

Translation invariance in a network of oscillatory units

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

IBM T.J. Watson Research Center

Yorktown Heights, NY 10598

{ravirao,gcecchi,cpeck,kozloski}@us.ibm.com

ABSTRACT

One of the important features of the human visual system is that it is able to recognize objects in a scale and translational invariant manner. However, achieving this desirable behavior through biologically realistic networks is a challenge.

Neurons may be modeled as oscillatory dynamical units. It is possible for a network of these units to exhibit synchronized oscillations under the right conditions. The synchronization of neuronal firing patterns has been suggested as a possible solution to the binding problem (where a biological mechanism is sought to explain how features that represent an object can be scattered across a network, and yet be unified). Networks consisting of such oscillatory units have been applied to solve the signal deconvolution or blind source separation problems. However, the use of the same network to achieve properties that the visual system exhibits, such as scale and translational invariance have not been fully explored.

Some approaches investigated in the literature (Wallis 1996) involve the use of non-oscillatory elements that are arranged in a hierarchy of layers. The objects presented are allowed to move, and the network utilizes a trace learning rule, where a time averaged value of an output value is used to perform Hebbian learning with respect to the input value. This is a modification of the standard Hebbian learning rule, which typically uses instantaneous values of the input and output.

In this paper we present a network of oscillatory amplitude-phase units connected in two layers. The types of connections include feedforward, feedback and lateral. The network consists of amplitude-phase units that can exhibit synchronized oscillations. We have previously shown that such a network can segment the components of each input object that most contribute to its classification. Learning is unsupervised and based on a Hebbian update, and the architecture is very simple.

We extend the ability of this network to address the problem of translational invariance. We show that by a specific treatment of the phase values of the output layer, limited translational invariance is achieved. The scheme used in training is as follows. The network is presented with an input, which then moves. During the motion the amplitude and phase of the upper layer units is not reset, but continues with the past value before the introduction of the object in the new position. Only the input layer is changed instantaneously to reflect the moving object. This is a promising result as it uses the same framework of oscillatory units, and introduces motion to achieve translational invariance.

Keywords: synchronization, motion, object representation, classification

1. INTRODUCTION

One of the problems that the human visual system has to solve is that of object classification. This starts with retinal input, which can be considered to consist of isolated pixels of activity, and ends with a high-level representation at an area such as IT (inferior temporal cortex), where a small set of neurons, or even a single neuron encodes a particular object such as a square or a face. In addition to solving this problem, the visual system must also address the issue of 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).

The problem of segmentation has been attacked more effectively with non-neural approaches.¹ Indeed, it is extremely difficult for a traditional computational model of a neural network to solve this problem, due to the superposition catastrophe as Rosenblatt observed.² The early efforts in computational neural networks ignored the temporal dynamical aspect of communication between real neurons. These temporal dynamics provide additional information separate from the amplitude of the neural signals, that can then be used to overcome the superposition catastrophe. Research by Singer³ and Varela⁴ uncovered biological evidence for the role of synchronization in neural responses to several motor and cognitive tasks, and in particular in perceptual recognition.

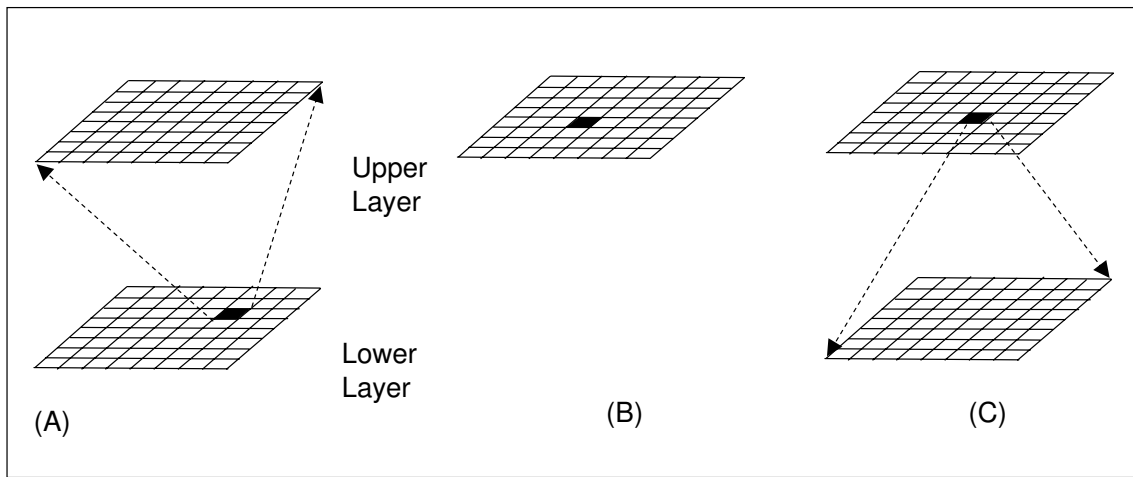


Figure 1. Illustrating the network connectivity. (A) shows feedforward connections. (B) shows lateral connections (C) shows feedback connections.

