Neural Networks, Vol.7, No.6/7, p.1019-1030, 1994

A Fast Dynamic Link Matching Algorithm for Invariant Pattern Recognition*

Wolfgang K. Konen
Thomas Maurer
Christoph von der Malsburg†
Institut für Neuroinformatik, Ruhr-Universität Bochum, FRG

Abstract

When recognizing patterns or objects, our visual system can easily separate *what* kind of pattern is seen and *where* (location and orientation) it is seen. Neural networks as pattern recognizers can deal well with noisy input patterns, but have difficulties when confronted with the large variety of possible geometric transformations of an object. We propose a flexible neural mechanism for invariant recognition based on correlated neuronal activity and the self-organization of dynamic links. The system can deal in parallel with different kinds of invariances such as translation, rotation, mirror-reflection and distortion. It is shown analytically that parts of the neuronal activity equations can be replaced by a faster, but functionally equivalent, algorithmic approach. We derive a measure based on the correlation of activity which allows an unsupervised decision of whether a given input pattern matches with a stored model pattern ("what"-part). At the same time, the dynamic links specify a flexible mapping between input and model ("where"-part). In simulations, the system is applied to both artificial input data and grey level images of real objects.

Keywords: *dynamic links, pattern recognition, self-organization, correlation.*

Requests for reprints should be sent to:
Thomas Maurer, Institut für Neuroinformatik,
Universitätsstr. 150, 44780 Bochum, FRG
Phone: +49-234-700-7996
Email: maurer@neuroinformatik.ruhr-uni-bochum.de

Running Title: "A Fast Dynamic Link Matching Algorithm"

*Supported by grants from the German Federal Ministry for Science and Technology (413-5839-01 IN 101 B/9) and from AFOSR (F 49620-93-1-0109).

†also at the Department of Computer Science, University of Southern California

1 Introduction

Invariant pattern recognition requires a flexible matching process

Invariant pattern recognition is one of the hardest problems in the theory of perception and signal processing, namely in vision. The problem is that the perceived signal of an object can undergo various changes while the object under observation still remains the same. In the case of vision, the possible changes under which object perception should be invariant include geometric transformations (translation, rotation, scaling, mirror symmetries), changes in lighting conditions, distortion of deformable objects (e. g., faces) and so on. Furthermore, an object may appear in quite different surroundings requiring segmentation as a base for perception.

Invariant pattern recognition can be formulated as a kind of flexible match process between a stored pattern and a perceived pattern where local features in the perceived pattern have to be matched with their counterparts in the stored pattern. The basic problem which makes invariant pattern recognition such a hard task can be sketched as a kind of chicken-egg dilemma: To find unambiguously the counterpart of a local feature, (at least some rough) knowledge about the global transformation is needed. (We mean by global transformation the combined invariance transformations appropriate for the pair of patterns to be matched.) On the other hand, in order to estimate the unknown global transformation, at least for some local features, their correct counterpart is needed.

Pattern recognition with invariance against local distortions or incomplete patterns can be performed with associative memories [1, 2, 3]. Simple associative memories can not incorporate geometric invariances, e. g., translation or rotation, since they do not retain the topographic structure among the features. Extensions to the Hopfield-type associative memory have been suggested [4] which can recognize topographically equivalent representations of small graphs. Recently, a modified Hopfield network has been proposed [5] which can *learn* a geometric invariance from transformation examples and can perform invariant pattern recognition with respect to this transformation. However, the principle is not well extensible to transformation groups with more than one parameter.

Translation (resp. scaling) invariance can be achieved by applying the Fourier transform (resp. Mellin transform) to the pattern and retaining only its magnitude [6, 7, 8]. Systems based on the magnitude of Fourier components, however, have the draw-back of being sensitive to distortions and they are specialized to certain kinds of invariances.

Approaches which contain the least commitment to any specific invariance occuring in pattern recognition are based on topography conservation such that the relative position between local features is retained in some way. Invariant pattern recognition can then be formulated as labeled graph matching [9] and has been demonstrated successfully for retrieval of superimposed patterns [10, 11] or distortion invariant object recognition [12].

Topography as the only constraint is a fairly general principle, i.e., in performing a match under this constraint one has to choose among a large number of possible transformations. It is desirable to have a locally distributed representation of possible global transformations which makes it unnecessary to fixate attention to only *one* transformation hypothesis at a given time and avoids exhaustive hypothesize-and-test strategies. A neuronal architecture with the necessary structure to provide such a representation was formulated by von der Malsburg [13, 14, 15] in the framework of the *Dynamic Link Architecture* (DLA). Within this architecture dynamic links can provide a binding between local

features. The dynamic links of a local feature pose a local constraint on possible global transformations. (A feature may have – at least initially – more than one dynamic link). Neighboring dynamic links with consistent constraints can cooperate – at the expense of other links – for an emergent global transformation. Thus a selection-in-parallel among many possible choices can be performed. A detailed description of the underlying neuronal mechanism can be found elsewhere [16, 17]; it will be reviewed briefly in Section 2 below.

Outline of the paper

The general goal of this paper is to describe a flexible structure for invariant pattern recognition which can deal with a large class of possible invariances in parallel. We will describe in Section 2 how the flexible match process can be formulated in a neuronal style using the concepts of the Dynamic Link Architecture (DLA). However, this style may not be optimal from a computational point of view when simulated on sequential hardware. Therefore, in Section 3 we present a mathematical abstraction of the neuronal model which economizes computational effort and replaces differential equations by a simpler potential approach. This finally leads to a simple mathematical recipe, the *Fast Dynamic Link Matching* (FDLM) algorithm. The functional equivalence between the FDLM and the underlying neuronal model is demonstrated with concrete computer simulations in Section 4. We develop a correlation criterion which allows the network to decide without supervision whether the match was successful. Finally, in Section 4 we present simulation results where FDLM is applied to a visual pattern recognition task on grey level images with realistic distortion and noise conditions.

2 A neuronal formulation of the match process using dynamic links

Problem formulation

Consider the following task: Given two patterns each of which consist of $N \times N$ local features arranged in two 2D-layers X and Y. The problem is to decide whether the two patterns can be matched by some geometric transformation and if so, to determine this transformation. In the present case, geometric transformations include translation, rotation, point symmetry, mirror symmetry along arbitrary axes and local distortions.

Topography constraint as a guideline

If one is looking for a basic method for invariant pattern recognition which can deal with all – or at least with a great variety – of the invariances mentioned above, some guideline is needed to help pick the correct global transformation from among all of the possibilities. As such a guideline we will use here the topography constraint: Whatever transformation is needed to match a local feature f of the stored pattern onto its counterpart in the perceived pattern, it is very likely that almost the same transformation is applicable to match also the neighbors of feature f onto their counterparts. It should be noted that neighborhood does not necessarily imply short Euclidean distance, but can mean

any kind of prewired connectivity in both pattern layers. However, in the present work "neighborhood" will always refer to close Euclidean distance within the layers.

The neuronal system

Finding a match between patterns means finding a set of mutually corresponding cells $a \in X$ and $b \in Y$. [1] A cell a may be considered as a neuron or neuronal group capable of two functions: *(i)* coding for a local feature f_a imposed by the actual pattern; *(ii)* representing an activity state x_a which can be transmitted to other cells. Correspondence is expressed in our model by the binding of cells a and b through a dynamic link $J_{ba} \geq 0$. The links converging on a given cell b are subject to a normalization condition $\sum_a J_{ba} = 1$. Thus they can be interpreted as the probability that a cell a is the correct correspondence for b. Local features assigned to a and b may help as a guideline for the correspondence problem, but will not solve it unambiguously, since a cell $b \in Y$ with feature f_b will have usually more than one candidate match $a \in X$ with the same feature. Let us define a similarity matrix T_{ba} which has high entries for all candidate matches with similar features. In the present case we simply take

$$T_{ba} = \delta(f_b, f_a). \tag{1}$$

An initial – ambiguous – guess for the dynamic link network J_{ba} is $J_{ba} = T_{ba} / \sum_{a'} T_{ba'}$, i. e. each cell b is linked to all cells a with similar features such that the total link strength converging on each cell b is 1. A solution of the match problem must resolve these ambiguities such that each cell b has at most one corresponding a, for instance by assigning to one dynamic link J_{ba} a much higher value than to all other links $J_{b'a}$ emerging from cell a. A desired quality of the neuronal system is its robustness: If some of the cells carry faulty feature information such that both patterns are not exactly but only roughly transformed versions of each other, the solution of the match problem should also be found. A neuronal activity mechanism capable of finding such an solution has been described in [16, 17] and is reviewed briefly below.

