Segmentation of Medical Images Using LEGION

Naeem Shareef, DeLiang L. Wang,* *Member, IEEE*, and Roni Yagel, *Member, IEEE*

***Abstract*—Advances in visualization technology and specialized graphic workstations allow clinicians to virtually interact with anatomical structures contained within sampled medical-image datasets. A hindrance to the effective use of this technology is the difficult problem of image segmentation. In this paper, we utilize a recently proposed oscillator network called the locally excitatory globally inhibitory oscillator network (LEGION) whose ability to achieve fast synchrony with local excitation and desynchrony with global inhibition makes it an effective computational framework for grouping similar features and segregating dissimilar ones in an image. We extract an algorithm from LEGION dynamics and propose an adaptive scheme for grouping. We show results of the algorithm to two-dimensional (2-D) and three-dimensional (3-D) (volume) computerized topography (CT) and magnetic resonance imaging (MRI) medical-image datasets. In addition, we compare our algorithm with other algorithms for medical-image segmentation, as well as with manual segmentation. LEGION's computational and architectural properties make it a promising approach for real-time medical-image segmentation.**

***Index Terms*— LEGION, medical images, oscillatory correlation, segmentation.**

I. Introduction

ADVANCES in imaging, visualization, and virtual environments technology are allowing the clinician to not only visualize, but also to interact with a virtual patient [32], [33]. In many cases, anatomical information contained within sampled image datasets is essential to clinical tasks. One typical example is in surgery where presurgical planning and postoperative evaluations are not only enhanced, but in many cases depend upon sampled image data for successful results [12], [26]. A number of imaging modalities are currently in widespread clinical use for anatomical and physiological imaging. Among them, two commonly used devices to image bony structures and soft tissue are computerized tomography (CT) and magnetic resonance imaging (MRI), respectively. Datasets are realized as a sequence of two-dimensional (2-D) cross sectional slices that together represent the three-dimensional (3-D) sample space. The entire image stack may be viewed as a 3-D array of scalar (or vector) values, called a volume, where each voxel (volume pixel) represents a measured physical quantity at a location in space. Advances in the fields of surface and volume graphics now make it possible to render a volume dataset with high image quality using lighting and shading models. Graphic workstations equipped with specialized hardware and texture-mapping capabilities show promise for real-time rendering [32]. Currently, clinicians view images on photographic sheets containing adjacent image slices, and must mentally reconstruct 3-D anatomical structures. Even though clinicians have developed the necessary skills to make use of this presentation, computer-based tools will allow for a higher level of interaction with the data.

A major hurdle in the effective use of this technology is the accurate identification of anatomical structures within the volume. Computer vision literature typically identifies three processing stages before object recognition: image enhancement, feature extraction, and grouping of similar features. In this paper we address the last step, image segmentation, where pixels are grouped into regions based on image features. The goal is to partition an image into pixel regions that together represent objects in the scene. Segmentation is a very difficult problem for general images, which may contain effects such as highlights, shadows, transparency, and object occlusion. On the other hand, sampled image datasets lack these effects with a few exceptions. One such exception is ultrasound datasets which may still contain occluded objects. Challenges to segment sampled image datasets involve handling noise artifacts introduced during the acquisition process and dataset size. With new imaging technology, increasing sizes of volume datasets are an issue for most applications. For example, a medium-size dataset with dimensions $256 \times 256 \times 125$ contains over eight million voxels. The MRI, CT, and color-image datasets for an entire human male cadaver from the National Library of Medicine's Visible Human Project [1] require approximately 15 Gbyte of storage. An additional challenge is that objects may be arbitrarily complex in terms of size and shape.

Many segmentation methods proposed for medical-image data are either direct applications or extensions of approaches from computer vision. image-segmentation algorithms can be classified in many ways [14], [21], [34]. We identify three broad classes that divide algorithms to segment sampled image data: manual, semiautomatic, and automatic. Reviews of algorithms from each class can be found in [10] and [24]. The image-segmentation algorithm presented in this paper is a

Manuscript received September 23, 1997; revised September 1, 1998. This work was supported in part by the ONR under Grant N00014-93-1-0335, by the NSF under Grant IRI-9423312, and by the DOD under Grant USAMRDC 94228001. The work of D. L. Wang was supported in part by a Young Investigator Award from the ONR. The Associate Editor responsible for coordinating the review of this paper and recommending its publication was C. R. Meyer. *Asterisk indicates corresponding author*.

N. Shareef and R. Yagel are with the Department of Computer and Information Science, Ohio State University, Columbus, OH 43210 USA (e-mail: shareef@cis.ohio-state.edu; yagel@cis.ohio-state.edu.).

*D. L. Wang is with the Department of Computer and Information Science and the Center for Cognitive Science, Ohio State University, Columbus, OH 43210 USA (e-mail: dwang@cis.ohio-state.edu).

Publisher Item Identifier S 0278-0062(99)02025-X.

semiautomatic approach. The manual method requires a human segmenter with knowledge of anatomy to use a graphical software tool to outline regions of interest. Obviously this method produces high-quality results, but is time consuming and tedious.

Semiautomatic methods require user interaction to set algorithm parameters, to perform initial segmentation, or to select critical features. They can be classified according to the space in which features are grouped together [14]. Measurement space methods map and cluster pixels in a feature space. A commonly used method is global thresholding [14], [15], [24], where pixel intensities from the image are mapped into a feature space called a histogram. Thresholds are chosen at valleys between pixel clusters so that each pair represents a region of similar pixels in the image. This works well if the target object has distinct and homogeneous pixel values, which is usually the case with bony structures in CT datasets. On the other hand, spatial information is lost in the transformation, which may produce disjoint regions.

Spatial-domain methods use spatial proximity in the image to group pixels. Edge-detection methods use local gradient information to define edge elements, which are then combined into contours to form region boundaries [10], [14]. For example, a 3-D version of the Marr–Hildreth operator was used to segment the brain from MRI data [6]. However, edge operators are generally sensitive to noise and produce spurious edge elements that make it difficult to construct a reasonable region boundary. Region growing methods [2], [4], [14], [34], on the other hand, construct regions by grouping spatially proximate pixels so that some homogeneity criterion is satisfied over the region. In particular, seeded-region-growing algorithms [2] grow a region from a seed, which can be a single pixel or cluster of pixels. Seeds may be chosen by the user, which can be difficult because the user must predict the growth behavior of the region based on the homogeneity metric. Since the number, locations, and sizes of seeds may be arbitrary, segmentation results are difficult to reproduce. Alternatively, seeds may be defined automatically, for example, the min/max pixel intensities in an image may be chosen as seeds if the region mean is used as a homogeneity metric [2]. A region is constructed by iteratively incorporating pixels on the region boundary. In addition, active-contour-based methods and neural-network-based classification methods have also been proposed to perform image segmentation. These methods will be described in some detail in Section V where comparisons are drawn.

In contrast to these approaches, we utilize a new biologically inspired oscillator network, called the locally excitatory globally inhibitory oscillator network (LEGION) [25], [30], [31], to perform segmentation on sampled medical-image datasets. The network was proposed based on theoretical and experimental considerations that point to oscillatory correlation as a representational scheme for the workings of the brain. The oscillatory correlation theory assumes that the brain groups and segregates visual features on the basis of correlation between neural oscillations [28], [30]. It has been found that neurons in the visual cortex respond to visual features with oscillatory activity (see [22] for review). Oscillations from neurons detecting features of the same object tend to synchronize with zero phase shift, whereas oscillations from different objects tend to desynchronize from each other. Thus, objects seem to be segregated in time. In addition to biological plausibility, LEGION exhibits unique computational advantages to be discussed in Section II.

We describe LEGION dynamics in Section II. In Section III, an algorithm derived from LEGION dynamics is described, which is based on a local neighborhood for grouping and chooses seeds automatically. In Section IV we present results of segmenting CT and MRI image datasets. In Section V we compare our algorithm with other segmentation algorithms for medical images and compare our results with manual segmentation. Finally, we give some concluding remarks in Section VI.

II. LEGION Model

Recent experimental evidence and earlier theoretical considerations point to neural oscillations in the visual cortex as a possible mechanism by which the brain detects and binds features in a visual scene. It is well known that neurons in the visual system respond to features such as color, orientation, and motion and are arranged in regular structures called columns and hypercolumns. In addition, these neurons respond only to stimuli from a particular part of the visual field. Recent experimental work shows persistent stimulus-dependent oscillations around 40 Hz in the visual cortex [[11], [13], [22]. Furthermore, synchronized behavior is observed between spatially separate neuronal groups. This supports earlier theoretical considerations [28] which suggest that cells acting as visual-feature detectors bind together through correlation of their firing activities. Previously proposed oscillator networks [5], [16], [23], represent an oscillatory event by a single-phase variable. These networks are limited when applied to image segmentation. Oscillations in these networks are built into the system rather than stimulus dependent. More substantially, these systems rely on fully connected network architecture to achieve synchronization, which results in indiscriminate grouping and loss of topological (spatial) information. LEGION is able to overcome these deficiencies with stimulus-dependent oscillations and fast- and long-range synchrony by using local connections. In addition, LEGION achieves fast desynchrony with a global inhibitory mechanism.

LEGION was proposed by Terman and Wang [25], [30] as a biologically plausible computational framework for image analysis and has been used successfully to segment binary and gray-level images [31]. It is a network of relaxation oscillators, each constructed from an excitatory unit x and an inhibitory unit y as shown in Fig. 1. Unit x sends excitation to unit y which responds by sending inhibition back. When external input stimulus I is continuously applied to x, this feedback loop produces oscillations. Neighboring oscillators are connected via mutual excitatory coupling, as well as the global inhibitor [see (1a) below].

LEGION is formally defined and analyzed in [25] and [31]. The behavior of each oscillator, indexed by i in a network, is

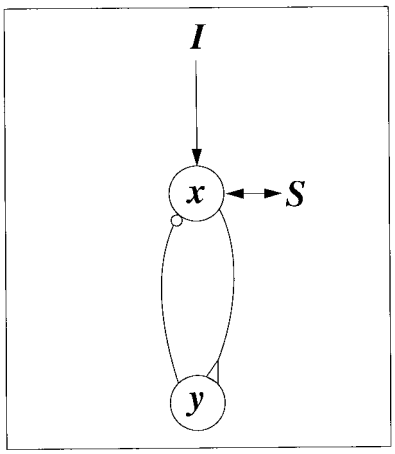


Fig. 1. Diagram of a single oscillator with an excitatory unit x and an inhibitory unit y in a feedback loop. A triangle indicates an excitatory connection and a circle indicates an inhibitory connection. I indicates external input and S indicates the coupling with the rest of the network.

