

# An optimization approach to achieve unsupervised segmentation and binding in a dynamical network

A. Ravishankar Rao, Guillermo A. Cecchi, Charles C. Peck, James R. Kozloski

**Abstract**—We present a novel network of oscillatory units, whose behavior is described by the amplitude and phase of oscillations. While building on previous work, the system presented in this paper significantly improves existing formulations by simplifying the network architecture required, presents a simple objective function to understand the system behavior, and demonstrates the ability to solve deconvolution and segmentation problems in an unsupervised manner.

We derive the network dynamics from an objective function that rewards both the faithfulness and the sparseness of representation. The resulting network architecture is simple, and the dynamics are straightforward to interpret. This network functions in an unsupervised manner, and is able to form unique representations for a set of inputs. Once the set of inputs is learnt, the network can deconvolve mixtures of inputs. A significant capability of the network is that it segments its inputs into components that most contribute to the classification of a given input object. The network dynamics is such that the units exhibit synchronization through phase locking after an initial settling period. The behavior of deconvolution is determined by the amplitude of units in an output layer, while segmentation is simultaneously determined through phase similarity between the input and output layer units. Thus, the network exhibits the binding of specific input units and the output units that represent a classification of these input units by means of phase similarity.

Learning is unsupervised and based on a straightforward modification of the traditional Hebbian update by including relative phase information. We demonstrate that efficient segmentation can be achieved even when there is considerable superposition of the inputs.

**Index Terms**—deconvolution, binding problem, phase correlation, synchronization, oscillations.

## I. INTRODUCTION

Deconvolution and blind deconvolution (i.e. identifying the presence of specific objects in the visual field) have been extensively studied in the neural network literature [1]. On the other hand, segmentation, which refers 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), has been addressed more effectively with non-neural approaches [2].

However, inspired by experimental evidence of a role for synchronization of neural responses in a variety of motor and cognitive tasks, and in particular in perceptual recognition [3], [4], Malsburg and Shneider were among the first to propose the use of synchronization to perform segmentation of a mixture of signals [5]. 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 amongst 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 [5] inherit the same shortcomings [6], [7], [8], 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 overlapping one-dimensional inputs, and can potentially be generalized to higher dimensions.

Segmentation can also be considered a solution to the binding problem [9], an issue extensively discussed in the neuroscience literature. The problem, which can be traced back to Rosenblatt, states that neural networks do not have the capacity to encode superimposed inputs (Rosenblatt's superposition catastrophe [10]). 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. In the presence of distractors at the low level, consisting say of other objects, traditional neural networks cannot segment the edges corresponding to the square from the other elements. One way of tackling this issue in a neural network 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.

## II. SEGMENTATION

We will first introduce an objective function for vector quantization or sparse representation (cf. [15]), in which it is assumed that the inputs  $\mathbf{x}$  drawn from an input ensemble are represented by an output layer  $\mathbf{y}$  through synaptic weights  $\{W_{ij}\}$ , and such that a non-negativity condition is imposed on the output layer,  $y_i \geq 0 \forall i$  (see Figure 1). Note that the output  $\mathbf{y} \neq \mathbf{W}\mathbf{x}$  due to the presence of lateral interactions,

The authors are at the T.J. Watson IBM Research Center, Yorktown Heights, NY 10598. For correspondence, contact ravirao@us.ibm.com or gcecchi@us.ibm.com

and the use of a squashing function to make  $\mathbf{y}$  non-negative. The objective function  $E$  is expressed as

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

where  $\mathcal{E}$  represents the input ensemble. The first term is related to the faithfulness of representation, and rewards the alignment between the network's output and the feed-forward input. 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:

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

where  $\mathcal{N}$  represents the network, consisting of  $N$  units. Given the imposition of non-negativity of  $y_i$ , this term rewards the sparseness of the representation. Imposing normalization on the synaptic weights and whitening of the inputs, the objective function can be simplified as:

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

assuming that synaptic normalization is enforced during the maximization process. Applying gradient ascent to the objective function *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.

In the above 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  $n^{\text{th}}$  unit in the lower layer and upper layer respectively.