A neuronal activity process for the self-organization of dynamic links

The self-organizing process proceeds through a number of iteration steps in the neuronal network. The wiring of the network contains in addition to its dynamic, inter-layer links J_{ba} also static, homogeneous intra-layer connections in both X and Y which are described through an interaction kernel $k(d) = \gamma \exp(-d^2/2s^2) - \beta$ which consists of short-range excitatory connections with range s and global inhibitory connections of relative strength β. It can be shown (cf. Appendix) that with this type of interaction kernel a layer has a connected active region or "blob" as equilibrium solution.

Each iteration step consists of a simultaneous blob formation process in both layers X and Y. This is achieved through a set of coupled differential equations starting from initial conditions $x(0) = y(0) = 0$:

$$\dot{x}_a = -\alpha x_a + (k * X)_a + I_a^{(x)}, \qquad X_a = \sigma(x_a), \tag{2}$$

$$\dot{y}_b = -\alpha y_b + (k * Y)_b + I_b^{(y)}, \qquad Y_b = \sigma(y_b). \tag{3}$$

[1] We use lower-case symbols a, b, a', b' for index vectors $a = (a_1, a_2)$ specifying row a_1 and column a_2 of a cell within a 2D-layer.

where $\sigma(\cdot)$ denotes a sigmoidal output function. Eqs. (2) and (3) differ only in their input terms: The layer X receives its input from an independent noise source $I_a^{(x)2}$ which is slowly varying compared to the dynamics of X and Y. Its value at $t = 0$ breaks the translational symmetry of Eq. (2), letting the blob appear at a different position in each iteration step.

On the other hand, the activity of layer Y is coupled to X through the input term $I_b^{(y)} = \varepsilon_I \sum_a J_{ba} T_{ba} X_a$ with coupling strength ε_I. The activity flow from an active cell $a \in X$ to a cell $b \in Y$ is proportional to the dynamic link J_{ba} connecting them and proportional to the similarity T_{ba} of the local features assigned to a and b. The resulting blob will appear in a region of layer Y which receives the largest activity flow.

When the activity in both layers X and Y has converged to its equilibrium blob solution, the dynamic links between active cells are strengthened according to

$$\Delta J_{ba} = \epsilon J_{ba}\, T_{ba}\, Y_b X_a. \tag{4}$$

Since the desired solution must resolve the local ambiguities, links converging on a given cell $b \in Y$ must compete with each other. This is achieved in our model with a simple sum rule constraint $\sum_a J_{ba} = 1$, which is applied after each update (4) by multiplicative normalization.

An iteration step is completed by resetting the active zones, i. e. $x_a = y_b = 0\ \forall a, b$. The next iteration step starts with a new noise amplitude $I_a^{(x)}$ yielding a blob in X at a different position. The blob formation in Y is influenced by the previous iteration steps through the current state $\{J_{ba}\}$ of the dynamic link network which acts in this sense as a kind of "short term memory".

Where does this process make use of the topography constraint mentioned above? The key feature is that a cell a is not activated alone but always together with some of its neighbors. If those neighbors have strong links to the corresponding region in plane Y, the links of cell a will also grow predominantly in this direction.

3 Functional abstraction of the neuronal process

The formulation of the match process given in Section 2 aimed at demonstrating the neuronal feasibility of the proposed process. The neuronal activity is described by the differential equations (2) and (3). Within each iteration step, these equations have to be solved numerically. While this in principle can be performed quickly on parallel (neuronal) hardware, it becomes costly when large layers are simulated on a sequential machine. Furthermore, the solutions in each iteration step resemble each other strongly, namely there is always one blob in X and one blob in Y. Thus the question arises as to wether there is a simpler approach to getting the desired solution without explicitly solving the differential equations.

The induced potential

It can be shown [18] that the system of differential equations given by Eq. (2) possesses a Lyapunov functional $L[x]$, which is given in the Appendix. A stationary solution of Eq. (2)

[2]The noise for every pair of cells a, a' is uncorrelated, i. e. $\left\langle I_a^{(x)} I_{a'}^{(x)} \right\rangle = \delta_{aa'}$.

is characterized by a local minimum of $L[x]$. Let $x_a = B(a)$ be a stationary solution of Eq. (2) having its maximum at the origin. For spatially homogeneous input $I^{(x)}$, each solution is degenerate with respect to translations[3]: With $B(a)$ being a solution of Eq. (2), $B(a - a_c)$ is also a solution of Eq. (2) for any fixed a_c. Small random inhomogeneities added to the input $I^{(x)}$ (small compared to the intra-layer coupling term $(k * X)$) will lift the degeneracy, but will not alter the *form* of the solution. Thus, instead of solving Eq. (2) explicitly for random input $I^{(x)}$, we can approximate the solution by $x_a = B(a-a_c)$ where $a_c \in X$ is randomly choosen.

Similar remarks now apply to the dynamics of Eq. (3): For homogeneous, stationary input $I^{(y)}$ all stationary solutions $y_b = B(b - b_c)$ with arbitrary b_c are possible. A small inhomogeneous input $I^{(y)}$ will lift the degeneracy in favor of the solution which minimizes the input-dependent term of the Lyapunov functional $L[y]$, which reads (cf. Appendix)

$$V(b_c) = -\sum_b \sigma(B(b - b_c))I_b^{(y)}. \tag{5}$$

The true solution for Eq. (3) can be approximated by $y_b = B(b - b_c)$ where b_c minimizes the potential of Eq. (5). This potential has an intuitive interpretation, since minimizing $V(b_c)$ means maximizing the spatial overlap of the Y solution with the input that layer Y receives. If the blob B is sufficiently large, small local minima in $I^{(y)}$ will be washed out and $V(b_c)$ is a smooth potential with only one dominant minimum.

The approximations above rest on two assumptions about the dynamics of Eqs. (2) and (3):

- Input inhomogeneities are sufficiently weak such that all stationary solutions can be approximated in shape by one generic function B which is sperically symmetric around its center.
- The local minimum found by minimzing Eq. (5) coincides with the local minimum of the Lyapunov functional found by the dynamics. If $V(b_c)$ has only one local minimum, this assumption is obviously fulfilled.

It will be verified by numerical simulations (see next section) that these approximations are indeed valid in the present context.

One can go even one step further in the functional abstraction and replace the equilibrium blob solution B of the differential equation by any other prescribed blob of arbitrary and simple shape. For example one could use a window function

$$B(b) = \begin{cases} 1 & \text{if } \|b\| < l/2 \\ 0 & \text{else.} \end{cases} \tag{6}$$

This can be used for a much faster implementation of the algorithm. Simulations show that the self-organization of dynamic links is not altered qualitatively by such a replacement.

FDLM: The Fast Dynamic Link Matching Algorithm

We summarize the results of the previous paragraph by giving a simple algorithm for the fast self-organization of dynamic links.

[3]Here we do not consider border effects which would be negligible in large layers. In the simulations we use periodic boundary conditions (wrap around).

Before starting the algorithm one has to choose a unimodal blob function $B(b)$ with center at $b = 0$. This may be either an equilibrium solution of a differential equation like Eq. (2) or a window function as defined in Eq. (6). Let $\sigma(\cdot)$ be a sigmoidal function (which may be the unit step function in the simplest case).

(i) Initialize the dynamic links with $J_{ba} = T_{ba} / \sum_{a'} T_{ba'}$.