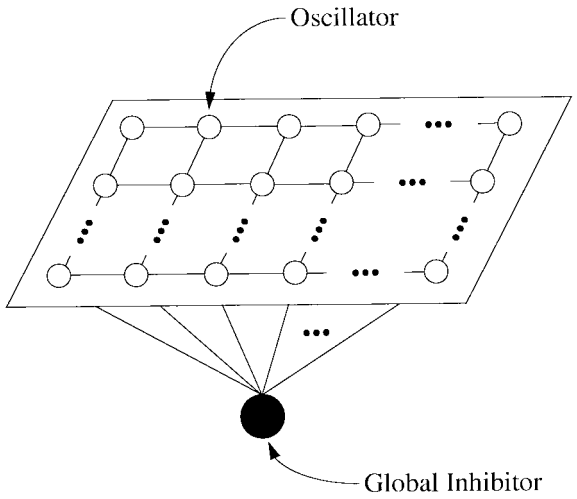


Fig. 2. A 2-D LEGION network with four-neighborhood connections. The global inhibitor is indicated with a black circle and is coupled with the entire network.

defined by the following equations:

$$\dot{x}_i = 3x_i - x_i^3 + 2 - y_i + \rho + I_i H(p_i + \exp(-\alpha t) - \theta) + S_i \tag{1a}$$

$$\dot{y}_i = \varepsilon(\gamma(1 + \tanh(x_i/\beta)) - y_i) \tag{1b}$$

$$\dot{p}_i = \lambda(1 - p_i) H\left[\sum_{k \in N_1(i)} T_{ik} H(x_k - \theta_x) - \theta_p\right] - \mu p_i \tag{2}$$

$$S_i = \sum_{k \in N_2(i)} W_{ik} H(x_k - \theta_x) - W_z H(z - \theta_z) \tag{3}$$

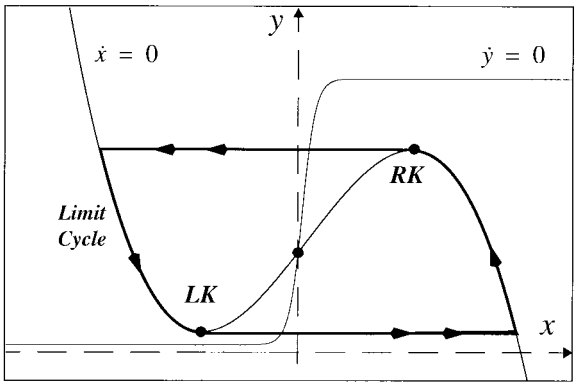
$$\dot{z} = \phi(\sigma_\infty - z). \tag{4}$$

The dynamics of x are defined in (1a) which contains a cubic function. The subtractive term y_i represents inhibition from unit y, I_i is external stimulus, and S_i represents excitatory coupling with neighboring oscillators and coupling with the global inhibitor. We call i stimulated if $I_i > 0$ and unstimulated if $I_i \leq 0$. The Heaviside step function H is defined as $H(a) = 1$ if $a \geq 0$ and $H(a) = 0$ if $a < 0$. The function H determines oscillatory behavior by multiplying I_i. The variable p_i is called the lateral potential of the oscillator and is used to suppress noisy regions. Parameter θ is set in the range $0 < \theta < 1$ and is used as a threshold for p_i and an exponential term which decays at rate α. The oscillator whose lateral potential exceeds θ is referred to as a leader. Parameter ρ is the amplitude of Gaussian noise and plays a role in assisting the separation of synchronized groups of oscillators. The behavior of y is defined in (1b) which contains a sigmoid function with β chosen small. The role of γ will be discussed below. Parameter ε is chosen to be small, i.e., $0 < \varepsilon \ll 1$ and determines that the oscillator is a relaxation oscillator with two time scales [27].

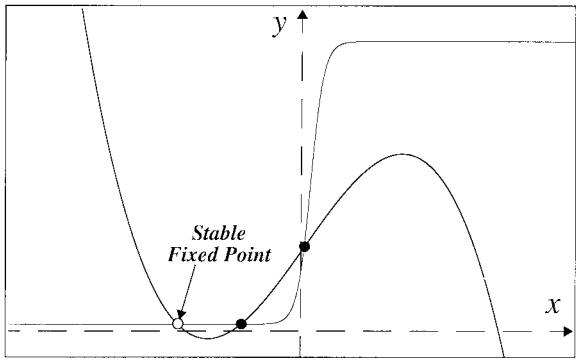
The potential term p_i plays the role of removing the oscillations of noisy regions. Its value is determined by the activities of its coupled neighbors. As shown in (2), if the activity of each neighbor $k \in N_1(i)$ is larger than threshold θ_x, T_{ik}, which is a permanent connection weight defining the topology of the network, is accumulated. If the sum is greater than threshold θ_p, then the outer Heaviside function will be one and p_i will grow its value to one by the term $\lambda(1 - p_i)$ where λ is a constant and chosen to be $O(1)$. In this case, oscillator i becomes a leader. If $H = 0$, then the potential will decay to zero with rate μ chosen to be $O(\varepsilon)$. $N_1(i)$ is called the potential neighborhood of i and $N_2(i)$ is called the recruiting neighborhood of i.

Equation (3) defines the coupling to oscillator i, which includes excitation from neighboring oscillators and inhibition from a global inhibitor z. A threshold θ_x is applied to the activity of each coupled oscillator x_k, $k \in N_2(i)$. The resultant excitation is weighted by W_{ik}, which is a dynamic weight used to achieve weight normalization to improve synchronization [29]. If the activity of the global inhibitor z is above the threshold θ_z then W_z is subtracted [see (3)]. The global inhibitor is activated when at least one oscillator in the network is active. In (4) σ_∞ is one if $x_i > \theta_z$ for at least one oscillator i and zero otherwise, and ϕ is a parameter. Fig. 2 shows a 2-D network architecture with four-neighborhood coupling. The global inhibitor, shown with a black circle, is coupled with the entire network.

The behavior of an oscillator is qualitatively shown in the phase-plane diagram (see Fig. 3). Fig. 3(a) illustrates the oscillatory behavior as a limit-cycle trajectory. When $\dot{x}_i = 0$ and $\dot{y}_i = 0$ two nullclines are defined called the x nullcline and the y nullcline, respectively. The x nullcline is a cubic where the left and middle branches connect at a point called the left knee (LK) and the middle and right branches connect at the right knee (RK). The y nullcline is a sigmoid function and, with a small β, the sigmoid is close to a step function. The two nullclines intersect at an unstable fixed point along the middle branch of the cubic. A stimulated oscillator starting in an arbitrary position will be attracted to a counter-clockwise limit cycle trajectory illustrated in Fig. 3(a). The section of the orbit that lies on the left branch is called the silent phase because x has low activity. Similarly, the section on the right branch is called the active phase. The oscillator exhibits two time scales due to ε. The slow time scale occurs during the two phases, denoted by the single arrows in Fig. 3(a). Parameter



(a)



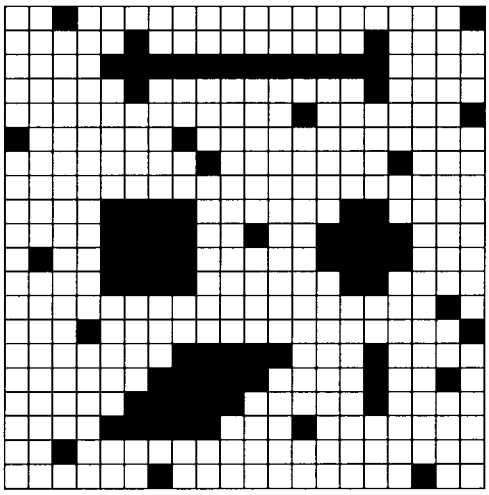
(b)

Fig. 3. Phase plane diagrams illustrating two states of a single oscillator. (a) An oscillatory state occurs when LK of the cubic is above the left part of the sigmoid. (b) A nonoscillatory state occurs when LK is below the left part of the sigmoid. Two fixed points are created on two sides of LK, where the fixed point to the left of LK is stable and acts as a gate to prevent the oscillator from oscillating.

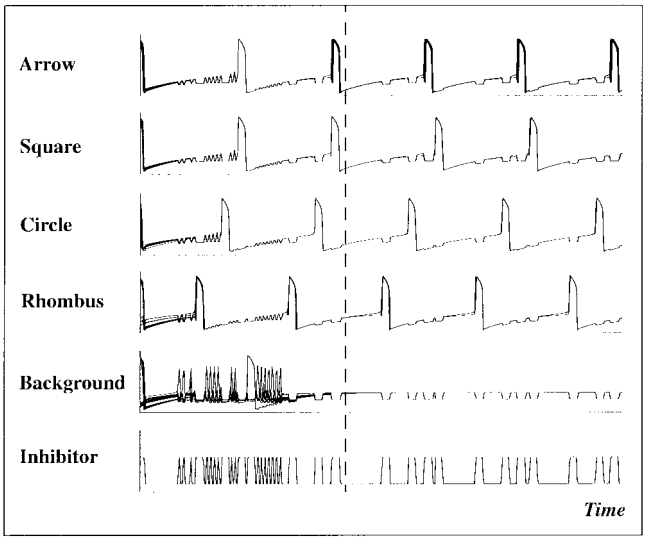
γ in (1b) controls the relative times an oscillator spends in the two phases whereby a larger γ leads to a shorter time in the active phase. The fast time scale, denoted by double arrows in Fig. 3(a), occurs when an oscillator alternates or jumps between the two phases at either LK or RK.

In (1a), I and S have the effect of vertically shifting the cubic. The position of the cubic relative to the sigmoid defines two states for an oscillator. The first is an enabled state where the cubic intersects the sigmoid at exactly one point and a limit cycle occurs [Fig. 3(a)]. The excitable state occurs when the cubic shifts downward so that LK is below the left part of the sigmoid [Fig. 3(b)]. Two fixed points are created, and one of them is stable, thus preventing the oscillator from jumping up.

The LEGION network defined in (1)–(4) has been rigorously analyzed by Terman and Wang [25], [31]. To summarize, their analytical results imply that after a number of oscillation cycles a block of oscillators corresponding to a major image region will oscillate in synchrony, while any two oscillator blocks corresponding to two different major regions will desynchronize from each other. Those stimulated oscillators whose corresponding pixels do not belong to a major region will stop oscillating shortly after the system starts, and these pixels are collectively called the background. A major region is a region that produces at least one leader. Furthermore,



(a)



(b)

Fig. 4. Computer simulation of a 20 × 20 LEGION network to segment a binary image. (a) The input image with four objects (arrow, square, cross, and rhombus) and some background noise. (b) Temporal activity of stimulated oscillators shown for 13 750 integration steps. Except for the global inhibitor, the vertical axis shows the x value and the horizontal axis indicates time. The activities of all oscillators in a group are shown together in one trace. Both synchronization and desynchronization occur after two cycles, denoted by the dashed line. The networks parameters are: $\varepsilon = 0.02$, $\alpha = 0.003$, $\beta = 0.1$, $\gamma = 20.0$, $\theta = 0.8$, $\lambda = 2.0$, $\theta_x = -0.5$, $\theta_p = 7.0$, $W_z = 2.0$, $\mu = 0.0006$, $\phi = 3.0$, $\rho = 3.0$, $T_{ik} = 2.0$, and $\theta_z = 0.1$. Dynamic weights to a stimulated oscillator are normalized to 8.0.

they established that the number of cycles required for both synchrony and desynchrony is no greater than the number of major regions plus one. This result, in particular, gives an upper bound on how long the system takes to achieve full segmentation.

To illustrate LEGION dynamics, Fig. 4 shows a computer simulation of a 20 × 20 LEGION using four-neighborhood connectivity for both N_1 and N_2 (see Fig. 2). Equations (1)–(4) were solved using a fourth-order Runge–Kutta method. The input image is a 20 × 20 binary image, shown in Fig. 4(a), with four objects (arrow, square, cross, and rhombus) and some background noise. Pixels are mapped to oscillators in one-to-

one correspondence and network connections preserve pixel adjacency. An oscillator receives a positive input stimulus if its associated pixel is black. Fig. 4(b) shows plots of the temporal activity of oscillators in each group. Each trace combines the activities for all oscillators in a group. The horizontal axis is time and the vertical axis is the normalized x activity, except for the global inhibitor which is the z activity. The activity of the oscillators representing the background is shown together. Initially, each stimulated oscillator has random activity and is enabled. Proper synchronization and desynchronization occur after only two cycles, consistent with the analysis. The near steady limit cycle behavior occurs after the vertical dotted line because all synchronized groups are separated, and oscillators representing the background have entered the excitable state.

III. Segmentation Algorithm

Using direct computer simulation of a LEGION network to segment large 2-D images and volume datasets is computationally infeasible because it requires numerically integrating a huge number of differential equations. Based on LEGION's computational characteristics, Wang and Terman [31] gave an algorithmic implementation of LEGION which follows major steps of the oscillatory dynamics. The simplifications made in their algorithm are summarized as follows.

- Jumping between the active and the silent phase takes one time step only.
- Leaders are computed during initialization.
- When every oscillator is in the silent phase the leader closest to the jumping point LK is selected to jump.
- Permanent weight T_{ik} is set to equal dynamic weight W_{ik}.
- An oscillator i in the silent phase jumps up to the active phase as soon as $S_i > 0$.
- All the oscillators in the active phase jump down if no oscillator is recruited to jump up in the previous time step.

The above simplifications were justified to be legitimate approximations of the underlying oscillatory dynamics [31]. We shall not repeat these justifications here. On the basis of these approximations, we further simplify the Wang and Terman algorithm for efficiency purposes, which are particularly needed for volume data segmentation. We also extend their algorithm in order to produce better results for the medical data that we deal with. More specifically, we made the following changes.

1) We approximate the state of an oscillator by a binary variable indicating which phase it lies in.
2) We introduce two neighborhoods, N_1 and N_2 [see (2) and (3), respectively] whereas only one neighborhood is used in [31].
3) We introduce an adaptive-tolerance scheme to adaptively weight the coupling strength between two oscillators that correspond to two pixels. In [31] a uniform W_z is used for this purpose.

