin{aligned} \mathcal{C}_{\text{total}}(\{x^I_i\}) &:= \lambda\mathcal{C}_e + \mathcal{C}_v \\ &= \lambda \sum_{(i,j)\in E} \mathcal{S}_e(\vec{\Delta}^I_{ij}, \vec{\Delta}^M_{ij}) - \sum_{i\in V} \mathcal{S}_v(\mathcal{J}^I(x^I_i), \mathcal{J}^M_i) , \end{aligned} \quad (9)$$

which is a linear combination of an edge term and a vertex term. The coefficient λ controls the rigidity of the image graph, large values penalizing distortion of the graph I with respect to the graph M. The graph rigidity can even be varied dynamically during optimization, a strategy we use in our two-stage optimization process as described below.

An object to be memorized is extracted from an image as a model graph by placing a rectangular grid of points over the object and by recording the corresponding jets. Our images have 128 $\times$ 128 pixels with 256 gray levels, our grid had 7 $\times$ 10 points spaced by 11 pixels (8 $\times$ 10 points for the office items). For more details see below, Section III-G.

To compare stored model graphs with current image data, a two-stage optimization process is performed, varying the image graph to minimize the cost $\mathcal{C}_{\text{total}}$ of its match to the model. During the first stage of optimization, the image graph is shifted while keeping its form rigid. This corresponds to the limit $\lambda \to \infty$ in $\mathcal{C}_{\text{total}}$. To initialize the process, the shape of the model graph is positioned arbitrarily in the image plane (e.g., centered, or in the lower-left corner). The rigid graph diffuses with a given maximum step size (e.g., 10 pixels). After each step, the total cost $\mathcal{C}_{\text{total}} = \mathcal{C}_v = -\sum_{i \in V} \mathcal{S}_v$ is computed and the new grid position is accepted if it reduces $\mathcal{C}_{\text{total}}$. Since all of our images contain just one object, this global move procedure is able to position the graph on that object. The total cost surface $\mathcal{C}_{\text{total}}$ correspondingly shows just one pronounced minimum (see Fig. 4), even if the graphs I and M belong to two different objects. A more sophisticated segmentation procedure will be required for images with several candidate objects.

Our simulations show that this first stage is very robust. It already works using only the lowest frequency band (center frequency $\pi/8$, width $\sigma = 2\pi$) of the jets. Even small subsets of the 70 vertices and only a few iterations (< 50) are sufficient to position the graph.

During the second stage of the matching procedure the rigidity parameter λ is set to a finite value to permit small graph distortions and the vertices in the image domain can diffuse: they are visited sequentially and in random order and are shifted by a random vector below a preset maximum length. The update procedure terminates when a set of vertex positions $\{x_i^I | i \in V\}$ has been found that constitute a local minimum of $\mathcal{C}_{\text{total}}$ (see Fig. 6).

Both optimization stages correspond to a simulated annealing procedure at zero temperature. As shown in Fig. 5, the local jet potential is also smooth enough to guarantee a straight descent to the minimum. Each stage is terminated once a predefined number of trials have failed to improve the cost value.

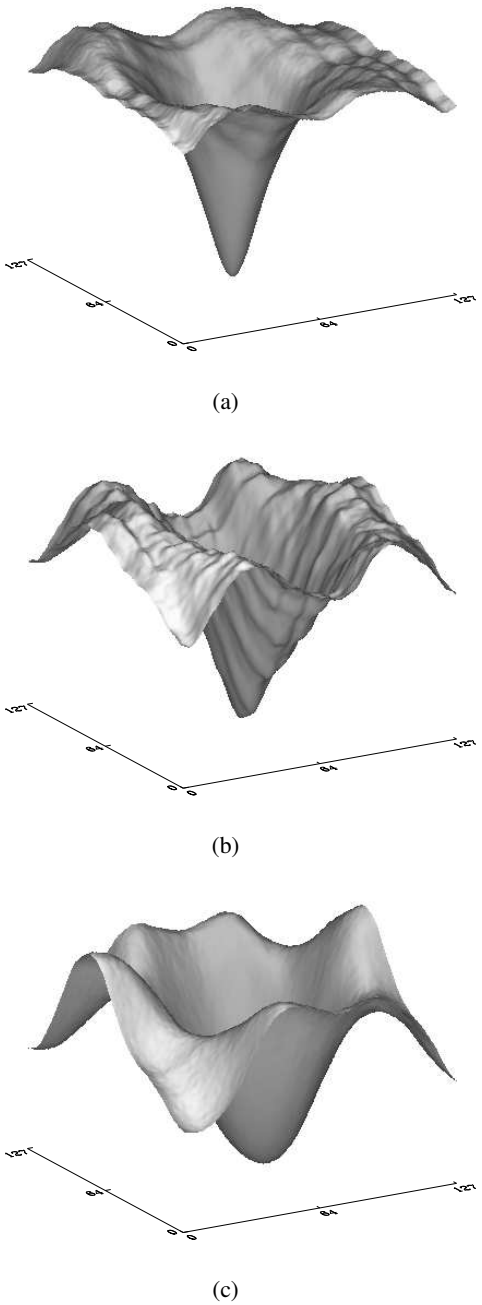

Fig. 4. Global field potential. (a) Contour plot of the total cost of an object graph compared with an undistorted image graph against its location. Both graphs are derived from the same image. (b) The same graphs extracted from different faces. The minimum of this potential gives the rough location of the face in the image. (c) Same potential as (b), but only the lowest frequency level is evaluated. The second minimum is an echo introduced by the wrap-around inherent in the FFT. Due to the fact that our starting grids do not transgress the image boundary we are safe from getting caught there.

