

# Unsupervised Segmentation With Dynamical Units

A. Ravishankar Rao, *Senior Member, IEEE*, Guillermo A. Cecchi, Charles C. Peck, and James R. Kozloski

**Abstract**—In this paper, we present a novel network to separate mixtures of inputs that have been previously learned. A significant capability of the network is that it segments the components of each input object that most contribute to its classification. The network consists of amplitude-phase units that can synchronize their dynamics, so that separation is determined by the amplitude of units in an output layer, and segmentation by phase similarity between input and output layer units. Learning is unsupervised and based on a Hebbian update, and the architecture is very simple. Moreover, efficient segmentation can be achieved even when there is considerable superposition of the inputs. The network dynamics are derived from an objective function that rewards sparse coding in the generalized amplitude-phase variables. We argue that this objective function can provide a possible formal interpretation of the binding problem and that the implementation of the network architecture and dynamics is biologically plausible.

**Index Terms**—Binding problem, deconvolution, oscillations, phase correlation, separation of mixtures, synchronization.

## I. INTRODUCTION

THE binding problem [1] is a long-standing issue in the field of neural computation [2]. One formulation of the problem, which can be traced back to Rosenblatt, states that neural networks (NNs) do not have the capacity to encode superimposed inputs (i.e., there is a superposition catastrophe [3]). The essence of the binding problem is that relationships that exist between features of an object at a given level of abstraction may be lost when the features are distributed across a network at multiple levels of abstraction. For instance, consider the human visual system (HVS) looking at four edges that make a square. The HVS is able to identify the edges, a low-level representation, with a square, a high-level representation. Both the high- and low-level representations are bound together by the concept “square.” One way of tackling this issue in an NN architecture is to employ a variable independent from the amplitude, which can provide additional information about the state of the units in the network. An example of such an independent variable is the phase of ongoing oscillations within elements of the network. Wang [2] has argued that the time domain is essential in overcoming Rosenblatt’s superposition catastrophe. Our work utilizes oscillatory phase, which is one mechanism for abstracting information in the time domain.

Let us consider the development of the HVS, and identify a few of the key problems it must solve. First, it must be able to form the high-level object categories in an unsupervised manner

[4]. The learning paradigm that can be used in this case is one where the system is repeatedly presented with instances from a collection of objects, and it learns to form distinct higher level categorizations of these objects. A second problem that the HVS must solve is that given a visual scene consisting of superposed objects, e.g., a cup on top of a table, it must identify the high-level object categories in the scene. Thus, an input scene consisting of “cup on a table” is recognized as being composed of two objects, “cup” and “table” that have been previously presented to the system and learned, and not as an unknown object. This problem may be viewed as one of separating a mixture of inputs into its constituents. A third problem is that the HVS must be able to identify the lower level features (e.g., image pixels or edges) that correspond to the higher level objects that they are part of. Thus, this would enable the handle of the cup, a lower level feature, to be identified as a feature belonging to the higher level category “cup.” This process may be viewed as one of segmentation, whereby high-level percepts and supporting low-level features are grouped. The significance of segmentation from a biological perspective is that it allows the identification of high-level percepts to be followed by appropriate action, such as lifting of the cup, which requires information about its location as provided by the lower level features.

Though there are other problems the HVS must address, e.g., the processing of color and motion, the progress in addressing these problems has been very limited [2]. In this paper, we investigate a solution to these problems that the HVS must solve, which consist of an unsupervised learning method to separate and segment mixtures of inputs. The separation of mixtures and blind deconvolution (i.e., identifying the presence of specific objects in the visual field) have been extensively studied in the NN literature [5]. The problem we investigate in this paper goes beyond separation however, and involves the additional step of segmentation. By segmentation, we refer to the ability to identify the elements of the input space that uniquely contribute to each specific object (i.e., establishing a correspondence between the pixels or edges and the higher level objects they belong to).

Typically, this segmentation problem has been attacked with nonneural approaches [6]. It is also possible to consider an approach where one performs an exhaustive trace back, i.e., examines each one of the inputs to a high-level category that has been activated, and determines where the support for the high-level activity originated. This approach is computationally expensive, especially, for multilayer networks, and there is little biological evidence to support it. Hence, we are still faced with the problem of determining an appropriate segmentation mechanism that has a basis in neural computation.

A promising research direction to pursue has been provided by the analysis of temporal domain phenomena. Inspired by experimental evidence of a role for synchronization of neural responses in a variety of motor and cognitive tasks, and, in partic-

Manuscript received December 24, 2005; revised October 27, 2006 and May 11, 2007; accepted May 13, 2007.

The authors are with the T. J. Watson IBM Research Center, Yorktown Heights, NY 10598 USA (e-mail: ravirao@us.ibm.com; gcecchi@us.ibm.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNN.2007.905852

ular, in perceptual recognition [7]–[9], Malsburg and Shneider were among the first to propose the use of synchronization to perform segmentation of a mixture of signals [10]. Their model consists of a layer of excitatory units connected with lateral excitation. Each of these excitatory units receives sensory input. Furthermore, every excitatory unit is connected to a global inhibitory unit which receives excitatory inputs, and sends inhibitory signals to each of the excitatory units. Segmentation is exhibited in the form of temporal correlation among the activities of the different excitatory units, so that the units that are synchronized represent the same input class. Some of the limitations of this approach include the need for a global inhibitory unit, and the inability of this network to disambiguate objects with partial overlap. Indeed, a number of approaches derived from [10] inherit the same shortcomings [11]–[13], and therefore, the issue of effective segmentation by networks of synchronizing units still needs to be addressed. In subsequent sections, we will introduce a novel network architecture that can efficiently segment superposed inputs, and can potentially be generalized to higher dimensions. Segmentation can also be considered a solution to the binding problem [1], an issue extensively discussed in the neuroscience literature.

The main contribution of this paper is to use an optimization approach to state the desired behavior of a network of oscillatory units. The network dynamics are derived in a principled manner by using an objective function that rewards sparse encoding of an input space. We show that the network dynamics can be implemented in a simple network with feedforward, lateral, and feedback connections. The network is able to separate and segment mixtures of inputs that have been learned in an unsupervised manner, and is able to cope with a considerable degree of spatial overlap of the inputs, in contrast with similar previous schemes.

## II. BACKGROUND

The original network proposed by Malsburg and Schneider [10], [11] has been influential in formulating a theory for the use of synchrony as a solution to segmentation. However, the specific implementation proposed in their paper has several shortcomings. First, a global inhibitory neuron is required. Our model overcomes this restriction and spreads inhibition across the entire network, which is more biologically plausible. Second, learning in their model requires a combination of short- and long-term synaptic modification, which in our model is reduced to a single generic rule. Third, the test cases used in their model did not involve any overlap among the spectral inputs to be separated. Our model allows complete overlap and shows that successful separation and segmentation is still possible.

Buhman and Malsburg [11] explicitly introduced oscillatory units into the model, but their model suffers from earlier noted shortcoming in that the presence of a global inhibitory unit is required. The subsequent works of Chen *et al.* [13] and Wang and Liu [13] offer enhancements of the original model, but maintain the essential aspect of utilizing a global inhibitor.

The work of Izhikevich [14] is mainly theoretical, and does not present any specific methodology to address the problem

of segmentation. Hoppensteadt and Izhikevich [15] illustrate their method with a single example using three inputs and have not applied their methodology to a larger number of inputs or test cases, or addressed the segmentation problem. Furthermore, they raise the issue that the Hebbian learning rule they use may not be the best. In contrast, our formulation uses the Hebbian rule, which is simple, and we have shown that it works extremely well. The method of Sun *et al.* [16] requires the use of visual motion to perform segmentation, and hence is not applicable to static inputs as we have investigated. Furthermore, their scheme relies on supervised training and uses backpropagation learning.

Zemel *et al.* [17] used a similar approach in which units in their network possess both amplitude and phase. Their learning algorithm was derived through trial and error, rather than an explicit optimization approach as we will show. Their approach requires supervised training, as they set the target phases of different contours in the training images to specific values. Our approach does not require the setting of target phase values during training. Furthermore, their network makes explicit use of complex weights, rendering biological interpretations more abstruse.

Chen *et al.* [18] describe an oscillatory model to perform image segmentation. However, their model does not segment superposed objects, and thereby does not address the issue of Rosenblatt’s superposition catastrophe. Cosp and Madrenas [19] describe a hardware implementation of the method in Chen *et al.*, which is capable of 2-D image segmentation but not the separation of superposed inputs.

More recently, Eckhorn *et al.* [20] have shown that both short- and long-range synchronizations aid the associative processing of signals in the cortex.

Lee [21] observes that most of the existing models for oscillatory NNs derive from the Hodgkin–Huxley, FitzHugh–Nagumo, and Wilson–Cowan models, and are “either too simplified to simulate any “real” chaotic neural behaviors or too complicated to be applied as feasible artificial NNs for applications.” In other words, there is an issue with modeling the neural behavior at the right level of abstraction, such that we capture the temporal dynamics of the neurons and use that to demonstrate significant useful behaviors such as object classification and discrimination between components of multiple objects. Indeed, there are few attempts in the literature which have succeeded at this goal and we have explicitly tried to address this issue in this paper.

There may be many mechanisms that are at play in solving the binding problem. Van der Velde and de Kamps [22] discuss the role that an object-based attention mechanism plays through top–down activation from higher to lower level cortical areas. Based on this mechanism, they are able to explain the occurrence of binding between the visual features of form and position. Furthermore, neural synchronization may also help bind nonvisual entities in general in the brain, including semantic and lexical concepts [22]. A recent paper by Weng *et al.* [23] explores the combination of supervised and unsupervised Hebbian learning in understanding the phenomenon of feature binding.

## III. SEGMENTATION

In this section, we address the problem of designing a system based on neural computation which learns in an unsupervised

manner and achieves the separation and segmentation of mixtures. We start with a two-layer system, consisting of a lower layer that captures the input and an upper layer that is responsible for a higher level categorization of the input. The input consists of a 2-D representation of visual objects, encoded as gray-level images. The input layer is connected via feedforward weights to the output layer and feedback weights connect the output layer to the input layer. There are also lateral connections between the units in the upper layer.