Malsburg and Shneider were among the first to propose the use of synchronization to perform segmentation of input 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. Several efforts have been undertaken⁶⁻⁸ to build on the early work of Malsburg and Shneider.⁵

Another important problem that the human visual system has to solve is to build invariant representations of objects. Some invariances of interest are translation and scale invariance. The advantage in building invariant representations is that they overcome the combinatorial explosion problem: say if every translated version of an object had its own unique representation, there would be too many combinations that the visual system would have to keep track of, especially when multiple objects are considered. Some research in using a neural-network architecture to solve translation invariant object representation is reported by Lades *et al.*⁹

A particularly challenging and open problem is to start with a network of oscillatory units (along the lines as proposed by Malsburg and Shneider⁵) and use it to perform both segmentation of inputs as well as achieve translationally invariant representations of visual objects. The research presented in this paper is an early attempt to address this problem.

2. BACKGROUND

In this section we describe a computational model using a network of oscillatory units. This network forms the basis for our investigation of translation invariance.

2.1. A network of dynamical units

In previous work,¹⁰ we investigated a network of dynamical units, where each unit is an oscillator characterized by an amplitude, a phase, and a frequency. The network is organized into two layers as shown in Figure 1. Inputs from the lower layer are denoted by \mathbf{x} , and the output of the upper layer by \mathbf{y} .

Each unit in the upper and lower layers is an oscillator, which possesses an amplitude, frequency and phase. 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 behavior of the above network can be derived by proposing an objective function for vector quantization or sparse representation (cf.¹¹). Consider inputs \mathbf{x} drawn from an input ensemble, which are then represented by an output layer \mathbf{y} through synaptic weights $\{W_{ij}\}$, such that a non-negativity condition is imposed on the output layer, $y_i \geq 0 \forall i$. A learning rule can be derived for the weights $\{W_{ij}\}$ such that the output set \mathbf{y} is sparse. The details of this derivation are beyond the scope of this paper, but the resulting network dynamics are captured as follows. Let the n^{th} unit in the lower layer have an amplitude x_n , a phase ϕ_n , and a period of oscillation τ_{xn} . The n^{th} unit in the upper layer has an amplitude y_n , a phase θ_n and a period of oscillation τ_{yn} . The period of oscillation of these units are drawn randomly from the range [1.9,2.1].

In the following equations, the quantity Δt represents the integration step used to approximate a temporal derivative.

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

$$\begin{aligned} \frac{\Delta \theta_n}{\Delta t} &\sim \sum_j W_{nj} x_j \sin(\phi_j - \theta_n) \\ &- \gamma \sum_k y_k \sin(\theta_k - \theta_n) \end{aligned} \quad (2)$$

$$\frac{\Delta \phi_n}{\Delta t} \sim \sum_j W_{jn} y_j \sin(\theta_j - \phi_n) \quad (3)$$

The learning rule utilized is

$$\Delta W_{ij} \sim y_i x_j [1 + \cos(\phi_j - \theta_i)] \quad (4)$$

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

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 a stronger depressing effect on the amplitude of synchronized units, and correspondingly a weaker effect for de-synchronized units.

When performing a simulation, the above equations are calculated by using an integration time step of $\Delta t = 0.05$. For instance,

$$y_n(t+1) = y_n(t) + \Delta t \left\{ \sum_j W_{nj} x_j [1 + \cos(\phi_j - \theta_n)] - \alpha y_n(t) - \gamma \sum_k y_k [1 + \cos(\theta_k - \theta_n)] \right\} \quad (5)$$

and similarly for $\Delta \phi_n$ and $\Delta \theta_n$. The Hebbian learning rule is calculated as

$$W_{ij}(t+1) = W_{ij}(t) + \epsilon y_i x_j [1 + \cos(\phi_j - \theta_i)] \quad (6)$$

where ϵ is the learning rate.