E. Complexity Considerations

Many researchers feel that the computational complexity of attributed graph matching is prohibitive. Their argument is mainly based on the NP-completeness of the general subgraph matching problem [22]. However, this *worst case* argument is not applicable (and convincing) for two reasons: i) pattern recognition problems like the one addressed here are solved satisfactorily in almost all situations if good approximations to the best matching solution can be found for the average instance. In particular, we do not require to find the globally optimal solution in the hardest instance which makes the general subgraph matching problem intractable. Neural network systems are especially designed for an efficient search of average case approximations due to their stochastic and analogue nature [23]. Furthermore, ii) the combinatorial explosion of possible matching solutions is dramatically reduced by the constraint that nodes are distinguishable by their attributes and that we are mainly dealing with planar or other low-dimensional graphs (see, e.g., [24] for a discussion of efficient algorithms). The hierarchical representation of our image data suggests that the average running time of our matching

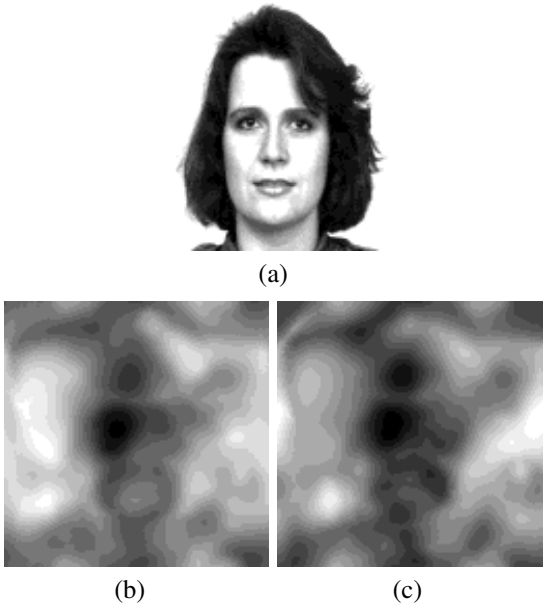

Fig. 5. Local field potential. (b) shows the comparison of a single jet [cross in image (a)] with all the other jets in the same image (gray level proportional to similarity). In (c) the same jet is compared to an image with the same person looking 15° to her right. After the jets are positioned roughly in the first stage, these local potentials are minimized for each model jet with edge costs as a constraint. Note that the potential is smooth near the minimum. The numerous secondary minima do not hurt because the jets have been brought close to their final location in the first stage of the matching.

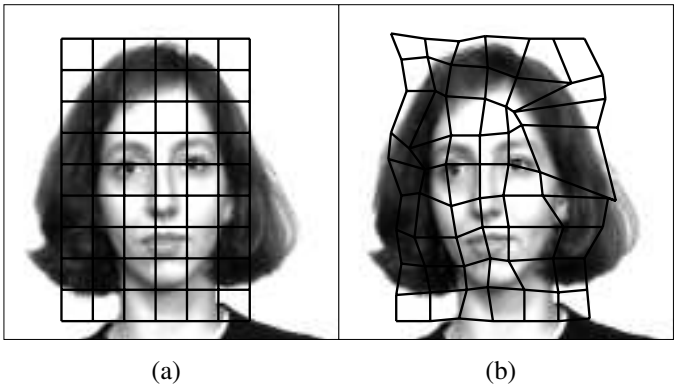

Fig. 6. (a) Example of a stored object, represented by a model graph in the form of a rectangular grid. The vertices are labeled with jets. The overlayed photograph is the one used to store the object. (b) Image that was presented to the system. Graph matching is initialized with an undistorted copy of the object graph. This graph is then first positioned by "global moves," and it is then modified by individual jet diffusion. The grid shows the graph which was accepted as the best match, characterized by $C_{\mathrm{total}} = -67.0$ (the possible optimum for identical images is -70.0).

algorithm scales linearly with the number of resolution levels which translates into a logarithmical scaling behavior as a function of the image size. Rapid convergence times for our algorithm can also be expected from the smooth cost surface as shown in Fig. 5. Local minima which dramatically complicate hard optimization problems (as the travelling salesman problem) do not slow down our diffusion-type graph matching procedure. The actual small convergence times we experience with full-scale images prove that scaling with image size is not a problem.