(ii) Choose a random center $a_c \in X$ and place the blob there: $x_a = B(a - a_c)$. Compute the resulting input to layer Y

$$I_b^{(y)} = \sum_a J_{ba} T_{ba} \sigma(x_a). \tag{7}$$

(iii) Use $I_b^{(y)}$ to compute the minimum $b_c \in Y$ of the potential V in Eq. (5) and place the blob there: $y_b = B(b - b_c)$.

(iv) Update the links between active cells such that the total link strength converging on each cell b is kept constant:

$$J_{ba} = \frac{J_{ba} + \epsilon J_{ba} T_{ba} \sigma(y_b) \sigma(x_a)}{\sum_{a'} (J_{ba'} + \epsilon J_{ba'} T_{ba'} \sigma(y_b) \sigma(x_{a'}))}. \tag{8}$$

(v) Proceed with step (ii).

In its simplest form, where B is given by Eq. (6) and $\sigma(B) = \theta(B) = B$, the algorithm has only two free parameters: the size l of the blob and the update parameter ϵ.

Although the algorithm in its present form has some resemblance to the Kohonen algorithm [19, 22], an important difference should be noted: While in the Kohonen algorithm any cell b has only one weight vector with the dimensionality of the input space, here each b has a distribution of weights J_{ba} to all input cells a with similar features. This initially shallow distribution allows a search-in-parallel among many possible mappings. The resulting flexibility resembles the difference of fuzzy logic concepts as opposed to ordinary logic.

4 Simulations and Results

Neuronal activity process and potential approach yield the same results

The dynamic link algorithm was investigated for patterns of size 8×8 with each cell having one out of $F_{max} = 10$ possible local features. To avoid boundary effects due to the small pattern size, we use wrap-around intra-layer coupling. The fact that the neuronal activity process described in Section 2 is capable of detecting with good reliability various symmetry transformations between two patterns has been demonstrated before [17, 20] where a success rate of 98% was found (3 symmetry classes). We show some typical results for a variety of geometric transformations in Fig. 1. In addition to the examples with global symmetries, Fig. 1(C) shows a pattern with a mixed symmetry: the lower

half of the front pattern maps one-to-one onto the rear pattern, the upper half contains a mirror reflection. Although the network *a priori* does not know anything about existence and location of the boundary, it quite successfully segments the different regions.

In order to test the validity of the approximations made in Section 3, we compare the results from both versions (ODE and induced potential) by applying both methods to the same pair of feature patterns with the same X activity. This activity is fed into two versions of the link network: In the first one, the Y activity is computed from the differential equation (3), in the latter a generic blob is placed in Y according to the FDLM [4]. Both networks perform the same link update rule Eq. (8). Parameters are $\{\alpha, \epsilon_I, \epsilon\} = \{0.30, 1.8, 0.8\}$ and $\{\beta, \gamma, s\} = \{0.73, 1.33, 4.0\}$ for the convolution kernel k which is cut off at a size of 5×5. The random input is parameterized by $I_a^{(x)} = 0.6 + \chi_a$ where χ_a is white noise with amplitude 0.6.

If the assumptions of Section 3 are correct, the self-organization of the dynamic links should proceed in both versions in the same manner. As a measure for the degree of self-organization we use the sum D of squared deviations of the actual link matrix elements J_{ba} from the ideal link matrix elements $J_{ba}^{(ideal)}$, the latter being 1 for corresponding pairs (b, a) and 0 elsewhere:

$$D = \sum_{a,b} (J_{ba} - J_{ba}^{(ideal)})^2. \tag{9}$$

The evolution of the measure D during iterations is shown in Fig. 2 for two typical simulation examples, showing that the agreement between both versions is very good. The remaining small differences are explained by the fact that in the ODE version the shape of the equilibrium blobs not always matches exactly the shape of the generic blob.

Correlation as a criterion for successful match

How can we measure the success of a pattern recognition attempt in the DLA if the ideal link matrix $J^{(ideal)}$ is not known to the system? This is a question of great importance for any pattern recognition system, because the real strength of such a system can be judged best by its ability to reject non-corresponding pattern pairs as "No invariance found" while at the same time detecting the correct geometric transformation in corresponding pairs. So the network should be able to discriminate *without supervision* between successful and unsuccessful matches. Furthermore, if the DLA is a meaningful element as a building block for neuronal computation, a dynamic link network should be able to communicate its "computational results" (i.e. success of match and if so, the matching geometric transformation) to other parts of the system.

Both requirements are not readily obtained from the dynamic links themselves for two reasons:

- In a neuronal system the links or synapses are *local* variables; their strengths are not directly accessible to third parties.
- Even if the links were accessible, it would be difficult to tell from the state $\{J_{ba}\}$ of the dynamic links alone (without knowledge of $J^{(ideal)}$) if a match was found or not, because locally there might be strong links between cells with similar features although no global match exists.

[4] The shape of the generic blob $B(b)$ is computed once from Eq. (3) with the input inhomogeneity $I^{(y)}$ being a unimodal function with its maximum at the origin.

It is interesting to note that the self-organization process itself offers a solution to both problems: Consider the correlation of neuronal activity as it is expressed in the co-activity of specific pairs of blobs in X and Y. Correlated activity of neurons can be easily recorded by other neurons and is thereby accessible to third parties. The correlation C_{ba} between two cells $a \in X$ and $b \in Y$ can be computed as a by-product of the self-organization process of Eqs. (2) – (4). If we denote by X_a and Y_b the output activity of cells a and b, respectively, at the end of each iteration step, the correlation after n steps is given by

$$C_{ba} = \frac{\langle Y_b X_a \rangle - \langle Y_b \rangle \langle X_a \rangle}{\Delta Y_b \, \Delta X_a} \qquad \text{with} \qquad \Delta A = \sqrt{\langle A^2 \rangle - \langle A \rangle^2} \tag{10}$$

where $\langle \cdot \rangle$ denotes averaging over iteration steps. C_{ba} has the value $+1$ (-1) if the cells a and b are always (never) in the same state and approaches 0 if the activity of a and b is uncorrelated.

The number of pairs with correlation above a critical value $C_{min} = 0.9$, weighted by their correlation, gives an excellent discriminant function between successful and non-successful matches:

$$\mathcal{C} = \sum_{\{a,b|C_{ba} \geq 0.9\}} C_{ba}. \tag{11}$$

Only if a globally consistent matching transformation exists, there will be a large number of highly correlated pairs, i. e. pairs which show correlated behaviour in almost all iteration steps.

Ten simulations were performed with input patterns that could be matched onto each other, and another ten with non-matching input patterns. The plots in Fig. 3 show the averages of the ten simulations with the error bars indicating the plus or minus one standard deviation intervals. The initially high values in the curves for $\mathcal{C}$ are not significant but an artefact of poor statistics: Each two cells which have been activated only once and together in the initial steps contribute a value 1 to the measure $\mathcal{C}$. Note that:

- No global information about the geometric transformation is necessary to compute $\mathcal{C}$ (no supervision).
- The same discriminating power can not be achieved by using the link values J_{ba} instead of the correlations C_{ba} and formulating a similar criterion as in Eq. (11), e. g.

  $$\mathcal{J} = \sum_{\{a,b|J_{ba} \geq 0.9\}} J_{ba}. \tag{12}$$

  Even if no global match exists, the dynamics of Eq. (4) obviously leads to a considerable number of links J_{ba} with high values. A closer inspection shows that those links connect pairs of cells having accidentally the same feature but otherwise random spatial arrangement.
- In the case of "No match", even high values in $\mathcal{J}$, caused by "accidental" links, do not lead to significant enhancement of $\mathcal{C}$.

Variation of the Blob Size

To study how the efficiency of the algorithm depends on the blob size, we used a square window function as a blob and varied the square size from 2 to 7 in the 8×8 layer. The

criterion for successful match was the correlation measure $\mathcal{C}$ of Eq. (11). The recognition process was judged to have succeeded if, for ten consecutive steps, $\mathcal{C}$ was in the range $64 \pm 20\%$ and to have failed if this criterion was not fulfilled within 200 iteration steps.