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.

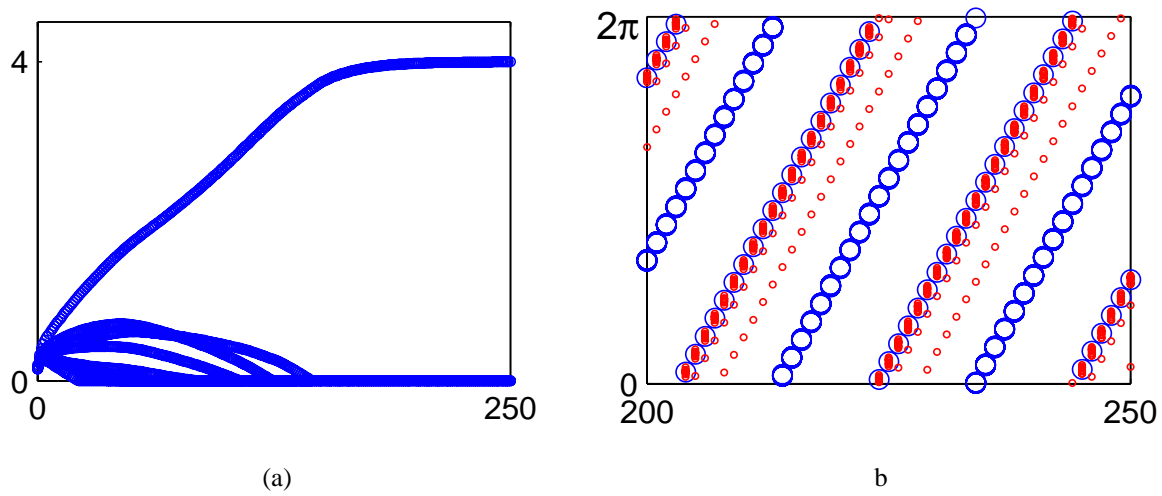


Figure 2. Behavior of the network after learning:(a) amplitude response upon presentation of an input from the training ensemble. The x axis shows the iteration number, and the y axis the amplitude of the units in the upper layer. (b)The phase response, shown after convergence. Blue circles correspond to upper layer units, and red ones to lower layer units. Time is in simulation steps.

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 values provide information about where that object is localized in the lower layer. This behavior could form a foundation to tackle the binding problem.¹² The phase behavior of the system is investigated in¹⁰ and is outside the scope of this paper.

Figure 2.1 shows the system behavior after it undergoes training according to equation 3. The upper layer essentially acts as a vector quantizer, and classifies the inputs it receives such that a unique winner in the upper layer represents a specific input. The phase information displayed in Figure 2.1(b) shows that the units in the two layers can get synchronized with each other. This synchrony aids in segmentation and deconvolution, as shown in.¹⁰

The question we address in this paper is whether such a network can exhibit invariant representations of objects. We develop an approach to answer this question by considering the requirements that this objective imposes, and designing a methodology to satisfy these requirements.

2.2. Requirements on a dynamic network that achieves translation invariance

We consider a few requirements on a dynamic network that achieves translation invariance, some of which are derived from biological considerations.

1. The network must learn in a self-organized manner. This means that supervised classification cannot be used. This requirement arises from observations in primates where the early visual system matures by learning through self-organizing mechanisms.¹³
2. Translated versions of a given object should have the same representation as the given object. This requirement is derived from Occam's razor, in that the smallest number of symbols will be thus required for encoding translated versions of an object. An additional benefit of this requirement is that it addresses the issue of combinatorial explosion, as the organism can unify its behavior to multiple translations of the same object.

Another way of stating this requirement is to posit that the representation of a static object should match the representation of the object as it undergoes translation.

3. Unique input objects should have unique representations in the system. In other words, unique inputs x should result in unique outputs y .

4. Ideally, the output y should be sparse, so that only a few output units encode a given input x , with the extreme being that a single unit y_i encodes an input vector x . This corresponds to a winner-take-all situation.

The solution we propose is based on observations of the biological processing of visual motion. It is likely that translational invariance is an outcome of observing visual objects undergoing motion. The intuition behind our approach is the following. If a system learns to categorize static objects first, and then observes these objects in motion, it should be possible for the system to associate the displaced objects with the objects in their original position. Thus, with an appropriate learning rule and training regimen, it should be possible to associate translated versions of an object with the original object. In this paper we restrict our attention to simple linear translations. More complex forms of motion such as rotation or random walks will be studied in forthcoming papers.

