UNIVERSITY OF SOUTHAMPTON

FACULTY OF PHYSICAL AND APPLIED SCIENCES

Electronics and Computer Science

An Attractor Network of Weakly-Coupled Excitable Neurons For General Purpose of Edge Detection

by

Shaobai Li

Thesis for the degree of Doctor of Philosophy

January 2015

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL AND APPLIED SCIENCES
Electronics and Computer Science

Doctor of Philosophy

AN ATTRACTOR NETWORK OF WEAKLY-COUPLED EXCITABLE NEURONS FOR GENERAL PURPOSE OF EDGE DETECTION

by Shaobai Li

The prospect of emulating the impressive computational capacities of biological systems has led to much interest in the design of analog circuits, potentially implementable in VLSI CMOS technology, that are guided by biologically motivated models. However, system design inevitably encounters the contrary constraints of size(or complexity) and computational power (or performance). From a high level design point of view, we believe that theoretical analysis of the model properties will undoubtedly benefit the implementation at a lower level. This thesis focuses on this simple aim to provide an extensive study of task-specific models based on dynamical systems in order to reduce model complexity or enhance the performance of algorithms.

In many examples, it is the self-evolving dynamics of the model that allows the intrinsic parallel computations of algorithms, which are traditionally expressed by differential equations and systems. For instance, simple image processing tasks, such as the detection of edges in binary and grayscale images, have been performed by a reaction diffusion equation using the FitzHugh-Nagumo model as the reaction term in the previous work done by Kurata et al. (2008); Nomura et al. (2003, 2008, 2011b,a). Once the initial condition is correctly assigned according to a processed image, system states of this model will automatically evolve to the final result.

From an application of view, the spatial distribution of system state can be regarded as a grid network with a proper discrete pattern; each network node becomes a FitzHugh-Nagumo type of neuron, while the diffusion term turns out to be the nearest couplings among them, where the coupling strength k is proportional to the original coefficient of diffusion D. So, one neuron (node) in the network deals with one pixel in the processed image. However, in previous study, this one-to-one mapping of image pixels to component neurons makes the size of the network a critical factor in any such implementation. The wrong edges are found due to the intrinsic mechanism of the algorithm when the diffused the processed image are used to pick out edges among the grayscale intensity levels and their most successful method solves this problem by a doubling of the size of the network.

In the thesis, we propose two main improvements of the original algorithm in order for the smaller complexity and the better performance. We treat dynamics of the coupled system for the purpose of edge detection as a k-perturbation of the uncoupled one. Based on stability analysis of system state for both uncoupled and coupled cases, the system used for edge detection is identified as a *Multiple Attractor* type network and the final edge result corresponds to an *attractor* in high dimensional space. Hence, we conclude that the edge detection problem maps an image to an initial condition that is correctly located within the attraction domain of an expected attractor. For the first improvement, in order to get rid of the wrong edges, we provide a way of quantify the excitability of uncoupled neurons based on the Lyapunov exponents so that the boundary of attraction domain of the attractors can be well estimated. Moreover, an anisotropic diffused version of processed image is used for the further enhancement on the performance. For the second improvement, in order for diffusion of the processed image being accomplished by the hardware, we introduce a self-stopping mechanism to the original equation. Moreover, we link the basic design rules on system parameter settings to the fundamental theorem of WCNN (weakly coupled neural network) (Hoppensteadt and Izhikevich, 1997), which states that an uncoupled neuron must be near a threshold (bifurcation point) in order for rich dynamics to be presented in the coupled network. We apply our techniques to detect edges in data sets of artificially generated images (both noise-free and noise-polluted) and real images, demonstrating performance that is as good if not better that the results of (Nomura et al., 2011b,a) with a smaller size of the network.

Contents

Declaration of Authorship **xiii**

Acknowledgements **xv**

1 Introduction **1**

1.1 Previous Works . . . 3

1.2 Objective . . . 4

1.3 Outline of the Thesis . . . 4

2 Image Processing Basics **7**

2.1 Digital Image Representation . . . 7

2.2 Edge Detection . . . 9

2.2.1 Edge Map and Evaluation of Edge Detection Algorithm . . . 9

2.3 Test Images . . . 10

3 Dynamical System For Edge Detection **13**

3.1 Basics on ODEs . . . 13

3.2 FitzHugh-Nagumo Model . . . 16

3.3 The Model of Edge Detection . . . 22

3.3.1 From Reaction Diffusion System to Coupled Neuronal Network . . . 23

3.3.2 Excitability and Fast Slow Dynamics . . . 25

3.4 Nomura Edge Detection Algorithms . . . 30

3.4.1 Binary Edge Detection Algorithm . . . 30

3.4.2 Gray Level Image Edge Detection Algorithm . . . 31

3.4.3 Improved Gray Level Image Edge Detection Algorithm . . . 33

3.4.4 Conclusion . . . 34

4 Theory of Weakly Coupled Neural Network **35**

4.1 Stability of Coupled Networks at Origin . . . 36

4.1.1 Coupling Network and Laplacian Matrix . . . 36

4.2 Attractor Network . . . 40

4.2.1 Edge Map: High Dimensional Attractor after Bifurcations . . . 40

4.2.2 Effect of Coupling . . . 42

4.3 Discussion and Conclusion . . . 49

5 Quantification of Excitability Using Continuous Lyapunov Exponents **53**

5.1 Problem Caused by Variable Threshold . . . 53

5.2 Proposed Edge Detection Method by Using Anisotropic Diffusion . . . 57

5.2.1 Image Threshold: Boundary of Attraction Domain 57
5.2.2 Computation of Lyapunov Exponents 58
5.2.3 Anisotropic Diffusion 62
5.3 Results 65

6 Network Stability **75**
6.1 Self-Stopping Diffusion 76
6.1.1 Modified Model Equations 79
6.2 Stability of Multi-Attractor Type Network 79
6.2.1 Analysis on Jacobian: Hopf Bifurcation in Coupled Network 80
6.3 Parameter Settings for Uncoupled Neurons 84
6.4 Anisotropic Coupling Strength 88
6.5 Results 91

7 Conclusion **97**
7.1 Comparison on Models and Methods 98

A Center Manifold Reduction of Planar FitzHugh-Nagumo Model **103**

B Silicon Neurons and Networks **107**

References **113**

List of Figures

2.1 Convention of the coordinates of the digital image representation 7

2.2 Artificial images and the corresponding ground truth of the edge data. . . 11

2.3 Real images used for testing algorithms of edge detection. 2.3(a) Traffic cone. 2.3(b) Tire. 2.3(c) Pillow. 2.3(d) Video camera. 2.3(e) Stairs. 2.3(f) Briefcase. All the real images are provided by Heath et al. (1997) which are available on the website "Edge Detector Comparison". 12

3.1 Phase portrait of FitzHugh-Nagumo model 16

3.2 Nullclines of FitzHugh-Nagumo models for different values of the parameters a and b 18

3.3 Bifurcation structure of single FitzHugh-Nagumo model 20

3.4 Phase portrait of FitzHugh-Nagumo system with three equilibria 21

3.5 Network structure of the system 25

3.6 Excitability of single FitzHugh-Nagumo model 27

3.7 Phase portraits of FitzHugh-Nagumo model in Equation 3.3 with different values of ϵ. For the parameter settings, $a = 0.25$, $b = 1$, and ϵ is chosen as 0.001 for (a), 0.01 for (d) and 0.1 for (g). For all the values of ϵ, two types solutions starting respectively with $(0.26, 0)$ (blue trajectories) and $(0.24, 0)$ (green trajectories) are plot in the phase portraits followed by their corresponding curves in time domain. The initial conditions $(0.24, 0)$ and $(0.26, 0)$ is separated by $(a, 0) = (0.25, 0)$, while the two trajectories are widely separated only when ϵ is set very small. 29

3.8 Edge detection results on one dimensional step function in Equation 3.20 and Equation 3.21 with constant image threshold. For other parameter settings, $\epsilon = 0.001$, $b = 1$, $k_v = 4$ and $k_w = 20$ 31

3.9 Color maps of one dimensional edge detection 32

4.1 Network Structure 37

4.2 Bifurcation diagram on coupling strength as bifurcation parameters 43

4.3 The effect of v coupling on system dynamics. The first four figures (a), (b), (c) and (d) present simulation for $\epsilon = 0.001$ and the rest four figures (e), (f), (g) and (h) present that for $\epsilon = 0.0005$. Figures in the left column illustrate the simulation of the uncoupled models, while figures in the right column are that of the coupled models. In all the simulation of coupled models, $k_w = 10$ and k_v is swept from 0 to 10 with the step of 0.5. For other parameter settings, $a_1 = a_2 = a = 0.25$,$b_1 = b_2 = b = 4$. . . 45