In Fig. 4A the number of misses (recognition failures) out of 100 trials (every trial with a new pair of patterns) is plotted against square size. For each of the 100 patterns all square sizes were tested. Feature sets of two sizes were used: $F_{max} = 10$ (squares) and $F_{max} = 6$ (diamonds), the latter one resulting in 60% more noise (i. e. accidental coincidences) in the similarity matrix. In Fig. 4B the average number of iterations necessary for a successful recognition is plotted against square size. The average is calculated by taking into account only successful matches. Both figures show that optimal performance is obtained if the blobs cover about half of the layer area. For $F_{max} = 6$ some of the random generated patterns have locally clustered ambiguities and in that region the correct transformation is nearly impossible to find. But the algorithm does not detect a wrong transformation, it simply does not come close enough to the ideal value $\mathcal{C} = 64$ in this case.

Robustness

Until now we have demonstrated a solution to the recognition task with pixel patterns which could be translated or reflected but were identical otherwise. A neural algorithm, however, must be insensitive to perturbations as well. We take the input pattern as some translation or reflection of the stored pattern and change every pixel with probability p to a different feature $f_{new} = F_{max} - 1 - f_{old}$. Using a square size of 5, $F_{max} = 10$ possible features and the same criterion for matching success as above we get the results shown in Fig. 5. In Fig. 5A the number of misses out of 100 trials is plotted against p, in Fig. 5B the average number of iterations necessary for successful match is shown. With 20% perturbation the algorithm still succeeds in more than 85% of the pattern examples. The average number of steps needed for successful matches is only weakly rising with increasing perturbation.

Implementation and Complexity

Computer simulations were carried out on a Sun Sparc 2 Station using the Neural Simulation Language (NSL) written by Alfredo Weitzenfeld [21]. The fastest version of the algorithm with binary squares for the blobs needs 10 ms per iteration on a 8×8 layer. Because it typically requires around 30 iterations until a successful match is reached, the recognition process needs around 300 ms. The full simulation of the neuronal activity process with the differential equations Eq. (2) is slower by a factor of 12 to 14.

Two questions arise in connection with the complexity of the algorithm: How does the computation time for a single iteration increase with the size N of the layer and how does the number of required iterations increase with N? (N is the number of neurons per dimension, each layer having N^2 neurons in total.) If S time steps are needed for the blob formation in the neuronal activity process (we used $S = 20$), the computation time per iteration increases in leading order proportional to SN^2A with A being the area of the blob. For the induced-potential version or FDLM complexity is proportional to N^2A operations per iteration. Consequently, FDLM does not decrease complexity but, differing with the simulation of the differential equations, the program is faster by more than a

constant factor of 10. – Concerning the second question about the necessary number of iterations, simulation results indicate that the number of iterations needed to fulfill a given match criterion increases only linearly with the layer size N ($N = 4, 6, \ldots, 20$) [22].

Dynamic link matching between images

Has the dynamic link matching algorithm sufficient flexibility and robustness to deal with invariant pattern recognition in realistic vision problems? To test this ability we applied the DLM algorithm to the task of identifying an object (or part of it) within a different image containing the same object in a different view. This task amounts to identifying the correct region in the image and specifying the transformation which maps to that region on the model of the object.

The general setup is shown in Fig. 6: The picture in Fig. 6(A) is a mirror-reflected view of the object in (C) and contains, in addition, a slight out-of-plane rotation. Note that both front legs are visible in (A) but not in (C). The image in Fig. 6(A) is covered with a rectangular grid whose nodes are analogous to the X cells in Eq. (2). Similarly, the part to be matched, Fig. 6(C), is covered by a smaller grid, its nodes being analogous to the Y cells of Eq. (3). Note that both grids in (A) and (C) are placed arbitrarily, i. e. the nodes of X and Y do in general not fall on corresponding pixels.

To each pair of nodes $a \in X$, $b \in Y$ we assign a feature similarity T_{ba} in the following way: At each node we extract features from the image by applying a set of local filters. In the present context we use five DoG filters (Laplacians) with five different scales. The use of DoG filters is motivated by their resemblance to ON-center-OFF-surround cells of biological vision systems and because we need rotationally invariant filters. Depending on the similarity of the five filter coefficients of each node, we assign to T_{ba} a real value between 1 (strong similarity) and 0 (dissimilarity).

Before we apply the FDLM algorithm of Sec. 3 two further aspects have to be taken into account:

- *Boundary effects:* Since wrap-around boundaries – as we used them in the artificial pattern case – are not appropriate for a vision task, we have to cope with the problem that nodes at the grid border are less frequently hit by blobs and are mapped with a systematic bias towards the middle of the grid. Two solutions may be suggested: Shrinking the blob size slowly towards a minimal blob during the matching process or making both grids larger than the actual region of interest. We choose the latter possibility and extend the Y grid by a 2-node-wide border region (not shown in Fig. 6(C)). The X grid, which covers the whole image, is already sufficiently large.

- *Sparse grids:* In order to have a managable number of dynamic links we have covered the images in Fig. 6(A) and (C) with rather sparse grids. This implies that in general the matching pixel p of a node $b \in Y$ does not fall on a node $a \in X$ but may lie somewhere between several nodes in X. The solution of the FDLM process should account for this by developing links to all those nodes such that their center of gravity

$$C^b = \sum_{a' \in X} a' J_{ba'} \tag{13}$$

  lies at p. To allow final states where the non-zero links of a node b cover an extended region of X ("dynamic receptive field", see Fig. 6F), we modify the link update rule

Eq. (4) to

$$\Delta J_{ba} = \epsilon(J_{ba} + J_0)(T_{ba} + T_0)\, Y_b X_a. \tag{14}$$

with constant growth parameters $J_0, T_0 \geq 0$. In our simulations we set $J_0 = 1.0, T_0 = 0.4$. Both parameters facilitate the growth of links between cells which are frequently co-active: The parameter J_0 facilitates growth irrespective of the "history" of the link (i. e. its previous value). The parameter T_0 facilitates growth between nodes which belong – loosly speaking – into the correct context (are often co-active), although their features are quite dissimilar.

The task of the system is now to map the small pattern defined in Fig. 6(C) onto the full image of Fig. 6(A). We apply the FDLM algorithm of Section 3 with the modified link update rule of Eq. (14) and $\epsilon = 0.01$. We use approximately circular blobs with a bell-shaped profile and a diameter of 4 nodes. The result is displayed as a gravity map in Fig. 6(B): For each node $b \in Y$ we compute the center of gravity C^b within X of the links converging on this node and connect every two C^b's whose nodes b are neighbours in Y. The resulting gravity map in Fig. 6(B) is completely unfolded (contains no crossovers) and is correctly mirror-reflected as can be seen from the corner marked with an '$\times$'. Fig. 6(D) and (E) show as second example the result of a matching process on a 60^o rotated image. In spite of the fact that the network operates only on the regular (non-rotated) grids (D) and (C), the C^b's of the links can arrange themselves in a map which is correctly tilted by approximately 60^o.

Both results have been obtained after 250 iteration steps of the algorithm. This number is somewhat larger than in the case of the artificial 8×8 patterns (cf. Fig. 4) because we operate on larger grids, the local feature information is noisy and because we want to establish here a local map specifying the transformation of each node. A global measure (e. g. "(A) matches (C)") can be obtained already at an earlier stage of the iteration.

In conclusion, these results show that dynamic link matching has the capability to perform visual pattern recognition tasks. The computational results can be visualized in the form of elastic planar graphs as shown in Fig. 6. Previous approaches to invariant object recognition with similar results were based on elastic graph matching algorithms [12, 11] which minimize a global cost function. Our results show how a neural system can perform and extend the functionality of graph matching without explicit reference to a global cost function. The system architecture is based on local interactions and network self-organization, thus being in accordance with paradigms from neural information processing.

