Int. J. of Pattern Recognition and Artificial Intelligence, 7(4):935-948 (1993)

A Neural System for the Recognition of Partially Occluded Objects in Cluttered Scenes*
— A Pilot Study —

Laurenz Wiskott and Christoph von der Malsburg†
Institut für Neuroinformatik, Ruhr-Universität Bochum
44780 Bochum, Germany, email laurenz@neuroinformatik.ruhr-uni-bochum.de

Abstract

We present a system for the interpretation of camera images of scenes composed of several known objects with mutual occlusion. The scenes are analyzed by the recognition of the objects present and by the determination of their occlusion relations. Objects are internally represented by stored model graphs. These are formed in a semi-automatic way by showing objects against a varying background. Objects are recognized by dynamic link matching. Our experiments show that our system is very successful in analyzing cluttered scenes. The system architecture goes beyond classical neural networks by making extensive use of flexible links between units, as proposed in the dynamic link architecture. The present implementation is, however, rather algorithmic in style and is to be regarded as a pilot study that is preparing the way for a detailed implementation of the architecture.

Keywords: neural networks, dynamic link matching, object recognition, occluded objects, scene analysis.

1 Introduction

The great ease with which our visual system analyses a cluttered dinner table and represents it in terms of the objects present and their spatial relationships is deceptive. Machine vision is still a long way from emulating that ability. Some of the reasons that make this a complicated problem are the variance in the appearance of individual objects, mutual occlusion between them, and the presence of accidental feature configurations.

With the system we are describing here we can demonstrate highly successful recognition and localization of objects in gray level images of complex scenes in spite of extensive mutual occlusion. Objects are made known to the system in a semi-automatic fashion. The performance level demonstrated may be comparable to that of classical computer vision systems. However, our system is distinguished by utmost simplicity and flexibility, which are the hall-mark of neural systems.

Classical vision systems, such as [1] or those reviewed in [2], have several distinct layers of data structures and process types of profoundly differing nature, perhaps characterized

*This work has been funded by grants from the German Federal Ministry of Science and Technology (ITR-8800-H1), from AFOSR (88-0274), and by the Stimulus Programme of the European Community (BRAIN).

†also Dept. of Computer Science and Section for Neurobiology, University of Southern California, Los Angeles, CA 90089.

best by the terms *signal processing* on a low level, *physics* on an intermediate level, and *geometry* or *reasoning* on a high level. It is the goal of the neural approach to compact all of this into the same basic data structure and the same fundamental dynamic process, variance being permitted only on the level of specific data (mainly connection strengths), which are evolved by autonomous adaptation from a simple initial state. All specific structures are to emerge as epiphenomena instead of being rigidly built into algorithms. The ultimate goal of the neural venture is to make the creation of more complex and flexible intelligent systems easier.

The system presented here is part of a larger effort [3] to create a neural architecture that can form a basis for this goal. It enriches conventional neural networks with the equivalent of pointers. These are realized as “dynamic links” with the help of signal synchronization and rapidly switching synapses. Dynamic links are controlled by the process of network self-organization, which favors sparse graphs with a maximum of cooperation between the surviving links.

Our approach is based on object recognition by dynamic link matching [3, 4, 5, 6] which we have demonstrated to be capable of robust recognition of three-dimensional objects under realistic practical conditions [7]. Model graphs representing two-dimensional views of three-dimensional objects are matched to parts of images (several two-dimensional views —or fusion graphs [8]— would be necessary for the full representation of a three-dimensional object). Model graphs are semi-automatically formed with the help of images of the object on varying background. No programming is needed to deal with novel objects, irrespective of variations in object type. Objects are recognized in composite scenes in spite of partial occlusion by other objects. As a result of the matching process, parts of the scene are labeled by the objects recognized. The depth sequence of the objects can be read off the occlusion relationships.

Our system uses a data format — fields of units with local receptive fields in the form of oriented edge detectors on different resolution levels — that is closely related to that of vertebrate visual cortex [9, 10]. The system as presented here is, however, not fully neural in detail, taking advantage of the specific capability of computers to represent high precision floating point numbers and free addressing. In the fully neural implementation of the dynamic link architecture [3], there would be only one homogeneous data format — neural activity signals and synaptic strengths varying on different time scales — and only one type of dynamical process, network self-organization. Models would be part of an associative memory for graphs in which they can be dynamically activated (as has been demonstrated in [4]). The image domain would be coupled to the model domain with the help of feature-preserving connections. During a match process, a sparse subset of these would be dynamically activated to establish a topographical mapping between the model graphs and a region of the image occupied by the corresponding object, as has been demonstrated in [6]. In regions of overlap between objects in the image, link dynamics would take the form of a territorial fight between the relevant model graphs, and occluded objects would achieve only a partial match. The work we are reporting here is in the form of algorithmic caricatures of these network dynamical processes. (For instance, in the matching process, we are kinematically restricting the connectivity between a model graph and the image to strictly topographical and rigidly non-distorting connection patterns that have only the freedom of a collective shift in the image and partial disconnection to account for occlusion.)