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. Consider

$$E_s = E + \beta \text{Re}[\mathcal{C}(E)] \quad (4)$$

where  $\mathcal{C}[E] = E(\mathbf{p}, \mathbf{q})$  is the complex extension of the energy

$$\mathcal{C}(E) = \mathbf{q}W\overline{\mathbf{p}} + \frac{1}{2}\lambda S(\mathbf{q}) - \frac{1}{2}\mathbf{q}\overline{\mathbf{q}} \quad (5)$$

where  $p_n = x_n e^{i\phi_n}$ ,  $q_n = y_n e^{i\theta_n}$ ,  $\overline{(\cdot)}$  is the conjugate operation, and

$$S(\mathbf{q}) = \sum_{n=1}^N q_n \overline{q_n} - \frac{1}{N} (\sum_{n=1}^N q_n) (\sum_{n=1}^N \overline{q_n}) \quad (6)$$

which is analogous to equation 2. In these equations, the parameter  $\beta$  determines the weight given to the energy arising from oscillations, such that the case  $\beta = 0$  reduces to the case of the traditional neural network without oscillatory

units. The effect of  $\beta$  on the performance of the network will be examined in Section V-A.

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

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

where  $\Psi_{nm} = \theta_n - \phi_m$ ,  $\Phi_{nm} = \theta_n - \theta_m$ ,  $\alpha = (N - \lambda(N - 1))/2N$ ,  $\gamma = \lambda/2N$ . This functional form is similar to a hybrid Ising/XY model [16].

### III. NETWORK AND LEARNING DYNAMICS

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

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

Using this in equation 7 we obtain

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

Where  $\alpha = (N - \lambda(N - 1))/2N$ ,  $\gamma = \lambda/2N$ . To maximize the objective function over the entire input ensemble, at a slower time-scale, we perform gradient descent over the synaptic weights, yielding the learning update rule:

$$\Delta W_{ij} \sim y_i x_j [1 + \beta \cos(\phi_j - \theta_i)] \quad (10)$$

Observe that this is a simple extension of the traditional Hebbian learning rule.

### IV. NETWORK CONFIGURATION

We present here a network to perform dynamical segmentation that implements the dynamics described in the previous section. The activation and phase variables are simply interpreted by 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, which was  $\tau \in [2.0, 2.1]$  msec. The network dynamics determine how the initial amplitudes and phases of the units evolve over time.

The network is designed as follows: **(a)** A bottom layer receiving input from an input signal, and consisting of

dynamical units. The amplitude output of these units is equal to their inputs (the grayscale images depicted in Figure 2), whereas the phase is a function of their natural frequency and feedback interactions with a top layer; **(b)** A top layer consisting of dynamical units that receive input from the bottom layer through feed-forward 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; **(c)** 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.

The network operates in two stages: learning and performance. Only during the learning stage are the feed-forward and feedback connections modified, whereas the inhibitory connections are 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 feed-forward, inhibitory and feedback inputs, weighted by their amplitude and the receiving unit's amplitude, as well as by a non-linear 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 the equations in 9 is the following: **(a)** the effect of feed-forward inputs on the amplitude is stronger for synchronized units, ie when the argument to the cosine term is small; **(b)** excitatory feed-forward and feedback connections are such that units that are simultaneously active tend towards phase synchrony (as the phase is not perturbed if the argument to the sin term is small); and **(c)** inhibitory connections tend towards de-synchronization; at the same time, they have a stronger depressing effect on the amplitude of synchronized units, and correspondingly a weaker effect for de-synchronized units.

The system is organized into two layers as shown in Figure 1. The lower layer consists of 8x8 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.

Figure 2 shows the input images used to test the system. These images are of size 8x8, and possess gray levels in the range 0-255. They represent 16 different 2D objects such as a square, triangle, cross, circle etc.

In order to understand the behavior of this system of equations, we consider Figure 3. As can be seen, the amplitude activity in the network reaches a steady state after approximately 200 iterations, where the updates of Eq. 9, derived from the objective function of Eq. 1 are applied at

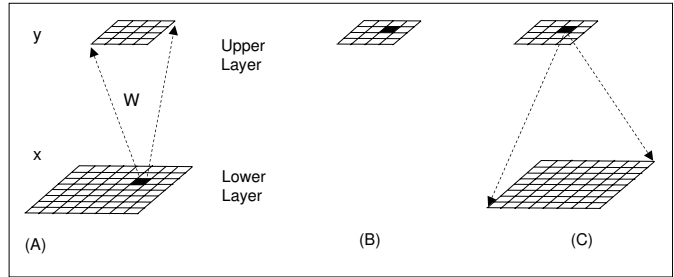


Fig. 1. Illustrating the network connectivity. (A) Feedforward connections. (B) Lateral connections. (C) Feedback connections.