F. Hardware Implementation

The system as described in the previous section requires a large amount of computing power. However, its design is highly data-parallel, and it is therefore easily implemented on parallel computers.

We use a system based on the transputer, a microprocessor with integrated support for a message-passing, distributed memory MIMD architecture [25]. Our machine consists of 23 transputers, one of which hosts the development system, one is a combined frame grabber/graphics display, and the others are used as a processor farm. We programmed in `occam`, an implementation of the Communicating Sequential Processes model [26]. One of the advantages of this language is the ease with which the real-time parts of the system (e.g., control of the frame grabber) can be implemented. It also permits very good exploitation of the concurrency inherent in the hardware (overlap of computation and communication). Parallel parts of our program reach efficiencies between 0.75 and 0.90 on 21 processors.

The convolution of a 128×128 pixel image with 40 wavelet filters requires less than 7 s. Comparison of an image to a stored object takes between 2 and 5 s on one transputer, depending on parameters. A recognition run, comparing one image to a gallery of 87 stored objects, thus takes about 25 s.

G. Acquisition of Image Galleries

In order to test our program on a large number of images, we acquired a series of images from 87 persons and from a small set of office items. They were obtained with a CCD camera providing a standard video signal, and digitized at 640×512 pixels with 8 bits of resolution by our transputer-based frame grabber. A section 512 pixels square was then low-pass filtered and decimated down to 128×128 pixels, and was stored on disk.

For each person we acquired three images in a standardized setting with constant lighting and magnification factor (see Fig. 7 for examples). Due to the automatic gain control of our camera, however, some differences in contrast level were introduced, for which we compensated to a certain degree by stretching the gray-level histogram of each image.

Our model domain was formed by storing a separate model graph for one standard image per person. The graph was formed by placing a square grid over the object. The grid had 7×10 points (8×10 for the office items) with a spacing of 11 pixels between neighbors. For a single first image this placement was done by hand. For all subsequent images the placement was done by matching the image to the first graph in the model domain by the global move step ($\lambda = \infty$) described in Section III-D.

H. Parameter Settings

Hard boundaries for image resolution as given by Nyquist's theorem result in 7 octaves for our 128×128 images. Our Gaussian envelope functions allow only for about 5 octaves to be investigated without aliasing. The maximum frequency

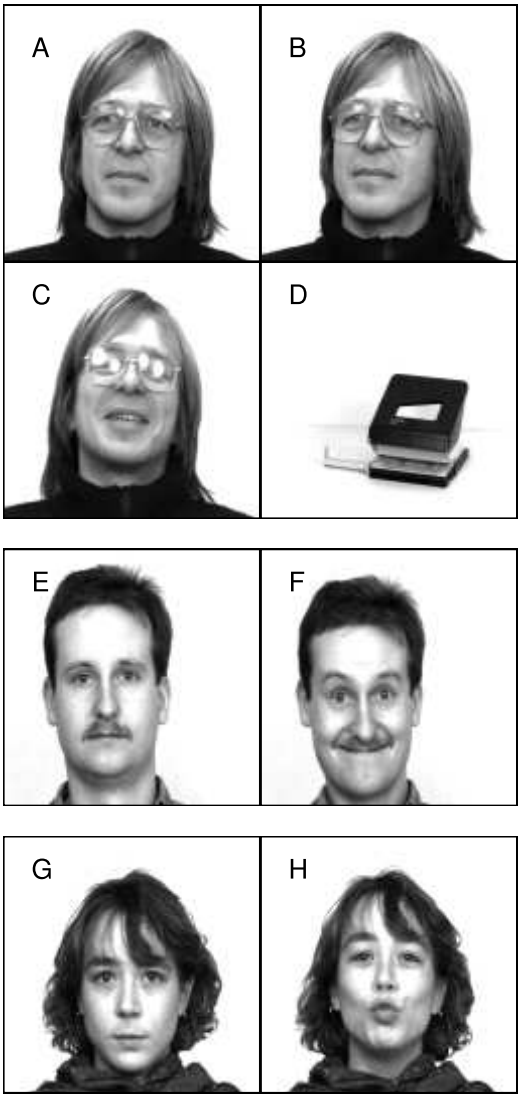

Fig. 7. Examples of the images used. Altogether, four complete galleries of 87 people were taken. (a), (e), (g) Images from standard image database. Subjects were asked to look straight into the camera. (b) In gallery 1 they looked 15° to their right. (c), (f), and (h) In gallery 2, subjects were asked for a different facial expression of their own choice. (d) One of the office items.

must be chosen such as to avoid strong wrap-around effects in the frequency space representation of the transformation kernels. This restricts us to 5 octaves with $k_{\max}$, the center frequency of the highest band, at $3\pi/4$, where π is the bandlimit. However, for our images the actually useful band is much narrower. An object is never as wide as the whole image and single pixels are quite noisy, so that only the range between $\pi/32$ and $\pi/2$ contains information useful for object recognition. To cover this band a trade-off has to be made between the number of frequency levels in one jet and the number of vertices in the graph.