This formulation has the advantage of algorithmic simplicity but pays a price in terms of inflexibility (when dealing with object distortions, for instance) and the need for algorithmic design, compromising the final goal of neural systems. However, the system design goes one important step beyond matching of unstructured feature sets, employing a neural description in terms of topographically linked sets of features. This seems to be of great importance for a system dealing with a topographically organized visual world (and here in particular in attacking the problem of mutual occlusion).

2 The System

2.1 Data Structures

Let $I(\vec{x})$ be the gray level distribution of the input image. Our image representation is based on the complex convolution

$$(\mathcal{W}I)\left(\vec{k},\vec{x}_0\right) := \int \psi_{\vec{k}}\left(\vec{x}_0-\vec{x}\right) I\left(\vec{x}\right)\, d^2x = \left(\psi_{\vec{k}} * I\right)\left(\vec{x}_0\right)\ , \tag{1}$$

the operator $\mathcal{W}$ symbolizing the convolution with a family of wavelet kernels $\psi_{\vec{k}}$,

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

which are parameterized by the two-dimensional vector $\vec{k}$ (the notation $\vec{k}^2$ or $\vec{k}\vec{x}$ refers to the scalar product). These kernels have the form of a plane wave restricted by a Gaussian envelope function, $\vec{k}$ determining wavelength and orientation of the wave. We have corrected for the DC value of the kernels (that is, a non-vanishing value of the integral $\int \psi_{\vec{k}}\left(\vec{x}\right) d^2\vec{x}$), as described in [7], although it is negligible for the large window width $\sigma = 2\pi$ we are using.

For our images of 128×128 pixels with 256 gray levels (obtained by subsampling from 512×512 after low-pass filtering) we employed a discrete lattice of $\vec{k}$ values

$$\vec{k}_{\nu\mu} = \begin{pmatrix} k_\nu \cos\phi_\mu \\ k_\nu \sin\phi_\mu \end{pmatrix},\quad k_\nu = \pi 2^{-\frac{\nu+2}{2}},\ \phi_\mu = \frac{\pi\mu}{8},\quad \nu \in \{0,1,2\},\ \mu \in \{0,\ldots,7\}. \tag{3}$$

The wavelet transform yields a complex number of oscillating phase as a function of position. This leads to local similarity optima when matching stored wavelet features to the transform of an image. Instead of the linear wavelet transform we therefore use the absolute values

$$\mathcal{J}_{\nu\mu}\left(\vec{x}_0\right) := \left|(\mathcal{W}I)\left(\vec{k}_{\nu\mu},\vec{x}_0\right)\right|\ . \tag{4}$$

It is this array of numbers that we use as feature vector to characterize the image at location $\vec{x}_0$. We will refer to this vector as a "jet".

For two jets $\mathcal{J}$ and $\mathcal{J}'$ a similarity value $\mathcal{S}(\mathcal{J},\mathcal{J}')$ is defined as the normalized scalar product (considering jets simply as vectors), raised to the fourth power (the motivation for which will be discussed later):

$$\mathcal{S}\left(\mathcal{J},\mathcal{J}'\right) := \left(\frac{\mathcal{J}\cdot\mathcal{J}'}{\|\mathcal{J}\|\ \|\mathcal{J}'\|}\right)^4. \tag{5}$$

This is used when comparing jets taken from two images or from an image and an object model.

The total system is composed of an image domain I and a model domain M. Images and object models are represented as labeled graphs. Labels on the nodes are jets. Edges connect neighboring nodes. In the work reported here we have treated the edge labels implicitly by just storing the positions of nodes as two-dimensional coordinates and creating distance information from coordinates whenever needed. Thus the natural substrate to represent an image graph is based on a jet and a pixel index pair for each of the 128^2 pixels. Permanently stored model graphs are formed from sparse subsets of pixels forming a square grid with a lattice spacing of 7 (or, in some cases, 5) pixels, linked by up to four edges per node.

To specify the state of the scene interpretation system completely it is necessary in addition to represent which regions of the image have been recognized by which model graphs and what the occlusion relations between the objects are. In our system we describe the relation of the model domain to the image domain with the help of a few binary variables that decide on the recognition status of a model and the visibility or occlusion of its individual nodes, plus a single position vector for the placement of the model graph in the image. These variables will be introduced in the next section.