4.4 Structural stability of the uncoupled system to the extremely large coupling through the state variable w. For parameter settings, $a_1 = a_2 = 0.25$, $\epsilon = 0.001$. For coupling strength $k_{v1} = k_{v2} = 1$ and $k_{w1} = k_{w2} = 5 \times 10^3$. Two straight lines respectively indicates the boundary of saddle-node bifurcation (solid line) and Hopf bifurcation (dashed line). The boundaries exhibits the symmetry due to the same setting of a_i. Generally, the uncoupled network requires to be set near to its bifurcation points, to ensure that it undergoes the bifurcations after the coupling is included. 46

4.5 Two groups of simulations to check the structural stability of the uncoupled system to the extremely large coupling through the state variable w. For parameter settings, $a = 0.25$, $\epsilon = 0.001$ in both simulations. However, b is chosen to be 2 in first group and 4 in second one. In the coupled systems for both simulation, k_w is swept from 10^0 to 10^3. 48

5.1 Image threshold $\theta_i(\tau)$ with τ equal to 0.015, 0.030, 0.060, 0.120, 0.200, 0.300, 0.500 and 1.000, respectively corresponding to the cases provided in Fig. 5.2 54

5.2 Edge detection results on one dimensional step function in Equation 5.1 with $a_i = \theta_i(\tau; d = 10)$ which is the variable image threshold obtained from Equation 3.24 with different values of τ. For other parameter settings, $b = 1$, $\epsilon = 0.001$ $k_v = 4$ and $k_w = 20$. 56

5.3 Lyapunov exponents of the single FitzHugh-Nagumo model with different values of parameter a. 61

5.4 The quasi-linear relationship between system parameter a and image threshold θ^*. 62

5.5 Artificial images and the corresponding ground truth of the edge data. . . 67

5.6 Edge detection results for artificial images. For the Nomura method, the system parameters are set as $b = 1$, $\epsilon = 0.001$, $k_v = 4$ and $k_w = 20$, and the two diffusive constants $d^0 = 40$ and $d^1 = 200$ in Equation 3.24. Both the diffusion stop times are $\tau = 1.0$. For the proposed method, the system parameters are set as $b = 3.5$, $\epsilon = 0.001$ and $k_w = 5$. The constant $K = 10$ and the threshold $\eta = 0.0$ in Equation 5.13. The steady state time $\tau_s = 1.0$ in the model equations for both algorithm. 68

5.7 Robustness test of Algorithm 2 in comparison to Canny detector. The value of measures are taken as the average of 50 simulations. 69

5.8 Robustness test of Algorithm 2 in comparison to the method using pure local gradient magnitude with different values of η in Equation 5.13. In each figure, the dashed line stands for the method using local gradient magnitude and the solid line stands for Algorithm 2. $\eta = 0.05$ for 5.8(a) and 5.8(b). $\eta = 0.15$ for 5.8(c) and 5.8(d). $\eta = 0.25$ for 5.8(e) and 5.8(f). The false positives become much fewer when value of η increases for both algorithms. Algorithm 2 has largely reduced fp_r curves with slight loss of tp_r. The value of measures are taken as the average of 50 simulations. . 70

5.9 Edge detection results for real images, (a) Traffic Cone [604×437]. (e) and Tire [512 × 512]. (b) and (f) provide the edge detection results obtained by adopting the Nomura method (Nomura et al., 2011a), in which the system parameters are $b = 1$, $\epsilon = 0.001$, $k_v = 4$ and $k_w = 20$, and the two diffusive constants in Equation 3.24 for the diffusive a_i^0 and a_i^1 are respectively set as $d^0 = 40$ and $d^1 = 200$. Both the diffusion stop times are $\tau = 1.0$ and the steady state time $\tau_s = 1.0$ for the model equation Equation 3.26. (c) and (g) provide the edge detection results with the proposed method in Algorithm 2 using anisotropic diffusion, in which the system parameters are $b = 3.5$, $\epsilon = 0.001$, $k_v = 0$ and $k_w = 5$. The constant $\widetilde{d} = 10$ and the threshold $\eta = 0.05$ in Equation 5.13 for all the images. And the anisotropic diffusion stopping time is $\tau = 1.0$. The steady state time $\tau_s = 1.0$ for the model in Equation 4.12. ((d)) and ((h)) are the results obtained by the Canny's algorithm where the thresholds are automatically determined by the program provided in MATLAB. . . . 72

5.10 Edge detection results for real images, (a) Pillow [468 × 552]. (e) Video camera [435 × 577]. (b) and ((f) provide the edge detection results obtained by adopting the Nomura method (Nomura et al., 2011a) ,in which the system parameters are $b = 1$, $\epsilon = 0.001$, $k_v = 4$ and $k_w = 20$, and the two diffusive constants in Equation 3.24 for the diffusive a_i^0 and a_i^1 are respectively set as $d^0 = 40$ and $d^1 = 200$. Both the diffusion stop times are $\tau = 1.0$ and the steady state time $\tau_s = 1.0$ for the model e-quation Equation 3.26. ((c)) and ((g)) provide the edge detection results with the proposed method in Algorithm 2 using anisotropic diffusion, in which the system parameters are $b = 3.5$, $\epsilon = 0.001$, $k_v = 0$ and $k_w = 5$. The constant $\widetilde{d} = 10$ and the threshold $\eta = 0.05$ in Equation 5.13 for all the images. And the anisotropic diffusion stopping time is $\tau = 1.0$. The steady state time $\tau_s = 1.0$ for the model in Equation 4.12. (d) and ((h)) are the results obtained by the Canny's algorithm where the thresholds are automatically determined by the program provided in MATLAB. . . . 73

6.1 Diffusion of Step 76

6.2 Diffusion of Step 78

6.3 Bifurcation structure of a single FitzHugh-Nagumo model 85

6.4 Edge map obtained by Algorithm 2 and Algorithm 3 using both self-stopping diffusion and diffusion with a stopping time. In Algorithm 2, the parameter settings are $b = 3.5$, $\epsilon = 0.001$, $\eta = 0$, $\widetilde{d} = 10$ and $\tau = 1$. For Algorithm 3, $\mu = 0.25$ and $\nu = 0$. For the self-stop diffusion, $\xi = 3$, otherwise for the original diffusion, a stopping time τ is chosen to be 1. The results are all the same as shown in (c), where $tp = 5758$, $tp_r = 98.37\%$, $fp = 0$ and $fp_r = 0\%$. 92

6.5 Robustness test for the proposed method in comparison with the previous algorithm in Section 5.2.3. Performance measures on the previous method is illustrated by the dashed lines while measures on the proposed method is illustrated by the solid lines. Generally, the results obtained by the proposed method have less false positives. However, the results obtained by the previous method have more true positive points. 93

6.6 The true positive rates of these two edge maps are nearly the same, $tp_r = 0.8112$ for 6.6(b) and $tp_r = 0.8110$ for 6.6(b). The false positive rate of the later ($fp_r = 0.0025$) is lower than that of the former ($fp_r = 0.0030$). . 94

6.7 Edge detection results for real images, (a) Pillow [468 × 552]. (d) Video camera [435×577]. (b) and (e) provide the edge detection results obtained by Algorithm 2, in which the system parameters are $b = 3.5$, $\epsilon = 0.001$, $k_v = 0$ and $k_w = 5$. The constant $\widetilde{d} = 10$ and the threshold $\eta = 0.05$ in Equation 5.13 The stopping time of the anisotropic diffusion is $\tau = 1.0$. The steady time $\tau_s = 1.0$ for all the models. (c) and (f) provide the edge detection results obtained by Algorithm 3, in which the parameter b is set according to Equation 6.38 where $\mu = 0.25$ and the coupling strength k_i is set according to Equation 6.39 where $\nu = -0.05$. The diffusive coefficient $\xi = 3$ in Equation 6.8. The steady state time $\tau_s = 1.0$ for the model. . . . 95