One viewpoint to describe the function of the output layer in such a system is that it performs a sparse encoding of the input space. This approach has been used by Olshausen and Fields [24] to understand the role of the cortex. Sparse encoding can also be viewed as a form of vector quantization. The function of sparse representation may be captured by an objective function designed as follows. Let the inputs  $\mathbf{x}$  be drawn from an input ensemble. Let an output layer  $\mathbf{y}$  represent these inputs through synaptic weights  $\{W_{ij}\}$ . For notational convenience, let  $\mathbf{x}$  and  $\mathbf{y}$  be row vectors. Further, impose a nonnegativity condition on the output layer  $y_i \geq 0 \forall i$ . The objective function  $E$  is defined as

$$E = \left\langle \mathbf{y} \mathbf{W} \mathbf{x}^T - \frac{1}{2} \mathbf{y}^2 - \frac{1}{2} \sum_n \mathbf{W}_n^2 + \frac{1}{2} \lambda \mathcal{S}(\mathbf{y}) \right\rangle_{\mathcal{E}} \quad (1)$$

where  $\mathcal{E}$  represents the input ensemble. The first term captures the faithfulness of representation and rewards the alignment between the network's output and the feedforward input. Note that the output  $\mathbf{y} \neq \mathbf{W} \mathbf{x}^T$  due to the presence of lateral and feedback interactions, and the use of a squashing function to make  $\mathbf{y}$  nonnegative. A stable value of  $\mathbf{y}$  is obtained only after the network settles as will be demonstrated in Figs. 4 and 5.

The second term is a constraint on the global activity and the third term is derived from imposing normalization of the synaptic weight vectors. The last term is defined as

$$\mathcal{S}(\mathbf{y}) = N (\langle y_n^2 \rangle_{\mathcal{N}} - \langle y_n \rangle_{\mathcal{N}}^2) = \sum_{n=1}^N y_n^2 - \frac{1}{N} \left( \sum_{n=1}^N y_n \right)^2 \quad (2)$$

where  $\mathcal{N}$  represents the network consisting of  $N$  units. This term is the variance of the output  $y$ , which is high when the distribution of output values is skewed, i.e., sparse. Given the imposition of nonnegativity of  $y_i$ , this term can be considered to reward the sparseness of the representation. By imposing normalization on the synaptic weights and whitening of the inputs, the objective function can be simplified as

$$E = \left\langle \mathbf{y} \mathbf{W} \mathbf{x}^T + \frac{1}{2} \lambda \mathcal{S}(\mathbf{y}) - \frac{1}{2} \mathbf{y}^2 \right\rangle_{\mathcal{E}} \quad (3)$$

assuming that synaptic normalization is enforced during the maximization process, as discussed in Appendix I. Applying gradient ascent to the objective function with respect to (wrt)  $\mathbf{y}$ , one obtains the dynamics that maximizes it upon presentation of each input, and applying it wrt  $\mathbf{W}$  one obtains the optimal learning update. Within an appropriate parameter range, learning leads to a winner-take-all (WTA) dynamics upon presentation of one of the learned inputs; moreover, when two learned inputs are presented, two winners arise, as depicted in Fig. 4. This issue is investigated in further detail in Appendix II.

In the previous formulation, each unit in the network is represented by a scalar value, say, by an amplitude. If we allow the units in the network to be oscillatory, each unit is now represented by an amplitude, frequency, and phase of oscillation. If the frequencies of all the units are close together, we can effectively describe each unit in terms of phasors of the form  $x_n e^{i\phi_n}$  for the lower layer and  $y_n e^{i\theta_n}$  for the upper layer. Here,  $\phi_n$  and  $\theta_n$  are the phases of the  $n$ th unit in the lower and upper layers, respectively. Phasors are a convenient way to represent the activity of units in an oscillatory network and will be used henceforth.

We will now introduce a generalization of the objective function, denoted by  $E_s$ , which is based on the behavior of these oscillatory units. We show that the maximization of  $E_s$  leads to an efficient segmentation of the inputs. For notational convenience, we define the input phasors to be  $p_n = x_n e^{i\phi_n}$  and the output phasors to be  $q_n = y_n e^{i\theta_n}$ . Let  $\mathcal{C}(E)$  be a complex version of the objective function  $E$  of (1)

$$\mathcal{C}(E) = \left\langle \mathbf{q} \mathbf{W} \overline{\mathbf{p}} + \frac{1}{2} \lambda \mathcal{S}(\mathbf{q}) - \frac{1}{2} \mathbf{q} \overline{\mathbf{q}} \right\rangle \quad (4)$$

where  $\overline{(\cdot)}$  is the conjugate transpose operation. Furthermore

$$\mathcal{S}(\mathbf{q}) = \sum_{n=1}^N q_n \overline{q_n} - \frac{1}{N} \left( \sum_{n=1}^N q_n \right) \left( \sum_{n=1}^N \overline{q_n} \right) \quad (5)$$

which is analogous to (2). Again, this term can be interpreted as the variance of the complex form of the output  $q$ . We note that the variance of  $\mathbf{q}$  is maximized when the population of  $q$  values is sparse. Since these are complex variables, the implication is that *both* the amplitude and phase values are sparse.

We now define  $E_s$  to be

$$E_s = E + \beta \text{Re}[\mathcal{C}(E)]. \quad (6)$$

The first term  $E$  has been defined in (1) and depends only on the amplitude information in the network and not on phase values. The second term involving  $\mathcal{C}(E)$  contains both amplitude and phase values. The coefficient  $\beta$  defines the relative weight given to the phase information in the network arising from oscillations. When  $\beta = 0$ , (6) reduces to the case of the traditional NN without oscillatory units. The effect of  $\beta$  on the performance of the network will be examined in Section VII-A.

An important point is that the elements of the weight matrix  $\mathbf{W}$  in (4) are real valued. This is a departure from some of the methods in the literature that use complex weights, such as [17], and will be examined in Section IX.

We can gain further insight into the nature of the objective function by regrouping the terms

$$E_s = \left\langle \sum_{n,m} y_n W_{nm} x_m (1 + \beta \cos \Psi_{nm}) - \alpha \sum_n y_n^2 (1 + \beta) - \gamma \sum_{n \neq m} y_n y_m (1 + \beta \cos \Phi_{nm}) \right\rangle_{\mathcal{E}} \quad (7)$$

where  $\Psi_{nm} = \theta_n - \phi_m$ ,  $\Phi_{nm} = \theta_n - \theta_m$ ,  $\alpha = \lambda(2 - N)/2N$ , and  $\gamma = \lambda/N$ . This functional form suggests that we are in the presence of a hybrid Ising/XY model [25]; further analysis

along this line is beyond the scope of this paper and will be developed in future publications.

#### IV. NETWORK AND LEARNING DYNAMICS

To obtain the network dynamics, we derive the network updates to maximize the objective function over a short-time scale, using the method of gradient ascent. Given the condition of non-negativity on the amplitudes, we can choose the gradient in polar coordinates

$$\Delta y_n \sim \frac{\partial E_S}{\partial y_n} \quad \Delta \theta_n \sim \frac{1}{y_n} \frac{\partial E_S}{\partial \theta_n}. \quad (8)$$

Setting for simplicity  $\beta = 1$ , we obtain

$$\begin{aligned} \Delta y_n \sim & \sum_j W_{nj} x_j [1 + \cos(\phi_j - \theta_n)] - \alpha y_n \\ & - \gamma \sum_k y_k [1 + \cos(\theta_k - \theta_n)] \end{aligned} \quad (9)$$

$$\begin{aligned} \Delta \theta_n \sim & \sum_j W_{nj} x_j \sin(\phi_j - \theta_n) \\ & - \gamma \sum_k y_k \sin(\theta_k - \theta_n) \end{aligned} \quad (10)$$

$$\Delta \phi_n \sim \sum_j W_{jn} y_j \sin(\theta_j - \phi_n) \quad (11)$$

where  $\alpha = 1 - \lambda(N - 1)/N$  and  $\gamma = \lambda/N$ . Equations (9)–(11) describe the instantaneous evolution of the dynamics of the system, given a set of initial conditions. The initial conditions describe the starting values of the amplitudes and phases of the units in the two layers. For instance, the initial values of  $x$  can be equal to a set of image intensity values in an input scene, the initial values of  $y$  can be zero, and the phases can be randomized. During this evolution, the weights are kept fixed. In order to maximize the objective function over the entire input ensemble, we perform gradient descent over the synaptic weights with a slower time scale. In other words, the synaptic weights are modified only after  $y$  settles following the application of (9)–(11). This yields the learning update rule

$$\Delta W_{ij} \sim y_i x_j [1 + \cos(\phi_j - \theta_i)]. \quad (12)$$

Observe that this is a simple extension of the traditional Hebbian learning rule. The operation of these equations will become clear when we examine a concrete example in Figs. 4 and 5.

#### V. NETWORK CONFIGURATION

The objective function approach we have described thus far is quite general and it does not depend on a specific network topology. We present here a network to perform dynamical segmentation that implements the dynamics described in Section IV. The activation and phase variables are simply interpreted as oscillating units described by an amplitude and a phase. The phase for a given unit is derived from an ongoing oscillation whose natural period is fixed for that unit. The period for a given unit is randomly drawn from within a small range, chosen to be  $\tau \in [2.0, 2.1]$  ms. The network dynamics

determine how the initial amplitudes and phases of the units evolve over time.

The network is designed as follows.

- 1) A bottom layer receives input from an input signal and consists of dynamical units. The amplitude output of these units is only a function of their inputs, whereas the phase is a function of their natural frequency and feedback interactions with a top layer.
- 2) A top layer consists of dynamical units that receive inputs from the bottom layer through feedforward connections. For these units, the amplitude and the phase are computed by integrating inputs as a function of their amplitude and their phase difference with respect to the receiving phase.
- 3) The top layer sends feedback to the bottom layer, which is used to modify only the phase of the bottom layer's units as a function of the incoming amplitudes and phase differences with respect to the receiving phases. This behavior has been described by (9)–(11).

The network operates in two stages, learning and performance. Only during the learning stage are the feedforward and feedback connections modified, whereas the inhibitory connections stay fixed throughout. During the learning stage, elements of the input ensemble are presented to the network, upon which the response of the network is dynamically computed. A unit's phase update is the result of its internal frequency and of integrating all feedforward, inhibitory, and feedback inputs, weighted by their amplitude and the receiving unit's amplitude, as well as by a nonlinear function of their relative phases with respect to the receiving unit. For the amplitude update, the incoming amplitudes are weighted by a function of the relative phases and limited by a leakage function of the receiving unit's amplitude.