2.2 Model Graph Formation

For the formation of a model graph, three images are taken, each with the object in identical position on a different background. In two of the images, the background is formed by a horizontal or vertical lattice of black and white stripes approximately 3.5 pixels wide. The third image has a white background. A square lattice of points with a spacing of 7 pixels (or, for some delicate objects, with 5 pixels) is selected in the image. For each of the selected lattice points $\vec{x}_i$ the similarity $\mathcal{S}(\mathcal{J}^h(\vec{x}_i), \mathcal{J}^v(\vec{x}_i))$ between jets taken from the images with horizontal and vertical stripes, respectively, is computed according to equation 5. These similarity values are high within the area covered by the object and low over the background. (Similarities for nodes inside an object but near its border are lower to the extent their wavelets reach over into the background.) Now all lattice points are selected for which this similarity is above a threshold (which we chose in the range $.12^4$ – $.33^4$, depending on the object. A more sound treatment of occluding boundaries will have to supplant this ad hoc treatment later). The procedure may produce several mutually disconnected graphs. From among these, the largest is picked. (Two graphs are connected if their minimal distance is one lattice spacing.) For the graph thus obtained, lattice points are labeled with the jets taken from the third image, which had been taken with a blank background. The resulting graph is stored as a model graph in memory. This graph formation process has the advantage of positioning the nodes automatically in those regions of the object which are least sensitive against background variations. For example, the process avoids placing nodes in openwork regions of an object. We also successfully experimented with other background textures, not only horizontal and vertical stripes.

2.3 Matching an Object

When matching a model graph against the image of a scene, a replica of its set of node positions $\{\vec{x}_i^M\}$ is placed in the image, creating the set of points $\{\vec{x}_i^I\}$, where $\vec{x}_i^I = \vec{x}_i^M + \vec{x}^s$ with an offset vector $\vec{x}^s$ common to all nodes. The graph formed by the set of points $\{\vec{x}_i^I\}$ is called the *image graph*. The model graph is compared to the image graph with offset $\vec{x}^s$ in terms of the cost function

$$\mathcal{C}_{total}(\vec{x}^s) := -\frac{1}{N_V} \sum_{i \in V} \mathcal{S}(\mathcal{J}_i^I, \mathcal{J}_i^M), \tag{6}$$

where $\mathcal{J}_i^I$ is the image jet taken at position $\vec{x}_i^I$, V is the set of visible nodes (visibility being defined below), and N_V is their number. Now, the matching cost (6) is minimized by varying $\vec{x}^s$. First, the offset is taken through all points on a square grid with a spacing of five pixels for which the replica of the model graph lies entirely within the image. Around the lattice point with minimal cost a better minimum is then found for a finer lattice with spacing 1. The resulting image graph is taken as the candidate match. In distinction to the work reported in [7], here we do not consider distortions of the image graph with respect to the model graph.

When an object is occluded to a large extent, the total cost of its match is degraded by the many nodes that come to fall on the image of other objects and that correspondingly have only average similarity values. It is decisive that this correct match cannot be outdone by a false match in some region of the image picked such that many nodes have above average similarity. In order to favor the correct match, we give its correctly matching nodes an advantage over the only averagely fitting nodes by raising the inner product in (5) to the fourth power. (In our experiments, analysis of correctly interpreted scenes shows that correctly matched nodes have a mean similarity value of $\mathcal{S}_c = 0.64$, whereas with graphs matched to scenes not containing the object nodes have a mean similarity of $\mathcal{S}_w = 0.35$. Random pairs of jets have a mean similarity of $\mathcal{S}_r = 0.32$.)

2.4 Scene Interpretation, Algorithm One

In this first, simple, scene interpretation algorithm, each model graph is matched separately to the image to decide if and where it fits and to what extent it is occluded. The algorithm has the advantage that there is no need for all objects in the scene to be known to the system. The algorithm examines all graphs in the model domain. First, a graph is matched to the image. Then, all nodes under a threshold of 0.52 for $\mathcal{S}$ are marked as occluded. (This parameter could be obtained automatically by collecting a similarity histogram for a large set of model jets and image jets. According to our experience, this histogram tends to be bimodal, and the threshold can be set near the minimum between the modes.) Since we assume that occlusion occurs for coherent regions, the algorithm proceeds to revise the occlusion decision for each node according to its neighborhood in the graph. For occluded nodes that have a majority of visible neighbors within the model graph the decision is reversed. Likewise for visible nodes that have a majority of occluded neighbors. In this way, all nodes are visited repeatedly, in the arbitrary sequence inherent in our graph administration system, until no more changes occur.

If the part of the graph now considered visible has average node similarity better than 0.61 and a visible area of at least 1300 pixels, it is accepted for the scene. (Although not the point of this algorithm, it is convenient for display purposes to order the accepted models in depth according their mutual occlusion indices, which are computed as explained in the next section.)