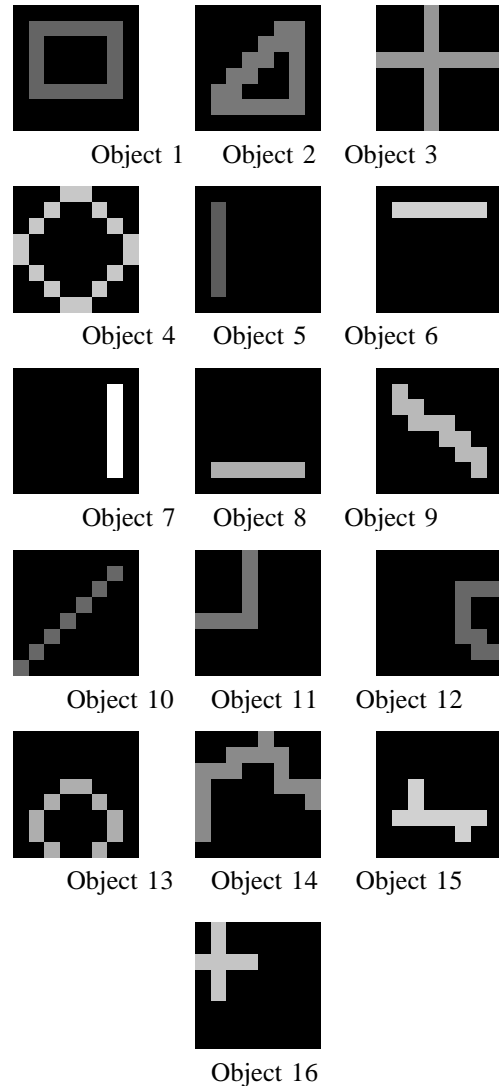


Fig. 2. Input images.

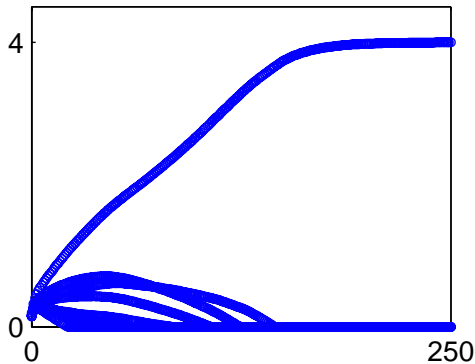


Fig. 3. Amplitude dynamics of elements in the upper layer of the network. For a single input, there are multiple upper layer units active initially, but after settling, a single winner emerges.

each iteration. The learning rule of Eq 10 is applied only after the network amplitude activity has settled down. We choose a nominal settling period of  $T = 200$  iterations. In Section V-B we examine the effect of choosing a given settling time  $T$  on the performance of the network. Note that since the period of oscillation  $\tau \approx 2$  msec, and the integration step to compute the updates is 0.1 msec, this effectively implies that the settling period is approximately 10 cycles of oscillation.

Within an appropriate parameter range, learning leads to a winner-take-all dynamics upon presentation of one of the learned inputs; moreover, when two learned inputs are presented two winners arise, as depicted in Fig. 5. A detailed explanation for this behavior is beyond the scope of this paper, and is provided in [21].

## V. DYNAMICAL SEGMENTATION

The system described in section IV is presented with a randomly chosen image from this set of images. The inputs are pre-processed to convert them to zero mean and unit norm. Upon settling of the transient behavior, which takes 200 iterations, the Hebbian learning rule in Eq. 10 is applied. This process is repeated for 1,000 presentations. The typical behavior of the system is that a single unit in the output layer emerges as a winner for a given input. Furthermore, after the 1,000 trials, a unique winner is associated with each input.

The receptor field for each unit in the upper layer is shown in Figure 4. Each image encodes the feedforward weights  $W_{ij}$  for the  $i^{\text{th}}$  unit in the upper layer. Thus we can see that the majority of receptor fields correspond to the outlines of the objects used for training. This shows that the upper layer has learnt in an unsupervised manner the correct representations for the objects it has been presented with. Though such a capability can be exhibited by traditional neural networks, what is significant is that the current network can also perform segmentation of the inputs, as explained below.