It is not difficult to see that these changes do not alter the essential dynamics of the underlying LEGION network. Thus, on the basis of [31], our resulting algorithm is functionally equivalent to LEGION with appropriately chosen system parameters. Our algorithm is given below for gray-level images which we deal with in this paper.

A. Segmentation Algorithm for Gray-Level Images

At the bottom of the page we show the segmentation algorithm for gray-level images. Only the binary state of oscillator i, x_i, is used in the algorithm. I_i indicates the value of pixel i.

When the algorithm terminates in Step 2, those oscillators that still have not jumped form the background, which corresponds to scattered noisy regions that cannot produce a leader. In the algorithm, jumping down occurs immediately

---

1. Initialize
   - 1.1 Form effective connections
     $W_{ik} = 1/(1 + |I_i - I_k|), \quad k \in N_1(i) \quad \text{or} \quad N_2(i)$
   - 1.2 Identify leaders
     $p_i = H\{\sum_{k \in N_1(i)} H[W_{ik} - 1/\omega(\text{Max}(I_i, I_k))] - \theta_p\}$
   - 1.3 Place all the oscillators in the silent phase. Namely $x_i(0) = 0$. Thus $z(0) = 0$.
2. Find a leader j that has not jumped and make j jump to the active phase; terminate if every leader has jumped.
   $x_j(t+1) = 1; \quad z(t+1) = 1$ {jump up}
3. Iterate
   - If $(x_i(t) = 1$ and $z(t) > z(t-1))$
     - $x_i(t+1) = x_i(t)$ {stay on the right branch}
   - else if $(x_i(t) = 1$ and $z(t) \le z(t-1))$
     - $x_i(t) = 0; \quad z(t+1) = z(t) - 1$ {jump down}
     - If $(z(t+1) = 0)$ go to step 2
   - else
     - $S_i(t+1) = \text{Max}_{k \in N_2(i)}[W_{ik} x_k(t) - 1/\omega(\text{Max}(I_i, I_k))]$
     - If $(S_i(t+1) > 0)$
       - $x_i(t+1) = 1; \quad z(t+1) = z(t) + 1$ {jump up}
     - else
       - $x_i(t+1) = x_i(t)$ {stay on the left branch}

when all of the oscillators stimulated by the same pattern have jumped up. We note that, different from seeded region growing techniques, leaders in our algorithm may or may not produce a resultant region. In fact, a general image produces many more leaders than segments and a leader can recruit other leaders in Step 3 to form a segment. Also note that maximization is performed when computing S_i instead of summation as used in (3). The maximization operation is also used in [31] where justification is provided. Because of the maximization operation, another property of our algorithm is that segmentation results are not sensitive to the order in which pixels are evaluated and thus can be readily reproduced.

We now specify $\omega(\mathtt{Max}(\mathtt{I_i}, \mathtt{I_k}))$ in the algorithm. We call ω the tolerance function, because the larger ω is the easier of the two corresponding oscillators to synchronize. Most of the interesting structures in sampled medical-image datasets have relatively brighter pixel intensities when compared with other structures. The separation between objects is usually defined by a relatively large change in the intensities of pixels. For example, see the bony and soft tissue structures in the CT image shown in Fig. 6(a) and the cortex and the extracranial tissue in the MRI image shown in Fig. 7(a). A uniform W_z used in [31] implies a constant tolerance ω to W_{ik}. Thus, small values of ω tend to restrict region expansion in areas such as the soft tissue in Fig. 6(a) and the brain in Fig. 7(a). Large values of ω, however, may cause undesirable region merging between darker pixels and brighter ones, i.e., the flooding problem. This consideration led us to use an adaptive function for ω as given below.

We have three intuitive criteria for defining segments (groups) on an image. The first is that leaders should be generated from both homogeneous and brighter parts of the image. Second, brighter pixels should be considered similar to wider ranges of pixels than darker ones. The third criterion stipulates that the boundaries of segments are given where pixel intensities have relatively large variations. We use an adaptive-tolerance scheme that satisfies the above three criteria. In this scheme, larger tolerances are used when brighter pixels participate in grouping. A mapping is constructed between the gray-level intensities $[0, I_{\max}]$ known from the imagery type and a range of tolerance values $[\omega_{\min}, \omega_{\max}]$ given by the user. When an oscillator i attempts to recruit another neighboring oscillator k, their corresponding tolerance is found based on the brighter pixel, thus, $\mathrm{Max}(I_i, I_k)$ is used as the argument to ω and applied to W_{ik}. The function ω returns the mapped tolerance value for a given pixel. This mapping determines how much a segmented region depends upon the local pixel intensity variation. In the segmentation experiments to be reported in Section IV, we use three ω functions that are linear, square, or cubic. Specifically we define $\omega(I) = (\omega_{\max} - \omega_{\min})(I/I_{\max})^n + \omega_{\min}$ for $n = 1, 2,$ or 3. Note that ω is a monotonically increasing function. Since the range of pixel intensities is finite, the ω function may be precomputed and stored in a lookup table. This greatly reduces the segmentation time, especially when ω is complex, because the function is used heavily in the recruiting process.

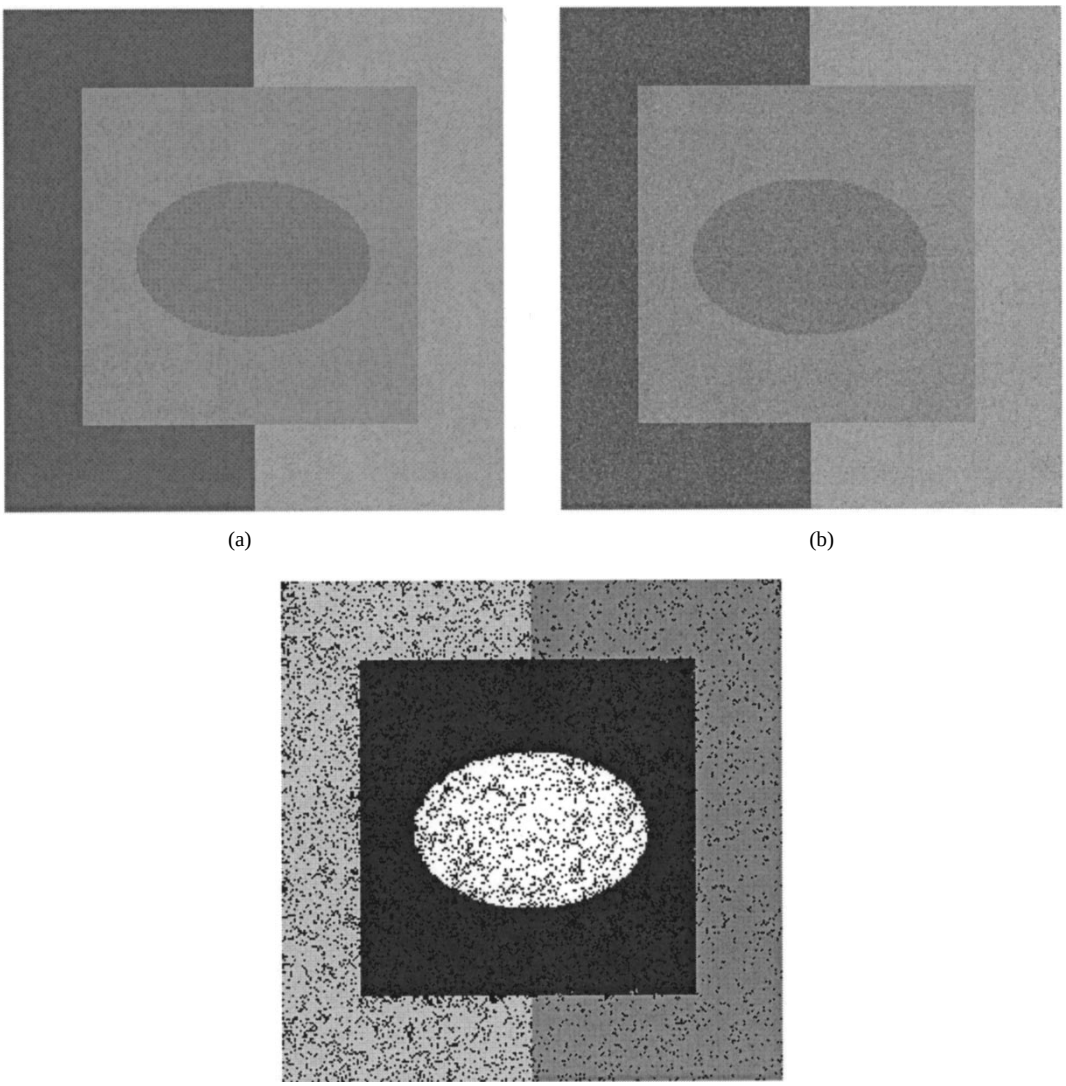
In addition to the tolerance function, our algorithm requires that the user set three parameters N_1, N_2, and θ_p. Each parameter has an influence on the number and sizes of segmented regions. It is easy to see that leaders lie in homogeneous parts of an image. N_1 and θ_p determine the identification of leaders and the minimum size for a segmented region in an image. Usually the number of leaders increases as N_1 decreases. Since more than one leader may be incorporated into the same region, a smaller potential neighborhood does not necessarily produce more segmented regions. A smaller threshold θ_p usually yields many tiny regions in the final segmentation, and a larger value will produce fewer and larger regions. This is because when θ_p is small, tiny regions in the image can produce leaders. N_2 affects the sizes and number of final segmented regions because it determines the spatial extent from a leader in the recruiting process. A smaller recruiting neighborhood implies that recruiting is restricted and usually produces smaller regions. In addition, more regions are produced because fewer leaders will be placed into the same region.

Three-dimensional segmentation is readily obtained by using 3-D neighborhood kernels. For example, in 2-D a 4-neighborhood kernel defines the neighboring oscillators to the left of, right of, above, and below the center oscillator; an eight-neighborhood kernel contains all oscillators with one oscillator away, including diagonal directions, and a 24-neighborhood kernel includes all oscillators that are two oscillators away. In 3-D, a neighborhood may be viewed as a cube. A six-neighborhood kernel contains oscillators (correspondingly voxels) that are face adjacent, a 26-neighborhood kernel contains oscillators that are face, edge, and vertex adjacent and a 124-neighborhood includes oscillators that are two voxels away.

IV. Results

We show results of our segmentation algorithm on 2-D and volume CT and MRI medical datasets of the human head. The user provides six input parameters: the potential neighborhood N_1; the recruiting neighborhood N_2; the threshold θ_p; the power n of the adaptive-tolerance-mapping function; and the tolerance range-variables $\omega_{\min}$ and $\omega_{\max}$.

Before presenting the results on medical imagery our algorithm is first used to segment a phantom image where ground truth is known. The 2-D phantom is composed of four regions, as shown in Fig. 5(a), and is then corrupted by additive Gaussian noise, as shown in Fig. 5(b). Such type of noise is characteristic of MRI acquisition [7], [20]. Figure 5(c) shows four regions plus a background, segmented by our algorithm using a square adaptive threshold function, a 24-neighborhood N_1, a four-neighborhood N_2, $\theta_p = 23$, $\omega_{\min} = 1$, and $\omega_{\max} = 4$. In this, as well as all following figures, we use a gray map to indicate the results of segmentation where each gray level indicates a distinct segment and black indicates the background generally composed of scattered areas. With only 0.05% pixels incorrectly labeled, the labeling of the pixels that belong to one of the four segmented regions is near perfect. The background accounts for 14.25% of total pixels. Notice that noisy variations in the phantom are mainly collected into the background, and thus can be easily incorporated into segments by a simple postprocessing step [see Fig. 13(b) and



(a) (b)

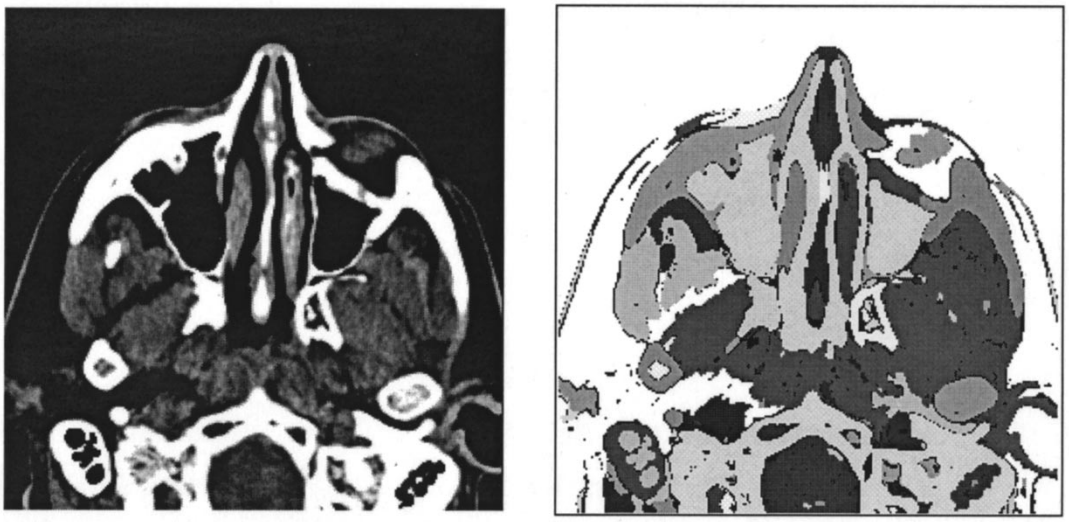
(c)

Fig. 5. Segmentation of a 256 × 256 phantom image. (a) Original image containing four regions with intensity values of 98, 118, 138, and 158. (b) The phantom with additive zero-mean Gaussian noise with variance of five for all four regions. (c) A gray map showing the result of segmenting Fig. 5(b).