2.5 Scene Interpretation, Algorithm Two

For this algorithm to work, there must be models for all objects in the scene. Posing such a constraint has the advantage that the relative occlusion relations can be determined and used for a more reliable interpretation of the scene. For two graphs A and B, this relation will be characterized by the "occlusion index" $\mathcal{Q}_{AB}$. When it is computed, the system may already have decided that third object(s) are occluding parts of A or B so that only part of their graphs are visible in the image. The visible region of a particular model's image graph is then the set of all pixels that lie within squares of size d centered around visible nodes of the graph (d being the spacing of nodes in the model's graph, 7, or sometimes 5, pixels).

Let us define a similarity function $\mathcal{S}_A(\vec{x})$ for a model A, defined over the whole image after the model has been matched. For each pixel $\vec{x}$ of the image it gives the similarity value of the nearest node, and zero outside the visible region of the graph. Further, let R be the region of overlap between the visible parts of A and another model B. Then the occlusion indices of A with respect to B is defined as

$$\mathcal{Q}_{AB} = \sum_{\vec{x} \in R} \mathcal{S}_A(\vec{x}) - \mathcal{S}_B(\vec{x})\,. \tag{7}$$

For this we have the relations $\mathcal{Q}_{AB} = -\mathcal{Q}_{BA}$, $\mathcal{Q}_{AA} = 0$, and for graphs A and B without overlap we have $\mathcal{Q}_{AB} = 0$. If $\mathcal{Q}_{AB} > 0$, A is occluding B, and if $\mathcal{Q}_{AB} \geq 0$, A is said not to be occluded by B.

To start scene interpretation, all stored models are first matched to the image. Each model will yield a "match", that is, the graph placement fully inside the image with minimal cost (6). These graph placements will not be changed during all of the following steps. The effect of the following iterative process will result in the graduation of some of the models in two steps, first to the status as candidate, then to the status of being accepted. The process has the following steps:

Step 1 Turn those models into candidates (i) for which the average node similarity is better than 0.45, (ii) for which the visible region is bigger than 1300 pixels and (iii) which haven't yet been accepted.

Step 2 Stop if there are no candidates.

Step 3 Accept one of the candidates: First, determine the mutual occlusion indices for all pairs of candidates. Then select the "unoccluded" candidate(s), that is, those for which all occlusion indices are non-negative. If there are several, select the one with lowest best cost (6). If there is no candidate for which all occlusion indices are non-negative, pick the one that is least occluded, that is, for which the least occlusion index is largest.

Step 4 Insert the model just accepted in the depth sequence of all accepted models. For this, the new model has to work its way from the back of the list forward. The model is advanced one step if its occlusion index relative to the model in front of it is non-negative. (This comparison is based on the overlap of all of the new model with the visible part of the model in front of it.) The advancement stops as soon as the new model hits one with which it has a negative occlusion index.

Step 5 Now the occlusion status of the nodes is updated. Those in the newly accepted graph are occluded by the territory of the models in front of it. The nodes of the model graphs now put behind are occluded by the new one. Also, the territory of the newly accepted graph is declared invisible for all as yet uncommitted model graphs, whose visible areas and cost values are modified correspondingly, see equation 6. After that, proceed with Step 1.

2.6 Implementation

Our system was developed as part of a larger package of C++ objects for object recognition, including, for example, dynamic link matching, and facilities for object recognition invariant with respect to scale and rotation (M. Lades, forthcoming publication). It was developed in cooperation with M. Lades, partly on the basis of the LEDA library for handling graphs that was developed by S. Näher at the Max–Planck–Institut für Informatik in Saarbrücken.

The program currently runs on Sun and NeXT workstations. The wavelet transform of an image takes about 3 minutes on a NeXT workstation (Motorola 68040 running at 25 MHz). After the transformation, scene interpretation takes about 280 seconds, of which 10 seconds are taken for the loading and initialization of the graphs, 250 seconds for the matching of the 13 model graphs, and 20 seconds for the interpretation proper. We have made no attempt to optimize the program with respect to time.

With respect to computational complexity, the following scaling laws are relevant. Our wavelet transform is based on an FFT and correspondingly it scales with $n \log n$, where n is the number of pixels. Matching a single model against the image scales linearly in model size in terms of number of nodes. Scaling with image resolution is not a problem. Since we are testing for a match at a hierarchy of grid points (2%–4% of all pixels in our case) the computational complexity only grows logarithmically. Scene interpretation scales linearly with the number of models in the gallery and likewise linearly in the number of objects in the scene.