6.8 Edge detection results for real images, (a) Stairs [441×579]. (d) Briefcase [419 × 577]. (b) and (e) provide the edge detection results obtained by Algorithm 2, in which the system parameters are $b = 3.5$, $\epsilon = 0.001$, $k_v = 0$ and $k_w = 5$. The constant $\widetilde{d} = 10$ and the threshold $\eta = 0.05$ in Equation 5.13 The stopping time of the anisotropic diffusion is $\tau = 1.0$. The steady time $\tau_s = 1.0$ for all the models. (c) and (f) provide the edge detection results obtained by Algorithm 3, in which the parameter b is set according to Equation 6.38 where $\mu = 0.25$ and the coupling strength k_i is set according to Equation 6.39 where $\nu = -0.05$. The diffusive coefficient $\xi = 3$ in Equation 6.8. The steady state time $\tau_s = 1.0$ for the model. . . . 96

7.1 One pixel to one neuron structure of the network model 97

List of Tables

4.1 List of all the high dimensional equilibria $\overline{\mathbf{X}} = (v_1, w_1, v_2, w_2)^T$ for the coupled two neurons in Equation 4.8. For the parameter settings, $a_1 = 0.1$, $a_2 = 0.2$, $b_1 = b_2 = 4$. For the settings of coupling strength, $k_v = 1$ and $k_w = 5$. . . . 41

4.2 List of eigenvalues $\mu|_{\overline{\mathbf{X}}} = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)^T$ of Jacobian in Equation 4.9 for all the equilibria in Table 4.1. Specifically, $\overline{\mathbf{X}}_2$ and $\overline{\mathbf{X}}_3$ are stable equilibria, because all the real parts of their eigenvalues of Jacobian are negative. . . . 41

5.1 Quantitative evaluations of edge detection algorithms . . . 65

6.1 $k_1 = 0.47157$, $k_2 = 0.23815$, $k_3 = 0.23455$, $k_4 = 0.1$ . . . 87

6.2 $k_1 = 0.48150$, $k_2 = 0.23828$, $k_3 = 0.23699$, $k_4 = 0.1$ . . . 87

6.3 $k_1 = 0.54$, $k_2 = 0.48259$, $k_3 = 0.47759$, $k_4 = 0.1$ . . . 87

6.4 $k_1 = 0.24548$, $k_2 = 0.12334$, $k_3 = 0.12264$, $k_4 = 0.1$ . . . 89

6.5 $k_1 = 0.24548$, $k_2 = 0.24754$, $k_3 = 0.24534$, $k_4 = 0.1$ . . . 89

7.1 Comparison on steps among Nomura's method, previous method and proposed algorithms . . . 101

Declaration of Authorship

I, Shaobai Li , declare that the thesis entitled *An Attractor Network of Weakly-Coupled Excitable Neurons For General Purpose of Edge Detection* and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- none of this work has been published before submission

Signed:...

Date:..

Acknowledgements

I would like to first thank Dr. Koushik Maharatna for leading me to such a great research topic. I would like to express my deepest gratitude to Dr. Srinandan Dasmahapatra for his excellent guidance, caring and great patience during all these years of my research. To those from whom I have learned, received inspiration, feedback and support, I am beholden. Taihai Chen, Rob Mills, Ke Yuan, Wenkang Xu: this thesis could not have happened without you. I would also like to thank all my families, especially my dear mum, for her long wait and selfless giving. Finally, special thanks goes to my girl friend (now my wife), Qian Sun. She has always been standing by me through the good times and bad.

Chapter 1

Introduction

Biological neural networks carry out massively parallel, robust computation while dissipating low power and have brought many neuroscientists and neuromorphic system designers under their spells. On the neuroscientific side, in 1952, Hodgkin and Huxley (1952) modelled the generation and propagation of the *spike* among the cells in the giant axon of the squid by using nonlinear partial differential equations. *Spike* is a sudden change in the neuronal membrane potential which is also referred to as *action potential.* Hodgkin-Huxley's model provides a new approach to understand the mechanism of this most common phenomena in neuron cells by using dynamical system theory. Since then, such neurophysiological models are analyzed, simplified and further developed with the help of dynamical system theory (DST). One of the most famous jobs was done by FitzHugh (1961) and Nagumo et al. (1962). They use a planar system (a two-dimensional system, originally the dimension of Hodgkin-Huxley is four) to replicate the complicated dynamics presented in Hodgkin-Huxley model. On the engineering end, Mahowald and Douglas (1991) proposed the concept of a "silicon neuron" - a silicon device whose physics resembled that of a neuronal membrane, and which could thereby replicate neural behaviour in real time by analog electronic circuitry. Since then, the techniques that transform the neuron models described by the dynamical systems into corresponding integrated circuits have been thoroughly studied and a large number of silicon neurons and neuromorphic silicon chips have been built (Basu and Hasler, 2010)(Cosp et al., 2008)(Wijekoon and Dudek, 2008). However, all of these designs face a fundamental trade-off between, on the one hand, the level of abstraction at which the neurons are modelled and circuits designed, often using DST (Arthur and Boahen, 2011); and on the other, the size of the silicon area on which they are fabricated. As a consequence, even though many successful neuromorphic and neurophysiological neuron chips emulating the sensory visual network, such as artificial silicon retina (Zaghloul and Boahen, 2006), have been designed, they perform image processing tasks at lower resolution (normally 50×50 pixels) compared to digital processors. Hence, system size is the most important issue when designing the neurally inspired circuit for the purpose of image processing.

In fact, there have always been protracted disputes on the level of abstraction for neural models among neuroscientists and neuromorphic system designers. In 1943, inspired by the topology of real neural networks, i.e., synaptic organization, and in order to explore the emergence of logical reasoning within such structures, McCulloch and Pitts (1943) proposed a mathematical model of simplified neurons with an activation threshold and binary outputs. Each single neuron can take as input, an arbitrary number of outputs of other neurons. The coupling of neurons within such a network is composed of direct connections and each connection has a weight. A neuron is set to its active state if the weighted sum of all the incoming signals(i.e. all the states of presynaptic neurons) exceeds the threshold, and inactive if not. In order to use a network of such simplified neurons to emulate the dynamics of logical reasoning, their first applied this model to compute a series of boolean functions.

Clearly, the McCulloch-Pitts neuron is a rough imitation which mainly draws upon the morphological properties of neurons. With advances in neurobiology, the artificial models have incorporated key biological features, such as activation functions instead of binary thresholds, and with a continuous set of possible output values. According to Maass's classification (Maass, 1997, 2001), networks of McCulloch-Pitts neurons with binary valued functions are called first generation neural network models, while their extensions which contain continuous activation functions and real-valued outputs, called *firing rates*, are called second generation neural network models. Many physiologists agree that firing rates describe how neurons encode and transmit information. Such neural networks can implement complicated mathematical functions, and have been shown to be capable of learning such functions. A considerable number of supervised learning and unsupervised learning rules have been proposed, many of which can be traced back to Hebb's work on synaptic plasticity (Hebb, 1949) in 1949, driven by coincident firing of connected neurons.

Nevertheless, this type of artificial neural networks are not satisfying for all neuron model designers. There are still a large number of researchers who believe that this kind of neuron is strongly restrictive for large amounts of data processing due to its lack of employment of spike dynamics. Arguments against the firing rate coding as neural information transmission can be found especially in vision area. Thorpe (1990) pointed out Poisson-like rate codes failed to satisfy rapid sensory processing in human vision. Moreover, Eckhorn et al. (1989) discovered that binary images are created through neuronal oscillation in the cat midbrain whose features are used to create the actual image in the further layer. Other evidence found in the electrosensory system of electric fish (Heiligenberg, 1991), in the auditory system of echo-locating bats (Kuwabara and Suga,

1993) confirm that neurons process sensory information based on action potentials (numbers, timing, phase etc.) or their interactive activities (network oscillation, concurrent rhythms etc.).

1.1 Previous Works

Dynamical neuron models are also widely adopted in many cases of computer vision and image processing. Chen and Wang (2002) have performed very innovative work using oscillators for image segmentation. They constructed oscillator networks with local excitatory lateral connections and a global inhibitory connection. Yen and Finkel (1998) simulated facilitatory and inhibitory interactions among oscillators to do contour integration. Li (2001) has proposed elaborate visual cortex models with oscillators and applied them on lattice drawings. Yu and Slotine (2009) proposed a network of oscillatory neurons of Morris-Lecar model (a generalised FitzHugh Nagumo model) for the purpose of visual grouping. Nomura and his colleagues (Nomura et al., 2003, 2008, 2011b,a; Kurata et al., 2008) use a discrete reaction diffusion system for detecting edges in real grayscale images. Moreover, such kinds of reaction-diffusion networks are built using CMOS technology (Karahaliloğlu and Balkır, 2004).