(c) for example]. This way of handling noise is characteristic of our algorithm. Segmented regions tend to reflect target objects well and uncertain parts of the image are collected into the background. The background would be a primary focus in further grouping (or postprocessing) for improving segmentation results. One way to achieve such postprocessing in LEGION is to use summation instead of maximization in Step 3 of the algorithm (see [31] for further discussion). Though the variance of the noise in Fig. 5(b) is five, a modest amount indicative of MRI images, our algorithm can correctly separate the four regions in the phantom for a variance of at least seven.

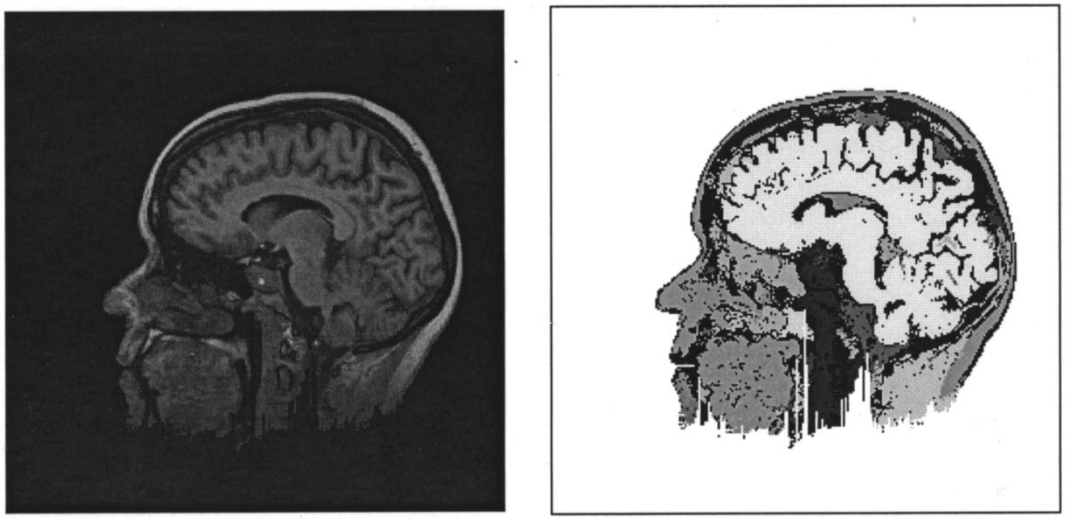
Fig. 6(a) is a CT image of a view of a horizontal section extracted at the nasal level of a human head. The white areas are bone and gray areas are soft tissue. Fig. 6(b) shows results of our algorithm using an eight-neighborhood N_1 and a 24-neighborhood N_2, $\theta_p = 7.5$, $n = 1$, $\omega_{\min} = 3$, and $\omega_{\max} = 32$. The gray map in Fig. 6(b) contains 105 segmented regions and the background. Due to limited discriminability of the printer, spatially separate regions that appear to have the same gray level are in fact different segmented regions. Whereas a global thresholding method would collect all bony structures into a single region and would have difficulty distinguishing the soft tissue, our algorithm is able to segment both. Furthermore, each spatially distinct bony structure or soft tissue area is separately labeled, as well as surrounding nasal cavities and passages.

MRI-image data have the advantage of being able to display soft-tissue structures better than CT-image data. All the MRI images used in this study were obtained using a 1.5-T MR scanner with a multielement resonant head coil. The field of view in the horizontal plane was set to 22 × 22 cm^2 and the image slices for the volume acquisition sequence were 1.7 mm thick with a pixel size of 0.86 × 0.86 mm^2 on a slice, resulting in a voxel size of 0.86 × 0.86 × 1.7 mm^3. They are T1 weighted with $\mathrm{TR/TE} = 33$ ms/40 ms, a 40-degree flip angle, and a 3-D spoiled gradient. The dataset in Fig. 12 used a Gd-dtpa contrast enhancement.



(a) (b)

Fig. 6. Segmentation of a 256 × 256 CT image. (a) Original gray-level image showing a horizontal section of a human head. (b) A gray map showing the result of segmentation.



(a) (b)

Fig. 7. Segmentation of a 256 × 256 MRI image. (a) Original gray-level image showing a midsagittal section of a human head. (b) A gray map showing the result of segmentation.

MRI imagery is more difficult to segment than CT imagery because objects are usually inhomogeneous and adjacent objects may have a low contrast in MRI. For example, see the MRI image shown in Fig. 7(a) which is a midsagittal section of the head showing many structures, including brain, vertebrae, oral cavity, extracranial tissue, bone marrow, and muscle. Fig. 7(b) shows our result of segmentation using an eight-neighborhood N_1 and a 24-neighborhood N_2, $\theta_p = 3.5$, $n = 3$, $\omega_{\min} = 1$, and $\omega_{\max} = 95$. As shown in the gray map of Fig. 7(b), the MRI image is segmented into 70 regions plus a background. The entire brain is segmented as a single region. Other significant segments include the extracranial tissue, the chin and the neck parts, the vertebrae, etc.

Among the six parameters, the effects of N_1, N_2, and θ_p are discussed in the previous section. The roles of the adaptive-tolerance measure and its parameters are illustrated in Figs. 8 and 9. Fig. 8(a) displays an MRI image of a horizontal section of a human head at the eye level, showing the structures of brain, two eyeballs, the ventricle at the center, extracranial tissue, and scattered bone marrow. Fig. 8(b) shows the segmented brain by a human expert, and this will be used later for a quantitative measure on the performance of our algorithm. To apply our adaptive-tolerance method, pixel intensities are first mapped to tolerance values. As the power n increases relatively larger tolerances are assigned to brighter pixels, which means that brighter pixels have relatively larger ranges of grouping. This is desirable if brighter pixels represent more interesting structures, as is usually the case for sampled medical images. Fig. 8(c)–(e) shows segmentation results using the cubic, square, and linear tolerance functions,

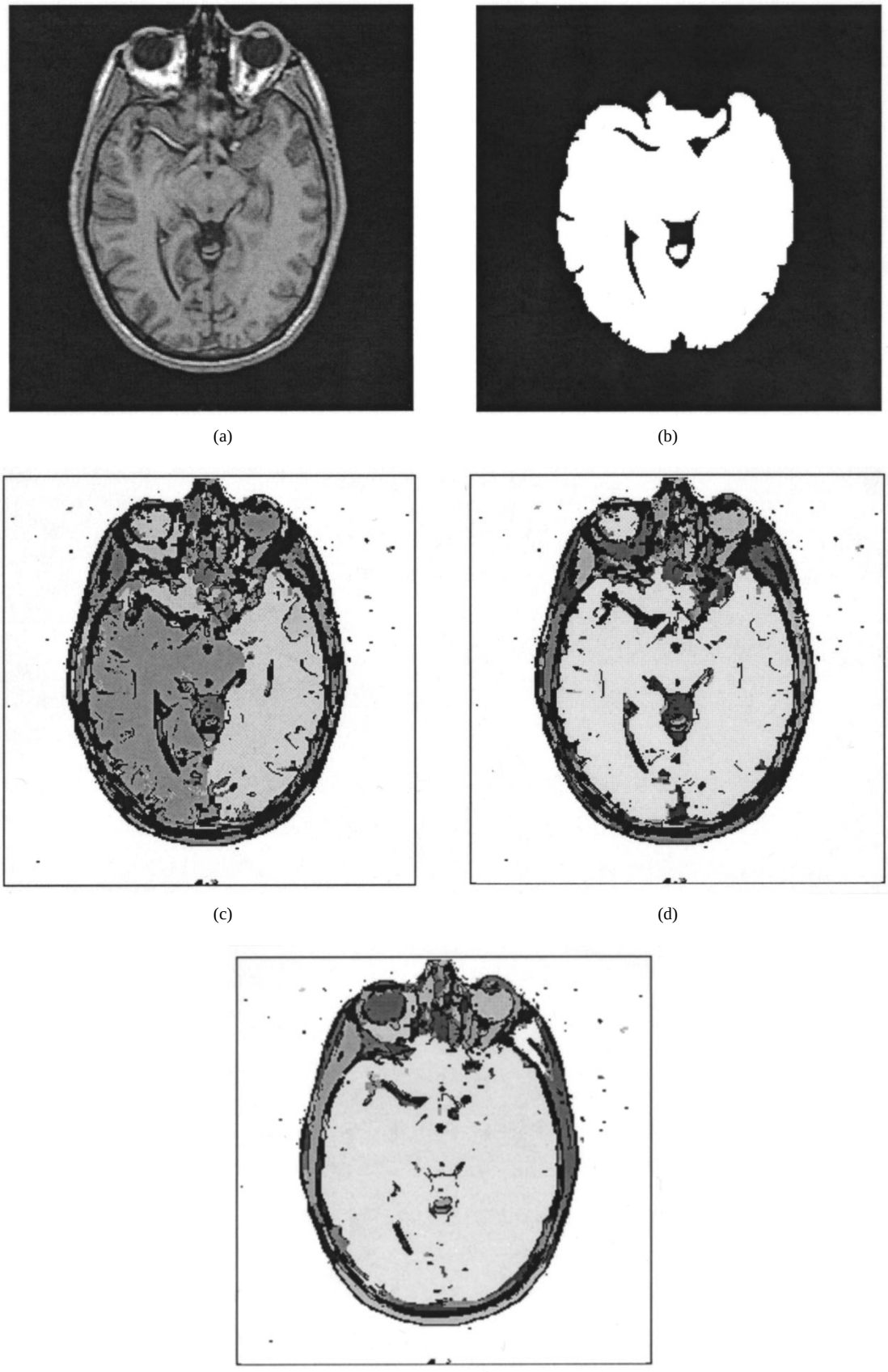
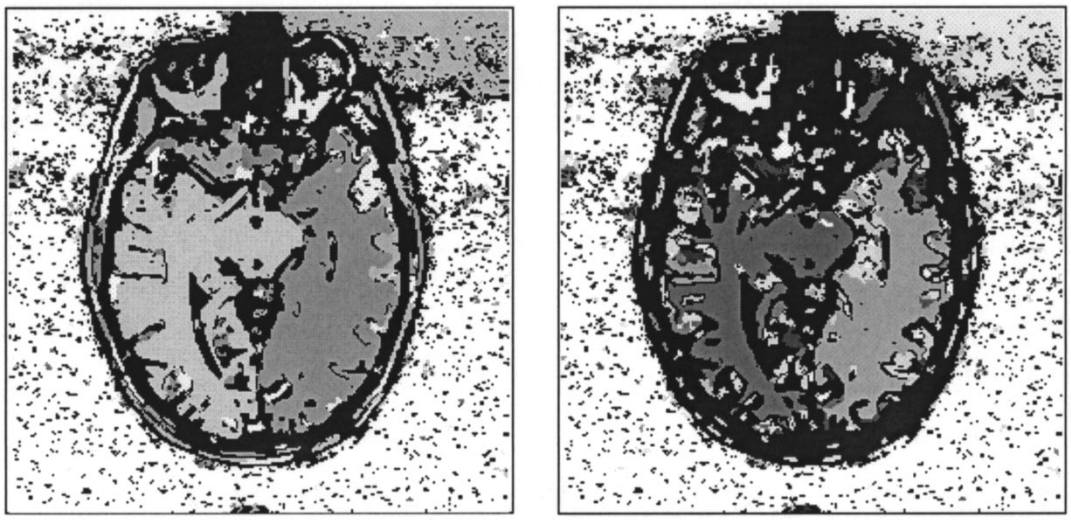


Fig. 8. Segmentation of a 256 × 256 MRI image. (a) Original gray-level image showing a horizontal section of a human head. (b) Manually segmented brain from (a). (c) Segmentation results when the tolerance function is cubic. (d) Segmentation results when the tolerance function is square. (e) Segmentation results when the tolerance function is linear.

