PERGAMON

Neural Networks 15 (2002) 423–439

Neural Networks

www.elsevier.com/locate/neunet

Contributed article

A dynamically coupled neural oscillator network for image segmentation

Ke Chen[a,*], DeLiang Wang[b]

[a]School of Computer Science, The University of Birmingham, Edgbaston, Birmingham B15 2TT, UK
[b]Department of Computer and Information Science, Center for Cognitive Science, The Ohio State University, Columbus, OH 43210, USA

Received 13 September 2000; revised 1 February 2002; accepted 1 February 2002

Abstract

We propose a dynamically coupled neural oscillator network for image segmentation. Instead of pair-wise coupling, an ensemble of oscillators coupled in a local region is used for grouping. We introduce a set of neighborhoods to generate dynamical coupling structures associated with a specific oscillator. Based on the proximity and similarity principles, two grouping rules are proposed to explicitly consider the distinct cases of whether an oscillator is inside a homogeneous image region or near a boundary between different regions. The use of dynamical coupling makes our segmentation network robust to noise on an image, and unlike image processing algorithms no iterative operation is needed for noise removal. For fast computation, a segmentation algorithm is abstracted from the underlying oscillatory dynamics, and has been applied to synthetic and real images. Simulation results demonstrate the effectiveness of our oscillator network in image segmentation.

Keywords: Neural oscillator network; Dynamic link; Image segmentation; Oscillatory correlation; Synchronization; Desynchronization; Locally excitatory globally inhibitory oscillator networks; Multiscale processing

1. Introduction

One of the fundamental tasks in vision is image segmentation. It is the process to partition an image into a number of coherent regions; the pixels belonging to the same region should have similar properties in terms of intensity, color, and texture, etc. while two adjacent regions should have dissimilar properties. Although humans perform the task without difficulty, image segmentation remains one of the most challenging problems in machine vision.

In the computer vision community, many attempts have been made for image segmentation (for reviews see Haralick & Shapiro, 1985; Pal & Pal, 1993; Zhu & Yuille, 1996; Zucker, 1976). First, pixel classification provides a segmentation technique that associates a pixel with a specific label by means of a classification method. A typical technique in this category is thresholding where a pixel is labeled if a measure of the pixel is above a given threshold. The thresholding technique has been extended to more sophisticated forms, such as a multiple-threshold test (Haralick & Shapiro, 1985; Kohler, 1981) and a recursive thresholding test (Cheriet, Said, & Suen, 1998). As a salient feature, edge information can be used to separate two adjacent regions, and hence edge-based segmentation forms another category. In general, such methods consist of two consecutive stages, i.e. edge detection and contour completion. Consequently, regions embraced by closed contours lead to final segmentation (Foresti, Murino, Regazzoni, & Vernazza, 1994; Geman, Geman, Graffigne, & Dong, 1990). Region-based techniques are yet another class of methods for image segmentation. Such methods operate directly on regions, e.g. region growing and split-and-merge (Horowitz & Pavlidis, 1976; Revol & Jourlin, 1997). The main idea underlying these methods is to iteratively group (split) pixels into connected regions in accordance with some prespecified homogeneity criteria. More recently, hybrid techniques combining two or more kinds of aforementioned techniques (Ahuja, 1996; Haddon & Boyce, 1990; Zhu & Yuille, 1996) have been suggested to achieve reliable segmentation. Computationally, all of the above algorithms are of serial nature though some partially parallel algorithms have been developed (Liou, Chiu, & Jain, 1991; Manjunath & Chellappa, 1993). Due to many uncertain factors for image segmentation (Wilson & Spann, 1996), solutions are often difficult to obtain and segmentation is, to a great extent, viewed as an unsolved problem.

Neural networks have been successfully applied in many fields. However, little has been reported on image segmentation. Due to the lack of an efficient representational

* Corresponding author. Tel.: +44-121-414-4769; fax: +44-121-414-4281.

E-mail addresses: k.chen@cs.bham.ac.uk (K. Chen), kchen@ieee.org (K. Chen).

0893-6080/02/$ - see front matter
PII: S0893-6080(02)00028-X

framework for encoding multiple objects simultaneously, image segmentation is particularly challenging for neural networks. Most of neural network methods pose image segmentation as pixel classification (Pal & Pal, 1993) and typically use a supervised learning process. Other segmentation methods include Boltzmann machines (Sejnowski & Hinton, 1987), multilayer perceptrons (Mozer, Zemel, Behrmann, & Williams, 1992), and biologically motivated neural networks (Grossberg & Wyse, 1991). However, all of these methods achieve only limited success. Recently, Kohonen's self-organization maps have been also applied to segmentation (Bhandarkar, Koh, & Suk, 1997; Kohonen, 1995). A major shortcoming in this approach is that there is no autonomous control over the number of regions that should be segmented.

An elegant way to encode the binding of features is suggested by the temporal correlation theory (von der Malsburg, 1981; Milner, 1974), whereby an object is represented by the temporal correlation of the firing activities of the distributed cells that detect different features of the object. A special form of temporal correlation is called *oscillatory correlation* (von der Malsburg & Schneider, 1986; Wang & Terman, 1995). The discovery of synchronous oscillations in the brain supports the oscillatory representation to implement temporal correlation; where each object is represented by a group of synchronized oscillators and different objects are represented by different groups whose oscillations are desynchronized from each other. Recently, Terman and Wang (1995) proposed a class of *locally excitatory globally inhibitory oscillator networks* (LEGION) on the basis of oscillatory correlation. LEGION can rapidly achieve both synchronization and desynchronization, and has been successfully applied to real image segmentation (Wang & Terman, 1997). However, due to pair-wise coupling between oscillators, the LEGION network is sensitive to noise; for example, it does not work for the noisy image in Fig. 4(A).

In this paper, we propose a dynamically coupled network on the basis of oscillatory correlation and LEGION. We use an ensemble of oscillators in a local region for grouping instead of pair-wise coupling. Dynamical coupling is proposed to generate a local coupling structure for an oscillator in terms of its surrounding active oscillators, and a local coupling structure provides the basis for computing local coherent properties of an object. Two grouping rules are proposed for examining the similarity between oscillators, which leads to robust performance for image segmentation. For fast computation, we extract a segmentation algorithm from oscillatory dynamics. The algorithm has been successfully applied to both synthetic and real images.

Unlike previous methods for robust segmentation, no iterative smoothing is used in our method. For iterative smoothing, it is well known that the performance highly depends upon *when* the algorithm is terminated, which, coupled with the fact that these algorithms generally converge to a uniform intensity image (Perona & Malik, 1990; Weickert, 1997), causes what we refer to as the termination problem. Without iterative operations at the image level, the termination problem does not occur in our approach.

In Section 2, we describe the proposed oscillator network, including dynamical coupling structures and grouping rules. In Section 3, model behavior is analyzed and discussed. In Section 4, an algorithm derived from oscillatory dynamics is presented for gray-level image segmentation. Section 5 reports the segmentation results on both synthetic and real images. Further discussions are given in Section 6.

2. Model description

In this section, we describe our network architecture and its dynamics, which can be viewed as an extension of LEGION. Two grouping rules are proposed on the basis of dynamical coupling structures.

In our neural oscillator network, a single oscillator (i,j) consists of an excitatory unit x_{ij} and an inhibitory unit y_{ij} as defined by Wang and Terman (1997):

$$\dot{x}_{ij} = 3x_{ij} - x_{ij}^3 + 2 - y_{ij} + I_{ij}H(p_{ij} - \theta) + S_{ij} + \rho, \tag{1a}$$

$$\dot{y}_{ij} = \epsilon[\alpha(1 + \tanh(\beta x_{ij})) - y_{ij}]. \tag{1b}$$

Here $H(\cdot)$ is the Heaviside step function defined as $H(\nu) = 1$ if $\nu \geq 0$ and $H(\nu) = 0$ if $\nu < 0$. I_{ij} denotes the external stimulation to the oscillator, and S_{ij} represents overall coupling from other oscillators in the network. The parameter ρ denotes the amplitude of a Gaussian noise term, which is introduced to test the robustness of the system and to actively assist in desynchronizing different patterns. The parameters ϵ is chosen to be small, $0 < \epsilon \ll 1$. In this case, Eqs. (1a) and (1b) corresponds to a standard relaxation oscillator (van der Pol, 1926). The dynamics of a single relaxation oscillator is summarized as follows, where we drop all the subscripts to simplify the presentation. The x-nullcline of Eqs. (1a) and (1b), $\dot{x} = 0$, is a cubic function, while the y-nullcline, $\dot{y} = 0$, is a sigmoid function. The parameter β controls the steepness of a sigmoid function and is chosen to be large. For an input $I > 0$, the two nullclines intersect only on the middle branch of the cubic, and Eqs. (1a) and (1b) gives rise to a stable periodic trajectory. In this case, the oscillator is called enabled. As illustrated in Fig. 1(A), four bounding points for the periodic trajectory are *LC*, *LK*, *RK*, and *RC* (Wang & Terman, 1997), whose x values are $LC_x = -2$, $LK_x = -1$, $RK_x = 1$, and $RC_x = 2$, respectively. The periodic solution alternates between an *active phase* ($RK_x \leq x \leq RC_x$) and a *silent phase* ($LC_x \leq x \leq LK_x$). When the oscillator stays in the active phase, it is called active; otherwise, it is called silent. Within either phase, the oscillator exhibits near steady state behavior. In contrast, the transition between the two phases takes place on a fast time scale, referred to as jumping. The parameter α is introduced to control the relative times that the oscillator