The system is then presented with a superposition of two randomly selected objects from Figure 2. Figure 5 shows

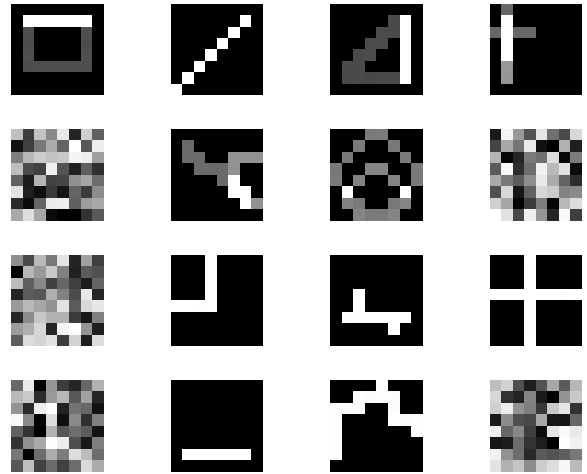


Fig. 4. The receptor field for each unit in the upper layer is plotted as a grayscale image.

resulting dynamics of the network after learning. Two aspects of the system response,  $\mathbf{y}$ , 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 deconvolution accuracy, which is defined as follows. Let unit  $i$  in the upper layer be the winner for an input  $\mathbf{x}_1$ , and let unit  $j$  be the winner for input  $\mathbf{x}_2$ . If units  $i$  and  $j$  in the upper layer are also winners when the input presented is  $\mathbf{x}_1 + \mathbf{x}_2$ , then we say the deconvolution is performed correctly, otherwise not. The ratio of the total number of correctly deconvolved cases to the total number of cases is the deconvolution accuracy.

In our experiment, the deconvolution accuracy was 90% over 100 trials. We used the following parameters to instantiate the model:  $\beta = 0.9$ ,  $\alpha = 0.5$ ,  $\gamma = 0.25$ ; the natural periods ( $\tau = 2\pi/\omega$ ) are drawn uniformly from  $[2, 2.1]$  msec, and learning takes place after 20 real time units, or approximately 10 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 presentation number, and  $T = 2000$ .

The phase behavior of the system can be understood through Figure 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 deconvolution 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  $\theta_i$  and  $\theta_j$  respectively. Consider a unit  $k$  in the lower layer with phase  $\phi_k$ . The behavior of the network is such that the phase of this  $k^{\text{th}}$  unit is usually synchronized with the phase of one of the winners in the upper layer.

Figure 6(a) shows the input resulting from a superposition of objects 1 and 3. Figure 6(b) 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

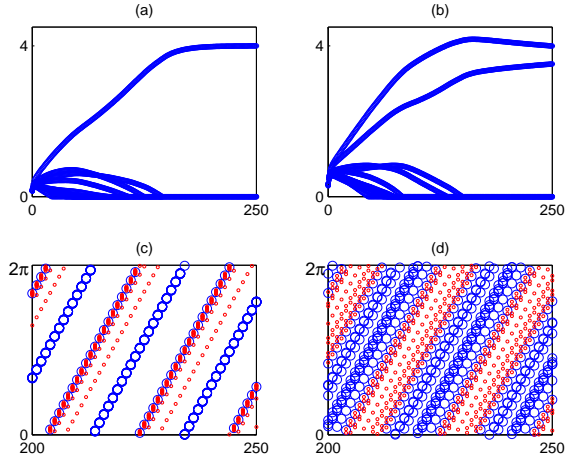


Fig. 5. Behavior of the network after learning; (a) and (c): amplitude and phase response upon presentation of an input from the training ensemble; (b) and (d): response to the presentation of a mixture. For the amplitude, the evolution is shown since the onset of the input. For the phase, only the behavior after convergence is shown. Blue circles correspond to upper layer units, and red ones to lower layer units. Time is in simulation steps.

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 Figure 2. Figure 6(c) and (d) show 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 Figure 6(b). 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  $d = \cos(\theta_i - \phi_k)$  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 anti-synchronization. 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 Figure 6(e) and those synchronized with the second winner are shown in Figure 6(f). Figure 6(e) 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. Similarly, the upper layer winner for object 3 is synchronized with lower layer units that represent object 3.