We investigated two values for the spacing of resolution levels, octaves and half-octaves. In the case of octaves, only the choice of $k_{\max} = \pi/2$ is sensible, as the kernels are very wide in frequency space. We used three levels and a grid of 9×13 vertices. For the case of half-octaves, we used five levels, a 7×10 grid, and $k_{\max} = 3\pi/4, \pi/2$, or $\pi/3$; of these choices, $\pi/2$ yielded the best results, by a slight margin. Some comparison runs with a preliminary set of parameters indicated that the choice of octaves does not lead to optimal results, probably due to the fact that we can only use a small number of levels, with the additional problem that the kernels overlap. The bulk of our results have been obtained with half-octaves and $k_{\max} = \pi/2$.

The parameter λ in (9) constrains the deformation of the graph, a smaller value being more permissive of object distortion in the image. On the two galleries we used (see below), we investigated a range of $10^{-6} < \lambda < 10^{-2}$. All values in the range $10^{-6} < \lambda < 3 \cdot 10^{-4}$ gave good results. The value $\lambda = 3 \cdot 10^{-5}$ proved to be optimal and we have used it for the results reported below, except where stated otherwise.

The parameters controlling diffusion should be chosen to find a good approximation to the minimum of (9) in the shortest possible time. For the maximum shift of a single vertex (see Section III-D), a value of half the (original) distance between two neighboring vertices is appropriate. The number of failed moves before diffusion is terminated should be set such that no further improvements occur except in rare cases. We examined the resulting distribution of cost values for a number of different settings of this parameter and determined that a value of 100 was sufficient. A lower value would lead to significant changes in final values of $\mathcal{C}_{\text{total}}$ and yield only little reduction in computing time.

I. Significance Criterion

The process of comparing an image with all models stored in a database always yields a best value for $\mathcal{C}_{\text{total}}$, irrespective of whether or not a corresponding image of the same person is contained in the database. For a recognition mechanism to be of use, we must find some criterion to evaluate the significance of a match.

Our results show that the answer can, with some reliability, be extracted from the statistics of the series of all $\mathcal{C}_{\text{total}}$ values. Let the series $\mathcal{C}_i$ denote these values ordered in ascending sequence, i.e., $\mathcal{C}_i < \mathcal{C}_{i+1}\ \forall i \in \{0, 1, \cdots, N-1\}$, and M_i be the model which gave the result $\mathcal{C}_i$. For the recognition to be significant we expect $\mathcal{C}_0$, which corresponds to M_0, the "candidate" model, to be clearly distinct from all the other values. This has been formalized as follows: If m is the mean and s the standard deviation of the series $\{\mathcal{C}_i \,|\, i = 1, 2, \cdots, N-1\}$ (not containing the candidate model), then we define the criteria for significance and acceptance of a match:

$$\kappa_1 := [r_1 > t_1]\ , \quad \text{where} \quad r_1 := \frac{\mathcal{C}_1 - \mathcal{C}_0}{s} \tag{10}$$

$$\kappa_2 := [r_2 > t_2]\ , \quad \text{where} \quad r_2 := \frac{m - \mathcal{C}_0}{s} \tag{11}$$

with parameters t_1 and t_2. The criteria can be combined to improve performance further:

$$\kappa := [\kappa_1 \vee \kappa_2]\ . \tag{12}$$

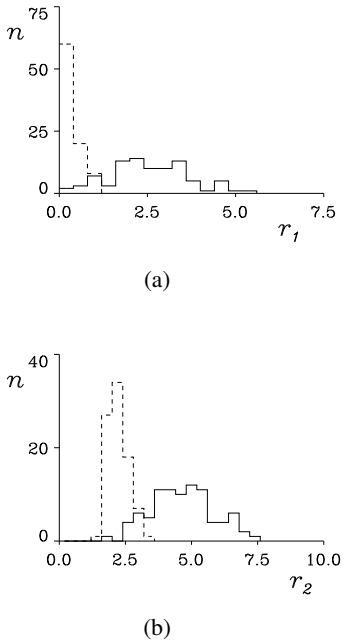

Fig. 8. Histograms for significance of recognition. (a) r_1 as defined in (10). (b) r_2 as defined in (11). Solid line: the correct model is in the database. Dashed line: the correct model is not in the database. Ideally, the distributions should not overlap. The upper limits of the dashed histograms were chosen as acceptance thresholds in order to rule out all false positive recognitions.

Of course, we will expect a tradeoff between ruling out all false recognitions and accepting all correct ones. Altogether there are six cases to be distinguished:

1) A correct model was in the database, was picked as the best match, and the match was judged significant.
2) The database contained no correct model and the best match was rejected.
3) A correct model was in the database, was picked as the best match, but the match was rejected.
4) A correct model was in the database, was not picked as the best match, and the match was rejected.
5) The database contained no correct model, but the best match was accepted.
6) A correct model was in the database but another one was picked as best match, and the match was accepted.

Obviously, cases 1 and 2 are desirable. Cases 3 and 4 are annoying and represent two slightly different versions of false negatives. Cases 5 and 6 are the ones we want to avoid, since they represent two cases of false positives.