The rationale for these equations is the following: 1) the effect of feedforward inputs on the amplitude is stronger for synchronized units, 2) excitatory feedforward and feedback connections are such that units that are simultaneously active tend towards phase synchrony, and 3) inhibitory connections tend towards desynchronization; at the same time, they have a stronger depressing effect on the amplitude of synchronized units, and correspondingly, a weaker effect for desynchronized units.

In order to simulate this network, we used the following configuration. The system is organized into two layers as shown in Fig. 1. The lower layer consists of  $8 \times 8$  units, each of which receives an image intensity value as input. Each unit in the lower layer is connected to every unit in the upper layer, which consists of 16 units. Furthermore, the units in the upper layer possess lateral connections such that each unit is connected to every other unit. Finally, each unit in the upper layer is connected to every unit in the lower layer through feedback connections.

Fig. 2 shows the input images used to test the system. These images are of size  $8 \times 8$ , and possess gray levels in the range 0–255. They represent 16 different 2-D objects such as a square, triangle, cross, circle, etc.

#### VI. DYNAMICAL SEGMENTATION

We first describe the training paradigm used for the network. Then, we examine the effect of presenting superposed inputs after the training phase is completed. We show that the network

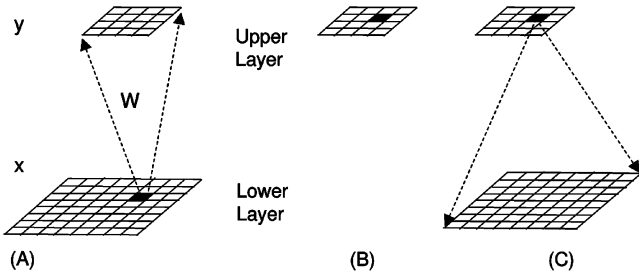


Fig. 1. Illustrating the network connectivity. (a) Input units  $X$  are arranged in a 2-D grid and can be thought of as image intensity values. The output units  $y$  also form a 2-D grid. Each input unit projects in a feedforward manner via a weight vector  $W$  to the output grid. (b) Identification of a particular unit in the output grid. (c) Feedback connections from this unit to the entire input grid.

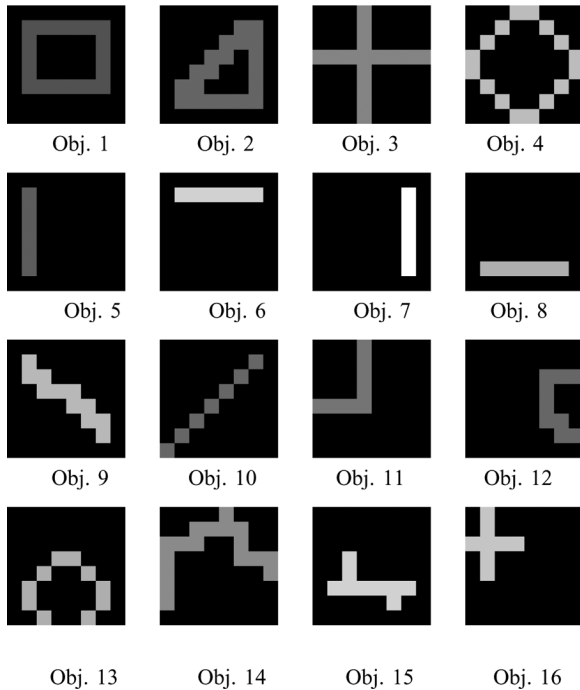


Fig. 2. Input images.

is able to achieve separation and segmentation of superposed inputs.

### A. Training

The system described in Section V is presented with a randomly chosen image from the set of images in Fig. 2. The inputs are preprocessed to convert them to zero mean and unit norm. The evolution of the network is determined by the application of (9)–(11). In order to understand the behavior of this system of equations, we consider Fig. 3. As can be seen, the amplitude activity in the network reaches a steady state after approximately 200 iterations. The learning rule of (12) is applied only after the network amplitude activity has settled down. We choose a nominal settling period of  $T = 200$  iterations.

We emphasize that the number of iterations is a function of the integration step size. Since the period of oscillation  $\tau \approx 2$  ms and the integration step to compute the updates is 0.1 ms, this effectively implies that the settling period is approximately ten

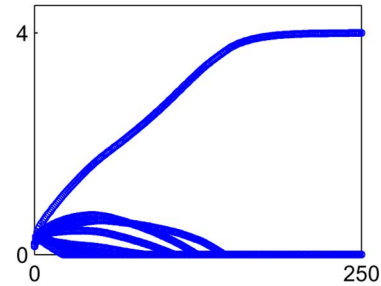


Fig. 3. Amplitude dynamics of elements in the upper layer of the network. Each curve shows the amplitude of an output unit plotted on the  $y$ -axis against time in iteration steps along the  $x$ -axis. For a single input, there are multiple upper layer units active initially, but after settling, a single winner emerges. The period of oscillation is  $\tau \approx 2$  ms and the integration step to compute the updates is 0.1 ms. Thus, the 250 iterations shown correspond to 12.5 cycles of oscillation. Though we show the time axis in iteration numbers in Figs. 4 and 5, it should be interpreted in terms of the number of oscillation cycles.

cycles of oscillation. In Section VII-B, we examine the effect of choosing a given settling time  $T$  on the performance of the network.

Within an appropriate parameter range, learning leads to a WTA dynamics upon presentation of one of the learned inputs. A detailed explanation for this behavior is provided in Appendix II.

This process of presentation of an input object, followed by a settling period and the application of the Hebbian learning rule, is repeated 1000 times. The typical behavior of the system is that a single unit in the output layer emerges as a winner. Furthermore, after the training, a unique winner is associated with each input. Note that the training has proceeded in an unsupervised fashion and that no explicit operation to compute the maximum output value in the network has been performed. The Hebbian learning rule is simply applied to all the weights in the network, regardless of which output unit has the maximum value.

### B. Behavior After Training: Superposed Inputs

After training, the system is presented with a superposition of two randomly selected objects from Fig. 2. Two aspects of the system response  $y$  are measured. The first aspect is to determine whether the winners for the superposed inputs are related to the winners when the inputs are presented separately. We term this measurement the separation accuracy, which is defined as follows. Let unit  $i$  in the upper layer be the winner for an input  $x_1$  and let unit  $j$  be the winner for input  $x_2$ . If units  $i$  and  $j$  in the upper layer are also winners when the input presented is  $x_1 + x_2$ , then we say the separation is performed correctly, otherwise not. The ratio of the total number of correctly separated cases to the total number of cases is the separation accuracy.

We used the following parameters to instantiate the model:  $\beta = 0.9$ ,  $\alpha = 0.5$ , and  $\gamma = 0.25$ ; the natural periods ( $\tau = 2\pi/\omega$ ) are drawn uniformly from  $[2, 2.1]$  and learning takes place after 20 real-time units or approximately ten cycles. Learning consists of 1000 presentations drawn at random from the training ensemble. The learning rate is reduced with an exponential schedule  $e^{-n/T}$ , where  $n$  is the presentation number and  $T = 2000$ .

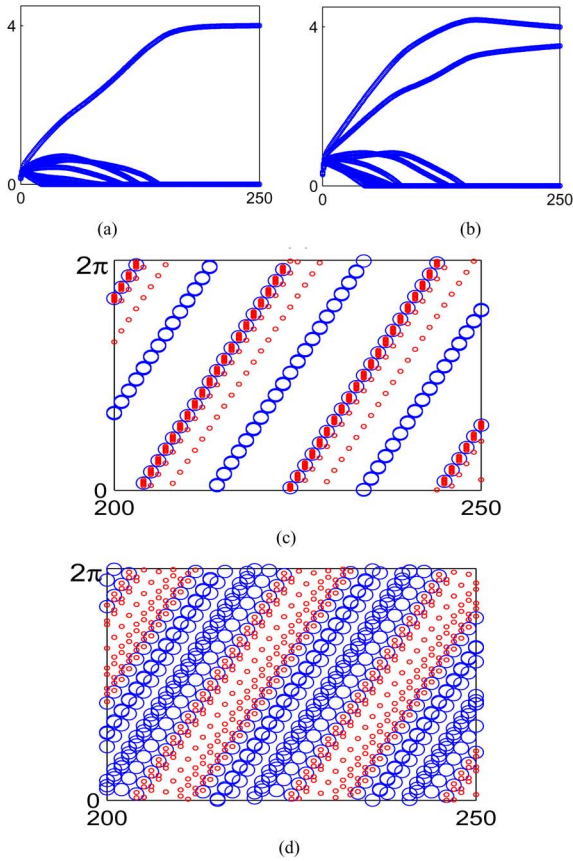


Fig. 4. Behavior of the network after learning. (a) Amplitude response upon presentation of an input from the training ensemble. The  $x$ -axis represents time shown as the number of iterations. The  $y$ -axis shows the value of the output  $y$  as a function of time. For the amplitude, the evolution is shown since the onset of the input. (c) Corresponding phase response. Phases values are displayed along the  $y$ -axis for each time step on the  $x$ -axis. Time is in simulation steps. Only the behavior after convergence is shown. The larger circles correspond to upper layer units and the smaller circles to lower layer units. (b) and (d) Response to the presentation of a mixture.

The behavior of the system is explained through the following figures. Fig. 4 shows the amplitude and phase responses of the units in the upper layer as a function of time.

When two inputs that have been learned are superposed and presented, we observe that two winners arise, as depicted in Fig. 4. This behavior is explained in Appendix II.

Fig. 5 shows the behavior of the system in phase space using a different visual representation. Inputs 2 and 4 from Fig. 2 were superposed and supplied as the input to the network. The activity of each unit in the network is plotted as a phasor, i.e., a vector with magnitude equal to the amplitude of the unit and angle equal to unit's phase. Initially, the network begins with randomized values for phase in the upper and lower layers and zero amplitude for the upper layer units. After ten iterations, several output units become active, indicated by increasing amplitudes. After 100 iterations, only two output units are active, as indicated by the phasor plot in the third row. (The two active output units correspond to the winners for each of the inputs when separately presented.) After 200 iterations, when the network activity has settled, we observe two active output units, which implies sparse coding in both the amplitude and phase space. There

are two clusters in the phase values of the input units. By comparing the histograms of the input unit phase values at the beginning and end, we observe that the distribution of input phases has also become more sparse.

