The Hong Kong Polytechnic University
香港理工大學
Department of Computing

A Study of the Synchronization and Desynchronization Dynamics of the Neural Oscillatory Network and the Applications in Scene Analysis

Gary Li Chung Lam

A thesis submitted in partial fulfillment of the requirements for the Degree of Master of Philosophy

August 2004

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

__________________(Signed)

Gary Li Chung Lam (Name of student)

Abstract

Theoretical studies of brain functions have asserted that the brain uses the temporal correlation of firing activities of neurons to represent objects in a scene. These studies have been directly supported by many experiments on the visual cortex and other brain regions. In these experiments, the brain exhibited synchronization between the firing of neurons when perceiving a scene. This points to a mechanism of neural oscillation as a representational framework in neural networks. The mechanism has been implemented using a class of neural networks called neural oscillator networks, which it is hoped will provide a physical foundation for the study of human perception as well as an effective computational model for real time scene segmentation.

This thesis proposes a neural oscillator network for scene segmentation called a solely excitatory oscillator network (SEON). SEON segments scenes in parallel by synchronizing and desynchronizing oscillators in the networks. The synchronization and desynchronization are very fast. The synchronization speed is also theoretically independent of the network size. This allows SEON to quickly and effectively segment large scenes. A further advantage of SEON is that it is able to reliably segment cluttered scenes into objects. These objects are encoded by the temporal correlation of oscillators. This encoding scheme directly implements the neural oscillation mechanism in human perception. SEON is particularly well suited to very-large-scale-integrated (VLSI) technology, and thus provides an elegant representation for real-time image processing. The performance of the proposed model is evaluated by comparing it with other contemporary methods. Experimental results show that the proposed model can segment scenes faster and more accurately than other methods.

The proposed model provides a comprehensive framework for studying human perception and selective attention. It also provides an effective method that is general enough to segment cluttered scenes (both grayscale and color) in real time. Experiments show that the proposed model can be applied to segment cluttered scenes reliably in real time.

Publications

Papers Published/Accepted

1. C. L. Li and R. S. T. Lee, "A Real-Time Scene Segmentation System Using Solely Excitatory Oscillator Networks (SEON)," accepted to appear in a *Special Issue on Real-World Applications of Data Mining of the Journal of Intelligent Manufacturing*.
2. C. L. Li and R. S. T. Lee, "Solely Excitatory Oscillator Network for Color Image Segmentation," In *Advances in Neural Networks – ISNN 2004* (Lecture Notes in Computer Science series LNCS3174), Part II, pp. 387-392, Springer-Verlag, 2004.
3. G. C. L. Li and R. S. T. Lee, "iJade Scene Segmentator - A Real-Time Scene Segmentation System Using Watershed-Based Neuro-Oscillatory Network," In *Knowledge-based Intelligent Information and Engineering Systems* (Lecture Notes in Computer Science series LNCS 3214), Part II, pp. 549-556, Springer-Verlag, 2004.
4. G. C. L. Li and R. S. T. Lee, "Scene Segmentation Using Solely Excitatory Oscillator Networks (SEON)," In *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications* (AIA 2004), Innsbruck, Austria, pp. 357-362, 2004.
5. G. C. L. Li and R. S. T. Lee, "Solely Excitatory Oscillator Networks (SEON) for Parallel Image Segmentation," In *Proceedings of the 5th ACM Postgraduate Research Day* (PGDAY 2004), Hong Kong, pp. 42-48, 2004.

Papers Submitted

1. G. C. L. Li and R. S. T. Lee, "A Solely Excitatory Oscillator Network for Scene Segmentation," submitted to *Neural Networks*, 1st revised on 12th March, 2004, 2nd revised on 4th June, 2004.

2. G. C. L. Li and R. S. T. Lee, “Color Image Segmentation Using Parallel Watershed-Based Solely Excitatory Oscillator Networks,” submitted to *International Journal of Pattern Recognition and Artificial Intelligence* (IJPRAI).

Acknowledgment

I wish to take this opportunity to express my sincere gratitude to the following bodies:

1. The Department of Computing of the Hong Kong Polytechnic University for providing necessary resources and facilities for my research and studies;
2. My supervisor Dr. Raymond S.T. Lee for his encouragement and guidance to me throughout my studies; and
3. My family, in particular, my grandmother and Vivian, for their care, love and support to me through the years of my studies.

I am certain that without the generous and tremendous support, encouragement and assistance from the above-mentioned bodies I would not be able to complete my studies in the Degree of Master of Philosophy.

Contents

List of Figures

List of Tables

Chapter 1. Introduction and Motivation of the Research

1.1. Background

Since its first introduction, neural networks have been motivated by the recognition that the brain, probably the most powerful computer (information – processing system) in the world, operates in an entirely different way from the traditional computer. With this recognition in mind, different researchers have studied neural networks for entirely different purposes. While engineers regard neural networks as a useful tool to solve more complex problems than those based on traditional techniques, neurobiologists regard neural networks as a research tool for the understanding of complex phenomena in the brain.

The viewpoints of engineers and neurobiologists are of equal importance. It is no doubt that engineers often design neural networks based on the analogy with the brain. These neural networks, although over-simplified comparing with the brain, provide elegant solutions in many areas such as pattern recognition, classification and clustering. On the other hand, neural networks modeling can give us more insight in the properties of the brain. For example, recurrent neural networks can help to explain many aspects of signal processing by the neurons that mediate the vestibulo-ocular reflex in oculomotor system (Anastasio, 1993).

This thesis proposes a neural network model that is motivated by many recent biological experiments on the visual cortex and the brain. In these experiments, the brain exhibits synchronization between the firing of different neurons when perceiving a scene (Eckhorn et. al., 1988; Gray et. al., 1989). These findings provide direct support to the temporal correlation hypothesis in neuroscience (von der Malsburg, 1981). The hypothesis asserted that the brain uses the temporal correlation of firing between different neurons to represent objects in a scene.

The temporal correlation hypothesis is implemented in a new class of neural networks called locally excitatory, globally inhibitory oscillator networks (LEGION) (Wang & Terman, 1995; Terman & Wang, 1995). In LEGION, each object is represented by a group of synchronized oscillators. Different groups, whose oscillations are desynchronized from each other, represent different objects. LEGION segments a scene by synchronizing oscillators within a group and desynchronizing oscillators between different groups. LEGION has several advantages when applied in scene segmentation. Firstly, LEGION segments scenes in parallel. Secondly, it has a strong neurobiological basis. Thirdly, it solves the binding problem that is particularly challenging for neural networks in scene segmentation.

1.2. Motivation of Research

This research is motivated by some biological experiments and studies about the visual cortex of cats (Gray et. al., 1989; Engel et. al., 1991; Yen et. al., 1999). Based on these experiments and studies, the author believes that excitatory oscillators could be an effective mechanism for synchronization and desynchronization in neural oscillator networks. Hence, one could use excitatory oscillators to achieve both synchronization and desynchronization in neural oscillator networks. Using excitatory oscillators to desynchronize oscillators has several advantages. Firstly, it could increase the speed of desynchronization. Secondly, it does not have the limitation of segmentation capacity, meaning that it can segment as many objects in the scene as possible. Thirdly, excitatory oscillators do not impose a correlation between all oscillators in the networks all the times. Fourthly, the desyncrhonization effect is localized within the desynchronized groups of oscillators, keeping it separate from the rest of the networks.

Another motivation of this research is the desire to segment cluttered scenes accurately and quickly. While LEGION has been successfully applied to segment some real images (Wang & Terman, 1997), it is not clear whether LEGION can segment cluttered scenes that are often encountered in real applications. For example, the segmentation of a human face in a cluttered scene is a very challenging problem for most segmentation methods. However, a fast scene segmentation method that is able to segment cluttered scenes accurately is fundamental to many image processing applications. It is hoped that oscillator networks could provide an effective solution for segmenting cluttered scenes in real times.

1.3. Aim of the Research

The aim of this research is to investigate and develop an oscillator network, which can be applied to segment cluttered scenes in real time and can help to explain the phenomenon of selective attention.

A solely excitatory oscillator network (SEON) has been developed to provide a feasible solution to cluttered scene segmentation. SEON is suitable for very-large-scale-integrated (VLSI) technology. Hence, it provides a feasible solution for real-time image processing. In addition, SEON is biological plausible. In particular, it could be used to explain selective attention of objects when the brain perceives a scene.

1.4. Organization of the Thesis

This thesis consists of six chapters, organized as follows:

1. Chapter 1 presents a general introduction to the whole thesis.
2. Chapter 2 introduces neural oscillator networks. This chapter gives the background and motivation of oscillator networks. It also describes the LEGION network and some properties of the network, and discusses the applications of neural networks in scene segmentation.
3. Chapter 3 introduces scene segmentation methods. This chapter provides the background of scene segmentation and discusses the traditional grayscale segmentation methods. In particular, watershed segmentation is discussed in details. Some general issues of color segmentation are also introduced. Finally, the watershed parallel segmentation is briefly introduced.

4. Chapter 4 proposes a solely excitatory oscillator network (SEON). This chapter discusses the motivation of SEON. It also introduces SEON by describing the single oscillator model, the synchronization and desynchronization dynamics between oscillators in SEON. In addition, this chapter discusses how SEON segments both grayscale and color scenes.
5. Chapter 5 presents the experimental results of the proposed model. This chapter introduces what experiments and simulations have been performed. The simulation results of the synchronization and desynchronization dynamics are discussed. It also presents the experimental results of grayscale segmentation and color segmentation.
6. Chapter 6 concludes this thesis. This chapter summarizes the contributions of this thesis. It also discusses how SEON can be used as a framework for the study of selective attention. Furthermore, it discusses the limitations of the proposed model and suggests some further research to enhance the model. Finally, this chapter presents some preliminary results of the suggested further research.

Chapter 2. Neural Oscillator Network

2.1. Introduction

2.1.1. Background and Motivation of Neural Oscillator Networks

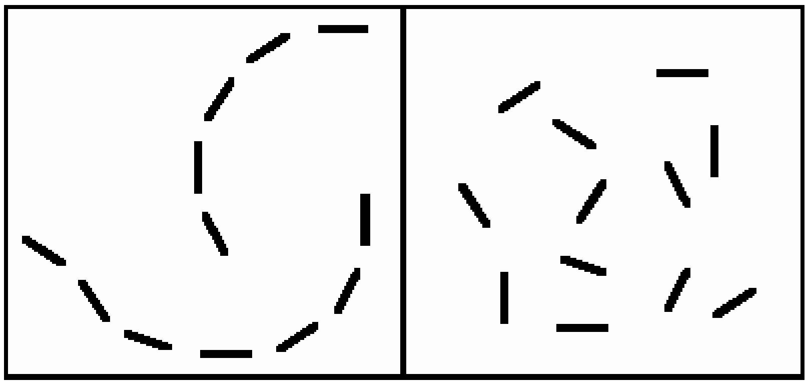
Research on neural networks has been motivated right from its inception by the recognition that the brain operates in an entirely different way from the conventional computer. It is generally believed that the brain is a highly complex, nonlinear and parallel information-processing system. It can organize its elements, known as neurons, to perform certain complex tasks such as pattern recognition, perception and motor control. For instance, human vision is the function of the visual system to provide a representation of the environment around us so that we can interact with the environment. Specifically, the brain consistently accomplishes the routine tasks of perceptual segmentation and recognition within very short periods of times, whereas the computer accomplishes the tasks of much lesser complexity with much longer times.

The design of a neural network is motivated by analogy with the brain, which is a living proof that fault tolerant parallel processing is not only physically possible but also fast and powerful. Neurobiologists look to neural networks as a research tool for the interpretation of neurobiological phenomena in the brain. On the other hand, engineers look to neurobiology for new ideas to solve complex problems. For example, some researchers have built electronic chips that mimic the structure of the retina (Mahowald, & Mead, 1989; Boahen, 1996). The chips emulate the retina in that it can adapt locally to changes in brightness, detect edges, and detect motion. Therefore, neurobiological analogy provides a hope and belief, and to a certain extent an existence of proof, that physical understanding of neurobiological structures could have a productive influence on the art of technology.

Recently, many experiments on the visual cortex and the brain have found that the brain exhibits synchronization between the firing of different neurons when perceiving a scene (Eckhorn et. al., 1988; Gray et. al., 1989). These findings directly support the temporal correlation hypothesis in neuroscience (von der Malsburg, 1981). The hypothesis asserted that the brain use the temporal correlation of firing between different neurons to represent objects in a scene. This points to a mechanism of temporal correlation as an encoding scheme to represent objects in neural networks.

To implement the temporal correlation hypothesis in neural networks, an elegant way to represent multiple objects in neural network is suggested by oscillatory correlation (von der Malsburg & Schneider, 1986; Wang & Terman, 1995). Oscillatory correlation is a special form of the temporal correlation hypothesis (von der Malsburg, 1981), where an object is represented by the temporal correlation of firing between different neurons that detect different features of objects. The oscillatory correlation is realized in a new class of oscillatory networks called locally excitatory, globally inhibitory oscillator networks (LEGION) (Wang & Terman, 1995; Terman & Wang, 1995). In LEGION, each object is represented by a group of synchronized oscillators. Different groups, whose oscillations are desynchronized from each other, represent different objects. LEGION can achieve synchronization and desynchronization rapidly, and has been successfully applied to parallel image segmentation with various neural oscillators (Wang & Terman, 1995; Terman & Wang, 1995; Wang, 1995; Campbell & Wang, 1996; Chen & Wang, 2002; Campbell et. al., 1999; Fox et. al., 2001). LEGION has a strong neurobiological basis and desirable parallel nature. It provides a computationally effective framework to implement the oscillatory correlation theory and segment scenes in real time.

Another contribution of LEGION is that it solves the binding problem in neural networks. The binding problem refers to how the coherence of an object, generally as a large set of features, is represented in a neural network. Since neural networks are well-established in pattern classification, most neural networks regard scene segmentation as a pixel classification problem (Pal & Pal, 1993). The problem of this approach is that only local information is used in classification. However, proper segmentation of objects depends on the image context of the objects (Wang & Terman, 1997). For example, Figure 1(a) and 1(b) have twelve line segments arranged in two different ways. Since both images have the same line segments, local classification leads to same segmentation results, even though the two images are different. LEGION solves the binding problem by implementing the temporal correlation hypothesis.



(a) (b)

Figure 1. An example of viewing scene segmentation as a classification problem.

Figure 1(a) and 1(b) have twelve line segments arranged in different ways. Classification leads to the same segmentation results even though the two images are different (Wang & Terman, 1997).

2.1.2. Organization of the Chapter

This chapter is organized in two sections. Section 2.2. introduces locally excitatory, globally inhibitory oscillator networks (LEGION) and their properties. Section 2.3. reviews the neural network approaches in parallel scene segmentation including both the non-oscillatory and oscillatory approaches.

2.2. Locally Excitatory, Globally Inhibitory Oscillator Networks (LEGION)

2.2.1. Model Description

LEGION is a two dimensional network of locally connected oscillators with a global inhibitor, as shown in Figure 2. Each oscillator is connected locally to its four nearest neighbours. The global inhibitor is connected to all oscillators in the network. In LEGION, an object is represented by a synchronized group of oscillators, where as different objects are represented different desynchronized groups of oscillators.

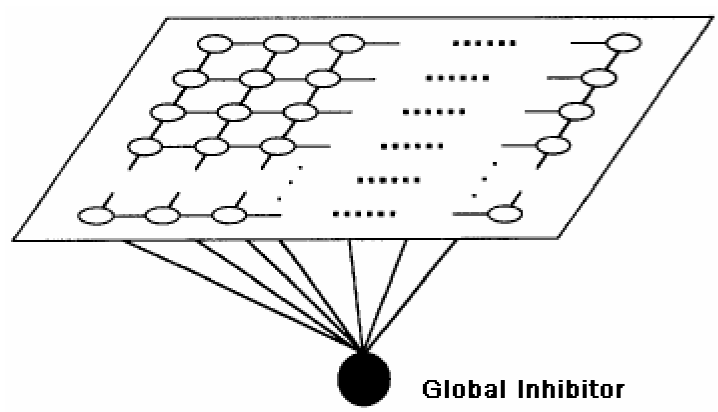


Figure 2. Network Architecture of LEGION with four nearest neighbours connectivity.

Each oscillator is connected to its four nearest neighbours. The global inhibitor is connected to all oscillators in the network (Wang & Terman, 1995).

In LEGION, each oscillator i in a network is a relaxation oscillator (Wang & Terman, 1995):

$$\dot{x}_i = 3x_i - x_i^3 + 2 - y_i + \rho + I_i + S_i \tag{1}$$

$$\dot{y}_i = \varepsilon\left[\gamma\left(1 + \tanh\left(x_i / \beta\right)\right) - y_i\right] \tag{2}$$

where x_i and y_i are the activities of an excitatory unit and an inhibitory unit respectively. ρ is the amplitude of a Gaussian noise term. I_i is an external stimulation. S_i represents the coupling from other oscillators in the network. ε, γ and β are parameters. Since the degree of equation (1) is 3, the x-nullcline of equation (1) is a cubic curve. The y-nullcline of equation (2) is a sigmoid function.

The parameter ε is a small positive number, which yields two time scales that define the property of relaxation oscillator. When $I_i > 0$, equation (1) and (2) become a stable limit cycle, which oscillates between a silent phase (small x values) and an active phase (large x values). Due to ε, the activity of oscillator changes slowly within the same phases but alternates quickly between two phases. The quick alternation from one phase to another phase is called jumping.

The term S_i is the overall input from the network. It is two parts: local excitatory term and global inhibitory term. The excitatory term represents the excitatory input from the four nearest neighbours. The inhibitory term represents the inhibitory input from the global inhibitor.

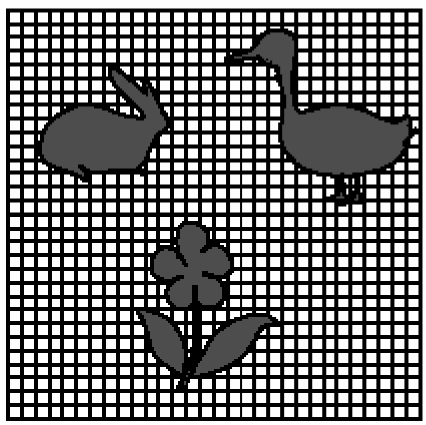
LEGION exhibits a mechanism of selective gating (Wang & Terman, 1995). When an oscillator jumps to active phase, its activity spreads to its neighbouring oscillators, which further spread the activity to their neighbours. This leads to synchronization of a group of oscillators in LEGION. On the other hand, oscillating groups inhibit each other through global inhibition so that at most one group can be in the active phase at a time. This leads to desynchronization between different groups of oscillators.

2.2.2. Properties of LEGION

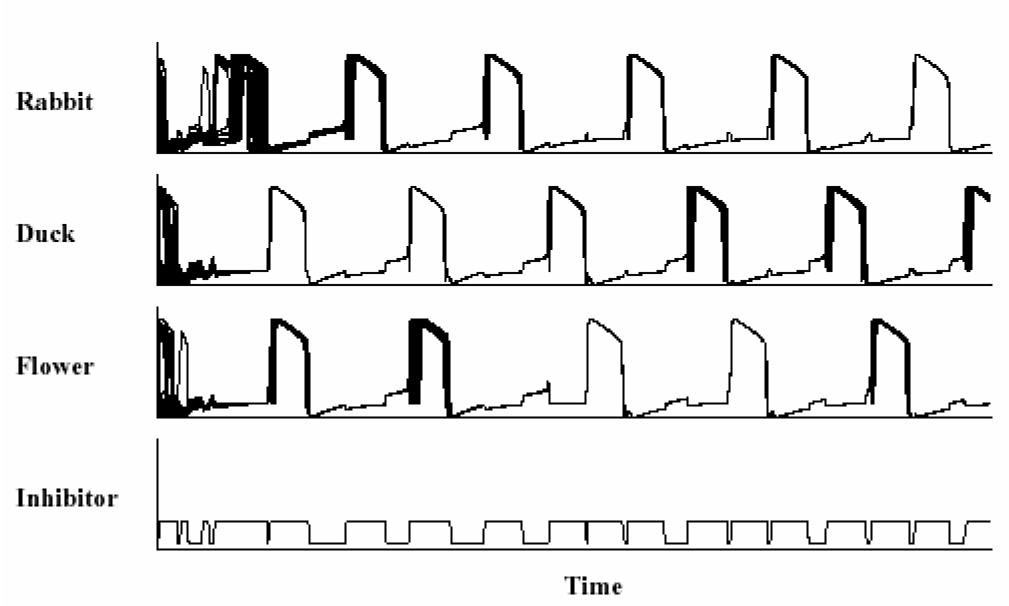
An important property of LEGION is the fast synchronization and desynchronization speed. It has been proved that LEGION can achieve synchronization and desynchronization in no greater than m cycles of oscillations, where m is the number of objects in the scene, as long as m does not exceed the segmentation capacity. The segmentation capacity refers to the maximum number of objects that can be segmented by LEGION. The typical capacity is about 5 to 7 (Wang & Terman, 1995). The capacity of LEGION was later enhanced using other oscillators (Campbell & Wang, 1996; Campbell et. al., 1999; Fox et. al., 2001).