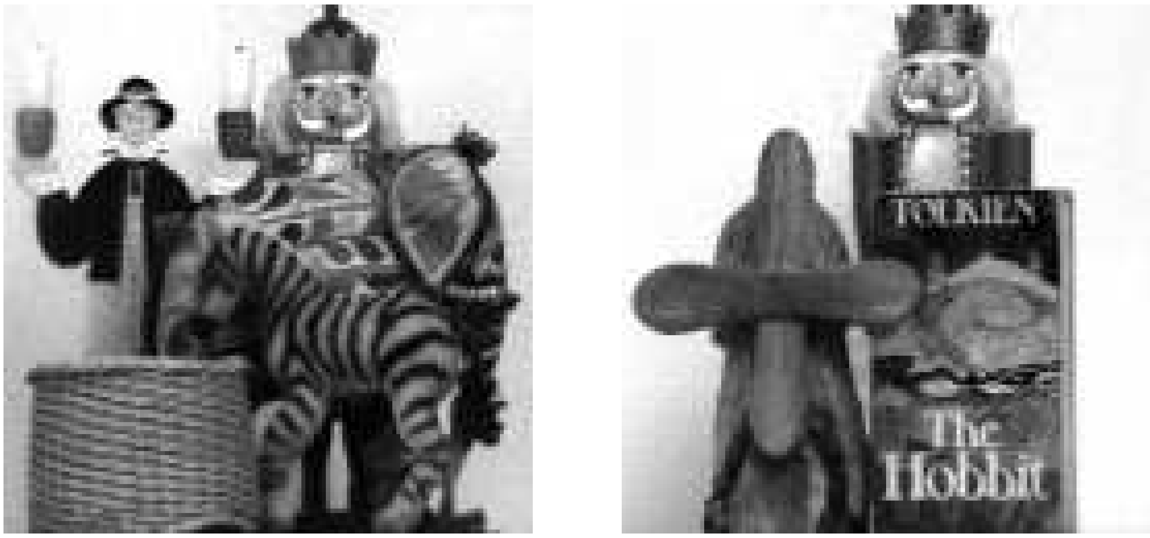
With appropriate hardware, it would be possible to attain real time performance for the purposes of an autonomously acting system, since the computationally intensive parts of the algorithm, wavelet transformation and model match, are amenable to parallel formulation. A moderately parallel implementation of a similar system in multi-transputer hardware has been described previously [7]. In an extreme case, each wavelet component in each image position could be computed simultaneously, and all models could be matched simultaneously against many positions in the image, such that only the time complexity logically inherent in comparing the different trial matches would need to be expended sequentially. Finally, even all jet comparisons in a single match could be done in parallel and the results would need just a few sequential steps to be collected into global match similarities. Algorithm 2 for the final scene interpretation after model matches has a few inherently sequential steps, which, however, need little time individually if the occlusion relations are computed in a parallel fashion, and the number of which is negligible in comparison to the inherently sequential steps involved in model matches.

3 Experiments

3.1 The Scenes

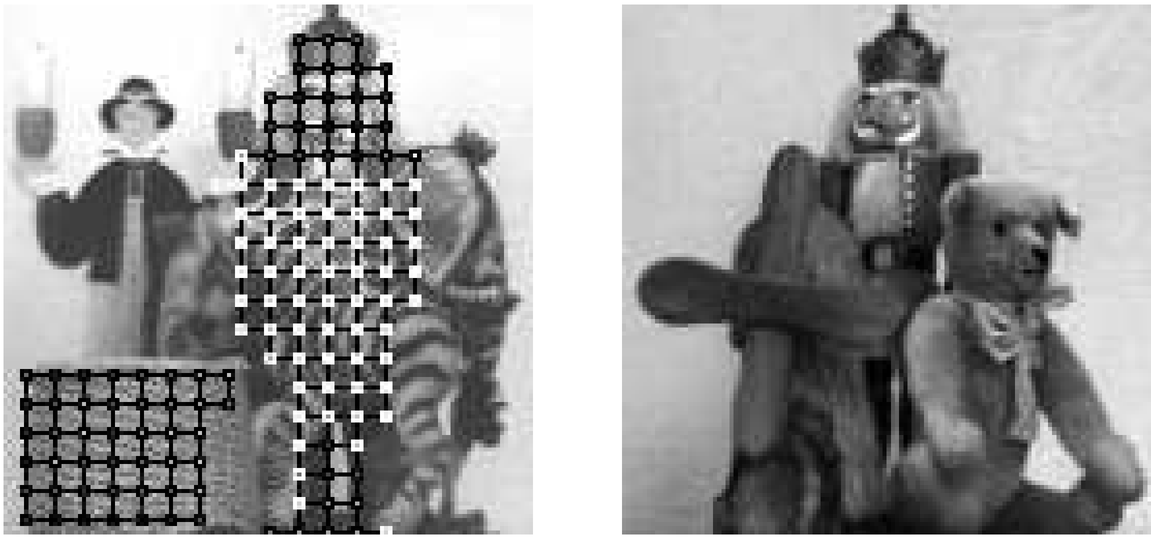
Our pictures are of size 128×128 pixels and have 256 distinguishable gray levels. They are derived from 512×512 pixel frames taken with a CCD camera, by low-pass filtering and subsampling. We took pictures of 30 scenes composed of 3 – 6 objects each. Scenes were taken at approximately 200 cm of distance, at which pictures are 42 cm wide. Distances to individual objects vary up to 10 cm, causing size variance of up to 5% relative to the images used for model graph formation. Individual objects were mostly presented in approximately the same orientation and perspective, although we also did individual experiments with slightly rotated objects, see Fig. 2b. To avoid visible shadows, we sometimes illuminated scenes with two light sources (although during model graph formation there had been only one light source). This fact is relevant for the issue of robustness to illumination, see below. We made sure the objects were visible to a certain minimal extent, 1300 pixels, corresponding to an area of approximately 140 cm^2.