Our work is based on the Nomura's network model for the purpose of edge detection. Specifically in his model, each neuron is connected only with all the four in its neighbourhood with diffusive coupling. Once the initial condition is correctly set according to the processed image, the system variable in neural model will automatically search for the edge map as the final steady state. They gave special importance to the stability of the output as a distinctive from other algorithm using reaction-diffusion systems. And the stability of the output makes the system more suitable in implementation with analogy circuitry. Their algorithm (Nomura et al., 2003) was first applied to detect edges in binary images by assigning the system parameter a with a constant threshold between the two intensity levels, which they referred to *image threshold*. The algorithm was then extended from binary to grayscale images (Nomura et al., 2011b) by replacing the constant image threshold with a variable one, which are obtained by diffusing the processed image. In doing so, they encountered some problems that they overcame by enlarging the size of the system for better performance (Nomura et al., 2011b). They failed to provide sufficient theoretical analysis of the algorithm to explain the ability of edge detection of the network. Consequently, they failed to provide the generalisation of the proper way to set system parameters. They attempted to explain the stability of the network model based on *Turing instability* which are denied in the later work done by themselves Kurata et al. (2008).

In this thesis, we emphasise the theoretical analysis of this network model, in order to deal with the problem they found. In the high level design procedure, we set out to refine in performance and reduce in size, a network model for edge detection, that can be more suitable to be fabricated in silicon, based on the dynamical system theory.

First, we provide the definition of the image threshold for a monostable FitzHugh-Nagumo model by evaluating the maximum Lyapunov characteristic exponent, so that we can correctly formulate the problem introduced when extending the algorithm to grayscale images. Next, according to the result of the analysis of two coupled FitzHugh-Nagumo models as a network, we also simplify the network system by eliminating the coupling term of the membrane potential. Finally, we propose a novel method of obtaining the threshold by anisotropic diffusion in order to solve the problem.

1.2 Objective

It is the explicit objective of this research to explore, analyze and create biologically inspired dynamical models of neural systems and networks for the purpose of information processing, which are also implementable and optimized for CMOS VLSI design technology. Our engineering application specifically focuses on the simple image processing task of edge detection. And this thesis focuses on the methodologies which is based on DST for the following purposes:

- simplify the existing structure of system dynamics for edge detection
- improve the resultant performance without increasing the system scale.

Based on the work done by Nomura et al. (2008, 2011a), the first task of our study is to identify the system model used for the purpose of edge detection as the *Multiple Attractor* network, according to the theory of *weakly coupled neural network* (WCNN). We correspond the shape of some of the non-hyperbolic equilibria $\overline{\mathbf{X}}_{\mathcal{M}}$ of the network model to the potential edge pattern $\mathcal{M}$ and illustrate their stability through simulations on small network. Then, in a second task, we study the "wrong edge" problem and ascribe it to the poor accuracy of the boundary domain $\partial\mathcal{B}(\overline{\mathbf{X}}_{\mathcal{M}})$ and the usage of isotropic diffusion of the processed image as image threshold. Finally, we attempt to redesign the system following the stability analysis of the system and present a reasonable method of setting system parameters and coupling strengths according to the fundamental theorem of weakly coupled neural network (WCNN).

1.3 Outline of the Thesis

This thesis is organised as below,

- In Chapter 2, we present some fundamental knowledge in the field of image processing. Some important concepts that are frequently referred to throughout the thesis, are formally defined. Particularly, we introduce the criteria for evaluating the edge detection performance of algorithms for artificial images used in this thesis.

- In Chapter 3, we provide a geometrical analysis of a planar single FitzHugh-Nagumo model. Specifically, we provide a few bifurcation patterns on a single FitzHugh-Nagumo model. Then, we derive, from the continuous reaction diffusion equations, the basic network model, of which the single element is a FitzHugh-Nagumo type neuron. After this, we link the detailed dynamical analysis on a single neuron to its neural excitability which is the inspiration of the edge detection algorithm. At last, we provide a brief review and discussion on the work by Nomura et al. (2003, 2008, 2011b,a) and Kurata et al. (2008) on the development of the edge detection algorithm. In particular, we point out the problems they encountered in defining thresholds in grayscale images.

- Chapter 4 covers a basic theoretical analysis of the edge detection ability of the network model introduced in Chapter 3. Specifically, we first provide an analytical analysis of the stability of system at the origin, in order to deny the original explanation provided in (Nomura et al., 2003, 2008, 2011b) using *Turing instability*. Then, we provide the stability analysis on the small network consisting of only two coupled neurons. After identifying the network type of the model, we change the coupling strength as bifurcation parameter. We find that Hopf bifurcation points in the coupled network determine the stability boundaries of attractors $\overline{\mathbf{X}}_{\mathcal{M}}$ representing the edge patterns $\mathcal{M}$. Based on this, we simplify the network model by eliminating the coupling of one specific state variable. Finally, we link the system model to the theory of Weakly Coupled Neural Network (WCNN) (Hoppensteadt and Izhikevich, 1997).

- Chapter 5 covers the modification on the original algorithm proposed by Nomura et al. (2008, 2011a,b) in order to deal with the accompany "wrong edge" problem. Firstly, we provide a detailed analysis on the such a problem of Nomura's model with an example of one dimensional edge detection. Secondly, we show that the boundaries of the attraction domain $\partial\mathcal{B}(\overline{\mathbf{X}}_{\mathcal{M}})$ are closely related to the excitability of each single neuron. Specifically, we provide the improved definition of the thresholds θ of excitability of the single neuron by using *Lyapunov exponent*. Thirdly, we construct our model to perform edge detection by introducing an anisotropic diffusion and we evaluate its performance on both artificial and real images. As a result, the edge detection performance for the artificial images are improved and the edge detection performance for the real images are also comparable to Nomura's method (Nomura et al., 2011a) while halving the network size. The results of this chapter were the following publication:

- S. Li, S. Dasmahapatra and K. Maharatna. Dynamical system approach to edge detection using coupled FitzHugh-Nagumo neurons. IEEE Transactions on Image Processing, Currently under review:-, 2014 expected.

- In Chapter 6, we mainly focus on the stability of the final output of detected edges to refine our design proposed in the previous chapter. Firstly, we redesign the process of diffusing processed image in order to assign the threshold of excitability θ. As a result, the diffusion process will be automatically stopped due to the intrinsic mechanism and it becomes more suitable for the hardware design as no stopping time is required to be computed. Secondly, we link the basic rules of design to the fundamental theorem of WCNN theory so that we are able to provide the rationale setting system parameters and coupling strengths, which the previous work failed to provide. Specifically, the system parameters are properly set in order for the uncoupled network being close to the bifurcation and the coupling strength is designed to be anisotropically controlled by the magnitude of gradient from a processed image. The results for both artificial and real images are provided in comparison with the method proposed in Chapter 5.

- Finally, Chapter 7 presents the most important conclusions drawn in the thesis.

Chapter 2

Image Processing Basics

The focus of this chapter is on the background of image processing to this thesis. In Section 2.1, we will start with some important concepts of image processing used throughout the whole thesis. And in Section 2.2.1, the evaluation criteria for the performance of the edge detection algorithms used in the thesis will be introduced. Finally in Section 2.3, we introduce the test images.

2.1 Digital Image Representation

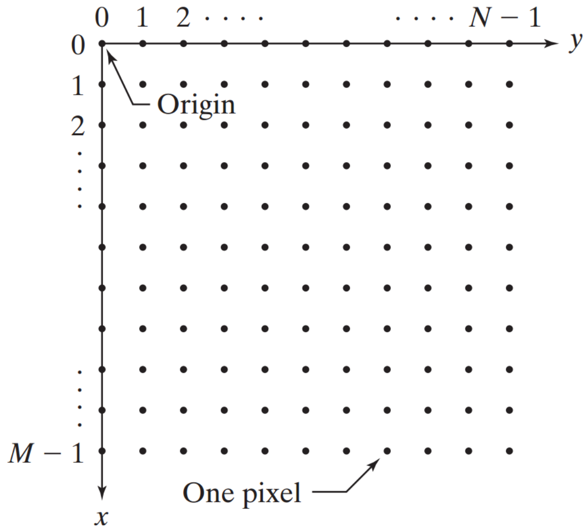


Figure 2.1: Convention of the coordinates of the digital image representation

In a digital image processing context (Gonzalez and Woods, 2008), an input image considered as an analog two-dimensional input is conventionally described by a continuous function $U(\mathsf{x},\mathsf{y})$ of two continuous variables x and y. The map

$$U : \mathsf{x}, \mathsf{y} \mapsto U(\mathsf{x}, \mathsf{y})$$

indicates the image intensity at a specific point (x, y) and thereby the whole function $U(\mathsf{x}, \mathsf{y})$ describes a spatial distribution of the image signal. In order to numerically process such a spatial signal in digital computer, it is usual to convert $U(\mathsf{x}, \mathsf{y})$ into a 2-D numerical arrays $U(\mathsf{m}, \mathsf{n})$ consisting of M rows and N columns. (m, n) is a pair of non-negative integers denoting the coordinate of original image intensity distribution, where

$$0 \leqslant \mathsf{m} \leqslant \mathsf{M} - 1, \quad 0 \leqslant \mathsf{n} \leqslant \mathsf{N} - 1$$