Appendix

Lyapunov functional and stationary solutions for the neuronal activity equation (2)

We want to study the layer dynamics driven by the equation

$$\dot{y}_i = -\alpha y_i + \sum_j k_{i-j}\, Y_j + \rho_i \ , \qquad Y_i = \sigma(y_i), \tag{A.1}$$

where y_i is the activity of neuron i. The layer input ρ_i and the lateral coupling k are time independent and $\sigma(y_i)$ is a strictly monotonous – hence invertible – neuronal output function, e. g.

$$\sigma(x) = \frac{1}{1+e^{-\lambda x}}. \tag{A.2}$$

The second term in Eq. (A.1) is the convolution of the layer output with the lateral coupling k, the latter being normally short range excitatory and long range inhibitory. For such a system a Lyapunov functional can be given [23, 24] which reads

$$H[y] = -\frac{1}{2}\sum_{i,j} k_{i-j}\, Y_i\, Y_j + \alpha \sum_i \int_0^{Y_i} \sigma^{-1}(Z)\, dZ - \sum_i \rho_i Y_i. \tag{A.3}$$

When the neuronal activities $y(t)$ evolve in time according to Eq. (A.1) the function $H[y(t)]$ falls monotonously, finally running into a minimum since H is bounded from below. Although the integrand σ^{-1} in Eq. (A.3) contains singularities, it is easy to show that the integral is always well defined.[5] This even holds in the limit $\lambda \to \infty$ where the sigmoid function approaches the step function $\sigma(x) \to \theta(x)$. In this limit the integral vanishes.

In the case of homogeneous input $\rho_i = \rho_0 = const$ all terms in the Lyapunov functional are invariant against a translation of the solution Y. If the input contains small inhomogeneities, among all solutions which differ only by a translation the one solution will be chosen which minimizes the term

$$V = -\sum_i \rho_i Y_i, \tag{A.4}$$

which we call the induced potential. The position of the stationary solution is specified by minimizing the induced potential, i. e. by maximizing the overlap between solution and layer input.

Next we want to show that the differential equation (2) has a one-blob state as equilibrium solution. We will derive an analytical expression for the size l of the blob. The calculation follows partly the analysis of Amari [25] and extends it to the wrap-around topology used in this work and to 2-dimensional layers. Furthermore we show that for global inhibition a solution with more than one blob is unstable.

In order to keep mathematical notation simple, we make the following modifications:

[5] Amari [18] has given a different Lyapunov functional for the same problem which does not require the inverse sigmoid function σ^{-1}.

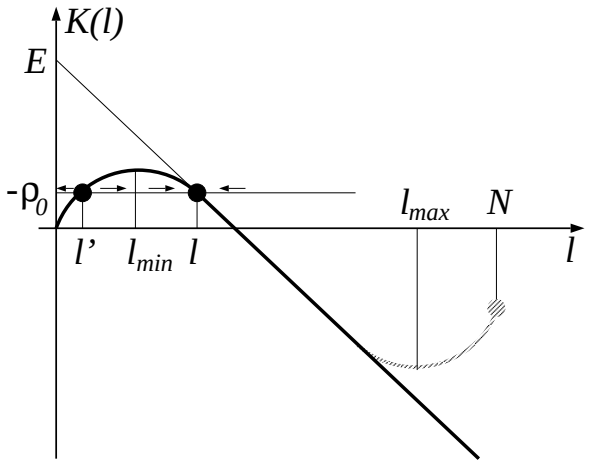

Figure A.1: *Sketch of $K(l)$ (Eq. (A.8)). Black line: infinite range of neural field; dashed line: finite neural field with wrap-around topology. $E = \gamma E(\infty)$ is the integral strength of the local excitation $\gamma k_E(r)$. Black points are fixed points and the arrows denote their stability.*

(a) Instead of neurons with activity x_a at discrete positions $a = (a_1, a_2)$ we consider a continuous neural field $x(\vec{r}, t)$. It is straightforward to rewrite the results obtained below for the discrete case.

(b) We first restrict ourselves to the 1D-case and extend the result to two dimensions at the end.

As a mathematical simplification we specify the sigmoidal function $\sigma(\cdot)$ of Eq. (2) to be a step function: $\sigma(x) = \theta(x)$. Eq. (2) now reads

$$\dot{x}(r,t) = -\alpha x(r,t) + \int dr' k(r - r')\theta(x(r',t)) + \rho(r). \tag{A.5}$$

Here the interaction kernel k consists of a local excitation k_E and a long range (global) inhibition β:

$$k(r) = \gamma k_E(r) - \beta. \tag{A.6}$$

The local excitation may be, for example:

(i) Gaussian, $k_E(r) = e^{-r^2/2s^2}$, as used here;

(ii) constant within a window, $k_E(r) = 1 \quad \text{for} \mid r \mid < d$.

We split the input term $\rho(r)$ into its mean, ρ_0, and the fluctuation $\rho_f(r)$ around that mean

$$\rho(r) = \rho_0 + \rho_f(r) \qquad , \qquad \langle \rho_f \rangle = 0.$$

For the current discussion we will consider only the constant mean. Simulations have shown that small inhomogeneities $\rho_f(r)$ with $\max \|\rho_f(x)\| \ll \rho_0$ do not alter substantially the shape of the equilibrium solution (the inhomogeneities are mainly influencing the position of the activity blob, as shown in Eq. (A.4)).

We define the integrals

$$E(l) = \int_0^l k_E(r)dr \tag{A.7}$$

$$K(l) = \int_0^l k(r)dr = \gamma E(l) - \beta l. \tag{A.8}$$

The function $K(l)$ is depicted in Fig. A.1. For an infinite neural field, $K(l)$ is monotonously decreasing for $l > l_{min}$. For a finite neural field of length N, $K(l)$ is only defined for $l < N$ and bounded by below. Due to wrap-around topology, $K(l)$ is increasing in the vicinity of $l = N$.

Theorem: Let a one-blob state of size l be defined as a function $x(r)$ with $x(r) > 0$ in a connected region of size l. A one-blob state is an equilibrium solution of Eq. (A.5) if its size l satisfies

$$K(l) + \rho_0 = 0 \tag{A.9}$$

and this solution is stable (unstable) if

$$K'(l) < 0 \quad (K'(l) > 0). \tag{A.10}$$

Proof: For our case – where $\sigma(x) = \theta(x)$ is the step function – the second term in the Lyapunov functional Eq. (A.3) vanishes and we get for a one-blob state of length l centered around c

$$H(l) = -\frac{1}{2}\int_{c-\frac{l}{2}}^{c+\frac{l}{2}} dr \int_{c-\frac{l}{2}}^{c+\frac{l}{2}} dr' k(r-r') - \int_{c-\frac{l}{2}}^{c+\frac{l}{2}} dr\,\rho_0 = -\int_0^l dr \int_0^r dr' k(r-r') - \rho_0 l. \tag{A.11}$$

The length condition Eq. (A.9) follows now directly from $\frac{dH}{dl} = 0$ and the stability condition Eq. (A.10) from $\frac{d^2H}{dl^2} = -K'(l) > 0$ ■.

If $0 < -\rho_0 < K(l_{min})$, the condition (A.9) allows two solutions, namely l and l' in Fig. A.1. However, due to Eq. (A.10), only l is a stable solution while l' is an unstable fixed point. More generally it is seen from Fig. A.1 that for any choice of parameters only those blob sizes l can be realized where $l > l_{min}$ (and additionally $l < l_{max}$ for the finite system).

Finally we comment on the case where more blobs than one start to grow in different regions (which may happen due to the input fluctuations ρ_f). We will here give an analysis only for the case of two blobs, but it can be shown that the result is also valid for the case of more than two blobs. Assume that at a given time t two blobs have sizes l_1 and l_2 with $l_1 > l_2$. Because the larger blob receives a larger excitation $E(l_1)$ while both blobs receive the same inhibition strength $\beta(l_1 + l_2)$, the larger blob will grow more rapidly and will be the only one to survive in the equilibrium solution.[6] This intuitive argument can be justified also more formally by considering the Lyapunov function for two blobs with length l_1, l_2 which are further apart than the range of the excitation kernel k_E:

$$H(l_1, l_2) = -\int_0^{l_1} E(r)dr - \int_0^{l_2} E(r)dr + \frac{1}{2}\beta(l_1 + l_2)^2 - \rho_0(l_1 + l_2). \tag{A.12}$$

[6] This is of course an effect of the global inhibition. If the inhibition is long range but finite, the result of Amari [25] holds that two blobs sufficiently far apart have no interaction, i. e. they can simply coexist.