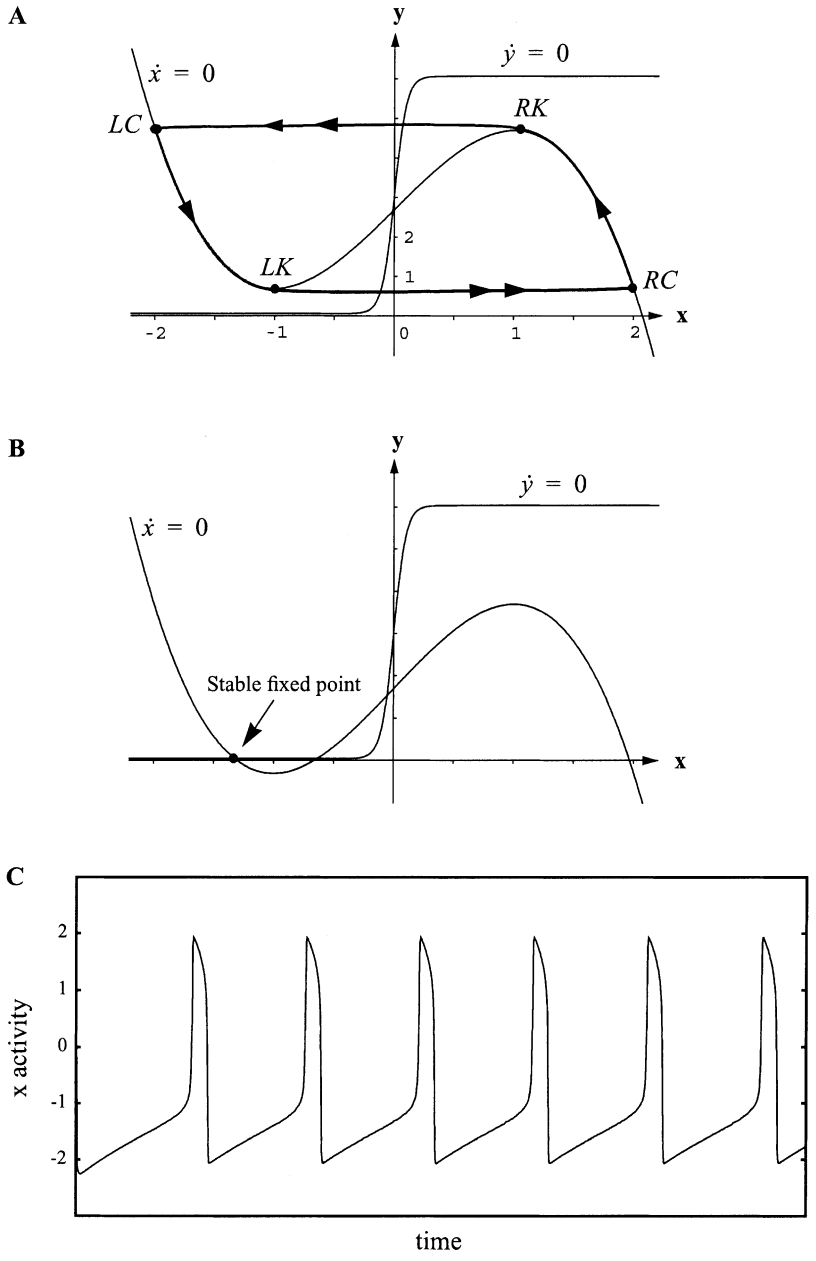

Fig. 1. Nullclines and trajectories of a single relaxation oscillator. (A) Behavior of an enabled oscillator. The bold curve shows the limit cycle of the oscillator, whose direction of motion is indicated by arrowheads with double arrowhead indicating jumping. *LC*, *LK*, *RC*, and *RK* indicate four bounding points for the periodic trajectory. (B) Behavior of an excitable oscillator. The oscillator approaches a stable fixed point. (C) Temporal activity of the oscillator. The x value of the oscillator is plotted. The parameter values used are $I = 0.8$, $\rho = 0.02$, $\epsilon = 0.04$, $\alpha = 9.0$, and $\beta = 0.1$.

stays in the two phases, with a larger α implying a shorter active phase. For an input $I < 0$, the two nullclines intersect on the left branch of the cubic, and Eqs. (1a) and (1b) produces a stable fixed point as illustrated in Fig. 1(B). In this case, the oscillator is called excitable. Fig. 1(C) shows the enabled behavior of the oscillator for a specific set of parameters.

In Eq. (1a), the Heaviside term provides a mechanism to distinguish between major regions and noisy fragments. Wang and Terman (1997) suggested that a major region must contain at least one oscillator, called a leader, which is located at the center of a large, homogeneous region. A noisy fragment does not contain such an oscillator. Basically, a leader receives large lateral excitation from its neighborhood. The variable p_{ij} denotes the lateral potential of oscillator (i,j), through the threshold θ, which critically determines whether oscillator (i,j) is a leader. The collection of the oscillators corresponding to noisy fragments forms the background, and such oscillators become excitable shortly after the system starts.

For image segmentation, we study two-dimensional oscillator networks. As illustrated in Fig. 2, there is a local fixed

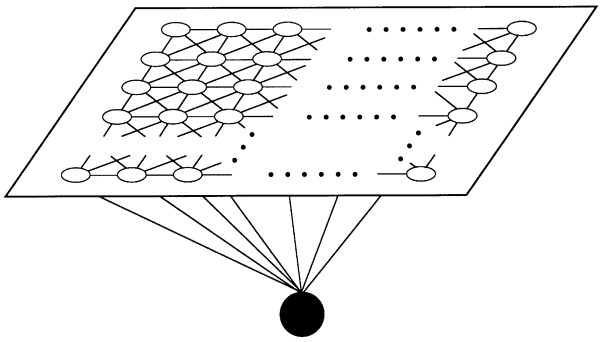

Fig. 2. Architecture of a two-dimensional neural oscillator network with eight nearest-neighbor fixed coupling. The global inhibitor is indicated by the black circle.

structure, where an oscillator is connected to only eight immediate neighbors except for those on the boundaries where no wraparound is used. The activation of an oscillator spreads through fixed connections only. The global inhibitor is connected with all the oscillators in the two-dimensional network, and its activity, z, is defined as

$$\dot{z} = \phi(\sigma_\infty - z), \tag{2}$$

where $\sigma_\infty = 0$ if $x_{ij} < \theta_z$ for every oscillator (i,j), and $\sigma_\infty = 1$ if $x_{ij} \geq \theta_z$ for at least one oscillator. θ_z is a threshold. If $\sigma_\infty = 0$, the global inhibitor is not triggered and the oscillators on the network receive no inhibition from the global inhibitor. If $\sigma_\infty = 1$, the global inhibitor is triggered and all the oscillators on the network receive inhibition. $\phi \gg \epsilon$ is a parameter.

In LEGION, when an oscillator is active, it triggers the global inhibitor, which in turn exerts inhibition on every oscillator in the network. On the other hand, the active oscillator propagates its activation through fixed connections to its immediate neighbors and from there to further neighbors. For gray-level image segmentation, we assume that every oscillator is uniformly stimulated when an image is presented. Thus, the phase of an oscillator is influenced only by the coupling term, S_{ij}, in Eq. (1a). It is desirable that a silent oscillator is activated if the oscillator is coherent with its surrounding active oscillators. Eventually, we want those oscillators corresponding to the same region to synchronize, while oscillators corresponding to different regions desynchronize.

In our model, oscillators that are in the active phase simultaneously form a single segment, and as a result they should correspond to a set of coherent pixels. Therefore, image attributes of these active oscillators provide a coherence measure of the entire group. In an image with considerable noise, group-wide measures are more robust than single pixel measures; the latter are used by Wang and Terman (1997). On the other hand, proximity is one of the most important grouping principles; remote pixels, even though they may be in the same segment, should contribute little to recruiting a particular pixel. Based on these considerations, we use the attributes of a *local* ensemble, called a neighborhood, of active oscillators for grouping.

How to determine the size and the shape of such a neighborhood for grouping? The neighborhood should not be uniform; for example, pixels near a boundary between different regions need to be treated differently than pixels in the interior of a region because discontinuity near a boundary causes local image attributes to change dramatically. In general, the interior of an image region must be treated differently, in terms of grouping, than areas near the region boundary. Within the interior of a region, a larger and more uniform neighborhood can be used, whereas near a region boundary the neighborhood should be tailored so that it does not cross different regions. In oscillatory correlation, this can be naturally achieved by limiting the neighborhood to the currently active oscillators because they correspond to the same region. To deal with these different situations, we introduce a set of dynamical coupling neighborhoods and propose two grouping rules for handling region interior and region boundary separately; these are all embodied in the overall coupling term, S_{ij}, in Eq. (1a).

We denote the fixed set of the eight immediate neighbors of oscillator (i,j) as $N(i,j)$ (see Fig. 2). To measure similarity for oscillators that lie inside a homogeneous region, we introduce a pair of neighborhoods, referred to as *pixel neighborhood* and *region neighborhood*, respectively. The pixel neighborhood is used to measure local image attributes of a single oscillator, while the region neighborhood is used to measure those of the active neighbors of the oscillator. The pixel neighborhood of oscillator (i,j) is defined as

$$N_1^a(i,j) = \{(k,l) | i - R_a \leq k \leq i + R_a; j - R_a \leq l \leq j + R_a\}, \tag{3}$$

where R_a is a parameter that determines the size of the pixel neighborhood. An example is shown in Fig. 3(A). While $N_1^a(i,j)$ is fixed, the corresponding region neighborhood is dynamical, defined in terms of currently active neighbors of (i,j) :

$$N^a(i,j) = \left\{ (k,l) | (k,l) \in \bigcup_{(m,n)\in N(i,j), x_{mn} \geq 0, H(u_{mn} - 0.5) = 0} N_1^a(m,n) \right\},$$

$$N_2^a(i,j) = \begin{cases} N^a(i,j) & |N^a(i,j)| > 0 \\ N_2^b(i,j) & \text{otherwise} \end{cases}. \tag{4}$$

Here u_{mn} is a variable indicating whether the corresponding pixel of the active oscillator (m,n) is near a boundary, to be defined later on. The above union operation is performed only over those active oscillators in $N(i,j)$ that are considered to be in the interior of a region. The motivation for this should be clear later when the second pair of neighborhoods are introduced for pixels that are near a region boundary. Fig. 3(B) gives an example of the region neighborhood corresponding to $|N^a| > 0$. $N_2^b(i,j)$ is an alternative region neighborhood to be defined below. The region

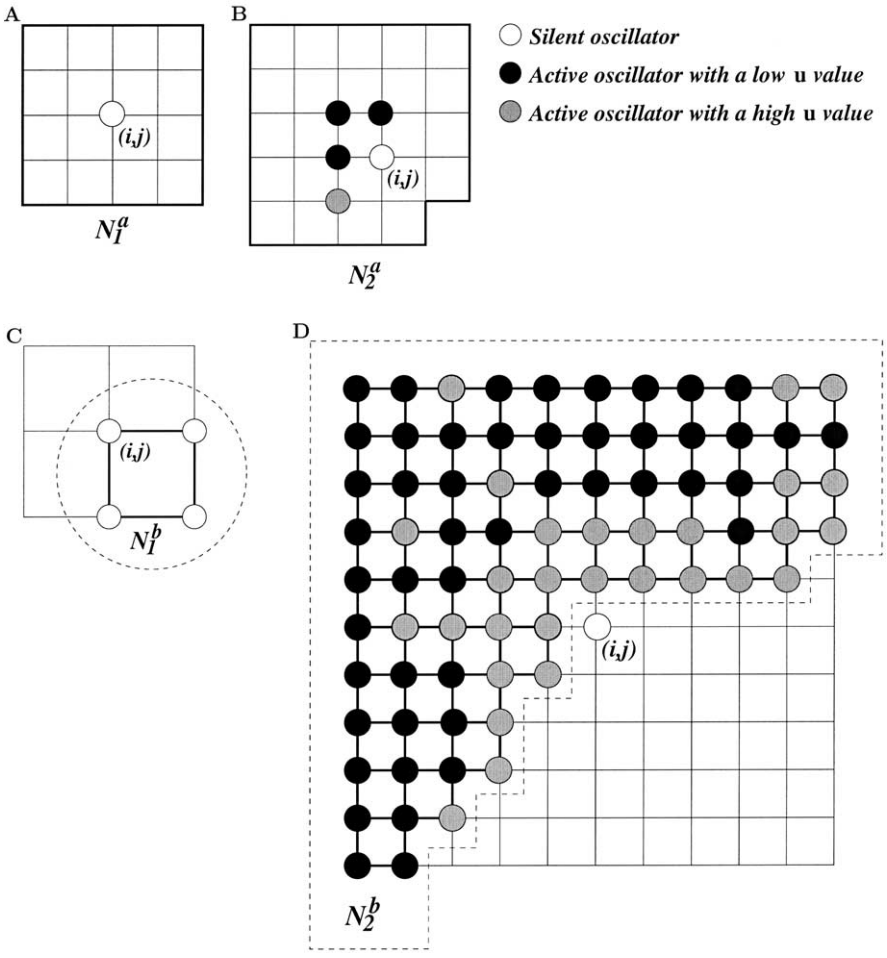

Fig. 3. Examples of various neighborhoods. (A) An example of the pixel neighborhood $N_1^a(i,j)$, where $R_a = 2$ is used. (B) An example of the region neighborhood $N_2^a(i,j)$, where $R_a = 2$ is used. (C) An example of the pixel neighborhood $N_1^b(i,j)$, where four silent oscillators constitute a dynamic coupling structure. (D) An example of the region neighborhood $N_2^b(i,j)$, where $R_0 = 5$ is used and all the active oscillators enclosed constitute a dynamic coupling structure.

