RC23838 (W0512-105) December 21, 2005
Computer Science

IBM Research Report

Unsupervised Segmentation with Dynamical Units

A. Ravishankar Rao, Guillermo A. Cecchi,
Charles C. Peck, James R. Kozloski
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

LIMITED DISTRIBUTION NOTICE: This report has been submitted for publication outside of IBM and will probably be copyrighted if accepted for publication. Ithas been issued as a Research Report for early dissemination of its contents. In view of the transfer of copyright to the outside publisher, its distributionoutside of IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or legally obtained copies of the article (e.g, payment of royalties). Copies may be requested from IBM T. J. Watson Research Center , P. O. Box 218, Yorktown Heights, NY 10598 USA (email: reports@us.ibm.com). Some reports are available on the internet at http://domino.watson.ibm.com/library/CyberDig.nsf/home

Unsupervised segmentation with dynamical units

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

***Abstract*— We present a novel network to deconvolve mixtures of inputs that have been previously learned, but more importantly, to segment 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 deconvolution is determined by the amplitude of an output layer, and segmentation by phase similarity between input and output layer units. Learning is unsupervised and based on Hebbian update, and the architecture is very simple. Moreover, 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 attacked 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. Besides the need for a global inhibitory unit, this network cannot 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 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, an issue extensively discussed in the neuroscience literature. The idea, which can be traced back to Rosenblatt, states that neural networks do not have the capacity to encode superimposed inputs (Rosenblatt's superposition catastrophe [9]), but which can be achieved by a variable independent from the amplitude, ie. the phase of ongoing oscillations.

Correspondence should be addressed to: gcecchi@us.ibm.com
A.R.R., G.A.C., C.C.P. and J.K. are at IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA

II. Segmentation

We will first introduce an objective function for vector quantization or sparse representation (cf. [18]), in which it is assumed that 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, $y_i \geq 0 \forall i$. We write then:

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

where $\mathcal{E}$ represents the input ensemble. The first term is related to faithfulness of representation, by rewarding 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 vectors. The last term is defined as:

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

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:

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

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. 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. III (see Appendix II).

We will now introduce a generalization of the objective function that leads to efficient segmentation of the inputs:

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

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

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

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

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

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

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

where $\Psi_{nm} = \theta_n - \phi_m$, $\Phi_{nm} = \theta_n - \theta_m$, $\alpha = \lambda(2-N)/2N$, $\gamma = \lambda/N$. This functional form makes evident that we are in the presence of a hybrid model of an Ising system with an XY model.

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:

$$\Delta y_n \sim \frac{\partial E_S}{\partial y_n} \qquad \Delta \theta_n \sim \frac{1}{y_n} \frac{\partial E_S}{\partial \theta_n} \tag{8}$$

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

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

$$\Delta \theta_n \sim \sum_j W_{nj} x_j \sin(\phi_j - \theta_n) - \gamma \sum_k y_k \sin(\theta_k - \theta_n) \tag{10}$$

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

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

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

Observe that this 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 is derived from an ongoing oscillation whose natural frequency (ie. in the absence of inputs) is determined by each unit, although within a small range for the entire ensemble.

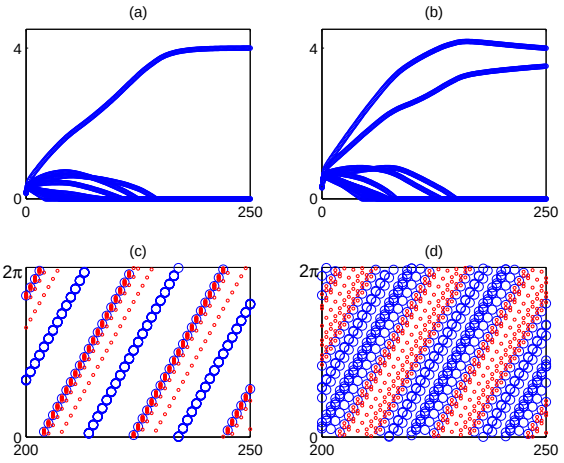

Fig. 1. Behavior of the network after learning; (a) and (c): amplitude and phase response upon presentation of an input from the training ensemble; (b) and (d): response to the presentation of a mixture. For the amplitude, the evolution is shown since input onset; for the phase, only the behavior after convergence is shown. Blue circles correspond to upper layer units, and red ones to lower layer units. Time is in simulation steps.

