

Learning Lateral Interactions for Feature Binding and Sensory Segmentation from Prototypic Basis Interactions

Sebastian Weng, Heiko Wersing, Jochen J. Steil, Helge Ritter

Abstract— We present a hybrid learning method bridging the fields of recurrent neural networks, unsupervised Hebbian learning, vector quantization, and supervised learning to implement a sophisticated image and feature segmentation architecture. This architecture is based on the competitive layer model (CLM) dynamic feature binding model which is applicable on a wide range of perceptual grouping and segmentation problems. A predefined target segmentation can be achieved as attractor states of this linear threshold recurrent network, if the lateral weights are chosen by Hebbian learning. The weight matrix is given by the correlation matrix of special pattern vectors with a structure dependent on the target labeling. Generalization is achieved by applying vector quantization on pairwise feature relations, like proximity and similarity, defined by external knowledge. We show the successful application of the method to a number of artificial test examples and a medical image segmentation problem of fluorescence microscope cell images.

Keywords— Recurrent Neural Network, Feature Binding, Perceptual Grouping, Supervised and Unsupervised Learning.

I. INTRODUCTION

The concept of feature binding has been proposed to provide elegant solution strategies to the segmentation problem in perception [21], [24], [25], [30]. Many feature binding models have thus tried to reproduce grouping mechanisms, like the Gestalt laws of visual perception, e.g. connectedness and good continuation, using temporal synchronization [1], [3], [13], [24], [25] or spatial coactivation [2], [4], [20], [30] for binding. In these models, grouping is based on lateral interactions between feature-representing neurons which characterize the degree of compatibility between features. Currently, in most of the approaches this lateral interaction scheme is chosen heuristically, since the experimental data on the corresponding connection patterns in the visual cortex is insufficient. Nevertheless, in more complex feature spaces this heuristic approach becomes infeasible, raising the question for more systematic learning methods for lateral interactions.

Mozer et al. [13] suggested supervised learning for a dynamic feature binding model of complex-valued directional units, where the connections to hidden units, guiding the grouping dynamics, were adapted by recurrent backpropagation learning. The application was limited to synthetic rectangle patterns. Sirosh et al. [22] combined self-organizing maps in feature space with Hebbian learning of lateral connections. Hofmann et al. [7] considered unsupervised texture segmentation by a pairwise clustering approach on feature vectors derived from Gabor filter banks at different frequencies and orientations. In their model the pairwise feature compatibilities were determined by a

divergence measure of the local feature distributions, which was shown to achieve good segmentation results for a range of image types. The problem of segmentation can also be phrased as a labeling problem, where relaxation labeling algorithms [21] have been used as a popular tool in a wide range of computer vision applications [10]. Pelillo & Refice [19] suggested a supervised learning method for the compatibility coefficients of relaxation labeling algorithms, based on minimizing the distance between a target labeling vector and the output after iterating a fixed number of relaxation steps. The main problem are multiple local minima arising in this highly nonlinear optimization problem.

As was shown recently, linear threshold (LT) networks provide interesting architectures for combining properties of digital selection and analog context-sensitive amplification [5], [29], [32] with efficient hardware implementation options [5]. Xie et al. [31] demonstrated how these properties can be used to learn winner-take-all competition between groups of neurons in an LT network with lateral inhibition. We use the competitive layer model (CLM), a large-scale, topographically organized recurrent network of linear threshold neurons which has proved its capability to solve a large variety of feature binding problems [16], [28], [30], mostly in computer vision. Like other feature binding methods, the CLM dynamics is based on evaluating a pairwise interaction function, which indicates the mutual degree of compatibility between two features with respect to the desired grouping. Suitable interactions for the CLM were derived from Gestalt principles [16], [20], [30] or had to be “hand-crafted” using problem-specific knowledge. The key to learning the compatibility function from a given labeled example is the fact that stable binding states of the CLM dynamics are characterized by a large number of consistency inequalities [30]. In [28], a quadratic consistency optimization (QCO) approach has been introduced to find feasible interaction weights considering all consistency inequalities. In this contribution, we show that the inequalities do not have to be explicitly computed because an interaction which is optimal with respect to a given example labeling can be derived from a form of one-shot Hebbian learning (HL). Generalization then can be achieved by projecting this optimal configuration to a number of basis functions which cover the whole space of potential feature combinations. We show the efficiency of this approach on a number of benchmark problems and a challenging medical image segmentation problem.

In section II we present the architecture and dynamics of the CLM. In section III we formulate the learning problem in terms of consistency inequalities with respect to a given target labeling and elaborate on the Hebbian learning result. This section further introduces the basis function approach and describes, how

S. Weng, J. J. Steil and H. Ritter are with the Neuroinformatics Department, Bielefeld University, Germany. E-mail: {sweng, jsteil, helge}@techfak.uni-bielefeld.de. H. Wersing is with the Honda Research Institute Europe GmbH, Offenbach, Germany. E-mail: heiko.wersing@honda-ri.de.

basis functions can be generated in a data driven way. Section IV shows the application of the learning architecture to controlled artificial binding problems of increasing complexity and to the real world problem of fluorescence cell segmentation. All results are compared to the earlier introduced quadratic optimization approach. Section V discusses the relation of the CLM to other feature binding architectures. Finally, we give a conclusion in section VI.

II. THE CLM ARCHITECTURE

A. The General Architecture

The CLM [20] consists of a set of L layers of feature-selective linear threshold neurons (see Fig. 1). The activity of a neuron at position r in layer α is denoted by $x_{r\alpha}$, and a *column* r denotes the set of the neuron activities $x_{r\alpha}$, $\alpha = 1, \dots, L$, sharing a common position r . Associated with each column r is a particular feature vector \mathbf{m}_r which contains a set of local properties, e.g. local edge elements characterized by position and orientation (x_r, y_r, φ_r) . A binding between two features \mathbf{m}_r and $\mathbf{m}_{r'}$ is expressed by simultaneous activities $x_{r\alpha(r)} > 0$ and $x_{r'\alpha(r')} > 0$ that share a common layer $\alpha(r) = \alpha(r')$. All neurons in a column r are equally driven by an external input h_r which represents the significance of the detection of feature r by a preprocessing step, for simplicity, we assume in this work all h_r to be equal to one. The afferent input h_r is fed to the activities $x_{r\alpha}$ with a connection weight $J > 0$. Within each layer α the activities are coupled via lateral connections $f_{rr'}$ which characterize the degree of compatibility between features \mathbf{m}_r and $\mathbf{m}_{r'}$. $f_{rr'} = f(\mathbf{m}_r, \mathbf{m}_{r'})$ is a symmetric function of the feature parameters, thus $f_{rr'} = f_{r'r}$. Positive values of $f_{rr'}$ express compatibility through excitatory connections, while negative values of $f_{rr'}$ express incompatibility through inhibitory connections.

The purpose of the layered arrangement in the CLM is to enforce a dynamical assignment of the input features to the layers using the contextual information stored in the lateral interactions. The unique assignment to a single layer is realized by a columnar Winner-Take-All (WTA) circuit which uses mutual symmetric inhibitory interactions with absolute strength $J > 0$ between neural activities $x_{r\alpha}$ and $x_{r\beta}$ that share a common column r . Due to the WTA coupling, for a stable equilibrium state of the CLM only one neuron from a single layer can be active within each column [30]. Thereby the whole attractor state describes a separation of the input features into several groups, where each group is active in a different layer. Remarkably the number of layers does not predetermine the number of active groups, since for sufficiently many layers only those are active that carry a salient group.

The combination of afferent inputs and lateral and vertical interactions is combined into the standard linear threshold additive activity dynamics

$$\dot{x}_{r\alpha} = -x_{r\alpha} + \sigma\left(J(h_r - \sum_{\beta} x_{r\beta}) + \sum_{r'} f_{rr'} x_{r'\alpha}\right), \quad (1)$$

where $\sigma(x) = \max(0, x)$. This corresponds to gradient de-

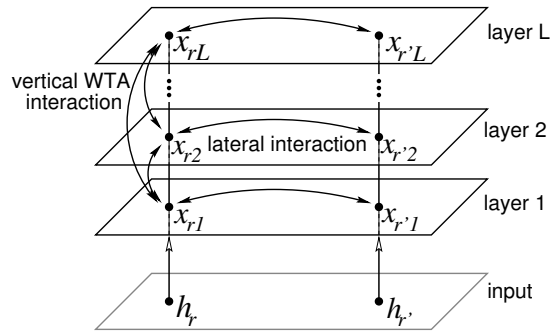


Fig. 1. The competitive layer model architecture (see text for description).

scent in the energy

$$E = - \sum_{r\alpha} J h_r x_{r\alpha} + \frac{1}{2} \sum_{r\alpha\beta} J x_{r\alpha} x_{r\beta} - \frac{1}{2} \sum_{\alpha r r'} f_{rr'} x_{r\alpha} x_{r'\alpha}. \quad (2)$$

The minima of (2) correspond to stable states of the CLM. It is proved [30] that for J large compared to the lateral weights $f_{rr'}$ a single active neuron in each column reproduces its afferent input, $x_{r\alpha(r)} \approx h_r$, while the other neurons of the column are inactive $x_{r\beta} = 0, \beta \neq \alpha(r)$. The assignments of the features r to the layers $\alpha(r)$ are characterized by consistency conditions for a minimum in E :

$$\sum_{r'} f_{rr'} x_{r'\beta} < \sum_{r'} f_{rr'} x_{r'\hat{\alpha}(r)} \quad \text{for all } r, \beta \neq \hat{\alpha}(r), \quad (3)$$

expressing that each feature r is assigned to the layer $\alpha(r) = \operatorname{argmax}_{\alpha} \sum_{r'} f_{rr'} x_{r'\alpha}$, where it receives the maximal support from the other features in this layer.

The CLM-dynamics can be implemented by a Gauss-Seidel approach that iteratively tries to find a fixed point solution of (1) for a randomly selected neuron. This implementation can be extended by an annealing technique introducing a pseudo temperature T to prevent suboptimal attractor states. We present the simulation algorithm in appendix I, details can be found in [30].

B. Figure-Background Separation

A special goal in many grouping and segmentation tasks is the separation of the relevant objects in the data from noisy or incoherent parts that form some kind of background for the desired information. This requirement can be easily integrated into the CLM architecture by using an additional layer b , where the lateral interactions are restricted to a self-interaction with strength m such that $f_{rr'}^b = m\delta_{rr'}$ ($\delta_{rr'}$ is the Kronecker delta). Using this expression as right respectively left hand side of the consistency conditions (3) yields that m defines a threshold of minimal mutual support that is necessary to assign a feature m_r to one of the relevant groups ($\alpha \neq b$). All features whose mutual support lies below this threshold are assigned to the background.

III. LEARNING OF LATERAL INTERACTIONS

A. Formulation of the Learning Problem

The learning problem consists of finding a suitable compatibility function $f_{rr'} = f(\mathbf{m}_r, \mathbf{m}_{r'})$ which expresses the preference to bind similar features $\mathbf{m}_r, \mathbf{m}_{r'}$ from a problem-specific

feature domain $\mathcal{F} = \{\mathbf{m}_1, \dots, \mathbf{m}_{N_{\mathcal{F}}}\}$ by positive values or the preference to segregate dissimilar features by negative values, respectively. We assume that a set of M labeled training patterns $\mathcal{P}^i, i = 1, \dots, M$ is given. For each \mathcal{P}^i a subset $\mathcal{R}^i = \{\mathbf{m}_1^i, \dots, \mathbf{m}_{N^i}^i\}$ of N^i different features and their corresponding target labels $\alpha^i(r) = \alpha(\mathbf{m}_r^i) \in \{1, \dots, L\}$ is obtained, where L is the maximal number of groups in a pattern and $r \in \{1, \dots, N^i\}$.

We first consider the case of a single training pattern $\mathcal{P}^i = \mathcal{P}$ to skip the index i and consider generalization to a larger training set later. Unless otherwise stated, we use the convention that $r \in \{1, \dots, N^i\}$ and that $\alpha(r)$ denotes the target label for feature \mathbf{m}_r while β and β' , denote other possible labels from $\{1, \dots, L\}$, ($\beta \neq \alpha(r) \neq \beta'$).

To obtain a target attractor \mathbf{y} for the CLM dynamics from the desired labels $\alpha(r), r = 1, \dots, N$, we set the activations of neurons that describe a correct assignment $\alpha(r)$ to one and all others to zero:

$$y_{r\alpha(r)} = 1; \quad y_{r\beta} = 0; \quad \text{for all } r, \beta \neq \alpha(r). \quad (4)$$

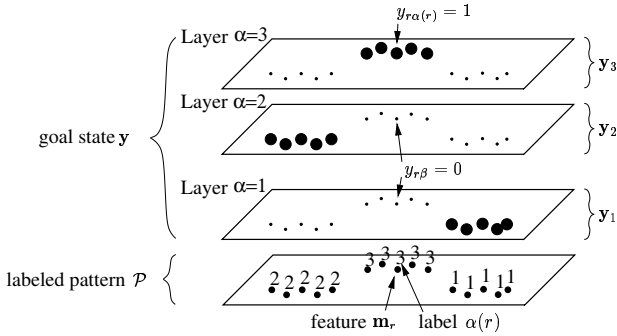


Fig. 2. Target state of the CLM: Activations of neurons that describe a correct assignment $y_{r\alpha(r)}$ are set to one, while all others are set to zero.

The learning goal is to choose $f_{rr'}$ such that these target states are stable states of the CLM grouping dynamics in the sense of minima of (2) which implies that they must be consistent with respect to (3). Therefore we substitute the target states (4) into (3) to obtain $N(L-1)$ target consistency conditions

$$\sum_{r'=1}^N f_{rr'} y_{r'\beta} < \sum_{r'=1}^N f_{rr'} y_{r'\alpha(r)} \quad \text{for all } r, \beta \neq \alpha(r). \quad (5)$$

In [28] and [26], the problem to violate the conditions (5) as little as possible is treated as a quadratic consistency optimization (QCO) problem in the variables $f_{rr'}$, for details see Appendix II. Already this approach yields remarkable grouping quality and will be compared with our new methods below. The drawback of the quadratic optimization is that it requires to compute and store all $N(L-1)$ inequalities in (5) which for most applications is computationally demanding or impossible. Below we show that this effort can be avoided, based on theoretical considerations on Hebbian learning.