The phase behavior of the system can be understood in more detail through Fig. 6. Suppose units  $i$  and  $j$  in the upper layer are the winners for a presentation consisting of a mixture of two inputs  $\mathbf{x}_1$  and  $\mathbf{x}_2$ , indicating that separation has taken place correctly. Here,  $i = 7$  and  $j = 2$ , for inputs corresponding to objects 1 and 3. Let the phases of units  $i$  and  $j$  in the upper layer be  $\theta_{2i}$  and  $\theta_{2j}$ , respectively. Consider a unit  $k$  in the lower layer with phase  $\theta_{1k}$ . The behavior of the network is such that the phase of this  $k$ th unit is usually synchronized with the phase of one of the winners in the upper layer.

Fig. 6(a) shows the activity of all the units in the lower layer displayed as a vector field. The magnitude of the vector reflects the amount of activity in the unit and the direction encodes the phase of the unit. The input layer in this case was formed by superposing objects 1 and 3 (rectangle and cross) in Fig. 2. Fig. 6(b) and (c) shows the phases of the two winners. As can be seen, units in the lower layer are synchronized with the winners in the upper layer. Furthermore, the units that have similar phase in the lower layer units tend to represent a single object, as can be seen from the silhouettes in the phase image of Fig. 6(a). In order to make this phenomenon more apparent, we display the segmented lower layer as follows. We display those units in the lower layer that are synchronized with the first winner in the upper layer. We allow a zone of synchronization, which is calculated as follows. Let  $d = \cos(\theta_{2i} - \theta_{1k})$  be a measure of the difference between the phase of an upper layer unit and a lower layer unit. (The cosine function is used to avoid the problem of taking the difference between two circular variables.) A value of  $d = 1$  represents perfect synchronization,  $d = 0$  represents no synchronization, and  $d = -1$  represents perfect antisynchronization. For the purpose of illustration, we assume that a value of  $d > 0.7$  represents synchronization. The units in the lower layer that are synchronized with the first winner in the upper layer are shown in Fig. 6(d) and those synchronized with the second winner are shown in Fig. 6(e). Fig. 6(d) shows that the phases of those lower layer members that represent object 1 are synchronized with the upper layer winner that also represents object 1. The upper layer winner for object 3 is synchronized with lower layer units that represent object 3.

Similarly, Fig. 7(a) shows the activity in the lower layer for a superposition of objects 3 and 4. The two winners in the upper layer again represent objects 3 and 4 and are also synchronized with the lower layer units that correspond to these same objects.

The implication of this result is that the phase information can be used to convey relationship information between different layers in a hierarchy. Thus, if some action is needed to be taken based on the identification of a certain object at a higher layer, the phase information provides information about where that object is localized in the lower layer. This is the essence of the binding problem as explained in Section I. For instance, suppose the presentation of an input image in visual area V1 of the cortex causes a unit in area IT (inferior temporal cortex) to fire

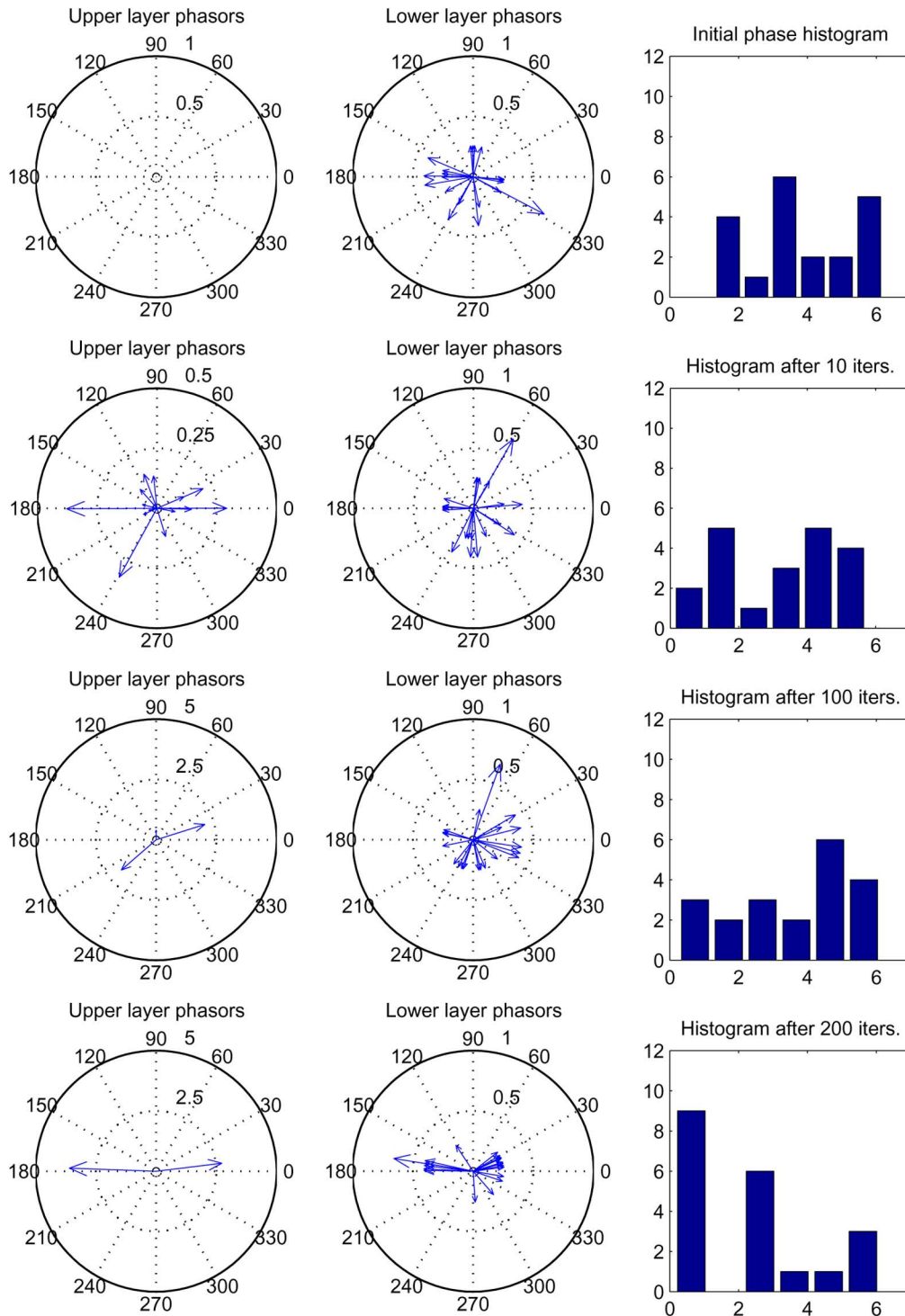


Fig. 5. Evolution of amplitudes and phases of the units in the system. The first column shows the phasors for the output layer units. The second column shows the phasors for the input layer units. The third column shows the histograms for the phases of the input layer units, using six bins to quantize the range  $0\text{--}360^\circ$ . Each row captures the network activity at a given iteration number. The first row is for iteration 0, followed by iteration 10, iteration 100, and finally, the last row represents iteration 200.

[26], indicating the presence of certain objects. Let us suppose that the prefrontal cortex generates a behavior that instructs the motor cortex to pick up a specific object. The motor cortex can infer the exact location of the object by utilizing the phase synchronization (binding) information that is jointly possessed by the high-level object representation in area IT and the low-level

units in area V1. Without this binding information, the precise location of the desired object cannot be identified.

## VII. SYSTEM PERFORMANCE

The phase relationship between the layers is not always as crisp as indicated in Fig. 6. The accuracy of phase segmenta-

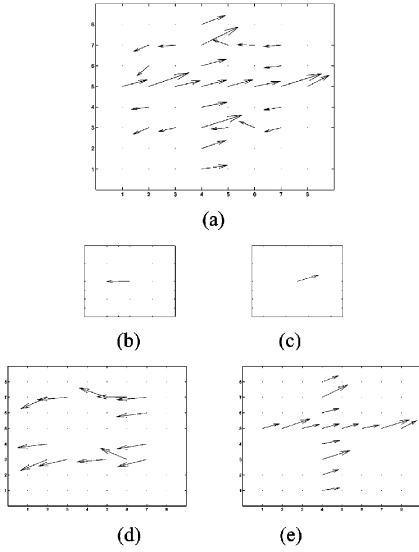


Fig. 6. Behavior of phase information. (a) Activity in the lower layer units, displayed as a vector field. The magnitude of the vector reflects the amount of activity in the unit and the direction encodes the phase of the unit. (b) Phase of the first winner, which is 3.147. (c) Phase of the second winner, which is 0.351. (d) Units in the lower layer that are synchronized with the first winner. (e) Units in the lower layer that are synchronized with the second winner.

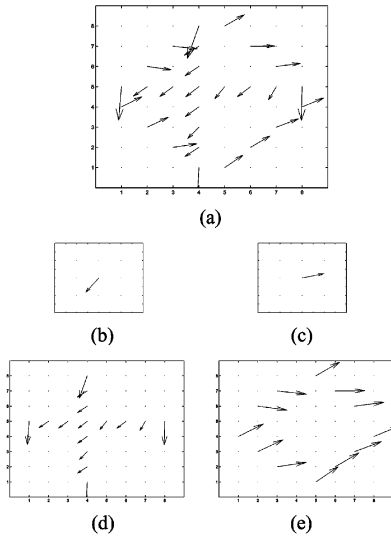


Fig. 7. Illustrating the behavior of phase information. (a) Activity in the lower layer units, displayed as a vector field. The magnitude of the vector reflects the amount of activity in the unit and the direction encodes the phase of the unit. (b) Phase of the first winner, which is 4.087. (c) Phase of the second winner, which is 0.241. (d) Units in the lower layer that are synchronized with the first winner. (e) Units in the lower layer that are synchronized with the second winner.

tion can be measured by computing the fraction of the units of the lower layer that correspond to a given object and are within some tolerance of the phase of the upper layer unit that represents the same object.