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 only a function of their inputs, 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 these equations is the following: (**a**) the effect of feed-forward inputs on the amplitude is stronger for synchronized units; (**b**) excitatory feed-forward and feedback connections are such that units that are simultaneously active tend towards phase synchrony; and (**c**) inhibitory connections tend towards de-synchronization; at the same time, they have

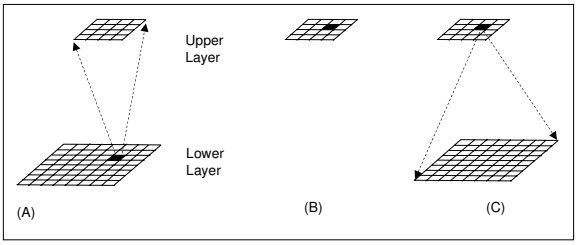

Fig. 2. Illustrating the network connectivity. (A) shows feedforward connections. (B) shows lateral connections (C) shows feedback connections.

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 2. 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 4x4 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 3 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.

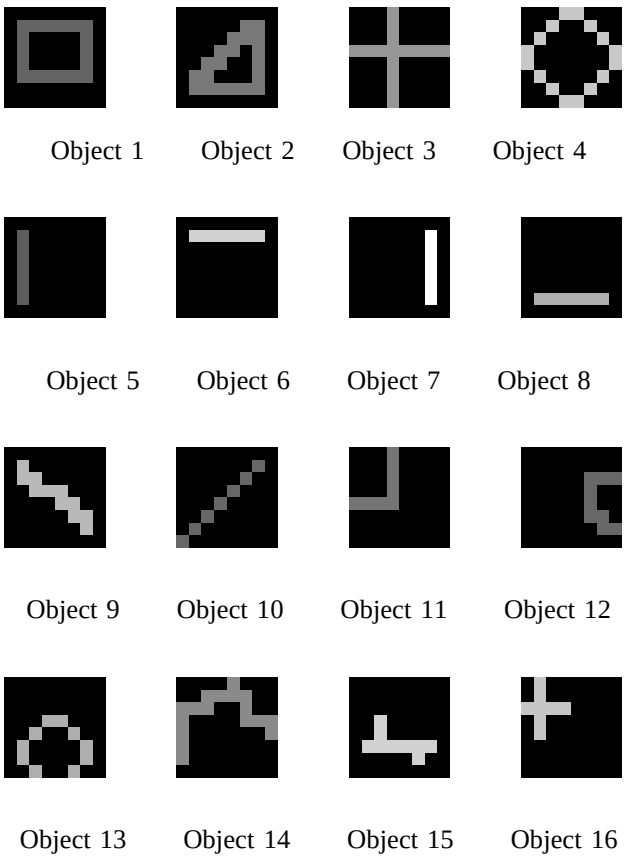

Fig. 3. Input images.

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 400 iterations, the Hebbian learning rule in Eq. 12 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. Furthermore, after the $1,000$ trials, a unique winner is associated with each input.

The system is then presented with a superposition of two randomly selected objects from Figure 3. 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.

We used the following parameters to instantiate the model: $\beta = 0.5, \gamma_A = 0.25, \gamma_P = 0.25,\ \mu = 0.5,\ \gamma_r = 0.1,\ \gamma_\theta = 0.3,\ \beta_f = 0.1\ \forall f,\ \sigma_\alpha = 2\pi/32\ \forall \alpha$ The natural periods ($\tau = 2\pi/\omega$) are drawn uniformly from $[2, 2.1]$, and learning takes place after 40 real time units, or approximately 20 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 $T = 2000$.

In our experiment, the deconvolution accuracy was 90% over 100 trials.

The phase behavior of the system can be understood through Figure 4. 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 θ_{2i} and θ_{2j} respectively. Consider a unit k in the lower layer with phase θ_{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.

Figure 4(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 Figure 3. Figure 4(b) and (c) 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 4(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 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 4(d) and those synchronized with the second winner are shown in Figure 4(e). Figure 4(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. Similarly, the upper layer winner for object 3 is synchronized with lower layer units that represent object 3.

Similarly, Figure 5(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 lever 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.

The phase relationship between the layers is not always as crisp as indicated in Figure 4. 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.