The equilibrium condition $\frac{\partial H}{\partial l_1} = \frac{\partial H}{\partial l_2} = 0$ leads to $E(l_1) = E(l_2)$. The corresponding solution is only stable if it is a local minimum of H. This requires the determinant of the Hessian matrix, $\left|\frac{\partial^2 H}{\partial l_i \partial l_j}\right|$, and its first element $\frac{\partial^2 H}{\partial l_1 \partial l_1}$ to be non-negative. Both conditions together lead to $\beta \leq 0$. Hence the 2-blob solution is unstable since β must be positive to represent an inhibition.

For the finite neural field there is a necessary condition on Eq. (A.9), namely $K(l_{max}) + \rho_0 < 0$. Otherwise the system will end in a positive saturation state $x_0(r) > 0 \quad \forall r$. Using Eq. (A.8), condition (A.9) can be written as

$$l = \frac{1}{\beta}(\gamma E(l) + \rho_0), \tag{A.13}$$

which in general can only be solved numerically. However, it is easy to compute the upper bound

$$l \leq \frac{\gamma E(\infty) + \rho_0}{\beta}. \tag{A.14}$$

The equality sign holds, if $k_E(r)$ has a finite support d_0 (i. e. $k_E(r) = 0 \forall \mid r \mid > d_0$) and $d_0 < l$. This allows a simple connection between the parameters of Eq. (A.5) and the length l of the resulting blob. Most importantly, it can be seen that the length l is inversely proportional to the global inhibition β.

The results of Eqs. (A.9) and (A.13) are readily extended to the 2D-case where l is now the diameter of the blob. We simply have to redefine

$$E(l) = \int\int_{\|\vec{r}\| \leq l/2} k_E(\vec{r} - \vec{l}/2) dr_1 dr_2 \tag{A.15}$$

$$K(l) = \int\int_{\|\vec{r}\| \leq l/2} k(\vec{r} - \vec{l}/2) dr_1 dr_2 = \gamma E(l) - \pi\beta\frac{l^2}{4} \tag{A.16}$$

where $\vec{l} = (l, 0)$. It is assumed that $k_E(\vec{r})$ and also the equilibrium one-blob state are spherically symmetric. The analogue of Eq. (A.14) now becomes an inequality[7]

$$l^2 < \frac{4}{\pi}\frac{\gamma E(\infty) + \rho_0}{\beta}. \tag{A.17}$$

However, the upper bound on the right hand side of Eq. (A.17) is usually close to the actual value for l^2 . For example with the parameter settings $\{\gamma, \rho_0, s\} = \{1.33, 0.6, 4.0\}$ and with k_E cut off at size 5×5 we numerically obtain $E(\infty) = 11.06$. With this we compute from Eq. (A.17) $l < 8.1$ for $\beta = 0.3$ and $l < 6.2$ for $\beta = 0.5$, while the simulations on a 16×16 layer yield $l = 8$ and $l = 6$, respectively.

Nomenclature

N	layer size, 2D-layer has N^2 neurons
n_{steps}	number of iteration steps

[7] This arises from the fact that for $E(\infty)$ the relevant integration region is the intersection of a disk (the support of k_E) with a half-plane, while for any finite l the integration region is the intersection of two disks.

X, Y	layer X, Y
a, b	neuron location index
a_1, a_2	row a_1, column a_2: $a = (a_1, a_2)$
J_{ba}	interlayer connection or coupling strength between cells a and b
T_{ba}	feature similarity between cells a and b
$k(r)$	intralayer connection or coupling strength for neurons with distance r
$k_E(r)$	excitatory part of $k(r)$
$K(r), E(r)$	integrals of $k(r), k_E(r)$
β, γ, s	parameters specifying the form of $k(r)$
α	neuron activity decay rate
x_a, y_b	activities of neuron a in layer X, neuron b in layer Y
X_a, Y_b	output of neuron a in layer X, neuron b in layer Y
$I_a^{(x)}$	external input to neuron a in layer X
ρ_i	external input to neuron i
ϵ_I	parameter for coupling strength between layers X and Y
ϵ	growth parameter for the interlayer links
$B(a)$	stationary activity function, blob
$L[x], H[x]$	Lyapunov function or Hamiltonian for layer X with activities x
$V(b_c)$	potential for blob centered around b_c
l	length or diameter of a blob
C_{ba}	correlation between cells a and b
$\mathcal{C}$	global correlation measure
$\mathcal{J}$	another match criterion
C^b	center of gravity for links J_{ba}
J_0, T_0	additional growth parameters for links

References

[1] D.J. Willshaw, O.P. Buneman, and H.C. Longuet-Higgins. Non-holographic associative memory. *Nature*, 222:960, 1969.

[2] T. Kohonen, E. Reuhkala, K. Mäkisara, and L. Vainio. Associative recall of images. *Biological Cybernetics*, 22:159–168, 1976.

[3] J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79:2554–2558, 1982.

[4] R. Kree and A. Zippelius. Recognition of topological features of graphs and images in neural networks. *Journal of Physics A*, 21:L813, 1988.

[5] A.C.C. Coolen and F.W.Kuijk. A learning mechanism for invariant pattern recognition in neural networks. *Neural Networks*, 2:495, 1989.

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

[7] 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*, 60:17–22, 1988.

[8] A. Fuchs and H. Haken. 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*, 60:107–109, 1988.

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

[10] C. von der Malsburg and E. Bienenstock. A neural network for retrieval of superimposed connection patterns. *Europhysics Letters*, 3:1243–1249, 1987.

[11] C. von der Malsburg. Pattern recognition by labeled graph matching. *Neural Networks*, 1:141–148, 1988.

[12] M. Lades, J.C.Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R.P. Würtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transaction on Computers*, 42:300–311, 1993.

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

[14] C. von der Malsburg. Nervous structures with dynamical links. *Berichte der Bunsengesellschaft für Physikalische Chemie*, 89:703–710, 1985.

[15] C. von der Malsburg. How are Nervous Structures Organized? In E. Başar, H. Flohr, H.Haken, and A.J. Mandell, editors, *Synergetics of the Brain, Proceedings of the International Symposium on Synergetics*, pages 238–249. Springer, Berlin, Heidelberg, New York, May 1983.

[16] D.J. Willshaw and C. von der Malsburg. How patterned neural connections can be set up by self-organization. *Proc. R. Soc. London B*, 194:431–445, 1976.

[17] W. Konen and C. von der Malsburg. Unsupervised symmetry detection: A network which learns from single examples. In I. Aleksander, editor, *Proceedings of the International Conference on Artificial Neural Networks*, pages 121–125. North-Holland, Amsterdam, 1992.

[18] S. Amari. Dynamical stability of formation of cortical maps. In M.A. Arbib and S. Amari, editors, *Dynamic Interactions in Neural Networks: Models and Data*. Springer, 1989.

[19] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

[20] W. Konen and C. von der Malsburg. Learning to generalize from single examples in the dynamic link architecture. *Neural Computation*, 5(5), 1993.

[21] A. Weitzenfeld. NSL Neural Simulation Language, Version 2.0. Technical report, Center for Neural Engineering, University of Southern California, 1990.

[22] K. Behrmann. Leistungsuntersuchung des 'Dynamischen Link-matchings' und Vergleich mit dem Kohonen-Algorithmus. Technical report, Institut für Neuroinformatik IR-INI 93-05, Ruhr-Universität Bochum, D-44780 Bochum, FRG, 1993.

[23] M. Cohen and S. Grossberg. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Trans. SMC*, 13:815–826, 1983.

[24] J.J. Hopfield. Neurons with graded responses have collective computational properties like those of two-state neurons. *Proc. Nat. Acad. of Sciences*, 81:3088–3092, 1984.

[25] S. Amari. Topographic organization of nerve fields. *Bulletin of Mathematical Biology*, 42:339–364, 1980.