B. Hebbian Learning

To fulfill (5) we choose a matrix $\hat{F} = (f_{rr'})$ of lateral interactions as the correlation matrix of all differential vectors

$(\mathbf{y}_\gamma - \mathbf{y}_\mu)$, where the vectors \mathbf{y}_γ and \mathbf{y}_μ represent all components in the γ th and μ th layer of the goal state \mathbf{y} (see fig. 2)

$$\hat{F} = \sum_{\gamma} \sum_{\mu \neq \gamma} (\mathbf{y}_\gamma - \mathbf{y}_\mu)(\mathbf{y}_\gamma - \mathbf{y}_\mu)^T. \quad (6)$$

Because of the special structure of the vectors $(\mathbf{y}_\gamma - \mathbf{y}_\mu)$ we can evaluate \hat{F} without explicit computation. Without restriction consider the features and target labels ordered, such that $r \leq r' \Leftrightarrow \alpha(r) \leq \alpha(r')$. By this permutation we get new goal state vectors \mathbf{y}_γ^{ord} and \mathbf{y}_μ^{ord} . Since the vectors $(\mathbf{y}_\gamma^{ord} - \mathbf{y}_\mu^{ord})$ have only entries 1, -1, and 0, and due to the ordering of the labels, we obtain the block-diagonal matrix

$$\frac{\hat{F}}{2} = \begin{pmatrix} [L-1]_{N_1 \times N_1} & [-1]_{N_1 \times N_2} & \cdots & [-1]_{N_1 \times N_L} \\ [-1]_{N_2 \times N_1} & [L-1]_{N_2 \times N_2} & \ddots & [-1]_{N_2 \times N_L} \\ \vdots & \ddots & \ddots & \vdots \\ [-1]_{N_L \times N_1} & [-1]_{N_L \times N_2} & \cdots & [L-1]_{N_L \times N_L} \end{pmatrix} \quad (7)$$

where $[\cdot]_{N_i \times N_j}$ is a constant matrix of size $N_i \times N_j$. We show a more descriptive derivation of (7) for the concrete example of 3 groups with 3, 2, and 1 feature in the appendix III.

We find that the positive entries are scaled against negative ones with a scalar factor $\lambda = \frac{1}{L-1}$ which only depends on the number of groups in the pattern. This reflects that there exist for each r a total of $L-1$ consistency inequalities (5). The block-diagonal structure of (7) fulfills (5), because all positive values $y_{r'\beta}$ on the left hand side are multiplied with negative weights $f_{rr'} < 0$, while all positive values $y_{r'\alpha}$ on the right hand side are multiplied with positive weights $f_{rr'} > 0$. This property still holds for all simultaneous permutations of the columns and rows of (7) that describe an arbitrary ordering of the features in a concrete pattern.

Obviously, for any positive value of λ the interaction matrix (7) makes the target labeling consistent for the CLM according to (5), because it preserves the block-diagonal structure of positive and negative values in the interaction matrix \hat{F} . For the moment we keep $\lambda = \frac{1}{L-1}$ fixed, while in the later chapters we will show how the modification of λ controls the segmentation level of the output labeling of the CLM.

To apply well-known arguments from Hebbian learning to store attractor states in a Hopfield network [8], we make the consistency inequalities (5) more restrictive by introducing further assumptions about the stable states of the CLM. We require the support of all features m_r belonging to a group $\alpha(r)$ to be of equal strength in all other layers β and β' which can be expressed by the equations:

$$\sum_{r'=1}^N f_{rr'} y_{r'\beta} = \sum_{r'=1}^N f_{rr'} y_{r'\beta'}. \quad (8)$$

On the other hand assume that the inequalities in (5) for the target label hold with a strict margin of one:

$$1 + \sum_{r'=1}^N f_{rr'} y_{r'\beta} = \sum_{r'=1}^N f_{rr'} y_{r'\alpha(r)}. \quad (9)$$

We collect all lateral weights $f_{rr'}$ corresponding to the feature \mathbf{m}_r as a row vector $\mathbf{f}_r = (f_{r1}, \dots, f_{rN})$ and all components $y_{\mu r}$ in the layer μ , of the target state \mathbf{y} as $\mathbf{y}_\mu = (y_{1\mu}, \dots, y_{N\mu})^T$ and rewrite (8) and (9) as scalar products

$$\mathbf{f}_r(\mathbf{y}_\beta - \mathbf{y}_{\beta'}) = 0 \Leftrightarrow \mathbf{f}_r(\mathbf{y}_{\beta'} - \mathbf{y}_\beta) = 0, \quad (10)$$

$$\mathbf{f}_r(\mathbf{y}_{\alpha(r)} - \mathbf{y}_\beta) = 1 \Leftrightarrow \mathbf{f}_r(\mathbf{y}_\beta - \mathbf{y}_{\alpha(r)}) = -1. \quad (11)$$

Now choose two arbitrary labels $\gamma, \mu \in \{1, \dots, L\}$ and collect all equations from (10) and (11) which contain the vector $(\mathbf{y}_\gamma - \mathbf{y}_\mu)$, then for all r :

$$\begin{aligned} \mathbf{f}_r(\mathbf{y}_\gamma - \mathbf{y}_\mu) &= 1, & \alpha(r) &= \gamma & (\Leftrightarrow (\mathbf{y}_\gamma - \mathbf{y}_\mu)_r &= 1), \\ \mathbf{f}_r(\mathbf{y}_\gamma - \mathbf{y}_\mu) &= -1, & \alpha(r) &= \mu & (\Leftrightarrow (\mathbf{y}_\gamma - \mathbf{y}_\mu)_r &= -1), \\ \mathbf{f}_r(\mathbf{y}_\gamma - \mathbf{y}_\mu) &= 0, & \gamma &\neq \alpha(r) \neq \mu & (\Leftrightarrow (\mathbf{y}_\gamma - \mathbf{y}_\mu)_r &= 0). \end{aligned}$$

Stacking the vectors \mathbf{f}_r to obtain the desired interaction matrix $F = (\mathbf{f}_1^T, \dots, \mathbf{f}_N^T)^T$ yields the matrix-vector product

$$(F(\mathbf{y}_\gamma - \mathbf{y}_\mu))_r = \begin{cases} 1 & : (\mathbf{y}_\gamma - \mathbf{y}_\mu)_r = 1 \\ -1 & : (\mathbf{y}_\gamma - \mathbf{y}_\mu)_r = -1 \\ 0 & : (\mathbf{y}_\gamma - \mathbf{y}_\mu)_r = 0 \end{cases} \quad (12)$$

If we now assume that we are not seeking the lateral connections for the linear threshold neurons of the CLM, but interpret the matrix F as a weight matrix of a Hopfield network with binary neurons, we see that the learning problem is similar to the problem of storing the $L(L-1)$ pattern vectors $\mathbf{y}_\gamma - \mathbf{y}_\mu$ in the weight matrix of a Hopfield network. According to standard Hebbian learning we can choose \hat{F} as the correlation matrix of all vectors $(\mathbf{y}_\gamma - \mathbf{y}_\mu)$ as in (6).

The argument of Hebbian learning holds exactly, if all pattern vectors $(\mathbf{y}_\gamma - \mathbf{y}_\mu)$ are orthogonal. In our case, every two vectors $(\mathbf{y}_\alpha - \mathbf{y}_\beta)$ and $(\mathbf{y}_\alpha - \mathbf{y}_{\beta'})$ have an overlap proportional to the size of the group α . Nevertheless, (6) is a reasonable choice, because essentially we are not interested in the exact reconstruction of the pattern vectors $(\mathbf{y}_\alpha - \mathbf{y}_\beta)$, but rather want to reconstruct the correct grouping indicated by \mathbf{y}_α . In this respect, the overlap causes a self-reinforcement of the groups proportional to their size rather than to distort the grouping. Consequently, the CLM dynamics becomes biased towards finding larger groups faster than smaller ones.

C. Generalization to New Patterns

In general, the feature domain $\mathcal{F} \in \mathbb{R}^d$ is a high-dimensional discrete or a non finite set and the training examples cover only a small and discrete subset of \mathcal{F} . Therefore we have to generalize the discrete interaction matrix \hat{F} obtained by the Hebbian learning as described above to an interaction function defined on the full feature domain $f : \mathcal{F}^2 \rightarrow \mathbb{R}$. As proposed already in the context of the quadratic optimization approach to CLM learning [28], generalization can be obtained by decomposing the interaction function into a linear combination of a set of K arbitrary symmetric basis interaction functions $g_{rr'}^j = g^j(\mathbf{m}_r, \mathbf{m}_{r'}) : \mathcal{F}^2 \rightarrow \mathbb{R}, j = 1, \dots, K$ which are defined on the whole feature space such that

$$f_{rr'} = \sum_j c_j g_{rr'}^j. \quad (13)$$

The analysis above provides an elegant way to choose suitable coefficients c_j by projecting the Hebbian correlation matrix \hat{F} in (6) onto the basis functions g^j

$$c_j = \sum_{r,r'} \hat{f}_{rr'} g_{rr'}^j / \| \mathbf{g}^j \|. \quad (14)$$

We see that the basis functions embody contextual knowledge that is used to reduce the dimension of the learning problem. Equation (13) is only a good approximation of $f_{rr'}$, if there is a high overlap between $\hat{f}_{rr'}$ and the shape of $g_{rr'}^j$ such that the coefficients c_j describe the relevant components of $\hat{f}_{rr'}$. Therefore the manual definition and adjustment of suitable basis interactions requires both problem-specific knowledge and detailed knowledge about the functionality of the CLM and results in much training and testing work for the user. Thus we turn now to the problem how suitable basis functions can be generated automatically without making assumptions about the shape of the interaction function [26].

Assume the basis functions as binary step functions ($g_{rr'}^j \in \{0, 1\}$) that describe a disjunct partitioning of the space \mathcal{F}^2 ($g_{rr'}^j g_{rr'}^i = \delta_{ij}$). This partitioning should be symmetric according to feature exchange ($g_{rr'}^j = g_{r'r}^j$) to ensure symmetry of the matrix F . Now we can simplify the projection of the theoretical interaction onto the basis functions (14) to the average interaction value within the respective region of \mathcal{F}^2 .

$$c_j = \sum_{r,r'} \hat{f}_{rr'} g_{rr'}^j / \sum_{rr'} g_{rr'}^j. \quad (15)$$

We can approximate this average by random sampling of feature pairs $(\mathbf{m}_r, \mathbf{m}_{r'})$ from the training set, evaluation of the basis functions, and using $\hat{f}_{rr'} = 1$, if the features belong to the same group, and $\hat{f}_{rr'} = -1/(L-1)$, if they belong to different groups, respectively.

We see, that (13) can only be a good approximation of $f_{rr'}$, if the partitioning described by the basis functions defines regions which consist either mainly of feature-pairs with $\hat{f}_{rr'} > 0$ or mainly of feature-pairs with $\hat{f}_{rr'} < 0$, so that the interactions coefficients can adopt high absolute values. Therefore the borders of the basis functions should as good as possible describe the borders between positive and negative values $\hat{f}_{rr'}$ in \mathcal{F}^2 .

There are two principal approaches to this problem: The first is to discretize \mathcal{F}^2 to a lattice structure. However through the curse of dimensionality in \mathcal{F}^2 for a fine discretization this approach results in an impractical high number of basis functions.

The second approach is to train a standard classifier in \mathcal{F}^2 to separate the regions of $\hat{f}_{rr'} > 0$ and $\hat{f}_{rr'} < 0$ from each other. This approach was tested in [27] by application of the Multi Layer Perceptron (MLP) and the Support Vector Machine (SVM) on the problem of fluorescence cell image segmentation and compared with results of the original QCO approach. The application of SVM showed good qualitative results, while in case of MLP it was difficult to choose an appropriate network architecture and learning parameters. However, practically the application of SVM and MLP seems not adequate, because in real world problems like the cell segmentation example the regions of $\hat{f}_{rr'} > 0$ and $\hat{f}_{rr'} < 0$ have a severe overlap, such that they cannot be separated exactly. As a result, the learned

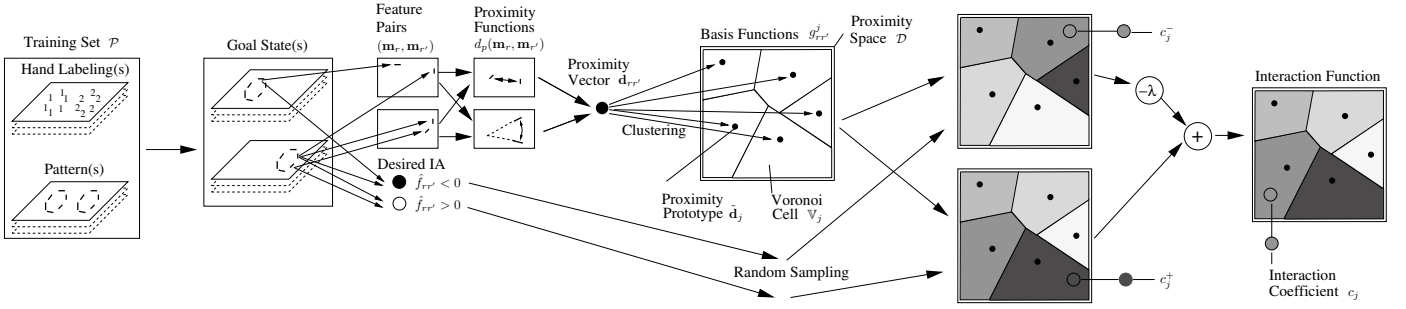


Fig. 3. Sketch of the Approximated Hebbian Learning (AHL). Starting from a set of hand-labeled patterns which can be interpreted as goal states of the CLM-dynamics, the learning algorithm extracts two kinds of information. Each pair of features $(\mathbf{m}_r, \mathbf{m}_{r'})$ is transferred into a proximity vector $\mathbf{d}_{rr'}$ by applying a set of predefined proximity functions $d_p(\mathbf{m}_r, \mathbf{m}_{r'})$, $p = 1, \dots, P$. Each feature pair is also associated with a desired positive or negative interaction value. In the first learning phase, a set of randomly chosen proximity vectors $\mathbf{d}_{rr'}$ is clustered to obtain a Voronoi partitioning of the proximity space. In the second learning phase, the distribution of positive and negative interaction within the proximity space is measured by counting feature pairs with positive and negative associated interaction within the Voronoi cells (dark color shows low number of feature-pairs and light color shows high number). These two density distributions are combined by the scaling parameter λ to the resulting interaction function (there dark color symbolizes negative interaction and light color symbolizes positive interaction).

classification functions from MLP and SVM were very complex and needed much more evaluation time than the results of QCO on the basis function approach here preferred. This point is essential for the application of the CLM, because the number of lateral weights $f_{rr'}$ is quadratic in the number of features in a pattern such that already the computation of the matrix F can be very resource demanding if the definition of the basis functions is too complex.

Learning of a qualitatively effective and computational efficient interaction function demands adaptation of learning parameters, like the kernel shape and margin size of SVM, to the presented learning problem.

We now present an alternative to these two approaches for finding basis functions based on the unsupervised learning principle of vector quantization (VQ) that results in a high evaluation speed.

D. Data Driven Generation of Basis Interactions

As $f_{rr'}$ and thus as well $g_{rr'}^j$ express some degree of mutual compatibility or proximity of a feature pair $(\mathbf{m}_r, \mathbf{m}_{r'})$ we first transform each pair into a generalized proximity space \mathcal{D} by forming vectors

$$\mathbf{d}_{rr'} = [a_1 d_1(\mathbf{m}_r, \mathbf{m}_{r'}), \dots, a_P d_P(\mathbf{m}_r, \mathbf{m}_{r'})]^T. \quad (16)$$

Each component $d_p(\mathbf{m}_r, \mathbf{m}_{r'})$ defines a proximity function according to some properties of the features \mathbf{m}_r and $\mathbf{m}_{r'}$ e.g. the local distance between two features. These components are normalized by their variance in the pattern or left unchanged, if the variance is zero:

$$a_p = \begin{cases} \frac{1}{\sigma^2(d_p(\mathbf{m}_r, \mathbf{m}_{r'}))} & : \sigma^2(d_p(\mathbf{m}_r, \mathbf{m}_{r'})) > 0 \\ 1 & : \sigma^2(d_p(\mathbf{m}_r, \mathbf{m}_{r'})) = 0 \end{cases}. \quad (17)$$

It has to be emphasized, that the proximity functions $d_p(\mathbf{m}_r, \mathbf{m}_{r'})$ need not define a metric in a strict sense. The only constraint is that they are symmetric under feature-exchange $d_p(\mathbf{m}_r, \mathbf{m}_{r'}) = d_p(\mathbf{m}_{r'}, \mathbf{m}_r)$, allowing a high freedom in defining grouping principles for the interaction function. Additional properties, like translation, rotation, and scale invariance of the proximity functions, can be required to achieve a generalization

of the desired grouping behavior according to object properties, like position, size and orientation.