VI. Neural Dynamics and Biological Constraints

It is possible to map the abstract network equations presented in the previous section 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

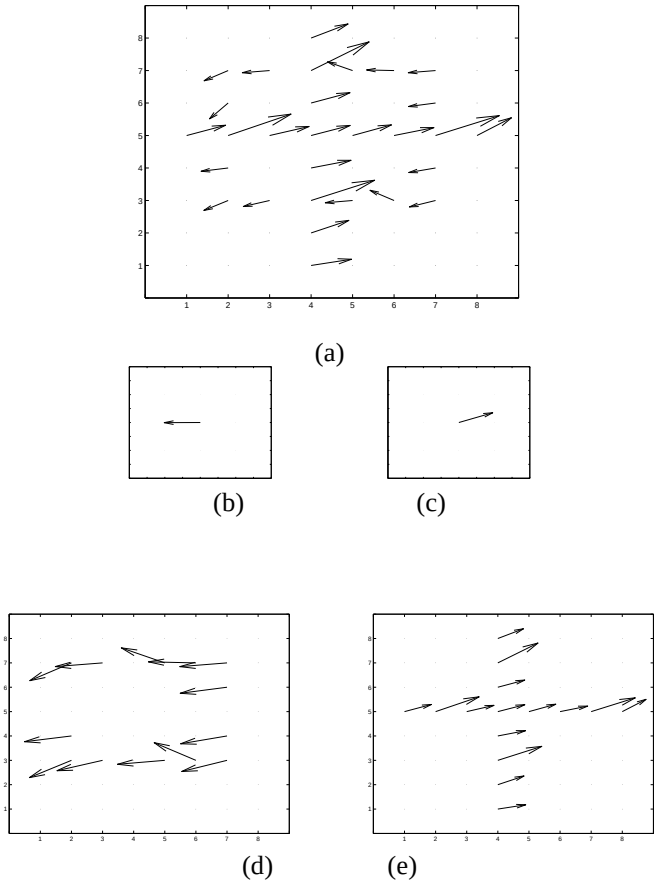

Fig. 4. Illustrating the behavior of phase information. (a) shows the activity in the lower layer units, displayed as a vector field. The magnitude of the vector reflects the amount of activity in the unit, and the direction encodes the phase of the unit. (b) shows the phase of the first winner, which is 3.147. (c) shows the phase of the second winner, which is 0.351. (d) shows the units in the lower layer that are synchronized with the first winner. (e) shows the units in the lower layer that are synchronized with the second winner.

Cowan [19]:

$$\tau \dot{E}_i = -E_i + S(\sum_j [w_{ij}^{EE} E_j - w_{ij}^{IE} I_i] + U_i) \tag{13}$$

$$\tau \dot{I}_i = -I_i + S(\sum_j [w_{ij}^{EI} E_j - w_{ij}^{II} I_i] + V_i) \tag{14}$$

where E, I are the excitatory and inhibitory local populations, U, V external inputs, and $S(\cdot)$ a monotonic function. The presence of generalized oscillations in the range of 40Hz 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 [11] 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

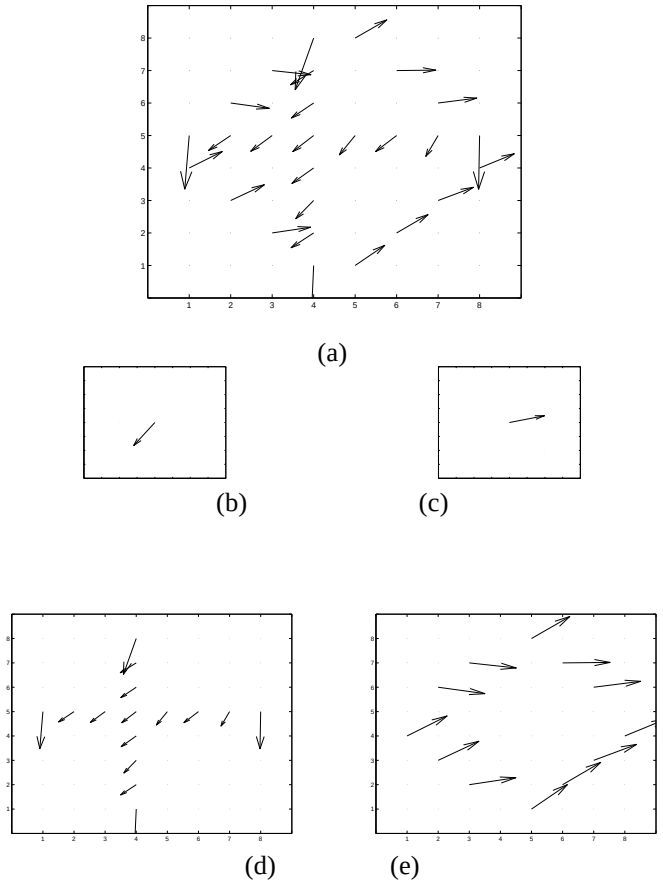

Fig. 5. Illustrating the behavior of phase information. (a) shows the activity in the lower layer units, displayed as a vector field. The magnitude of the vector reflects the amount of activity in the unit, and the direction encodes the phase of the unit. (b) shows the phase of the first winner, which is 4.087. (c) shows the phase of the second winner, which is 0.241. (d) shows the units in the lower layer that are synchronized with the first winner. (e) shows the units in the lower layer that are synchronized with the second winner.