Similarly, Figure 7(b) 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 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 [22], 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.

The phase relationship between the layers is not always as crisp as indicated in Figure 6. The accuracy of phase segmentation 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. The segmentation accuracy for our experiment was 81% over 100 trials.

The trials were carried out in the following manner. The entire network was randomized, and inputs were presented individually during the training phase. Once the network was trained, its performance for deconvolution and segmentation was measured for 10 pairs of randomly selected inputs. This entire process was repeated 10 times, giving rise to 100 trials.

#### A. Effect of varying $\beta$ on performance

In equation 4 we introduced a parameter  $\beta$  that controls the weight given to the phase update equations of an oscillatory network. When  $\beta = 0$ , this reduces to a traditional neural network without oscillations. We examine the effect of varying the  $\beta$  on the performance of the network, and the results are summarized in Figure 8. The parameter  $\beta$  was varied from  $\beta = 0.1$  to  $\beta = 4.0$ .

The deconvolution 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 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.

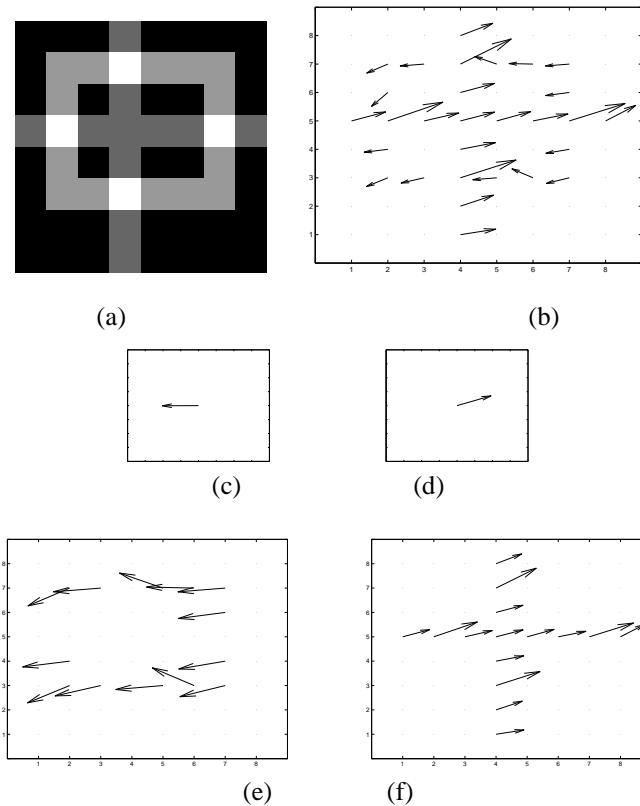


Fig. 6. Illustrating the behavior of phase information. (a) Shows the superposition of objects 1 and 3. (b) shows the activity in the lower layer units, displayed as a vector field. The magnitude of the vector reflects the amount of activity in the unit, and the direction encodes the phase of the unit. (c) shows the phase of the first winner, which is 3.147. (d) shows the phase of the second winner, which is 0.351. (e) shows the units in the lower layer that are synchronized with the first winner. (f) shows the units in the lower layer that are synchronized with the second winner.

### B. Effect of settling time on performance

In Section III 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 Eq 10. We examined the effect of varying the settling time  $T$  on the performance of the network, and the results are summarized in Figure 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 increase, 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