We created a gallery of 13 toy objects of which all scenes are composed. The objects are called *basket, bear, book, box, candleman, candlewoman, clock, elephant, glass-of-marbles, nutcracker, rattle, windmill,* and *zebra*. The objects actually used in the experiments on which we report here represent a slight selection. We have excluded objects that are too small, that have too little inner structure (although we included *basket, rattle*, and *glass-of-marbles* in spite of their poor inner structure), and that are not compact (although we included *clock* with its two holes and *zebra* with its thin legs). The problem with small objects was that with our resolution the corresponding model graphs contain too little information and yield good cost in wrong places. This is similarly the case with objects that are too homogeneous. For objects with openwork structure (e.g., letter scales that we tried) our extended receptive fields caused problems, being too sensitive to the background. We will return to these restrictions in the discussion.



a) b)

Fig. 1: Two of the 30 scene images. **a)** contains *zebra, basket, elephant, candleman*, and *nutcracker*, **b)** contains *windmill, book*, and *nutcracker*. There are altogether 13 models in the gallery, and 121 objects in the scenes, each scene containing between three and six objects. The resolution of the images is 128^2 pixels with 256 grey levels.



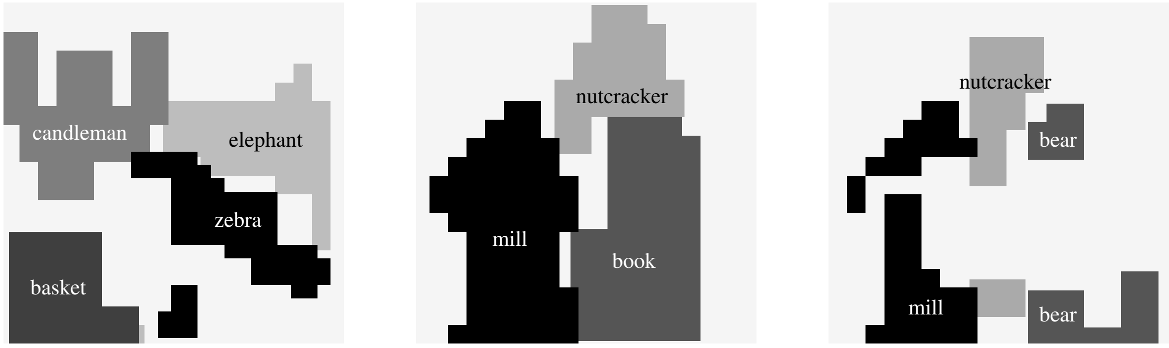
a) b)

Fig. 2: **a)** Picture of the image graphs for *basket* and *nutcracker* as matched in scene 1a. Black and white frames denote visible and occluded nodes, respectively. The state of visibility was determined by Algorithm 2. Center pixels of nodes code for similarity of the model jets to corresponding image jets, from white (0) to black (1). **b)** One example of a scene with rotated objects ($\approx 20^\circ$) not included in the statistical investigation. For interpretation see Fig. 3c and 4c.

3.2 Performance of the System

Two of the scenes used in our experiments are shown in Fig. 1. Fig. 3 shows interpretations obtained with Algorithm 1, and Fig. 4 with Algorithm 2.

Algorithm 1 gave the following performance. From a total of 121 objects, 24 were not recognized correctly, leaving 80% recognized correctly. Only 2 objects were erroneously accepted. Some decisions regarding occlusion were unsatisfactory (see, for instance, the candleman in Fig. 3a), which is not surprising since no interactions between objects were taken into account (and the point of Algorithm 1 has been recognition only, anyway).