J. Recognition Results

In order to set the parameter t_1 and t_2, we looked at the density distribution of the values of r_1 and r_2 defined in the previous section. These distributions were obtained in comparing the 88 models of gallery 1 (persons looking 15° to their right, see Fig. 7) to the standard image database. Fig. 8 shows the resulting histograms. We then selected thresholds such that no false positives (case 5) resulted. The corresponding values are $t_1 = 1.37$ and $t_2 = 3.50$.

TABLE I
RESULTS OF COMPARING TWO GALLERIES (GAL. 1, HEAD ROTATION BY 15°; GAL. 2, GRIMACES; SEE SECTION III-G) AGAINST THE STANDARD IMAGE DATABASE OF THE SAME PERSONS. ALL ENTRIES ARE EXPRESSED AS PERCENTAGES. COLUMNS CORRESPOND TO THE SIX CASES EXPLAINED IN SECTION III-I

Gallery	Criterion	Case 1	2	3	4	5	6
gal. 1	κ_1	86	100	11	2	0	0
	κ_2	83	100	15	2	0	0
	κ	88	100	10	2	0	0
gal. 2	κ_1	79	100	17	3	0	0
	κ_2	80	100	16	3	0	0
	κ	84	100	13	3	0	0

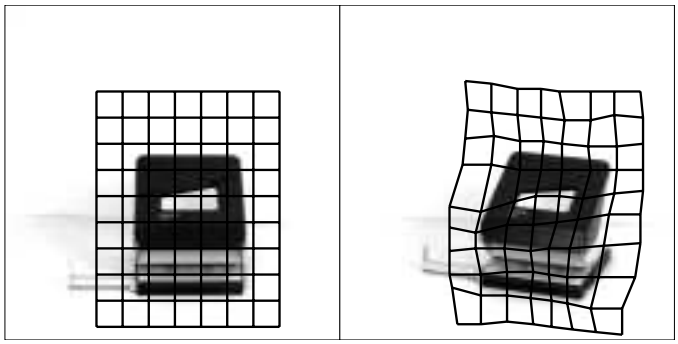

Fig. 9. (a) Example of a stored picture from the database of office items. The rectangular grid shows the positions of the stored jets. (b) A different view of the same object, with the deformed image graph after matching to the correct model graph.

Table I summarizes the results for two galleries, containing 88 and 87 models, respectively. Note that although some matches are declared not significant because of our desire to eliminate false positives, most of these actually found the correct object. Furthermore, just eliminating case 5 also avoids all instances of case 6. In the case of gallery 2 (grimaces), all instances of case 4 result from rotations of the image in the image plane. Note that the thresholds t_1 and t_2 obtained from gallery 1 are also effective for gallery 2. None of the results reported have been distorted by deliberately leaving out images.

In order to demonstrate that our system is by no means specialized to the recognition of faces and can equally well cope with other objects, we investigated a sample of 7 different objects from an office environment (see Fig. 9 for an example). Three different views were taken for each object. One view of each object was selected at random for the database and recognition performance was tested with the remaining 14 pictures. In 13 of these cases (93%) the system yielded the correct classification, while in one case the correct database entry was only in the second place.

The result was achieved with the parameter λ set to 10^{-3} in the cost function (9), corresponding to a higher deformation penality. The need for a larger λ can be understood qualitatively. For faces with their rich structure, jets from adjacent image locations have a higher variability. This prevents the vertices from moving too freely in the image plane during the diffusion process. For office items with their large even surfaces the variability is much lower, which would result in

a higher deformation during diffusion. A larger value of λ and a stiffer graph is therefore more appropriate (see Fig. 9).

IV. Related Work

The purpose of our study is to demonstrate and develop the power of the Dynamic Link Architecture. Nevertheless, it would be interesting to compare our system both conceptually and in its performance to related object recognition systems. However, we do not attempt performance comparisons. In the absence of common benchmark databases these are very difficult to make. (We are prepared, however, to share our database for such tests, and to test our system on databases provided to us by others.) On the conceptual side, a baseline to which our system may be compared are methods using correlation or template matching. In the simplest version, correlation is performed by rigidly comparing stored patterns to the image, either directly [27] or by first comparing a small set of canonical patterns (eigen-patterns characteristic for an object class) and then treating their set of amplitudes as a global feature vector for the recognition proper [28], [29]. These methods have the weakness that they cannot deal even with position invariance. Position (and size) invariance can be achieved by pre-processing the image by taking the magnitude of the Fourier-transform (resp., the Mellin transform) [30], [31], [32]. Systems based on the magnitude of Fourier components, however, have the drawback of being extremely sensitive to image distortions, as produced by perspective transformations, for example. An attempt (A. Goldstein, personal communication) to recognize faces from our galleries with the help of pattern correlation in all possible relative positions, evaluated with our significance criterion, had a recognition rate of 48% for gallery 1 (15^o head rotation) and 51% for gallery 2 (grimaces).