(a)

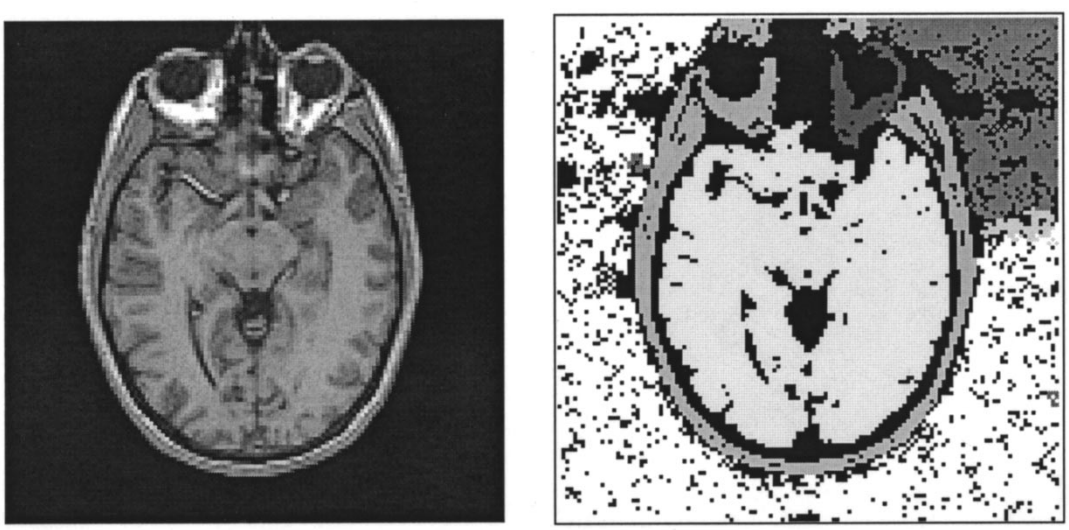


(b) (c)

Fig. 9. Effects of tolerance parameters in segmenting the MRI image of Fig. 8(a). A cubic tolerance function is used with $\omega_{\min}$ clamped to 1. (a) Segmentation results when $\omega_{\max}$ is 80. (b) Segmentation results when $\omega_{\max}$ is 40. (c) Segmentation results when $\omega_{\max}$ is 20.

respectively, on the MRI image of Fig. 8(a). In this case N_1 and N_2 are both set to eight-neighborhood $\theta_p = 4.5, \omega_{\min} = 4$, and $\omega_{\max} = 18$. The gray maps in Figs. 8(b)–(d) contain 288, 239, and 173 regions, respectively. The three gray maps show that areas with bright pixels are consistently segmented, such as the white structures behind the eyes, the brain, and the brighter parts of the extracranial tissue. Also consistently segmented are the two eyeballs. In the leader identification step of our algorithm, the tolerance function has a similar effect in determining leaders as do N_1 and θ_p. In the recruiting step, the tolerance function affects region expansion because it specifies which oscillators in N_2 are to be grouped. Region expansion depends on how fast pixel intensities change within a region. The boundaries between regions occur where intensity changes are too large. When the pixel intensities within a region change substantially, a tolerance function with a higher n usually tends to break the region apart. For example, the brain is segmented into many parts in Fig. 8(c), whereas it remains a single region in Fig. 8(d) and (e). In Fig. 8(e), however, boundary details of the brain are lost. Segmentation of the extracranial tissue shows the same effect. Thus, an appropriate choice of the tolerance function depends upon the intensity-variation characteristics of target objects.

The user can also control the grouping for darker and brighter pixels by selecting $\omega_{\min}$ and $\omega_{\max}$. When either parameter is changed, the tolerance function remaps all intensities to a new range of tolerance values. When $\omega_{\min}$ is increased, tolerance values will increase more quickly for darker pixels. A similar effect holds for brighter pixels when $\omega_{\max}$ is increased. Moreover, the effect is more dramatic. In Fig. 9, we show segmentation results using a cubic tolerance function clamping $\omega_{\min}$ to one and varying $\omega_{\max}$ from 80 in Fig. 9(a), to 40 in Fig. 9(b), and 20 in Fig. 9(c). The remaining parameters are the same as in Fig. 8. The results in Fig. 9(a)–(c) contain 278, 375, and 459 regions, respectively. The white areas behind the eyes, and the bright areas that cor-



(a) (b)

Fig. 10. Segmentation of a reduced resolution image of Fig. 8(a). (a) Original gray-level image after a reduction in resolution by half. (b) The result of segmentation.

respond to the brain and the extracranial tissue (see Fig. 8(a)) are grouped more fully in Fig. 9(a) than in Fig. 9(b) and (c).

Figs. 9 and 10 also illustrate how our algorithm handles the various types of noise artifacts commonly found in sampled image data. Fig. 8(a) contains noise inherent in the imaging process, as well as arbitrarily sized noisy areas caused by too small a sampling rate compared with the sizes of the structures being imaged, such as the nasal area between the two eyes. Fig. 9 shows that these noise artifacts are collected into the background and do not affect segmentation of other regions. To further illustrate the robustness of our algorithm to noise artifacts, we reduce the resolution of the image of Fig. 8(a) by half by throwing away every other pixel value. As shown in Fig. 10(a), this magnifies the noise artifacts, especially near the boundaries of objects. The result of segmenting Fig. 10(a) is shown in Fig. 10(b) using a 24-neighborhood N_1, an eight-neighborhood N_2, $\theta_p = 11.5$, $n = 3$, $\omega_{\min} = 1$ and $\omega_{\max} = 150$. The segmentation result contains ten regions. The algorithm is able to segment the brain, the areas behind the eyes, and the extracranial tissue while placing a large number of noisy areas into the background.

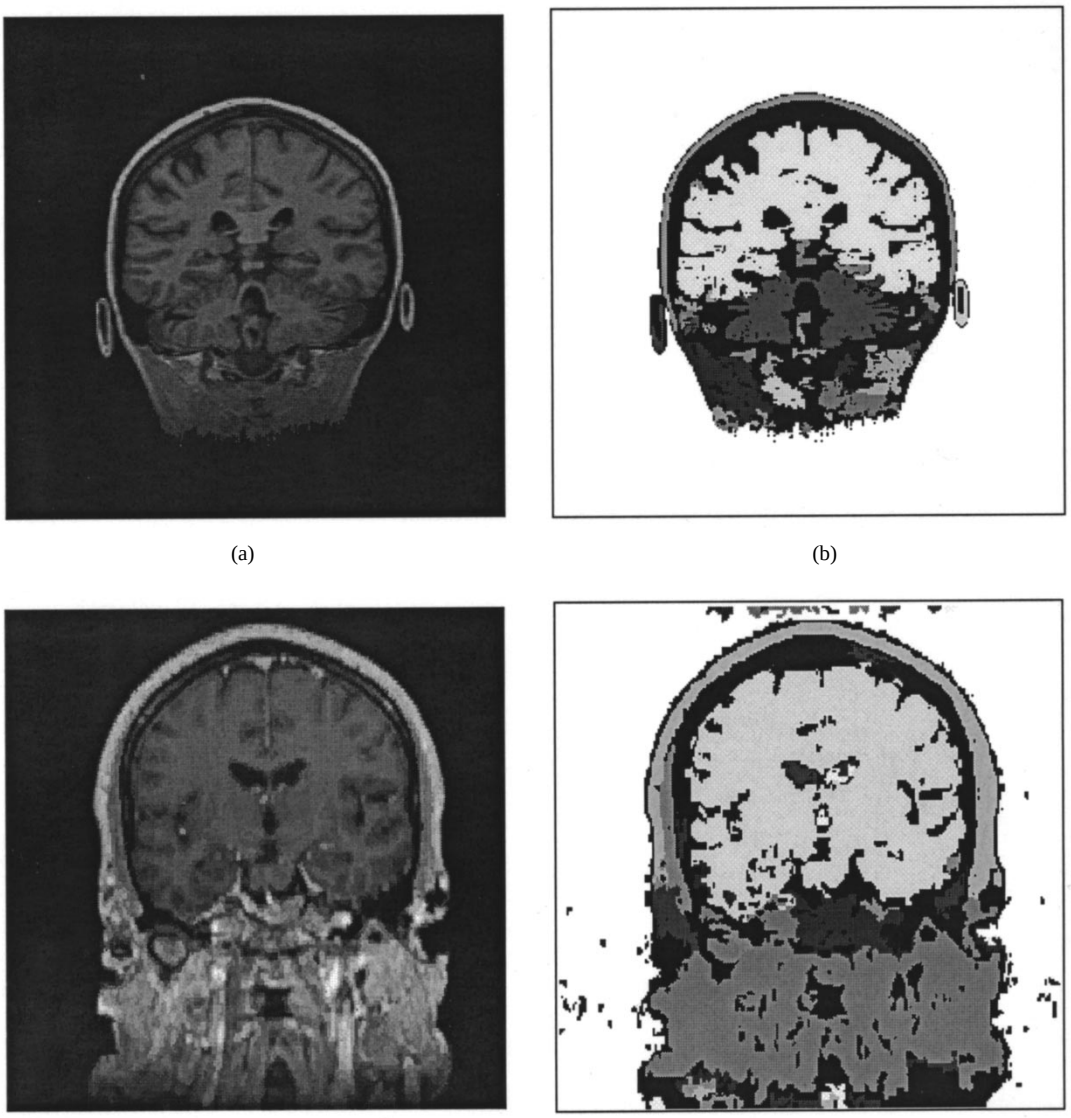
It is well known that, due to the lack of ground truth, quantitative evaluation of a segmentation algorithm is difficult to achieve. An alternative is to use manual-segmentation results as ground truth. One should, however, bear in mind that such manual segmentation is not perfect as ground truth (see more discussions in Section V-B). Nevertheless, some quantitative comparison with manual segmentation may provide a useful indication. We use Fig. 8(b), a segmented brain by a human expert, as the ground truth for a quantitative comparison. Since the brain is the target of manual segmentation, only the segmented brain region by our algorithm is used in the comparison. In addition, we do not use Fig. 8(c), Fig. 9(b) or Fig. 9(c) because the brain is fragmented in segmentation for the purpose of explaining parameters. Table I gives error rates in two metrics: false target counts, those pixels that are wrongly segmented as the target by the algorithm, and false nontarget counts, those pixels that are in but fail to be segmented as the target. The percentages in the table are calculated based on the number of pixels in the manually segmented target. In the overall best case [Fig. 8(e)] the error rate is less than 4% in both metrics. For other segmentation results the false nontarget rate is higher, but the false target rate is very low: less than 1%.