neighborhood provides a local measure of an image region being expanded.

To deal with those pixels near a region boundary separately, we introduce another pair of neighborhoods. The key for this pair is that neighborhoods should be limited to active oscillators only, since they belong to the same segment which corresponds to the same region. First, a pixel neighborhood of this kind is defined as

$$N_1^b(i,j) = \{(k,l) | (k,l) \in N(i,j), x_{kl} < 0\} \cup \{(i,j)\}. \quad (5)$$

An example of this pixel neighborhood is shown in Fig. 3(C). Theoretically speaking, a single silent oscillator—(i,j) itself—may be evaluated for recruiting. However, as our experiments with real images show, single pixels are too noise-sensitive to obtain a reliable measure. That is why in the definition of $N_1^b(i,j)$ we include the silent neighbors of (i,j). Notice that only silent oscillators should be included in N_1^b because (i,j) may belong to a region different from the currently expanding one. Finally, corresponding to N_1^b, a region neighborhood, $N_2^b(i,j)$, is defined as

$$N_2^b(i,j) = \{(k,l) | i - R_b \le k \le i + R_b, j - R_b \le l \le j + R_b, x_{kl} \ge 0\}. \quad (6)$$

Here the integer R_b determines the size of the region neighborhood, and satisfies the following condition: R_b is the minimal value guaranteeing that the number of active oscillators in $N_2^b(i,j)$ is no less than $(2R_0 + 1)^2/2$, where R_0 is a prespecified constant, and $R_b \le [\min(M,N)/2)]$ assuming that the size of the image is $M \times N$. Fig. 3(D) gives an example of $N_2^b(i,j)$. Compared to typical $N_2^a(i,j)$, which includes the *neighborhoods* of active oscillators, $N_2^b(i,j)$ includes only active oscillators themselves. This reflects the fact that we use N_2^b for grouping pixels near a region boundary, and a neighborhood of an active oscillator is likely to cross different regions. The latter situation will produce unreliable measures and thus should be avoided.

All the neighborhoods in Eqs. (3)–(6) are defined without explicitly considering the boundary of the entire image; and

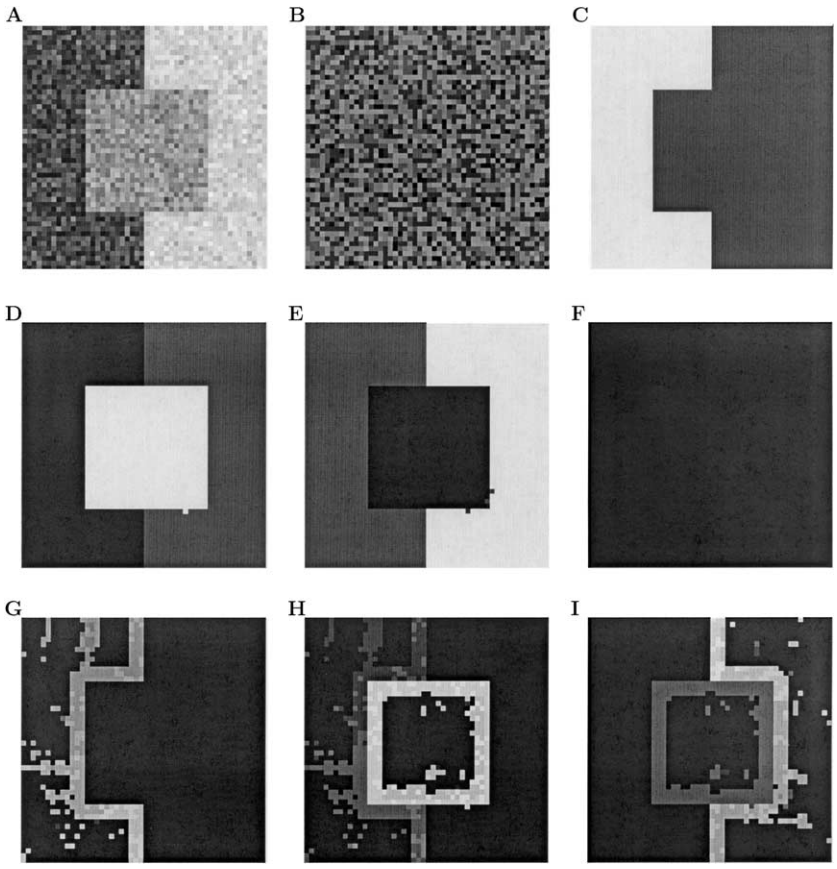

Fig. 4. Segmentation by a dynamically coupled oscillator network. (A) A 50×50 noisy image. The image is created from three regions with constant intensities of 42, 160, and 192, added with Gaussian noise of zero mean and standard deviation 24. In the figure (B)–(E) correspond to instantaneous x activity and (F)–(H) u activity. (B) A snapshot at the beginning of dynamic evolution. (C)–(E) A sequence of snapshots shortly after B. (F)–(H) A sequence of snapshots of u activity taken at the corresponding times of (C)–(E).

they are delimited by the boundary. Except for $N_1^a(i,j)$, shapes of all the neighborhoods defined above are dynamically determined by all the active oscillators surrounding a specified silent oscillator (i,j) and specify dynamical coupling structures associated with oscillator (i,j). Such dynamical coupling structures are incorporated by the grouping rules described later to yield robust segmentation performance.

To detect homogeneity in gray-level images, the first- and the second-order statistics are commonly used. Similarly, we adopt mean, μ, and standard deviation, σ, to measure homogeneity in a specific neighborhood. Note that the mean and standard deviation are set to zero if the neighborhood is the empty set. Based on the similarity principle, grouping takes place if the local attributes of a silent oscillator are similar to those of a local region corresponding to a set of local active oscillators. To detect the similarity, we suggest two grouping rules for the two situations considered above: region interior and region boundary. First, we examine similarity for the type a pair of neighborhoods, N_1^a and N_2^a. For oscillator (i,j), we have

$$S_\mu^a(i,j) = \omega_\mu^a - |\mu_{N_1^a(i,j)} - \mu_{N_2^a(i,j)}|, \tag{7a}$$

$$S_\sigma^a(i,j) = \omega_\sigma^a - |\sigma_{N_1^a(i,j)} - \sigma_{N_2^a(i,j)}|. \tag{7b}$$

Here ω_μ^a and ω_σ^a are two tolerance parameters for measuring similarity. The subscripts, $N_1^a(i,j)$ and $N_2^a(i,j)$, indicate where the statistics are calculated. $S_\mu^a(i,j)$ and $S_\sigma^a(i,j)$ are measures in terms of the mean and the standard deviation. Similarly, we examine similarity for the type b pair of neighborhoods, N_1^b and N_2^b, as follows:

$$S_\mu^b(i,j) = \omega_\mu^b - |\mu_{N_1^b(i,j)} - \mu_{N_2^b(i,j)}|, \tag{8a}$$

$$S_\sigma^b(i,j) = \omega_\sigma^b - |\sigma_{N_1^b(i,j)} - \sigma_{N_2^b(i,j)}|. \tag{8b}$$

Based on Eqs. (7a)–(8b), we formulate two grouping rules for oscillator (i,j) with the condition that there is at least one active oscillator in $N(i,j)$: (a) the oscillator will be recruited if $S_\mu^a(i,j) \geq 0$ and $S_\sigma^a(i,j) \geq 0$; (b) the oscillator is recruited if $S_\mu^b(i,j) \geq 0$ and $S_\sigma^b(i,j) \geq 0$. The two grouping rules recruit (i,j) in two different ways, and together implement both the proximity and the similarity principles. When an oscillator lies in a homogeneous region, Rule a should be used for recruitment. In contrast, when it is near a boundary between different regions, Rule a is unlikely to yield

recruitment and Rule b is used instead for expanding the region to produce a precise region boundary.

We now define u_{ij} introduced in Eq. (4) and want u_{ij} to approach unity only if oscillator (i,j) is activated by Rule b rather than Rule a. Therefore, u_{ij} is defined as

$$\dot{u}_{ij} = \lambda(\delta^b_{ij} - u_{ij})H(\dot{x}_{ij} - \eta) - \epsilon u_{ij}, \tag{9}$$

where $\delta^b_{ij} = [1 - H(S^a_\mu(i,j))H(S^a_\sigma(i,j))]H(S^b_\mu(i,j))H(S^b_\sigma(i,j))$. $\lambda \gg \epsilon$ is a parameter indicating the rate at which u_{ij} changes. The Heaviside function, within which η is a parameter, dictates that u_{ij} does not change on the fast time scale unless the derivative of x_{ij} increases significantly; in other words, u_{ij} does not change much unless the corresponding oscillator jumps up (see Fig. 1(A)).

The initial value of u_{ij} is set to zero. Thus, according to Eq. (9) u_{ij} stays at zero unless only Rule b is used in recruiting (i,j). In the latter case, u_{ij} approaches one quickly. Beyond a brief time of jumping up, u_{ij} slowly decays to zero since the first term in Eq. (9) is zero, and the second term acts on the slow time scale. The slow decay of u_{ij} ensures that once it is activated it stays at a high value for a while, say for the time period of the active phase. Taken together, u_{ij} serves the purpose of recording whether (i,j) has been recruited in the last cycle of oscillation by Rule b only, or whether pixel (i,j) is near a region boundary. Thus, Eq. (9) implements the requirement for $N^a_2(i,j)$, the only neighborhood that uses u_{ij}. Given that the pixels recruited by Rule b likely correspond to near-boundary ones, it should now be clear that these pixels should be excluded in the definition of type a neighborhoods, which are used to evaluate the recruiting of interior pixels of a region.