Neural models based on graph matching have been proposed which represent each potential link between image and model by a neuron [14]. Topology constraints can then be implemented in a soft fashion by connections between those neurons. Simic ([33], [34]) has analytically compared that class of graph matching methods to another one that is called elastic matching and that includes our type of approach. He found that graph matching systems that enforce constraints strictly (as does elastic matching) are in their scaling behavior vastly superior compared to networks that represent links (vertex pairings) explicitly by neurons.

V. Discussion

The Dynamic Link Architecture derives its power from a data format based on syntactically linked structures. This capability has been exploited here on three levels. Firstly, when an image is formed in the image domain, the local feature detectors centered at one of its points are bundled to form a composite feature detector (called a jet). A composite feature detector can be shipped to the model domain and can be compared as a whole to other composite feature detectors there. This frees the system from the necessity to train new individual neurons as detectors for complex features before new object classes can be recognized, a major burden on conventional layered systems. Secondly, links are used to represent neighborhood relationships within the image domain and within the model domain. Neural objects thereby acquire internal structure, and their communication can now be constrained to combinations with matched syntactical structure. This forms the basis for elastic graph matching. Finally, the dynamic binding between matched graphs, which in the present context is an unimportant by-product of recognition, will be useful to back-label the image with all the patterns recognized and to build up representations of composite objects and scenes.

This paper describes two versions of the system for object recognition in the Dynamic Link Architecture. One is fully neural and is described in qualitative form to convey the idea (Section II); the other is optimized for ease of implementation on current digital hardware, and is formulated in full detail (Section III). The latter falls short of being fully neural in detail, but it is computationally efficient and avoids some of the complexities of nonlinear dynamics. Although we have chosen human faces as objects to test the performance of our system, we made no specific efforts to optimize its structure just for that application (see Fig. 9), in order to retain its full conceptual generality.

Our system can deal successfully with a large gallery of objects, recognizing them under different circumstances like distortion and rotation in depth. In all retrieval operations the rate of false assignments was below 5%. Furthermore, we have determined a clear criterion on the significance of the recognition process. With this significance criterion all false assignments were rejected *and* at the same time no image was accepted if its corresponding model was temporarily removed from the gallery. This means that the capacity of the gallery to store distinguishable objects is certainly larger than its present size. No limits to this capacity other than a linear increase in computation time have been encountered so far.

This large capacity could be used in a further step to reduce the rate of about 15% where the system failed to identify a stored model: if, for each object, a small number of different views (e.g., from a different perspective or under different lighting conditions) are stored in the model domain as graphs, then graph dynamics can interpolate between and extrapolate from these different views, at the same time keeping the rejection capability constant.

Our object recognition system admittedly is processing-intensive. Most of the time is spent on image transformation and on optimizing the map between an image and individual stored models. Processing demands beyond image transformation grow linearly with the size of the model gallery. This would not slow a system with fully parallel hardware, but it would still be expensive. This expense can easily be reduced in future systems by arranging models in a decision tree and by searching this tree in a sequential fashion. Such a strategy would reduce the processing costs in the search to a scale logarithmic in the number of models. Moreover, early identification of object classes could serve to relabel image points in terms of object-class specific attributes and could thus enormously speed up later matches.

The general system described in Section II inherently generalizes over object position, due to the structure of feature

specific connections between the image and the object domain. Because Gabor-based wavelets are designed to be robust with respect to small distortions, including changes in size and in orientation, the system also generalizes over such changes. Since the wavelet kernels are dilated and rotated versions of each other, it would also be possible to let these connections fully generalize over size and orientation. This would lead, however, to dense connections between nodes and less specific single connections between features, and elastic graph matching would become difficult or impossible. An alternative, which we are pursuing at present [18], is to introduce global parameters for orientation and size and let these diffuse during the matching, in the same way the grid position is diffusing in the first step of the present matching procedure. It is interesting to note in this context that the reaction time of the human visual system grows for objects in unexpected orientations. This suggests that our visual system does not have invariance to orientation to the same degree to which it has translation invariance.

A somewhat more complex issue is invariance with respect to perspective movement. Our system is based on a two-dimensional representation. Essentially different views of a three-dimensional object have to be recognized with the help of multiple views. It is, however, mandatory that recognition be robust with respect to small changes in perspective. A system based on Gabor-based wavelets and topological mapping is ideally suited for this, as we have demonstrated here.

Some of the shortcomings of the present system can be overcome with the help of natural extensions. For instance, visual segmentation can be achieved with the help of mechanisms that induce temporal signal correlations within segments and anticorrelations between segments [7]–[9], [35], and composite objects and scenes can be represented in a hierarchically structured system where the nodes of graphs are themselves smaller graphs to some depth of recursion. Once developed to its full potential, the Dynamic Link Architecture may thus prove to be a natural basis for the implementation of a broad range of interesting cognitive processes.

ACKNOWLEDGEMENT

We are grateful to J.-M. Fellous and H. Haase for transputer software. We would also like to thank C. Anderson and P. König for stimulating discussions.