TABLE I
ERROR RATE FOR SEGMENTING FIG. 8(a)

Segmented image	False target	False nontarget
Fig. 8b	0.31%	10.65%
Fig. 8c	2.83%	3.95%
Fig. 9a	0.41%	10.93%
Fig. 10b	0.67%	9.1%

TABLE II
EXECUTION TIME

Segmented image	Image size	Time (in sec)
Fig. 6b	256x256	1.62
Fig. 7b	256x256	1.45
Fig. 8c	256x256	0.59
Fig. 8d	256x256	0.59
Fig. 8e	256x256	0.57
Fig. 9a	256x256	0.60
Fig. 9b	256x256	0.62
Fig. 9c	256x256	0.62
Fig. 10b	128x128	0.24
Fig. 11b	256x256	0.81
Fig. 11d	256x256	0.83
Fig. 11f	256x256	0.97
Fig. 11h	256x256	1.01
Fig. 12	256x256x128	58.62



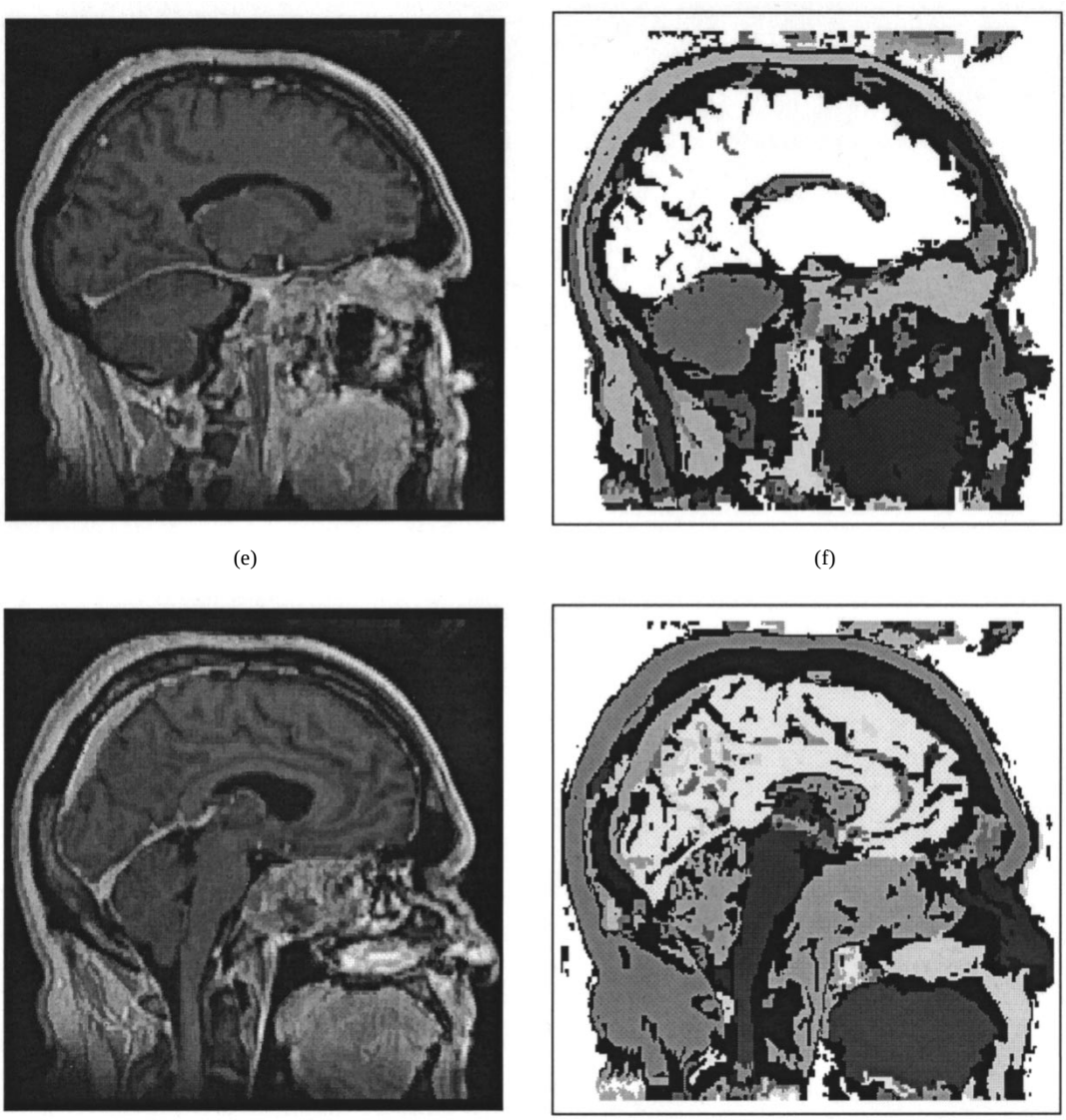
(a) (b)

(c) (d)

Fig. 11. Segmentation of 256 × 256 MRI images. For all segmentation results a cubic tolerance function is used. (a) Original gray-level image showing a coronal section of a human head. (b) Segmentation result for the image in (a) with a 24-neighborhood N_1, a four-neighborhood N_2, $\theta_p = 11.5$, $\omega_{\min} = 1$, and $\omega_{\max} = 316$. (c) Original gray-level image showing another coronal section of a human head. (d) Segmentation result for the image in (c) with an eight-neighborhood N_1, an eight-neighborhood N_2, $\theta_p = 7.5$, $\omega_{\min} = 7$ and $\omega_{\max} = 90$.

Fig. 11 shows segmentation results on a number of 256 × 256 MRI images with different sections. Parameters have been chosen in order to extract various meaningful structures. In Fig. 11(b), significant regions that are segmented include the cortex, the cerebellum, and the extracranial tissue, as well as two ear segments. In Fig. 11(d), the entire brain is segmented as a region, as are the extracranial tissue and the neck muscle. In Fig. 11(f), the cortex and the cerebellum are well separated. Other interesting regions that are segmented include the chin part and the extracranial tissue. In Fig. 11(h), again the cortex and the cerebellum are well segmented. In addition, the brainstem and the ventricle lying at the center of the brain are correctly separated. Other structures are also well segmented, as in Fig. 11(f).

As discussed before, our algorithm easily extends to segment volume data by expanding 2-D neighborhoods to 3-D. To illustrate 3-D segmentation we show the result of segmenting an entire MRI volume dataset from which the image in Fig. 8(a) was obtained. The volume dataset consists of 128 horizontal sections, and each section consists of 256 × 256 pixels, with a total of 256 × 256× 128 pixels. The dataset was partitioned into four stacks along the vertical direction. From superior to inferior: stack one consists of sections 1–49; stack two, sections 50–69; stack three, sections 70–89; and stack four, sections 90–128. We divide the entire dataset to four stacks for the purpose of dividing total computing to different stages, and for reflecting major anatomical shifts. It turns out that such divisions are approximately consistent with signal intensity variations introduced during the volume dataset acquisition process. The following parameters are common for all stacks: $N_2 = 26$, θ_p is half the size of N_1, a square tolerance function, and $\omega_{\min} = 1$. In addition, $N_1 = 26$ for stack one and six for the remaining stacks; $\omega_{\max} = 10$ for stack one, 25 for stack two, 30 for stack



(e) (f)

(g) (h)

Fig. 11. (*Continued.*) Segmentation of 256 × 256 MRI images. For all segmentation results a cubic tolerance function is used. (e) Original gray-level image showing a sagittal section of a human head. (f) Segmentation result for the image in (e) with a 24-neighborhood N_1, an eight-neighborhood N_2, $\theta_p = 17.5$, $\omega_{\min} = 7$ and $\omega_{\max} = 75$. (g) Original gray-level image showing another sagittal section of a human head. (h) Segmentation result for the image in (g) with a 24-neighborhood N_1, an eight-neighborhood N_2, $\theta_p = 19.5$, $\omega_{\min} = 4$ and $\omega_{\max} = 250$.

three, and 35 for stack four. We emphasize that both N_1 and N_2 are 3-D neighborhood kernels and segmentation is performed on 3-D datasets directly rather than on individual 2-D sections. The parameters in the algorithm are chosen to extract the 3-D brain and no attempt is made to correlate parameter values with signal variations between horizontal sections which are introduced during image acquisition. In this volume, dataset signal intensities vary systematically and by 5%–10% in the entire dataset. Through local connections, our algorithm can tolerate to a certain extent gradual variations without changing parameter values. Fig. 12(a) and (c) shows two views of the segmented 3-D brain using a volume-rendering software developed in [19]. Fig. 12(a) displays a top view with the front of the brain facing downward. Fig. 12(c) displays a side view of the segmented 3-D brain, with the front of the brain facing leftward. To put our results in perspective, Fig. 12(b) and (d) shows the corresponding views of manual segmentation of the same volume dataset by a human technician (more discussions in Section V). As shown in Fig. 12, the results of manual segmentation fit well with prototypes of our anatomical knowledge. On the other hand, as will be discussed in Section V, the results of our algorithm can better reflect details of a specific dataset.

All of the results reported above were obtained on an SGI Onyx workstation. Table II documents the execution times for all segmentation runs where a 256 × 256 brain section takes about one second. We note that our algorithm is about ten times as fast as those in Section V-A that have reported running times (after speed differences due to computer platforms are calibrated) where a 256 × 256 section typically takes approximately one minute on a Sun Sparc 10 workstation.

We have performed many other tasks of medical-image segmentation using our algorithm, including tasks such as segmenting blood vessels in other 3-D volume MRI datasets and

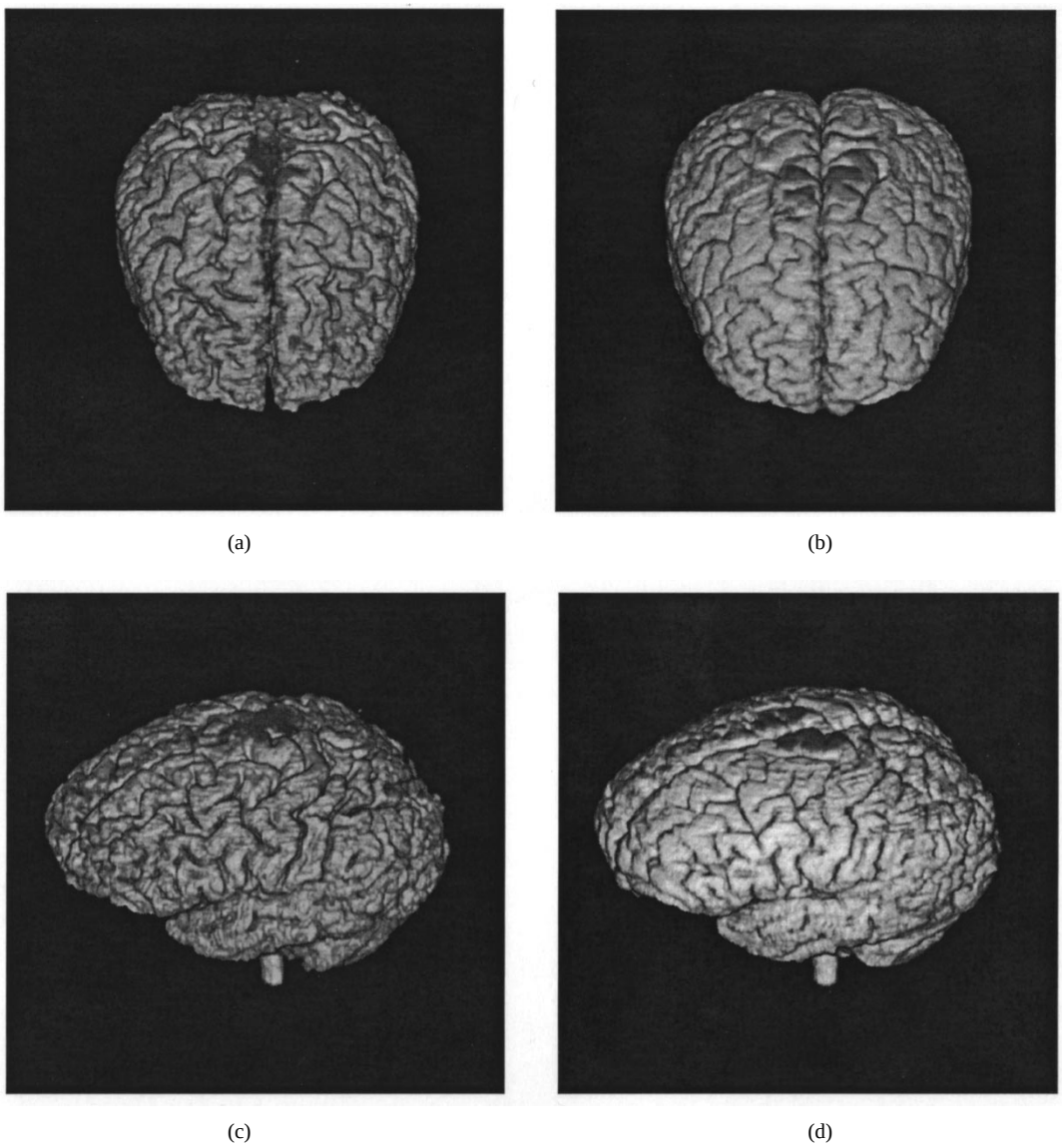


Fig. 12. Segmentation of a 3-D MRI volume dataset for extracting the brain. The segmentation results are displayed using volume rendering. (a) and (c) show the results with a top view (the front facing downward) and a side view (the front facing leftward), respectively. (b) and (d) show the corresponding results produced by slice-by-slice manual segmentation.

segmenting sinus cavities from the Visible Human Dataset [1]. The segmentation results are comparable with those illustrated above. We note that our segmentation results are usually robust to considerable parameter variations, that is, major segments are not sensitive to these variations.