We conducted trials to determine the accuracy of the system in performing separation and segmentation. The entire network was randomized and inputs were presented individually during the training phase. Once the network was trained, its performance for separation and segmentation was measured for 100

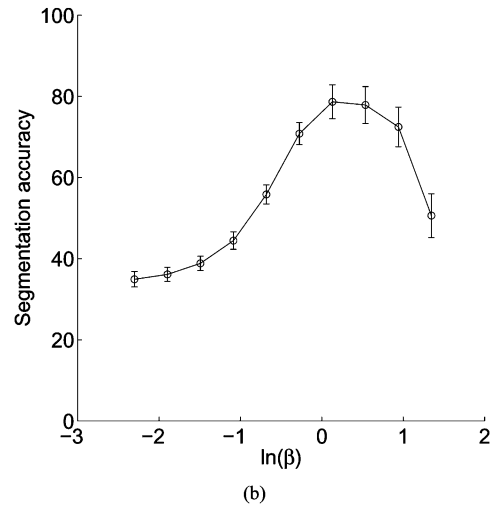
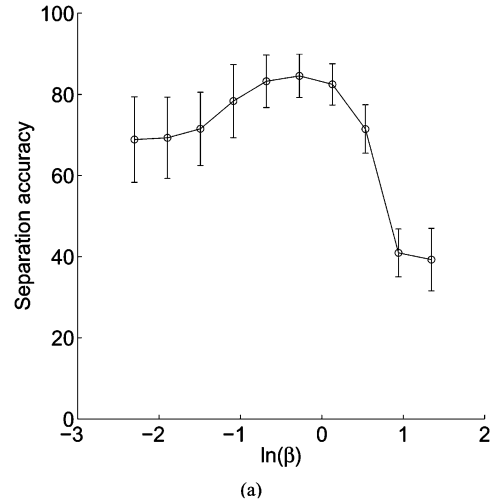


Fig. 8. (a) Separation accuracy of the network as a function of the parameter  $\beta$ . The natural logarithm of  $\beta$  is plotted on the  $x$ -axis. The range of  $\beta$  is 0.1–4.0. (b) Segmentation accuracy of the network as a function of  $\beta$ .

pairs of randomly selected inputs. This entire process was repeated 100 times. This enables us to measure the standard deviations of the accuracy estimates reliably.

#### A. Effect of Varying $\beta$ on Performance

The parameter  $\beta$  in (6) controls the weight given to the phase update equations. When  $\beta = 0$ , this reduces to a traditional NN without oscillations. We examine the effect of varying the  $\beta$  on the performance of the network and the results are summarized in Fig. 8. The parameter  $\beta$  was varied from  $\beta = 0.1$  to  $\beta = 4.0$ .

The separation accuracy of the network is reasonable even when  $\beta = 0$  and improves as  $\beta$  is increased. However, the segmentation accuracy is poor at low values of  $\beta$ , indicating that the phase information is crucial for performing accurate segmentation. As observed earlier, the phases of the units provide a source of information that is independent of the amplitude, and this allows the units of the network to perform associations between the winners in the upper layer and the members of the lower layer that gave rise to specific winners. Finally, observe that as  $\beta$  is increased beyond 1.0, the performance of the network degrades. This is because too much weight is being placed on the

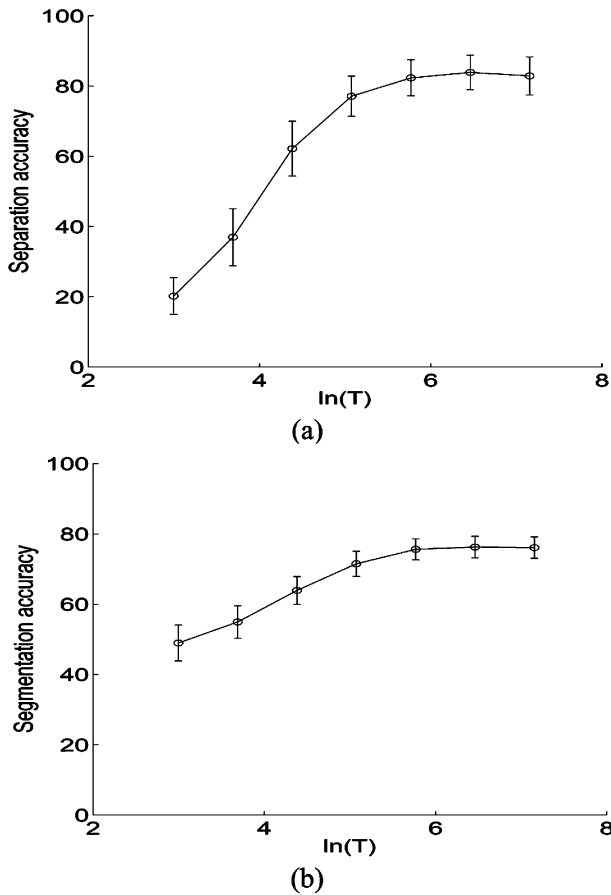


Fig. 9. (a) Separation accuracy of the network as a function of the settling time  $T$ . The natural logarithm of  $T$  is plotted on the  $x$ -axis. The range of  $T$  is 20–1280. The error bars indicate the standard deviation of the accuracy measurements. (b) Segmentation accuracy of the network as a function of  $T$ .

phase information and insufficient weight is given to the amplitude information. This suggests that optimal network performance is achieved when both phase and amplitude information are used in tandem.

### B. Effect of Settling Time on Performance

In Section IV, we presented the update equations that are applied to the network at each iteration and mentioned that the network is allowed to settle for  $T = 200$  iterations before applying the Hebbian learning rule of (12). We examined the effect of varying the settling time  $T$  on the performance of the network and the results are summarized in Fig. 9. The settling time was varied from  $T = 20$  to  $T = 1280$  in powers of two.

The network performs poorly when the learning rule is applied before the amplitudes are allowed to settle. As the number of settling iterations increases, the performance increases before beginning to plateau. Significantly more computation time is required for higher settling periods  $T$ . Hence, our choice of  $T = 200$  represents a tradeoff between reasonable computation time and accuracy of network performance.

### C. System Behavior With Noisy Inputs

We investigated the sensitivity of the network with respect to additive noise. The network was initially trained with noise-

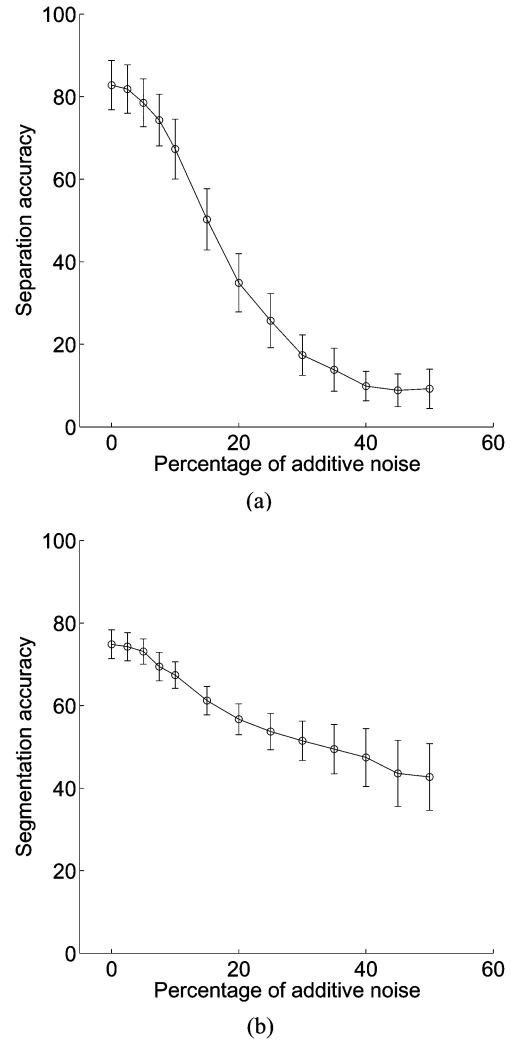


Fig. 10. (a) Separation accuracy of the network as a function of additive noise drawn from a Gaussian distribution. (b) Segmentation accuracy of the network as a function of additive noise.

free inputs, as shown in Fig. 2. Then, increasing amounts of additive noise were added to the inputs and the performance of the network was measured. We used two noise distributions: Gaussian and uniform noise.

For additive Gaussian noise, the noise at a given image pixel is generated by multiplying a random number drawn from a zero-mean unit variance distribution with a noise fraction  $f$ . Values of  $f$  ranged from 2.5% to 50% of the maximum input value. The inputs were remapped by adding an offset equal to the smallest negative value to make them positive, and then renormalized to the range  $[0, 1]$ . Fig. 10 shows the separation and segmentation accuracy of the network as a function of the percentage of additive Gaussian noise. As done earlier, 100 trials were used to generate each point in the graph.

For additive uniform noise, the noise at a given image pixel is generated by multiplying a random number drawn from a uniform distribution in the range  $[0, 1]$  with a noise fraction  $f$ . As before, values of  $f$  ranged from 2.5% to 50% of the maximum input value. The inputs were renormalized to the  $[0, 1]$  range. Fig. 11 shows the separation and segmentation accuracy of the network as a function of the percentage of additive Gaussian

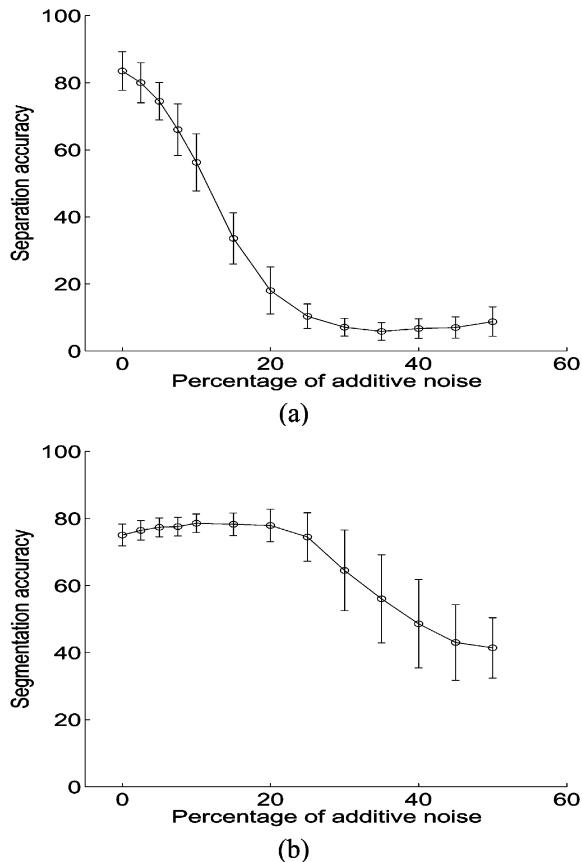


Fig. 11. (a) Separation accuracy of the network as a function of additive noise drawn from a uniform distribution. (b) Segmentation accuracy of the network as a function of additive noise.

noise. As done earlier, 100 trials were used to generate each point in the graph.