On the basis of two grouping rules, the overall coupling term, S_{ij}, in Eq. (1a) is given by

$$\begin{aligned} S_{ij} = \Bigg[& H\left(S^a_\mu(i,j)\right)H\left(S^a_\sigma(i,j)\right) \\ & + H\left(S^b_\mu(i,j)\right)H\left(S^b_\sigma(i,j)\right)\Bigg]H\left(\sum_{(k,l)\in N(i,j)} H(x_{kl}) - 0.5\right) \\ & - W_z H(z - \theta_z). \end{aligned} \tag{10}$$

Here W_z is the weight of inhibition from the global inhibitor z. In our network, both W_z and θ_z are always set to 0.5. Note that S_{ij} depends upon fixed connectivity, and there must be at least one active oscillator in $N(i,j)$ in order for (i,j) to be recruited. This yields a connectivity constraint. Under this condition, when either of two grouping rules is satisfied, $S_{ij} > 0$; otherwise, $S_{ij} < 0$ due to the inhibition. Note that leaders can jump up by themselves when the global inhibitor is not activated.

We summarize the dynamics of our network as follows. Suppose that all the oscillators are initialized in the silent phase. Once a leader enters the active phase, it triggers the global inhibitor, which exerts inhibition on the entire network. For a silent oscillator, its overall coupling is positive if either Rule a or b is satisfied, and in this case the oscillator jumps to the active phase. Otherwise, overall coupling is negative due to the global inhibition, which makes the oscillator stay in the silent phase. In this way, activation is propagated to other oscillators via local fixed connectivity until all the oscillators representing the same region are activated. Thus, the dynamics of our network preserves the property of LEGION. That is, it is the process of both synchronization through local excitation among neighboring oscillators and desynchronization through global inhibition (Terman & Wang, 1995; Wang & Terman, 1997).

3. Model behavior

From the dynamical system perspective, the system (1a)–(10) is very similar to that of Wang and Terman (1997). The only difference is in the definition of the overall coupling term S_{ij}. For Wang and Terman (1997), S_{ij} is simply a weighed sum within $N(i,j)$. In our model, in addition to $N(i,j)$, two pairs of neighborhood are introduced: N^a_1, N^a_2, N^b_1, and N^b_2. Out of the four neighborhoods, N^a_1 is fixed and the remaining three vary dynamically. One additional complexity lies in the fact that N^a_2 depends on the variable u. Despite these complexities in forming dynamical neighborhoods, from the standpoint of oscillator i, the net effect of all these neighborhoods is a binary value: the first term in Eq. (10).

The binary nature of lateral coupling has the same effect as dynamic normalization of lateral connections in Wang and Terman (1997), which simplified their analysis and improved the quality of synchronization within a block of oscillators that corresponds to the same segment. Thus, the cubic of any oscillator—leader or not—can take only one of three positions: $S_{ij} = 0$, $S_{ij} = 1 - W_z$, and $S_{ij} = -W_z$. It is possible that, for a very brief time period during jumping up, $S_{ij} = 1$; this possibility does not affect system behavior and can be eliminated through parameter tuning. From this as well as Eq. (1a), it is clear that one should set $0 < I_{ij} < W_z < 1$, so that an oscillator cannot jump to the active phase unless it receives strong lateral excitation, except for a leader which can jump to the active phase if the global inhibitor is not triggered.

As an oscillator block is jumping to the active phase dynamical neighborhoods change. As a result, an active oscillator may lower its cubic from $S_{ij} = 1 - W_z$ to $S_{ij} = -W_z$ and it may also shift in the reverse direction, when the block is expanding. Such phase shifts induce oscillators to move between the active phases of two cubics, while they are active. After a block has completely jumped to the active phase, such phase shifts stop because dynamical neighborhoods do not change until the block starts to jump down. Another difference caused by dynamical neighborhoods occurs when a block is jumping down to the silent phase.

In Wang and Terman (1997), when an oscillator jumps down it lowers the cubics of its active neighbors, thus encouraging them to jump down as well. This may not happen in our model, because an oscillator jumping down may have no effect on the cubics of its active neighbors. The reason is that the excitatory lateral coupling of an oscillator is combined into a single binary value in Eq. (10). These two distinctions make the quality of synchrony within each block in our model not as good as in theirs (see below), but they do not affect the overall system behavior of oscillatory correlation.

In a more detailed version that contains mathematical analysis, Wang and Terman (1996) presented the combined (binary) form of lateral excitation as an alterative definition to the model presented in Wang and Terman (1997). They noted that the combined from of lateral excitation has the benefit of being simpler without dynamical weight normalization, but leads to less precise synchrony within each block and less flexibility in choosing parameters. They also stated that the same analytical results obtained for their 1997 definition apply to the alternative definition. The behavior of our model is consistent with their analysis.

In Eq. (9), u_{ij} increases on the fast time scale and approaches 1 during a brief period of time when oscillator (i,j) jumps up because of Rule b. The parameters λ and η can be chosen so that the activation speed of u_{ij} is about the same as that of x_{ij} itself. Thus, when (i,j) is ready to influence its neighbors u_{ij} has already recorded the fact that (i,j) is recruited by Rule b. The slow decay of u_{ij} ensures that its activation persists while (i,j) is active. Thus N_2^a in Eq. (4) can be calculated as intended. Note that, because of the Heaviside function in Eq. (9), u_{ij} is not automatically activated when δ_{ij}^b is 1. The latter is possible even when (i,j) has just jumped to the silent phase and no oscillator in $N(i,j)$ is active (see Eqs. (5) and (6)). Also, δ_{ij}^b may switch its value between 0 and 1 when (i,j) is active. The Heaviside function simplifies the dynamics of u_{ij}.

Now consider an oscillator block that starts to jump down to the silent phase. Due to the binary nature of lateral coupling and the fact that the oscillators of the block may stay on two different cubics with $S_{ij} = 1 - W_z$ or $S_{ij} = -W_z$, there can be small differences in jumping time for these oscillators. Note that the phase differences within the block will be compressed when the block jumps up again in the next cycle. Moreover, before the last oscillator of the block jumps down, no oscillator in other block can jump up. This creates a phase separation between this block and the other oscillators of the network. As proven by Terman and Wang (1995, Theorem 4.3.1), once the block is separated from other oscillators it remains separated.

As done in Terman and Wang (1995) and Wang and Terman (1997), our above analysis is based on the assumption that segments corresponding to oscillator blocks are well-defined from an input image. It is difficult to analyze LEGION networks for arbitrary input images. The best way to observe LEGION behavior for arbitrary images is through algorithmic approximations, as we provide in Section 4. To summarize, our system behaves similarly as the LEGION network of Wang and Terman (1997) in terms of oscillatory dynamics. As demonstrated later, our new definition of S_{ij}, however, leads to very different segmentation results.

To illustrate the behavior of our system, we have numerically simulated a 50×50 oscillator network. The input image to the network, shown in Fig. 4(A), has the same size as the network so that an oscillator corresponds to a pixel. This noisy image contains three regions. Because of the image noise, the original LEGION algorithm of Wang and Terman (1997) cannot produce successful segmentation of the image (see Section 5 for more discussion). In our simulation all oscillators are simulated with $I = 0.2$. The system (1a)–(10) are solved using the fourth-order Runge–Kutta method with the following parameter values: $\epsilon = 0.004$, $\alpha = 4.0$, $\beta = 0.1$, $\rho = 0.02$, $\phi = 3.0$, $\lambda = 3.0$, $\eta = 0.5$, $R_a = 5$, $R_0 = 7$, $\omega_\mu^a = 2.5$, $\omega_\sigma^a = 4.0$, $\omega_\mu^b = 3.0$, $\omega_\sigma^b = 4.5$. Leaders are selected using the method introduced in Section 4 with $R_p = 7$ and $T_p = 12.0$. In the numerical simulation, Heaviside functions, evaluated in every integration step, may change their values in one step. In this sense they act in a super-fast time scale, in addition to the slow and the fast time scales that characterize the dynamics of a relaxation oscillator. The Heaviside functions induce 'hops' between three cubic positions: $S_{ij} = 0$, $1 - W_z$, or $-W_z$, which have been discussed earlier.

The phases of all the oscillators on the network are randomly initialized. Fig. 4(B)–(E) show the instantaneous activity (snapshot) of the network at various stages of dynamic evolution. The diameter of each black circle represents the x activity of the corresponding oscillator; if the range of x values of all the oscillators is given by $x_{\min}$ and $x_{\max}$, the diameter corresponding to an oscillator is set to be proportional to $(x - x_{\min})/(x_{\max} - x_{\min})$. Fig. 4(B) shows the snapshot at the beginning of the dynamic evolution, indicating the random initial conditions. Fig. 4(C) shows the snapshot shortly after Fig. 4(B). One can clearly see the effect of synchrony and desynchrony: all the oscillators that correspond to the left region are active while the oscillators corresponding to the rest of the image are silent. Thus, the left region is segmented from the rest of the image. A short time later, as shown in Fig. 4(D), the oscillators corresponding to the middle region are active and separated from the rest of the image. Fig. 4(E) shows another snapshot after Fig. 4(D). At this time, the right region has its turn to be activated and separates from the rest of the input. This successive 'pop-out' of the segments continues in a stable periodic fashion unless the input image is withdrawn. Except for the two pixels near the lower left corner of the middle square, the image is accurately segmented.

To indicate those oscillators that jump up because of Rule b only, Fig. 4(F)–(H) show the snapshots of u values at the time instants that correspond to those for Fig. 4(C)–(E), respectively. Note that u values are between

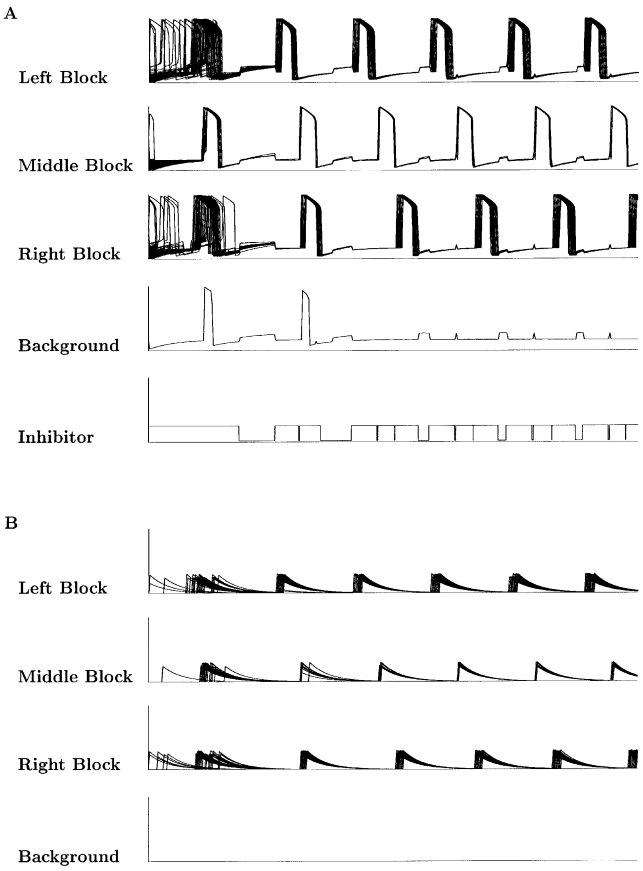

Fig. 5. Temporal traces of all oscillators. (A) The upper three traces show the combined x activities of the three oscillator blocks corresponding to three segmented regions. The fourth trace shows the background activity, and the bottom trace the global inhibitor. (B) The four traces show the combined u activities of the three oscillator blocks and the background. The entire simulation takes 32,000 integration steps.