We use a variation of vector quantization, the activity equilibration vector quantization AEV [6], to map the proximity vectors $\mathbf{d}_{rr'}$ to a set of K prototypes $\tilde{\mathbf{d}}_j$. Then we choose as j -th basis interaction g^j the membership function

$$g_{rr'}^j = \begin{cases} 1 & : (\mathbf{m}_r, \mathbf{m}_{r'}) \in V_j \\ 0 & : (\mathbf{m}_r, \mathbf{m}_{r'}) \notin V_j \end{cases} \quad (18)$$

of the corresponding j -th multidimensional Voronoi cell [26]:

$$V_j = \{(\mathbf{m}_r, \mathbf{m}_{r'}) \mid \forall i \neq j : \|\mathbf{d}_{rr'} - \tilde{\mathbf{d}}_j\| \leq \|\mathbf{d}_{rr'} - \tilde{\mathbf{d}}_i\|\}. \quad (19)$$

Since clustering techniques are dependent on the initialization of prototypes there can occur states with “unused” prototypes, which results in low activation of prototypes $\tilde{\mathbf{d}}_j$ measured by low values of $A_j = |V_j|$. The AEV uses a heuristic to reinitialize prototypes with low activation close to prototypes with a high activation that causes an activity equalization of the prototypes such that we neglect the normalization term in (15) in the following.

The representation of the basis functions as Voronoi cells in the proximity space simplifies the computation of the interaction weights $f_{rr'}$. For a given feature pair we have to compute the proximity vector $\tilde{\mathbf{d}}_{rr'}$, evaluate its nearest neighbor from the set of K prototypes $\tilde{\mathbf{d}}_j$, and return the interaction coefficient c_j of this prototype. Figure 4 shows a sketch of this procedure.

E. Aspects of Implementation

From the special structure of the theoretically derived interaction matrix \hat{F} in section III-B and the basis functions in the last section we come to an efficient straight-forward learning algorithm with the possibility to control the segmentation level of the grouping result.

The theoretical interaction weights $\hat{f}_{rr'}$ (6) can adopt only one positive and one negative value, where the negative is scaled against the positive one by the factor $\lambda = \frac{1}{L-1}$. Neglecting the normalization because of the activity equilibration we rewrite (15) as

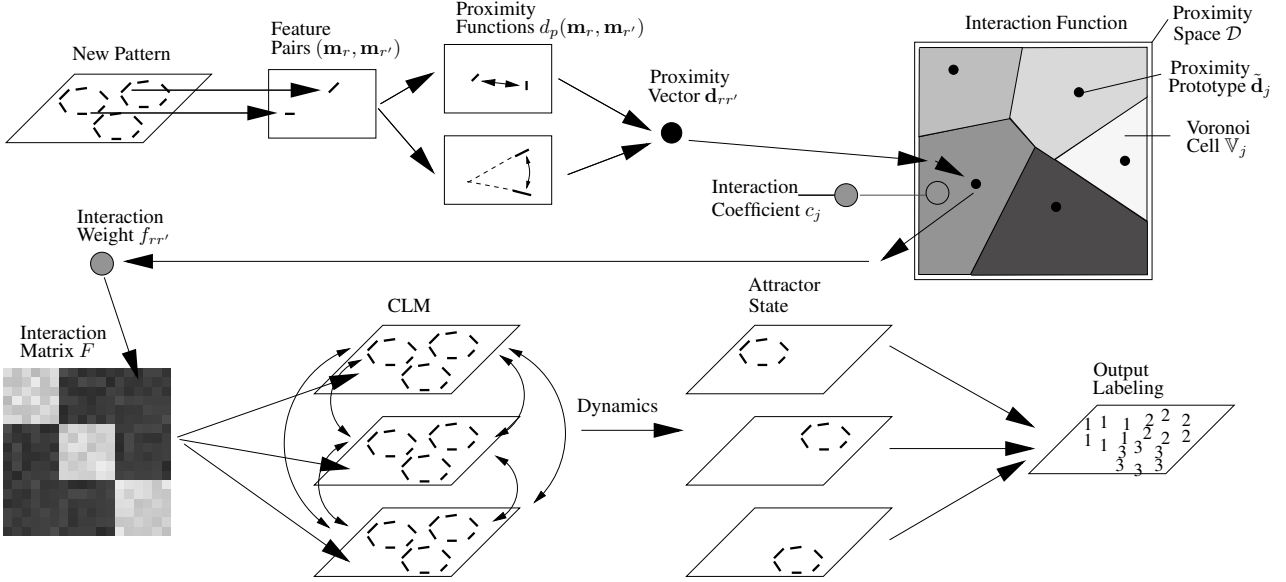


Fig. 4. Sketch of the application process. An interaction matrix F for a new pattern is constructed by mapping each pair of features $(\mathbf{m}_r, \mathbf{m}_{r'})$ onto their proximity vector $\mathbf{d}_{r,r'}$ and taking the interaction coefficient c_j of the nearest proximity prototype $\bar{\mathbf{d}}_j$ as their pairwise interaction. Then F is used as the weight-matrix for the lateral connections within the layers of the CLM and the dynamics of the CLM is simulated, until it converges to an attractor state which defines the output labeling.

$$c_j = \sum_{r,r'|\alpha(r)=\alpha(r')} g_{r,r'}^j - \sum_{r,r'|\alpha(r)\neq\alpha(r')} \lambda g_{r,r'}^j. \quad (20)$$

Denote the positive and negative part of the c_j by

$$c_j^+ = \sum_{\substack{(\mathbf{m}_r, \mathbf{m}_{r'}) \in V_j \\ \alpha(r)=\alpha(r')}} 1; \quad c_j^- = \sum_{\substack{(\mathbf{m}_r, \mathbf{m}_{r'}) \in V_j \\ \alpha(r)\neq\alpha(r')}} 1. \quad (21)$$

Then the computation of c_j means counting of feature pairs $(\mathbf{m}_r, \mathbf{m}_{r'})$ with positive and negative values $\hat{f}_{r,r'}$ within the corresponding Voronoi cells. Obviously

$$\mathbf{c} = \mathbf{c}^+ - \lambda \mathbf{c}^-, \quad (22)$$

where \mathbf{c} , \mathbf{c}^+ and \mathbf{c}^- are the vectors of the values c_j , c_j^+ and c_j^- , $j = 1, \dots, K$ respectively.

The quality of the so-constructed interaction function (13) depends on the angle between the vectors \mathbf{c}^+ and \mathbf{c}^- , because we can only get clear positive or negative interaction weights $f_{r,r'}$, if for each basis function $g_{r,r'}^j$ either the value of c_j^+ dominates against c_j^- or vice versa. Otherwise, the values of the c_j^+ and c_j^- annihilate each other and the resulting c_j are close to zero. In this sense the learned interaction is optimal, if $(\mathbf{c}^+)^T \mathbf{c}^- = 0$ holds, while in the other cases it depends on the strength of the factor λ and the relation between c_j^+ and c_j^- whether the corresponding c_j become positive or negative. Thus the angle between \mathbf{c}^+ and \mathbf{c}^- is a good measure for the evaluation of learning success. It yields information whether the basis functions have been chosen appropriately and whether the labeled training pattern is compatible with this choice.

Regarding the scaling factor λ as a variable parameter, we find (see Fig. 5) that equation (22) defines a line within the learning parameter space \mathbb{R}^K . The choice of λ affects the grouping behavior of the CLM: a decrease of λ biases the interaction

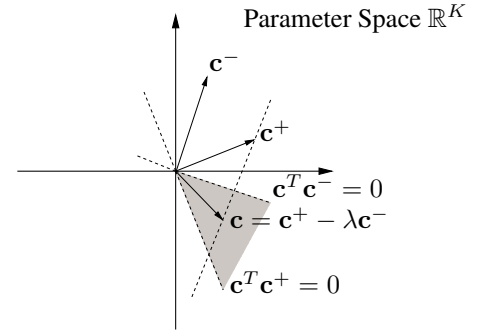


Fig. 5. Choice of λ : The vector \mathbf{c}^+ describes the projection of the positive components of F onto the basis functions $g_{r,r'}^j$, while \mathbf{c}^- shows the projection of the negative components of F to the basis functions. If we want to compute \mathbf{c} by $\mathbf{c} = \mathbf{c}^+ - \lambda \mathbf{c}^-$, the fulfillment of the consistency conditions is implied, if we choose λ in such a way, that \mathbf{c} lies between the lines $\mathbf{c}^T \mathbf{c}^- = 0$ and $\mathbf{c}^T \mathbf{c}^+ = 0$. We see, that the importance of the right choice of λ depends essentially on the angle between \mathbf{c}^+ and \mathbf{c}^- . If both vectors are linearly independent, we can choose $\lambda > 0$ almost arbitrarily, while the feasible interval for λ shrinks with the angle between \mathbf{c}^+ and \mathbf{c}^- . Therefore the qualitative performance of the learning approach depends on a good choice of basis functions which are orthogonal for feature pairs within and between groups.

function towards a higher attraction, which results in fewer but bigger groups in the segmentation of new patterns; an increase biases the interaction function towards a higher repulsion, which results in more but smaller groups in the output of the CLM. Thus after training of \mathbf{c}^+ and \mathbf{c}^- , we can use λ to adjust the CLM to a finer or rougher segmentation of the input pattern.

To constrain the range of λ , we once more inspect the consistency conditions (5) which are fulfilled, if the left hand side of equation (5) is always < 0 and the right hand side is always > 1 . By separate summation of all left hand sides and all right hand sides, the insertion of the superposition of basis functions (13), and the comparison of the result with the definitions of the

components c_j^+ and c_j^- in (21), we get the two inequalities:

$$\mathbf{c}^T \mathbf{c}^+ > 0 \quad \text{and} \quad \mathbf{c}^T \mathbf{c}^- < 0. \quad (23)$$

From (22) and (23) we get an upper and lower bound for λ (see Fig. 5) as:

$$\frac{(\mathbf{c}^+)^T \mathbf{c}^+}{(\mathbf{c}^-)^T \mathbf{c}^+} > \lambda > \frac{(\mathbf{c}^+)^T \mathbf{c}^-}{(\mathbf{c}^-)^T \mathbf{c}^-}. \quad (24)$$

We can perform a linear search between the upper and lower bound to adjust λ to a desired segmentation level. In the following we will use the convention, that \mathbf{c}^+ and \mathbf{c}^- will be normalized by $\frac{1}{\|\mathbf{c}^+\|}$ and $\frac{1}{\|\mathbf{c}^-\|}$, where $\|\cdot\|$ means the sum of all (per definition nonnegative) values c_j^+ respectively c_j^- to uniquely specify the value of λ .

In this framework, the only choice left to the user is the selection of appropriate feature properties, like color, position and orientation, an adequate distance function $d_p(\mathbf{m}_r, \mathbf{m}_{r'})$, and the number of basis functions K . From this choice, the learning is fully self-contained with λ as the only free parameter left for putting a bias on larger or smaller groups.

F. Extension to Training Sets with several Training Patterns.

Up to now, we have assumed that the training set consists only of a single pattern. This approach can be sufficient to learn a grouping behavior, because our learning algorithm mainly uses statistics on pairwise feature relations and their correlation according to the hand labeling. A single pattern can comprise a huge set of pairwise feature combinations and therefore can be sufficient to represent the statistics of the desired grouping behavior.

The algorithm can be applied to a training set with several patterns without major modifications. We can simply perform the scaling of proximity functions (17), the clustering step of AEV, and the estimation of \mathbf{c}^+ and \mathbf{c}^- (21) on a representative sample of feature pairs which is uniformly chosen from all patterns, provided the two features forming a pair are drawn from the same pattern. An implementation of the whole learning algorithm which we call Approximated Hebbian Learning (AHL), can be found in the appendix IV, a summary of the complete approach is sketched in Fig. 3. An additional sketch for the application process of the CLM is given in Fig. 4.

G. Estimation of the Background Layer Strength

If the CLM contains a special background layer g , the strength of self-interaction m in the background layer has to be weighted against the lateral interactions $f_{rr'}$. A feature \mathbf{m}_r is assigned to the background, if m is bigger than the support in all other layers, while it is assigned to the relevant groups, if the support in at least one layer is higher than m . Therefore m is constrained by additional consistency conditions, where m has to be bigger or smaller than the maximal lateral support $\max_{\alpha} \sum_{r'} f_{rr'} x_{r\alpha}$ of the feature \mathbf{m}_r in the other layers $\alpha \neq b$, whether the feature \mathbf{m}_r should be assigned to the background or not.

We try to estimate a suitable value for m by assuming the average maximal support of all features in the background as lower bound m_{low}

$$m_{low} = \frac{\sum_{\mathbf{m}_r | \alpha(r)=b} \max_{\alpha} (\sum_{r'} f_{rr'} y_{r'\alpha})}{\#\mathbf{m}_r | \alpha(r)=b}, \quad (25)$$

and the average maximum support of all features in the figure groups as upper bound m_{up} for m .

$$m_{up} = \frac{\sum_{\mathbf{m}_r | \alpha(r) \neq b} \max_{\alpha} (\sum_{r'} f_{rr'} y_{r'\alpha})}{\#\mathbf{m}_r | \alpha(r) \neq b}. \quad (26)$$

A simple heuristic which proved to work is, to set m between m_{low} and m_{up} , e.g.

$$m = (m_{low} + 3m_{up})/4. \quad (27)$$

IV. APPLICATION

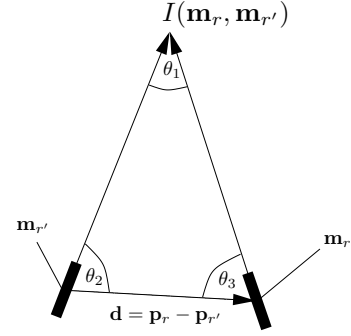


Fig. 6. Proximity measures for oriented line segments: Two line segments \mathbf{m}_r and $\mathbf{m}_{r'}$ define a triangle $(\mathbf{p}_r, \mathbf{p}_{r'}, I(\mathbf{m}_r, \mathbf{m}_{r'}))$, where $I(\mathbf{m}_r, \mathbf{m}_{r'})$ is the intersection point of the two line segments. The proximity of \mathbf{m}_r and $\mathbf{m}_{r'}$ is expressed by the local distance $\|\mathbf{d}\|$ and the three angles within the triangle, where we swap θ_2 and θ_3 , if $\theta_3 > \theta_2$ to ensure the symmetry of $\mathbf{d}_{r,r'}$. So $\mathbf{d}_{r,r'} = (a_1 \|\mathbf{d}\|, a_2 \theta_1, a_3 \theta_2, a_4 \theta_3)^T$. If $I(\mathbf{m}_r, \mathbf{m}_{r'})$ does not exist, because both features have parallel orientation, θ_1 is set to zero, describing an infinite sharp angle. θ_2 and θ_3 describe the angles between the orientation of the two features and the vector \mathbf{d} under the constraint $\theta_1 + \theta_2 + \theta_3 = \pi$.