The network is fairly robust in the presence of low to moderate levels of noise. As the noise level increases substantially, the performance degrades markedly, which is to be expected. The segmentation accuracy appears to show less degradation in the presence of increasing noise. The reason for this is that the segmentation is measured only for cases that exhibit correct separation. Thus, given that two inputs are correctly separated, the constituents of those inputs are typically well segmented, with an accuracy of greater than 75% in most cases. We compare Figs. 10 and 11. For a noise level at a given percentage of the input, Gaussian noise distorts the input less than uniform noise, and hence, the performance of the system is relatively superior in the presence of Gaussian noise.

#### D. Comparison With Other Approaches

We also compare the performance of our system with respect to a Kohonen network used as a vector quantizer (VQ), where the neighborhood size is zero. Our results show that the performance of our oscillatory network model is comparable to that of a Kohonen VQ for classification and separation. Furthermore, the degradation of the system performance of the two systems is very similar in the presence of increasing additive Gaussian noise for both these tasks.

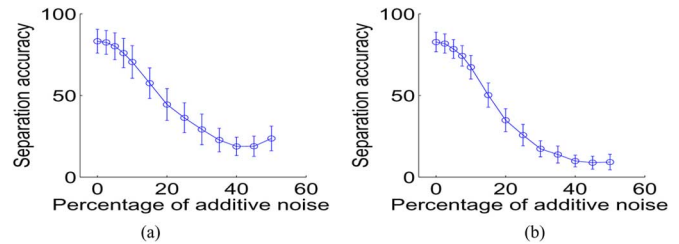


Fig. 12. (a) Separation accuracy of the WTA network as a function of additive Gaussian noise. (b) Separation accuracy of the oscillatory network as a function of additive Gaussian noise. The error bars represent the standard deviation of the separation accuracy.

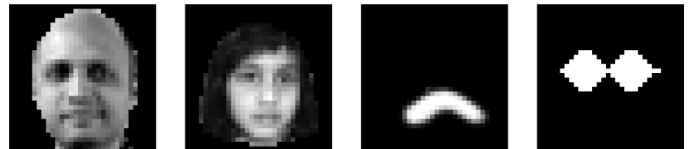


Fig. 13. Set of more complex inputs.

Fig. 12 shows the variation of separation accuracy with respect to the amount of additive Gaussian noise added to the inputs. This figure shows that the performance of the oscillatory network is comparable to that of the WTA network. Even though the oscillatory network is performing additional computation in order to achieve segmentation, it does not affect the ability of the network to separate mixtures of inputs. In other words, we do not have to give up the basic functionality of separation in order to achieve the more advanced functionality of segmentation, which the WTA network is not able to do; so the capabilities of separation and segmentation do not have to be traded off against each other, which is a distinct advantage of our method. Often, in the design of systems, the addition of new capabilities has to be traded off against existing capabilities. Our oscillatory model is more complex than a Kohonen VQ; this complexity is required to solve the problem of binding lower and higher level features in the presence of multiple objects. We conclude that our system can achieve efficient segmentation without compromising the basic capabilities of classification and separation.

Though it is difficult to make direct comparisons with other approaches, we can determine the relative degradation of the performance with respect to a baseline performance in the absence of noise. We compare our results with those obtained by Lee [21] who has performed such experiments. Lee's model showed a degradation of 8.7% in the correct recognition rate in the presence of a 10% noise level. The neural oscillatory elastic graph matching (NOEGM) model [21, p. 1239] showed a degradation of 25.9% for the same level of noise. Our system shows a performance degradation of 19% in separation accuracy and a degradation of 11% in segmentation accuracy for a 10% noise level.

#### E. Geometric Distortions

The network as presented in this paper is sensitive to geometric distortions of the input, such as translation, scale, and rotation variations. One approach to improving the system performance, say, in achieving translation invariance, is to use multiple layers, as compared with the two-layer system in this paper. Indeed, such an architecture exists in the human brain where

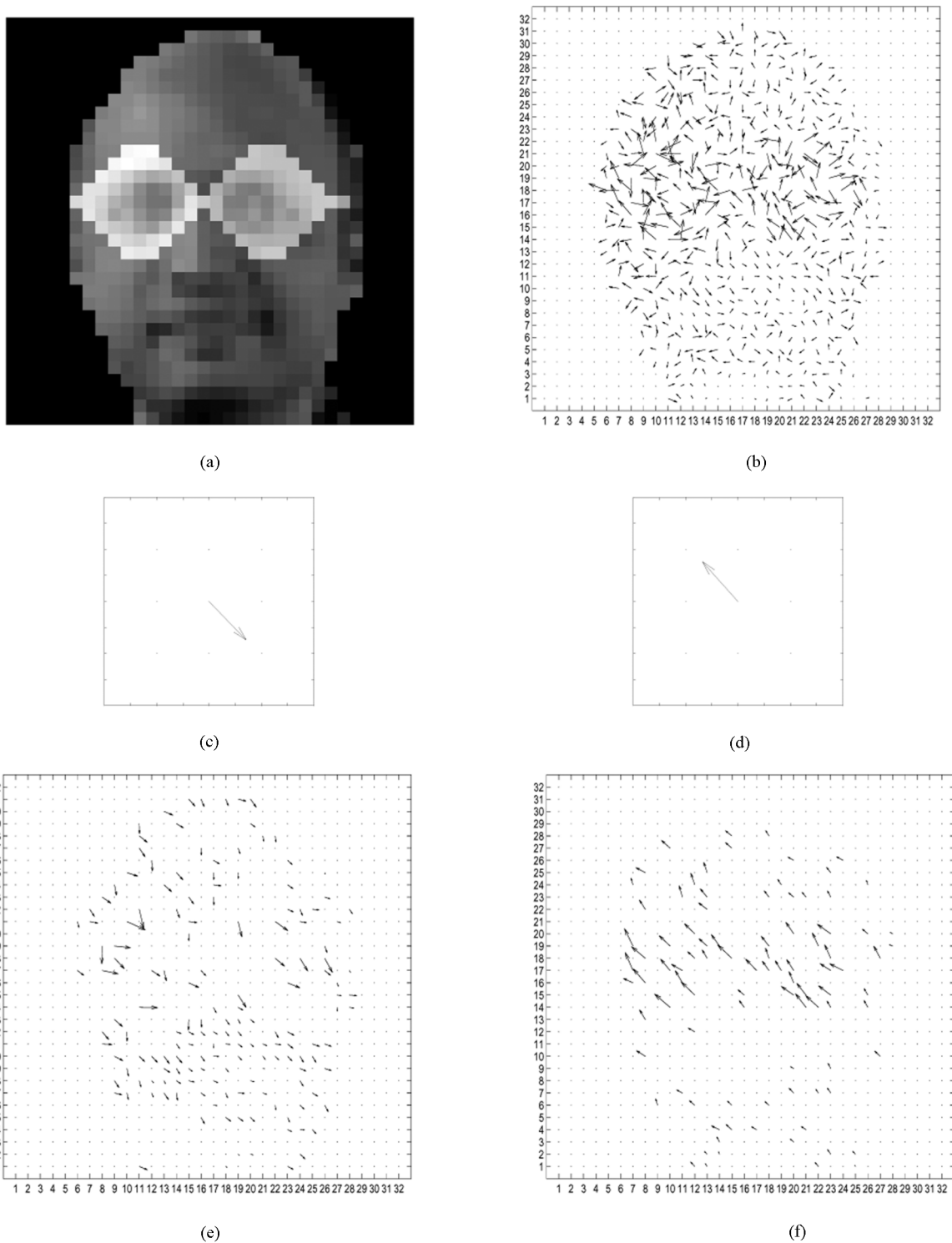


Fig. 14. Behavior of phase information. (a) Superposition of two images from Fig. 13. (b) Activity in the lower layer units, displayed as a vector field. (c) Phase of the first winner, which is 5.48. (d) Phase of the second winner, which is 2.297. (e) Units in the lower layer that are synchronized with the first winner. (f) Units in the lower layer that are synchronized with the second winner.

units in successively higher layers in the visual pathway possess larger receptive fields such that at the highest level, in the inferior temporal cortex [26], units respond to objects in a translation invariant manner.

Another approach is to use the temporal domain in training the network. Instead of learning to represent objects in static

locations, the learning takes place as the object is undergoing geometric distortion such as translation. This allows the categorization of the object at one location to be related to translated versions of the same object. We have been able to demonstrate limited translational variance [27] with such an approach. The main idea is to use a trace learning rule, as proposed by Wallis

[28], which allows associations to be formed between translated versions of a given object.

Let us define a moving average  $\tilde{y}$ , calculated as follows:

$$\tilde{y}(t+1) = \mu y(t) + (1 - \mu)\tilde{y}(t) \quad (13)$$

where  $\tilde{y}$  denotes a moving average of the value  $y$  and  $\mu = 0.8$ . The Hebbian learning rule in (12) is modified to use  $\tilde{y}$  as follows:

$$\Delta W_{ij} \sim \tilde{y}_i x_j [1 + \cos(\phi_j - \theta_i)]. \quad (14)$$

The effect of the trace learning rule is to establish equivalence between translated versions of an object. In [27], we show that it is possible to obtain translation invariance of  $\pm 3$  pixels for  $8 \times 8$  objects (i.e., for displacements of  $\pm 37.5\%$  of the image size) as shown in Fig. 2.

Though these early results are promising, further research is required to make the network more robust to geometric distortions and this topic is outside the scope of this paper. It is also possible to derive rotation, scale, and translation invariant features through the use of special processing of the input such as the use of Zernike moments in multilayer networks as shown in [29]. The techniques presented in this paper could be extended to multilayer networks and then combined with approaches to achieve invariance as presented in [29]. Another possibility is to use amplitude information in the feedback connections. An early investigation in this direction is presented in [30].

Other researchers, e.g., [21], have mentioned that they are extending oscillatory networks to handle complex scene analysis, including problems such as invariance. However, results have yet to be reported.

### F. Experiments With More Complex Images

We used a more complex data set, shown in Fig. 13. The images are of size  $32 \times 32$  and two of them contain real facial image data. The size of the output layer was  $4 \times 4$  units, as before, and the same parameters were used to instantiate the model as described in Section VI-B. Fig. 14 illustrates the behavior of phases in the upper and lower layer of the network for a superposition of objects 1 and 4. The objects consist of a face and a pair of glasses. The winners in the upper layer are almost perfectly out of phase, which is desirable. The phases of the lower layer units are grouped into two populations, those synchronized with the face and glasses, respectively, which is also desirable. Though the spatial locations of the lower layer phases matched to the winner representing the glasses agree with the locations of the pixels representing the glasses, there are spurious matches to locations corresponding to the face. The separation accuracy for this set of input images is 48.4%, and the segmentation accuracy is 41%. The separation and segmentation accuracy for this set of inputs is lower than obtained for the simpler class of inputs and it is likely due to the more complex nature of the input objects.