REFERENCES

[1] C. von der Malsburg, "The correlation theory of brain function," Max-Planck-Institut für Biophysikalische Chemie, Postfach 2841, D–3400 Göttingen, FRG, Internal Report, 1981.

[2] ——, "How are Nervous Structures Organized?" in *Synergetics of the Brain, Proc. Int. Symp. Synergetics*, E. Başar, H. Flohr, H.Haken, and A. Mandell, Eds. Berlin, Germany: Springer, May 1983, pp. 238–249.

[3] ——, "Nervous structures with dynamical links," *Berichte der Bunsengesellschaft für Physikalische Chemie*, vol. 89, pp. 703–710, 1985.

[4] J. Buhmann, J. Lange, and C. von der Malsburg, "Distortion invariant object recognition by matching hierarchically labeled graphs," in *IJCNN International Conference on Neural Networks, Washington*. IEEE, 1989, pp. I 155–159.

[5] E. Bienenstock and R. Doursat, "Issues of representation in neural networks," in *Vision and Vision Research*, A. Gorea, Ed. Cambridge University Press, 1991.

[6] C. von der Malsburg and W. Schneider, "A neural cocktail-party processor," *Biological Cybernetics*, vol. 54, pp. 29–40, 1986.

[7] W. Schneider, "Anwendung der Korrelationstheorie der Hirnfunktion auf das akustische Figur-Hintergrund-Problem (Cocktailparty-Effekt)," Universität Göttingen, 3400 Göttingen, F.R.G., Ph.D. Dissertation, 1986.

[8] O. Sporns, G. Tononi, and G. Edelman, "Modeling perceptual grouping and figure-ground segregation by means of active reentrant connections," *Proc. Nat. Acad. Sci. U.S.A.*, vol. 88, pp. 129–133, 1991.

[9] C. von der Malsburg and J. Buhmann, "Sensory segmentation in oscillatory neural networks," *Biological Cybernetics*, 1991.

[10] C. M. Gray, P. König, A. K. Engel, and W. Singer, "Oscillatory responses in cat visual cortex exhibit intercolumnar synchronization which reflects global stimulus properties," *Nature*, vol. 338, pp. 334–337, 1989.

[11] D. J. Burr, "Elastic matching of line drawings," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 3, pp. 708–713, 1981.

[12] ——, "A dynamic model for image registration," *Computer Graphics and Image Processing*, vol. 15, pp. 102–112, 1981.

[13] C. von der Malsburg, "Pattern recognition by labeled graph matching," *Neural Networks*, vol. 1, pp. 141–148, 1988.

[14] R. Krey and A. Zippelius, "Recognition of topological features of graphs and images in neural networks," *J. Phys. A*, vol. 21, pp. 813–818, 1988.

[15] D. J. Willshaw and C. von der Malsburg, "How patterned neural connections can be set up by self-organization," *Proceedings of the Royal Society, London*, vol. B 194, pp. 431–445, 1976.

[16] ——, "A marker induction mechanism for the establishment of ordered neural mappings," *Philosophical Transactions of the Royal Society, London*, vol. B 287, pp. 203–243, 1979.

[17] A. F. Häussler and C. von der Malsburg, "Development of retinotopic projections — an analytical treatment," *Journal of Theoretical Neurobiology*, vol. 2, pp. 47–73, 1983.

[18] J. Buhmann, M. Lades, and C. von der Malsburg, "Size and distortion invariant object recognition by hierarchical graph matching," in *Proc. Int. Conf. Neural Networks, San Diego*. IEEE, 1990, pp. II 411–416.

[19] D. Field, "Relations between the statistics of natural images and the response properties of cortical cells," *Journal of the Optical Society of America A*, vol. 4, no. 12, pp. 2379–2394, 1987.

[20] J. Jones and L. Palmer, "An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex," *Journal of Neurophysiology*, pp. 1233–1258, 1987.

[21] D. Burr, M. Morrone, and D. Spinelli, "Evidence for edge and bar detectors in human vision," *Vision Res.*, vol. 29, no. 4, pp. 419–431, 1989.

[22] M. Garey and D. Johnson, *Computers and Intractability*. New York: W.H. Freeman and Co., 1979.

[23] R. Durbin and D. Willshaw, "An analogue approach to the travelling salesman problem using an elastic net method," *Nature*, vol. 326, pp. 689–691, 1987.

[24] G. Miller, "Isomorphism testing for graphs of bounded genus," in *Proc. 12th ACM STOC Symp.*, 1980, pp. 218–224.

[25] R. P. Würtz, J. C. Vorbrüggen, C. von der Malsburg, and J. Lange, "A Transputer-based neural object recognition system," in *From Pixels to Features II – Parallelism in Image Processing*, H. Burkhardt, Y. Neuvo, and J. Simon, Eds. North Holland, 1991, pp. 275–294.

[26] C. Hoare, *Communicating Sequential Processes*. Hemel Hempstead: Prentice Hall International, 1989.