Figure 3 illustrates the selective gating mechanism. Figure 3(a) contains three objects, namely a rabbit, a duck and a flower. This figure is inputted to a 30 by 30 LEGION. The stimulated oscillators will oscillate, while those non-stimulated oscillators will not oscillate. Figure 3(b) shows the temporal evolution of every stimulated oscillator. The activities of the oscillators representing each object are combined together in the figure. Although the oscillators in the networks start with random initial values, the synchronization within each object and the desynchronization between different objects are achieved in only two oscillation periods.

Oscillatory correlation provides a unique way to label objects in a scene. As shown in Figure 3(b), segmentation in LEGION is performed in time. That is, each object pops out at a distinct time from the network and different objects alternate periodically in time. Once an object is in the active phase, all of its features, but none of the ones from other objects, are simultaneously available for late processing such as attention and recognition. Hence, it solves the binding problem by using synchronization to simultaneously represent different features of an object and desynchronization to represent different objects in a network (Wang & Liu, 2002).



(a)



(b)

Figure 3. An example of selective gating mechanism in LEGION.

(a) A 30 by 30 input image containing three objects. (b) The temporal evaluation of the three synchronized groups of oscillators representing each object and the global inhibitor (Wang & Terman, 1995).

Since LEGION was first introduced (Wang & Terman, 1995), it has been extended significantly. For example, LEGION has been extended to distinguish between main image regions and noisy fragments (Wang & Terman, 1997). To improve the performance of LEGION, an algorithm that follows LEGION dynamics has also been abstracted (Wang & Terman, 1997). Another enhancement is to use weight adaptation scheme to remove noise and preserve features in an image (Chen & Wang, 2000). The scheme is also transformed to a feature-preserving smoothing algorithm. In addition to these enhancements, the segmentation capacity and speed have also been enhanced using other oscillators (Campbell & Wang, 1996; Campbell et. al., 1999; Fox et. al., 2001).

2.3. Applications in Parallel Scene Segmentation

2.3.1. Non-Oscillatory Neural Network Approaches

Being inherently distributed and parallel, neural networks can be used as an efficient method for parallel scene segmentation. However, very little research has been conducted into applying neural networks to scene segmentation. There are two major reasons for this. The first reason is that it is difficult to represent segmentation results in neural networks, which is the binding problem mentioned in section 2.1.1. The second reason for the inadequacy of research in this area is the lack of efficient segmentation methods that can take full advantage of the parallel properties of neural networks while preserving the reliability of the segmentation results. Some early non-oscillatory approaches of neural network in parallel scene segmentation are discussed in this section. The oscillatory approach will be discussed in the next section.

In an early study, Boltzmann machine was used to separate figures from background (Sejnowski & Hinton, 1987). The network has two types of binary units: figure units and edge units. The units are locally connected with fixed weights. These local connections represent local cooperation with a connected region and local competition between figure and background. Two types of input are required for the network: bottom-up input that contains edges information and top-down input that provides information to select a figure in an image. The output of the system is the units representing a figure. Boltzmann machine produces desired results in small synthetic images. However, as pointed out by the authors, it is not clear whether this machine can be used to segment real images.

Another neural network model is FBF model (Grossberg & Wyse, 1991). This model consists of two systems: a feature contour system and a boundary contour system. The feature contour system detects local features different filters and performs diffusion within an image region. The boundary contour system detects local edges and performs contour completion. The model worked well in small images but its performance on real images is not clear.

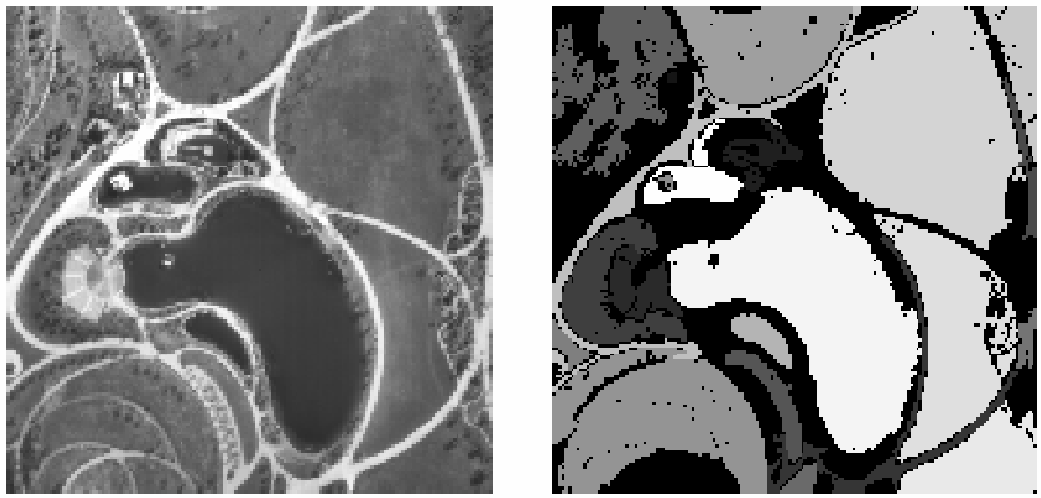
Some other methods use Hopfield network (Lin et. al., 1996), multi-layer perceptions (Mozer et. al., 1992) and Kohonen self-organizing map (Bhandarkar et. al., 1997). However, all these methods have had only limited success. The segmentation boundaries of these methods are often not continuous, and require a network training pre-processing procedure and an edge linking post-processing procedure.

2.3.2. Oscillatory Neural Network Approaches

Most early oscillatory models use harmonic oscillators and all-to-all connections to achieve synchronization. The fundamental problem of these models is that critical information about the topology of features is lost.

To solve the limitation of all-to-all connectivity, some researchers proposed to use local connectivity to model perceptual organization. A locally connected network was constructed to solve the problem of all-to-all connectivity (Sporns et. al., 1991). Reentrant connection is used to achieve proper synchronization. However, the network has only been tested using synthetic data.

As alluded to earlier, LEGION and its variants have been successfully applied in segmenting various real grayscale images, such as medical and satellite images, texture images, and image sequences (motion) (Wang & Terman, 1997). Figure 4 shows typical segmentation results of LEGION (Wang & Terman, 1997). Figure 4(a) shows a grayscale aerial image. Figure 4(b) shows the result of segmentation. The image is segmented into 23 regions. In the simulation, different segments pop out sequentially. As shown in Figure 4(b), almost all major regions are successfully segmented.



(a) (b)

Figure 4. A typical segmentation results of LEGION.

(a) A grayscale aerial image. (b) A segmented image. It consists of 23 regions after segmentation (Wang & Terman, 1995).

Chapter 3. Scene Segmentation

3.1. Introduction to Scene Segmentation

3.1.1. Background of Scene Segmentation

One of the most challenging tasks in computer vision is scene segmentation. Given its importance, scene segmentation has been extensively studied (Adams & Dischof, 1994; Ballard, 1981; Baugher & Rosenfeld, 1986; Brummer, 1991; Canny, 1986; Chang & Li, 1994). The main goal is to divide an image into parts that have a strong correlation with objects or areas of the real world contained in the image.

Most traditional scene segmentation methods only deal with grayscale scenes. These methods generally are based on one of the two basic properties of intensity values: discontinuity and similarity (Gonzalez, 1992). In the first category, the approach is to partition a scene based on the abrupt change in intensity, such as edges. Edge-based segmentation is a typical example in this category. In the second category, the approach is to partition a scene into regions that are similar according to a set of predefined criteria. Thresholding and region-based segmentation are examples in this category. In addition, some methods, such as morphological watershed segmentation, partition a scene based on both discontinuity and similarity. In general, these methods perform better than the other two approaches because they use both the concepts of the two approaches.

With the rapid advance of computational power of computers, color scene segmentation has attracted more attention than before. At present, most color scene segmentation methods are extended from the gray level segmentation methods by implementing them in different color spaces (Cheng et. al., 2001; Pal, & Pal, 1993). Therefore, the selection of color spaces is a major issue in color scene segmentation. The widely used RGB color space is not suitable for scene segmentation due to the high correlation between its RGB components and the lack of direct information for color (Gonzalez, 1992). Some color spaces, such as HVC color space, describe colors in terms that are practical for human interpretation (Hafner et. al., 1995). These color spaces may provide a better representational framework of color for scene segmentation. All of the above mentioned methods for grayscale and color segmentation are inherently sequential. Some partially parallel methods have been proposed (Manjunath & Chellappa, 1993) but parallel segmentation remains relatively unresearched.

3.1.2. Organization of the Chapter

The main body of this chapter is organized in three sections. Section 3.2. introduces the grayscale scene segmentation methods. The concept of watershed segmentation method is also discussed. Section 3.3. describes the techniques in color scene segmentation and introduces the HVC color space. Section 3.4 discusses the parallel watershed transformation, which is one of the few effective parallel segmentation methods.

3.2. Grayscale Scene Segmentation

3.2.1. Traditional Segmentation Methods

According to the dominant features employed, grayscale scene segmentation methods can be generally divided into three types – thresholding (Beghdadi et. al., 1995; Brink, 1995; Glasbey, 1993), edge-based segmentation (Ballard, 1981; Baugher & Rosenfeld, 1986; Brummer, 1991; Canny, 1986; Furst, 1986) and region-based segmentation (Adams & Dischof, 1994; Chang & Li, 1994; Kurita, 1995; Laprade, 1998).

Thresholding is the simplest scene segmentation process and has the advantage of being computationally inexpensive and fast. In thresholding, a brightness constant called a threshold is used to segment objects. Many modifications of thresholding exist, such as local thresholding, band thresholding, semi-thresholding and multi-thresholding (Glasbey, 1993; Gonzalez, 1992; Banks, 1990). The main problem of thresholding is that it is difficult in general to find appropriate threshold values, especially when the images have histograms that cannot be partitioned effectively.

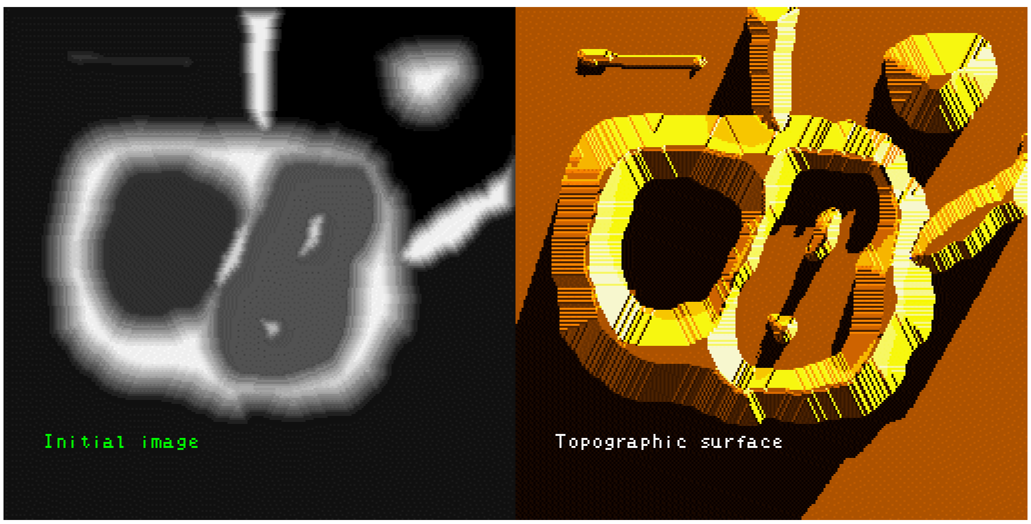
Edge-based segmentation relies on edges found in an image by edge detecting operators (Gonzalez, 1992; Banks, 1990). The most common problem of this method is an edge presence in locations where there is no border, and no edge presence where a real border exists. This problem is usually caused by noises or inappropriate information about an image. Also, a computationally expensive post-processing procedure of edge linking, such as Hough transform (Ballard, 1981; Gonzalez, 1992), is often needed to assemble edge pixels into meaningful edges.

Region-based segmentation constructs regions directly by iteratively merging or splitting pixels into connected regions in accordance with some homogeneity criteria such as the average gray-level of the region, colour properties, and simple textures properties (Gonzalez, 1992; Banks, 1990). This method is generally better in noisy images, where it is difficult to detect borders. In addition to the preceding three methods, more recent research combining two or more of these methods (Ahuja, 1996; Zhu & Yuille, 1996) has suggested promising segmentation results.

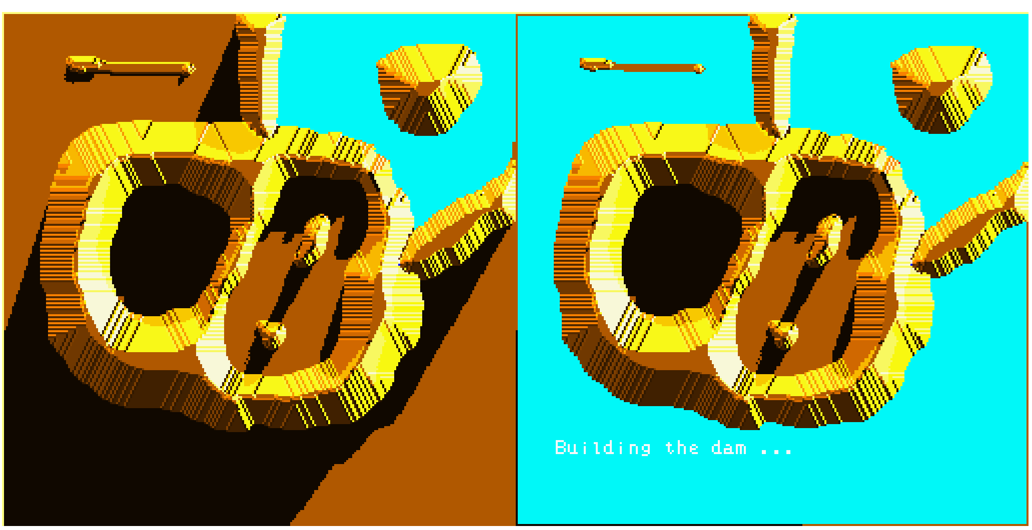
3.2.2. Morphological Watershed Segmentation and the Use of Markers

Watershed segmentation embodies many of the concepts of the three methods introduced in the preceding section and often produces more stable segmentation results, including continuous boundaries (Gonzalez, 1992; Banks, 1990; Bieniek & Moga, 2000). Having continuous boundaries obviates the need for the expensive post-processing procedure of edge linking. However, due to noises and other local irregularities of the gradient, direct application of watershed segmentation often leads to oversegmentation. This problem can be solved by the use of markers (Bleau & Leon, 2000; Banks, 1990; Gonzalez, 1992; Pal & Pal, 1993), which also provides a simple framework for incorporating knowledge-based constraints in the segmentation process.

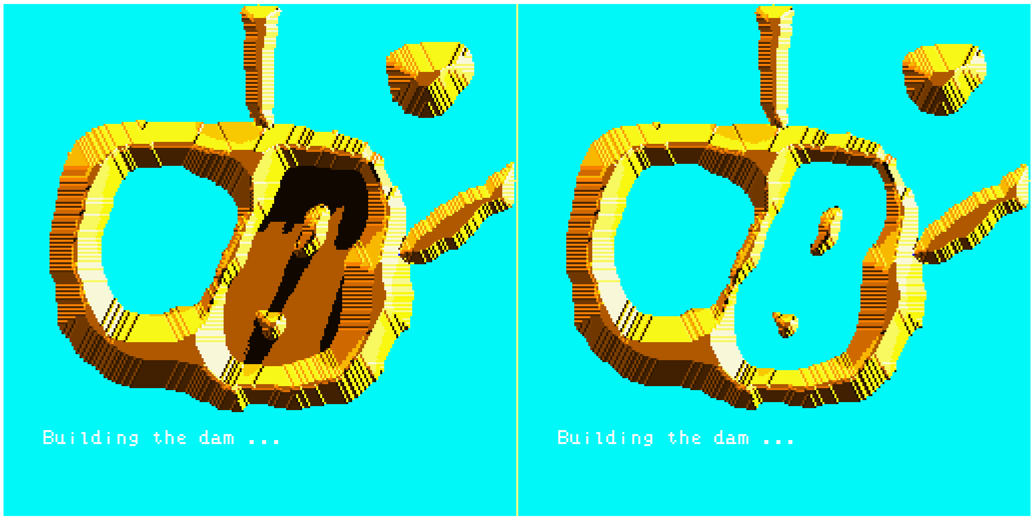
In watershed segmentation, a grayscale image can be considered as a topographic surface. If this surface is flooded from its minima and, if the merging of the waters coming from different sources is prevented, the image is partitioned into two different sets: the catchment basins and watershed lines. The idea is illustrated in Figure 5. Figure 5(a) shows a simple grayscale image and Figure 5(b) is a topographic view, where the height of the mountains is proportional to gray-level values in the image. Figures 5(c) to 5(f) show the different stages of flooding. Figure 5(g) shows the final watershed lines of the image.



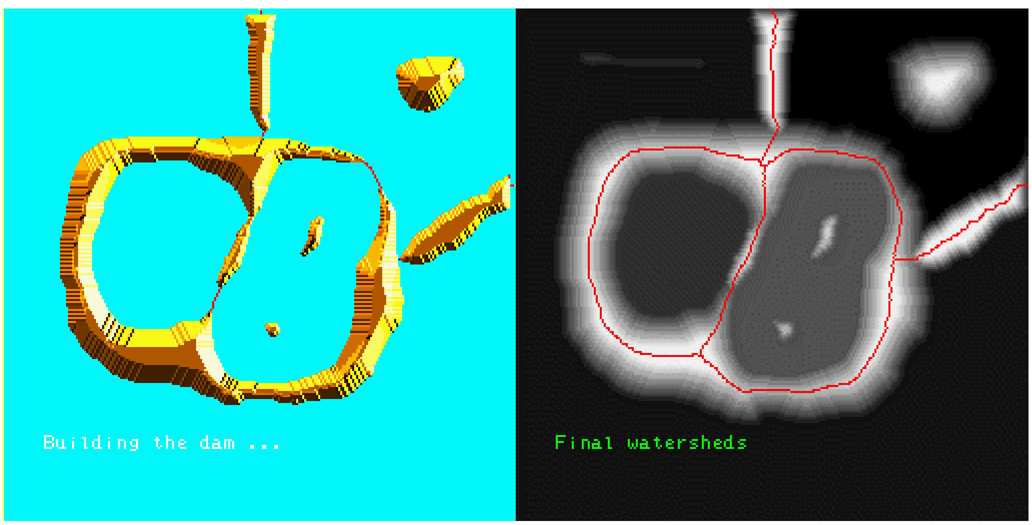
(a) (b)



(c) (d)



(e) (f)

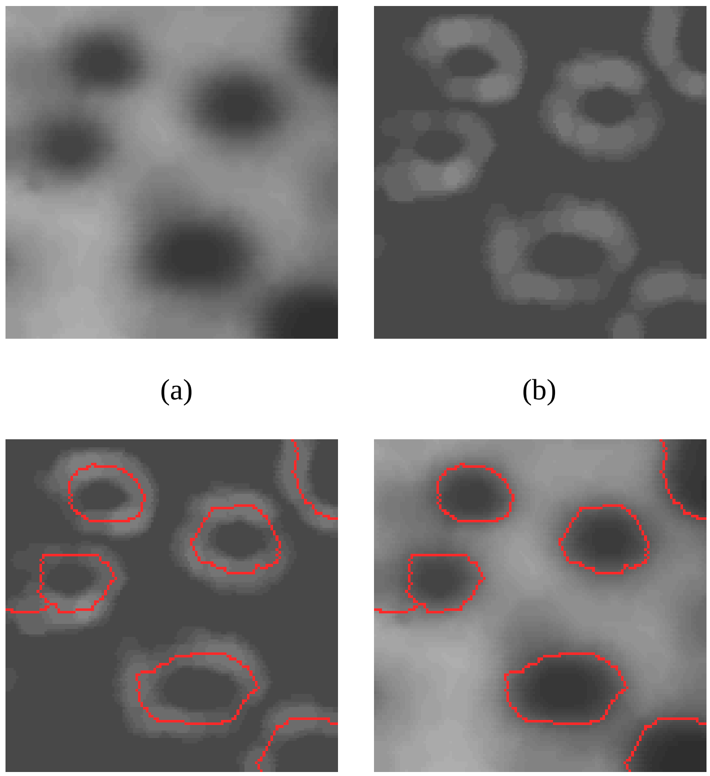


(g) (h)

Figure 5. Topographic view of watershed segmentation.

(a) Original image. (b) Topographic view (c) – (g) five stages of flooding. (h) Final watershed lines (Beucher & Meyer, 1992).

If watershed segmentation is applied to the gradient of an image, the catchment basins correspond to the homogenous gray level regions of the image. For example, an image and its gradient are shown in Figures 6(a) and 6(b), respectively. Application of the watershed segmentation results in the watershed lines of the gradient image shown in Figure 6(c). These segmentation boundaries are superimposed on the original image in Figure 6(d). It should be noted that the segmentation boundaries are continuous as mentioned at the beginning of this section.