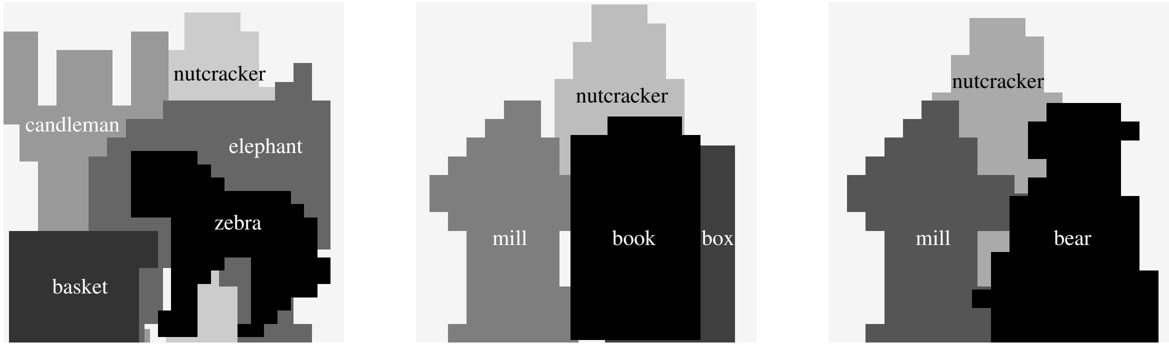


a) **b)** **c)**

Fig. 3: Interpretations of the scenes in Fig. 1a, b, and 2b respectively, by Algorithm 1. Visible regions of the matched model graphs are shown, from front (black) to back (light). **a)** The algorithm recognized *zebra, basket, candleman* and *elephant* in scene 1a, but it missed the *nutcracker*, not finding the lower part under the zebra and discarding the identified head region as too small in area. For the zebra, large parts are interpreted as occluded, because of perturbation of inner jets by overlap with the background. **b)** Scene 1b is interpreted correctly. Altogether 80% of the 121 objects were recognized correctly while 2 models were accepted erroneously by this algorithm. **c)** shows the interpretation of scene Fig. 2b. All objects are recognized correctly but again large parts are interpreted as occluded. Although we did not investigate the robustness of our system against rotation in depth systematically several examples like this suggest that this algorithm might be robust up to approximately 10°.

Algorithm 2 produced 21 completely correct interpretations from among the full set of 30 scenes. For 3 scenes it made errors regarding the occlusion order, always for pairs of weakly overlapping objects. In the remaining 6 scenes, 3 models were accepted although the corresponding object was not present, and 4 objects that were present were not recognized. Among the latter, 2 were matched at the wrong position and 2 objects were matched correctly but were rejected on the basis of too bad a cost value. In all, 96.7% of the objects are recognized correctly and with confidence.

Some of the errors committed by the system are instructive. In the total set of experiments, there were only 2 cases in which the best match for a model graph in the scene was found in the wrong place. One case concerned the *glass-of-marbles*, which has little internal structure and is partially transparent. In the other case, *candlewoman* was matched to the fairly similar *candleman* while the proper object was heavily occluded. To deal with this type of error, one could produce several matches for each model. The system would then have to manage more matches, but it would be more likely to find correct matches (and would be equipped to match multiple instances of the same object type).



a) **b)** **c)**

Fig. 4: Interpretations of the scenes in Fig. 1a, b and 2b by Algorithm 2. **a)** In scene 1a, all objects and their occlusion relations have been recognized correctly. **b)** In scene 1b, a box (rightmost model) was erroneously recognized. Besides, the book mistakenly was put in front of the windmill. Altogether, 96.7% of the 121 objects were recognized correctly, and 3 models were accepted erroneously by this algorithm. **c)** The third example, scene 2b, was also recognized correctly. Algorithm 2 seems to be more robust against rotation in depth than the first. Up to approximately 15° might be of little effect. Both algorithms could be improved in this respect if graph distortion were permitted, see [7].

4 Discussion

What we are describing here was undertaken as a pilot study, to identify the specific problems involved in analyzing cluttered scenes into the component objects and their relationships. Our previous work had been limited to scenes with single objects [7, 11, 12]. The success of the extremely simple system described here, with all its obvious deficiencies, came to our own surprise. This success is mainly due to a concept of object recognition based on dynamic link matching. In contrast to the classical mechanism of pattern recognition on the basis of unstructured sets of features, dynamic link matching is very robust to occlusion, and it furnishes detailed information as to where the identified object is located in the image. This is essential for back-labeling the scene with the recognition information and with occlusion relations.

One strength of our system is that in order for it to be extended to new objects it needs neither programming, as is necessary in the classical Artificial Intelligence style, nor does it need the extensive runs of statistical estimation that are required with many neural network systems. Instead, new objects are simply shown to the system once.

It is instructive to discuss some of the weaknesses of our system. Already during model graph formation it becomes evident that our treatment of occlusion boundaries needs modification. The similarity values between corresponding points on the pictures with one object and different background should be near to 1 inside the object and should drop sharply at the boundary of the object. Instead, they are already degraded for some distance inside the boundary, due to the increasing overlap of receptive fields of jet components with the varying background. Correspondingly, when a scene is interpreted, the quality of the match of an object is degraded by this sensitivity of the jets near the occlusion boundaries to the background. (It is this particular difficulty that forced us to discard small and openwork-type objects.) Instead, the precise shape of the occlusion boundary should be able to help to identify an object. This could to a large extent be achieved by simply improving the resolution of our system and correspondingly by putting more emphasis on smaller high-resolution jet components. (With better resolution we could have included objects that had little structure on the resolution level employed.) However, a more principled approach seems to be required to treat the problem properly.