For systematic benchmarking, we use artificially generated datasets and investigate the problem of binding local line segments to regular geometrical polygons of certain size and shape.

A polygon is described by the parameters $(x, y, R, \theta_{start}, S)$, where x, y and R define the center and radius of a circle. Starting on the circumference of this circle at the angle $\theta = \theta_{start}$, we get a regular geometrical object by successively increasing θ with $\frac{2\pi}{S}$ and stepping to the next point on the circumference at the angle θ . Therefore S defines the shape of the polygon as point, line, triangle, square, etc.

These lines are divided in into small line segments of equal length which define feature vectors $\mathbf{m}_r = (\mathbf{p}_r, \varphi_r)^T$ by their position $\mathbf{p}_r = (x_r, y_r)$ and orientation $\mathbf{o}_r = (dx_r, dy_r)$ mapped to an orientation angles $\varphi_r \in [0, \pi]$.

A pattern is given by one to five objects of same shape S and similar size R , where the remaining object parameters are chosen randomly as $x, y \in [R, 3R]$ and $\theta_{start} \in [0, 2\pi]$.

The CLM task is to segment patterns which contain objects of the same shape S and size R into the distinct objects, i.e. it has to adapt the compatibilities to detect typical angles and distances within the observed polygons. To model this task with basis functions, we assume that each two features define a triangle (see Figure 6). We combine the local distance between two features with the three angles within the corresponding triangle to a four-dimensional proximity space \mathcal{D} .

Note that this choice includes some redundancy, since a triangle is specified uniquely by one side length and two inner angles.

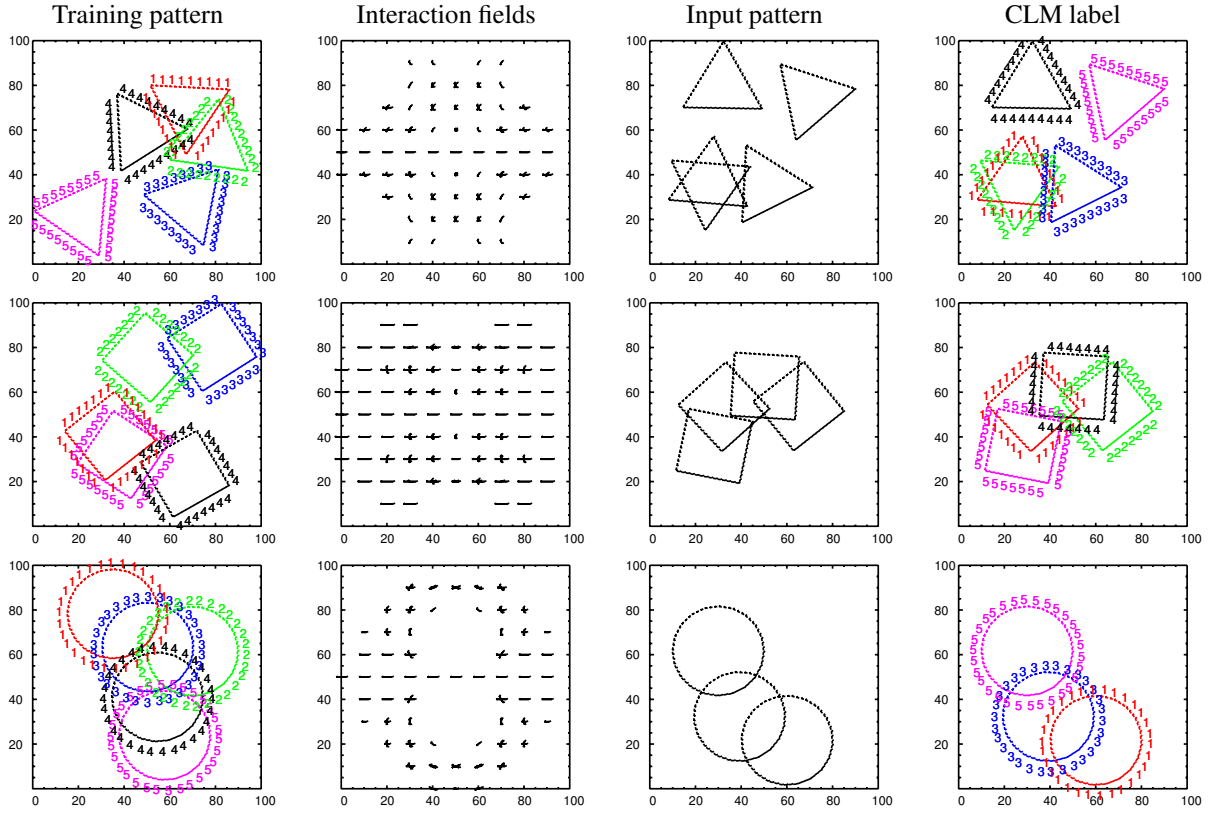


Fig. 7. Training results for contour grouping on artificial data: the first column shows three training patterns, with five ideal triangles, squares and circles of size $R = 20$. The labels of the features are displayed by color and number. The second column displays parts of interaction functions resulting from the application of AHL ($\lambda = 0.5$) on the training patterns. In each diagram a discretization (according to position and orientation) of line segments is plotted which have a positive interaction with a reference segment at position (50,50) with horizontal orientation. The length of the line segments reflects the strength of positive interaction. The negative interactions are not visualized here. They show the remaining feature orientations that are positioned on the same lattice structure as seen in the positive interaction fields. The third column shows unlabeled test patterns. The last column shows results of a ten-layered (nine figure-layer, one groundlayer) CLM which applies the learned interaction function to the test image.

However, it is a feature of the AHL learning approach that it can deal with redundant and spurious proximity dimensions, since positive and negative interactions detach along the relevant dimensions.

We cluster the proximity vectors $d_{r,r'}$ occurring in one typical training pattern to 100 prototype vectors \vec{d}_j whose Voronoi cells define the basis functions according to (18). Then we apply the algorithms QCO ($\kappa = 100$) and AHL ($\lambda = 0.5$) to estimate the interaction coefficients c_j of these basis functions. Some examples for training patterns, learned interaction function, input pattern and CLM-output are shown in Figure 7 for datasets containing triangles (S=3), squares (S=4) and approximated circles (S=20) of size $R = 20$.

Figure 8 shows the labelings resulting from the triangle interaction function in Figure 7 applied on two different configurations of two triangle input groups. The upper row shows two triangles in uncorrelated orientation, while the lower row shows a more complicated configuration with two neighboring triangle sides in parallel orientation. Each column belongs to a concrete choice of the segmentation control parameter λ .

Equation (24) suggests for the learned interaction function an upper bound $\lambda_{max} \approx 7.5$ and a lower bound $\lambda_{min} \approx 0.45$ for λ .

In the first case of uncorrelated orientations the choices of

$\lambda = 0.2$, $\lambda = 2$ and $\lambda = 4$ result in a correct labeling. This can be explained by the fact that only parallel line segments or line segments with an inner angle of 60 degrees have positive interaction, while all other pairs of line segments have negative interaction. Between $\lambda = 4$ and $\lambda = 6$ the triangles start to decompose into single lines, because the interactions between lines with an inner angle of 60 degrees are shifted into the negative quadrant. This splitting process proceeds with the increase of λ , until each layer of the CLM is occupied by one group. A further increase of λ has no effect on the output labeling, since the consistency conditions (3) imply that the groups with the least inter-group inhibition are merged in the same layers.

In the second case the parallel triangle sides are connected by additional positive interaction. For moderate values of λ , e.g. $\lambda = 2$ to 4, this attraction between the two groups is compensated by negative connections between high distance line segments of an inner angle 60 degrees such that the correct labeling is returned. At small values of λ , e.g. $\lambda = 0.2$, the necessary negative interaction is weakened and both groups are merged to the same layer, while larger values of λ result in an over segmentation of the pattern.

This example shows that the segmentation control parameter λ can be explored by seeking a trade off between an over and under segmentation of a test pattern. The range of feasible val-

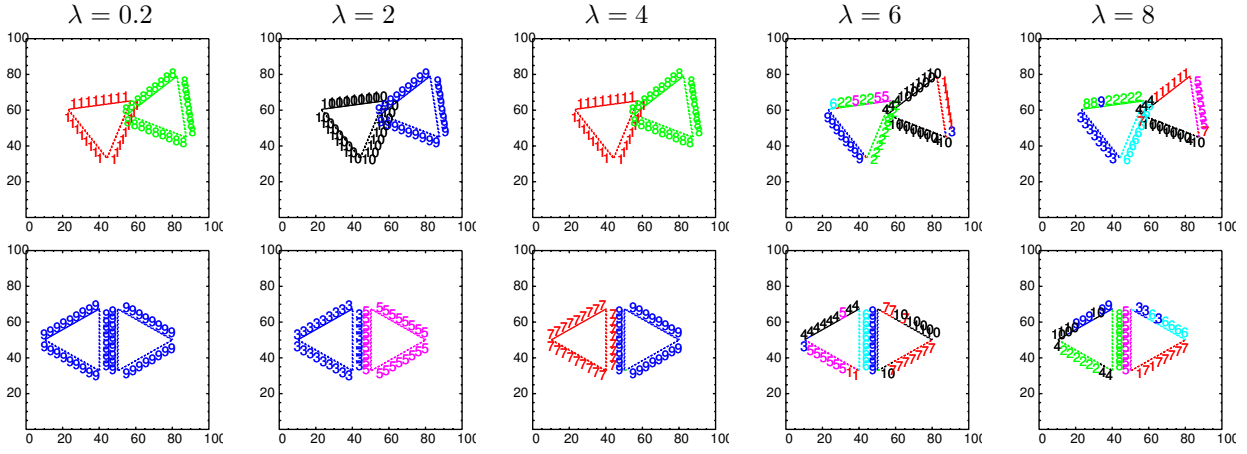


Fig. 8. Output labelings of a ten-layered CLM for two triangles in different configurations resulting from the triangle interaction function at different values of λ .

ues of λ depends on the concrete configuration in the presented input pattern. However, the bounds λ_{max} and λ_{min} indicate the size of the interval and give a hint to start the search at the center of $[\lambda_{min}, \lambda_{max}]$.

The following four experiments address the evaluation of our learning method according to the adaptation of object size and shape, the influence of spurious features and the influence of errors in feature position and orientation respectively. Therefore we have to measure the grouping quality achieved with the CLM and the learned interaction function.

Quality Measure: The resulting CLM labels $\alpha^C(r)$ are compared to the desired goal labels $\alpha^G(r)$ from the pattern generator by computing the percentage of features that have been labeled correctly. For an algorithmic description see appendix V.

Adaptation to object size and shape: We investigate the robustness of the algorithms QCO and AHL using $S \in \{3, 4, 20\}$ and $R \in \{15, 20, 25, 30, 35, 40, 45, 50, 55, 60\}$ to generate for each combination of (algorithm, S, R) 10 random training patterns with 5 objects. Each of the ten resulting interaction functions is tested with a ten-layered (one ground layer, nine figure layers) CLM on ten new patterns with the same object properties S and R , but with the number of objects varying from one to five. Figure 9 plots the average grouping quality over all 100 resulting CLM responses.

The results show that AHL is always able to adapt the correct compatibilities for the relevant feature combinations in the patterns. Remarkably the desired grouping behavior is learned from a single pattern only, where the property of rotation invariance is guaranteed from the design of the proximity functions.

QCO also performs well for R from 15 to 30. For larger values of R we observe a significant decrease in grouping quality for QCO. The reason is that a pattern with small R has a smaller number of features than a pattern with large R , because the length of the borderline grows, while the length of each local edge feature is fixed. In QCO, for each of these features, a set of $L - 1$ consistency conditions has to be computed which are extended by the margin κ which is uniform for all conditions (see Appendix II). The choice of $\kappa = 100$ seems not suitable for all sets of conditions that arise when varying R . Actually the qualitative performance of QCO for patterns with a higher

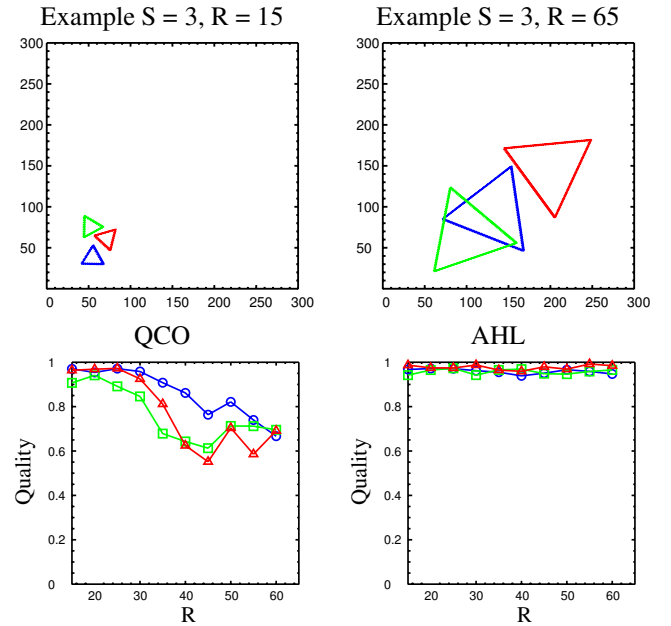


Fig. 9. Average Grouping Quality for Objects of different Size: Plots of grouping quality against object size R (15 to 65) for $S = 3$ (triangles), 4 (squares) and 20 (circles). Top row shows stimulus examples.

value of R can be improved, if the value of κ is increased with the number of features in the patterns. A second disadvantage of QCO against AHL lies in the higher computation time which becomes significant for patterns which consist of many features.

Influence of spurious features: In the other three experiments we investigate how different types of errors in the datasets affect the grouping performance of the CLM. Again we test the learning algorithms QCO and AHL on objects $S \in \{3, 4, 20\}$, but this time keep $R = 20$ fixed. Instead we vary $\eta \in \{5\%, 10\%, 15\%, 20\%, 25\%, 30\%, 35\%, 40\%, 45\%, 50\%\}$, the percentage of features which are removed from the pattern. These features are replaced by randomly generated features m_r with $x_r, y_r \in [0, 5R]$, $\varphi_r \in [0, \pi]$ that are labeled as background features. The average grouping quality over 100 CLM responses (10 tests for each of 10 training runs) is plotted in Figure 10 for

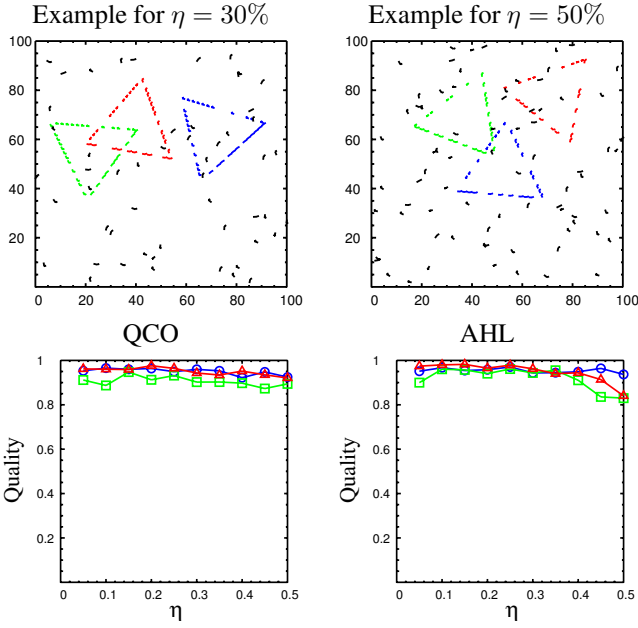


Fig. 10. Statistic Grouping Quality for Noisy Patterns: Plots of grouping quality against percentage of spurious features η (5% to 50%) for $S = 3$ (triangles), 4 (squares) and 20 (circles). Top row shows stimulus examples.

each combination of (algorithm, S , η).