Figures

Figure 1: **Various invariance transformations detected by the dynamic link network.** The figures **(A)**–**(C)** show the layers X (in front) and Y (in the rear) after 40 iteration steps. Different pattern features are represented by the grey level of the cells. White circles denote cells lying within the currently active blob. Both layers are fully connected by links with dynamically variable strength. In the figures only those links are shown which have more than 40% of their maximal strength ($J_{ba} \geq 0.4$). Input patterns have **(A)** vertical mirror symmetry, **(B)** point symmetry, **(C)** mixed symmetry (lower half: identity mapping, upper half: vertically mirror reflected). Only a relatively small number (40) of iterations leads to a well-organized state of the dynamic link network (with minor imperfections like a few missing links in **(A)** and two misplaced links in **(B)**). Example **(C)** shows that the network is able to segment regions of different invariances.

Figure 2: Deviation $D = \sum_{a,b}(J_{ba} - J_{ba}^{(ideal)})^2$ of the link network J_{ba} from its ideal configuration $J_{ba}^{(ideal)}$ as a function of the number n of link updates. Starting from a pair X, Y of identical patterns, the two lower curves show the result of the neuronal activity process (solid line) and of the FDLM (dashed line). The two upper curves show the same for another pattern pair. Obviously the precise time development of D is dependent on the (randomly chosen) initial conditions, but for any specific pattern both versions give nearly the same results.

Figure 3: **Unsupervised discrimination between corresponding and non-corresponding patterns.** As a function of the number n of iteration steps we show the correlation measure $\mathcal{C}$ (solid lines) as defined in Eq. (11) and the link measure $\mathcal{J}$ (dashed lines) as defined in Eq. (12). Thick lines (upper two curves): Results in the case of corresponding patterns (successful match); thin lines (lower two curves): non-corresponding patterns, i. e. no global match exists between the patterns. While $\mathcal{C}$ quickly drops to values near zero in the non-match case, the corresponding curve for $\mathcal{J}$ rises monotonically and does not allow good discrimination.

Figure 4: Performance with different sizes of square-shaped blobs. The 100 trial patterns had 6 (diamonds) resp. 10 (squares) different pixel features.

Figure 5: Effects of disturbing the local feature information by flipping each pixel with probability p to a false value (100 trial patterns).

Figure 6: **Invariant visual object recognition.** **(A)** Image of the elephant covered by a 12×12 X-grid. **(C)** Mirror-reflected and slightly depth rotated view of the elefant, covered by the Y-grid. **(B)** Within the DLM process each Y-node establishes links to its corresponding region in the X-grid. The picture shows the centers of gravity for the links of every Y-node, arranged in a map with the same topology as the Y-grid. One corresponding corner of the map and the grid in (C) is marked with an '$\times$', showing that the map is correctly mirror-reflected. **(D)** Same as (A), but with an additional 60^0 rotation. **(E)** Gravity map obtained by matching (C) on the image (D). Note that the gravity map is correctly tilted although the nodes in both (C) and (D) are arranged in regular, non-tilted grids. **(F)** Link strength distribution for a typical node of Y.