[27] T. Kohonen, *Self-organization and Associative Memory*. Springer Verlag, 1984.

[28] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *Journal of the Optical Society of America A*, vol. 4, 1987.

[29] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve procedure for the characterization of human faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, 1990.

[30] D. Cassasent and D. Psaltis, "Position, rotation and scale invariant optical correlation," *Applied Optics*, vol. 15, 1976.

[31] A. Fuchs and H. Haken, "Pattern recognition and associative memory as dynamical processes in a synergetic system. Translational invariance, selective attention, and decomposition of scenes," *Biological Cybernetics*, vol. 60, pp. 17–22, 1988.

[32] ——, "Pattern recognition and associative memory as dynamical processes in a synergetic system. II. Decomposition of complex scenes, simultaneous invariance with respect to translation, rotation, and scaling," *Biological Cybernetics*, vol. 60, pp. 107–109, 1988.

[33] P. Simic, "Statistical mechanics as the underlying theory of "elastic" and "neural" optimizations," *Network*, vol. 1, pp. 89–103, 1990.

[34] ——, "Constrained nets for graph matching and other quadratic assignment problems," *Neural Computation*, vol. 3, pp. 268–281, 1991.

[35] D. Wang, J. Buhmann, and C. von der Malsburg, "Pattern segmentation in associative memory," *Neural Computation*, vol. 2, pp. 94–106, 1990.

For the authors' pictures please refer to the printed version.

Martin Lades (S'90) received the diploma in physics from the University of Erlangen, Germany in 1988, with a topic in the area of optical computing.

After a year as research assistant at the University of Southern California he is currently pursuing the Ph.D. in neural engineering at the University of Bochum, Germany, concentrating on neural networks for scene analysis. Further research interests include optical implementations of neural networks and self-organizing systems.

Jan C. Vorbrüggen (M'91) is a research associate and Ph.D. candidate at the Institut für Neuroinformatik, Ruhr-Universität Bochum.

After receiving a diploma in physics from Bonn University, he joined Prof. v.d. Malsburg's group in 1988, initially at the Max-Planck-Institute for Brain Research in Frankfurt. During this time, he twice spent research visits at the University of Southern California, Los Angeles. His research interests include self-organizing systems, object recognition and scene understanding, and the architecture of parallel computers.

Joachim Buhmann (M'91) received the Ph.D. degree in theoretical physics from the Technical University of Munich in 1988. The dissertation focussed on associative memories.

He held postdoctoral positions at the University of Southern California and at the Lawrence Livermore National Laboratory. He is an associate professor for computer science at the University of Bonn, Germany. His current research interest covers the theory of neural networks and their applications to image understanding and signal processing. The scope of his research covers complex adaptive systems.

Jörg Lange received his diploma in physics in May 1986, and the Ph.D. degree in May 1991, both from the University of Göttingen.

From 1988 through 1990 he participated in von der Malsburg's group at the Computer Science Departement at USC, Los Angeles. He worked on computer models of gravitating many body systems and investigated neural concepts applied to computer vision problems. His doctoral thesis deals with invariant pattern recognition and substructuring 2-D image objects. Presently he is engaged in Visualization of dynamical NMR data at GMD, Bonn.

Christoph von der Malsburg received both the diploma and the Ph.D. in physics from the University of Heidelberg.

He then spent 17 years as staff scientist in the Department for Neurobiology of the Max-Planck-Institute for Biophysical Chemistry in Göttingen. In 1988 he joined the Computer Science Department and the Section for Neurobiology of the Biology Department at the University of Southern California in Los Angeles. In 1990 he took on a position as director at the Institut für Neuroinformatik at the Ruhr-Universität Bochum. His interests are in brain organization, mainly at the level of ontogenesis and function of the visual system.

Rolf P. Würtz received the diploma in mathematics from the University of Heidelberg, Germany in 1986 with a topic in abstract algebra.

He then turned to neural computing and spent two years at the Max-Planck-Institute for Brain Research at Frankfurt, Germany. This time was accompanied by two research visits to the University of Southern California, Los Angeles. Since 1990 he has been a research assistant at the Institut für Neuroinformatik of the University of Bochum. Current research interests include object recognition from real world images and optimization in neural networks.

Wolfgang Konen received the diploma and Ph.D. degrees in physics from the University of Mainz, Germany, in 1987 (experimental physics) and 1990 (theoretical physics), respectively.

Since 1990 he has been with the group of Prof. v. d. Malsburg in the Institute for Neuroinformatik at the University of Bochum, Germany, where he is working on the development of neural architectures. His current interests include learning algorithms and computer vision.

ERRATA

This electronic version corrects the following errors in the printed original:

- In equation 4 the complex exponential should be a 2-dimensional vector.
- In the third paragraph of Section III-D (page 304), the equation should read "$\mathcal{C}_{\text{total}} = \mathcal{C}_v = -\sum_{i \in V} \mathcal{S}_v$."
- In Fig. 8, parts (a) and (b) are interchanged in the printed version.