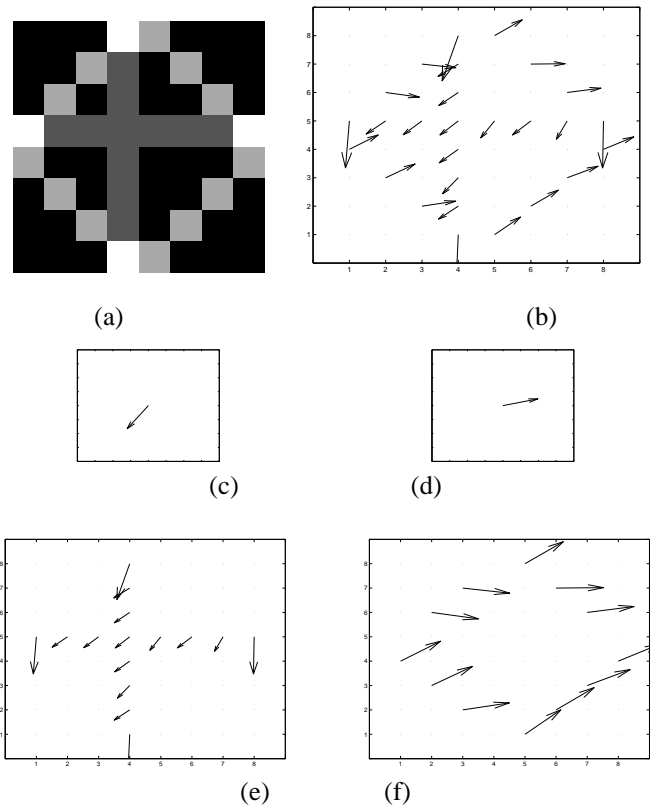
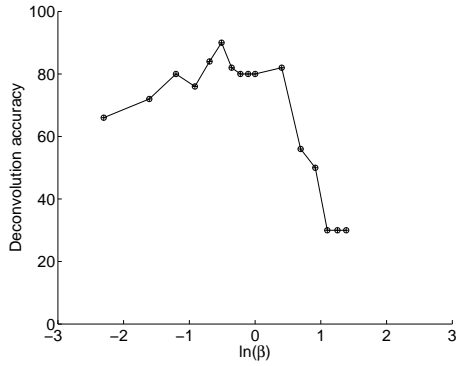


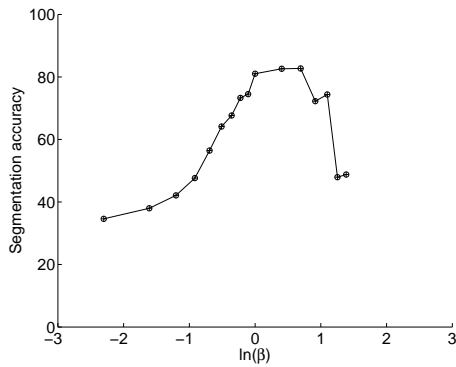
Fig. 7. Illustrating the behavior of phase information. (a) Shows the superposition of objects 3 and 4. (b) shows the activity in the lower layer units, displayed as a vector field. The magnitude of the vector reflects the amount of activity in the unit, and the direction encodes the phase of the unit. (c) shows the phase of the first winner, which is 4.087. (d) shows the phase of the second winner, which is 0.241. (e) shows the units in the lower layer that are synchronized with the first winner. (f) shows the units in the lower layer that are synchronized with the second winner.

noise-free input, as shown in Figure 2. Then, increasing amounts of additive noise were added to the inputs, and the performance of the network was measured. The noise at a given image pixel is generated by multiplying a uniform random number in the range  $[-1, 1]$  with a noise fraction  $f$ . Values of  $f$  ranged from 3% to 50% of the maximum input value. The inputs were then re-normalized to the range  $[0,1]$ . As done earlier, 100 trials were used to generate each point in the graph.

Figure 10 shows the deconvolution and segmentation accuracy of the network as a function of the percentage of noise. 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 shows less degradation in the presence of increasing noise. The reason for this is that the segmentation is measured only for cases that exhibit correct deconvolution. Thus, given that two inputs are correctly deconvolved, the constituents of those inputs are typically well segmented, with an accuracy of greater than 75 % in most cases.



(a)



(b)

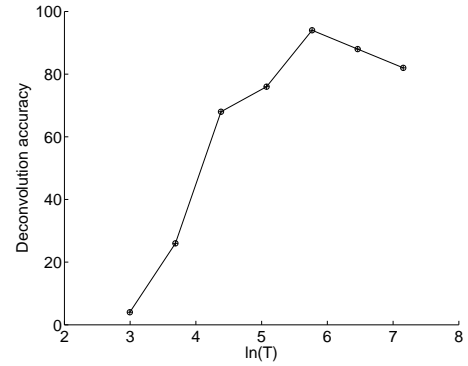
Fig. 8. (a) The deconvolution accuracy of the network as a function of the parameter  $\beta$ . The natural logarithm of  $\beta$  is plotted on the x-axis. The range of  $\beta$  is 0.1 to 4.0 (b) The segmentation accuracy of the network as a function of  $\beta$ .