We obtain a high grouping quality for AHL and QCO in all test runs. These results highlight both the robustness of the two learning methods and the robustness of the CLM against fragmentary data, which can be explained by the high redundancy in the matrix of lateral interactions and the redundant character of the CLM. These results also show that the CLM is able to separate the relevant groups from a noisy background using the background layer.

Influence of errors in feature position and feature orientation: In the third and fourth experiment we repeat the second experiment but substitute the error η by two other errors $\varepsilon_p \in \{0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5\}$ and $\varepsilon_o \in \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$. ε_p defines an error in feature position, where all features in the dataset are randomly shifted in x- and y-coordinate by $dx, dy \in [-\varepsilon_p, \varepsilon_p]$. ε_o defines an error in feature orientation, where all features in the dataset are randomly rotated by an angle $d\varphi \in [-\varepsilon_o, \varepsilon_o]$. The average grouping qualities of all combinations of (algorithm, S , ε_p) are plotted in Figure 11, while the average grouping qualities of all combinations of (algorithm, S , ε_o) are plotted in Figure 12.

The results show that the learned grouping principles are fairly robust against errors in feature extraction. For the errors in feature position both AHL and QCO show high grouping quality, which can be explained by the interaction fields in Fig. 7. There each feature gets positive feedback not only from features with the correct orientation, which are in a small interval of local distance, but also from features which are shifted parallel to the relevant orientation.

In the case of errors in feature orientation there is a noticeable decrease in grouping quality which is serious for AHL. This difference between AHL and QCO becomes clear, if we inspect

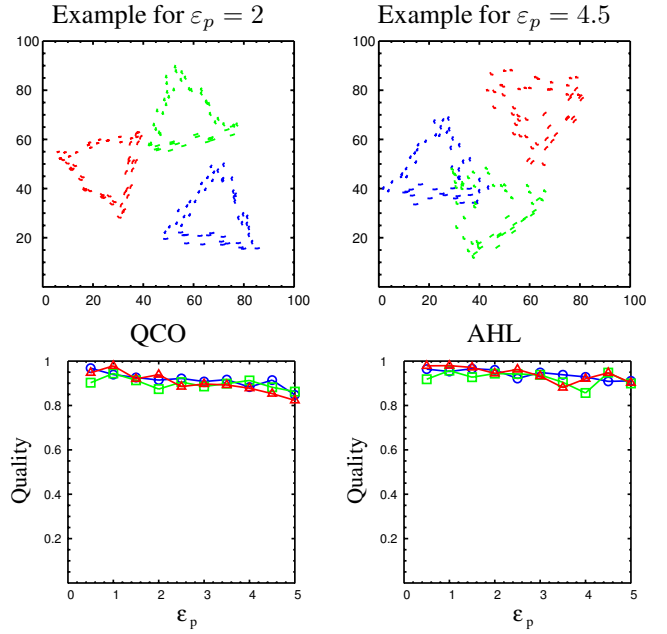


Fig. 11. Statistic Grouping Quality for Translation Errors: Plots of grouping quality against translation error ε_p (0.5 to 5) for $S = 3$ (triangles), 4 (squares) and 20 (circles). Top row shows stimulus examples.

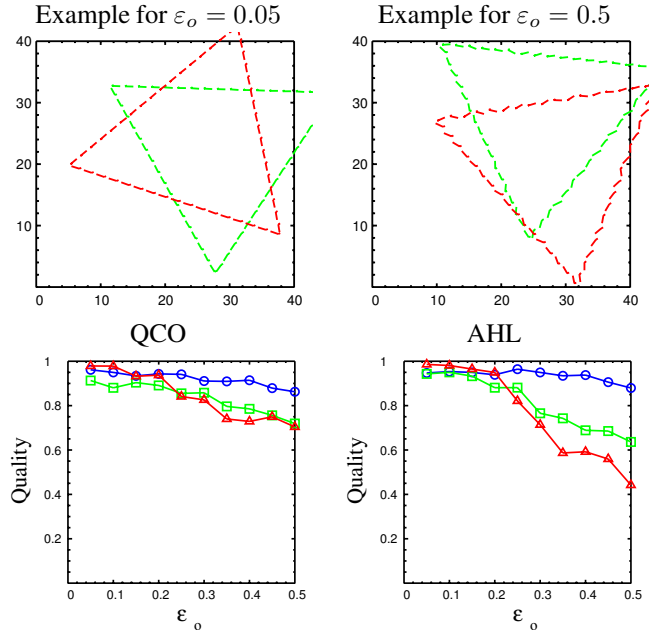


Fig. 12. Statistic Grouping Quality for Rotation Errors: Plots of grouping quality against rotation error ε_o (0.05 to 0.5) for $S = 3$ (triangles), 4 (squares) and 20 (circles). Top row shows stimulus examples.

the kind of errors the AHL-learned CLM makes. In patterns of triangles a triangle is often divided into three separate lines which causes a typical quality of 0.33. Similar in patterns of squares a square is often divided into two perpendicular groups of two parallel lines which causes a typical quality of 0.5. Consequently the interactions of feature pairs with the characteristic inner angles of 60 respectively 90 degrees are not excitatory enough. These errors can be reduced by decreasing the parameter λ to achieve a coarser segmentation of the data into bigger

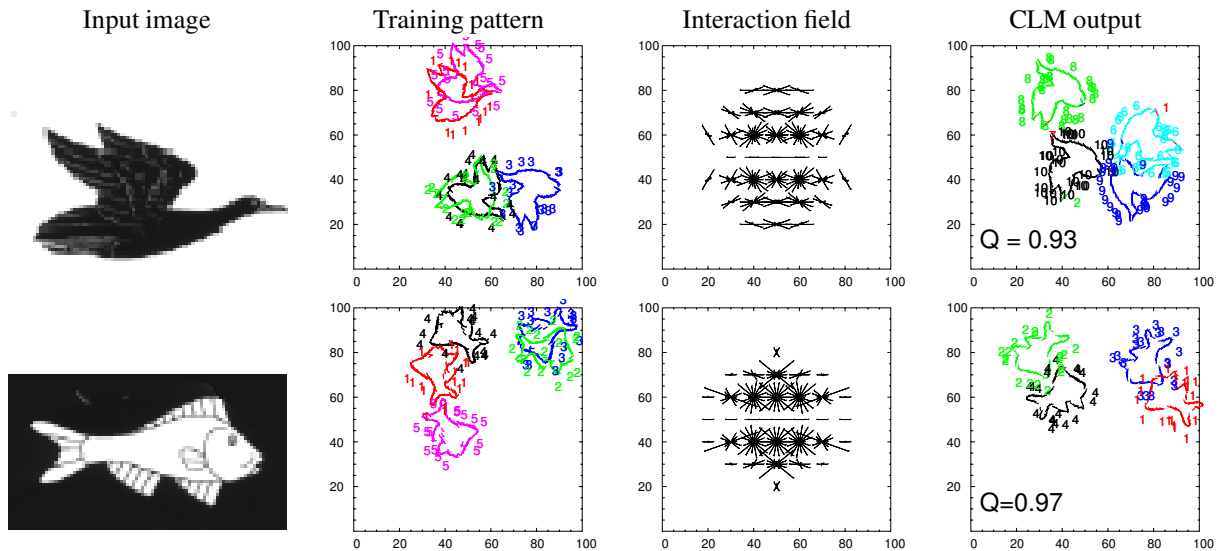


Fig. 13. Contour grouping for more complex objects of a fish and a bird.

groups.

Adaptation to more complex shapes and differentiation of shapes: The previous experiments have shown that the AHL approach can achieve an impressive amount of generalization and robustness, if the representation of the feature space \mathcal{F} and the pairwise proximity space \mathcal{D} are consistent with the grouping behavior specified by the target labeling. For the artificial polygon contours this consistency is a result of the high self-similarity of the polygon shapes, where the rotation invariant interaction function matches at each point of the contour.

In Fig. 13, we demonstrate the application of the contour grouping approach on the more complex and non symmetric contours of a bird and a fish. Therefore we apply sobel-x and sobel-y filters on the two images in column one of Fig. 13 and extract salient edge features $\mathbf{m}_r = (\mathbf{p}_r, \varphi_r)$. The sobel responses are thinned out to one pixel wide contours by non maximum suppression according to the intensity gradients strength and orientation followed by a thresholding with 50% of the maximal intensity gradient strength and a subsampling of the resulting edge set by the factor 2. The orientation angles $\varphi \in [0, \pi]$ are extracted from the sobel-responses $\mathbf{o}_r = (s_{x_r}, s_{y_r})$ at the remaining edge positions $\mathbf{p}_r = (x_r, y_r)$ neglecting the unique direction information such that the edge sets are invariant to a black/white switching of the foreground and background color.

Training and test patterns are constructed by copying randomly rotated and translated versions of the extracted edge sets into a 2D coordinate system, where the maximal distance of an edge to the center of its group is normalized to a fixed radius $R = 30$ (see the second column of Figure 13).

Since the contour of the bird and the fish show varying shape along the border line, the learning problem of extracting relevant inter edge angles and distances is less consistent than before. The higher overlap between positive and negative interaction within the pairwise proximity space can be interpreted as loss of higher order relation information in the AHL-learned ($\lambda = 2$) interaction functions in column three of Figure 13. Both interactions integrate several edge combinations typical for dif-

ferent points on the object contours, which makes them similar to an ordinary edge clustering interaction expressed by excitatory connections for short ranged local distances at all feature orientations. Thereby the interaction functions concentrate the excitatory connections to different types of object specific edge configurations.

The similarity to the edge clustering behavior weakens the ability of the CLM to separate overlapping and close-by contours, like it can be observed for the artificial polygon contours. Column four of Fig. 13 shows two examples, where CLM grouping performance reaches its limits ($Q = 0.93$ respectively $Q = 0.97$). However, the interaction functions are distinct enough from each other to differentiate the two types of contours.

To show this, we merge the two interaction functions into a single CLM network with layer specific interaction functions $f_{rr'}^\alpha$, $\alpha = 1, \dots, L$. As stated in [29], the theoretical convergence and assignment properties of the CLM can be extended to this case such that the CLM output labeling expresses more than a simple segmentation of the input by classifying each group according to the type of the layer the groups are assigned to.

Figure 14 demonstrates two examples for such a scenario. The first row shows output labelings of a $L = 12$ layered CLM on two patterns, each showing two clearly (locally) separable contours of the fish and the bird in random orientation and fixed position. The first six layers, indexed by $F1, \dots, F6$, apply the fish interaction $f_{rr'}^{F1, \dots, F6} = w^{fish} f_{rr'}^{fish}$ from Figure 13 and the other six layers, labeled by $B1, \dots, B6$ apply the respective bird interaction $f_{rr'}^{B1, \dots, B6} = w^{bird} f_{rr'}^{bird}$. The scaling of the interaction functions with the layer weights w^{fish} and w^{bird} is chosen by hand. In the output labeling, the layers $F1, \dots, F6$ prefer the collection of the fish-shaped contours, while the layers $B1, \dots, B6$ prefer the bird-shaped contours.

The second row of Figure 14 shows another scenario with more, simpler but also more similar shapes of the artificial letter contours I, L, T and X. Note that there is a subsumption ambiguity between the letter I and the other letters, and that the

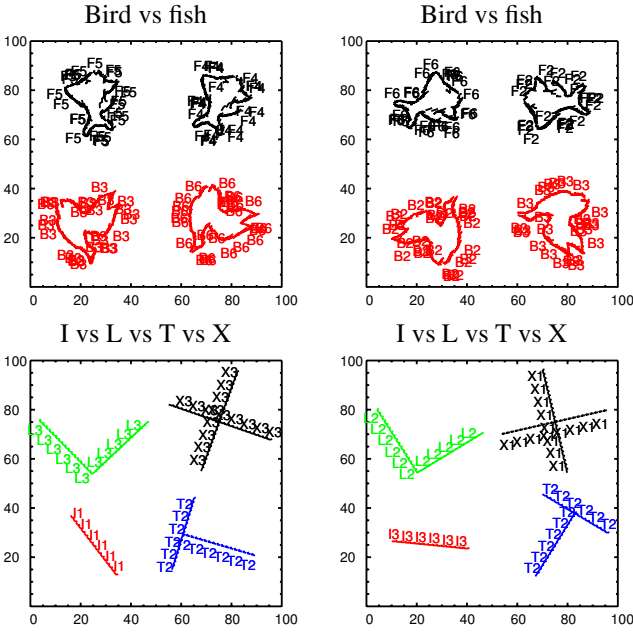


Fig. 14. Contour classification based on layer specific interaction functions. The first column shows labeling results from a $L = 6 + 6$ layered CLM, where the interaction function of layers 1-6 was trained on the fish contour and the layers 7-12 were trained on the bird contour. The second column shows the labeling results from a $L = 3 + 3 + 3 + 3$ layered CLM, where always the interaction functions of three layers were trained on the shape of one of the letters I, L, T and X.

letter T and L only differ in the proportion where the two inferior lines intersect. The output grouping of a $L = 12$ layered CLM is shown for two patterns consisting of (orientationally) uncorrelated and non overlapping instances of the letters I, L, T and X in random orientation and fixed position. Always three layers share the same interaction function respectively matrix trained with AHL ($\lambda = 0.5$) on the respective letter contour: $f_{rr'}^{I1, \dots, I3} = w^I f_{rr'}^I$, $f_{rr'}^{L1, \dots, L3} = w^L f_{rr'}^L$, $f_{rr'}^{T1, \dots, T3} = w^T f_{rr'}^T$, and $f_{rr'}^{X1, \dots, X3} = w^X f_{rr'}^X$. Under suitable layer weight w^I , w^L , w^T and w^X , this network can differentiate the letters I, L, T and X in random orientation, as long as the letters are uncorrelated in rotation and locally separated (compare discussion of the triangle interaction function in Figure 7 and 8).

These two examples demonstrate the principal abilities of the CLM/AHL based contour grouping approach to group and differentiate complex and similar shapes. A more detailed investigation of the pattern matching abilities of a multi-interaction CLM architecture is beyond the scope of this work, because we want to concentrate on the adaptivity of a single-interaction architecture to grouping and segmentation problems.

A. Application to Cell Segmentation

In a more practical domain, we apply the two learning approaches in the area of medical image processing. The goal is to derive an automatic method to estimate the number, positions and form of fluorescent lymphocytes cells on microscope images of blood probes as component of a neural cell detection system (NCDS), like in [12], [14], [15]. Examples of image patches from [15] are shown in the top of Figure 18.