This experiment shows that additional capabilities may need to be introduced into our model to make it work with larger, more complex images. A promising direction to explore is the

incorporation of amplitude in the feedback connections, as explored in [30]. In addition, it may be necessary to increase the number of layers in the network from two to three or more to effectively utilize feedback information. We would also like to point out that we do not perform any additional preprocessing of the images, such as Laplacian filtering, edge detection, or feature extraction as is the norm in most image recognition applications. Our computations are performed directly on the original image. It is likely that better results may be obtained with special feature extraction steps. Furthermore, we use an unsupervised learning framework, in contrast to supervised learning paradigms used in typical object recognition tasks.

### VIII. NEURAL DYNAMICS AND BIOLOGICAL CONSTRAINTS

It is possible to map the abstract network equations presented in Section VII to realistic neural dynamics. Neural oscillations have been described in terms of field dynamics of small ensembles of locally interacting neurons, exemplified by the cortical dynamics derived by Wilson and Cowan [31]

$$\tau \dot{E}_i = -E_i + S \left( \sum_j [w_{ij}^{EE} E_j - w_{ij}^{IE} I_j] + U_i \right) \quad (15)$$

$$\tau \dot{I}_i = -I_i + S \left( \sum_j [w_{ij}^{EI} E_j - w_{ij}^{II} I_j] + V_i \right) \quad (16)$$

where  $E$  and  $I$  are the excitatory and inhibitory local populations,  $U$  and  $V$  are external inputs, and  $S(\cdot)$  is a monotonic function. The presence of generalized oscillations in the range of 40 Hz has been documented in a large number of experiments, in particular, related to sensory processing and recognition. Interestingly, the Wilson–Cowan equations can generate oscillations under a wide range of conditions, and in particular, they can create type-II oscillations [32], so that the frequency of oscillation is relatively constant as a function of the input. The possibility of defining phases is predicated upon the existence of oscillations. This implies that the phase equations will include an additional term  $\Delta\psi_n \sim \Psi_n + \omega_n$ , where the first term is the previously determined interaction term and the second one is the natural frequency of the oscillations. However, if we assume that these natural frequencies are sufficiently similar, the effect on the gradient ascent of the objective function is negligible. The additional component of the online energy change is

$$\begin{aligned} \Delta E_\omega &\sim \sum_{i \in \mathbf{p}, j \in \mathbf{q}} \omega_i W_{ij} x_i y_j \frac{\partial E}{\partial \phi_i} \\ &+ \sum_{m \in \mathbf{q}, n \in \mathbf{p}} \omega_m y_m W_{mn} x_n \frac{\partial E}{\partial \theta_m} \end{aligned} \quad (17)$$

$$\begin{aligned} \Delta E_\omega &\approx \Omega \sum_{n,m} x_m W_{nm} y_n (\sin \Phi_{nm} + \sin \Phi_{mn}) \\ &- \Omega \gamma \sum_{n,m} y_n y_m \sin \Theta_{nm} = 0. \end{aligned} \quad (18)$$

The interaction terms in the update equations can be understood also in biological terms. We assume that the receiving

ensemble is oscillating at a frequency similar to that of the forcing, so that the phase of the receiving ensemble changes slowly relative to the forcing phase; moreover, we interpret the amplitude as representing the oscillating rate of the ensemble. Therefore, an instantaneous description of the ensemble and the forcing can be approximated by  $r(t) \sim [1 + \cos(\theta(t))]$  and  $f(t) \sim [1 + \cos(\theta(t) + \Phi)]$ , respectively, where  $\Phi$  is the (slowly varying) phase difference between the ensemble and the forcing. We can compute then the average change over a cycle as

$$\langle \Delta r \rangle \sim \langle f \rangle + k_0 \langle r f \rangle. \quad (19)$$

The rationale is as follows: For an ensemble of spiking or threshold elements with leakage, to a first order, the change in rate is proportional to the average forcing amplitude, and to a second order, to the coincidence between the forcing and the proximity to threshold. Thus, the forcing has the strongest effect when it peaks near the state at which the ensemble is closer in average to the threshold. Similarly, the phase change is proportional to the difference between the forcing's and ensemble's rate derivative times the ensemble's rate

$$\langle \Delta \theta \rangle \sim \left\langle r \left( \frac{df}{dt} - \frac{dr}{dt} \right) \right\rangle \approx \omega \left\langle r \left( \frac{df}{d\theta} - \frac{dr}{d\theta} \right) \right\rangle. \quad (20)$$

From (19), we obtain  $\langle \Delta r \rangle \sim 1 + \beta \cos \Phi$ , where  $\beta = k_0/(1 + k_0)$  and, from (20),  $\langle \Delta \theta \rangle \sim \sin \Phi$ . Indeed, simulations based on (8) show that the dynamical equations derived in Section IV are a reasonable approximation to the ensemble behavior, within a range of parameters. Further considerations are beyond the scope of this paper.

Finally, for biological considerations, one can interpret that in (11) (update of the input layer's phase), the weight matrix  $W_{jn} = (W_{nj})^T$  is replaced by a new set of feedback connections  $W_{nj}^{FB}$ . This will indeed be the case: Given that the Hebbian learning rule (12) is symmetric in its arguments,  $W_{nm} \xrightarrow[t \rightarrow \infty]{} W_{mn}^{FB}$ .

## IX. DISCUSSION

We have shown that the approach derived from the simple objective function in (6) can do the following: 1) provide very simple neural dynamical and learning rules, 2) achieve good computational results in solving the segmentation problem, and 3) provide a reasonable biological interpretation of segmentation, including the ability of the network to behave without supervision. Moreover, the scheme presented here lends itself relatively easily to generalizations, in particular, by extending the feedback connections to affect not only the phase of lower units, but also their amplitude. We are indeed working towards an integrated model to account for both segmentation and inference in the presence of partial information [30]. We have also shown the ability of the network presented in this paper to achieve translation invariant encoding of objects, with a minor modification to the learning rule [27].

While building on previous work, the system presented in this paper significantly improves existing formulations by simplifying the network architecture required, presenting a simple objective function to understand the system behavior and demonstrating the ability to solve separation and segmentation problems in an unsupervised manner.

### A. Practical Considerations

The oscillatory network model presented in this paper is able to perform both separation and segmentation of mixtures. The ability to perform segmentation arises from the use of oscillations, which are computationally expensive to simulate. Indeed, this has been the experience of other researchers as well, which has prompted the investigation of oscillator-based hardware solutions [33]. We have chosen to implement the system on a parallel platform, an IBM p690 shared memory machine with 24 processors. The parallelization of the code was performed with the pthreads library.

We used small images in order to demonstrate that an optimization approach can be used to tackle the binding problem. Further research needs to be undertaken to scale the model up to more realistic image sizes and content. We also observe that one of the requirements imposed on our model was that it should function in an unsupervised manner, due to our desire to explain biological phenomena. Systems of practical nature designed to solve specific problems such as image retrieval may not have such a requirement, which means they can use alternate learning methods such as supervised learning to achieve superior performance. At the same time, practical systems for image retrieval may not need to solve the binding problem because the actions they undertake may not require identification of the inputs that caused a certain classification to arise. For instance, the reporting of a class label would constitute a satisfactory action.

Though the temporal domain may be essential in solving the binding problem [2], the creation of robust systems that utilize NN-based temporal approaches remains a significant challenge. This challenge can be addressed by creating the right formalization to study and model temporal NNs and also engaging in detailed experimentation to evaluate and improve the implementation of the models.

## X. CONCLUSION

In this paper, we presented a biologically motivated network configuration aimed at addressing the binding problem. The system requirements are that it learns in an unsupervised fashion and is able to separate and segment mixtures of inputs that have been learned. By meeting these requirements, we address an important aspect of development in the HVS, where learning is initially unsupervised and the ability to recognize and segment superposed objects is essential for interactions with the real world.

Oscillatory networks have been investigated as a possible solution to the binding problem. We presented an optimization approach to state the desired behavior of an oscillatory network. The network dynamics are derived in a principled manner by using an objective function that rewards sparse encoding of an

input space. We showed that the network dynamics can be implemented in a simple network, and demonstrated its performance in terms of binding through phase synchronization. The sparseness of encoding was demonstrated with real visual inputs. The network is able to separate and segment mixtures of inputs that have been learned in an unsupervised manner, and is able to cope with a considerable degree of spatial overlap of the inputs.

In summary, our model presents many interesting novel features with a rich potential for formalization and generalization. There are several fruitful extensions of our model, which we have begun pursuing [27], [30]. Further work needs to be done to investigate the robustness of oscillatory networks in the handling of geometric distortions.

#### APPENDIX I NORMALIZATION CONSTRAINT

Imposing normalization on the synaptic weights and whitening of the inputs, it is easy to see that the third term in (1) can be dropped. Applying gradient descent on the synaptic vectors, we find  $\Delta \mathbf{W} \sim \langle \mathbf{y}\mathbf{x}^T \rangle - \mathbf{W}$ . The normalization constraint implies  $\mathbf{W}^{(t+1)} = \mathbf{v}/|\mathbf{v}|$  and  $\mathbf{v} = \mu\mathbf{y}\mathbf{x}^T + (1 - \mu)\mathbf{W}^{(n)}$ .

Given the normalization constraint on the synaptic vectors and  $\mu \ll 1$ ,  $\mathbf{W}_{n+1} \approx \mathbf{W}_n + (\mu/1 - \mu)\mathbf{y}\mathbf{x}^T$ , and therefore,  $\Delta \mathbf{W}_n \sim \mathbf{y}\mathbf{x}^T$ , so that the objective function can be simplified as  $E = \langle \mathbf{y}\mathbf{W}\mathbf{x}^T + (1/2)\lambda S(\mathbf{y}) - (1/2)\mathbf{y}^2 \rangle_{\mathcal{E}}$ , assuming that synaptic normalization is imposed during the maximization process.