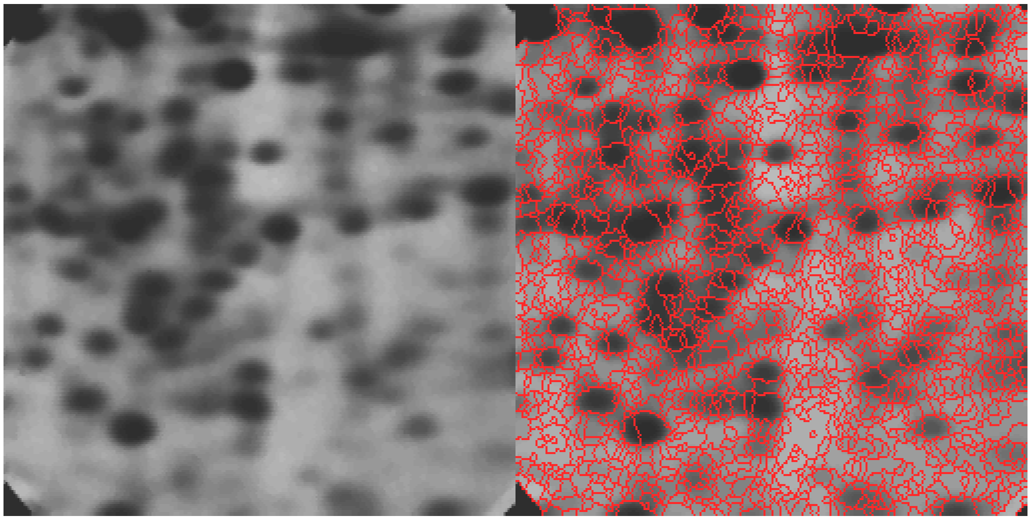


(a) (b) (c) (d)

Figure 6. An example of watershed segmentation.

(a) Original image. (b) Image gradient. (c) Watershed lines. (d) Watershed lines superimposed on original image (Beucher & Meyer, 1992).

Although watershed segmentation is a powerful segmentation method, a direct application of the method often produces the problem of oversegmentation due to noise and other local irregularities in the gradient. Figure 7 shows that oversegmentation can be very serious and makes the segmentation result useless. The oversegmented image generally has a large number of segmented regions.

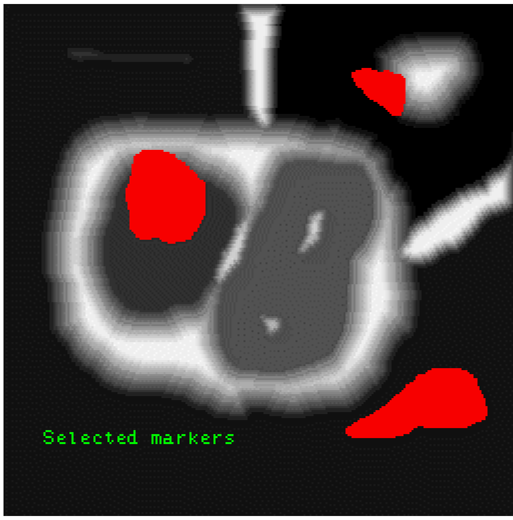


(a) (b)

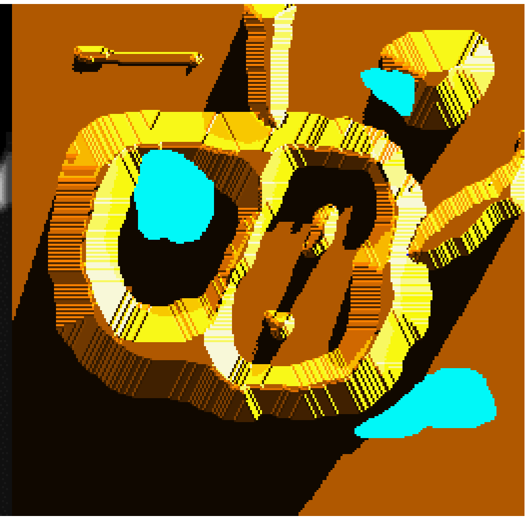
Figure 7. An example of oversegmentation.

(a) Original Image. (b) Oversegmented image (Beucher & Meyer, 1992).

To solve the problem of oversegmentation, a mechanism that limits the number of allowable regions is required. An approach used to control oversegmentation is the use of markers. Markers define the regions where flooding starts. Other non-marked regions are not flooded at the beginning of the algorithm. Figure 8(a) shows an image with three markers. Figure 8(b) shows that flooding starts from the three marked regions. Figures 8(c) to 8(e) shows the flooding around the three marked regions. Figure 8(f) shows the final watershed lines. It should be noted that the markers have controlled the final watershed lines so that only three regions are generated.

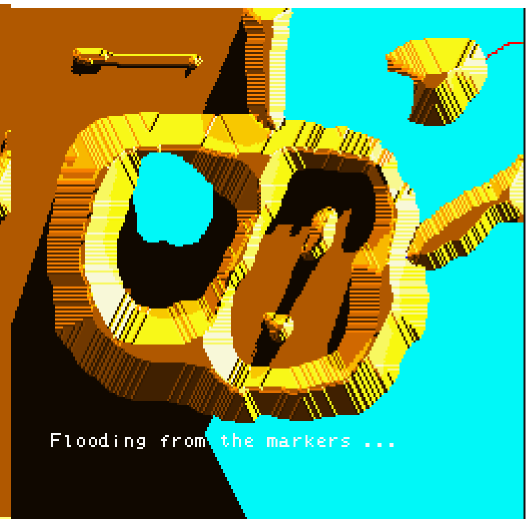


(a)

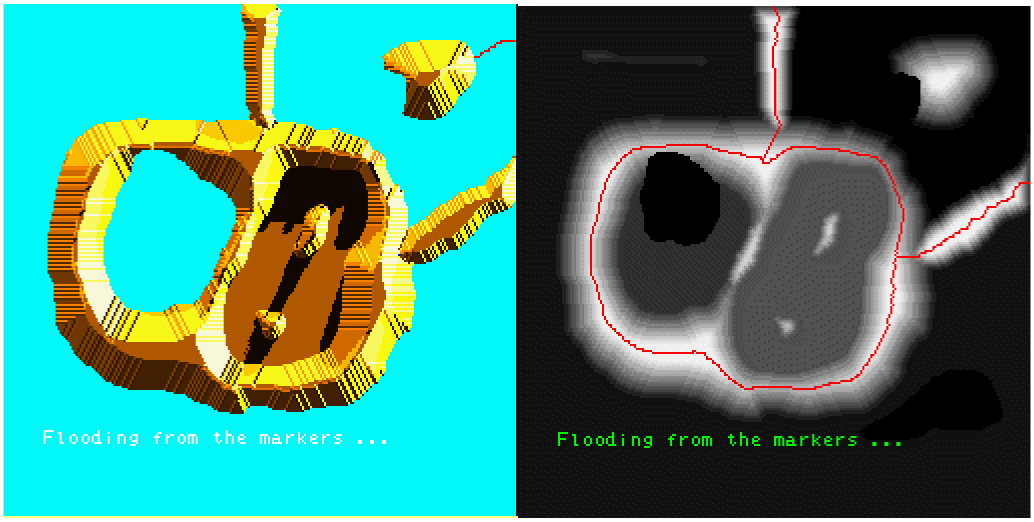


(b)

(c)



(d)

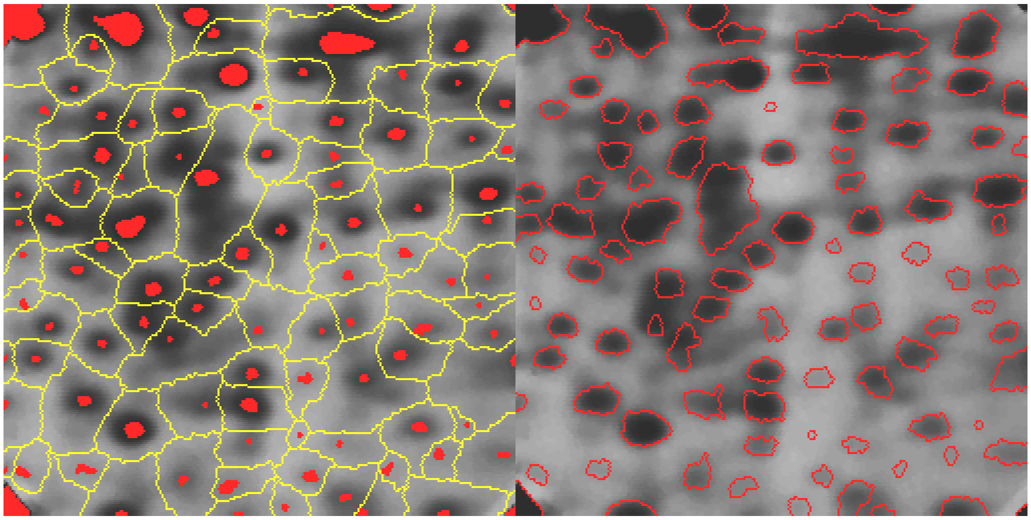


(e) (f)

Figure 8. Marker-controlled watershed segmentation.

(a) Figure with three markers. (b)-(e) The flooding around the three markers (f) Final Watershed lines (Beucher & Meyer, 1992).

With the use of markers, the oversegmentation problem in Figure 7 can be solved as shown in Figure 9. The markers in Figure 9(a) are selected based on the gray-level values and connectivity. In general, marker selection can use more complex descriptions such as size, shape, location, relative distances, texture context, and so on (Gonzalez, 1992). It should be noted that using markers brings a prior knowledge to bear on the segmentation problem. Hence, watershed segmentation provides a framework that can make effective use of prior knowledge in the segmentation process, which is a significant advantage of the method (Gonzalez, 1992).



(a) (b)

Figure 9. An example of marker-controlled watershed segmentation.

(a) Figure with selected markers. (b) Final watershed lines (Beucher & Meyer, 1992).

3.3. Color Scene Segmentation

3.3.1. General Concerns of Color Segmentation Methods

There are four principle reasons that account for the increased interest in color image segmentation (Cheng et. al., 2001; Gonzalez, 1992; Pal, & Pal, 1993). Firstly, color images contain more information than gray level images, which often simplify the tasks of image analysis. Secondly, human can discern many more shades of color than shades of gray. Thirdly, the computational power of personal computers is increasing rapidly, allowing them to process color images efficiently. Fourthly, and perhaps most importantly, most of the images people use in real life are actually color images. However, there has been little research into color image segmentation (Cheng et. al., 2001; Pal, & Pal, 1993). At present, most color image segmentation methods are extended from the gray level segmentation methods by implemented in different color spaces (Cheng et. al., 2001; Pal, & Pal, 1993).

One central issue in color image segmentation is the choice of a suitable color space (Cheng et. al., 2001). The traditional and widely used RGB color space is ideally suited for hardware implementation and display. However, due to the high correlation between its RGB components and the lack of direct information for color, it is not good for color image segmentation. The high correlation between RGB components makes the three components dependent on each other. There is also a strong association between the RGB components and intensity. This makes it very difficult to use RGB space to discriminate between highlights, shadows and shading in color images (Cheng et. al., 2001; Gonzalez, 1992). Likewise, human does not discern colors by giving the percentage of their corresponding RGB values (Gonzalez, 1992).

3.3.2. Hue, Value, Chroma (HVC) Color Space

When calculating color difference based on color perceptual similarity, a color space describing colors close to human perception is essential. The hue, value, chroma (HVC) color space describes colors in terms that are close to human color perception and practical for human interpretation (Hafner et. al., 1995). As a result, the color descriptions are natural and intuitive to human. It has been found that, with human judgment of the similarity between the query and database images, the HVC space obtains the best performance in the experiments of different color spaces (Hafner et. al., 1995). Another advantage of HVC space is that the National Bureau of Standards (NBS) unit color difference is defined in HVC space (Judd, 1963), which calculates the color difference between two colors based on human color perceptual similarity.

Table 1 shows the relationship between the human color perception and the NBS distance (Judd & Wyszeki, 1963). Two colors are perceived almost the same by human if the NBS value is less than 1.5. If the NBS distance is less than the threshold, two pixels are regarded as similar.

NBS Value	Human Perception
0 – 1.5	Almost the same
1.5 – 3.0	Slightly different
3.0 – 6.0	Remarkably different
6.0 – 12.0	Very different
> 12.0	Different color

Table 1. The relationship between human color perception and NBS Value distance.

When calculating the NBS values, the RGB color values are transformed to HVC color values. Given a set of RGB color value (*R, G, B*), the transformation from the RGB color value to the HVC color value (*H, V, C*) is shown as follows (Hafner et. al., 1995, Judd & Wyszeki, 1963):

---

1. Transform the RGB value to xyz value.

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 0.620 & 0.178 & 0.204 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.056 & 0.942 \end{pmatrix} \times \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

2. Based on the xyz value, calculate H_1, H_2, and H_3.

$$H_1 = f(x) - f(y)$$
$$H_2 = f(z) - f(y)$$
$$H_3 = f(y)$$

where $f(x) = \dfrac{18.51u}{u + 17.58\left(1 + \dfrac{5.146u}{u + 30.07}\right)}$

3. Calculate S_1 and S_2.

$$
\begin{aligned}
M_1 &= H_1 \\
M_2 &= 0.4H_2 \\
S_1 &= (8.880 + 0.996\cos\theta)M_1 \\
S_2 &= (8.025 + 2.558\cos\theta)M_2
\end{aligned}
$$

where $\theta = \tan^{-1}\left(\frac{M_2}{M_1}\right)$

4. Obtain the HVC color value.

$$
\begin{aligned}
H &= \tan^{-1}\left(\frac{S_2}{S_1}\right) \\
V &= H_3 \\
C &= \sqrt{S_1^{\,2} + S_2^{\,2}}
\end{aligned}
\qquad (3)
$$

For two colors (H_1, V_1, C_1) and (H_2, V_2, C_2) in the HVC space, the NBS color distance is defined by (Judd & Wyszeki, 1963):

$$
d_{NBS} = 1.2\sqrt{(2C_1C_2)(1-\cos(2\pi\Delta H/100)) + (\Delta C)^2 + (4\Delta V)^2} \qquad (4)
$$

where $\Delta H = |H_1 - H_2|, \Delta V = |V_1 - V_2|,$ and $\Delta C = |C_1 - C_2|$.

The color difference d_{NBS} can be used to determine whether the colors of two pixels are considered different in human perception. For example, if we adopt a NBS value of 1.5 as threshold ($\theta_{NBS} = 1.5$) in Table 1, two colors are different if $d_{NBS} > \theta_{NBS}$.

3.4. Parallel Watershed Segmentation

All of the methods outlined in this chapter so far are inherently sequential. There is relatively little research in parallel scene segmentation. Some partially parallel methods have been proposed (Manjunath & Chellappa, 1993) but the area remains relatively unresearched.

Moga and Gabbouj developed a parallel watershed transformation method (Alina & Moncef, 1997; Alina, 1998; Alina et. al., 1998). The algorithm starts by detecting the regional minima of the gradient image. The image is then lower-complete transformed and represented as an acyclic directed graph or forest. Pixels are represented as vertices and edges exist between two vertices if their lower-complete values satisfy the given ordering relation. The labels of minima are propagated to non-minimum pixels through the edges.

Parallel watershed transformation has been successfully implemented in multi-processor architecture to image segmentation. The method, however, is not fully parallel. To compute the lower distances of every plateau, a sequential breadth-first scanning is still required for every processor, which requires heavy sequential computation for each individual processor in a multi-processor architecture.

Chapter 4. Solely Excitatory Oscillator Network (SEON)

4.1. Introduction

4.1.1. Network Architecture

In this chapter, a solely excitatory oscillator network (SEON) is proposed for scene segmentation. SEON uses the Peskin oscillator as a basic computational unit. The architecture of SEON consists of a two dimensional matrix of Peskin oscillators and a global excitatory separator (ES). Each oscillator is locally connected to its eight nearest neighbours, forming a locally connected 2D grid. The global excitatory separator is connected to all oscillators in the network, as shown in Figure 10. The network uses a group of synchronized oscillators to represent an individual segment in a scene. Different segments are represented using different desynchronized groups. This allows SEON to use the temporal correlation of oscillators to represent segments in a scene.

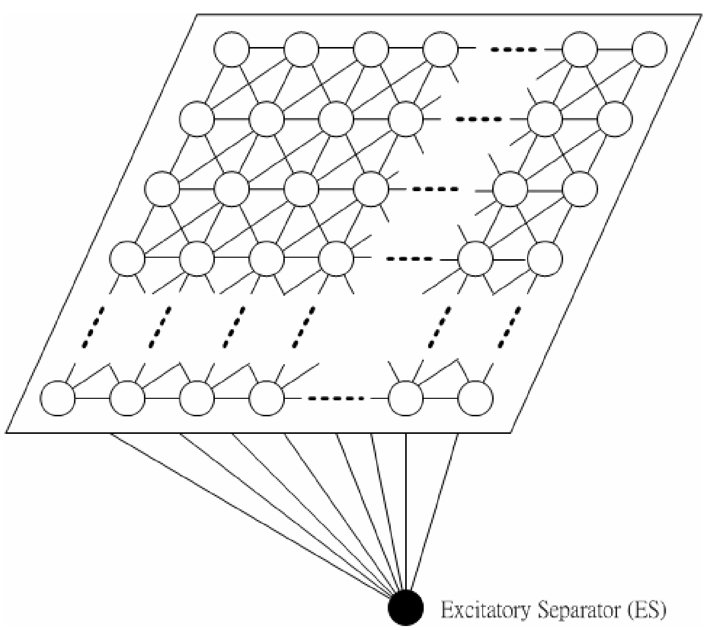


Figure 10. Network architecture of SEON with eight nearest neighbours connectivity.

Each oscillator is connected to its eight nearest neighbours. The excitatory separator (indicated by the black circle) is connected to all oscillators in the network.

4.1.2. Motivation of the Proposed Model

The proposed SEON model is motivated by some biological experiments and studies about the visual cortex of cats (Gray et. al., 1989; Engel et. al., 1991; Yen et. al., 1999).

In most traditional oscillator networks, the structure that all oscillators have connections both to and from a global inhibitor is not biologically plausible (Yen et. al., 1999). Traditional oscillator networks usually use a global inhibitor to desynchronize oscillators in the networks (Wang & Terman, 1997; Campbell et. al., 1999). To desynchronize oscillators, the global inhibitor continuously sends inhibition signals to all oscillators in the networks all the times. The connections between the global inhibitor and all oscillators always exist. However, such connection structure is not biological plausible (Yen et. al., 1999). To provide a more biologically plausible model, the connection between the global inhibitor and all oscillators in the network should not always exist. A global excitatory separator (ES) is proposed as a more biologically plausible model. ES desynchronizes oscillators by sending excitation signals. In most of times, ES stays inactive and does not have connections to all oscillators. It only has connections with individual oscillators when they reach the threshold of ES (θ_{ES}). If any oscillator i reaches ES threshold, ES will send a pulse to that particular oscillator i instantaneously for desynchronization. After the pulse excitation, the connection between the oscillator i and ES is closed. Hence, most of the times, ES does not have connection with all oscillators in the networks, which is more biologically plausible.

Many experiments showed that different objects do not have temporal correlation at all (Engel et. al., 1991). When oscillator networks use global inhibitions to desynchronize different objects, the global inhibitor correlates the whole network all the times by continuously sending inhibitions to all oscillators. However, this is not consistent with the experimental findings. In SEON, different segments (objects) do not have temporal correlation most of the times.

Desynchronization effect is localized by ES. ES only sends a pulse to a single oscillator. The oscillator then triggers the chain reaction of firings within its group for desynchronization. Desynchronization only affects the desynchronized groups of oscillators and does not affect the rest of the network. After desynchronization, the desynchronized groups do not have any temporal correlation.

While many experiments supported the temporal correlation hypothesis that individual objects in a scene are represented by synchronization (Gray et. al., 1989), the experiments did not support that global inhibition is the mechanism for desynchronization. In fact, von der Malsburg has stated that the use of global inhibitions is only to prevent global synchronization of all oscillators (von der Malsburg, 1981). Hence, if other mechanisms can achieve desynchronization without using inhibition, these mechanisms can also be used to represent different objects in the correlation theory. The use of excitatory separator therefore does not conflict with existing experimental findings and correlation theory. By using ES as a desynchronization mechanism, SEON implements the correlation theory in a more biologically plausible manner.

4.1.3. Organization of the Chapter

The rest of the chapter describes the solely excitatory oscillator network (SEON) in details. Section 4.2. introduces the Peskin oscillator, which is the basic computational unit of SEON. Section 4.3. describes the compulsory firing mechanism for synchronization. Section 4.4. deals with the global excitatory separator (ES) as a desynchronization mechanism. Section 4.5. and 4.6. describe how SEON segments grayscale and color scenes respectively.

4.2. A Single Oscillator Model

The Peskin oscillator was originally devised for the study of the synchronization of cardiac rhythms and was later adapted to deal with synchronizing neurons by Mirollo and Strogatz

(Mirollo, & Strogatz, 1990). It is a simplified form of the integrate-and-fire oscillator that has been adopted in LEGION by Campbell et. al. (Campbell et. al., 1999) to perform scene segmentation.

The activity of a single Peskin oscillator *i* (Peskin, 1975), which is simplified from an integrate-and-fire oscillator, is described by a voltage-like state variable x_i, which obeys

$$\dot{x}_i = S_i - \gamma x_i, \quad i = 1,\ldots,n \tag{5}$$

and assume the values

$$0 \le x_i \le 1 \tag{6}$$

where S_i is a non-negative stimulus of the oscillator (i.e. $S_i \ge 0$) and γ is a constant.