0 and 1, and they all initialized to 0. From these figures, it is clear that these oscillators tend to be those near a region boundary.

Fig. 5 shows the temporal evolution of all the oscillators. The activities of the oscillators of each segment are combined together as one trace in the figure, and for the background. If any group of oscillators synchronizes, they together appear like a single oscillator. The three upper traces in Fig. 5(A) represent the activities of the three oscillator blocks corresponding to the three regions of Fig. 4(A). The fourth trace represents the background, which, in this case, consists of just one oscillator (the one near the right boundary of the square in Fig. 4(E)). The bottom trace in Fig. 5(A) represents the activity of the global inhibitor. From Fig. 5(A), it is easy to see that the noisy image of Fig. 4(A) is completely segmented into three regions, in about three cycles. Additionally, synchrony is not as tight as that shown in Wang and Terman (1997) with dynamical weight normalization. All of these are consistent with our analysis. Fig. 5(B) provides the corresponding traces of u values.

4. Segmentation algorithm

For an $M \times N$ gray-level image, a network consisting of $M \times N$ oscillators and a global inhibitor is used to perform segmentation, whereby each oscillator corresponds to one pixel of the image. Segmentation of real images with large numbers of pixels involves integrating a large number of differential equations, which results in prohibitively expensive computation on a serial personal computer. Similar to Wang and Terman (1997), we extract a segmentation algorithm from the dynamics of our network to reduce numerical computation. Basically, the algorithm preserves the essential properties of relaxation oscillators, e.g. two time scales, and network dynamics, e.g. synchronization and desynchronization.

For computational efficiency, we make several approximations to the dynamics of our network in the segmentation algorithm, as similarly done by Wang and Terman (1997). These include the selection of leaders, one time step to jump an enable oscillator, one time step for alteration between the active phase and the silent phase, and simultaneous jumping down all of the active oscillators belonging to the same region. Computation of overall coupling in Eq. (10) can be simplified as follows: Rule a is first evaluated and only if its conditions are unsatisfied Rule b is further evaluated. In this way, Rule a has precedence and Rule b may not need to be evaluated. The dynamical coupling structures are also generated accordingly. Similarly, we can also simplify the computation of u_{ij}. As mentioned before, the variable is used to distinguish between two classes of active oscillators. Therefore, u_{ij} is only useful when oscillator (i,j) is in the active phase. In our algorithm, u_{ij} is treated as binary and initialized to zero, and it is set to unity once a silent oscillator is activated only by Rule b.

Based on the lateral potential concept (Wang & Terman, 1997), here we use an alternative method to select leaders for our network. We observe that oscillators near the center of a homogeneous region tend to have a high lateral potential. Based on this observation, we calculate the standard deviation on a fixed neighborhood defined as

$$N_p(i,j) = \{(k,l) | i - R_p \le k \le i + R_p, j - R_p \le l \le j + R_p\},$$

where R_p determines the size of the neighborhood. Oscillator (i,j) is designated as a leader if and only if $\sigma_{N_p(i,j)} \le T_p$, where T_p is a threshold. Intuitively, a larger value of R_p results in leaders that are generated from larger homogeneous regions, while a smaller value of T_p results in leaders from more homogeneous regions.

All the parameters, including R_p and T_p for selecting leaders, R_a and R_0 for determining pixel and region neighborhoods, and the tolerance parameters, ω_μ^a, ω_σ^a, ω_μ^b, ω_σ^b, for testing the similarity, are specified in advance. The algorithm is summarized as follows:

```
1. Initialization
```

1.1 Set $z(0) = 0$.
1.2 For every oscillator (i, j) in the network, set $u_{ij}(0) = 0$.
1.3 Place all the oscillators randomly in the silent phase so that for every oscillator (i, j), $LC_x < x_{ij}(0) < LK_x$.
1.4 Find leaders
$p_{ij} = H(T_p - \sigma_{N_p(i,j)})$.
2. Find one oscillator (i, j) satisfying the following conditions:
(a) $p_{ij} = 1$; (b) $x_{ij}(t) \geq x_{kl}(t)$, where oscillator (k, l) in the silent phase and $p_{kl} = 1$. Then,
$x_{ij}(t+1) = RK_x$; $z(t+1) = 1$
{jump up to the active phase}
$x_{kl}(t+1) = x_{kl}(t) + (LK_x - x_{ij}(t))$,
for $(k, l) \neq (i, j)$ and $p_{kl} = 1$.
In this step, the leader in the silent phase, which is closest to LK, is selected. This leader jumps up to the active phase, and all the other leaders move towards LK.
3. Iterate until all the leaders have been to the active phase.
If ($x_{ij}(t) = RK_x$ and $z(t) > z(t-1)$)
$x_{ij}(t+1) = x_{ij}(t)$
{stay in the active phase}
else if ($x_{ij}(t) = RK_x$ and $z(t) \leq z(t-1)$)
$x_{ij}(t) = LC_x$; $z(t+1) = z(t) - 1$
{jump down to the silent phase}
If $z(t+1) = 0$, go to step 2.
else {oscillator (i, j) is silent}
Form $N_2^a(i, j)$; Evaluate $S_{ij}(t+1)$ using Rule a
If $S_{ij}(t+1) > 0$
$x_{ij}(t+1) = RK_x$; $z(t+1) = z(t) + 1$
{jump to the active phase}
else
Form $N_1^b(i, j)$; and $N_2^b(i, j)$;
Evaluate $S_{ij}(t+1)$ using Rule b
If $S_{ij}(t+1) > 0$
$x_{ij}(t) = RK_x$;
{jump up to the active phase}
$z(t+1) = z(t) + 1$
$u_{ij}(t+1) = 1$
else
$x_{ij}(t+1) = x_{ij}(t)$
{stay in the silent phase}

5. Simulation results

We first apply our algorithm to a typical noisy synthetic image illustrated in Fig. 6(A), which was used by Sarkar and Boyer (1991). The image contains four parallelograms and an image background. Due to substantial noise, the LEGION algorithm of Wang and Terman (1997) fails to segment the image into five meaningful regions. This is because their segmentation is based on pair-wise coupling within a fixed neighborhood (eight nearest-neighbor coupling). As a result, grouping is vulnerable to noise. When applied to Fig. 6(A), their algorithm shows two kinds of error: breaks a region into fragments and conjoins regions with the image background. The same kinds of error occur to the image of Fig. 4(A). In the simulation using our algorithm, an oscillator network consisting of 230×240 oscillators and a global inhibitor is used so that each oscillator corresponds to one pixel in the image. After the image is presented, the algorithm quickly produces segregated regions at different time steps. Fig. 6(B)–(F) display the five segments that have been produced alternately. In these resulting images, a black pixel indicates that its corresponding oscillator stays in the active phase and a gray pixel indicates in the silent phase. The first four segments correspond to the four parallelograms and the last segment corresponds to the image background. In addition, all the excitable oscillators constitute the background corresponding to those parts with high intensity variations. To display the complete scenario of segmentation, we use the so-called *gray map* convention, where each gray level indicates a distinct segment (Wang & Terman, 1997). Fig. 6(G) is the resulting gray map of this simulation, where the background is indicated by the black scattered areas. It is evident that our algorithm segments the noisy image into five meaningful regions.

The next image that our algorithm is tested on is an aerial image shown in Fig. 7(A), a portion of a large aerial image to be used later. There are several objects, including water body, forest, island, and land. Fig. 7(B) and (C) illustrate the edge maps produced by a Canny edge detector (Canny, 1986) with two different scales. The edge maps indicate that the image is noisy and a single scale hardly extracts all salient features. Therefore, it is difficult to segment the image with an edge-based approach. Such difficulty remains even when automatic scale selection methods (Elder & Zucker, 1998; Lindeberg, 1998) are applied. Shortly after the image is presented to our network, five segments pop out at different times, which are shown in Fig. 7(D)–(H). The five segments correspond to the island, the water body, the forest, and two parts of land. Fig. 7(I) shows the entire segmentation result by a gray map, where the background is indicated by the black areas. Our algorithm successfully segments the noisy image into the meaningful regions.

Extracting a hydrographic object refers to grouping the pixels corresponding to a water body, e.g. river, together and putting other regions into the background. It is a major part of a geographic information system. Hydrographic objects tend to be more homogeneous in comparison with other kinds of objects. When our network is applied, we can utilize this property to generate leaders so

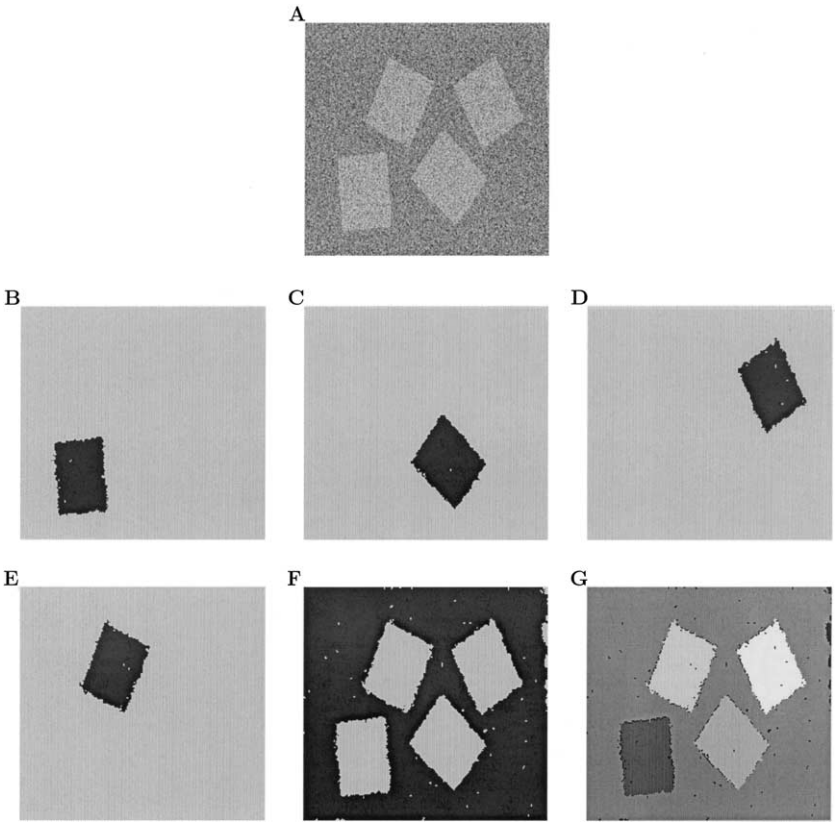

Fig. 6. Segmentation result for a synthetic image. The parameters used in this simulation are $R_p = 7$, $T_p = 12.0$, $R_a = 5$, $R_0 = 7$, $\omega_\mu^a = 2.5$, $\omega_\sigma^a = 4.0$, $\omega_\mu^b = 3.0$, and $\omega_\sigma^b = 4.5$. (A) Original image consisting of 230×240 pixels. (B)–(F) Five segments produced alternately by the network. (G) A gray map showing the entire segmentation result.