Determining appropriate algorithm parameters is not as difficult as it may appear because each parameter has an intuitive meaning and its effect on segmentation is fairly predictable. The method we use to set the parameters is an incremental one where each parameter is set individually while holding the others constant. First, target structures in the dataset are determined for segmentation. Usually these structures correspond to bright and relatively homogeneous regions within images. To reduce the number of extraneous regions in segmentation, N_1 and θ_p should both be set large initially. As an initial step, we suggest for a 2-D image a 24-neighborhood N_1 and $\theta_p = 16$ (two-thirds of the size of N_1). They essentially act as filters for removing small and nontarget regions. A cubic tolerance function should be used first, in order to identify bright regions. To limit expansion into extraneous regions, N_2 can be chosen small initially (e.g., eight-neighborhood). Choosing $\omega_{\min}$ and $\omega_{\max}$ is relatively tedious and may require some trial and error to produce best results.

The code for our algorithm is written in C and is compiled to run on both the SGI and HP platforms. The user is able to set each of the six algorithm parameters through a graphical user interface (GUI) written in Motif. The software displays a 3-D volume with integer tags so that the user can select one particular segmented 3-D region for viewing purposes. The latter utility is also written in C and Motif for the GUI.

V. Comparisons

Medical-image segmentation, particularly for MRI images, is a well-studied problem and there is a large body of literature on the topic. In Section V-A, we compare our method with other segmentation methods for medical imagery. Such a

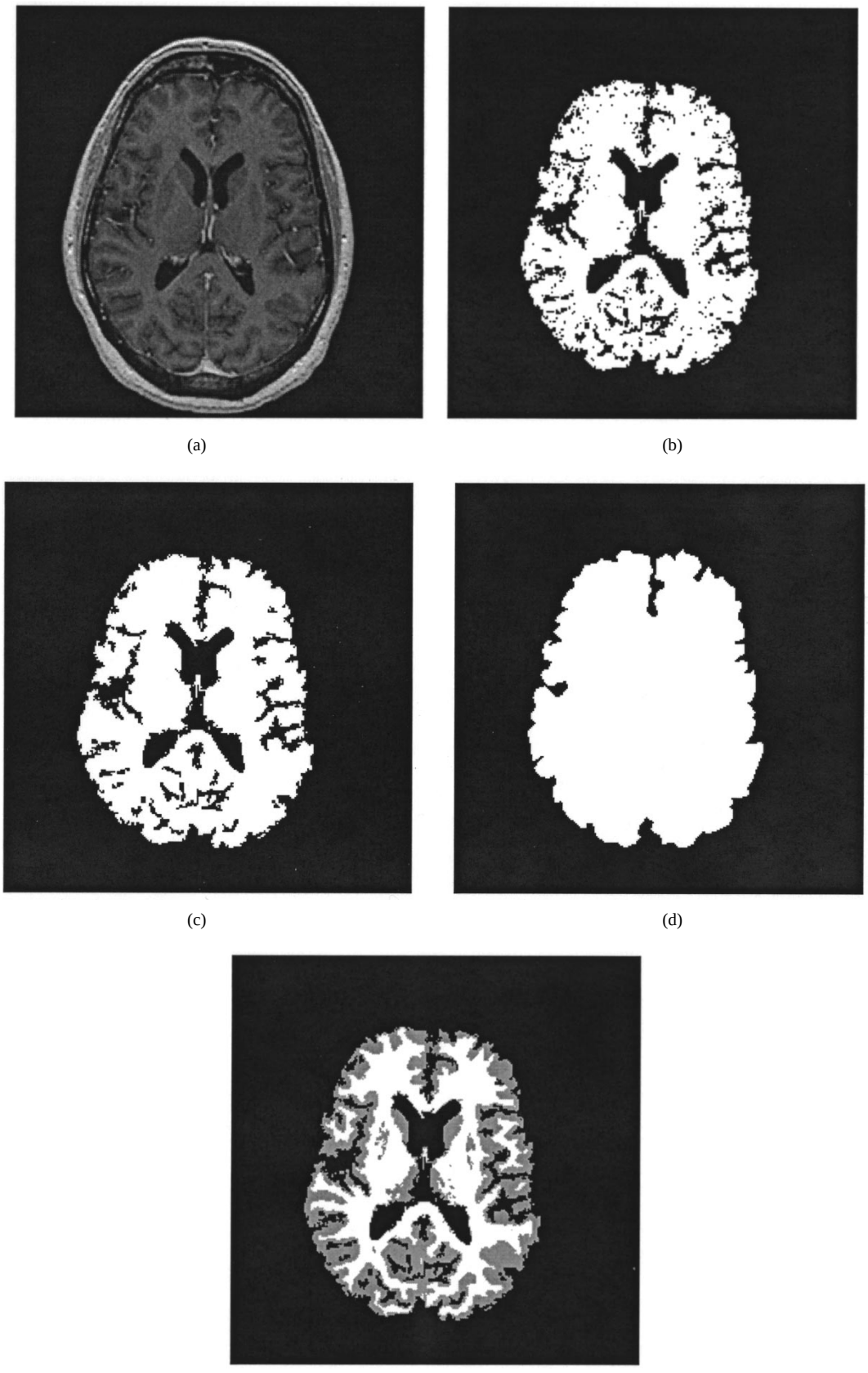


Fig. 13. Comparison between computer and manual segmentation. (a) Original gray-level image showing a horizontal section of the volume data set used in Fig. 12. (b) Segmentation result of the image in (a) from 3-D segmentation in Fig. 12 by our algorithm. (c) Result of (b) after tiny holes merge with the brain segment. (d) Segmentation result of the image in (a) from 3-D manual segmentation in Fig. 12. (e) Result of separating white matter from gray matter based on the segmented brain in (c).

comparison is by no means complete, as a comprehensive comparison can only be done in a survey paper given the size of the literature. Instead, we have selected several methods that have been recently and successfully applied to medical imagery. In Section V-B, we conduct another, perhaps more meaningful, type of comparison, with manual segmentation.

A. Comparison with Other Methods

In this subsection, we compare our algorithm to four recent approaches that have been successfully used to segment MRI images of the brain. The first is the commonly used global-thresholding approach. In [15], global thresholding and space filling were used to segment the white and gray matter of the brain from sagittally oriented MRI images. In this method, a user manually identifies regions of interest separately for the white- and gray-matter regions in order to define two pixel-intensity ranges for applying global thresholding. As was previously mentioned, the loss of spatial information is an inherent problem in this approach. Joliot and Mazoyer [15] tried to overcome this problem with a connectivity algorithm to fill the white- and gray-matter segmentation, again with user intervention, to seed a primary region of interest. White-matter segmentation is done first, and then gray-matter segmentation starts on the surface of the white matter based on anatomical knowledge that gray matter surrounds white matter. Many other *ad hoc* techniques are also employed to overcome the problems caused by high intensity variation and the lack of topology. Compared with this global thresholding method, our method embodies topology naturally in oscillator-network connectivity through local connections that can tolerate gradual variations in pixel intensities in one region, even though the intensity range of the whole region may be large. Also, our method is more general, not limited to segmenting prespecified objects, and requires less user intervention. On the other hand, because of generality, domain-specific knowledge is not utilized as directly in our method as in [15].

Statistical models, such as Markov or Gibbs random field models, have been widely used to segment medical imagery [8], [20]. In this approach, connectivity and smoothness constraints on desired segmentation are imposed by Gibbsian priors. Using Bayesian formulation, the segmentation algorithm attempts to estimate the maximum *a posteriori* probability (MAP). Given good parameter estimation and prior knowledge of the number of target objects to be segmented, statistical models can produce good segmentation results. In both [8] and [20] gray matter, white matter, and cerebral spinal fluid were the target segments. Compared to this approach, our method is more flexible in terms of what is segmented and does not require prior specification of the number of segments. In addition, our algorithm is less intensive computationally, as indicated by reported running times [8], [20]. On the other hand, if the target segments are known beforehand, this knowledge can be incorporated into the statistical approach for producing better results.

Recent image-segmentation methods have used the idea of fitting a contour to the boundary of a targeted object, called active contours or snakes [17], based on an energy-minimizing spline that converges to the boundary of a target region. In contrast to edge-detection methods that first identify edges and then try to construct a closed boundary from the edges, this approach shapes a predefined contour to match the boundary of the object. An energy function is generally defined in terms of an internal energy that measures the smoothness of the boundary and an external energy that measures image properties as well as user defined constraints. The contour is iteratively updated so that the energy is minimized. Chiou and Hwang [9] noted the local minimum problem with the active-contour method for segmenting MRI images of the brain, which prevents the spline from converging to its desired boundary. They used a two-layer perceptron to train on the image data to better identify pixels on the boundary of the brain. In addition, pushing forces were added to further help the spline to avoid local minima, so as to reduce the requirement that the initial contour must be placed close to the target boundary. To work for noisy images, the model needs to be further augmented with a stochastic decision mechanism.

Active-contour methods require considerable user intervention in placing the initial contour and for [9] additional user input of desired boundary pixels for supervised training of multilayer perceptrons. In order to provide such intervention, the user has to solve the segmentation problem to a considerable extent. Note that, besides this kind of user input, the system still needs to properly set a number of parameters. Also, energy minimization is generally computationally expensive, and the local-minimum problem is always a major concern. Our method does not suffer from these problems. Besides parameter setting our algorithm is entirely automatic, requiring no user intervention. Our method represents a uniform framework, whereas the method of Chiou and Hwang [9] is a mixture of several unrelated techniques. In terms of segmentation results, our results appear to be at least as good in identifying contours of the brain.

Recently, a neural network method was proposed by Alirezaie *et al.* [3] who used learning-vector quantization (LVQ) for segmenting 2-D MRI images of the brain. This method treats the segmentation problem as classifying pixels based upon features that are extracted from multispectral images and incorporate spatial information. The network is a typical self-organizing map [18] which requires prior knowledge of the number of objects to be segmented. For proper initialization a training set is needed for each image and it is selected by manual segmentation of multispectral MRI images. It is particularly instructive to compare the method of Alirezaie *et al.* [3] with our method, since both are neural network models. As described above, this LVQ method requires considerable user intervention to provide essential information for segmentation, whereas ours does not. Although spatial information is utilized in the LVQ method, it can be incorporated only in a limited way specified by system parameters. In our method, spatial information is naturally incorporated into lateral connections between oscillators and, through temporal dynamics, oscillators can influence each other to an unbounded extent even though oscillators have only local direct connections. Perhaps more importantly, segmentation in our method is the result of emergent behavior of the entire oscillator network that takes

the whole image as the input. As a result, the LEGION network, besides its biological plausibility, is especially feasible for parallel-hardware implementation, which would be important for real-time segmentation of volume datasets. In contrast, classification-based methods, including that of Alirezaie *et al*. [3], operate on a single pixel at a time.

B. Comparison with Manual Segmentation

Since much of the motivation for medical-image segmentation is to automate all or part of manual segmentation, it is perhaps more important to compare the results of our algorithm with those of manual segmentation. Manual segmentation generally gives the best and most reliable results when identifying structures for a particular clinical task. Usually, a medical practitioner with knowledge of anatomy utilizes a mouse-based software to outline or fill regions of target structures on each image slice in an image stack, i.e., a volume. The segmenter needs to mentally reconstruct a structure in 3-D because only 2-D image slices can be viewed. Segmentation of each slice requires that the segmenter view adjacent images in the stack in order to correctly reconstruct an object in 3-D. The task is very tedious and time consuming for the segmenter, and thus does not serve the needs of daily clinical use well. On the other hand, no fully automatic method exists that is able to provide comparable segmentation quality to manual segmentation. Our algorithm lends itself to a semiautomatic approach that is easy to use and fast to generate results, thus providing the segmenter a valuable tool to attain acceptable segmentation quality.