The oscillator behaves differently depending on the value of S_i. At an initial time, individual value x_i obeying inequality (6) is prescribed by equation (5). If $S_i>0$, x_i will increase and reach the value 1 as the time passes. Then, the oscillator fires by emitting a pulse and x_i jumps back to zero. At an infinitesimally later time t^+, the values x_j of all other coupled oscillators are increased by an amount ε or the other oscillators fire when reaching $x_j=1$. In short,

$$x_i(t) = 1 \Rightarrow x_j(t^+) = \min(1, x_j(t) + \varepsilon), \quad \forall j \in N(i) \tag{7}$$

where *N(i)* represents all oscillators coupled with the *i*th oscillator. In SEON, the couplings between oscillators are local, and only exist between the nearest neighbours. *N(i)* is defined as the nearest eight neigbhors of oscillator *i*. In addition, a pulse is transmitted instantaneously between oscillators, and the propagation speed is infinite. Hence, the model includes no conduction delays. The differential equation (5) of a single uncoupled oscillator can be solved analytically. The trajectory of an oscillator is $x_i(t)=S_i/\gamma - K\exp(-\gamma t)$ given that $S_i>0$. As $t\rightarrow\infty$, $x_i\rightarrow S_i/\gamma$, which is the upper limit of x_i. Since the oscillator will fire when $x_i=1$, $S_i/\gamma >1$ is a necessary condition for firing, which is important for the synchronization. If $S_i=0$, x_i decays exponentially toward zero, no firing will occur.

4.3. Synchronization Mechanism

Synchronization of a group of oscillators in SEON is achieved by a compulsory firing mechanism. This mechanism uses a different firing convention of Peskin oscillator: all oscillators can fire in the same iteration no more than once. Under this mechanism, when oscillator *i* reaches its threshold, it will fire and reset its value to zero. The oscillator *i* then sends an instantaneous pulse to its coupled oscillator *j*. If oscillator *j* is induced to fire, then its value will also reset to zero. According to the firing convention, all oscillators can only fire once in the same iteration. Hence, when firing its coupled oscillators, oscillator *j* will not fire oscillator *i* again, even though oscillator *i* is coupled with it.

The purpose of the mechanism is to prevent the large coupling strength problem in a network of integrate-and-fire oscillators Model A called by Hopfield and Herz (Hopfield & Herz, 1995; Campbell et. al., 1999). In Model A, when the coupling strength ε nears 1, firings of oscillators change the relative phases of oscillators very slowly. The oscillators will keep firing and resetting frequently without a significant phase change. This limits the value of coupling strength. However, a large coupling strength is very important for rapid synchronization.

The interaction of oscillators in the network is defined by the interaction equation (Hopfield & Herz, 1995). An oscillator *j* fired by oscillator *i* resets its value with the following equation

$$x_j(t^+) = x_j(t^-) + \varepsilon_{ji} - 1 \qquad (8)$$

where ε_{ij} is the coupling strength from oscillator *i* to oscillator *j*. Oscillator *i* will also receive excitatory pulse from oscillator *j* and increase from zero (since it has just been fired) to ε_{ij}. If the coupling strength ε nears 1, the oscillators will fire and reset frequently but change their relative phases very slowly. The synchronization time will then be infinite. At $\varepsilon = 1$, the equation (5) becomes meaningless because the oscillators are constantly firing and resetting.

While an excessively large coupling strength destroys synchronization, a large coupling strength is crucial for fast synchronization. If the coupling strength is too small, the synchronization process will usually take a significantly longer time to complete. The proposed firing convention in the mechanism, together with equation (7), allows the use of large coupling strength ($\varepsilon \geq 1$) to achieve synchronization instantaneously when an oscillator fires.

By the mechanism, a locally coupled network of n Peskin oscillators can synchronize in one cycle when pulse strength is sufficiently large. In the following synchronization algorithm, x_i is voltage-like variable of oscillators described in equation (5). To indicate whether each oscillator is fired or not, the oscillator has an internal state variable of either “fired” or “unfired”.

---

1. The oscillator values x_i are randomly chosen from the range [0,1].
2. All oscillators are set to an “unfired” state.
3. Repeat this step until no oscillators reach threshold

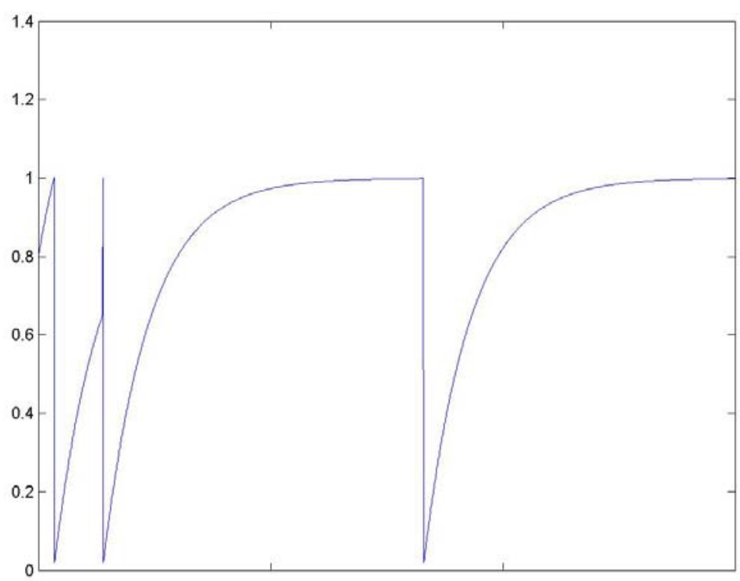
   If any oscillator i reaches its threshold $\theta_i = 1$

      The oscillator i fires by setting itself to a “fired” state. Its value x_i is reset to zero. The values x_j of its coupled oscillators j that are in an “unfired” state are increased by equation (7).
4. Go to step 2.

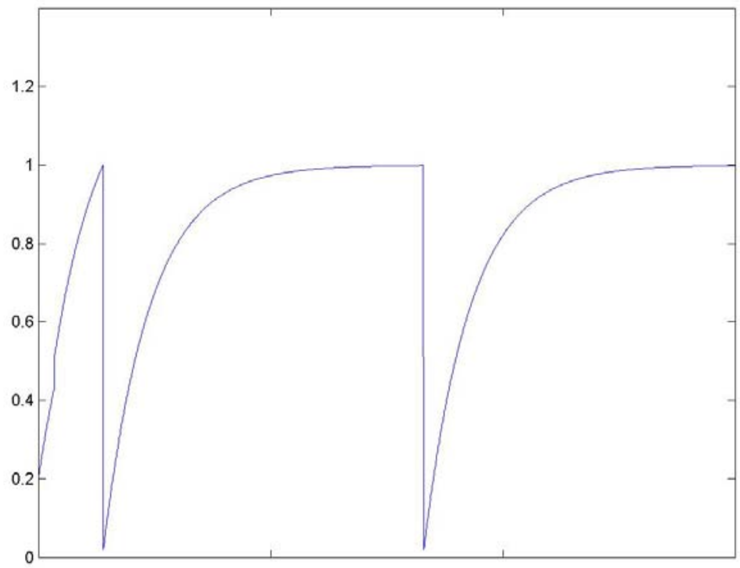
At the beginning of each iteration, all oscillators are set to the “unfired” state, which allows them to fire. When the oscillator fires, it is set to a “fired” state. An oscillator in a “fired” state will not be fired by other oscillators. This allows all oscillators to reset to zero after the chain of firings. The oscillators will then be synchronized because they have the identical dynamic.

The temporal evolution of a pair of pulse-coupled Peskin oscillators with $\varepsilon < 1$ is shown in Figure 11. At the beginning, the two oscillators 1 and 2 have different initial values of 0.8 and

0.2, respectively. When the value of one oscillator, in this case x_1, reaches the threshold, the oscillator fires and resets its value to zero. It then emits an instantaneous pulse to oscillator 2. The value of oscillator 2 (x_2) will be increased by equation (7). Since $\varepsilon < 1$, x_2 increases only by ε, and oscillator 2 does not fire. Therefore, the two oscillators do not synchronize at the first firing. At the second firing, the firing of oscillator 2 causes oscillator 1 to fire and reset the value to zero. This causes them to synchronize. Since an oscillator can only fire once in the same iteration, oscillator 2 will not fire back at oscillator 1. Hence, the values of both oscillators are zero. They then become perfectly synchronized. They remain synchronized thereafter because their dynamics are identical. However, if $\varepsilon \geq 1$, oscillator 2 is guaranteed to fire and then reset its value to zero at the first firing. Both oscillators then reset their values to zero, enabling them to be synchronized thereafter.



(a)



(b)

Figure 11. Synchronization of a pair of oscillators.

(a) The temporal evolution of oscillator 1 with an initial value of x_1=0.8. (b) The temporal evolution of oscillator 2 with an initial value of x_2=0.2. Both oscillators use $S_o = 1$, $\gamma = 0.5$ and $\varepsilon = 0.1$.

Figure 12 shows the temporal evolution of 25 oscillators with $\varepsilon \geq 1$ in a 5 by 5 network. When oscillator *i* reaches the threshold, it fires the neighbouring oscillators and resets its value to zero. Given that $\varepsilon \geq 1$, the firing of oscillator *i* will always cause the neighbouring oscillators to fire and to reset their values to zero. Since each oscillator can only fire once in the same iteration, the neighbouring oscillators will fire their eight nearest neighbours excluding oscillator *i*. The process repeats, leading to a chain reaction of firing to all coupled oscillators, as shown in Figure 13. The chain reaction causes all coupled oscillators to synchronize in one cycle. Since synchronization is accomplished with the instantaneous chain reaction of firing, synchronization will not slow when the number of oscillators increases. In fact, in contrast to LEGION where the synchronization speed will decrease when the network size increases (Wang & Terman, 1995;

Terman & Wang, 1995; Wang, 1995; Campbell & Wang, 1996; Chen & Wang, 2002; Campbell et. al., 1999; Fox et. al., 2001), the speed of synchronization is independent of the number of oscillators *n*. Theoretically, no matter how many oscillators are in the segment, the segment can synchronize instantly by firing one oscillator.

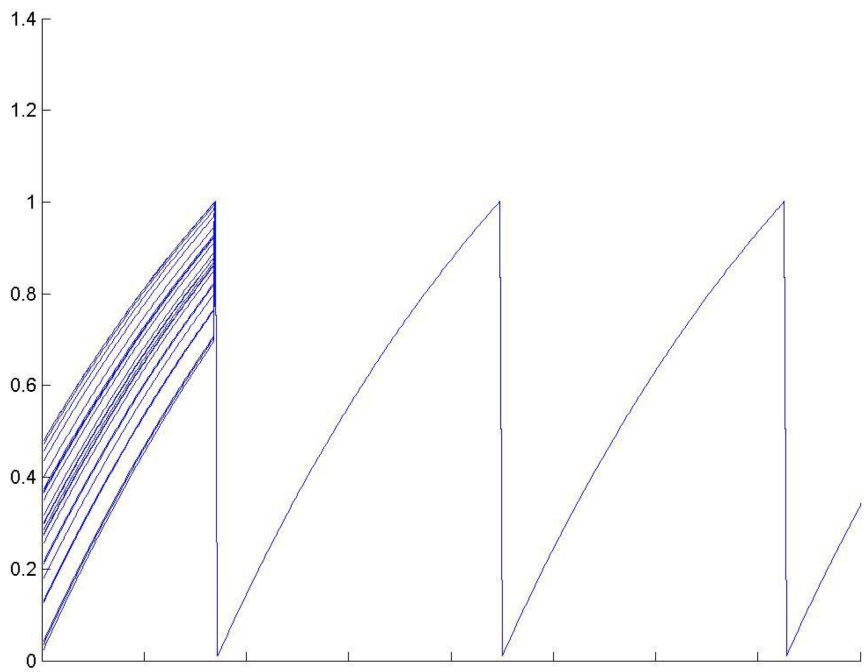


Figure 12. The temporal evolution of 25 oscillators in a 5 by 5 2D network.

All oscillators have different initial values randomly chosen from the range [0,0.5]. All oscillators use S_o = 1, γ = *0.5* and ε = 1.

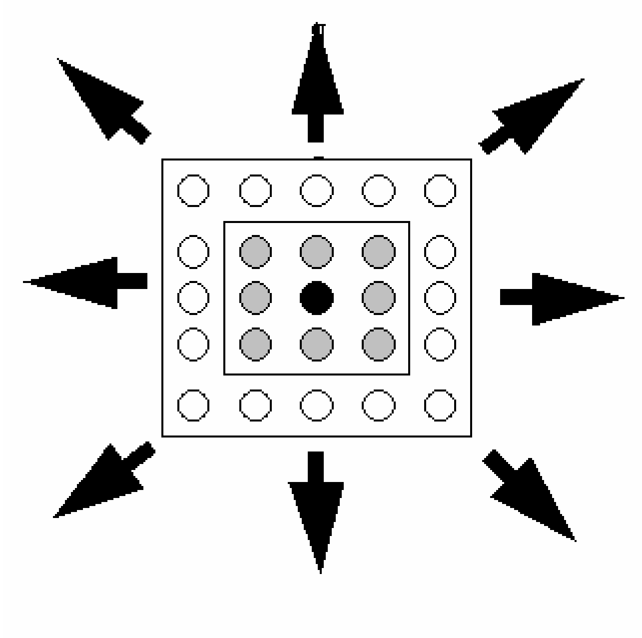


Figure 13. An instantaneous chain reaction of firing from an oscillator to its neighbours.

When a single oscillator fires, it triggers a chain reaction to achieve synchronization instantly.

4.4. Desynchronization Mechanism

In SEON, desynchronization is achieved by an excitatory separator (ES) that is connected to all oscillators in the network, as shown in Figure 10. When two segments (groups of oscillators) accidentally synchronize, ES will send an excitatory pulse to an oscillator *i* in one of the segments. This leads to early firing of oscillator *i*. The early firing triggers the chain reaction of firings, which in turn triggers the early synchronization of the segment containing oscillator *i*. The early synchronization creates a phase difference between the two segments, which effectively desynchronize them. Hence, each firing of ES can desynchronize one segment among all accidentally synchronized segments. The process will repeat to desynchronize all synchronized segments. In the following desynchronization algorithm, θ_{ES} is the threshold of ES and θ_i is the threshold of oscillator *i*. To indicate whether ES is busy or not, ES has an internal state variable of either "busy" or "ready". T_{busy} is the number of iterations that ES stays at the "busy" state. $T_{seperation}$ is the number of iterations that ES will stay at the "busy" state.

---

1. The ES is set to the "ready" state.
2. Repeat this step until the end of simulation

   If ES is in a "busy" state and T_{busy}=$T_{seperation}$

      ES is set to the "ready" state.

   If any oscillator reaches the ES threshold θ_{ES} where $\theta_{ES} < \theta_i$ and ES is in the "ready" state

      ES sends an excitatory pulse to the oscillator. The state is set to "busy" and T_{busy} is set to zero. The value of the oscillator is increased to 1. This causes the oscillator to fire.

   If ES is in a "busy" state

      $T_{busy} = T_{busy} + 1$

Initially, ES is set to the "ready" state. When any oscillator reaches the threshold of ES (θ_{ES}), ES sends an excitation to the oscillator. θ_{ES} can be any positive real number smaller than the threshold of oscillators, and is set to 0.8. Also, the state and T_{busy} are set to "busy" and zero, respectively. T_{busy} is used to count the duration in which the ES stays at the "busy" state. Upon receiving the excitation, the oscillator value is set to 1, which triggers the early synchronization of the segment containing the oscillator. SEON uses the internal state variable to ensure that, at any given time, only one segment will perform an early synchronization when reaching the ES threshold (θ_{ES}). When ES is in a "busy" state, it will not send excitation to the oscillators even when the oscillators reach the threshold. When T_{busy} reaches a specific duration $T_{separation}$, ES is set to "ready" state again, allowing it to desynchronize other segments. Hence, $T_{separation}$ represents the minimum phase difference between two desynchronised segments. Since the network will eventually separate all segments with at least a $T_{separation}$ phase difference and there is no limitation on the simulation duration, theoretically there is no limitation on the number of objects the network can segment.

Figure 14 shows the temporal evolution of two accidentally synchronized segments. When the activity of an oscillator in a segment reaches the ES threshold (θ_{ES} = 0.8), the excitatory separator will send an excitatory pulse to that oscillator, leading the oscillator to fire, and instantaneously triggering the chain reaction of synchronization. This pre-empts the synchronization of oscillators within the same segment and creates a phase difference between the two synchronized segments. In other words, the excitatory separator pre-empts the synchronization of one segment between the two synchronized segments, which effectively desynchronize them. Since the excitatory separator will only fire one oscillator at any given time and stay at “busy” for a specific duration $T_{separation}$, it will desynchronize one segment among all of the synchronized segments at each firing.

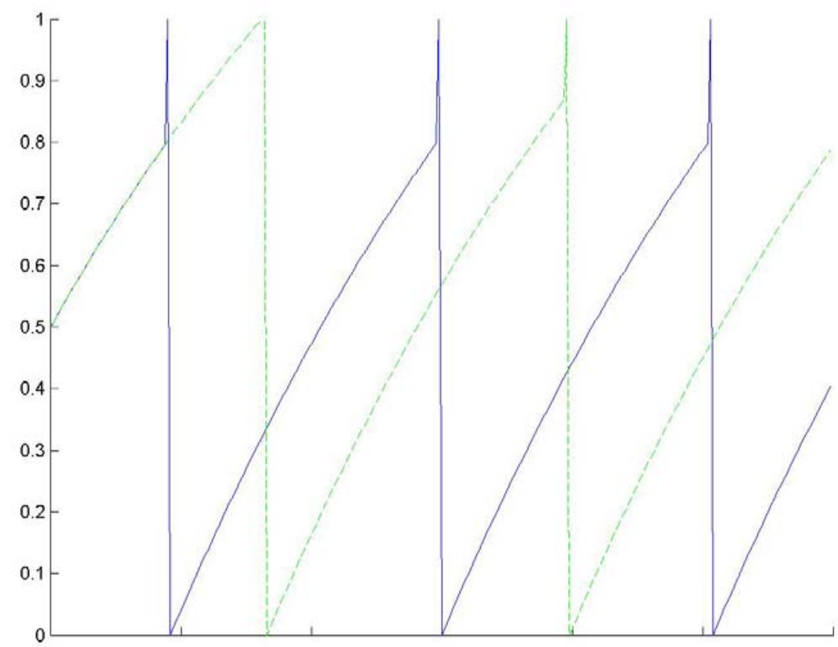


Figure 14. The temporal evolution of two accidentally synchronized segments (θ_{ES} = 0.8).

When ES sends an excitation to an oscillator in a segment, there are three possible outcomes – desynchronization effect, no synchronization effect and resynchronization effect (The stimulations of the three outcomes are given in section 5.2). Desynchronization occurs when the excitation is sent to an oscillator in some globally synchronized segments. This desynchronization effect is the main purpose of the ES. If an excitation is sent to an oscillator in a single desynchronized segment, there will either be no synchronization effect or there will be a resynchronization effect. In most cases, an excitation to an oscillator in a desynchronized segment will result in no synchronization effect. This happens when the desynchronization process has been completed. However, the excitation of ES may lead to accidental resynchronization in the early state of desynchronization. An excitation to an oscillator in a desynchronized segment may cause two desynchronized segments to accidentally resynchronize again. The desynchronization algorithm cannot prevent the problem of accidental resynchronization. However, the resynchronized segments will be desynchronized again immediately by the ES in the next cycle. The problem of resynchronization will only appear in the early state of desynchronization. After all of the segments have been completely desynchronized, they will progress in the same sequence periodically. In this stage, an excitation from ES will only result in no synchronization effect for the segments.

A rule of thumb is used to increase the desynchronization speed and prevent resynchronisation. Since ES sends an excitation every $T_{separation}$ duration, in order to increase the probability of the complete desynchronization of all globally synchronized segments in the first cycle, the parameters of the model can be adjusted such that

$$T > 3(n-1) \times T_{separation} \tag{9}$$

where T is the period of the oscillators. This inequality increases the probability that the globally synchronized segments will receive a sufficient number of excitations (n-1 excitations) from ES in one cycle. This increases the probability that the desynchronization process completes in one cycle. After all of the segments are desynchronized, they will progress in the same sequence periodically. The excitations from ES will not create any synchronization effect except to shorten their periods. In this stage, accidental resynchronization will not occur again. The inequality increases the probability of complete desynchronization in the first cycle, and no resynchornization will happen afterward.

4.5. Grayscale Scene Segmentation

Using the idea of sequential watershed segmentation (Gonzalez, 1992; Banks, 1990; Bieniek & Moga, 2000), SEON utilizes its properties of parallel nature and invariant synchronization speed to produce reliable segmentation results in parallel. As in the watershed segmentation, SEON is applied to the gradient of an image, rather than to the image itself. In the following segmentation algorithm, p_i is the gray level value of the ith pixel and is in the range [0,255]. $T(i)$ represents the four nearest neighbours of oscillator i (four nearest neighbors are used here, rather than eight nearest neighbors. It will be explained later in this section).

---

1. The stimulus values S_i of all oscillators are set to 1.
2. For gray level value k from 0 to 255

    For any oscillator i with gray level value k

    If the oscillator i has not been marked

    For $j \in T(i)$

    If no neighbor with $p_j \leq k$ belongs to any region

    Create new regional minimum by marking the oscillator i to an unique number M_r.