#### APPENDIX II WTA DYNAMICS

The dynamics derived from (1) result in the emergence of a unique winner after learning and at least two winners when two inputs are superimposed, as shown in Fig. 4. To understand this, we start by writing  $\dot{y}_n = I_n - y_n - \lambda \sum_{m \neq n} y_m$ , where  $I_n = \mathbf{W}_n \cdot \mathbf{x}$  is the input to unit  $n$ . In steady state,  $\dot{y}_n = 0 \forall n$ ; let us assume that, for the maximal input,  $y_M \approx I_M$ , and therefore,  $y_n \approx 0 \forall n \neq M$ . In this case, the condition for stability implies  $x_n - \lambda x_M < 0 \forall n \neq M$ , or equivalently,  $I_M > x_N/\lambda \forall n \neq M$ ; this condition can be achieved if the weight vectors are properly aligned after learning.

When two vectors are presented to the network after learning, a similar analysis shows that the solution of two winners is a stable one, provided that  $(I_{M_1}^{(1)} + I_{M_2}^{(1)})/(1 + \lambda) + (I_{M_1}^{(2)} + I_{M_2}^{(2)})/(1 + \lambda) > I_n^{(1)}/\lambda + I_n^{(2)}/\lambda$ .

#### ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions, which improved this paper.

#### REFERENCES

- [1] C. Von der Malsburg, "The what and why of binding: The modeler's perspective," *Neuron*, pp. 95–104, 1999.
- [2] D. Wang, "The time dimension for scene analysis," *IEEE Trans. Neural Netw.*, vol. 16, no. 6, pp. 1401–1426, Nov. 2005.
- [3] F. Rosenblatt, *Principles of Neurodynamics: Perception and the Theory of Brain Mechanisms*. Washington, DC: Spartan Books, 1962.
- [4] S. Edelman, *Representation and Recognition in Vision*. Cambridge, MA: MIT Press, 1999.
- [5] A. Bell and T. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Comput.*, vol. 7, pp. 1129–1159, 1995.
- [6] S. Ullman, M. Vidal-Naquet, and E. Sali, "Visual features of intermediate complexity and their use in classification," *Nature Neurosci.*, vol. 5, no. 7, pp. 682–7, 2002.
- [7] C. Gray, P. König, A. Engel, and W. Singer, "Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties," *Nature*, vol. 338, no. 6213, pp. 334–337, 1989.
- [8] E. Rodriguez, N. George, J.-P. Lachaux, J. Martinerie, B. Renault, and F. Varela, "Perception's shadow: Long-distance synchronization of human brain activity," *Nature*, vol. 397, no. 6718, pp. 430–433, 1999.
- [9] E. Basar, C. Basar-Eroglu, S. Karakas, and M. Schürmann, "Oscillatory brain theory: A new trend in neuroscience," *IEEE Eng. Med. Biol. Mag.*, vol. 18, no. 3, pp. 56–66, May/Jun. 1999.
- [10] C. von der Malsburg and W. Schneider, "A neural cocktail-party processor," *Biol. Cybern.*, vol. 54, no. 1, pp. 29–40, 1986.
- [11] J. Buhmann and C. Von Der Malsburg, "Sensory segmentation by neural oscillators," in *Proc. Int. Joint Conf. Neural Networks II*, 1991, pp. 603–607.
- [12] K. Chen, D. Wang, and X. Liu, "Weight adaptation and oscillatory correlation for image segmentation," *IEEE Trans. Neural Netw.*, vol. 11, no. 5, pp. 1106–1123, Sep. 2000.
- [13] D. L. Wang and X. Liu, "Scene analysis by integrating primitive segmentation and associative memory," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 32, no. 3, pp. 254–268, Jun. 2002.
- [14] E. Izhikevich, "Weakly pulse-coupled oscillators, FM interactions, synchronization, and oscillatory associative memory," *IEEE Trans. Neural Netw.*, vol. 10, no. 3, pp. 508–526, May 1999.
- [15] F. Hoppensteadt and E. Izhikevich, "Pattern recognition via synchronization in phase-locked loop neural networks," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 734–734, May 2000.
- [16] H. Sun, L. Liu, and A. Guo, "A neurocomputational model of figure-ground discrimination and target tracking," *IEEE Trans. Neural Netw.*, vol. 10, no. 4, pp. 860–884, Jul. 1999.
- [17] R. Zemel, C. Williams, and M. Mozel, "Lending direction to neural networks," *Neural Netw.*, vol. 8, no. 4, pp. 503–512, 1995.
- [18] K. Chen, D. Wang, and X. Liu, "Weight adaptation and oscillatory correlation for image segmentation," *IEEE Trans. Neural Netw.*, vol. 11, no. 5, pp. 1106–1123, Sep. 2000.
- [19] J. Cosp and J. Madrenas, "Scene segmentation using neuromorphic oscillatory networks," *IEEE Trans. Neural Netw.*, vol. 14, no. 5, pp. 1278–1296, Sep. 2003.
- [20] R. Eckhorn, A. M. Gail, A. Bruns, A. Gabriel, B. Al-Shaikhli, and M. Saam, "Different types of signal coupling in the visual cortex related to neural mechanisms of associative processing and perception," *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1039–1052, Sep. 2004.
- [21] R. Lee, "A transient-chaotic autoassociative neural network based on lee oscillators," *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1228–1243, Sep. 2004.
- [22] F. van der Velde and M. de Kamps, "From knowing what to knowing where: Modeling object-based attention with feedback disinhibition of activation," *J. Cogn. Neurosci.*, vol. 13, pp. 479–491, 2001.
- [23] S. Weng, H. Wersing, J. J. Steil, and H. Ritter, "Learning lateral interactions for feature binding and sensory segmentation from prototypic basis interactions," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 843–862, Jul. 2006.
- [24] B. Olshausen and D. Fields, "Natural image statistical and efficient coding," *Network: Comput. Neural Syst.*, vol. 7, pp. 333–339, 1996.
- [25] J. Kosterlitz and D. Thouless, "Ordering, metastability and phase transitions in 2d systems," *J. Phys. C, Solid State Phys.*, vol. 6, pp. 1181–1203, 1973.
- [26] R. Quiroga *et al.*, "Invariant visual representation by single neurons in the human brain," *Nature*, vol. 435, no. 2, pp. 1102–7, Jun. 2005.
- [27] A. R. Rao, G. A. Cecchi, C. C. Peck, and J. R. Kozloski, "Translation invariance in a network of oscillatory units," *Proc. SPIE—Int. Soc. Opt. Eng.*, vol. 6064, Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning, pp. 606411–606419, 2006.
- [28] G. Wallis, "Using spatio-temporal correlations to learn invariant object recognition," *Neural Netw.*, pp. 1513–1519, 1996.
- [29] S. Perantonis and P. J. G. Lisboa, "Translation, rotation and scale invariant pattern recognition by high-order neural networks and moment classifiers," *IEEE Trans. Neural Netw.*, vol. 3, no. 2, pp. 241–251, Mar. 1992.

- [30] Y. Liu *et al.*, "Inference and segmentation in cortical processing," *Proc. SPIE—Int. Soc. Opt. Eng.*, vol. 6057, Human Vision and Electronic Imaging XI, pp. 6057OY1–6057OY10, 2006.
- [31] H. Wilson and J. Cowan, "Excitatory and inhibitory interactions in localized populations of model neurons," *Biophys. J.*, vol. 12, pp. 1–24, 1972.
- [32] F. Hoppensteadt and E. Izhikevich, *Weakly Connected Neural Networks*. New York: Springer-Verlag, 1997.
- [33] D. Fernandes and P. Navaux, "An oscillatory neural network for image segmentation," in *Lecture Notes in Computer Science*. Berlin, Germany: Springer-Verlag, 2003, vol. 2905/2003, Progress in Pattern Recognition, Speech and Image Analysis, pp. 667–674.



**A. Ravishankar Rao** (SM'00) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, India, in 1984 and the Ph.D. degree in computer engineering from the University of Michigan, Ann Arbor, in 1989.

Currently, he is a Research Staff Member at the Computational Biology Center, IBM T. J. Watson Research Center, Yorktown Heights, NY. His research interests include image processing, computer vision and computational neuroscience. His work has resulted in seventeen patents and over

fifty publications. He has published a book entitled *A Taxonomy for Texture Description and Identification* (New York: Springer-Verlag, 1990).

Dr. Rao chaired The International Society for Optical Engineering (SPIE) Conference on Machine Vision and Applications from 1996 to 2007. He served as the Tutorials Chair for the IS&T PICS Conference from 1999 to 2001. He is an Associate Editor of the journals *Pattern Recognition* and *Machine Vision and Applications*. He was named a Master Inventor at IBM Research, in recognition of his sustained and valuable invention activity.



**Guillermo A. Cecchi** received the M.S. degree in physics from the University of La Plata, La Plata, Argentina, in 1992 and the Ph.D. degree in physics and biology from The Rockefeller University, New York, NY, in 1999.

He was a Postdoctoral fellow at the Laboratory of Mathematical Physics, The Rockefeller University, and in 2000, he joined the Functional Neuroimaging Laboratory, Department of Psychiatry, Cornell University Medical School, New York, NY, as a Physics Fellow. In 2001, he joined IBM Research, Yorktown

Heights, NY, as a Research Staff Member, and since then, he has been working on computational neuroscience, brain imaging, and applications of statistical network theory in biology.



**Charles C. Peck** received the Ph.D. degree in electrical engineering from the University of Cincinnati, Cincinnati, OH, in 1994.

Currently, he leads IBM's Biometaphorical Computing Research Group, dedicated to analyzing and modeling the brain for scientific, medical, and technology applications. This includes data-driven modeling via the Blue Brain collaboration with Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, and theory driven modeling of global brain function and individual structures.

Dr. Peck was one of nine people from Lockheed Martin to win the NOVA Award for Technical Excellence, the corporation's highest honor, in 1998. He was also selected by the National Academy of Engineering (NAE) as one of America's top young engineers.



**James R. Kozloski** received the Ph.D. degree in neuroscience from the University of Pennsylvania, Philadelphia, in 1999.

He subsequently held a research position at Columbia University, New York, NY, in the laboratory of Dr. R. Yuste, before joining the research staff of IBM, Yorktown Heights, NY, in 2001. He was named Adjunct Professor at Columbia in 2006. His research interests, primarily in computational biology, include structural biology, neural system modeling, functional simulations of neocortex, and

molecular biology. He invents in the area of neurotechnology and designs parallel computing software architectures and interfaces for both simulation and data analysis problems in neuroscience