This task is difficult for standard segmentation algorithms

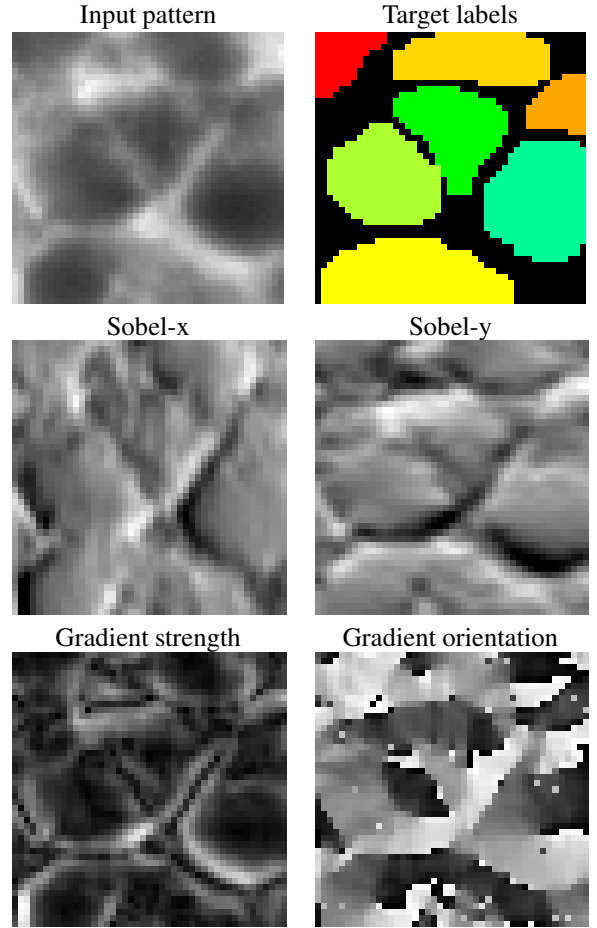


Fig. 15. Feature extraction from fluorescence cell images.

based on contour detection, because changing environment parameters, like concentration of contrast medium or radiology strength, make the corona of the cells highly irregular. This is illustrated in Figure 15 where the sobel-x and sobel-y responses are visualized for an example pattern. In contrast to the contour grouping approach for the artificial contour data not only the salient edge features are extracted for grouping, but each pixel of the input image has to be labeled by the CLM.

The cells show a characteristic structure of dark gray values at the center and brighter values at the border regions. Therefore we try to bind regions with intensity gradients pointing from inside to outside together.

As features we use directed edge vectors $\mathbf{m}_r = (\mathbf{p}_r, \mathbf{o}_r)$, $\mathbf{o}_r = (s_{x_r}, s_{y_r})$, where the orientation vectors \mathbf{o}_r are given by the responses of sobel-x and sobel-y filters at the image position \mathbf{p}_r . Again we assume that two features define a triangle and want to define their proximity by their local distance and the three angles within the triangle. However, since we now have to distinguish equal and opposed directions of the features, we have to improve the proximity functions by defining θ_1 , θ_2 , and θ_3 to be positive or negative whenever the orientation vectors \mathbf{o}_r and $\mathbf{o}_{r'}$ point towards the intersection point $I(\mathbf{m}_r, \mathbf{m}_{r'})$ of the two features or not respectively (see Figure 16). The angles θ_1 , θ_2 , and θ_3 are combined with the local distance $\|\mathbf{p}_r - \mathbf{p}_{r'}\|$ to a four dimensional proximity space.

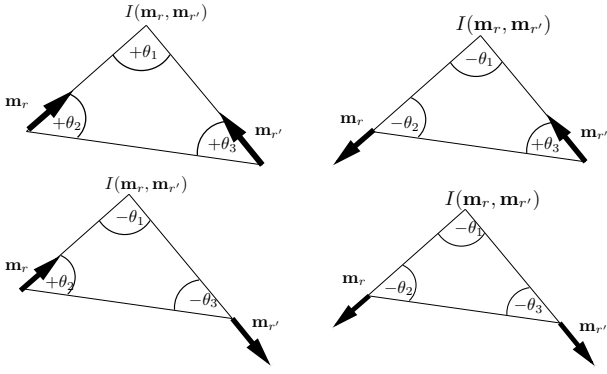


Fig. 16. Proximity functions for directed edge features: The proximity functions for oriented line segments have to be enhanced by additional case differentiations for the directions of the features. The angles θ_1 , θ_2 , and θ_3 are counted positively, if both \mathbf{m}_r and $\mathbf{m}_{r'}$ point towards their intersection point $I(\mathbf{m}_r, \mathbf{m}_{r'})$. They are counted negatively, if both feature point away from $I(\mathbf{m}_r, \mathbf{m}_{r'})$. In the case that only one feature points towards $I(\mathbf{m}_r, \mathbf{m}_{r'})$, the angle at this feature is counted positive, while the others are counted negative. To preserve symmetry under feature exchange θ_2 and θ_3 are swapped, if $\theta_3 > \theta_2$.

The first two rows of Figure 18 show the complete data set of the fluorescence cell images together with the desired hand-labelings. Each image consists of 45×45 pixel, which results in a number of 2025 features that should be grouped by a ten layered CLM (one ground layer, nine figure layer).

To show that our test application has an adequate degree of difficulty, we first try to solve it with a naive approach of direct clustering of the feature vectors \mathbf{m}_r . We apply the standard k-Means-algorithms in two ways on each pattern to generate a set of feature prototypes: In the first case we apply the k-Means with the number of prototypes equal to the layers in the CLM ($k = 10$). In the second case we give the k-Means more problem knowledge by setting the number of prototype vectors to the number of groups in the goal labeling ($k = L^i$).

Then we use each of these sets of prototypes, to divide the corresponding pattern by assigning each feature \mathbf{m}_r to its next neighbor prototype. The results are shown in the last two rows of figure 18. This approach fails as the resulting clusters in the images show arbitrary shape which has no similarity to the structure of cells. The measured quality lies between 20 and 30%.

Therefore, we apply the algorithms QCO ($\kappa = 100$) and AHL (at three different values for λ) to estimate an interaction function $f_{rr'}$ for the CLM from a single pattern and its target labeling. This function consists of 100 basis functions, whose interaction coefficients are adapted from a hand labeling.

Figure 17 visualizes an example of an AHL ($\lambda = 2$) trained interaction function. It shows the interaction field of all edges lying on a 5 pixel wide lattice to the feature at position $\mathbf{p}_r = (0, 0)$ and orientation $\mathbf{o}_r = (1, 0)$. The colors of the displayed sectors describe the interactions of features with the respective position and orientation to the central feature. Light color shows positive interaction and dark color shows negative interaction.

The interaction field shows long range inhibitory and short range excitatory connections which describe the radius of the cells. Further gradient pairs which point towards each other, like they occur between the pixels of neighboring cells, inhibit each other, while gradient pairs which point away from each other,

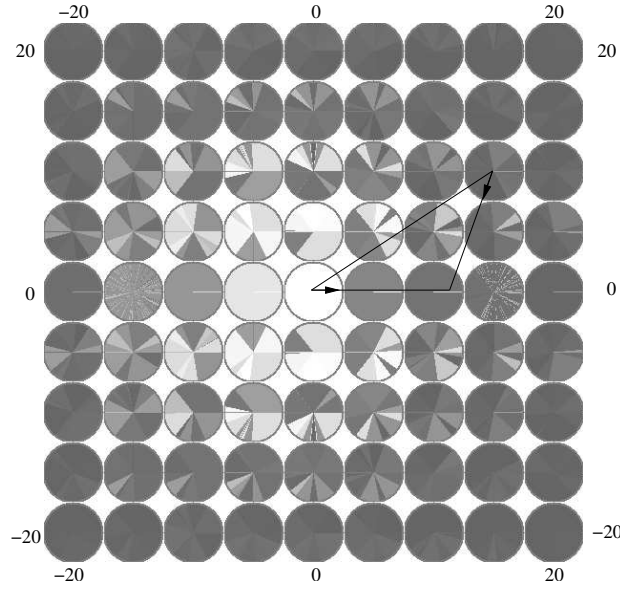


Fig. 17. Interaction field resulting from the application of AHL ($\lambda = 2$) on pattern 1 and its target labeling. All features on a five pixel wide lattice are plotted in the color corresponding to their interaction to the feature with position $\mathbf{p}_r = (0, 0)$ and orientation $\mathbf{o}_r = (1, 0)$. Light color shows positive interaction, dark color shows negative interaction. The plotted triangle sketches the proximity triangle from Fig. 16.

like they can be observed between pixels in same cell, support each other.

Figure 18 contrasts some of the results from the two algorithms AHL and QCO to the results of the direct clustering, while Figure 19 displays the statistics of the qualities of the different approaches according to the whole data set. The third row of Figure 18 shows that, as a result of training with QCO on the first pattern, the CLM is able to separate the different cell regions from each other. This is indicated by an average grouping quality of roughly 70% of the images. However, if we look at the output labels, we see that these results are suboptimal, because the images are slightly oversegmented in cell regions that are split in the output.

Now inspect the results of AHL in the fourth row of figure 18. In this training run, the weight λ is set to a perfect balance of positive and negative interaction at $\lambda = 1$. The results show that AHL has learned the correct grouping behavior since the borders between different labels coincide with borders between cells in the images. But there occur errors, where mainly in the first three images different cells are merged to the same label. Therefore the average grouping performance lies slightly below that of QCO.

Since the groups in the result of AHL seem to be too large and some regions are merged, we conclude that we have to shift the balance between attraction and repulsion of the feature in direction of a higher repulsion to get better results. In this case this shift is done by trial and error, which is shown in the rows five and six of figure 18. We see that we get better results for $\lambda = 2$, where we no longer have a merging of different regions. This enhances the average grouping performance of AHL such that it outperforms the results of QCO. A further increase of λ yields no significant improvement of the grouping behavior.

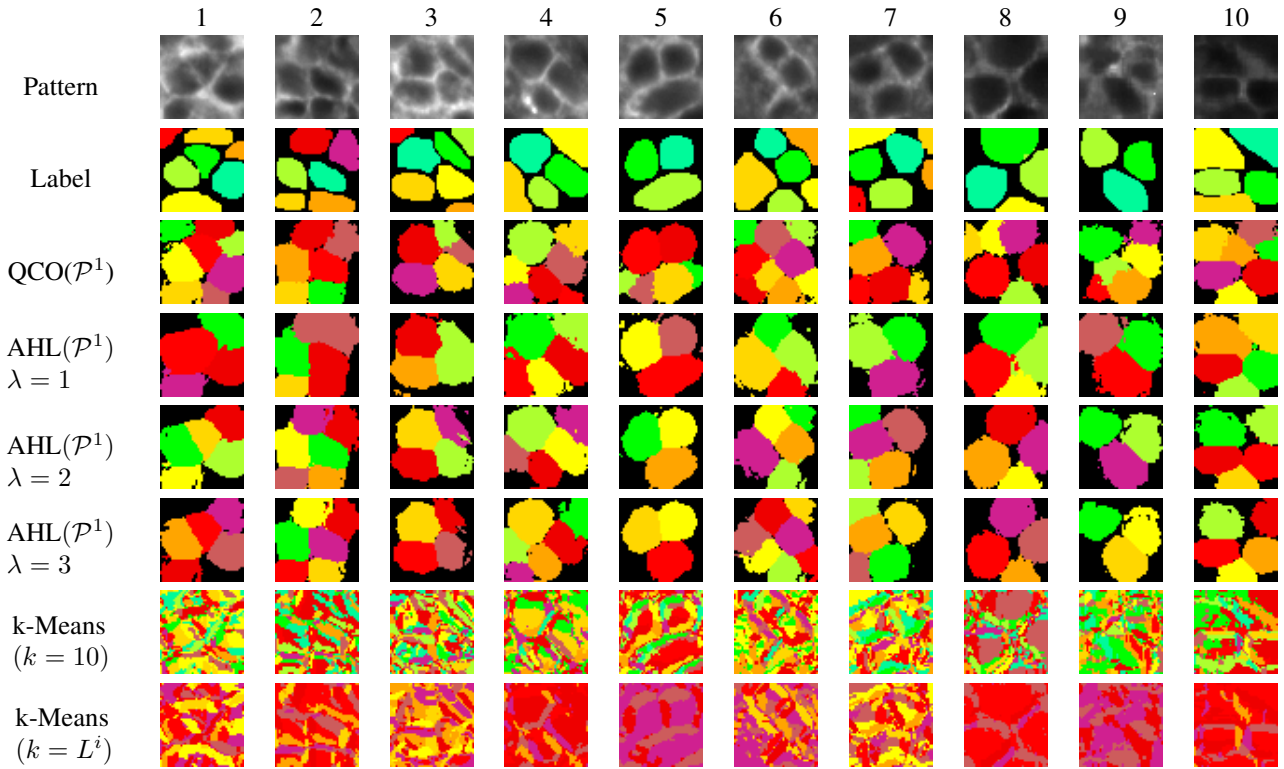


Fig. 18. Fluorescence cell image segmentation: The first two rows show a set of ten training patterns together with their desired labeling. Rows three to six show segmentation results of the learning approaches QCO and AHL from training only with pattern one. The last two rows show results from a direct feature clustering with k-Means-Clustering.

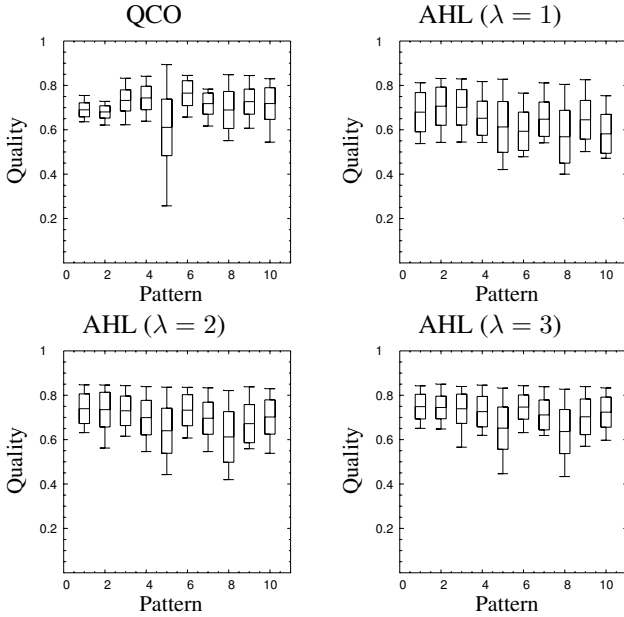


Fig. 19. Statistics of Grouping Quality: The diagrams describe the grouping results of the algorithms QCO and AHL (for different values of λ) on the 10 cell images in Fig. 18. The x-axes enumerate the training pattern, while the y-axes show the grouping quality from the test run over all ten patterns given by average quality, standard deviation of quality and best and worst quality.

Instead we see in the sixth row of figure 18 that at $\lambda = 3$ we reach a parameter region, where oversegmentation starts and the labeled regions become too small.

V. DISCUSSION

The here applied feature binding mechanism of spatial coactivation implemented by the CLM simulation algorithm competes with a huge class of models, like Locally Excitatory Globally Inhibitory Oscillator Networks (LEGION) [1], [25], Multiple-Object Adaptive Grouping of Image Components (MAGIC) [13] and the Spiking Laterally Interconnected Synergetically Self-Organizing MAP (SLISSOM) [3] which are based on the mechanism of temporal correlation.