    Else if the neighbors with $p_j \leq k$ belongs to only one region

    Mark the oscillator i to belong to that region. Create connections between oscillator i and j. The connection is described by equation (7).

    Else if the neighbors with $p_j \leq k$ belongs to more than one regions

    If one of the neighbors is marked as “boundary region” or there exist some neighbors with $p_j < k$

    Mark the oscillator i as “boundary region” by setting S_i to 0.

    If all neigbhors satisfy $p_j = k$

    Mark the oscillator i to belong to any one of the regions and merge the regions. Create connections between oscillator i and j. The connection is described by equation (7).

In order to segment the scenes, the segmentation algorithm determines how the connection is defined and the values of S_j are chosen. Two oscillators have a connection only if they are neighbors (SEON uses the eight nearest neighbors of oscillator *i* defined in *N(i)* here) and if they belong to the same region. The connection between the two oscillators is defined using equation (7). In equation (7), when oscillator *i* fires, the values x_j of all connected oscillators *j* are increased by an amount ε or the other oscillators fire when reaching their thresholds (x_j=1). The stimulus S_i of oscillator *i* is set to 1 if the oscillator *i* belongs to a region. S_i is set to 0 if the oscillator *i* belongs to a "boundary region". Each segment in the scene is therefore represented by a group of synchronized active oscillators ($S_i=1$). Different groups, whose oscillations are desynchronized from each other, represent different segments. Boundary regions are represented by inactive oscillators ($S_i=0$).

The method is not completely parallel because it runs sequentially for every gray-level step. However, at each gray-level step, all oscillators perform segmentation concurrently based on their four nearest neigbhors. Each oscillator only compares its gray level value with its four nearest neighbors, and does not carry out operations that are computationally onerous. This allows SEON to keep all oscillators as simple as possible, which is essential for an efficient neural network. Since the number of gray-level steps must be less than 256 regardless of the network size, SEON can perform segmentation rapidly in a very large network, utilizing its invariant synchronization speed property. The algorithm efficiency can be improved by using only those gray level values that exist in the images. These values can be obtained in the histogram of the images.

At the beginning of the segmentation process, if the neighbors of oscillator *i* do not belong to any region, they will be marked as a new regional minimum. When SEON runs through the gray-level steps, many small regions will be created. These small regions grow by marking their neighbors as belonging to them. If the neighbors of an oscillator belong to only one region, the oscillator is marked as belonging to that region. If the neigbhors belong to more than one region, further checking is required to distinguish between the case that they are boundary region and the case that they belong to same region. These two cases are distinguished by the fact that, at each gray-level step k, the neigbhors with $p_j \leq k$ in the same region should have the same gray level value. Therefore, these regions can be merged into a single region. The local merging operation eliminates many meaningless small segments. If $p_i < k$, a boundary is detected. The oscillator is then marked as boundary region. Boundary region is represented by the inactive oscillators by setting S_j to 0. Eventually, all oscillators will be placed in their own regions or boundary region. Since each oscillator only makes simple comparison based on the information from its four neigbhors, all oscillators have only very light computational requirements and can run concurrently at each gray-level step.

The probability of generating continuous boundaries is increased by immediately marking an oscillator as within the boundary region if at least one of its neighbors is a boundary region. Most discontinuous boundaries are merged into other regions. The resulting boundaries are therefore often continuous. The boundaries may not be one pixel thick. They may be thinly connected boundaries of several pixels thick. However, the boundaries will not be too thick because all pixels near the boundaries will be marked to the neighbor regions first, rather than to the boundary region. Only those pixels that are in the middle of the edge will be marked as boundary region.

Since *T(i)* are the four nearest neighbors, SEON can also detect vague boundaries that are only connected diagonally. The four nearest neighbors mechanism prevents the problem that other regions merge with the vague boundaries. If the eight nearest neighbors are used, when regions grow, it will merge with the vague boundaries as the region can grow by using the information from the four corner pixels. The four nearest neighbors mechanism allows SEON to detect vague boundaries that are connected diagonally because it does not use the four corner pixels.

The four nearest neighbors mechanism also makes SEON more sensitive to noises and more vulnerable to oversegmentation. The problem can be solved by the use of markers (Bleau &

Leon, 2000; Banks, 1990; Gonzalez, 1992; Pal & Pal, 1993) which has been successfully applied to solve the oversegmentation problem in watershed segmentation (Marker-controlled SEON will be discussed in section 6.5.). The eight nearest neighbors mechanism can also be used to segment noisy images.

4.6. Color Scene Segmentation

SEON can be easily extended to segment color scenes based on human color perception. In color scene segmentation, SEON performs segmentation on both the corresponding gray level scenes and HVC color space. SEON performs segmentation mainly on the gray level version of the color image, which effectively is gray level image segmentation as introduced in the previous section. To incorporate the color information in the segmentation process, SEON constructs additional segment boundaries using the color difference. NBS color distance is adopted to calculate the color difference in the HVC color space (Hafner et. al., 1995, Judd & Wyszeki, 1963). The value of 1.5 in Table 1 is used as a threshold when determining whether two colors are different (i.e. θ_{NBS} = 1.5).

SEON mainly segments the gray-level version of the color image. It discriminates different colors in the HVC space by the NBS distance to construct additional boundary based on color information. The rationale of my approach is that, while gray level image segmentation usually produces satisfactory results, it loses the rich color information during the process. However, when the gray level segmentation methods are extended to incorporate color information, they are usually over complicated and computationally inefficient. In fact, as far as image segmentation is concerned, color information is only useful in that it provides additional boundary information. If the color of the whole image is generally the same, color information is not useful for improving the segmentation results. This is the major reason why fingerprint images are mainly gray level images. Since the whole finger is the same color, color information is usually not useful for fingerprint analysis. The proposed approach effectively segments color images in parallel by simple comparison operations without the loss of color information. This allows SEON to produce reliable segmentation results in a computationally inexpensive and parallel manner.

Color information is incorporated in the segmentation by measuring the color difference $d_{NBS}(i,j)$, defined by equation (4) in section 3.3.2., between each oscillator *i* and its neighbors *j*. If $d_{NBS}(i,j) > \theta_{NBS}$, the color of oscillator *i* is different from the color of oscillator *j*. Oscillator *i* is then marked as belonging to boundary region. Since the oscillator only compares colors with its neighbors, all oscillators can compare colors concurrently. The simple NBS value comparison places a small computational burden on SEON. This allows SEON to segment color images efficiently. Obviously, by omitting this step, SEON can also be applied to gray level image segmentation.

Chapter 5. Simulations and Experiments

5.1. Introduction

5.1.1. Performance Analysis of the Proposed Model

Three kinds of experiments are carried out to study SEON. Firstly, SEON is applied to black and white image segmentation for the investigation of the oscillatory dynamics. The segmentation methods described in sections 4.5 and 4.6 are not used to simplify the studies of oscillatory dynamics of SEON. Secondly, performance analysis is conducted for grayscale image segmentation to evaluate the segmentation performance and the speed of SEON in comparison with other methods including sequential watershed segmentation (Gonzalez, 1992; Banks, 1990; Bieniek & Moga, 2000), LEGION (Wang & Terman, 1997; Campbell et. al., 1999) and parallel watershed transformation (Alina & Moncef, 1997; Alina et. al., 1998). Thirdly, similar performance analysis is also done for color image scene segmentation. Color SEON is compared with sequential watershed segmentation (Gonzalez, 1992; Banks, 1990; Bieniek & Moga, 2000), parallel watershed transformation (Alina & Moncef, 1997; Alina et. al., 1998) and RGB vector gradient method (Gonzalez, 1992).

5.1.2. Organization of the Chapter

This chapter is organized in three sections. Section 5.2. describes the simulation of oscillatory dynamics of SEON in the context of black and white image segmentation. The synchronization and desynchronization dynamics in SEON are investigated in the simulation. Section 5.3 presents the experimental results in grayscale image segmentation. The performance of SEON is evaluated in term of both segmentation capability and speed. Section 5.4 deals with the experiments of color image segmentation.

5.2. Oscillatory Dynamics Simulation

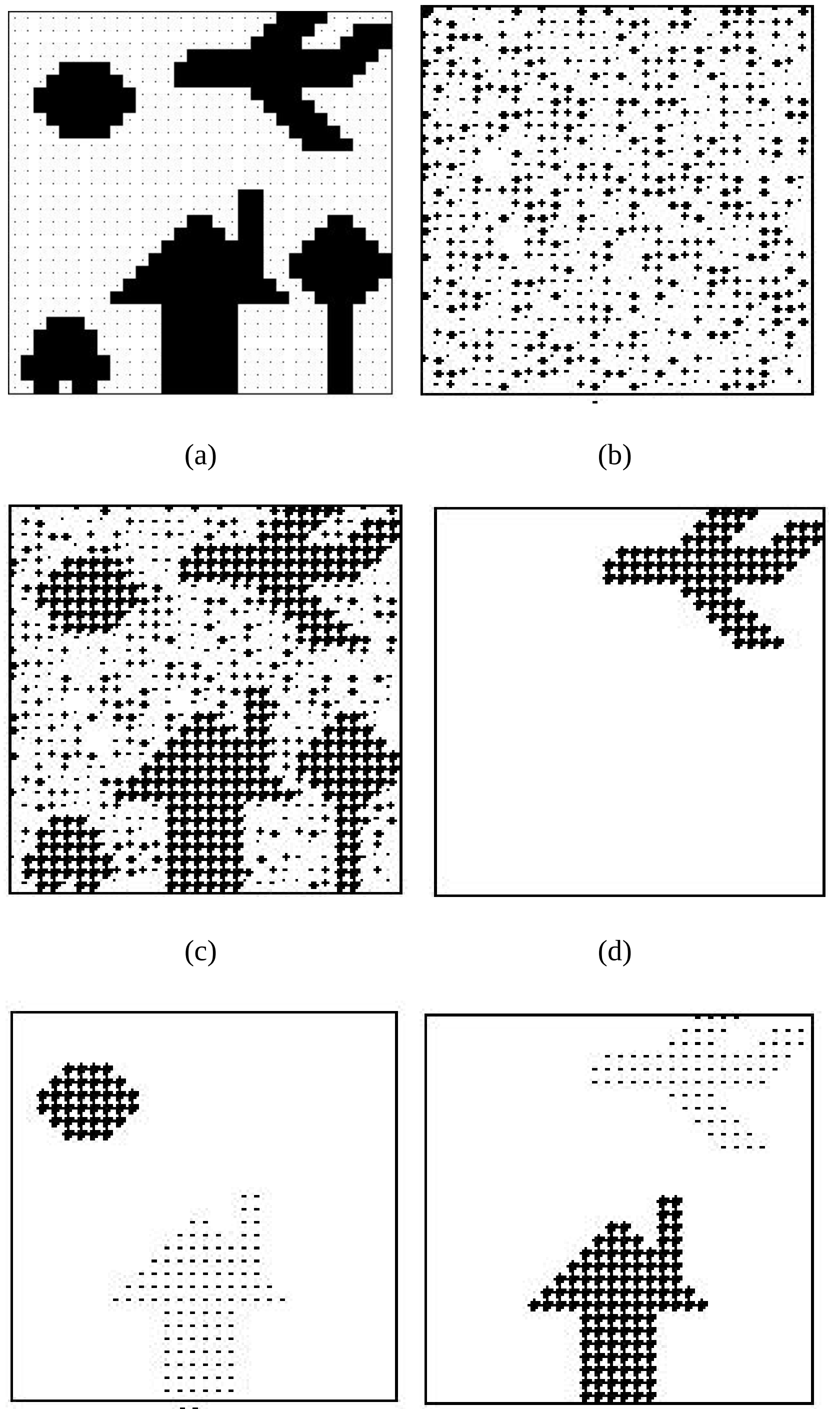
To study the oscillatory dynamics of SEON, a 30 by 30 2D SEON network is applied to black and white image segmentation similar to the one performed by Campbell et al. (Campbell et. al., 1999). The parameter values used in the simulation are listed in Table 2 (These values will be used for all the experiments in this thesis unless otherwise stated). Figure 15(a) shows the input image. The black and white squares correspond to those oscillators that receive stimulus (S_i=1) and do not receive stimulus (S_i=0) respectively. The diameter of the black circle is directly proportional to the value of the corresponding oscillator x_i and is calculated as follows:

$$d_i = k \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{10}$$

where k, x_{min} and x_{max} are the scaling parameter, the minimum value and maximum value of all oscillators, respectively. Figures 15(b) – (h) shows the network activities at some specific time steps. In Figure 15(b), initial values x_i of the oscillators are randomly chosen from the range [0,1]. Figure 15(c) shows the network activity after the input image is inputted to the network. Figures 15(d) – (h) show the segments that resemble the aeroplane, sun, house, car and tree at their highest activation respectively. Each segment appears clearly once every cycle. The segments will appear periodically in the same sequence.

Model Parameters	Values Used in the Simulation
γ	0.5
ε	1
θ_{ES}	0.8
θ_i	1

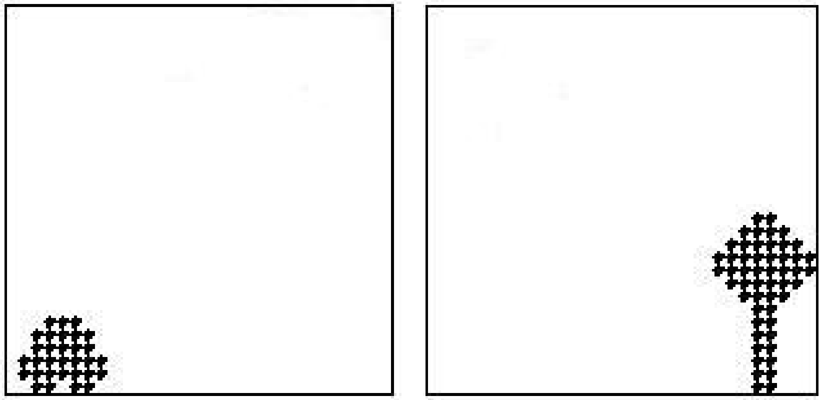
Table 2. Parameter values used in black and white image segmentation experiments.



(a) (b)

(c) (d)

(e) (f)

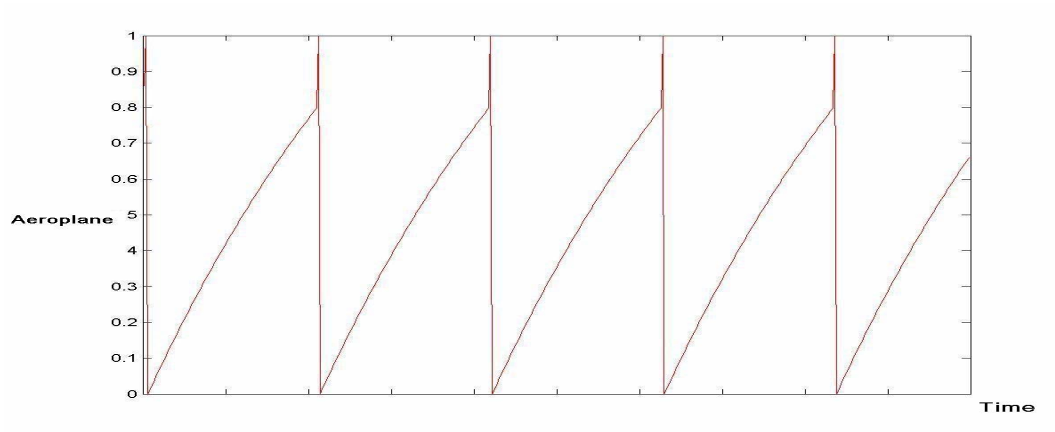


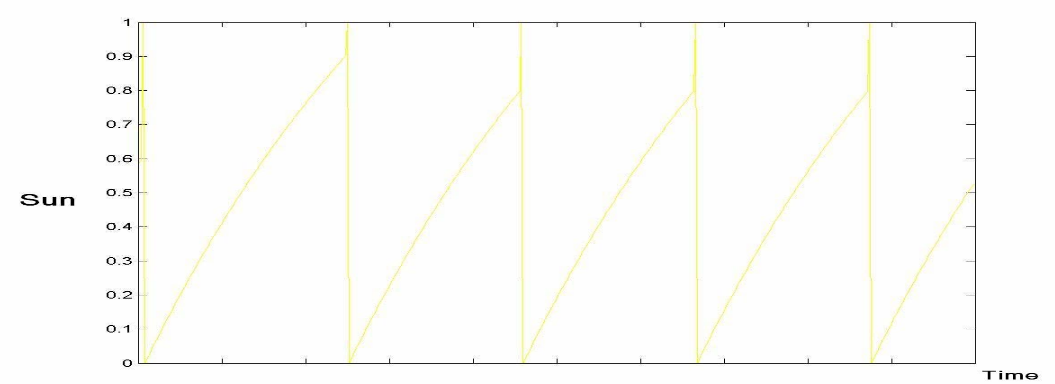
(g) (h)

Figure 15. Black and white image segmentation.

(a) Input image. (b) The network is initially randomized from the range [0,1]. (c) The image in (a) is inputted to the network. (d) – (h). Each picture shows the maximal activities for each group of oscillators at some specific time steps. Other parameters are shown in Table 2.

Figure 16 shows the temporal evolution of the combined activities of oscillators corresponding to the five segments for the first few cycles. In Figure 16, at the beginning of the simulation, the undesirable global synchronization occurs between segments corresponding to sun, car, aeroplane and tree. However, when the group of oscillators corresponding to the aeroplane segment first reaches the threshold of ES (θ_{ES}=0.8), ES sends the first excitation to one of the oscillators in the group, and desynchronizes them from the other three groups of oscillators. After a $T_{separation}$ time, ES is set to the ready state again. It sends the second excitation and desynchronizes the sun segment from other two segments. Both the desynchronization of the aeroplane and sun segments from other segments happens rapidly in the first cycle. Undesirable accidental resynchronization happens when the house segment receives the third excitation from ES in its second cycle. The excitation from ES causes the house segment to accidentally synchronize with the car and tree segments. Although the possibility of accidental resynchronization is not very high (many trials of simulations have been performed in order to achieve this), the desynchronization algorithm cannot prevent this problem. However, in the next cycle, ES sends the six and seventh excitation to the house and car segments respectively. This rapidly desynchronizes the synchronized house, car and tree segments again. The problem of accidental resynchronization can thus be also solved by ES. It should be noted that the fourth and fifth excitations sent to the aeroplane and sun segment respectively, do not create any synchronization effect. The clearly desynchronized five segments will then keep the same sequence, and appear periodically in the sequence order as shown in Figure 16.





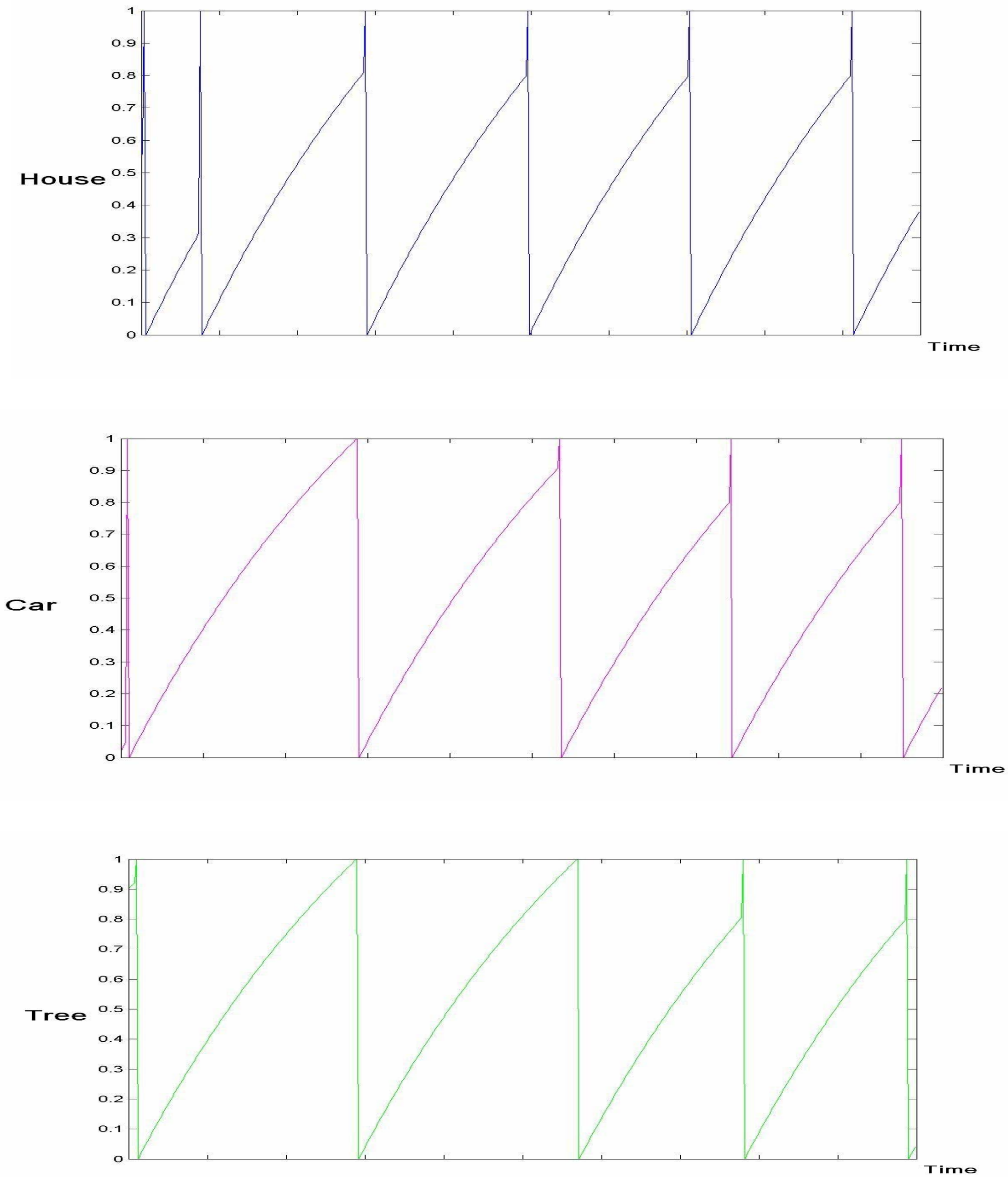


Figure 16. The temporal evolutions of five segments during image segmentation process.