Figure 2.1 shows the coordinate representation of an digital image $U(\mathsf{m}, \mathsf{n})$. Suppose m and n are the lengths of the image in x or y direction and spatial variables (x, y) are discretised uniformly in both directions with the finite difference $\Delta\mathsf{h} = \mathsf{l}_\mathsf{x}/\mathsf{M} = \mathsf{l}_\mathsf{y}/\mathsf{N}$. The $\mathsf{M} \times \mathsf{N}$ numerical arrays illustrated in Figure 2.1 can be written in a more traditional matrix notation:

$$U = \begin{pmatrix} U_{(0,0)} & U_{(0,1)} & \cdots & U_{(0,\mathsf{N}-1)} \\ U_{(1,0)} & U_{(1,1)} & \cdots & U_{(1,\mathsf{N}-1)} \\ \vdots & \vdots & & \vdots \\ U_{(\mathsf{M}-1,0)} & U_{(\mathsf{M}-1,1)} & \cdots & U_{(\mathsf{M}-1,\mathsf{N}-1)} \end{pmatrix} \tag{2.1}$$

Each element of this matrix $U_{(\mathsf{m},\mathsf{n})}$ is called an *image element* or a *pixel*. Clearly, $U_{(\mathsf{m},\mathsf{n})} = U(\mathsf{x} = \mathsf{m} \cdot \Delta\mathsf{h}, \mathsf{y} = \mathsf{n} \cdot \Delta\mathsf{h})$. Sometimes for the sake of convenience of expression, we also pack the pixels in U into an $\mathsf{MN} \times 1$ column vector in column order. If denote the sequence number of element in the column vector as i, then the mapping the coordinate (m, n) in U to the index i can be defined as,

$$\begin{aligned} f_c : \quad & \mathbb{Z}^{\geqslant 0} \times \mathbb{Z}^{\geqslant 0} \to \mathbb{Z}^{+} \\ & (\mathsf{m}, \mathsf{n}) \mapsto f_c(\mathsf{m}, \mathsf{n}) = i, \\ & i = \mathsf{n} \cdot \mathsf{M} + \mathsf{m} + 1 \end{aligned} \tag{2.2}$$

Correspondingly, the inverse of the map will be,

$$\begin{aligned} f_c^{-1} : \quad & \mathbb{Z}^{+} \to \mathbb{Z}^{\geqslant 0} \times \mathbb{Z}^{\geqslant 0} \\ & i \mapsto f_c^{-1}(i) = (\mathsf{m}, \mathsf{n}), \\ & \mathsf{m} = (i - 1) \mod \mathsf{M} \\ & \mathsf{n} = \lceil \frac{i}{\mathsf{M}} \rceil \end{aligned} \tag{2.3}$$

$\lceil x \rceil$ is the *ceiling* function denoting minimum integer greater than x. So, in the rest of the thesis, we will use both notations with different subscripts, such as U_i and $U_{(\mathsf{m},\mathsf{n})}$, to denote the same concept in different cases.

Normally, a pixel $U_i = U_{(\mathsf{m},\mathsf{n})}$ has four neighbors whose coordinates are given by

$$(\mathsf{m}+1,\mathsf{n}),(\mathsf{m}-1,\mathsf{n}),(\mathsf{m},\mathsf{n}+1),(\mathsf{m},\mathsf{n}-1)$$

This set of pixels, also called the 4-*neighbors* (Gonzalez and Woods, 2008) of the pixel U_i, is denoted by $N_4(U_i)$. And correspondingly, the set of the coordinates of these pixels in $N_4(U_i)$ is denoted by $\mathcal{P}_{(\mathsf{m},\mathsf{n})}$ or $\mathcal{P}_i$. Normally, for the coordinates within the digital image, both m and n are located in the ranges $[0,\mathsf{M}-1]$ and $[0,\mathsf{N}-1]$ respectively. But it is worth noting that for those pixels on the border or corner of an image, their neighbouring pixels lie outside the image. In these cases, we also use $\mathsf{m}=-1,\mathsf{M}$ and $\mathsf{n}=-1,\mathsf{N}$ to denote the coordinate outside of the image.

The intensity levels of digital images are also quantized as the equally spaced discrete values (Gonzalez and Woods, 2008). The number of the discrete intensity levels is denoted by L and it is typically an integer power of 2 ($L = 2^n$) so that

$$U_i \in [0, L-1].$$

Conventionally, when a digital image U_i has 2^n intensity levels, it is convention to refer to the image as an "n-bit image." For instance, an image with 256 possible discrete intensity levels ranging from 0 to 255 is called an 8-bit image. The grayscale images used in this thesis are all 8-bit images.

2.2 Edge Detection

2.2.1 Edge Map and Evaluation of Edge Detection Algorithm

Edge pixels are defined as pixels where the function of the image intensities U_i changes abruptly and edges are defined as sets of connected such edge pixels. As a result of edge detection, the edge information in the processed image are provided in an *edge map* denoted by $\mathcal{M}_{(\mathsf{m},\mathsf{n})}$, which is usually a binary image of the same size as the processed image. If we denote the set consisting of all the coordinates of edge pixels in a processed image as S, then an edge map of such image can be expressed as follows,

$$\mathcal{M}_{(\mathsf{m},\mathsf{n})} = \begin{cases} 1, & (\mathsf{m},\mathsf{n}) \in S \\ 0, & (\mathsf{m},\mathsf{n}) \in \urcorner S \end{cases}$$

In order to compare the performance of the different edge detection algorithms, it is important to determine the criteria by which edge detection performance can be evaluated. Canny (1986) provided a good summary on evaluating a generally formulated edge detection problem, in which he outlined the performance criteria which are widely accepted in the areas of computer vision and image processing. We list them as follows, *Good detection.* First, as many edges as possible in the original image are required to be detected in the edge image. In other words, the error rate of missing edges has to be small for the good detection. Second, as many as the edges in the edge image should truly exist in the original image, which means the error rate of wrong edges has to be small for the good detection.

Good position. The edge detected in the edge image should reflect its original position of the edge in the processed image as correctly as possible.

Generally, it is still difficult to produce a convincing quantitative results for assessing the edge detection algorithms according to the criteria stated above. Several studies have shown the problems and most of them is based on using artificial images. Normally, because the artificial images contain merely simple geometric patterns, their ground truth data of edges can be precisely specified. The evaluation of an edge detection algorithm can be carried out by comparing the resultant edge map $\mathcal{M}_{(\mathsf{m},\mathsf{n})}$ with the ground truth $\mathcal{M}_{gt(\mathsf{m},\mathsf{n})}$. In particular, in order to compare the performance of edge detection of algorithms for artificial images, we will the following four measures, the true positive tp, the rate of the true positive tp_r, the false positive fp and the rate of false positive fp_r.

$$\begin{aligned} tp &= \sum_{(\mathsf{m},\mathsf{n})\in S_{gt}} \mathcal{M}_{(\mathsf{m},\mathsf{n})}, \quad tp_r = \frac{tp}{\sum \mathcal{M}_{(\mathsf{m},\mathsf{n})}} \\ fp &= \sum_{(\mathsf{m},\mathsf{n})\in \neg S_{gt}} \mathcal{M}_{(\mathsf{m},\mathsf{n})}, \quad fp_r = \frac{fp}{\sum (1-\mathcal{M}_{gt(\mathsf{m},\mathsf{n})})} \end{aligned} \tag{2.4}$$

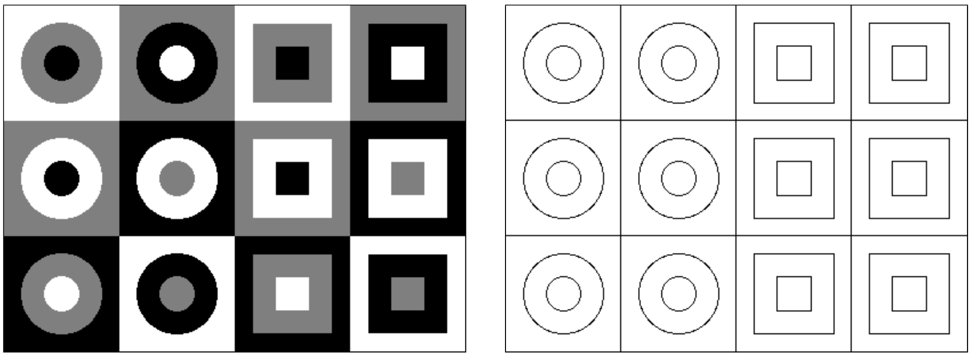
where, S_{gt} denotes the set of the coordinates of edge pixels in the ground-truth data. The true positive pixels are the ones in the ground truth image $\mathcal{M}_{gt}$ and also detected in the edge image $\mathcal{M}$. And the false positive pixels are the ones only detected in the edge image $\mathcal{M}$ but not in the ground truth image $\mathcal{M}_{gt}$.