3. METHODS

Here we describe a system that addresses the requirements specified in section 2.2. We extend the ability of the network described in 3.1 to address the problem of translational invariance. We show that by a specific treatment of the phase values of the output layer, limited translational invariance is achieved. The scheme used in training is as follows. The network is presented with an input, which then moves. During the motion the amplitude and phase of the upper layer units is not reset, but continues with the past value before the introduction of the object in the new position. Only the input layer is changed instantaneously to reflect the moving object. This is a promising result as it uses the same framework of oscillatory units, and introduces motion to achieve translational invariance.

There appears to be some biological grounding for this type of computation, as there are microsaccades during which no phase resetting occurs. It is only when a completely new saccade is undertaken, while looking at a different visual field, that phase resetting occurs. This is discussed in more detail in section 5.

3.1. Network configuration

The network described in Figure 1 is used.

3.2. Input presentation

Figure 3 shows the input images used to test the system. These images are bi-level, and of size 8x8. They represent four different 2D objects. Each input object undergoes translation, to produce different representations of the same object. In our experiments we used translations in the horizontal direction (along the x-axis).

3.3. Learning

Two stages of learning were employed. The first stage involves 2000 presentations, where each presentation randomly selects an object. This object is presented statically, at a fixed position, as shown in Figure 3, column 1. All the phases in the system, θ_i and ϕ_i are set to 0. The activity in the upper layer of the network is allowed to stabilize after 100 iterations, and the Hebbian phase-dependent learning rule of equation 6 is applied. Only the feedforward and feedback weights are learnt, and the lateral weights are left unchanged. The weights are normalized after each update.

The second stage also involves 2000 presentations. In each presentation, an object is selected at random, and a direction of motion (positive or negative) is selected at random. This then determines the sequence of inputs that are presented, e.g. object 1 translated to the right, which would involve object 1 presented at its fixed position first, followed by a version translated 1 pixel to the right, then a version translated 2 pixels to the right and so on. In our experiments, a maximum of a 3 pixel translation was used. During the successive presentation of inputs in the second stage, the network dynamics are changed in the following fashion.

First, the phase values θ_i in the upper layer are not reset after the object is translated, but rather continue to evolve with their old value, and the value of the inputs received. Second, a moving average \tilde{y} is calculated as follows.

$$\tilde{y}(t+1) = \mu y(t) + (1-\mu)\tilde{y}(t) \quad (7)$$

Here, \tilde{y} denotes a moving average of the value y .