Each diagram represents the combined x_i activities of all of the oscillators of the corresponding segment.

The parameter values are shown in Table 2.

5.3. Grayscale Scene Segmentation

The performance analysis of SEON in grayscale image segmentation was done in two ways. Firstly, it was done through the comparison on segmentation capability between SEON and other existing methods. These methods include the traditional sequential method such as watershed segmentation (Gonzalez, 1992; Banks, 1990; Bieniek & Moga, 2000) and the recent parallel methods such as LEGION (Wang & Terman, 1997; Campbell et. al., 1999) and parallel watershed transformation (Alina & Moncef, 1997; Alina et. al., 1998). Secondly, the segmentation speeds between these methods are compared. The relationship between the speed and image size is also investigated.

5.3.1. Segmentation Performance Test

The experiments use a scene gallery of 1200 grayscale images from GreenStreet™ from GST Technology Ltd. as scenery images. For systematic validation and analysis of segmentation performance for different level of complexity of the nature scenes, the 1200 images are divided into six categories in the ascending order of complexity level:

(1) Objects with clear boundaries in a homogenous background

(2) Objects with vague boundaries in a homogenous background

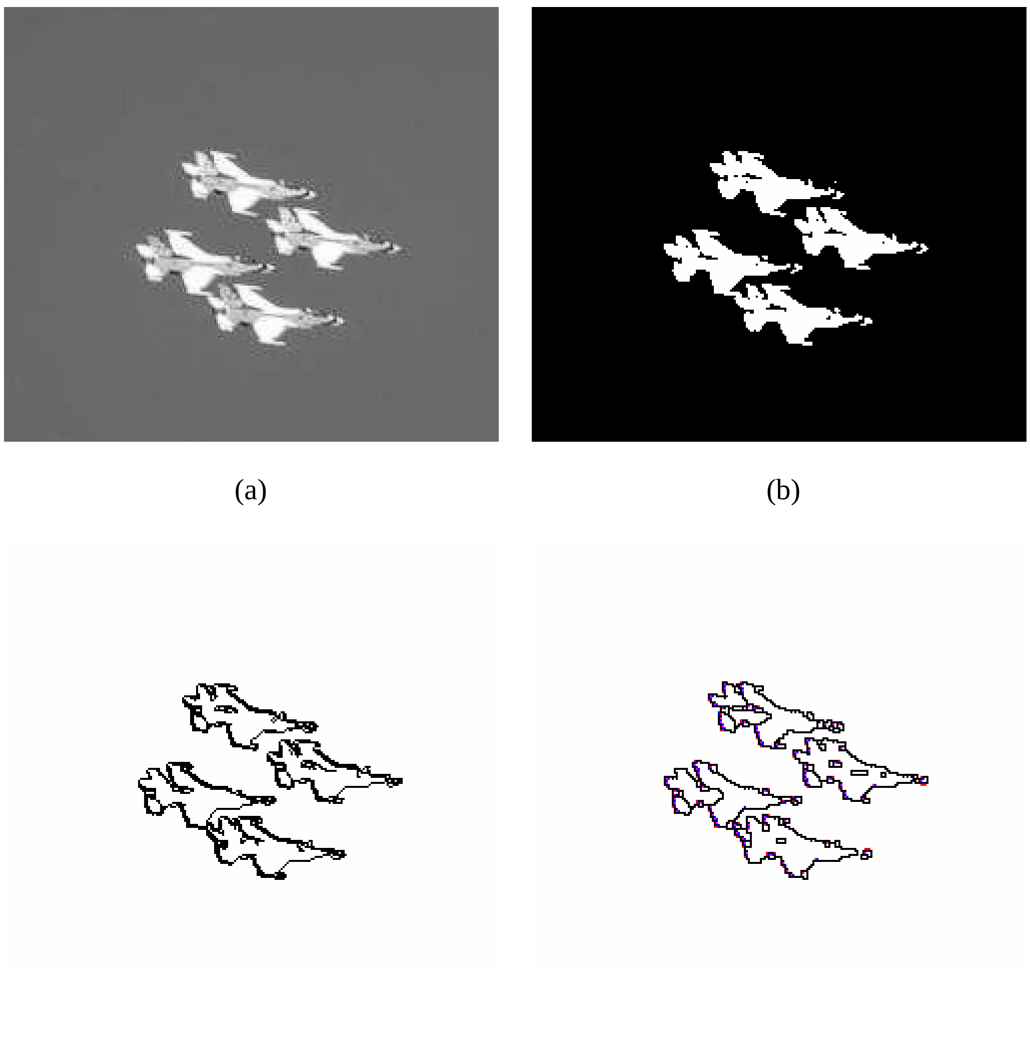
(3) Objects with complicated boundaries in a homogenous background

(4) Objects with clear boundaries in a nonhomogenous background

(5) Objects with vague boundaries in a nonhomogenous background

(6) Objects with complicated boundaries in a nonhomogenous background

Figures 17-21 show the sample images from the first five categories and the corresponding segmented images with SEON, sequential watershed method and LEGION. The segmented images with parallel watershed transformation are not shown because the method produces same segmentation result as the sequential version. The major difference between these two methods is the segmentation speed. In the segmented images of SEON, each region represents a synchronized group of oscillators. Inactive oscillators comprising the boundaries are black.



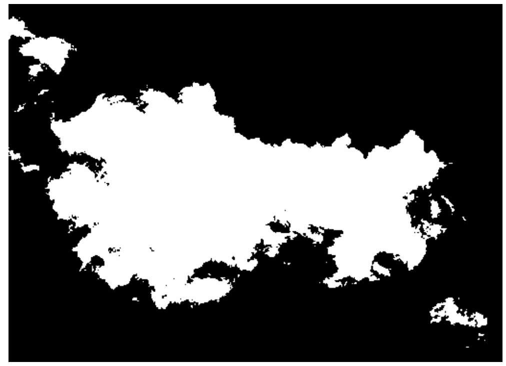
(a) (b)

(c) (d)

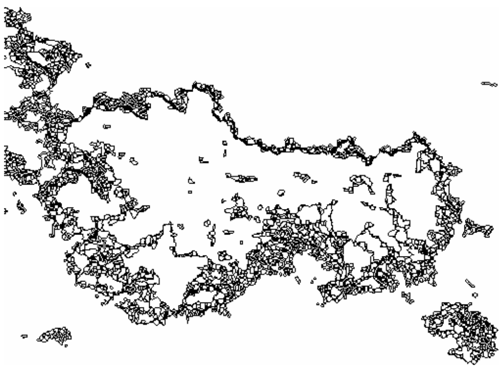
Figure 17. Sample grayscale image of category (1) and the corresponding segmented images.

(a) Original image (b) Segmented by LEGION (c) Segmented by SEON (d) Segmented by Watershed.

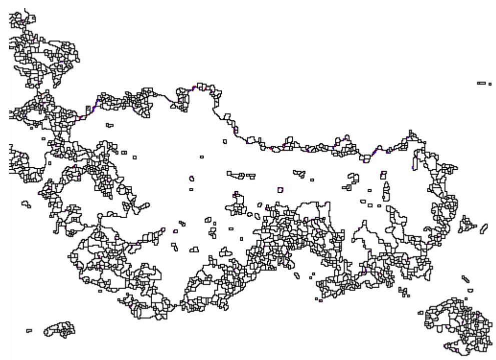
(a)



(b)



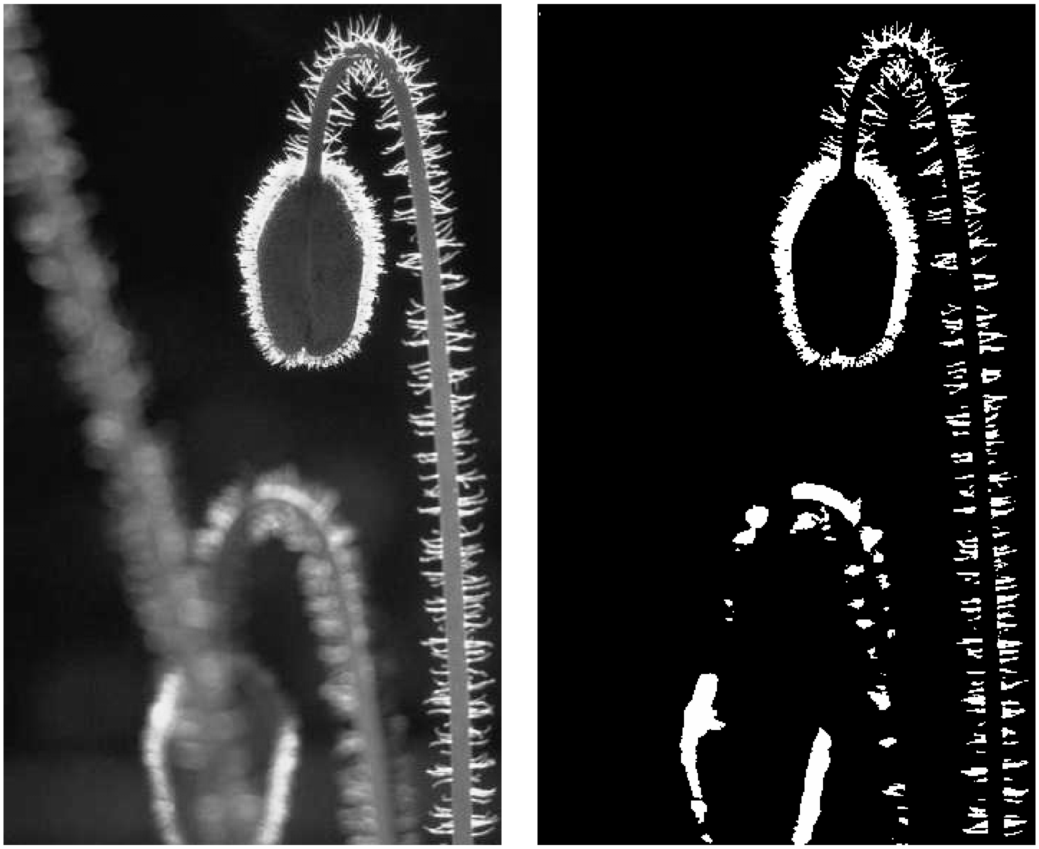
(c)



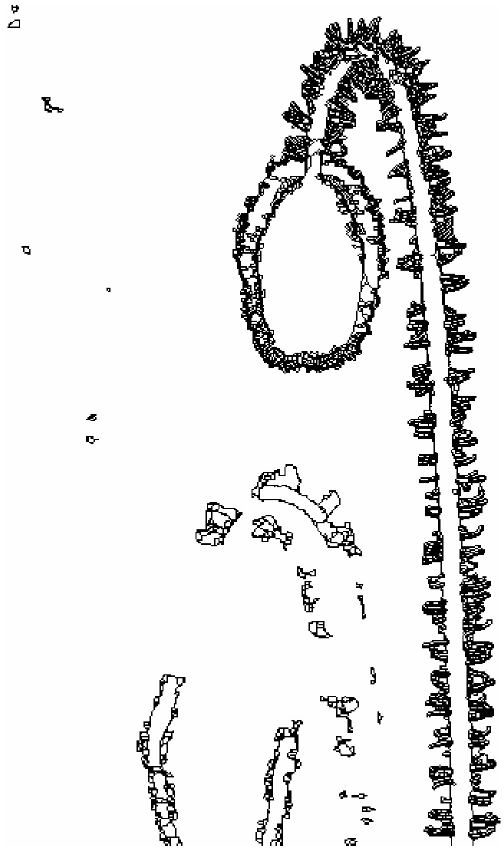
(d)

Figure 18. Sample grayscale image of category (2) and the corresponding segmented images.

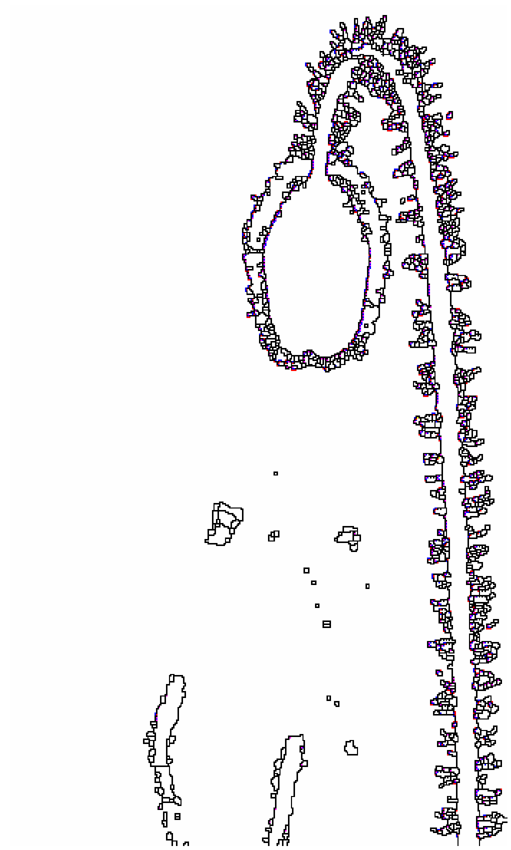
(a) Original image (b) Segmented by LEGION (c) Segmented by SEON (d) Segmented by Watershed.



(a) (b)



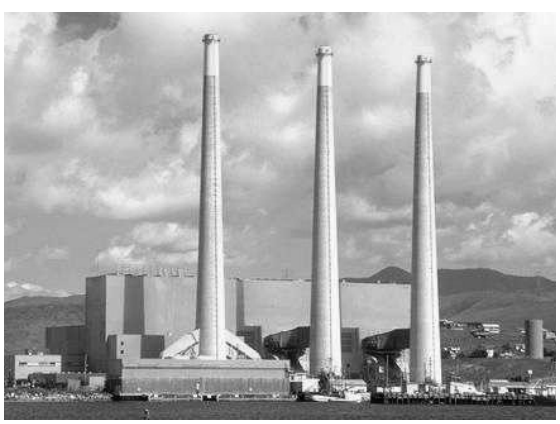
(c)



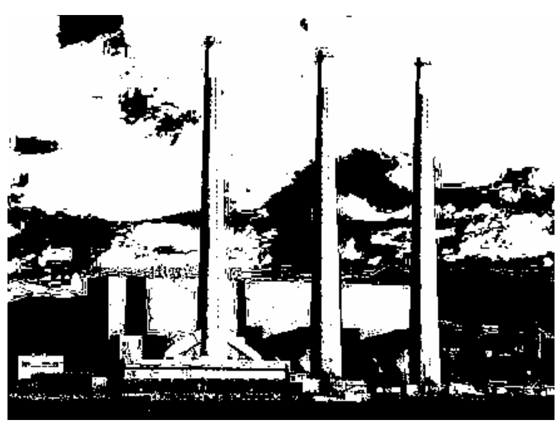
(d)

Figure 19. Sample grayscale image of category (3) and the corresponding segmented images.

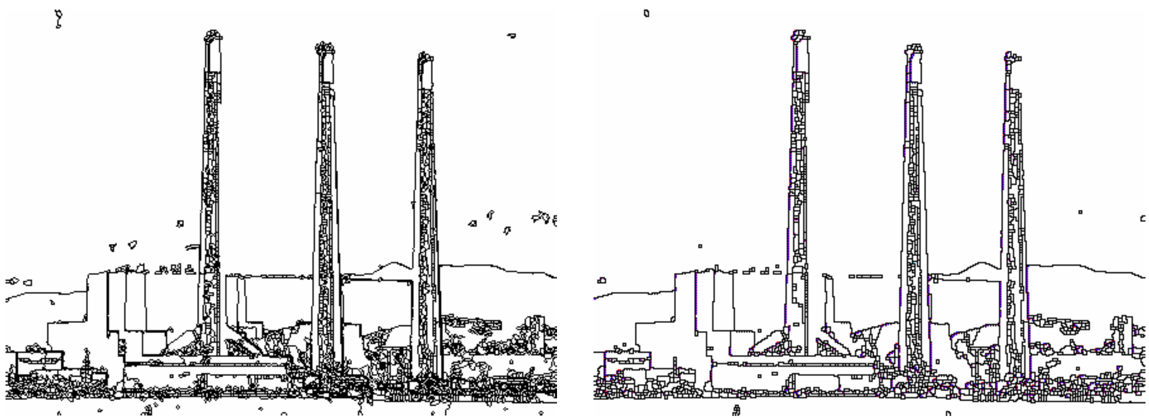
(a) Original image (b) Segmented by LEGION (c) Segmented by SEON (d) Segmented by Watershed.



(a)



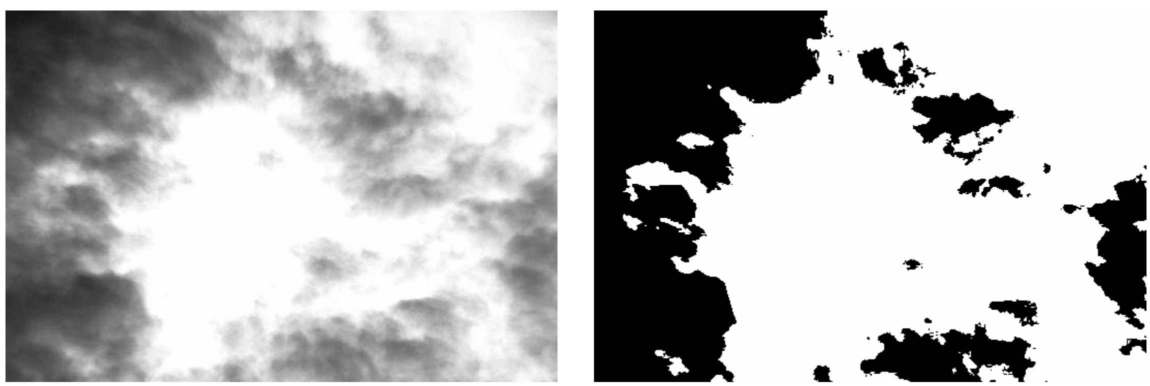
(b)



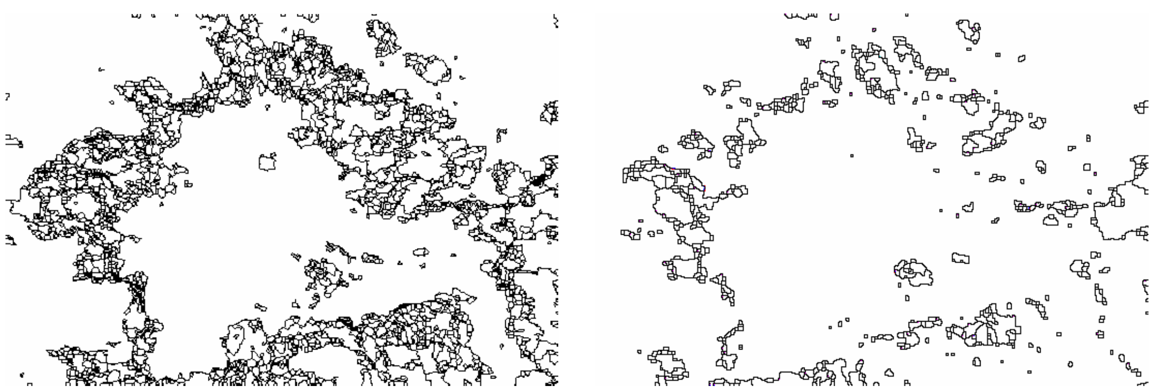
(c) (d)

Figure 20. Sample grayscale image of category (4) and the corresponding segmented images.

(a) Original image (b) Segmented by LEGION (c) Segmented by SEON (d) Segmented by Watershed.



(a) (b)



(c) (d)

Figure 21. Sample grayscale image of category (5) and the corresponding segmented images.

(a) Original image (b) Segmented by LEGION (c) Segmented by SEON (d) Segmented by Watershed.

SEON and watershed segmentation produce more reliable results than LEGION. The advantage of reliable segmentation is more significant for complicated objects (e.g. plants) and nonhomogenous background in categories (3), (4) and (5). In general, LEGION cannot efficiently segment complicated objects such as plants. LEGION also cannot segment images effectively if the background is nonhomgenous. It only produces reliable segmentation for simple objects in homogenous background such as white aeroplanes and clouds in clear sky. In addition, both SEON and watershed segmentation produce continuous boundaries, rather than discontinuous ones in LEGION. This significantly reduces the needs of edge-linking procedure.