In these models the feature representative neurons are implemented by oscillatory units, where excitatory lateral connections provide synchronization of neurons in the same group, while inhibitory connections result in a desynchronization of different groups. Though different groups are active at different time phases of the network simulation in these models a single output layer is sufficient to describe the grouping result. However, as it was discussed in the framework of LEGION by Wang & Terman [25] a single layer has only a limited capacity of representing groups at different phase shifts depending on the concrete timing of the inhibitory and excitatory neurons fire rates. The segmentation of patterns with a higher number of groups than the present capacity of the network demands an adaptation of the models parameters. To overcome these limits Wang & Terman suggest a multistage architecture with multiple layers.

In the CLM the capacity for different groups is directly specified by the number of layers. If the number of groups, defined by the lateral interaction weights, exceeds the number of layers, the consistency conditions (3) imply that the groups with the least inter-group inhibition are merged in the same layer. In

this case the capacity of the CLM can be increased by providing additional layers, until layers stay unused.

The competitive connections between the layers of the CLM provide a direct possibility to combine several grouping behaviors in the same network. Applying layer specific lateral interaction matrices with components $f_{rr'}^\alpha, \alpha = 1, \dots, L$ the WTA loops of the network not only implement an assignment of the groups to different layers, but also assign each group to the layer it best fits in. Such an architecture opens new learning problems, like an automatic scaling of the interaction functions with layer weights w^α or the adjustment of layer specific segmentation control parameters λ^α .

If this model is extended further by symmetric inter-layer weights $f_{rr'}^{\alpha\beta}$, expressing the compatibility of the assignment of feature \mathbf{m}_r to label α with the assignment of feature $\mathbf{m}_{r'}$ to label β , it can be compared with the method of relaxation labeling introduced by Rosenfeld et al [21]. For this method it is preferred to use positive weights $f_{rr'}^{\alpha\beta}$ to apply an iterative update for the assignment probabilities $x_{r\alpha}$ under the constraint $\sum_\alpha x_{r\alpha} = 1$, which attracts states of consistent labeling. However, there exist also implementations that can deal with negative compatibilities e.g. [9], which can be trained and interpreted as correlation or mutual information between assignments [18], [19].

Another class of familiar architectures are the Contextual-Constraint Based Hopfield Neural Cube (CCBHNC) [2], the Competitive Hopfield Neural Networks (CHNN) [4] and the Columnar Competitive Model (CCM) [23]. Those networks show the same layer-wise organization of neurons as the CLM with the difference that they consist of binary neurons. Therefore the winner-takes-all-process within the columns of the net is implemented by a simple maximum selection rule, where in a column only the neuron with maximal input stays active such that there is no need for explicit vertical interactions J . Instead the lateral connections, like in the case of relaxation labeling, are extended by a subset of cross layer connections $f_{rr'}^{\alpha\beta}$. In [2], [4] and [23] these connections are chosen strictly negative, based on distances between features.

In contrast to CHNN, CCBHNC and CCM the CLM uses both excitatory and inhibitory lateral connections which allow to change the segmentation level of the output by shifting the interaction weights into the negative or positive quadrant. The linear threshold neurons of the CLM encode real valued network states which provide a richer dynamics than binary recurrent networks, e.g. the self-inhibitory annealing of the simulation algorithm (see appendix I) implements an gradual activation the winner-takes-all behavior within the columns of the CLM to avoid local minima in the CLM energy (2) which correspond to suboptimal grouping states.

In terms of learning the here provided AHL approach shows similarities to the learning approach for the SLISSOM respectively its basis architecture the LISSOM [22]. There in a first stage a self-organizing map adapts the input weights from a feature extracting sensory layer. After the neurons of the network have reached stable firing rates, the lateral interaction weights and the input weights from the sensory layer are adapted by Hebbian learning in form of shifting the weights stepwise towards the correlation of the average firing rates of the connected neurons.

In AHL the SOM-learning step is performed within a pairwise proximity space. The following sampling of positive and negative feature pairs shall approximate the correlation matrix resulting from averaging Hebbian learning steps on a sequence of layer difference vectors specified in the target state of the CLM. The differences to the SLISSOM are that the prototype vectors of the unsupervised SOM learning steps are not necessarily connected to form a feature map and that the input vectors that are adapted by the principle of Hebbian learning are specified in a supervised way from the target labeling.

Surely the construction of suitable proximity functions, as it is demonstrated in this work, faces the same problems like designing heuristics for complete interaction functions. However, these problems occur on a much simpler level, where grouping principles can be described by the underlying distance functions of elementary features, e.g. smoothness and continuity by angles between edges, similarity of color by distances in color spaces and similarity of texture by distance of Gabor jets responses, while the further parameters, like the range and strength of excitatory and inhibitory connections and the weighting of different proximity principles, are extracted from the target labeling.

As the contour grouping examples have shown, the proximity functions can be used to specify generalization properties, like invariance to rotation and translation, as they are demanded in [11] for a robust shape differentiation method. Other demands, like the invariance to scale or the adaptation to highly complex contours, are not so straightforward to achieve from our bottom-up approach which employs only on local features from an early perceptual processing stage.

VI. CONCLUSION

We have shown the successful application of our new approximate Hebbian learning (AHL) approach for the competitive layer model architecture. Apriori knowledge about the desired grouping behavior enters this learning approach basically in two ways. Firstly, an appropriate parameterized feature model has to be selected. Secondly, a number of symmetric distance functions have to be defined for making pairwise comparisons between parameterized feature vectors. These distances are then the basis for actually learning the similarity measure between pairs of features from a labeled training set and incorporate apriori assumptions about translation and rotation invariance of the derived similarity function. This strongly improves the generalization capability of the approach and leads to good segmentation results even from single hand-labeled training images. Differences in the dimensional properties of the set of distance functions are captured by the AEV vector quantization that effectively adapts the resolution of the pairwise similarity function to the different scaling of feature parameters, like orientation angle or spatial distance. We note that this vector quantization method can also be replaced by a simple interval binning of feature parameters, if the feature parameters have a uniform interval distribution on the training set.

Compared to earlier approaches to segmentation and feature binding, where feature compatibility functions have to be designed by careful manual tuning, our hybrid learning approach provides a strong simplification for real practical applications.

This method also enables the learning of appropriate similarity functions in high-dimensional abstract feature spaces, where no intuition on the functional form of the feature similarity is available. Possible further applications, which are, however, beyond the scope of this paper, can be found in general pairwise data clustering problems that need not be related to computer vision or perceptual grouping.

ACKNOWLEDGMENTS

This work was supported by DFG grant GK-231 and carried out at the Faculty of Technology, University of Bielefeld. The authors thank Tim Nattkemper for discussion and Walter Schubert for providing the cell image data.

APPENDIX

I. IMPLEMENTATION OF CLM-DYNAMICS

The CLM-dynamics can be implemented as:

1. Initialize all $x_{r\alpha}$ with small random values around

$$x_{r\alpha}(t=0) \in [h_r/L - \epsilon, h_r/L + \epsilon].$$

Initialize T with the largest eigenvalue of the matrix $\{f_{rr'}\}$.

Choose $J = \max_r \sum_r f_{rr'}$.

2. Do $N \cdot L$ times: Choose (r, α) randomly and update $x_{r\alpha} = \max(0, \xi)$, where

$$\xi := \frac{J(h_r - \sum_{\beta \neq \alpha} x_{r\beta}) + \sum_{r' \neq r} f_{rr'} x_{r'\alpha}}{J - f_{rr} + T}$$

3. Decrease T by $T := \eta T$, with $0 < \eta < 1$.

Go to step 2 until convergence.

The pseudo temperature T controls an additional self-inhibitory term added to (1), see [30]. Starting at $T_c = \lambda_{\max}\{f_{rr'}\}$, which is the maximal eigenvalue of the matrix of lateral weights, an annealing process is implemented by decreasing T to zero. To run the algorithm without annealing simply set $T = 0$.

II. THE QUADRATIC OPTIMIZATION APPROACH(QCO)

The idea of QCO [28] is, that we can directly substitute the definition of the basis functions (13) into the consistency conditions (5) to get a set of dimension reduced consistency conditions of the form:

$$\sum_j c_j Z_j^k < 0 \quad \text{for all } k = (i, r \mid \mathbf{m}_r \in \mathcal{R}^i, \beta \neq \alpha^i(r)) \quad (28)$$

where k is a super-index for the consistency conditions running over all combinations of pattern i , feature r , and label $\beta \neq \alpha^i(r)$. The values Z_j^k describe the information from the training set and the basis interactions by a projection of the target layer vectors $(\mathbf{y}_{\hat{\alpha}} - \mathbf{y}_{\beta})$ onto the basis functions $g_{rr'}^j$:

$$Z_j^k = Z_j^{ir\beta} = \sum_{r' \mid \alpha^i(r') = \beta} g_{rr'}^j - \sum_{r' \mid \alpha^i(r') = \alpha^i(r)} g_{rr'}^j. \quad (29)$$

If we write the values Z_j^k as vectors $\mathbf{Z}^k = (Z_1^k, \dots, Z_K^k)$, each vector \mathbf{Z}^k corresponds exactly with one consistency condition in (5), we therefore call them consistency vectors.

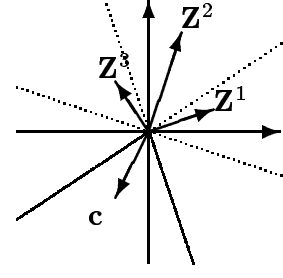


Fig. 20. Geometry Representation of the Learning Problem of QCO: the consistency vectors \mathbf{Z}^k describe the learning problem, to fulfill the k th consistency condition, we have to choose the vector of learning parameters \mathbf{c} from the opposite half space of the weight state space. Therefore each new consistency condition restricts the area of suitable interaction weights further (thick lines).

If we consider these vectors in the weight space of learning parameters c_j , we get a simple geometrical interpretation of the learning problem (see Figure 20). To fulfill the k th consistency condition, we have to choose $\mathbf{c} = (c_1, \dots, c_K)^T$ from the opposite half space of \mathbf{Z}^k . Therefore each new consistency condition restricts the area of suitable interaction weights \mathbf{c} . In order to achieve a better generalization to further yet unseen patterns, we can restrict this area further by the introduction of a positive margin variable $\kappa > 0$ in (28).

$$\sum_j c_j Z_j^k + \kappa < 0 \quad \text{for all } k = (i, r \mid \mathbf{m}_r \in \mathcal{R}^i, \beta \neq \alpha^i(r)). \quad (30)$$

In [26], [28], the optimization problem, to satisfy (30), is solved by minimizing the quadratic error

$$E_{QCO} = \sum_k \left(\sum_j c_j Z_j^k + \kappa \right)^2. \quad (31)$$

This approach is more restrictive than the original learning problem (30), because it demands that all consistency conditions are fulfilled in the same manner. Applications of a similar approach for designing BSB associative memories have shown that it is competitive to more sophisticated optimization methods [17]. The minimum of (31) is found by gradient descent under additional constraints $|c_j| \leq 1$. These constraints prevent the interaction coefficients from obtaining large values, which might disturb the group formation process, e.g. if one basis function describes a pure self interaction $g_{rr'}^i = \delta_{rr'}$ it follows that all Z_i^k are equal to one such that the minimum of (31) can be found trivially by setting c_i to κ and all other parameters $c_j, j \neq i$ to zero. The result would be $f_{rr'} = \kappa \delta_{rr'}$ which obviously can not result in reasonable grouping since all lateral interactions are set to zero. However, if $|c_j| \leq 1$ this trivial solution is not allowed for $\kappa > 1$ such that all c_j have to be adapted to find a minimum of (31).

The QCO-approach is computationally expensive because all consistency conditions (5) have to be transformed by equations (29) and (30) to the space of learning parameters c_j , before the optimization step according to the error function (31) is performed.

III. EXAMPLE FOR THE CONSTRUCTION OF THE INTERACTION MATRIX

Assume a set of six arbitrary features $\mathbf{m}_r, r = 1, \dots, 6$ that have to be separated into three groups containing three, two, and one features. We describe this problem by assigning each feature to a label $\alpha(r) \in 1, 2, 3$, where we assume, that the features are ordered according to their label, such that we get the labeling vector:

$$\alpha^{ord} = [1 \ 1 \ 1 \ 2 \ 2 \ 3]^T. \quad (32)$$

Using this labeling in (4), we generate a three layered target state of the CLM, containing the layer vectors:

$$\begin{aligned} \mathbf{y}_1^{ord} &= [1 \ 1 \ 1 \ 0 \ 0 \ 0]^T \\ \mathbf{y}_2^{ord} &= [0 \ 0 \ 0 \ 1 \ 1 \ 0]^T \\ \mathbf{y}_3^{ord} &= [0 \ 0 \ 0 \ 0 \ 0 \ 1]^T. \end{aligned} \quad (33)$$

Now compute the block-diagonal correlation matrix for the difference vectors between these layer vectors according to (7) as

$$\hat{F} = \begin{bmatrix} 4 & 4 & 4 & -2 & -2 & -2 \\ 4 & 4 & 4 & -2 & -2 & -2 \\ 4 & 4 & 4 & -2 & -2 & -2 \\ -2 & -2 & -2 & 4 & 4 & -2 \\ -2 & -2 & -2 & 4 & 4 & -2 \\ -2 & -2 & -2 & -2 & -2 & 4 \end{bmatrix}. \quad (34)$$

The negative entries of \hat{F} are scaled against the positive ones by the factor $\lambda = \frac{1}{3-1}$. For any other ordering of the six features, we simply have to permute the rows and the columns of (34) in the same way as the features to make the interaction matrix consistent with the new labeling according to (5).

IV. AHL LEARNING ALGORITHM

We are given a set of training patterns \mathcal{P}^i , where each training pattern consists of a set of N^i features \mathbf{m}_r^i , with a target labeling $\alpha^i(r) = \alpha^i(\mathbf{m}_r^i)$. The distances in the proximity space are computed by a set of P similarity functions $d_p(\mathbf{m}_r, \mathbf{m}_{r'})$, $p = 1, \dots, P$.

We first compute the individual mean μ_p and variance σ_p for each distance function on the training set:

$$\begin{aligned} \mu_p &= \frac{1}{\sum_i N_i^2} \sum_i \sum_{r, r'} d_p(\mathbf{m}_r^i, \mathbf{m}_{r'}^i), \\ \sigma_p &= \sqrt{\frac{1}{\sum_i N_i^2} \sum_i \sum_{r, r'} (d_p(\mathbf{m}_r^i, \mathbf{m}_{r'}^i) - \mu_p)^2}. \end{aligned}$$

The distances are then normalized by their variance using

$$a_p = \begin{cases} \frac{1}{\sigma_p} & : \sigma_p > 0 \\ 1 & : \sigma_p = 0 \end{cases},$$

and we define $D_p^{irr'} = a_p d_p(\mathbf{m}_r^i, \mathbf{m}_{r'}^i)$ as the normalized distance for all p, i, r, r' . The set of distance vectors $\mathbf{D}^{irr'} = (D_1^{irr'}, \dots, D_P^{irr'})^T$ is quantized using a set of K prototypes

$\tilde{\mathbf{d}}_j \in \mathbb{R}^P, j = 1, \dots, K$. The Voronoi cell set V_j of prototype $\tilde{\mathbf{d}}_j$ is defined as

$$V_j = \{(i, r, r') \mid \|\mathbf{D}^{irr'} - \tilde{\mathbf{d}}_j\| < \|\mathbf{D}^{irr'} - \tilde{\mathbf{d}}_k\| \text{ for all } k \neq j\},$$

which carries the training pairs assigned to this prototype. This can be further subdivided into a set V_j^+ , where features are assigned to the same layer, and V_j^- , where they are in different layers¹:

$$\begin{aligned} V_j^+ &= \{i, r, r' \in V_j \mid \alpha^i(r) = \alpha^i(r')\}, \\ V_j^- &= \{i, r, r' \in V_j \mid \alpha^i(r) \neq \alpha^i(r')\}. \end{aligned}$$

The algorithm can be performed using simple vector quantization or the more advanced AEV algorithm for resetting unused prototype vectors. For AEV, an activation score A_j is computed for each prototype, based on the population of the Voronoi cell set V_j . Based on an activation threshold Θ_{AEV} , and a given update probability P_{AEV} prototypes are then reinitialized.