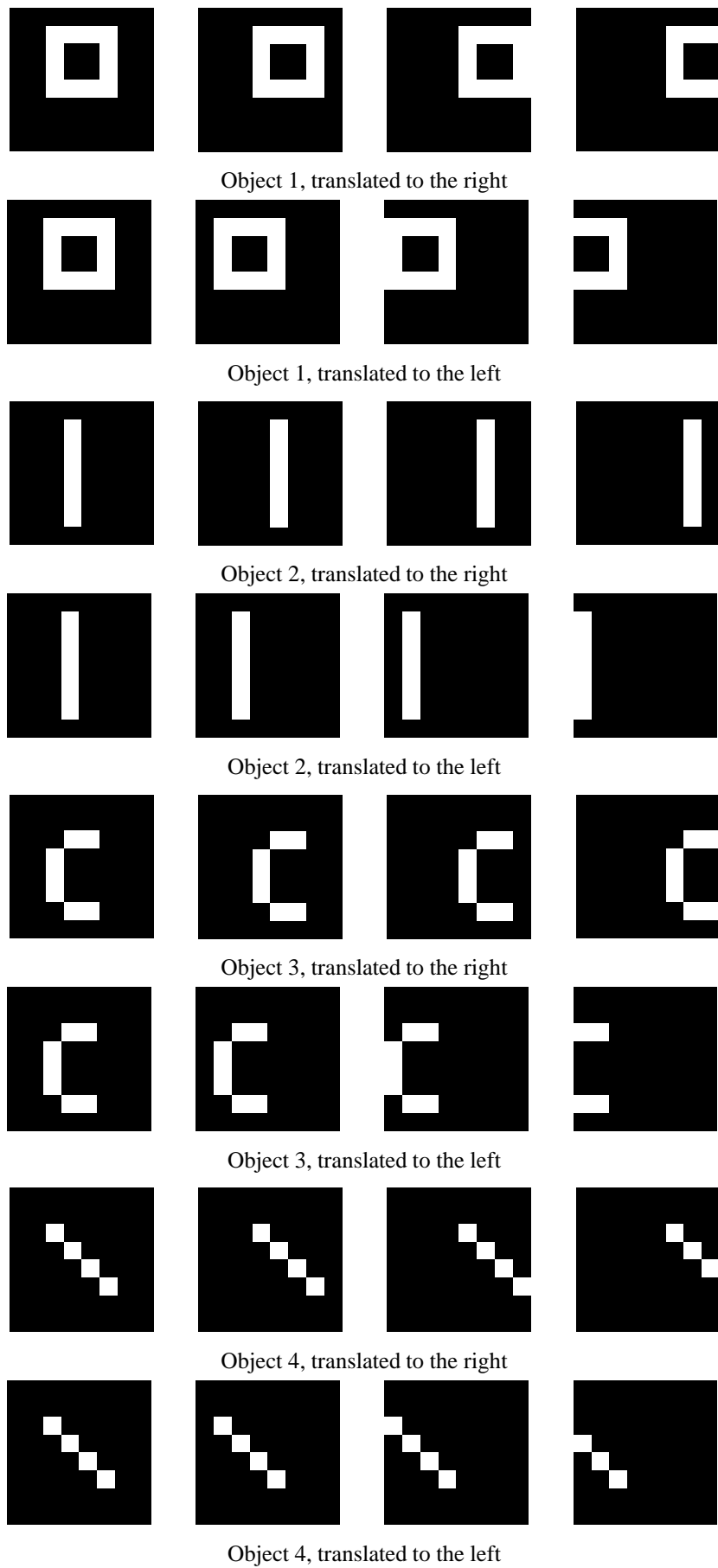


Figure 3. Input images. The left-most column shows each object at its initial position. The subsequent columns show translated versions of the object, either to the right or to the left.

The Hebbian learning rule is modified to use \tilde{y} as follows

$$\Delta W_{ij} \sim \tilde{y}_i x_j [1 + \cos(\phi_j - \theta_i)] \quad (8)$$

We term this the trace learning rule, as it employs the trace of y . This follows the terminology of.¹⁴

Third, the update equations 3 are applied for only 1 iteration rather than 100. The justification for doing this is that we want to create an association between the translated version of the object and its original version. We take advantage of the inertia in the system through the moving averaged value, \tilde{y} , and a reduced settling time. In biological systems, there appears to be support for this type of update, as there are two pathways in the visual system, one to deal with static object representations, and the other to deal with motion. The motion pathway operates at a much faster time scale than the static pathway.

4. EXPERIMENTAL RESULTS

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 8x8 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.

We used the following parameters to instantiate the model: $\alpha = 0.5, \gamma = 0.25$. For the learning rate we used $\epsilon = 0.025$, and the integration step $dt = 0.05$. The learning rate underwent an exponential decay, with a reduction of 15% every 200 iterations. The value of $\mu = 0.8$ was used in equation 7 to calculate the moving average.

After the static phase of learning, ie presentation of the objects before they were translated, the network formed four unique winners for the four objects presented 96% of the time.

Subsequently, the objects were translated, and the network learned the association between the translated versions of the object and the object in its original position. After learning, the accuracy of the association was measured as follows. Let unit m be the winner in the upper layer for object k in its original position. We translated object k to form objects k_1, k_2, \dots, k_n where n is the total number of displacements used. In our experiments, $n = 6$, and included displacements of ± 3 pixels from the original position. If the winner for the translated object k_i matches the winner m , then this is counted as a successful association. The total number of successful associations is measured across all the translated versions of the four objects. The accuracy of association was 76.4% over 100 trials. This indicates good performance of the network in learning translation invariance.

We should mention that though we did not dwell on the phase behavior of the network in this paper, the performance of the network degrades significantly if phase information is not used in the learning rules.

5. DISCUSSION

The methods and approach presented in this paper demonstrate that it is possible for a network of oscillatory units to achieve translation invariant object representations, provided the suitable learning paradigm is used. One plausible mechanism is for the system to use a trace learning rule to establish equivalence between translated versions of an object. As pointed out in previous work,¹⁴ this is a biologically plausible mechanism.

The human visual system employs two pathways for processing visual inputs, the “what” and the “where” pathways.¹⁵ The framework provided in this paper can form the basis for a deeper exploration of the correspondence between these two pathways, and invariant object representation at multiple levels of abstraction.