gradient ascent of the objective function is negligible. The additional component of the on-line energy change is:

$$\Delta E_\omega \sim \sum_{i\in\mathbf{p},j\in\mathbf{q}} \omega_i W_{ij} x_i y_j \frac{\partial E}{\partial \phi_i} + \sum_{m\in\mathbf{q},n\in\mathbf{p}} \omega_m y_m W_{mn} x_n \frac{\partial E}{\partial \theta_m} \quad (15)$$

$$\Delta E_\omega \approx \Omega \sum_{n,m} x_m W_{nm} y_n \left(\sin\Phi_{nm} + \sin\Phi_{mn}\right) - \Omega\gamma \sum_{n,m} y_n y_m \sin\Theta_{nm} = 0 \quad (16)$$

The interaction terms in the update equations can be understood also in biological terms. We will to 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 will 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 Φ is the (slowly varying) phase difference between the ensemble and the forcing. We can compute then the average change over a cycle as:

$$\langle \Delta r \rangle \sim \langle f \rangle + k_0 \langle rf \rangle \quad (17)$$

The rational is as follows: for an ensemble of spiking or threshold elements with leakage, to first order the change in rate is proportional to the average forcing amplitude, and to second order to the coincidence between the forcing and the proximity to threshold, i.e. 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:

$$\langle \Delta\theta \rangle \sim \langle r(\frac{df}{dt} - \frac{dr}{dt}) \rangle \approx \omega \langle r(\frac{df}{d\theta} - \frac{dr}{d\theta}) \rangle \quad (18)$$

From Eq. 17 we obtain $\langle \Delta r \rangle \sim 1 + \beta \cos\Phi$, where $\beta = k_0/(1+k_0)$ and from Eq. 18 $\langle \Delta\theta \rangle \sim \sin\Phi$.

Finally, for biological considerations, one can interpret that in the last equation (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^{FB}_{nj}. This will indeed be the case: given that the Hebbian learning rule (Eq. 12) is symmetric in its arguments, $W_{nm} \underset{t\to\infty}{\longrightarrow} W^{FB}_{mn}$.

VII. Discussion

The original network proposed by Malsburg and Shneider [5], [6] has been influential in advancing 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 earlier noted shortcoming in that the presence of a global inhibitory unit is required. The subsequent work of Chen, Wang and Liu [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.

In summary, although much work remains, our model presents many interesting novel features with a rich potential for formalization and generalization.

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 Eq. II 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 $\mathbf{W}^{(t+1)} = \mathbf{v}/|\mathbf{v}|, \ \mathbf{v} = \mu \mathbf{y}\mathbf{x}^T + (1-\mu)\mathbf{W}^{(n)}$.

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

APPENDIX II
WINNER-TAKE-ALL DYNAMICS

The dynamics derived from Eq. II result in the emergence of a unique winner after learning, and at least two winners when two inputs are superimposed, as shown in Fig. III. 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's 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 $x_n - \lambda x_M \ \forall n \neq M$, 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 $(I^{(1)}_{M_1} + I^{(1)}_{M_2})/(1+\lambda) + (I^{(2)}_{M_1} + I^{(2)}_{M_2})/(1+\lambda) > I^{(1)}_n/\lambda + I^{(2)}_n/\lambda$.

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] F. Rosenblatt (1962) *Principles of Neurodynamics: Perception and the Theory of Brain Mechanisms*, Washington: Spartan Books.

[10] The simulation parameters are: $\mu = 0.5, \ \gamma_r = 0.1, \ \gamma_\theta = 0.3, \ \beta_f = 0.1 \ \forall f, \ \sigma_\alpha = 2\pi/32 \ \forall \alpha$ The natural periods ($\tau = 2\pi/\omega$) are drawn uniformly from $[2, 2.1]$, and learning takes place after 40 real time units, or approximately 20 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 $T = 2000$.

[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] H. Haken (2002) *Brain dynamics: synchronization and activity patterns in pulse-coupled neural nets with delays and noise.* Berlin: Springer-Verlag.

[16] A. Winfree (1980) *The geometry of biological time.* New York: Springer-Verlag.

[17] Y. Kuramoto (1984) *Chemical oscillations, waves, and turbulence.* Berlin: Springer-Verlag.

[18] B.A. Olshausen & D.J. Fields (1996) Natural image statistical and efficient coding. *Network: Computation in Neural Sustems*, **7**, 333-339.

[19] H.R. Wilson & J.D. Cowan. (1972) Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical Journal*, **12**:1-24.