Our system is extreme in the sense of building high-level models directly by gluing together lowest-level features, which are derived directly from gray-level profiles. However, the feature similarity function of our system is insensitive to changes in contrast and brightness. It is clear that mechanism for medium-level structural description (edge or corner detection, shape from X, stereo depth) would be of great help. We will make efforts to include these in future systems, incorporating them in a homogeneous neural architecture. It would probably be wise to postpone a direct comparison of our system to the vertebrate visual system until mid-level features have been incorporated (although the types of features that we do use — wavelets— correspond closely to the form of receptive fields found in early visual areae [9]). The speed and the robustness with which our visual system can recognize individual objects can probably only be explained with the help of mid-level structural descriptions.

Another weakness is the somewhat artificial way of forming model graphs by taking several pictures with different background and the object in the same position, and by manually controlling two parameters. Once the problem with similarity degrading near the occluding boundary mentioned above is solved, the obvious procedure for model graph formation is the visual tracking of the object in a complex scene under perspective movement, which would automatically vary the background. This capability has been shown in [8], which describes the automatic construction of a fusion graph from multiple image frames. (That work also demonstrated the successful recognition of objects by dynamic link matching in spite of gross perspective changes.)

Although in the bulk of our experiments we chose not to permit relative distortion between image graph and model graph and merely permitted ca. 5% size variation, some experiments with rotation in depth by 10–20^o were performed (see Fig. 2b). These showed that the system is fairly robust with respect to distortions. Nevertheless, it would fail with substantial rotations in depth or in the image plane or size changes. In [12] it was shown how size variations can be handled by a minor system extension, rotation in the image plane being amenable to a similar treatment, and in another system [8] similar in style arbitrary rotations of objects in depth were successfully dealt with.

Finally, the pilot study presented here must still be reformulated in full neural detail, that is, entirely in terms of differential equations for neural signals and dynamic link variables. Much of what we have rigidly fixed here in terms of specific algorithms must then emerge on the basis of general principles of self-organization. The difficulty will be to safeguard the system against a host of unwanted instabilities. The eventual advantage will be a system that can flexibly and autonomously deal with new situations and that will be weaned from the continuous intervention of a programmer.

Acknowledgements: We wish to thank Martin Lades for his contributions to the development of the program package and Wolfgang Konen for useful discussions.

References

[1] A.R.Hanson and E.M.Riseman. Segmentation of natural scenes. In A.R.Hanson and E.M.Riseman, editors, *Computer Vision Systems*, pages 129–163. Academic Press, New York, 1978.

[2] P. Suetens, P. Fua, and A.J. Hanson. Computational strategies for object recognition. *ACM Computing Surveys*, 24:5–61, 1992.

[3] C. v.d.Malsburg. The correlation theory of brain function. Internal report, 81-2, Max-Planck-Institut für Biophysikalische Chemie, Postfach 2841, 3400 Göttingen, FRG, 1981.

[4] C. von der Malsburg. Nervous structures with dynamical links. *Ber. Bunsenges. Phys. Chem.*, 89:703–710, 1985.

[5] E. Bienenstock. Neural like graph-matching techniques for image processing. In W. von Seelen, G. Shaw, and U.M. Leinhos, editors, *Organization of Neural Networks.* VCH Verlags-Gesellschaft, Weinheim, Germany, 1988.

[6] E. Bienenstock and C. von der Malsburg. A neural network for invariant pattern recognition. *Europhysics Letters*, 4:121–126, 1987.

[7] Martin Lades, Jan C. Vorbrüggen, Joachim Buhmann, Jörg Lange, Christoph v.d. Malsburg, Rolf P. Würtz, and Wolfgang Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans. Comput.*, 42(3):300–311, 1992.

[8] K. Reiser. Learning persistent structure. Doctoral thesis, res. report 584, Hughes Aircraft Co., 3011 Malibu Canyon Rd. Malibu, CA 90265, 1991.

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

[10] R.L. DeValois and K.K. DeValois. *Spatial Vision.* Oxford Press, 1988.

[11] Joachim Buhmann, Jörg Lange, and C. von der Malsburg. Distortion invariant object recognition by matching hierarchically labeled graphs. In *IJCNN International Conference on Neural Networks, Washington*, pages I 155–159. IEEE, 1989.

[12] J. Buhmann, M. Lades, and C.v.d.Malsburg. Size and distortion invariant object recognition by hierarchical graph matching. In *Proceedings of the IJCNN International Joint Conference on Neural Networks*, pages II 411–416, San Diego, June 1990. IEEE.