SEON detects vague boundaries more efficiently than watershed segmentation. This is demonstrated by the images in categories (3), (4) and (5). In general, SEON can detect vague boundaries that cannot be detected by watershed segmentation. For example, SEON can detect the vague boundaries of clouds in Figure 21, whereas watershed segmentation cannot produce a complete boundary of the clouds.

5.3.2. Segmentation Speed Test

Four methods including SEON, LEGION, parallel watershed transformation and sequential watershed transformation have been implemented on an Intel® Pentium® 4 2000MHz machine with 1024MB RAM for the comparison of segmentation speed. Sequential watershed segmentation (Gonzalez, 1992; Banks, 1990; Bieniek & Moga, 2000) is performed by Image Processing Toolbox 3.0 in MATLAB 6.0 from The MathWorks, Inc. SEON, LEGION (Wang & Terman, 1997; Campbell et. al., 1999) and parallel watershed transformation (Alina & Moncef, 1997; Alina et. al., 1998) are implemented by Java™ programming language in a multithreading manner. The segmentation speeds of the three parallel methods are studied. The segmentation speed of sequential watershed method is only shown for reference.

In SEON, each pixel in the image is represented by one oscillator. Segmentation of real images will involve a large number of oscillators and require integrating a large number of differential equations. To reduce the numerical computation, the segmentation method is extracted from the oscillatory dynamics of the network while preserving the essential properties of Peskin oscillators such as synchronization and desynchronization dynamics.

Table 3 shows the average execution times that the four different methods segment various image sizes of X by X pixels. Although sequential watershed method is the fastest method for 128 by 128 images, due to the sequential nature, its execution time rises significantly when the image size increases. On the other hand, the three parallel methods do not increase as much as the sequential method for large image size.

On average, SEON is the fastest of the four methods when the image size is larger than 256 by 256 pixels. If the image is smaller than 256 by 256 pixels, due to the 256 sequential gray-level steps, SEON is lower than LEGION. However, SEON is still faster than parallel watershed transformation for 128 by 128 images.

The efficiency of SEON also increases when image size increases. While SEON segments small images (256 by 256 pixels) only 128% faster than LEGION, it segments large images (4096 by 4096 pixels) 448% faster than LEGION. In addition, while SEON segments small images (256 by 256 pixels) only 155% faster than parallel watershed transformation, it segments large images (4096 by 4096 pixels) 808% faster than parallel watershed transformation. This shows that the improvement of SEON for large images is greater than LEGION and parallel watershed transformation.

Image Size X	Average segmentation execution times (in seconds)

(X by X Pixels)	SEON	LEGION	Par. Watershed	Seq. Watershed
128	0.183	0.179	0.19	0.167
256	0.352	0.453	0.547	0.793
512	0.697	1.34	1.618	2.673
1024	1.389	3.435	4.729	11.563
2048	2.746	9.064	14.694	38.792
4096	5.397	24.167	43.619	119.376

Table 3. Average segmentation execution times of SEON and other methods

The average execution time of each image size X is obtained from 200 images.

5.4. Color Scene Segmentation

The performance analysis of SEON in color scene segmentation was also done in two ways. Firstly, it was done through the comparison on segmentation capability between SEON and other existing methods. These methods include gray-level sequential watershed segmentation (Gonzalez, 1992; Banks, 1990; Bieniek & Moga, 2000), gray-level parallel watershed transformation (Alina & Moncef, 1997; Alina et. al., 1998) and color sequential RGB vector gradient method (Gonzalez, 1992). Secondly, the segmentation speeds between these methods were compared.

5.4.1 Segmentation Performance Test

The experiments use a scene gallery of 600 color images from GreenStreet™ from GST Technology Ltd. as scenery images. For systematic validation and analysis of segmentation performance for different level of complexity of the nature scenes, the 600 images are divided into six categories in the ascending order of complexity level:

(1) Objects with clear boundaries in a homogenous background

(2) Objects with vague boundaries in a homogenous background

(3) Objects with complicated boundaries in a homogenous background

(4) Objects with clear boundaries in a nonhomogenous background

(5) Objects with vague boundaries in a nonhomogenous background

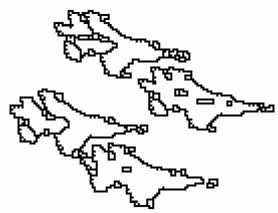
(6) Objects with complicated boundaries in a nonhomogenous background

Figures 22-26 show the sample images from the first five categories and the corresponding segmented images with SEON (with the additional boundaries detected by NBS color difference of equation (4)), color RGB vector gradient method and gray-level sequential watershed method. The segmented images with parallel watershed transformation are not shown because the method produces same segmentation result as the sequential version. The major difference between these two methods is the segmentation speed. In the segmented images of SEON, each region represents a synchronized group of oscillators. Inactive oscillators comprising the boundaries are black.

(a) (b)

(c)

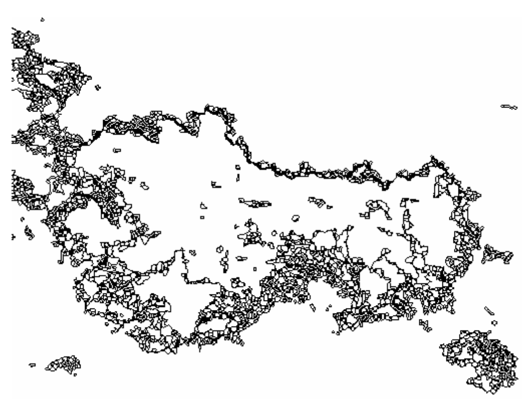
(d)



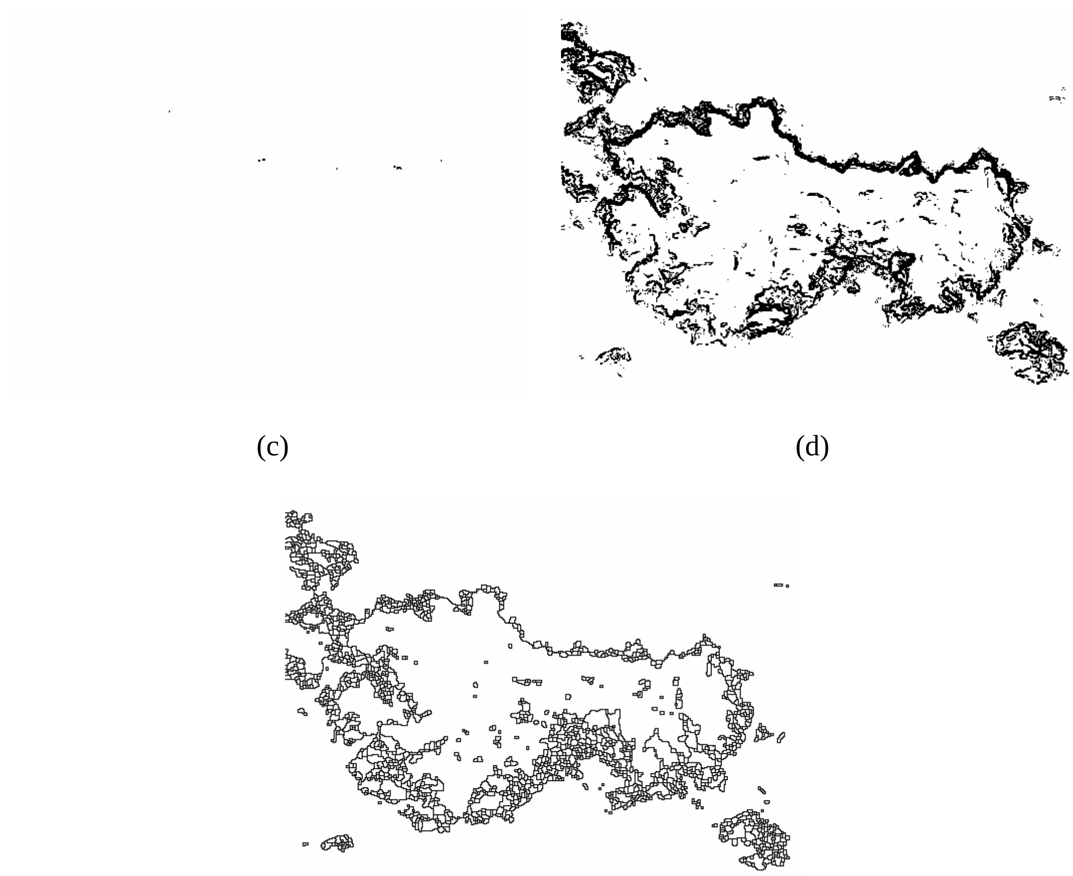
(e)

Figure 22. Sample color image of category (1) and the corresponding segmented images.

(a) Original image (b) Segmented by SEON (c) Additional boundaries detected by NBS value (d) Segmented by RGB gradient. (e) Segmented by Watershed.



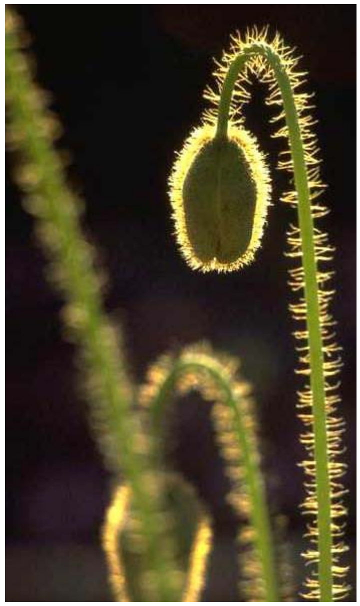
(a) (b)

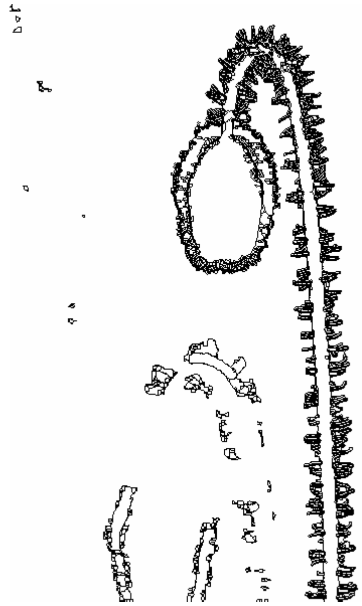


(e)

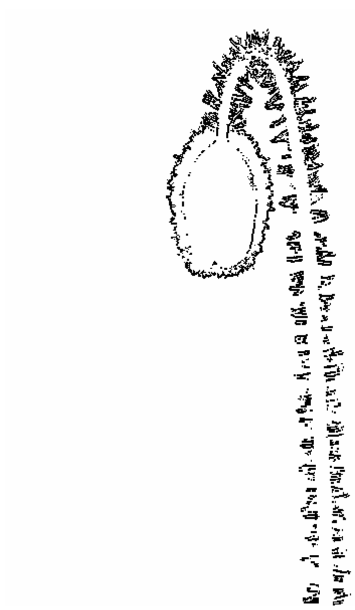
Figure 23. Sample color image of category (2) and the corresponding segmented images.

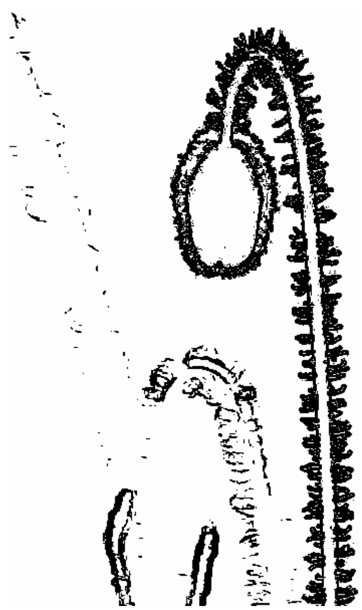
(a) Original image (b) Segmented by SEON (c) Additional boundaries detected by NBS value (d) Segmented by RGB gradient. (e) Segmented by Watershed.



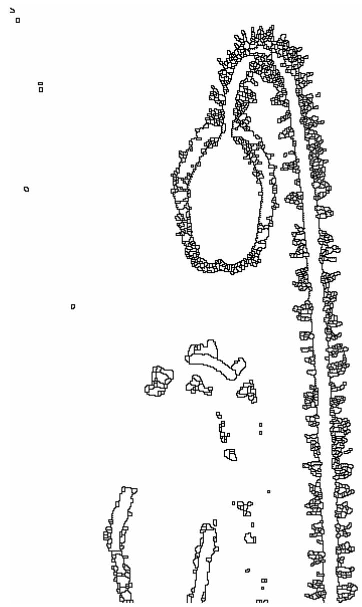


(a) (b)





(c) (d)

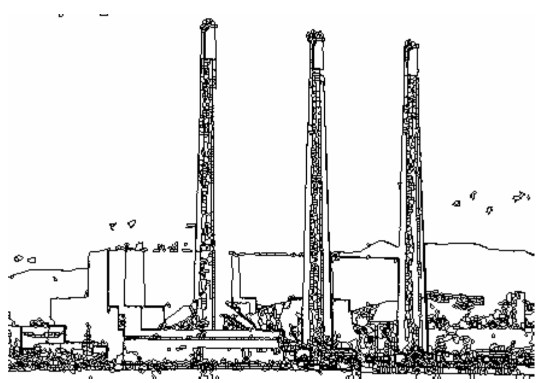


(e)

Figure 24. Sample color image of category (3) and the corresponding segmented images.

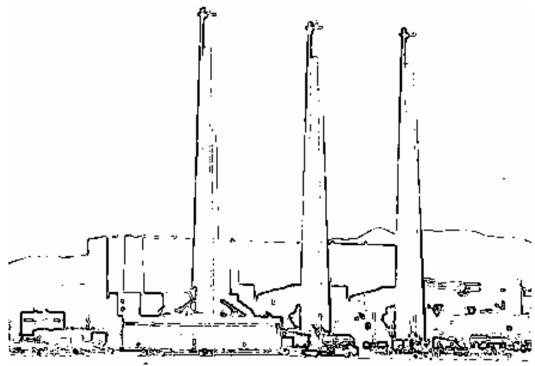
(a) Original image (b) Segmented by SEON (c) Additional boundaries detected by NBS value (d) Segmented by RGB gradient. (e) Segmented by Watershed.

(a)

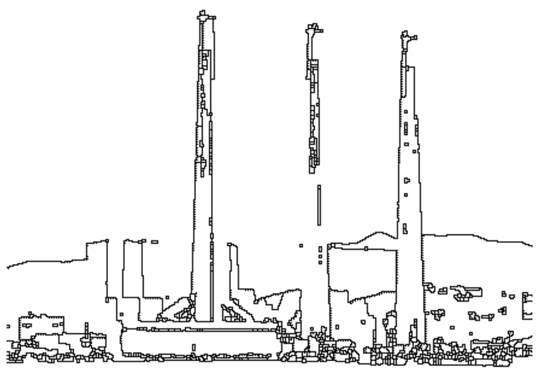


(b)

(c)



(d)

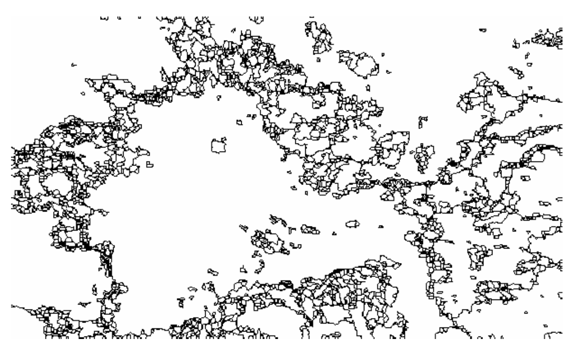


(e)

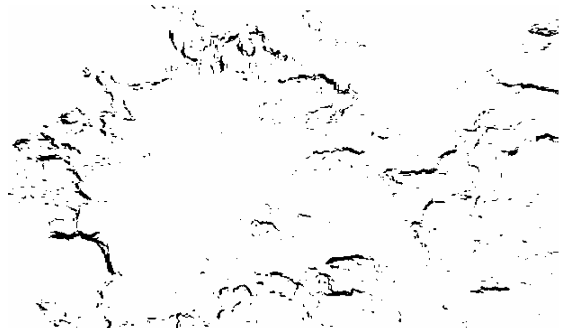
Figure 25. Sample color image of category (4) and the corresponding segmented images.

(a) Original image (b) Segmented by SEON (c) Additional boundaries detected by NBS value (d) Segmented by RGB gradient. (e) Segmented by Watershed.

(a)



(b)



(c) (d)

(e)

Figure 26. Sample color image of category (5) and the corresponding segmented images.

(a) Original image (b) Segmented by SEON (c) Additional boundaries detected by NBS value (d) Segmented by RGB gradient. (e) Segmented by Watershed.

SEON produces more reliable results than color RGB vector gradient method and watershed segmentation. The advantage of reliable results is more significant for complicated objects and nonhomogenous background in categories (4) and (5). In the sample images of the three categories, only SEON can produce the complete contour of the objects. For example, SEON can detect the complete contour of the factory in Figure 25, but RGB vector gradient method and watershed segmentation can only produce part of the contours.

SEON also detects vague boundaries more effectively than RGB vector gradient method and watershed segmentation. This is demonstrated by the images in categories (5). SEON can detect the vague boundaries of clouds in Figure 26, whereas RGB vector gradient method and watershed segmentation cannot produce a complete boundary of the clouds. In addition, both SEON and watershed segmentation produces continuous boundaries, rather than discontinuous ones in RGB vector gradient method. This significantly reduces the need of edge-linking procedure.

Usually, in images that contained many different colors such as Figures 22(a), 24(a) and 25(a), the NBS color difference provides useful additional boundaries as shown in Figures 22(c), 24(c) and 25(c). However, in images consisted of only few colors such as Figures 23(a) and 26(a), the NBS color difference provides only few additional boundaries. Usually SEON has already detected the boundaries in the gray level version of the images. Therefore, even though there are some different colors in Figures 23(a) and 26(a), the NBS color difference does not necessarily provide additional useful information.

5.4.2. Segmentation Speed Test

Four methods including SEON, sequential watershed segmentation, parallel watershed transformation and sequential RGB vector gradient method have been implemented on an Intel® Pentium® 4 2000MHz machine with 1024MB RAM for the comparison of segmentation speed. Sequential watershed segmentation (Gonzalez, 1992; Banks, 1990; Bieniek & Moga, 2000) and RGB vector gradient method (Gonzalez, 1992) are performed by Image Processing Toolbox 3.0 and Digital Image Processing Using MATLAB (DIPUM) Toolbox (Gonzalez et. al., 2004) in MATLAB 6.5 from The MathWorks, Inc, respectively. SEON and parallel watershed transformation (Alina & Moncef, 1997; Alina et. al., 1998) are implemented by Java™ programming language in a multithreading manner. The segmentation speeds of the two parallel methods are the focus of the study. The segmentation speeds of the sequential methods are only shown for reference.

In SEON, each pixel in the image is represented by one oscillator. Segmentation of real images will involve a large number of oscillators and require integrating a large number of differential equations. To reduce the numerical computation, the segmentation method is extracted from the oscillatory dynamics of the network while preserving the essential properties of Peskin oscillators such as synchronization and desynchronization dynamics.

Table 4 shows the average execution times that the four different methods segment various image sizes of X by X pixels. Although sequential RGB vector gradient method is the fastest method for 128 by 128 images, due to the sequential nature, its execution time rises significantly when the image size increases. The sequential watershed method has also the same problem. On the other hand, the two parallel methods do not increase as much as the sequential method for large image size.

On average, SEON is the fastest of the four methods when the image size is larger than 256 by 256 pixels. SEON also segments large images more efficiently than parallel watershed transformation. For example, while SEON segments small images (256 by 256 pixels) only 155% faster than parallel watershed transformation, it segments large images (4096 by 4096 pixels) 808% faster than parallel watershed transformation. This shows that the improvement of SEON for large images is greater than parallel watershed transformation.

Image Size X (X by X Pixels)	Average segmentation execution times (in seconds)			
	SEON	Par. Watershed	RGB gradient	Seq. Watershed
128	0.183	0.19	0.092	0.167
256	0.352	0.547	0.561	0.793
512	0.697	1.618	2.038	2.673
1024	1.389	4.729	9.278	11.563
2048	2.746	14.694	32.542	38.792
4096	5.397	43.619	108.375	119.376

Table 4. Average segmentation execution times of color SEON and other methods.

The average execution time of each image size X is obtained from 100 images.

Chapter 6. Conclusions and Further Research

6.1. Organization of the Chapter

This chapter concludes the thesis and is organized in four sections. Section 6.2. summarizes the major contributions of the thesis. Section 6.3. describes the psychophysical plausibility of SEON. Section 6.4. points out some limitations of the proposed SEON model. Finally, section 6.5. suggests possible further research of SEON. Some recent results of the suggested further research are also briefly described.

6.2. Summary of Major Contributions

A SEON network is proposed for cluttered scene segmentation and the study of synchronization mechanism in human perception. The major contributions of this thesis can be summarized in three areas namely, segmentation performance, segmentation speed and psychological plausibility.

SEON improves the segmentation performance of traditional oscillator networks. The proposed model has been successfully applied to segment complicated objects (e.g. human face, plants) in cluttered scenes, whereas traditional oscillator networks are usually only applied to segment binary images or simple grayscale images. The segmented images of SEON have continuous boundary, obviating the need for expensive edge linking procedure. This advantage does not exist in most neural network approaches to scene segmentation.