## VI. DISCUSSION

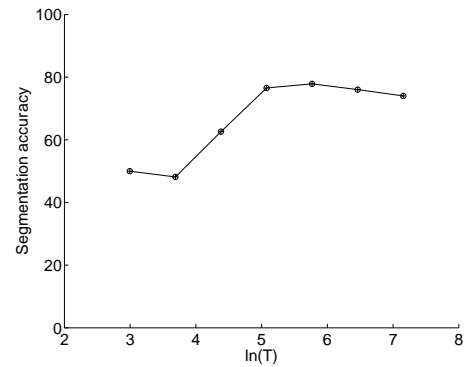
We have shown that the approach derived from the simple objective function in Section II can : (a) provide very simple neural dynamical and learning rules (b) achieve good computational results in solving the segmentation problem, (c) provide a reasonable biological interpretation of segmentation, including 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 [17]. 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 [18].

We have investigated the relationship between the model presented in this paper and biologically observed oscillations. However this subject is outside the scope of the current paper, and details can be found in [21].

While building on previous work, the system presented in this paper significantly improves existing formulations by



(a)



(b)

Fig. 9. (a) The deconvolution 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 to 1280. (b) The segmentation accuracy of the network as a function of  $T$ .

simplifying the network architecture required, presenting a simple objective function to understand the system behavior, and demonstrating the ability to solve deconvolution and segmentation problems in an unsupervised manner. The original network proposed by Malsburg and Schneider [5], [6] 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. Firstly, a global inhibitory neuron is required. Our model overcomes this restriction and spreads inhibition across the entire network, which is more biologically plausible. Secondly, learning in their model requires a combination of short-term and long-term synaptic modification, which in our model is reduced to a single generic rule. Thirdly, the test cases used in their model did not involve any overlap amongst the spectral inputs to be separated. Our model allows complete overlap, and shows that successful separation and segmentation is still possible.

Buhman and Malsburg [6] explicitly introduced oscillatory units into the model, but their model suffers from the earlier noted shortcoming in that the presence of a global inhibitory unit is required. The subsequent work of Chen, Wang and Liu

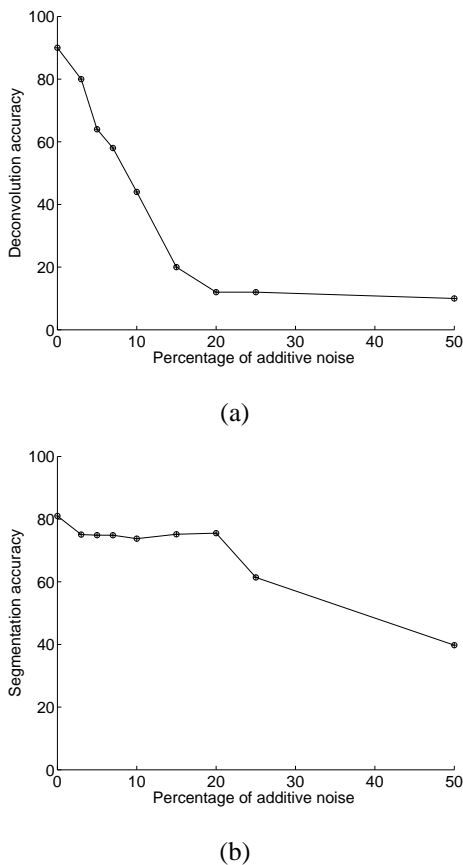


Fig. 10. (a) The deconvolution accuracy of the network as a function of additive noise. (b) The segmentation accuracy of the network as a function of additive noise.

[8], and Wang and Liu [8] offer enhancements of the original model, but maintains the essential aspect of utilizing a global inhibitor.

The work of Izhikevich [12] is mainly theoretical, and does not present any specific methodology to address the problem of segmentation. Hoppensteadt and Izhikevich [13] 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* [14] 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 back-propagation learning.

Finally, Zemel *et al.*[19], used a similar approach in which units in their network possess both an amplitude and phase. They employ complex weights, which results in increased computational overhead and costs. The model presented in our paper is simpler as it uses real valued weights. The learning algorithm used in [19] was derived through trial and

error, rather than an explicit optimization approach as done in this paper. Furthermore, their approach requires supervised training, as they set the target phases of different contours in the training images to specific values. Our approach on the other hand, does not require the setting of target phase values during training.