The target of the learning are the coefficients c_j^+ and c_j^- of the basis interaction prototypes, derived from the vector quantization in the proximity space.

The AHL learning algorithm is implemented by:

Initialize $(\tilde{\mathbf{d}}_j)_p(t=0) = \mathcal{N}(a_p \mu_p, 1)$ ²;
For iterations $i = 0, \dots, N$
perform step 1 to 3:

1. For all $j \in 1, \dots, K$:
set $\tilde{\mathbf{d}}_j(t+1) = \frac{1}{A_j(t)} \sum_{i, r, r' \in V_j(t)} \mathbf{D}^{irr'}(t)$,
where $A_j(t) = \sum_{i, r, r' \in V_j(t)} 1$;
2. For all $j \in 1, \dots, K$:
compute $V_j(t+1)$, based on
the new prototypes $\tilde{\mathbf{d}}_j(t+1)$;
3. AEV step with $P_{AEV} = 1 - \frac{i}{N}$:
For all $j \in 1, \dots, K$ with $\frac{A_j}{\sum_k A_k} < \Theta_{AEV}$:
choose $s \in [0, 1]$ randomly;
if $s < P_{AEV}(t)$:
set $(\tilde{\mathbf{d}}_j)_p \in \mathcal{N}(a_p \mu_p, 1)$;
otherwise:
choose another prototype $\tilde{\mathbf{d}}_n$;
place $\tilde{\mathbf{d}}_j$ near $\tilde{\mathbf{d}}_n$:
 $(\tilde{\mathbf{d}}_j)_p \in \mathcal{N}((\tilde{\mathbf{d}}_n)_p, 0.01)$;

For all $j \in 1, \dots, K$:
compute:
 $c_j^+ = \sum_{i, r, r' \in V_j^+} 1$ and $c_j^- = \sum_{i, r, r' \in V_j^-} 1$;
For all $j \in 1, \dots, K$:
normalize:
 $c_j^+ := c_j^+ / \sum_{i=1}^K c_i^+$ and $c_j^- := c_j^- / \sum_{i=1}^K c_i^-$;

The AEV step is omitted by removing step 3 from the algorithm. For all the simulations in this contribution we choose

¹Feature-pairs, where both features are assigned to the background (like in Fig. 18) or one feature is not assigned to any label, should be treated as assigned to different labels to prevent spurious support in the the background.

² $\mathcal{N}(\mu, \sigma)$ is a normal distribution with mean μ and variance σ .

$\Theta_{AEV} = \frac{1}{2K}$ and perform eleven iterations of step 1 to 5 ($N = 10$). Finally, we note that all summations over (i, r, r') that run over all feature pairs from the training patterns, can be approximated by a randomly selected subset. In all presented simulations the size of this subset is 10000 feature pairs.

If we want to apply the learned interaction to a new test pattern \mathcal{P}^T with N^T features $\mathbf{m}_r, r = 1, \dots, N^T$, we compute the lateral interactions as:

$$f_{rr'} = f(\mathbf{m}_r, \mathbf{m}_{r'}) = c_{I(\mathbf{m}_r, \mathbf{m}_{r'})}^+ - \lambda c_{I(\mathbf{m}_r, \mathbf{m}_{r'})}^-,$$

where $I(\mathbf{m}_r, \mathbf{m}_{r'})$ is the index of the corresponding proximity prototype

$$I(\mathbf{m}_r, \mathbf{m}_{r'}) = \operatorname{argmin}_j \left(\sum_p (a_p d_p(\mathbf{m}_r, \mathbf{m}_{r'}) - (\tilde{\mathbf{d}}_j)_p)^2 \right).$$

The separation strength λ can be chosen according to the discussion in section III-E.

V. QUALITY MEASURE

We measure the grouping quality Q of the CLM for a hand-labeled test pattern \mathcal{P}^T by comparing the goal labeling $\alpha^G(r) \in \{1, \dots, L^G\}$ with the labeling from the CLM-output $\alpha^C(r) \in \{1, \dots, L^C\}$. Therefore we initialize the components of the $L^G \times L^C$ Matrix $O^{(0)}$ with the number of features that are assigned in the goal-labeling to α^G and in the CLM-answer to α^C . Starting from $Q = 0$, we assume, that the maximum entry of O , which is the largest overlap between a group in the goal labeling and the CLM labeling, is labeled correctly, and add it to Q . Since this group must be separated from all other groups, the features in the same column and row of this entry must be labeled incorrectly. We delete all these values from O and seek iteratively the next largest overlap between the goal labeling and the CLM labeling in the remaining submatrix of O , until all entries of O are equal to zero. Finally, we normalize Q with the total number of features in the pattern such that Q lies within the interval $[0, 1]$ and we can interpret it as percentage of overlap between goal labeling and CLM-labeling. The quality measure is implemented in the following way:

Initialize

$$Q := 0$$

$$O_{\alpha^G \alpha^C}^{(0)} = \#\{\mathbf{m}_r \mid \alpha^G(r) = \alpha^G \wedge \alpha^C(r) = \alpha^C\}.$$

for $(t = 0; t < L^G)$

$$(\alpha_{MAX}^G, \alpha_{MAX}^C) = \operatorname{argmax}_{(\alpha^G, \alpha^C)} O_{\alpha^G \alpha^C}^{(t)}$$

$$Q := Q + O_{\alpha_{MAX}^G \alpha_{MAX}^C}^{(t)}$$

for all (α^G, α^C)

$$O_{\alpha^G, \alpha^C}^{(t+1)} = \begin{cases} O_{\alpha^G, \alpha^C}^{(t)} & : \alpha^G \neq \alpha_{MAX}^G \wedge \alpha^C \neq \alpha_{MAX}^C \\ 0 & : \text{else} \end{cases}$$

$$Q := \frac{Q}{\#\mathbf{m}_r}$$

REFERENCES

- [1] E. Cesmeli, D. Wang: Texture Segmentation Using Gaussian-Markov Random Fields and Neural Oscillator Networks. IEEE Trans. on Neural Networks 12(2) (2001), pp. 394-404.
- [2] C.Y. Chang, P. C. Chung: Medical Image Segmentation Using a Contextual-Constraint-Based Hopfield Neural Cube. Image and Vision Computing 19 (2001), pp. 669-678.
- [3] Y. Cho, R. Miikkulainen: Self-Organization with Laterally Connected Spiking Neurons. Proc. of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97, Nagoya Japan) (1997), pp. 1120-1125.
- [4] P.C. Chung, C.T. Tsai, E.L. Chen, Y.N. Sun: Polygonal Approximation Using a Competitive Hopfield Neural Network. Pattern Recognition Vol. 27(11) (1994), pp. 1505-1512.
- [5] R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, H. S. Seung: Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature 405 (2000), pp. 947-951.
- [6] G. Heidemann, H. Ritter: Efficient Vector Quantization using the WTA-rule with Activity Equalization. Neural Processing Letters 13(1) (2001), pp. 17-30.
- [7] T. Hofmann, J. Puzicha, J. Buhmann: Unsupervised texture segmentation in a deterministic annealing framework. IEEE Trans. Pattern Analysis and Machine Intelligence 20(8) (1998), pp. 803-818.
- [8] J.J. Hopfield: Neural Networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. USA 79 (1982), pp. 2554-2558.
- [9] R. A. Hummel, S. W. Zucker: On the Foundations of Relaxation Labeling Processes. IEEE Trans. PAMI 5(3) (1983), pp. 267-287.
- [10] J. Kittler, J. Illingworth: Relaxation Labeling Algorithms - A Review. Image Vision Comput. 3 (1985), pp. 206-216.
- [11] L. J. Latecki, R. Lakämper: Application of planar shape comparison to object retrieval in image databases. Pattern Recognition 35 (2002), pp. 15-29.
- [12] K. M. Lee, W. N. Street: An Adaptive Resource-Allocating Network for Automated Detection, Segmentation, and Classification of Breast Cancer Nuclei Topic Area: Image Processing and Recognition. IEEE Trans. on Neural Networks 14(3) (2003), pp. 680-687.
- [13] M. Mozer, R. S. Zemel, M. Behrmann, C. K. I. Williams: Learning to segment images using dynamic feature binding. Neural Computation 4(5) (1992), pp. 650-665.
- [14] T. W. Nattkemper, H. Ritter, W. Schubert: A Neural Classifier Enabling High-Throughput Topological Analysis of Lymphocytes in Tissue Sections. IEEE Trans. on Inf. Techn. in Biomed. vol. 5, no. 2 (2001), pp. 138-149.
- [15] T. W. Nattkemper, H. Wersing, H. Ritter, W. Schubert: A Neural Network Architecture for Automatic Segmentation of Fluorescence Micrographs. Neurocomputing, 48, (4), (2002), pp 357-367.
- [16] J. Ontrup, H. Wersing, H. Ritter: A Computational Feature Binding Model of Human Texture Perception. Cognitive Processing., 5(1), (2004), pp. 32-44.
- [17] J. Park, H. Cho, D. Park: On the design of BSB associative memories using semidefinite programming. Neural Computation 11 (1999), pp. 1985-1994.
- [18] S. Peleg, A. Rosenfeld: Determining compatibility coefficients for curve enhancement relaxation processes. IEEE Trans. Syst. Man Cybern. 8 (1978), pp. 548-555.
- [19] M. Pelillo, M. Refices: Learning compatibility coefficients for relaxation labeling processes. IEEE Trans. PAMI 16(9) (1994), pp. 933-945.
- [20] H. Ritter: A spatial approach to feature linking. In Proc. International Neural Network Conference Paris Vol.2 (1990), pp. 898-901.
- [21] A. Rosenfeld, R.A. Hummel, S.W. Zucker: Scene Labeling by Relaxation Operations. IEEE Trans. Syst. Man Cybern. 6(6) (1976), pp. 420-433.
- [22] J. Sirosh, R. Miikkulainen: Topographic Receptive Fields and Patterned Lateral Interaction in a Self-Organizing Model of the Primary Visual Cortex. Neural Computation 9 (1997), pp. 577-594.
- [23] H. Tang, K. C. Tan, Z. Yi: A Columnar Competitive Model for Solving Combinatorial Optimization Problems. IEEE Trans. on Neural Networks 15(6) (2004), pp. 1568-1573.
- [24] C. von der Malsburg: The what and why of binding: The modeler's perspective. Neuron 24 (1999), pp. 95-104.
- [25] D. Wang, D. Terman: Image segmentation based on oscillatory correlation. Neural Computation 9(4) (1997), pp. 805-836.
- [26] S. Weng, J. J. Steil: Data Driven Generation of Interactions for Feature Binding. In Proc. ICANN (2002), pp. 432-437.
- [27] S. Weng, J. J. Steil: Learning Compatibility Functions for Feature Binding and Sensory Segmentation. In Proc. ICANN (2003), pp. 60-67.
- [28] H. Wersing: Learning Lateral Interactions for Feature Binding and Sensory Segmentation. Advances in Neural Information Processing Systems NIPS Vol. 14(2001), pp. 1009-1016.
- [29] H. Wersing, W.-J. Beyn, H. Ritter: Dynamical stability conditions for recurrent neural networks with unsaturating piecewise linear transfer functions. Neural Computation 13(8) (2001), pp. 1811-1825.
- [30] H. Wersing, J. J. Steil, H. Ritter: A competitive layer model for feature binding and sensory segmentation. Neural Computation 13(2) (2001), pp. 357-387.

- [31] X. Xie, R. Hahnloser, H. Seung: Learning winner-takes-all competition between groups of neurons in lateral inhibition networks. In Advances in Neural Information Processing Systems 13, The MIT Press (2001).
- [32] Z. Yi, K. K. Tan: Multistability of Discrete-Time Recurrent Neural Networks with Unsaturating Piecewise Linear Activation Functions. IEEE Trans. on Neural Networks 15(2) (2004), pp. 329-336.



Sebastian Weng received the Diploma in Computer Science from the University of Bielefeld, in 2001. He currently pursuing a Ph.D. program in Computer Science at the University of Bielefeld, working within the Neuroinformatics Group and the graduate program "Strukturbildungsprozesse" ("Structure forming phenomena"). He actually researches learning algorithms for the Competitive Layer Model (CLM) a recurrent neural network that is able to fulfill grouping and segmentation tasks on the basis of various Gestalt laws. Main application fields are pattern recognition and

computer vision.



Heiko Wersing received the Diploma in Physics in 1996 from the University of Bielefeld, Germany. In 2000 he received the Ph.D. in science from the Faculty of Technology at the University of Bielefeld, Germany. In 2000 he became a member of the Future Technology Research Group of Honda R&D Europe GmbH, Offenbach, Germany. Since 2003 he holds a position as a principal scientist in the Honda Research Institute Europe GmbH at Offenbach, Germany. His current research interests include recurrent neural networks, models of perceptual grouping and feature

binding, principles of sparse coding, and biologically motivated approaches to object recognition.



Jochen J. Steil received the Diploma in Mathematics from the University of Bielefeld, Germany, in 1993. Since then he has been a member of the Neuroinformatics Group at the University of Bielefeld working mainly on recurrent networks. In 1995/96 he stayed one year at the St. Petersburg Electrotechnical University, Russia, supported by a German Academic Exchange Foundation (DAAD) grant. In 1999, he received the Ph.D. Degree with a Dissertation on "Input-Output Stability of Recurrent Neural Networks", since 2002 he has been appointed tenured senior researcher

and teacher (Akad. Rat). J.J. Steil is staff member of the special research unit 360 "Situational Artificial Communicators" and the Graduate Program "Task Oriented Communication" heading projects on robot learning and intelligent systems. Main research interests of J.J. Steil are analysis, stability, and control of recurrent dynamics and recurrent learning, and cognitive oriented architectures of complex robots for multimodal communication and grasping. He is member of the ENNS and the IEEE neural networks society.



Helge Ritter studied Physics and Mathematics at the Universities of Bayreuth, Heidelberg and Munich. After a Ph.D. in Physics at Technical University of Munich in 1988 he visited the Laboratory of Computer Science at Helsinki University of Technology and the Beckman Institute for Advanced Science and Technology at the University of Illinois at Urbana-Champaign. Since 1990 he is head of the Neuroinformatics Group at the Faculty of Technology, Bielefeld University. His main interests are principles of neural computation, in particular for self-organization and learning,

and their application to build intelligent systems. In 1999, Helge Ritter was awarded the SEL Alcatel Research Prize and in 2001 the Leibniz Prize of the German Research Foundation DFG.