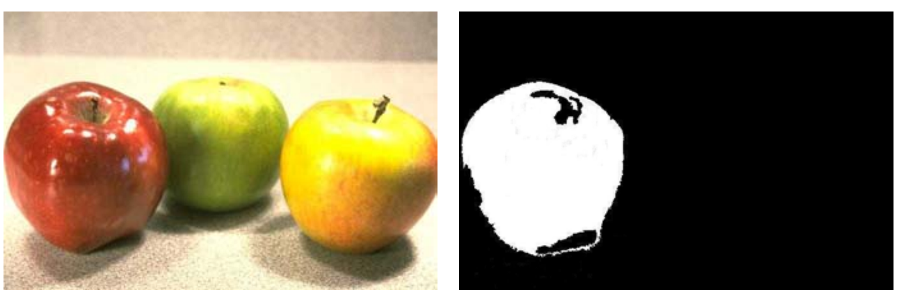
SEON further improves the segmentation speed of traditional oscillator networks. SEON further improves the synchronization speed to one cycle in a very large network. The synchronization speed will not decrease when the network size increases. This allows SEON to segment large scenes very efficiently.

SEON may provide a more extensive framework for the study of perceptual organization and selective attention than traditional oscillator networks. In particular, SEON can be viewed as the stage of perceptual organization, whereas the use of markers in SEON can be viewed as the stage of selective attention. In addition, the potential use of marker allows SEON to use a richer set of object properties as selection criteria in selective attention. The detail is given in the next section.

6.3. A Framework for Selective Attention

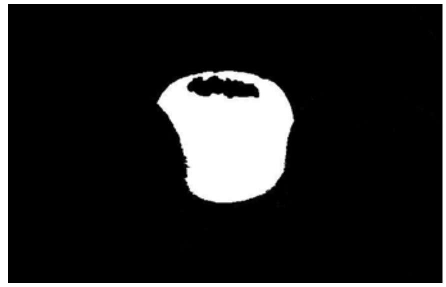
SEON may provide an extensive framework for the study of perceptual organization and selective attention (Mack et. al., 1992). While SEON can be viewed as the stage of perceptual organization (Mack et. al., 1992), the use of marker (Bleau & Leon, 2000; Banks, 1990; Gonzalez, 1992; Pal & Pal, 1993) in SEON can be viewed as the stage of selective attention (Mack et. al., 1992).

SEON can be viewed as the stage of perceptual organization or preattentive processing (Mack et. al., 1992). In SEON, multiple segments will pop out periodically in sequence. These segments can be interpreted as the multiple organizations generated in perceptual organization (Mack et. al., 1992; Trick & Pylyshyn, 1994). Figure 27 shows the three largest organizations (segments) that SEON produces when segmenting Figure 27(a).



(a) (b)

(c)



(d)

Figure 27. The three largest organizations (segments) generated by SEON.

The three organizations (segments) are selected manually among many other small organizations (segments) based on the size of organizations. These three organizations (segments) pop out periodically in sequence.

The organizations generated in perceptual organization will be processed sequentially in selective attention. This is consistent with the studies that selection attention is based on segment-by-segment scrutiny and serial shifting (Julesz, 1995; Neumann & Sanders, 1996). SEON is also consistent with the studies that the processing of perceptual organization is parallel in nature (Mack et. al., 1992; Trick & Pylyshyn, 1994).

By using markers (Bleau & Leon, 2000; Banks, 1990; Gonzalez, 1992; Pal & Pal, 1993), SEON can select objects based on a rich set of objects properties including size, color, motion, texture and familiarity. The use of markers in SEON can therefore be viewed as a stage of selective attention. In Figure 27, the three largest organizations can be selected among other small segments based on their sizes. A selection network based on object size has been proposed (Wang, 1996). However, size is only one of the object properties. A more comprehensive model of selective attention should select objects based on various properties of objects. SEON inherits the advantage of watershed segmentation (Gonzalez, 1992) that can be easily extended to incorporate markers to bring prior knowledge to the segmentation process. Using different kinds of markers (Bleau & Leon, 2000; Banks, 1990; Gonzalez, 1992; Pal & Pal, 1993) allows SEON to select objects based on a rich set of properties. Many markers have been proposed in the literatures

(Bleau & Leon, 2000; Banks, 1990; Gonzalez, 1992; Pal & Pal, 1993). This allows SEON to use various object properties for selective attention. While an in-depth research of using markers in SEON is left to future works, some preliminary results of marker-controlled SEON will be discussed in section 6.5.

In addition, SEON is consistent with Gestalt psychology (Koffka, 1935; Rock & Palmer, 1990). Firstly, SEON does not require any training before segmentation. This is consistent with the fact that the first five grouping principles in Gestalt psychology (proximity, similarity, common fate, connectedness and good continuation) do not require any learning. In contrast, the traditional neural networks for image segmentation often require a long training time (Kosko, 1992; Pal & Pal, 1993). Secondly, the use of markers to bring prior knowledge allows SEON to incorporate the last grouping principle (prior knowledge) of Gestalt psychology (Koffka, 1935; Rock & Palmer, 1990).

The fast segmentation speed of SEON is not only important for fast scene segmentation, but also for the neural plausibility. It is known that a human can identify an object within 100ms (Biederman, 1987). Hence, the fast segmentation speed of SEON is more consistent with the human visual system. It should be noted that the author does not mean to claim that the existing SEON system has the same performance as the human vision system. Instead, the proposed method, to the best of the author's knowledge, is one of the fastest segmentation methods in the literatures (Pal & Pal, 1993,Gonzalez, 1992; Banks, 1990), which is relatively closed to the fast segmentation performance of human vision. The strong segmentation capability in cluttered scenes also makes SEON closer to the segmentation ability of human vision model.

6.4. Limitations

While the synchronization speed of SEON is independent of the network size, the desynchronization mechanism of SEON is its performance bottleneck. Since ES can only desynchronize one segment among all synchronized segments at every firing, the performance of the network decreases when the number of globally synchronized segments increases. Thus, although SEON can theoretically separate an infinite number of segments, the performance of SEON is likely to decrease when the number of segments increases because there may be a greater number of globally synchronized segments. We can reduce $T_{separation}$ to improve the performance as long as the network can still clearly separate different segments.

The parallel segmentation method inherits the problem of oversegmentation from the sequential watershed segmentation. While the four neighbors mechanism enables SEON to detect vague boundaries more effectively, the mechanism also makes SEON more vulnerable to oversegmentation. This not only seriously affects the segmentation results, but also reduces the desynchronisation speed. Many useless small segments will be created because of oversegmentation, which affects the desynchronization speed.

6.5. Further Research and Recent Research Results

Further enhancement of the model is now under investigation in three ways. Firstly, I am developing an efficient extension of the parallel network that incorporates markers. An efficient marker mechanism should not only solve the oversegmentation problem, but should also allow SEON to bring a prior knowledge to bear on the segmentation problem. Secondly, I am applying multi-resolution scene analysis to SEON, which allows course-to-fine image segmentation. Thirdly, SEON is particularly suitable for very-large-scale-integrated (VLSI) technology (for an actual chip implementation exploring phase locking see Andreou & Edwards, 1994 and Madrenas et. al., 2004). The implementation of SEON using VSLI technology is another possible research direction. In this section, I will conclude this thesis by presenting some recent results of using markers in SEON.

As mentioned before, SEON can be extended to incorporate markers to bring prior knowledge to the model for scene segmentation. There are several advantages for using markers. Firstly, markers provide a mechanism to select objects in a scene based on a rich set of object properties in selective attention. Secondly, markers solve the problem of oversegmentation. Thirdly, markers indirectly improve the segmentation speed by avoiding oversegmentation.

When segmenting a scene, SEON can use the useful information provided by markers to simplify the segmentation tasks. By using markers, only those regions that have markers can be created. Therefore, when an oscillator has neighbors belonging to more than one region, SEON no longer needs to determine whether two regions should be merged or not. Instead, the two regions must be different regions because irrelevant region will not be created.

A variation of the segmentation method described in section 4.5. could be used for SEON to use markers during segmentation. In the following segmentation algorithm, p_i is the gray level value of the *i*th pixel and is in the range [0,255]. $T(i)$ represents the four nearest neighbours of oscillator *i*.

---

1. The stimulus values S_i of all oscillators are set to 1.
2. Repeat until all oscillators are marked to a region

    For gray level value *k* from 0 to 255

      For any oscillator *i* with gray level value *k*

        If the oscillator *i* has not been marked

          For $j \in T(i)$

            If no neighbor with $p_j \leq k$ belongs to any region

              If the oscillator has a marker

                Create new regional minimum by marking the oscillator state

                to an unique number M_r.

              Else

Mark the oscillator as "boundary region" by setting S_j to 0.

Else if the neighbors with $p_j \leq k$ belongs to only one region and no neighbor is $p_j < k$

Mark the oscillator to belong to that region. Create connections between oscillator *i* and *j*. The connection is described by equation (7).

Else if the oscillator is marked as "boundary region"

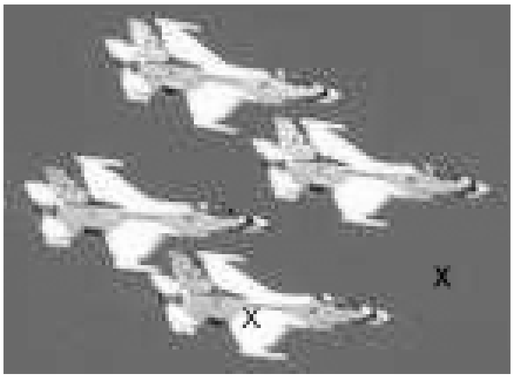
If the neighbors with $p_j \leq k$ belongs to only one region and neighbor is $p_j < k$

Mark the oscillator to belong to that region. Create connections between oscillator *i* and *j*. The connection is described by equation (7).

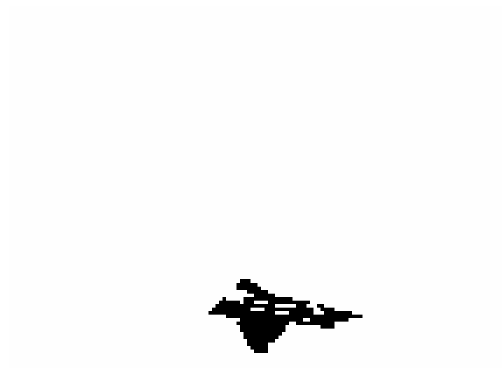
The major difference between the above method and the segmentation method in section 4.5. is that SEON does not need to determine whether two regions should be merged or not. A new regional minimum can only be created if the oscillator has a marker. Other irrelevant regions will not be created because they do not have markers. Therefore, all regions created by SEON are of interests and should not be merged.

Figure 28(a) shows an image that has four planes. As indicated by the cross, the plane at the bottom of the image has a marker. The background has another marker. The other three planes do not have any marker. Figures 28(b) and 28(c) show the segmented objects at their highest activities. Only two objects are obtained, namely the plane at the bottom and the background. Other three planes are considered as parts of the background.

It should be noted that the marker-controlled SEON creates a clear segmentation with only two segmented objects, while the normal SEON create more than ten segments (most segments are very small and meaningless) when segmenting a similar image in Figure 17(a). Thus, marker-controlled SEON has a good potential to provide an effective solution for the problem of oversegmentation.



(a)



(b)

(c)

Figure 28. An example of scene segmentation by marker-controlled SEON

(a) An image with four planes. The two crosses represent the markers in the scene. Only the plane at the bottom and the background have markers. (b)-(c). Each picture shows the maximal activities for each object. Since only the plane at the bottom and the background have markers, the segmented image only consists of the planes at the bottom and the background. Other three planes are not segmented and are considered as background.

Although marker may provide an effective solution to oversegmentation, the selection of markers itself is highly heuristic and domain-dependent. The selection of markers has been extensively studied (Gonzalez, 1992). However, most methods cannot apply without specific domain knowledge. One possible approach for domain independent selection of markers is information theory, which could be an interesting area for further research.

References

Adams, R., & Dischof, L. (1994). Seeded region growing. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *16*, 641-647.

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.

Alina, N. M. (1998). Parallel marker-based image segmentation with watershed transformation. *Journal of Parallel and Distributed Computing*, *51*, 27-45.

Alina, N. M., Bogdan, C., & Moncef, G. (1998). Parallel watershed transformation algorithms for image segmentation. *Parallel Computing*, *24*, 1981-2001.

Alina, N. M., & Moncef, G. (1997). Parallel image component labeling with watershed transformation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*, 5, 441-450.

Anastasio, T. J. (1993). Modeling vestibulo-ocular reflex dynamics: from classical analysis to neural networks. Neural System: analysis and modeling (pp. 407-430). Norwell.

Andreou, A. G., & Edwards T. G. (1994). VSLI phase locking architectures for feature linking in multiple target tracking systems. Advances in Neural Information Processing Systems 6. Morgan Kaufmann. CA.

Ballard, D. H. (1981). Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition. 13*, 111-122.

Banks, S. (1990). Signal Processing, Image Processing and Pattern Recognition (pp. 215-237). Prentice Hall.

Baugher, E. S., & Rosenfeld, A. (1986). Boundary localization in an image pyramid. *Pattern Recognition*, *19(5)*, 373-396.

Beghdadi, A., Negrate, A. L., & de Lesegno, P. V. (1995). Entropic thresholding using a block source model. *Graphical Models and Image Processing*, *57*, 197-205.

Beucher, B., & Meyer, F. (1992). The morphological approach of segmentation: the watershed transformation. *Mathematical Morphology in Image Processing*, Dougherty, E. (ed.), Marcel Dekker, New York.

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

Biederman, I. (1987). Recogntion-by-component: A theory of human image understanding. *Psychol. Rev., 94*, 115-147.

Bieniek, A., & Moga, A. (2000). An efficient watershed algorithm based on connected components. *Pattern recognition*, *33*, 6, 271-280.

Bleau, A., & Leon, L. J. (2000). Watershed-based segmentation and region merging. *Computer Vision and Image Understanding*, *77*, 317-370.

Boahen, K. A. (1996). A retinomorphic vision system. *IEEE Micro, 16*, 30-39

Brink, A. D. (1995). Minimum spatial entropy threshold selection. *IEE Processings Vision, Image and Signal Processing*, *142*, 128-132.

Brummer, M. E. (1991). Hough transform detection of the longitudinal fissure in tomographic head images. *IEEE Transaction on Medical Imaging*, *10(1)*, 74-18.

Campbell, S. R., & Wang, D. L. (1996). Synchronization and desynchronization in a network of locally coupled Wilson-Cowan Oscillators. *IEEE Transactions on Neural Network*, *7(3)*, 541-554

Campbell, S. R., Wang, D. L., & Jayaprakash, C. (1999). Synchrony and Desynchrony in Integrate-and-Fire Oscillators. *Neural Computation*, *11*, 1595-1619

Canny, J. F. (1986). A computational approach to edge detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *8(6)*, 679-698.

Chang, Y. L., & Li, X. (1994). Adaptive image region-growing, *IEEE Transaction on Image Processing*. *3*, 868-872.

Chen, K., & Wang, D. L. (2002). Weight Adaptation and oscillatory correlation for image segmentation. *IEEE Transaction on Neural Network, 11(5)*, 1106-1123

Chen, K., & Wang, D. L. (2002). A dynamically coupled neural oscillator network for image segmentation. *Neural Networks*, *15*, 423-439

Cheng, H. D., Jiang, X. H., Sun, Y., & Wang, J. L. (2001). Color image segmentation: Advances and prospects. *Pattern Recognition*. *34*, 2259-2281.

Eckhorn, R., Bauer, R., Jordan, W., Brosch, M., Munk, M., & Reitboeck, R. J. (1988). Coherent oscillations: a mechanism of feature linking in the visual cortex? Multiple electrode and correlation analysis in the cat. *Biol. Cybernetics. 60*, 121-130

Engel, A. K., Konig, P., & Singer, W. (1991). Direct physiological evidence for scene segmentation by temporal coding. *Proc. Natl. Acad. Sci. USA*, *88*, 9136-9140.

Fox, J. J., Jayaprakash, C., Wang, D. L., & Campbell, S. R. (2001). Synchronization in Relaxation Oscillator Networks with Conduction Delays. *Neural Computation. 13*, 1003-1021

Furst, M. A. (1986). Edge detection with image enhancement via dynamic programming. *Computer Vision, Graphics, and Image Processing*, *33*, 263-279.

Gray, C. M. (1999). The temporal correlation hypothesis: still alive and well. *Neuron*, *24*, 31-47.

Gray, C. M., Konig, P., Engel, A. K., & Singer, W. (1989). Oscillatory responses in cat visual cortex exhibit intercolumnar synchronization with reflects global stimulus properties. *Nature*, *338*, 334-337

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

Glasbey, C. A. (1993). An analysis of histogram-based thresholding algorithms, *CVGIP – Graphical Models and Image Processing*, *55*, 532-537.

Gonzalez, R. C. (1992). Digital Image Processing. Wiley.

Gonzalez, R. C., Woods, R. E., & Eddins, S. L. (2004) Digital Image Processing Using Matlab (DIPUM) Toolbox. URL: http://www.imageprocessingbook.com/DIPUM/mfiles/dipum_toolbox.htm

Hafner, J., Sawhney, H. S., Equitz, W., Flickner, M., & Niblack, W. (1995). Efficient color histogram indexing for quadratic form distance functions. *IEEE Trans. Pattern Anal. Machine Intell.*, 16, 729-736.

Hopfield, J. J., & Herz, A. V. M. (1995). Rapid local synchronization of action potentials: Toward computation with coupled integrate-and-fire oscillator neurons. *Proceedings of the National Academy of Sciences of the United States of America*, *92*, 6655-6662

Judd, D. B., & Wyszeki, G. (1963). Color in Business, Science, and Industry. Wiley.

Julesz, B. (1995). Dialogues on perception. MIT Press, Cambridge MA.

Koffka, K. (1935), Principles of Gestalt Psychology. Harcourt, New York.

Kurita, T. (1995). An efficient clustering algorithm for region merging. *IEICE Transactions on Information and Systems*, *E78-D*, 1546-1551.

Laprade, P. H. (1998). Split-and-merge segmentation of aerial photographs. *Computer Vision, Graphics, and Image Processing*, *44(1)*, 77-86.

Lin, J. S., Cheng, K. S., & Mao, C. W. (1996). A fuzzy Hopfield neural network for medical image segmentation. *IEEE Transactions on Nuclear Science*, 43, *4*, 2389-2398.

Mack, A., Tang, B., Tuma, R., Kahn, S., & Rock, I. (1992). Perceptual organization and attention. *Cognit. Psychol.*, *24*, 475-501.

Madrenas, J., Cosp, J., Lucas, O., Alarcón, E., Vidal, E., & Villar, G. (2004). BIOSEG: A Bioinspired VLSI Analog System for Image Segmentation. In *Proceedings of the Twelfth European Symposium on Artificial Neural Networks* (ESANN'2004), Bruges, Belgium, pp. 411-416.

Mahowald, M. A., & Mead, C. (1989). Silicon retina. Wiley.

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.

Mirollo, R. E., & Strogatz, S. H. (1990). Synchronization of pulse-coupled biological oscillators. *SIAM Journal on Applied Mathematics. 50(6)*, 1645-1662

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

Neumann, O., & Sanders, A. F. (1996). Handbook of perception and action: Attention. Academic Press, San Diego.

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

Peskin, C. S. (1975). Mathematical aspects of heart physiology (pp. 268-278). Courant Institute of Mathematical Sciences, New York University, New York.

Rock, I., & Palmer, S. (1990). The Legacy of Gestalt Psychology. *Scientific American, 263*, 84-90.

Sejnowski, T. J., & Hinton, G. E. (1987), Separating figure from ground with a Boltzmann machine, Vision, Brain, and Cooperative Computation. MIT Press. Cambridge MA.

Singer, W. (1999). Neuronal synchrony: a versatile code for the definition of relations? *Neuron*, *24*, 49-65

Singer, W., & Gray, C. M. (1995). Visual feature integration and the temporal correlation hypothesis. *Annual Review Neuroscience, 18*, 555-586.

Sporns, O., Tononi, G., & Edelman, G. M. (1991). Modeling perceptual grouping and figure ground segregation by means of active re-entrant connections. *Proc. Natl. Acad. Sci. USA , 88*, 129-133.

Trick, L. M., & Pylyshyn, Z. W. (1994). Why are small and large numbers enumerated differently? A limited-capacity preattentive stage in vision. *Psychol. Rev*, *101*, 80-102.

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

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.

Wakefield, J. (2001). A mind for consciousness, *Scientific American*, *285*, 26-27.

Wang, D. L. (1995). Emergent synchrony in locally coupled neural oscillators. *IEEE Transactions on Neural Network*, *6*, 941-948.

Wang, D. L., & Liu, X. (2002). Scene Analysis by integrating primitive segmentation and associative memory. *IEEE Transactions on Syst. Man Cybern. – Part B: Cybern.*, *32(3)*, 254-268.

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. (1997). Image segmentation based on oscillatory correlation. *Neural Computation*, *9*, 805–836. (For errata see *Neural Computation*, *9*, 1623–1626, 1997.)

Yen, S. C., Menschik, E. D., & Finkel, L. H. (1999). Perceptual grouping in striate cortical networks mediated by synchronization and desynchronization, *Neurocomputing*, *26-27*, 609-616.

Zhu, S. C., & Yullie, 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.