In summary, our model presents many interesting novel features with rich potential for formalization and generalization. There are several fruitful extensions of our model, which we have begun pursuing [17], [18].

## REFERENCES

- [1] A.J. Bell and T.J. Sejnowski (1995) An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, **7**:1129-1159.
- [2] S. Ullman, M. Vidal-Naquet, E. Sali E. (2002) Visual features of intermediate complexity and their use in classification. *Nature Neuroscience* **5**(7):682-7.
- [3] C.M Gray, P. König, A.K. Engel and W. Singer (1989) Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. *Nature*, **338**(6213):334-337.
- [4] E. Rodriguez, N. George, J.-P. Lachaux, J. Martinerie, B. Renault and F.J. Varela (1999) Perception's shadow: long-distance synchronization of human brain activity. *Nature*, **397**(6718):430-433.
- [5] Ch. von der Malsburg and W. Schneider (1986) A neural cocktail-party processor. *Biol. Cybern.*, **54**(1):29-40.
- [6] J. Buhmann and C. Von Der Malsburg (1991) Sensory segmentation by neural oscillators. *International Joint Conference on Neural Networks*, Part II, pp. 603-607.
- [7] K. Chen and D. Wang and X. Liu (2000) Weight Adaptation and Oscillatory Correlation for Image Segmentation. *IEEE Transactions on Neural Networks*, **11**(5):1106-1123.
- [8] D. L. Wang and X. Liu (2002) Scene analysis by integrating primitive segmentation and associative memory. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, **32**(3):254-268.
- [9] C. Von der Malsburg (1999) The what and why of binding: The modeler's perspective, *Neuron*, pp. 95-104.
- [10] F. Rosenblatt (1962) *Principles of Neurodynamics: Perception and the Theory of Brain Mechanisms*, Washington: Spartan Books.
- [11] F.C. Hoppensteadt and E.M. Izhikevich (1997) *Weakly connected neural networks*, Springer.
- [12] E.M. Izhikevich (1999) Weakly Pulse-Coupled Oscillators, FM Interactions, Synchronization, and Oscillatory Associative Memory. *IEEE Transactions on Neural Networks*, **10**(3):508-526.
- [13] F.C. Hoppensteadt and E.M. Izhikevich (2000) Pattern Recognition Via Synchronization in Phase-Locked Loop Neural Networks. *IEEE Transactions on Neural Networks*, **11**(3):734.
- [14] H. Sun, L. Liu and A. Guo (1999) A Neurocomputational Model of Figure-Ground Discrimination and Target Tracking. *IEEE Transactions on Neural Networks*, **10**(4):860-884.
- [15] B.A. Olshausen & D.J. Fields (1996) Natural image statistical and efficient coding. *Network: Computation in Neural Systems*, **7**, 333-339.
- [16] J.M. Kosterlitz & D.J. Thouless (1973) Ordering, metastability and phase transitions in two-dimensional systems. *J. Phys. C* **6**, 1181-1203.
- [17] Y. Liu *et al.* (2006) Inference and segmentation in cortical processing. *Proc. Human vision and electronic imaging*, Electronic Imaging 2006 Conference, SPIE, to appear.
- [18] A. R. Rao, G. A. Cecchi, C. C. Peck and J. R. Kozloski (2006) Translation invariance in a network of oscillatory units *Proc. Electronic Imaging 2006 Conference*, SPIE, to appear.
- [19] R. S. Zemel, C. K. Williams & M. C. Mozel (1995) Lending direction to neural networks. *Neural Networks*, **8**(4), 503-512.
- [20] H. R. Wilson & J. D. Cowan. (1972) Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical Journal*, **12**:1-24.
- [21] A. R. Rao, G. A. Cecchi, C. C. Peck and J. R. Kozloski, Unsupervised segmentation with dynamical units, *IBM Research Technical Report*, <http://domino.watson.ibm.com/library/cyberdig.nsf/Home>, 2005.
- [22] R. Quiroga *et al*, Invariant visual representation by single neurons in the human brain, *Nature*, June 2005, **435**(2)3, pp. 1102-7.