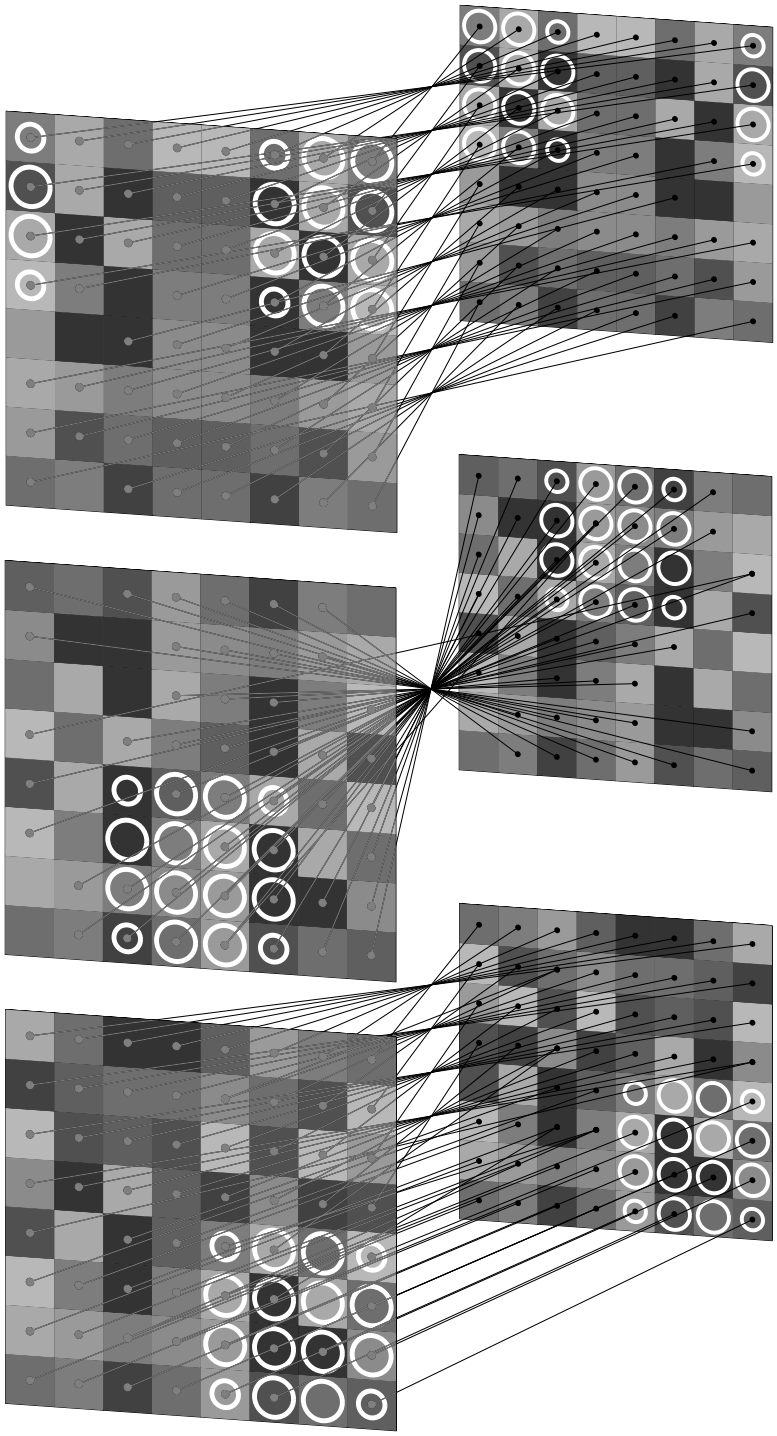

Fig. 1

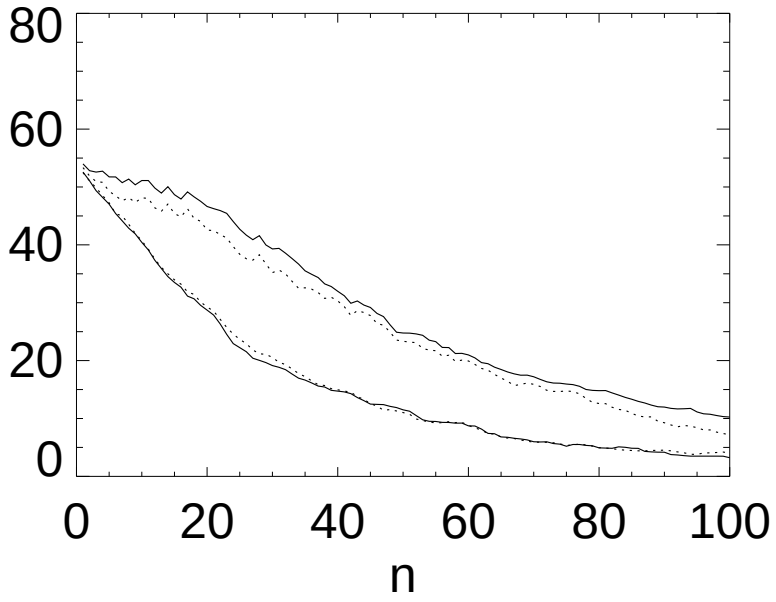

Fig. 2

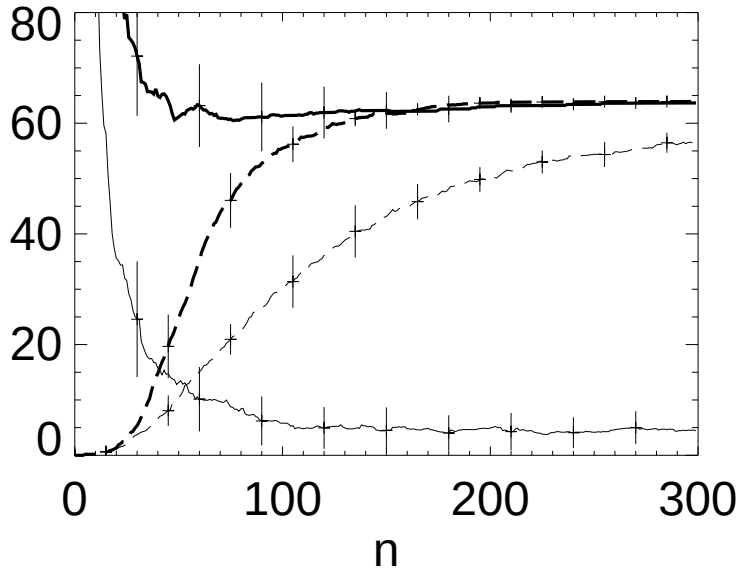

Fig. 3

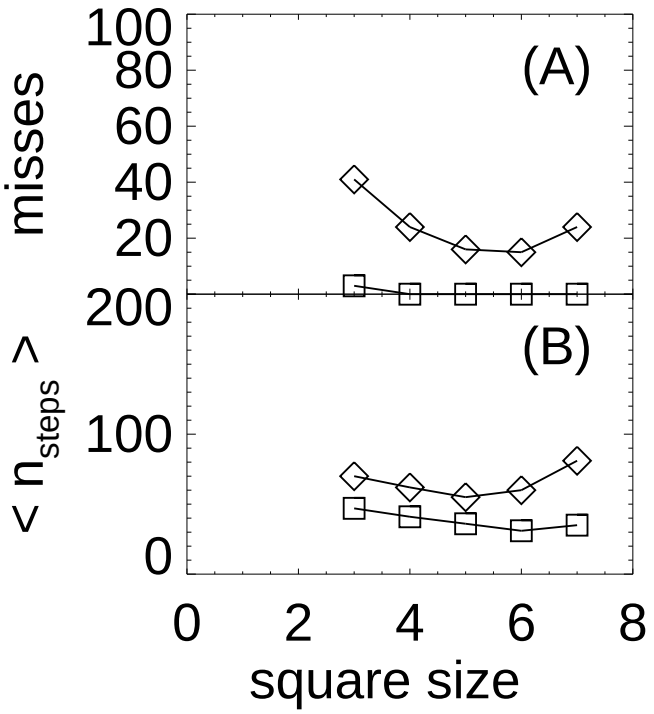
100
80
60
40
20
misses
(A)
200
(B)
100
0
< n steps >
0
2
4
6
8
square size

Fig. 4

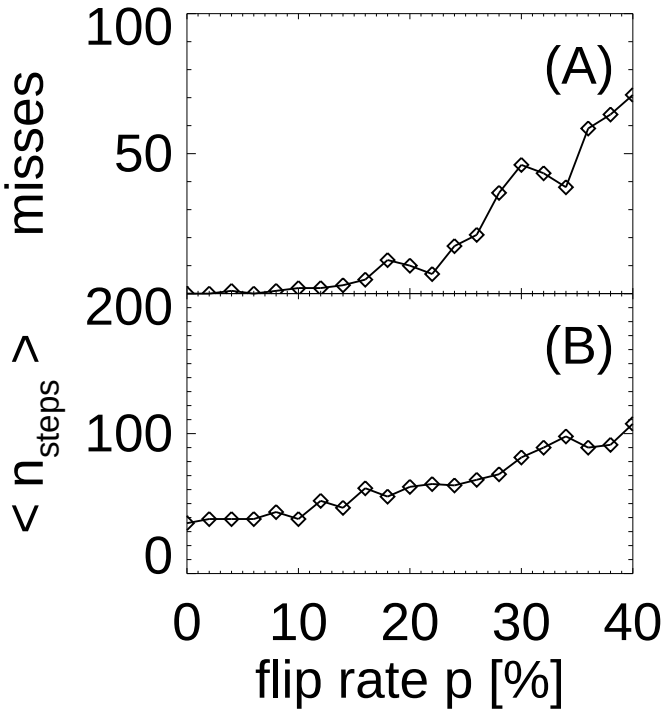
100
50
misses
(A)
200
(B)
100
0
< n steps >
0
10
20
30
40
flip rate p [%]

Fig. 5

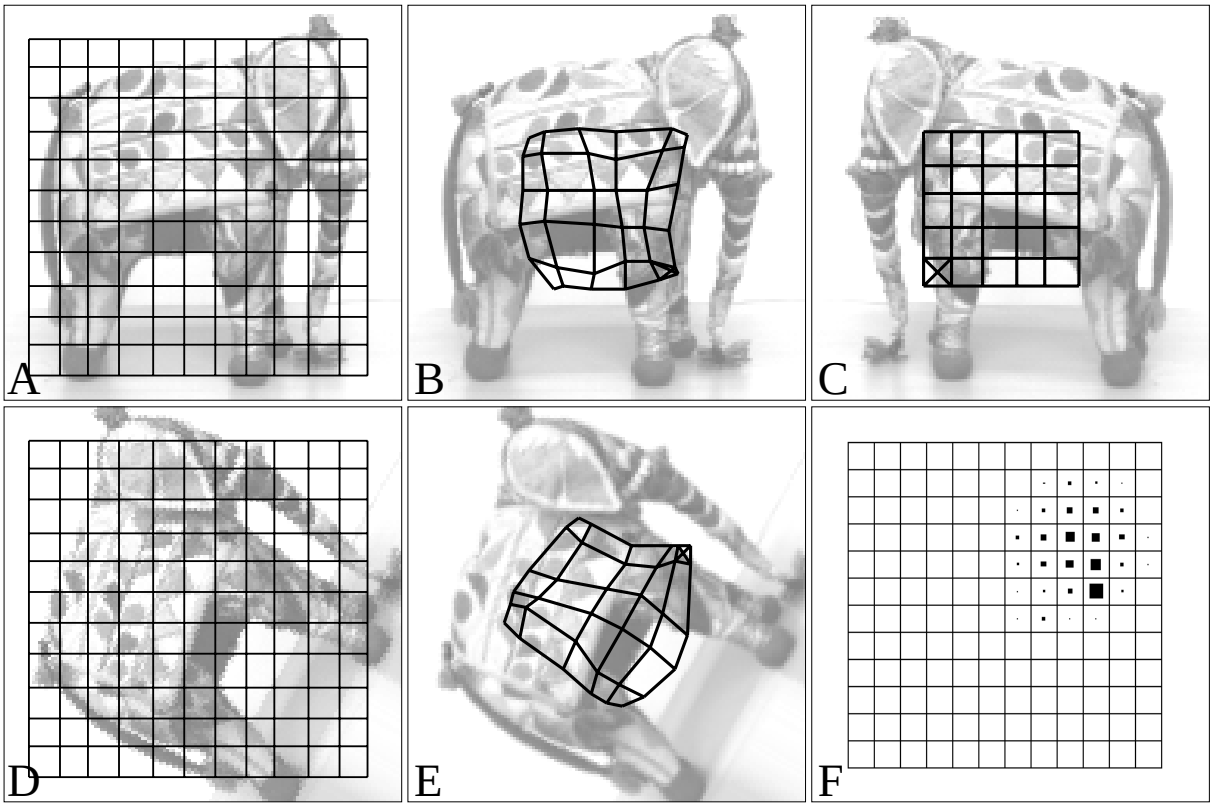

Fig. 6