that only oscillators corresponding to hydrographic objects can be identified as leaders. Thus, other objects are naturally put into the background. Our algorithm has been applied to this task. Here we demonstrate its performance using two aerial images provided by the US Geological Survey. Fig. 8(A) shows an aerial image with a river that has several branches. A portion of the image is used in Fig. 7. As shown before, the boundaries between the river and its banks are quite noisy. Fig. 8(B) shows the extraction result. To clearly display the result, the river extracted is marked as white and superimposed on the original image. As the figure illustrates, the river is extracted well and the islands, including a small one, are preserved precisely. Fig. 8(C) shows another aerial image containing a river. In the image, the docks cause the river boundaries to be irregular. Fig. 8(D) shows our extraction result. These simulation results demonstrate that our algorithm performs well for hydrographic object extraction.

Magnetic resonance imaging (MRI) techniques have been extensively applied in medicine. However, automatic segmentation of MRI images is a challenging task since objects in an MRI image are usually nonhomogeneous and have low contrasts. We use MRI images of human heads to further test our algorithm. Fig. 9(A) is a midsagittal MRI image, and our algorithm segments the image into 29 regions plus a background, as shown in the gray map of Fig. 9(B). The background is indicated by the black areas. Many salient regions are separated such as the cerebral cortex, the cerebellum, the corpus callosum, the fornix, a part of the bone marrow, and several other anatomical structures. The next image is another sagittal section shown in Fig. 9(C). Fig. 9(D) shows the segmentation result, and the image is segmented to 28 regions plus a background indicated by the black areas. Similarly, salient regions segregated include the cortex, the cerebellum, the lateral ventricle, the eye ball, and the sinus. In particular, the cerebellum is segregated with good accuracy though it has a low contrast with the adjacent cortex. These two MRI images have been used to test the original LEGION algorithm (Wang & Terman, 1997), and our segmentation results are better. Our segments do not contain 'pepper and salt' like holes. Also, the cerebellum is segmented better for both Fig. 9(A) and (C) in our results than in theirs. Fig. 9(E) is an image of a coronal section, and Fig. 9(F) shows its segmentation results with 33 regions plus a background. Salient regions segmented include the cortex and the cerebellum. Fig. 9(G) shows a horizontal section of human head, and Fig. 9(H) shows the segmentation result, where the image is segmented into 24 regions plus a background. Salient regions segregated include two hemispheres, two eyeballs, and the brain ventricle near the center.

We have compared our network with several recent

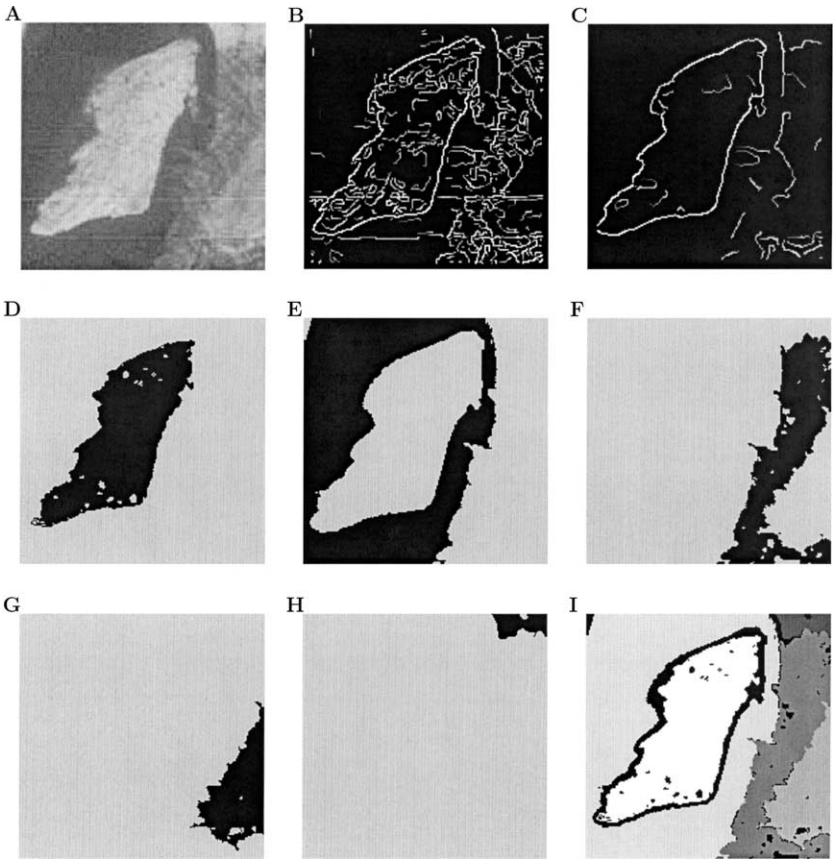

Fig. 7. Segmentation result for an aerial image. The parameters used in this simulation are $R_p = 5$, $T_p = 7.0$, $R_a = 2$, $R_0 = 5$, $\omega_\mu^a = 2.0$, $\omega_\sigma^a = 3.8$, $\omega_\mu^b = 3.5$, and $\omega_\sigma^b = 4.8$. (A) Original image consisting of 181×181 pixels. (B)–(C) Two edge maps generated from the original image by the Canny edge detector. (D)–(H) Five segments produced by the network. (I) A gray map showing the entire segmentation result.

image processing methods for segmenting the above images. These methods include a multiscale adaptive segmentation method (Caelli & Bischof, 1997) and several nonlinear smoothing methods (Weickert, 1997) to remove noise prior to segmentation. Due to limited space, we report only comparative results for two images that have been used for testing our algorithm. The two images belong to two different types: an aerial image and an MRI image. As suggested by Caelli and Bischof (1997), we use three scales in their algorithm. Fig. 10(A) and (B) show gray maps of the segmentation results by the multiscale adaptive method. In Fig. 10(A), the aerial image in Fig. 7(A) is segmented to 48 regions. The river and the island are segmented, and the boundary of the segmented island is reasonably accurate. However, the boundary between the river and its bank to the left is not correctly segmented due to a significant amount of noise on the image. In Fig. 10(B), the MRI image in Fig. 9(E) is segmented to 98 regions, which include the cerebral cortex, the cerebellum, and parts of the extracranial tissue. Although much of the cerebellum is extracted, its right side is shrunk, and so is the segmented cortex. Furthermore, we report a comparison with a recent nonlinear smoothing algorithm—edge-enhanced diffusion (Weickert, 1996)—followed by a Fisher-distance based region growing algorithm, where the mean and variance of image intensities over a 5×5 neighborhood are used for growing (Haralick & Shapiro, 1985). In simulations, we determine parameter values in both smoothing and region growing by searching for the parameter space as suggested in their original work, and we report only the best results here. Fig. 10(C)–(F) shows the smoothed images and the corresponding segmentation results. In Fig. 10(D), the aerial image is segmented into five regions, and several black scattered areas which together would correspond to the background in LEGION segmentation. The island is accurately segmented, but the boundary between the river and its bank is not. In Fig. 10(F), the MRI image is segmented to 31 regions plus back scattered areas. Like ours, most of the significant regions are successfully extracted by this method. Due to noise removal by smoothing, segmented regions by this method tend to appear smoother; on the other hand, our network tends to yield more precise boundaries between different regions. Note also that nonlinear smoothing suffers from the termination problem, which generally requires human intervention to select best results. In summary, the comparisons show that our segmentation algorithm performs better than the multiscale adaptive segmentation method, and at a comparable

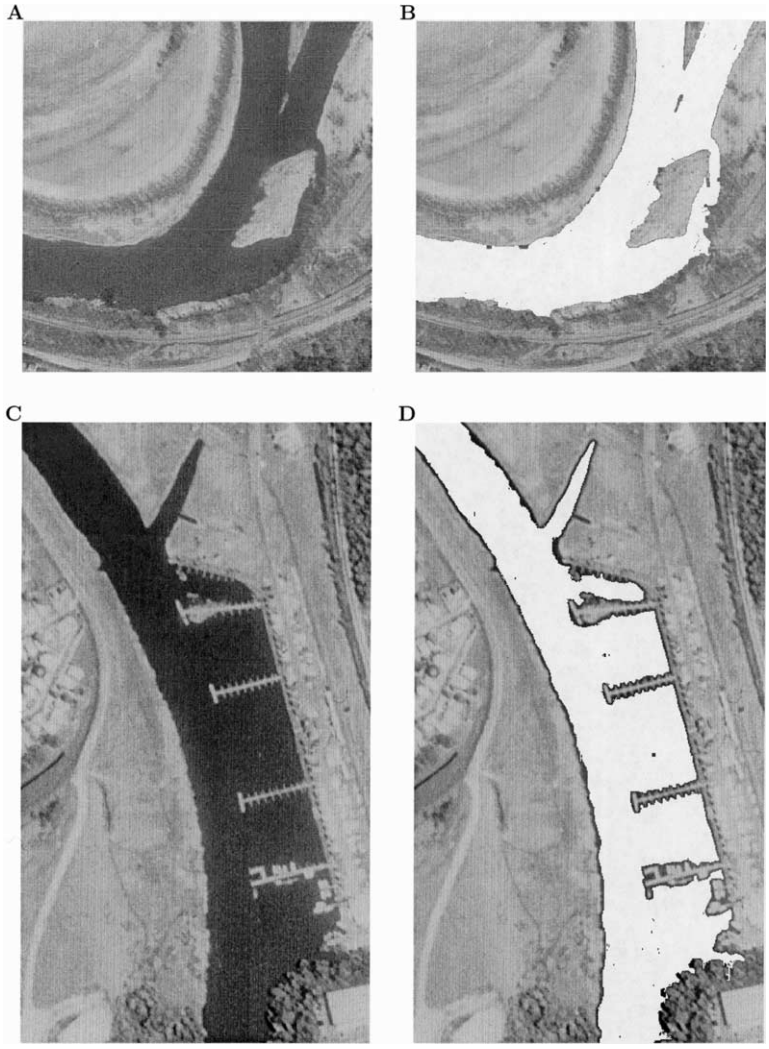

Fig. 8. Simulation results for hydrographic object extraction. The parameters used in simulations are $R_p = 25$, $T_p = 4.0$ and the other parameters are the same as listed in the caption of Fig. 7. (A) An aerial image consisting of 640×640 pixels. (B) The extraction result for the image in (A). (C) Another aerial image consisting of 378×670 pixels. (D) The extraction result for the image in (C).

level with an effective nonlinear smoothing technique plus subsequent segmentation.