2.3 Test Images

The images used to test the performance of the edge detection algorithm studied in this thesis include both artificial images and real images. Moreover, white noise is added to the images when a robustness test is carried out.

Artificial Images

The basic structure of the artificial image used in this thesis is provided in Figure 2.2(a). It consists of 12 groups of small square patterns [101 × 101], so the size of whole image is [303 × 404]. The intensity levels of the artificial image are 63, 127 and 191 and either group of circles or squares has all permutations of these intensity levels. The corresponding ground truth data is provided in Figure 2.2(b). The ground truth data of the edges denoted by $\mathcal{M}_{gt_{(\mathsf{m},\mathsf{n})}}$ is automatically computed by programs in Matlab.



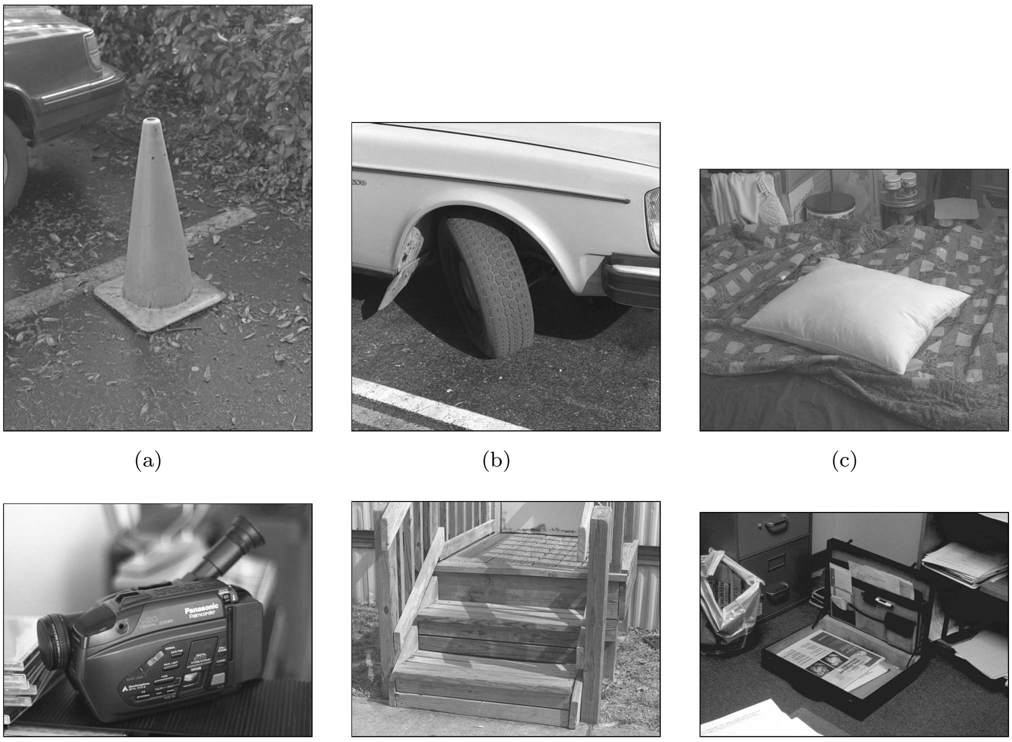
(a) Artificial images with the intensity levels 0, 127 and 255

(b) Ground-truth data of the artificial images

Figure 2.2: Artificial images and the corresponding ground truth of the edge data.

Real Images

Figure 2.3 presents all the real images used in this thesis, which are, respectively, traffic cone [604 × 437], tire [512 × 512], pillow [468 × 552], video camera [435 × 577], stairs [441 × 579] and briefcase [419 × 577]. They are all downloaded from the website “Edge Detector Comparison” based on the work done by Heath et al. (1997).Original photographs are in colours and we convert them into grayscale images with the built-in function in Matlab.



(a) (b) (c)

(d) (e) (f)

Figure 2.3: Real images used for testing algorithms of edge detection. 2.3(a) Traffic cone. 2.3(b) Tire. 2.3(c) Pillow. 2.3(d) Video camera. 2.3(e) Stairs. 2.3(f) Briefcase. All the real images are provided by Heath et al. (1997) which are available on the website “Edge Detector Comparison”.

Chapter 3

Dynamical System For Edge Detection

This chapter introduces the background theory of ordinary differential equations (ODEs) and dynamical system which are used in the following chapters. We highlight a selection of fundamental concepts of dynamical system theory. Then, we provide an example of studying a FitzHugh-Nagumo type of neuronal model in order to see how its properties are analysed by using dynamical system theory. Then, we focus on the basic model in the edge detection algorithm which is a set of continuous *reaction-diffusion* equations. We elaborate the system structure from the derivation of the discrete version of the reaction-diffusion equations and explain how the edge detection algorithm can be driven by the *excitability* of a single neuron. Finally, we provide a review of Nomura's edge detection algorithms based on the discrete reaction-diffusion equations.

The theory of dynamical system introduced in this chapter is based on the books (Perko, 1991; Izhikevich, 2006), which we recommend for further details and references.

3.1 Basics on ODEs

A general system of ordinary differential equations (ODEs) can be expressed as

$$\begin{cases} \dfrac{dx_1}{dt} = f_1(x_1, x_2, \ldots, x_n) \\ \dfrac{dx_2}{dt} = f_2(x_1, x_2, \ldots, x_n) \\ \qquad \vdots \\ \dfrac{dx_n}{dt} = f_n(x_1, x_2, \ldots, x_n) \end{cases}$$

where each component f_i is assumed to have continuous mixed partial derivatives with respect to all x_j up to a sufficiently high order. Introduce a state vector

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n$$

and define a map $f : \mathbb{R}^n \to \mathbb{R}^n$ by

$$x \mapsto f(x) = \begin{pmatrix} f_1(x_1, x_2, \ldots, x_n) \\ f_2(x_1, x_2, \ldots, x_n) \\ \vdots \\ f_n(x_1, x_2, \ldots, x_n) \end{pmatrix}.$$

Then the system can be written as

$$\dot{x} \equiv \frac{dx}{dt} = f(x), \quad x \in \mathbb{R}^n. \tag{3.1}$$

Denote a solution to Equation 3.1 with an initial condition $x_0 \in \mathbb{R}^n$ by $x(t, x_0)$. General theory of smooth differential equations (Perko, 1991) ensures the existence and uniqueness of such solution $x(t)$ for any $x_0 \in \mathbb{R}^n$.

Nullclines and Equilibria of Dynamical Systems

In order to analyse the behaviour of the dynamical system in Equation 3.1, one would find its nullclines and equilibria. Each nullcline of the dynamical system can be determined by

$$f_i(x) = 0.$$

And a point $\overline{x} \in \mathbb{R}^n$ is an *equilibrium* of the dynamical system if

$$f(\overline{x}) = 0.$$

For all the points on the specific nullclines $f_i(0) = 0$, the derivative of the specific system state $\dot{x}_i$ is zero. And at an equilibrium where $f(\overline{x}) = 0$, all derivatives $\dot{x} = 0$. So, any solutions $x(t, \overline{x})$ starting at the equilibrium $\overline{x}$ consist of only one point, namely the equilibrium $\overline{x}$.