Fig. 12 compares the performance of our approach in segmenting MRI volume datasets with manual segmentation performed by a medical technician. Using a segmentation software available from the National Institutes of Health on the Apple Macintosh platform, the task required the segmenter many hours. To have a closer comparison, we display in Fig. 13 one horizontal section and its segmentation in Fig. 12. The sample section from the volume dataset is shown in Fig. 13(a) and Fig. 13(d) shows the manually segmented brain for this section. Fig. 13(b) shows the segmented brain from the same sample image using our method. Fig. 13(c) shows the result after a simple postprocessing step which fills in tiny isolated holes in the background that have no more than 12 pixels. A comparison between Fig. 13(c) and (d) reveals that our algorithm is able to provide more details that are omitted in the corresponding manual segmentation. For example, the brain ventricles and many fissures are correctly outlined in Fig. 13(c) by our algorithm, but are too tedious to outline in manual segmentation.

One objective of medical-image segmentation is to separate white matter and gray matter. To achieve this objective, knowledge about the characteristics of white and gray matter is generally needed. Unlike algorithms that are specifically designed for this purpose (see for example [20] and [8]), our algorithm is intended to be more flexible for segmenting a variety of structures (see Fig. 11). However, it is interesting to note that once the brain is segmented, the separation of gray matter from white matter can be performed by our algorithm in another pass (or as the second layer, see discussions in Section VI). To illustrate this, we use Fig. 13(c) as the segmented brain, and Fig. 13(e) displays the result of segmenting white matter and gray matter. Given that the segmented brain consists of either white matter or gray matter, our second pass treats all segmented regions as white matter (likely discontiguous) by an appropriate selection of leaders and the background as gray matter. Leaders are simply chosen by checking if the average pixel intensity in an entirely stimulated potential neighborhood exceeds a threshold [75 in Fig. 13(e)]. Other parameter values for producing Fig. 13(e) are $N_1 = N_2 = 8$, $n = 1$, $\omega_{\min} = -16$, and $\omega_{\max} = 40$. Note that a negative tolerance value prohibits any grouping. The result in Fig. 13(e) is comparable with those of Chang *et al*. [8] and Rajapakse *et al*. [Fig. 1, 20] whose statistical algorithms, as discussed in Section V-A, are specifically designed to segment gray matter, white matter, and cerebral spinal fluid.

VI. Concluding Remarks

We propose to use a new neurally inspired approach to the problem of segmenting sampled medical-image datasets. Studies from neurobiology and the oscillatory-correlation theory suggest that objects in a visual scene may be represented by the temporal binding of activated neurons via their firing patterns. In oscillatory correlation, neural oscillators respond to object features with oscillatory behavior and group together by synchronization of their phases. Oscillator groups representing different objects desynchronize from each other. Our algorithm is derived from LEGION dynamics for image segmentation because of its desirable properties of stimulus-dependent oscillations, rapid synchronization for forming oscillator groups, and rapid desynchronization via a global inhibitor for separating oscillator groups.

Based on performance considerations, we derive an efficient algorithm that closely follows LEGION dynamics. To segment CT- and MRI-sampled datasets, we group oscillators based on the intensity contrasts of their corresponding pixels and introduce an adaptive-tolerance scheme to better identify structures of interest. Our results show that the LEGION approach is able to segment volume datasets and, with appropriate parameter settings, produces results that are comparable to commonly used manual segmentation. Our method also compares favorably with other segmentation methods for medical imagery. Our software is currently used to segment structures from sampled medical datasets from various sources including the Visible Human Project dataset.

Various extensions to our algorithm can be explored for further improving segmentation results and reducing user intervention in parameter setting. The neighborhood kernels used for both the potential and recruiting neighborhoods may be set in many ways. For example, a Gaussian kernel may be used as a recruiting neighborhood, with the connection strength between two oscillators falling off exponentially. Such a Gaussian neighborhood can potentially alleviate unwanted region merging, or the flooding problem (see [14]). Other tolerance functions may also be used to better define pixel similarity. For example, a tolerance mapping using a Gaussian function may provide proper tolerance values for both darker and brighter pixels.

So far, segmentation is performed by a single network layer only. Another layer may be added to process the result of the first layer. Because the second layer deals with segmented regions, it is easy to make the second layer extract only major, i.e., large, regions while putting other regions into a background. This can be implemented by choosing a larger potential neighborhood for the second layer while using the same recruiting neighborhood. With the addition of this second layer, we expect that a majority of small segments in the segmentation results of Section IV will be removed, and the number of final segments will be cut dramatically.

While our algorithm captures major components of LEGION dynamics and leads to much faster implementation, it should be pointed out that our algorithm is not equivalent to LEGION dynamics. For example, LEGION dynamics exhibits a segmentation capacity, only a limited number of segments can be separated, whereas our algorithm can produce an arbitrary number of segments. The ability to naturally exhibit a segmentation capacity may be a very useful property for explaining psychophysical data concerning perceptual organization. In addition, our algorithm is iterative in nature. The dynamical system of LEGION, on the other hand, is fully parallel, and does not require synchronous algorithmic operations.

To conclude, our results as well as our comparisons with other methods suggest that LEGION is an effective computational framework to tackle the problem of medical-image segmentation. Layers of LEGION networks may be envisioned that are capable of grouping and segregation based on partial results from preceding layers, and thus may further enhance segmentation performance. The network architecture is amenable to VLSI chip implementation, which would make LEGION a plausible architecture for real-time segmentation. Real-time segmentation would be highly desirable for large volume datasets.

ACKNOWLEDGMENT

The authors would like to thank S. Campbell for his invaluable help in preparing this manuscript, and the anonymous reviewers for their comments. The authors also wish to thank W. Mayr for his time and effort in providing manual segmentation, K. Mueller for providing the volume-rendering software and images of 3-D brain visualizations and D. Stredney for his manual segmentation in Fig. 8(b), as well as P. Schmalbrock, D. Stredney, and G. Wiet for very useful interactions. Volume datasets were provided by the James Cancer Research Institute of the Ohio State University, the Alcohol and Drug Abuse Research Center at the McLean Hospital of the Harvard Medical School, and the National Library of Medicine.

REFERENCES

[1] M. J. Ackerman, "The visible human project," *J. Biocomm.*, vol. 18, p. 14, 1991.
[2] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 641–647, 1994.
[3] J. Alirezaie, M. E. Jernigan, and C. Nahmias, "Neural network-based segmentation of magnetic resonance images of the brain," *IEEE Trans. Nucl. Sci.*, vol. 44, pp. 194–198, 1997.
[4] T. Asano and N. Yokoya, "Image segmentation schema for low-level computer vision," *Pattern Recognit.*, vol. 14, pp. 267–273, 1981.
[5] P. Baldi and R. Meir, "Computing with arrays of coupled oscillators: An application to preattentive texture discrimination," *Neural Comput.*, vol. 2, pp. 458–471, 1990.
[6] M. Bomans, K.-H. Hohne, U. Tiede, and M. Rieme, "3-D segmentation of MR images of the head for 3-D display," *IEEE Trans. Med. Imag.*, vol. 9, pp. 177–183, 1990.
[7] J. T. Bushberg, J. A. Seibert, E. M. Leidholdt, and J. M. Boone, *The Essential Physics of Medical Imaging*. Baltimore MD: Williams & Wilkins, 1994.
[8] M. M. Chang, M. I. Sezan, A. M. Tekalp, and M. J. Berg, "Bayesian segmentation of multislice brain magnetic resonance imaging using three-dimensional Gibbsian priors," *Opt. Eng.*, vol. 35, pp. 3206–3220, 1996.
[9] G. I. Chiou and J.-N. Hwang, "A neural network-based stochastic active-contour model (NNS-SNAKE) for contour finding of distinct features," *IEEE Trans. Image Processing*, vol. 4, pp. 1407–1416, 1995.
[10] L. P. Clarke et al., "Review of MRI segmentation: Methods and applications," *Magn. Resonance Imag.*, vol. 13, pp. 343–368, 1995.
[11] R. Eckhorn et al., "Coherent oscillations: A mechanism of feature linking in the visual cortex," *Biol. Cybern.*, vol. 60, pp. 121–130, 1988.
[12] M. Friedman, "Three-dimensional imaging for evaluation of head and neck tumors," *Arch. Otolaryng. Head Neck Surg.*, vol. 119, pp. 601–607, 1993.
[13] C. M. Gray, P. König, A. K. Engel, and W. Singer, "Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties," *Nature*, vol. 338, pp. 334–337, 1989.
[14] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques," *Comput. Graph. Image Processing*, vol. 29, pp. 100–132, 1985.
[15] M. Joliot and B. M. Mazoyer, "Three-dimensional segmentation and interpolation of magnetic resonance brain images," *IEEE Trans. Med. Imag.*, vol. 12, pp. 269–277, 1993.
[16] D. M. Kammen, P. J. Holmes, and C. Koch, "Cortical architecture and oscillations in neuronal networks: Feedback versus local coupling," in *Models of Brain Functions*, R. M. J. Cotterill Ed. Cambridge: Cambridge Univ. Press, pp. 273–284, 1989.
[17] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vision*, vol. 1, pp. 321–331, 1988.
[18] T. Kohonen, *Self-Organizing Maps*. Berlin, Germany: Springer–Verlag, 1995.
[19] K. Mueller and R. Yagel, "Fast perspective volume rendering with splatting by utilizing a ray-driven approach," in *Proc. IEEE*, pp. 65–72, 1996.
[20] J. C. Rajapakse, J. N. Giedd, and J. L. Rapoport, "Statistical approach to segmentation of single-channel cerebral MR images," *IEEE Trans. Med. Imag.*, vol. 16, pp. 176–186, 1997.
[21] S. Sarkar and K. L. Boyer, "Perceptual organization in computer vision: A review and a proposal for a classificatory structure," *IEEE Trans. Syst. Man Cybern.*, vol. 23, pp. 382–399, 1993.
[22] W. Singer and C. M. Gray, "Visual feature integration and the temporal correlation hypothesis," *Ann. Rev. Neurosci.*, vol. 18, pp. 555–586, 1995.
[23] H. Sompolinsky, D. Golomb, and D. Kleinfeld, "Cooperative dynamics in visual processing," *Phys. Rev.*, vol. 43, pp. 6990–7011, 1991.
[24] P. Suetens et al., "Image segmentation: Methods and applications in diagnostic radiology and nuclear medicine," *Eur. J. Radiology*, vol. 17, pp. 14–21, 1993.
[25] D. Terman and D. L. Wang, "Global competition and local cooperation in a network of neural oscillators," *Physics D*, vol. 81, pp. 148–176, 1995.
[26] M. W. Vannier, J. L. Marsh, and J. O. Warren, "Three-dimensional CT reconstruction images for craniofacial surgical planning and evaluation," *Radiology*, vol. 150, pp. 179–184, 1984.
[27] F. Verhulst, *Nonlinear Differential Equations and Dynamical Systems*. Berlin, Germany: Springer-Verlag, 1990.
[28] C. von der Malsburg, "The correlation theory of brain function," Max-Planck-Instit. Biophysical Chem., Internal Rep. 81-82, 1981.
[29] D. L. Wang, "Emergent synchrony in locally coupled neural oscillators," *IEEE Trans. Neural Networks*, vol. 6, pp. 941–948, Apr. 1995.
[30] D. L. Wang and D. Terman, "Locally excitatory globally inhibitory oscillator networks," *IEEE Trans. Neural Networks*, vol. 6, pp. 283–286, Jan. 1995.
[31] D. L. Wang and D. Terman, "Image segmentation based on oscillatory correlation," *Neural Comput.*, vol. 9, pp. 805–836, 1997 (for errata see *Neural Comput.*, vol. 9, pp. 1623–1626, 1997).
[32] G. Wiet et al., "Cranial base tumor visualization through high performance computing," in *Proc. Media Meets Virtual Reality 4, Health Care Information Age*, 1996, pp. 43–59.
[33] R. Yagel et al., "Multisensory platform for surgical simulation," in *Proc. IEEE Virtual Reality Annual Int. Symp.*, 1996, pp. 72–78.
[34] S. W. Zucker, "Region growing: Childhood and adolescence," *Comput. Graph. Image Processing*, vol. 5, pp. 382–399, 1976.