6. Discussion

The major difference between our network and the LEGION network of Wang and Terman (1997) lies in how to determine overall coupling, S_{ij}, in Eq. (1a). For gray-level image segmentation, pair-wise coupling is used in their network so that the coupling strength between two neighboring oscillators is determined by the similarity of two corresponding pixels (Wang & Terman, 1997). Such pair-wise coupling leads to a fixed coupling structure. Unlike their network, we adopt dynamical coupling structures and use two grouping rules to recruit a silent oscillator in two different ways. Thus, our network has the property that the labeling of an oscillator is considered at any time through two grouping rules on the basis of dynamical coupling structures. This property gives rise to robust performance, and does not exist in their network. On the other hand, both their network and our network share the property of local coupling that plays a fundamental role in image segmentation. As pointed out in Section 2, our network shares the same properties of LEGION dynamics that has been extensively analyzed by Terman and Wang (1995), e.g. fast synchronization within an oscillator group representing each object and desynchronization among oscillator groups representing different objects. Therefore, our neural oscillator network preserves the biological links of LEGION argued by Wang and Terman (1997).

LEGION has been used to handle different types of real images, e.g. range image segmentation (Liu & Wang, 1999), by setting up proper coupling between neighboring oscillators based on a task-oriented similarity measure. In our network, the simple statistics—mean and standard deviation are now used only to test the similarity for gray-level image segmentation. We expect that our network can also

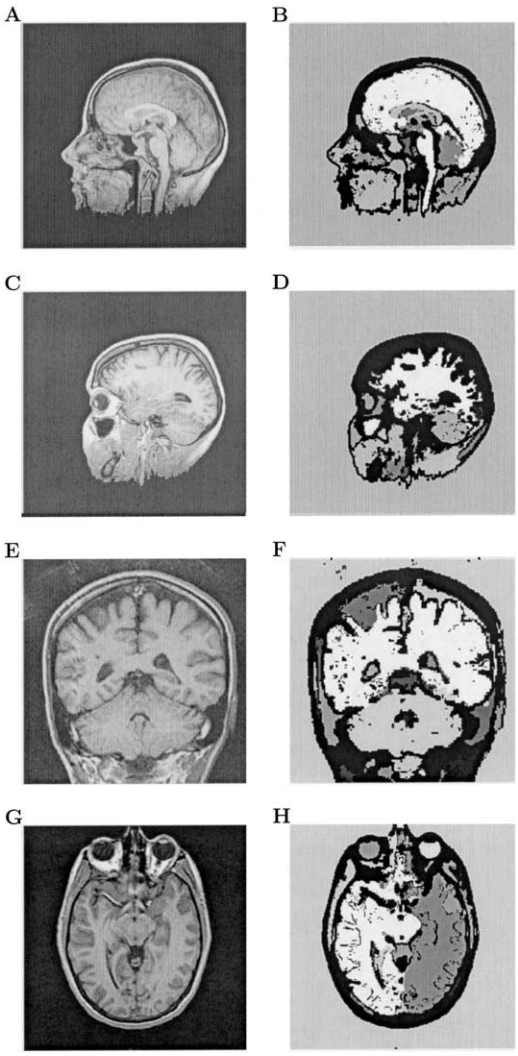

Fig. 9. Segmentation results for MRI images. The parameters used in all the simulations are $R_p = 3$, $T_p = 5.0$, $R_a = 1$, $R_0 = 4$, $\omega_\mu^a = 2.2$, $\omega_\sigma^a = 2.5$, $\omega_\mu^b = 3.0$, and $\omega_\sigma^b = 3.1$. (A) A midsagittal MRI image consisting of 257×257 pixels. (B) The segmentation result for the image in (A). (C) A sagittal MRI image consisting of 257×257 pixels. (D) The segmentation result for the image in (C). (E) An MRI image of a coronal section consisting of 175×175 pixels. (F) The segmentation result for the image in (E). (G) An MRI image of a horizontal section consisting of 256×256 pixels. (H) The segmentation result for the image in (G).

be extended to deal with other types of images through the use of a proper task-oriented measure.

As discussed elsewhere (Terman & Wang, 1995; Wang & Terman, 1997), LEGION with a fixed set of parameters segments a limited number of objects, typically ranging from five to seven. The capacity property of LEGION resembles the phenomenon that human attention (and short-term memory) is capacity-limited. The segmentation capacity of LEGION is determined by the ratio of the oscillation period to the duration of the active phase (see Fig. 5(A)), and it can be calculated analytically (Linsay & Wang, 1998). Thus, to segment an image with more objects than the capacity would require some hierarchical organization, whereby a congregate segment with more than one object would be further segmented at a later time. To facilitate segmentation of real images, however, the segmentation capacity is removed from the algorithmic abstraction in Section 4, and our algorithm can segment an arbitrary number of objects.

A salient feature of our network is its robust performance in noisy image segmentation. Noise is unavoidable in real image processing. In order to achieve good segmentation performance, a smoothing technique is often employed for noise removal prior to the use of a segmentation method, which results in a two-stage segmentation approach. In general, smoothing techniques can be classified into two categories: linear and nonlinear smoothing (Schalkoff, 1989). However, both of them have different problems in practice. Linear smoothing causes important discontinuities to be blurred so that meaningful objects cannot be separated during segmentation. The general idea behind nonlinear smoothing is to adapt pixel values to the local properties of an image. Although the use of nonlinear smoothing avoids, to a good extent, the problem existing in linear smoothing, it causes the termination problem (Perona & Malik, 1990; Saint-Marc, Chen, & Medioni, 1991; Weickert, 1997). Unlike the two-stage segmentation approach, in our network no iterative smoothing operation is involved so that the termination problem does not occur. This feature distinguishes our approach from those two-stage segmentation methods.

Our system expands a segment based on the information from a local front region of the segment, not from the whole segment. As a result, it segments an image roughly into smoothly varying regions. The structure of smoothness within each region depends on the choice of system parameters (see below). Our use of dynamical neighborhoods has some resemblance to multiscale image analysis. In terms of grouping, Rule a measures similarity in two relatively fixed image neighborhoods: N_1^a and N_2^a. Such neighborhoods define certain spatial scales, although N_2^a is not really fixed in our method. Rule b measures similarity between two groups of pixels: N_1^b and N_2^b, both of which vary considerably depending on the grouping process and the shape of the currently expanding segment. Because of the dynamic nature of neighborhood structures, our method differs significantly from multiscale approaches. A number of recent studies proposed systematic methods for selecting scales at different image locations for local feature detection (Elder & Zucker, 1998; Lindeberg, 1998). These studies aim at detecting local features at an optimal scale according to some criteria. Thus features of both coarse and fine scales may be detected simultaneously. We address a different issue: how to organize local features into larger structures—segments (see Fig. 10 for a comparison with a multiscale segmentation method). It is not clear whether such organization becomes easier when all of the local edges in an image are made simultaneously available by automatic scale control (Elder & Zucker, 1998).

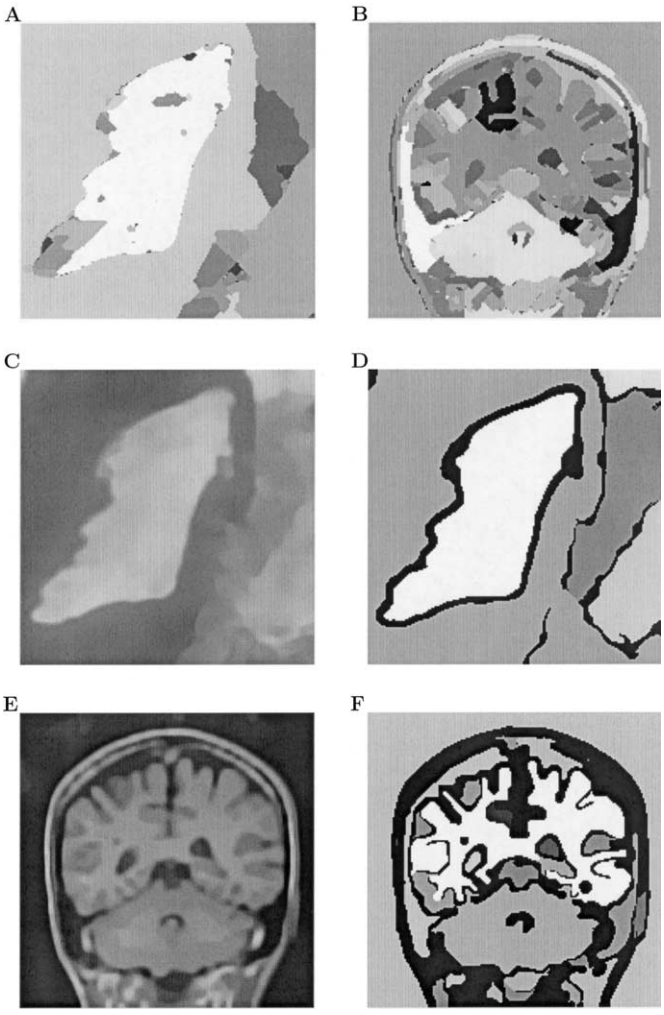

Fig. 10. Comparative results for two images. (A) The segmentation result for the image in Fig. 7(A) by a multiscale adaptive segmentation method. (B) The segmentation result for the image in Fig. 9(E) by the multiscale adaptive segmentation method. (C) The smoothing result for the image in Fig. 7(A) by an edge-enhanced diffusion algorithm, where the contrast parameter is 1.5 and the scale parameter is 2.5, and the result corresponds to iteration 25. (D) The segmentation result for the image in Fig. 7(A) based on the smoothed image in (C). (E) The smoothing result for the image in Fig. 9(E) by the edge-enhanced diffusion algorithm, where the contrast parameter is 1.0 and the scale parameter is 2.0, and the result corresponds to iteration 40. (F) The segmentation result for the image in Fig. 9(E) based on the smoothed image in (E).