The local stability of the equilibrium $\overline{x}$ is determined by the *Jacobian matrix*

$$J = D_x F = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$$

evaluated at $\overline{x}$. If we use λ_i to denote the ith eigenvalue of the Jacobian $J|_{\overline{x}}$,

$$J|_{\overline{x}} \mathbf{v}_i = \lambda_i \mathbf{v}_i \tag{3.2}$$

where $\mathbf{v}_i$ is the eigenvector corresponding to λ_i. A dynamical system is said to be *locally stable* at an equilibrium $\overline{x}$ if all the eigenvalues λ_i of the Jacobian matrix evaluated at the equilibrium $J|_{\overline{x}}$ have negative real part

$$\mathrm{Re}(\lambda_i) < 0.$$

Jacobian matrix J is said to be *hyperbolic* when none of its eigenvalues λ_i has zero real part, and to be *nonhyperbolic* when at least one of its eigenvalues λ_i has zero real part.

Bifurcation

Sometimes, one would be interested in how the qualitative behaviour of system in Equation 3.1 changes when the system dynamics defined by f changes with respect to system parameters μ. Consider a dynamical system

$$\dot{x} = f(x, \mu), \quad x \in \mathbb{R}^n,$$

where $\mu = (a_1, a_2, \cdots, a_m) \in \mathbb{R}^m$ denotes the system parameters. A parameter value μ_0 is said to be *regular*, if for any μ belongs to an open neighbourhood S of μ_0, the qualitative behaviour of dynamical system remains the same. In contrast, a parameter value μ_0 is said to be a *bifurcation value*, if there exist some value μ_1 in any neighbourhood S of μ_0 such that the qualitative behaviour of dynamical system is different for μ_0 and μ_1. This thesis deals with only two types of bifurcations which occur in dynamical systems; namely, *saddle-node* bifurcation and *Hopf* bifurcation. These two types of bifurcations are the simplest cases which occur at a nonhyperbolic equilibrium; the Hopf type of bifurcation occurs when the eigenvalues of J has a pair of purely imaginary eigenvalues. Saddle node bifurcation appears when one (just one) eigenvalue of J is zero.

3.2 FitzHugh-Nagumo Model

Generally, the FitzHugh-Nagumo model (FitzHugh, 1961; Nagumo et al., 1962) is a simplified model of neural dynamics, able to fire periodic spikes. However, for the parameter settings adopted here, the FitzHugh-Nagumo model is not an oscillator and can only fire a single action potential instead of periodic ones. The FitzHugh-Nagumo model which can be expressed as the following two-variable ordinary differential equations:

$$\begin{cases} \dot{v} = \frac{1}{\epsilon}[v(1-v)(v-a)-w] & = -\frac{1}{\epsilon}[v^3-(a+1)v^2+av+w] \\ \dot{w} = v - bw & = v - bw \end{cases} \tag{3.3}$$

where $x = (v, w)^T \in \mathbb{R}^2$ is the vector of state variables. Specifically in a neuroscience context (Ermentrout and Terman, 2010), v denotes the *membrane potential* and w is the *gating variable*. $\epsilon \ll 1$, $0 < a < 1$ and $b > 0$ are all system parameters. Such a two-dimensional dynamical system which is also called *planar* system is usually analysed by plotting its *phase portrait* in the state space $\mathbb{R}^2$.

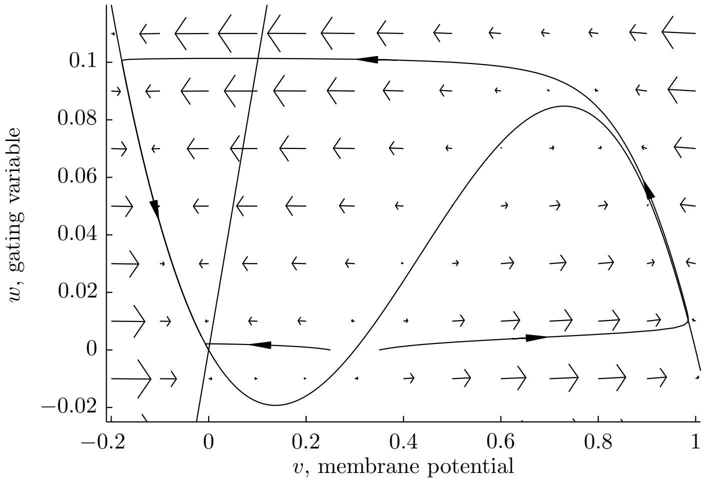


Figure 3.1: Phase portrait of FitzHugh-Nagumo model for $a = 0.3$, $b = 1$ and $\epsilon = 0.001$. The initial conditions of two illustrative solutions are $(0.25, 0)$ and $0.35, 0$.

Figure 3.1 provides an example of phase portrait of the FitzHugh-Nagumo system in Equation 3.3. The whole plane which is also called *phase* plane denotes the state space $\mathbb{R}^2$. For any point (v, w) on the phase plane, the vector $(f_1(v, w), f_2(v, w))$, where

$$f_1(v, w) = \frac{1}{\epsilon}[v(1-v)(v-a)-w],$$
$$f_2(v, w) = v - bw,$$

indicates the direction of change of the point. So the vectorised function f in Equation 3.1 is also referred to as the *vector field*. As introduced in Section 3.1, a *nullcline* refers to a set of points in state space of which the derivatives with respect to one of the state variables are all zeros. For the two-variable dynamical system in Equation 3.3, the two nullclines are two curves in the phase plane as the two black solid lines shown in Figure 3.1. The nullclines of the FitzHugh-Nagumo system defined by $\dot{v} = 0$ (v-nullcline) and $\dot{w} = 0$ (w-nullcline) can be expressed as

$$\begin{cases} w = v(1-v)(v-a) \\ w = \dfrac{1}{b}v \end{cases} \tag{3.4}$$

The v-nullcline of the system is a cubic curve and the w-nullcline is a straight line. Their intersection at the origin $(0,0)$ is an equilibrium. We also present two illustrative solutions with different initial conditions as two trajectories in Figure 3.1. The equilibrium $(0,0)$ is stable as these two different trajectories finally converge to it.

The general form of Jacobian of the FitzHugh-Nagumo model in Equation 3.3 is

$$J = \begin{pmatrix} -\frac{3}{\epsilon}v^2 + \frac{2(a+1)}{\epsilon}v - \frac{a}{\epsilon} & -\frac{1}{\epsilon} \\ 1 & -b \end{pmatrix} \tag{3.5}$$

The specific Jacobian matrix of the equilibrium at the origin is

$$\begin{aligned} J &= \begin{pmatrix} -\dfrac{1}{\epsilon}(3v^2 - 2(a+1)v + a) & -\dfrac{1}{\epsilon} \\ 1 & -b \end{pmatrix}\Bigg|_{(v,w)=(0,0)} \\ &= \begin{pmatrix} -\dfrac{a}{\epsilon} & -\dfrac{1}{\epsilon} \\ 1 & -b \end{pmatrix} \end{aligned} \tag{3.6}$$

where the *trace* τ and the *determinant* Δ

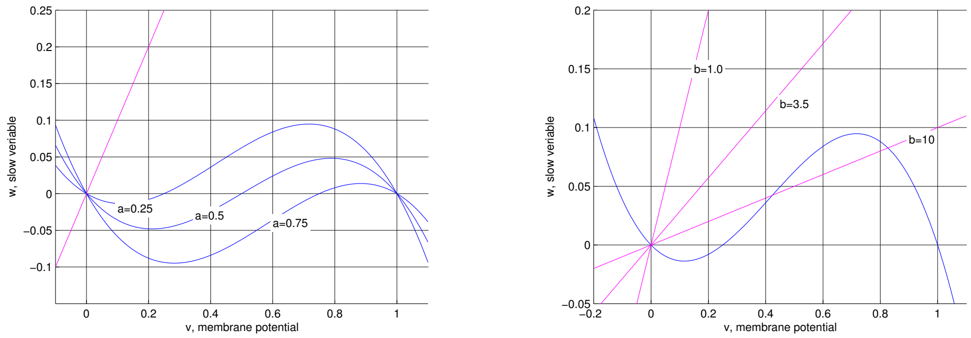
$$\begin{aligned} \tau &= -\frac{a}{\epsilon} - b \\ \Delta &= \frac{1}{\epsilon}(ab+1) \end{aligned}$$

And the eigenvalues of J can be obtained by solving the *characteristic equation*

$$\begin{vmatrix} -\dfrac{a}{\epsilon} - \lambda & -\dfrac{1}{\epsilon} \\ 1 & -b-\lambda \end{vmatrix} = 0$$

which can also be written in the polynomial form

$$\lambda^2 - \tau\lambda + \Delta = 0 \tag{3.7}$$



(a) Nullclines of FitzHugh-Nagumo models when $a = 0.25$, $a = 0.50$ and $a = 0.75$

(b) Nullclines of FitzHugh-Nagumo models when $b = 1.0$, $b = 3.5$ and $b = 10.0$