There are several areas in which the results of this paper can be improved. More complex motion patterns need to be understood, as objects in the real world can undergo a variety of displacements, including rotation. The effect of relaxing some of the assumptions in the current model need to be investigated, such as whether a static presentation of the objects is really necessary initially. The initial experiments presented in this paper used synthetic images as a proof-of-concept, and further experiments using real images need to be performed. The network connectivity employed in this paper is all-to-all. More realistic topographic mapping between layers can be used instead.

The method of Sun *et al*¹⁶ used a network of oscillatory units to achieve figure-ground separation, such that a moving object is segmented from a stationary background. They used supervised learning through back-propagation to train their network. This differs from the unsupervised learning mechanism used in our paper.

Discussions of mechanisms and phenomena that surround the issue of resetting cortical dynamics can be found in¹⁷ and .¹⁸ Furthermore, special neural activity has been recorded during saccades¹⁹ that explains why we are unaware of fast motion across the retina at this time, a phenomenon known as saccadic suppression. Based on these mechanisms and observations it is plausible that a phase resetting occurs at the time of a saccade, or equivalently at the time of presentation of a new stimulus. In addition, for the observation of motion within a saccade, phase resetting should not occur, as the motion across the retina is less rapid than when a saccade occurs.

REFERENCES

1. S. Ullman, M. Vidal-Naquet, and E. Sali, "Visual features of intermediate complexity and their use in classification," *Nature Neuroscience* **5**(7), pp. 682–7, 2002.
2. F. Rosenblatt, *Principles of Neurodynamics: Perception and the Theory of Brain Mechanisms*, Spartan Books, Washington, 1962.
3. 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* **338**(6213), pp. 334–337, 1989.
4. 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* **397**(6718), pp. 430–433, 1999.
5. C. von der Malsburg and W. Schneider, "A neural cocktail-party processor," *Biol. Cybern* **54**(1), pp. 29–40, 1986.
6. J. Buhmann and C. V. D. Malsburg, "Sensory segmentation by neural oscillators," in *International Joint Conference on Neural Networks*, **2**, pp. 603–607, 1991.
7. K. Chen, D. Wang, and X. Liu, "Weight adaptation and oscillatory correlation for image segmentation," *IEEE Transactions on Neural Networks* **11**(5), pp. 1106–1123, 2000.
8. D. L. Wang and X. Liu, "Scene analysis by integrating primitive segmentation and associative memory," *IEEE Transactions on Systems, Man, and Cybernetics Part B* **32**(3), pp. 254–268, 2002.
9. M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Wurtz, and W. Konen, "Distortion invariant object recognition in the dynamic link architecture," *IEEE Trans. Comput.* **42**(3), pp. 300–311, 1993.
10. A. R. Rao, G. A. Cecchi, C. C. Peck, and J. R. Kozloski, "Unsupervised segmentation with dynamical units." submitted to *IEEE Transactions on Neural Networks*, December 2005.
11. B. Olshausen and D. Fields, "Natural image statistical and efficient coding," *Network: Computation in Neural Systems* **7**, pp. 333–339, 1996.
12. C. V. der Malsburg, "The what and why of binding: The modeler's perspective," *Neuron* , pp. 95–104, 1999.
13. K. Obermayer and G. Blasdel, "Geometry of orientation and ocular dominance columns in monkey striate cortex," *J. Neuroscience* **13**, pp. 4114–4129, 1993.
14. G. Wallis, "Using spatio-temporal correlations to learn invariant object recognition," *Neural Networks* , pp. 1513–1519, Dec 1996.
15. M. Goodale and A. Milner, "Separate visual pathways for perception and action," *Trends Neuroscience* **15**, pp. 20–5, Jan. 1992.
16. H. Sun, L. Liu, and A. Guo, "A neurocomputational model of figure-ground discrimination and target tracking," *IEEE Transactions on Neural Networks* **10**(4), 1999.
17. G. Francis, S. Grossberg, and E. Mingolla, "Cortical dynamics of feature binding and reset: Control of visual persistence," *Vision Research* **34**(8), pp. 1089–1104, 1994.
18. D. M. Eagleman and T. J. Sejnowski, "Motion integration and postdiction in visual awareness,"
19. A. Thiele, P. Henning, M. Kubischik, and K. P. Hoffmann, "Neural mechanisms of saccadic suppression," *Science* **295**, pp. 2460–2462, March 2002.