There are a number of parameters in our system, and these parameters fall into two kinds. The first kind contains parameters pertinent to oscillator dynamics: those in Eqs. (1a), (1b), (2), and (9). With an understanding of oscillatory dynamics, these parameters can be fixed. In fact, they are absorbed and do not appear in our segmentation algorithm given in Section 4. The second kind contains parameters that specify neighborhoods and grouping rules. These parameters must be specified for a segmentation task. Therefore, how to effectively tune the second kind of parameters is an important issue. These parameters can be further classified into three categories. The first category includes two parameters, R_p and T_p, used to select leaders. As mentioned in Section 4, these parameters must be determined in a task-specific way, by desired properties of segments, such as degree of homogeneity and size. The next two parameters, R_a and R_0, constitute another category to determine the size of pixel and region neighborhoods. R_a is used to generate the pixel neighborhood N_1^a and the region neighborhood N_2^a. The selection of R_a depends upon image types. When the detailed structures are needed, R_a should be set to a small value, and vice versa. The other parameter R_0 determines the minimal number of activated oscillators that yields a robust estimation of local attributes, and it can be determined in a similar way. In our simulations, we fix the two parameters for a type of image and can achieve satisfactory results. Moreover, our empirical studies indicate that the performance of our network is relatively insensitive to detailed values of the two parameters. The last category consists of four parameters, ω_μ^a, ω_σ^a, ω_μ^b, and ω_σ^b. In general, a smaller tolerance value causes less grouping. As a result, more and smaller regions are generated. In contrast, a larger tolerance value results in fewer and larger regions. In our simulations, we use of the following heuristic methods to tune these

parameters. From the viewpoint of statistics, the two grouping rules described in Section 2 can be regarded as the hypothesis-test procedures (Cox, 1987); we test the hypothesis whether the statistics associated with a pixel represented by a silent oscillator are equal to those of a region represented by a group of active oscillators. A tolerance parameter may thus be viewed as a confidence interval. Suppose that the image satisfies a Gaussian distribution. Then, for example, the $100(1-\gamma)\%$ confidence interval for the mean on a pixel neighborhood can be determined by the quantity $t(n-1;\gamma/2)\sigma/\sqrt{n}$, where σ is the estimation of the standard deviation, n is the population size, γ is the error probability, and $t(n-1;\gamma/2)$ is the value of t-distribution in terms of γ and n. In general, we cannot assume the Gaussian distribution and the population size varies in the grouping process due to dynamical coupling. Therefore, the confidence interval can provide only some heuristics to determine the tolerance parameters. On the basis of the heuristic information, we always use larger ω_μ^b and ω_σ^b in Rule b than ω_μ^a and ω_σ^a in Rule a. In addition, the histograms of local mean and standard deviation for a fixed neighborhood can provide some useful information to select the parameters, ω_μ^a and ω_σ^a. To a certain extent, these heuristic techniques overcome the difficulty in tuning the tolerance parameters. However, a general method is not available, and some user tuning is still needed as in many other methods.

As pointed out before (Wang & Terman, 1997; Wilson & Spann, 1996), in general there is no unique answer for image segmentation due to its uncertain nature. For example, the same house should be treated as one segment in some tasks, and separated into multiple segments, such as roof, door, and windows, in some other tasks. This fundamental uncertainty dictates that no set of parameter values fits all tasks, or some parameter tuning must be conducted for any segmentation algorithm. On the other hand, natural imagery of a certain type, e.g. aerial imagery from similar imaging conditions, is of similar scale and statistical properties. These common properties can be viewed as prior knowledge for segmenting images of a certain type. Such prior knowledge is encoded by a proper set of parameter values in our algorithm, which leads to desired results as we have demonstrated in this paper. From a practical point of view, one can acquire prior knowledge by analyzing a small number of images (just one in many cases) from a collection of the same type, as done in our simulations. In terms of methodology, this is a supervised learning process: acquiring image regularity from examples. A recent study by Liu, Chen, and Wang (2001) shows that parameters comparable to ours for leader selection can be determined by training a multilayer perceptron. This suggests a promising direction for addressing the issue of parameter tuning. The key there is how to choose representative training samples. Similarly, if one treats segmentation as classification, parameter determination may also be regarded as an optimization problem, which would lead to an optimal set of values given statistical properties of each class (Saha & Udupa, 2001).

To conclude, we have presented a new method to image segmentation, which is based on dynamical coupling and oscillatory correlation. Our study demonstrates that robust segmentation performance can be achieved based on the idea of using dynamical neighborhoods, which is inherently consistent with the dynamics of relaxation oscillators. Our results lend further support to oscillatory correlation as a potential solution to scene segmentation and perceptual organization (Wang & Terman, 1997).

Acknowledgements

We would like to thank X.W. Liu for discussions and providing the aerial images, and N. Shareef for providing the MRI images. This work was supported in part by an NSF grant (IIS-0081058), an AFOSR grant (F49620-01-1-0027), an ONR Young Investigator Award (N00014-96-1-0676) to D.L.W, and an NSFC grant (60075017) to K.C.

References

Ahuja, N. (1996). A transform for multiscale image segmentation by integrated edge and region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*, 1211–1235.

Bhandarkar, S. M., Koh, J., & Suk, M. (1997). Multiscale image segmentation using a hierarchical self-organizing map. *Neurocomputing*, *14*, 241–272.

Caelli, T., & Bischof, W. (1997). *Machine learning and image interpretation*, New York: Plenum Press.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *8*, 679–698.

Cheriet, M., Said, J. N., & Suen, C. Y. (1998). A recursive thresholding technique for image segmentation. *IEEE Transactions on Image Processing*, *7*, 918–921.

Cox, C. P. (1987). *A handbook of introductory statistical methods*, New York: Wiley.

Elder, J. H., & Zucker, S. W. (1998). Local scale control for edge detection and blur estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*, 699–716.

Foresti, G., Murino, V., Regazzoni, C. S., & Vernazza, G. (1994). Grouping of rectilinear segments by the labeled Hough transform. *CVGIP: Image Understanding*, *58*, 22–42.

Geman, D., Geman, S., Graffigne, C., & Dong, P. (1990). Boundary detection by constrained optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*, 608–628.

Grossberg, S., & Wyse, L. (1991). A neural network architecture for figure-ground separation of connected scenic figures. *Neural Network*, *4*, 723–742.

Haddon, J. F., & Boyce, J. F. (1990). Image segmentation by unifying region and boundary information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*, 929–948.

Haralick, R. M., & Shapiro, L. G. (1985). Image segmentation techniques. *Computer Vision Graphics and Image Processing*, *29*, 100–132.

Horowitz, S. L., & Pavlidis, T. (1976). Picture segmentation by a tree traversal algorithm. *Journal of ACM*, *23*, 368–388.

Kohler, R. (1981). A segmentation system based on thresholding. *Computer Vision Graphics and Image Processing*, *15*, 319–338.

Kohonen, T. (1995). *Self-organizing maps*, Berlin: Springer.

Lindeberg, T. (1998). Feature detection with automatic scale selection. *International Journal of Computer Vision*, *30*, 77–116.

Linsay, P. S., & Wang, D. L. (1998). Fast numerical integration of relaxation oscillator networks based on singular limit solutions. *IEEE Transactions on Neural Networks*, *9*, 523–532.

Liou, S. P., Chiu, A. H., & Jain, R. C. (1991). A parallel technique for single-level perceptual organization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*, 317–325.

Liu, X. W., & Wang, D. L. (1999). Range image segmentation using a relaxation oscillator network. *IEEE Transactions on Neural Networks*, *10*, 564–573.

Liu, X. W., Chen, K., & Wang, D. L. (2001). Extraction of hydrographic regions from remote sensing images using an oscillator network with weight adaptation. *IEEE Transactions on Geoscience and Remote Sensing*, *39*, 207–211.

von der Malsburg, C. (1981). *The correlation theory of brain function*, Göttingen: Max-Planck-Institute for Biophysical Chemistry Internal Report 81-2.

von der Malsburg, C., & Schneider, W. (1986). A neural cocktail-party processor. *Biological Cybernetics*, *54*, 29–40.

Manjunath, B. S., & Chellappa, R. (1993). A unified approach to boundary perception: Edges, textures, and illusory contours. *IEEE Transactions on Neural Networks*, *4*, 96–108.

Milner, P. M. (1974). A model for visual shape recognition. *Psychological Review*, *81*, 521–535.

Mozer, M. C., Zemel, R. S., Behrmann, M., & Williams, C. K. I. (1992). Learning to segment images using dynamic feature binding. *Neural Computation*, *4*, 650–665.

Pal, N. R., & Pal, S. K. (1993). A review on image segmentation techniques. *Pattern Recognition*, *26*, 1277–1294.

Perona, P., & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*, 629–639.

van der Pol, B. (1926). On 'relaxation oscillations'. *Philosophical Magazine*, *2*, 978–992.

Revol, C., & Jourlin, M. (1997). A new minimum variance region growing algorithm for image segmentation. *Pattern Recognition Letters*, *18*, 249–258.

Saha, P. K., & Udupa, J. K. (2001). Optimum image thresholding via class uncertainty and region homogeneity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*, 689–706.

Saint-Marc, P., Chen, J. S., & Medioni, G. (1991). Adaptive smoothing: A general tool for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*, 514–529.

Sarkar, S., & Boyer, K. L. (1991). On optimal infinite impulse response edge detection filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*, 1154–1171.

Schalkoff, R. J. (1989). *Digital image processing and computer vision*, New York: Wiley.

Sejnowski, T. J., & Hinton, G. E. (1987). Separating figure from ground with a Boltzmann machine. In M. A. Arbib & A. R. Hanson, *Vision, brain, and cooperative computation* (pp. 703–724). Cambridge, MA: MIT Press.

Terman, D., & Wang, D. L. (1995). Global competition and local cooperation in a network of neural oscillators. *Physica D*, *81*, 148–176.

Wang, D. L., & Terman, D. (1995). Locally excitatory globally inhibitory oscillator networks. *IEEE Transactions on Neural Networks*, *6*, 283–286.

Wang, D. L., & Terman, D. (1996). *Image segmentation based on oscillatory correlation*, Columbus, OH: Ohio State University Center for Cognitive Science Technical Report No. 19.

Wang, D. L., & Terman, D. (1997). Image segmentation based on oscillatory correlation. *Neural Computation*, *9*, 805–836 for errata see *Neural Computation*, vol. 9, pp. 1623–1626, 1997.

Weickert, J. (1996). Theoretical foundations of anisotropic diffusion in image processing. *Computing*, *11*, 221–236.

Weickert, J. (1997). A review of nonlinear diffusion filtering. In B. Romeny, L. Florack, J. Koenderink & M. Viergever, *Scale-space theory in computer vision* (pp. 3–28). Berlin: Springer.

Wilson, R., & Spann, M. (1996). *Image segmentation and uncertainty*, New York: Wiley.

Zhu, S. C., & Yuille, Y. (1996). Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*, 884–900.

Zucker, S. W. (1976). Region growing: Childhood and adolescence. *Computer Vision Graphics and Image Processing*, *5*, 382–399.