Figure 3.2: Nullclines of FitzHugh-Nagumo models for different values of the parameters a and b

And the two solutions of this characteristic equation are

$$\lambda_1 = \frac{-(\frac{a}{\epsilon}+b)+\sqrt{(\frac{a}{\epsilon}-b)^2-4}}{2}, \quad \lambda_2 = \frac{-(\frac{a}{\epsilon}+b)-\sqrt{(\frac{a}{\epsilon}-b)^2-4}}{2} \tag{3.8}$$

As mentioned above, the parameters a, b and ϵ are all greater than 0. If $(a/\epsilon - b)^2 - 4 < 0$, λ_1 and λ_2 are complex conjugate and they both have the negative real parts. If $(a/\epsilon - b)^2 - 4 > 0$, then it is easy to check that λ_2 is smaller than 0. While

$$\begin{aligned}
\lambda_1 &= \frac{-(\frac{a}{\epsilon}+b)+\sqrt{(\frac{a}{\epsilon}-b)^2-4}}{2} \\
&= \frac{-(\frac{a}{\epsilon}+b)+\sqrt{(\frac{a}{\epsilon}+b)^2-\frac{4ab}{\epsilon}-4}}{2} \\
&< \frac{-(\frac{a}{\epsilon}+b)+\sqrt{(\frac{a}{\epsilon}+b)^2}}{2} \\
&= 0
\end{aligned}$$

So, the equilibrium of the FitzHugh-Nagumo system in Equation 3.3 at the origin is a stable *node* within the given condition for the parameters $a > 0$, $b > 0$ and $\epsilon > 0$.

As shown in Figure 3.2, one can imagine that changing the values of system parameters a and b will also change the positions of these two nullclines. In particular, the parameter a determines the location of the intersection $(a, 0)$ of the v-nullcline and v-axis between 0 and 1 on the v-axis, and thereby, influences both values of two extrema (one is minimum and the other is maximum) near it. And the value of parameter $1/b$ determines the slope of w-nullcline so that it may have one, two or three intersections (equilibria) with the v-nullcline as shown in Figure 3.2(b). So that parameters a and b control the numbers of intersections of the two nullclines, namely the equilibria. And a straightforward calculation shows that the system has origin as its only fixed point if and only if

$$(a-1)^2 - \frac{4}{b} < 0 \tag{3.9}$$

or two fixed points, origin and $(\frac{a+1}{2}, \frac{a+1}{2b})$, if $(a-1)^2 - \frac{4}{b} = 0$ or three fixed points, origin and $(v_\pm, w_\pm)$, where

$$v_\pm = \frac{a+1 \pm \sqrt{(a-1)^2 - \frac{4}{b}}}{2}, \quad w_\pm = \frac{v_\pm}{b} \tag{3.10}$$

So either increasing b with a fixed or reducing a with b fixed will lead to the generation of two equilibria other than the origin.

Consider the situation where the system (3.3) has only two equilibria, i.e. $v = \frac{a+1}{2}$ and $b = \frac{4}{(a-1)^2}$, the Jacobian becomes

$$J = \begin{pmatrix} \frac{1}{\epsilon} \cdot \frac{(a-1)^2}{4} & -\frac{1}{\epsilon} \\ 1 & -\frac{4}{(a-1)^2} \end{pmatrix}$$

The *trace* τ and the *determinant* Δ of the Jacobian are respectively,

$$\begin{aligned} \tau &= \frac{1}{\epsilon} \cdot \frac{(a-1)^2}{4} - \frac{4}{(a-1)^2} \\ \Delta &= 0 \end{aligned} \tag{3.11}$$

and the eigenvalues are,

$$\lambda_1 = -\tau, \qquad \lambda_2 = 0;$$

So the system undergoes a saddle-node bifurcation.

The middle intersection of two nullclines (v_-, w_-) is always saddle with the given condition for a, b and ϵ. It can be figured out by checking its Jacobian matrix

$$J|_{(v_-,w_-)} = \begin{pmatrix} \frac{1}{\epsilon}(a + \frac{2}{b}) & -\frac{1}{\epsilon} \\ 1 & -b \end{pmatrix}$$

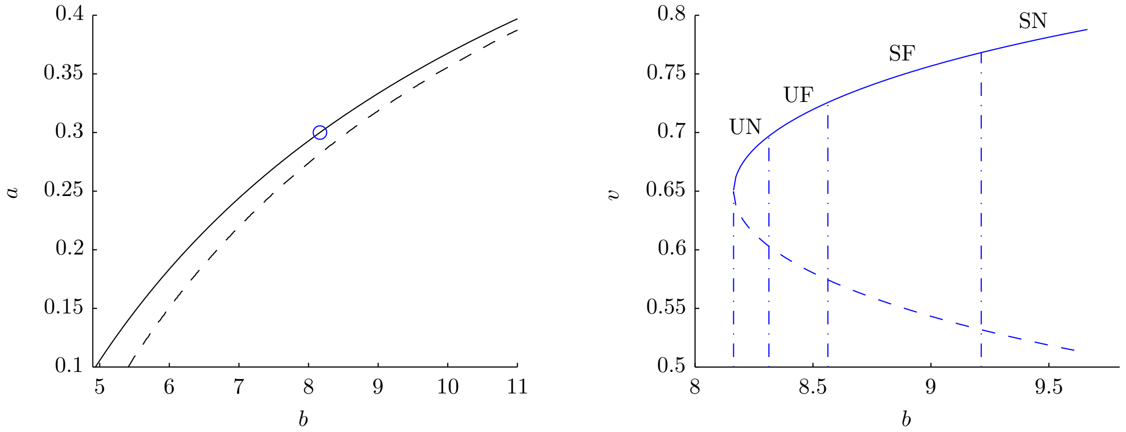
The corresponding

$$\Delta|_{(v_-,w_-)} = -\frac{1}{\epsilon}(ab+1) < 0$$

Hence, it is easy to check that

$$\begin{aligned} \lambda_1|_{(v_-,w_-)} &= \frac{\tau + \sqrt{\tau^2 - 4\Delta}}{2} > \frac{\tau + \sqrt{\tau^2}}{2} \geqslant 0 \\ \lambda_2|_{(v_-,w_-)} &= \frac{\tau - \sqrt{\tau^2 - 4\Delta}}{2} < \frac{\tau - \sqrt{\tau^2}}{2} \leqslant 0 \end{aligned}$$

However, the stability of upper equilibrium (v_+, w_+) changes according to the different values of system parameters. In order to track the stability of (v_+, w_+) after the saddle-node bifurcation, we look into a concrete example where we fix $a = 0.3$ and sweep the value of b. The results are provided in Figure 3.3(b) below.



(a) Boundary of Saddle node bifurcation (solid line) and Hopf bifurcation (dashed line).

(b) Bifurcation diagram of single FitzHugh-Nagumo model with $a = 0.3$.

Figure 3.3: Bifurcation structure of single FitzHugh-Nagumo model in Equation 3.3. (b) illustrates how a newly appearing equilibrium (v_+, w_+) changes after saddle-node bifurcation for a specific value of $a = 3$ as circled in (a).UN: unstable node, UF: unstable focus, SF: stable focus, SN: stable node.

The solid and the dashed lines respectively indicates v_+ and v_- changing along the parameter b. It is clear to see that there is a bifurcation at $b = 8.16$ which we have shown is a saddle-node bifurcation. Figure 3.3(b) shows that the upper equilibrium (v_+, w_+) is generally divided into four different partitions when b continues to increase after the bifurcation point $b = 8.16$. These four partitions illustrate four different stability states of the upper equilibrium (v_+, w_+). It can be clearly seen from Figure 3.3(b) that (v_+, w_+) becomes stable if $b > 8.56$; the vertical dashed line $b = 8.56$ in Figure 3.3(b) gives the stability boundary of the upper equilibrium (v_+, w_+). A direct calculation of J at (v_+, w_+) when $a = 0.3$ and $b = 8.56$ shows that

$$\lambda_{1,2} \approx \pm 0.3058i$$

which are a pair of purely imaginary values. So, $(a, b) = (0.3, 8.56)$ with the given value $\epsilon = 0.001$ is a point for Hopf bifurcation. Indeed, for each value of $0 < a < 1$, there will be two values of the parameter b, b_{SN} and b_H, respectively denoting the saddle-node and Hopf bifurcations. And Figure 3.3(a) provides these two sets of bifurcation points of (a, b) in the a-b plane. The analytical expression of the solid curve (saddle-node bifurcation) is given by

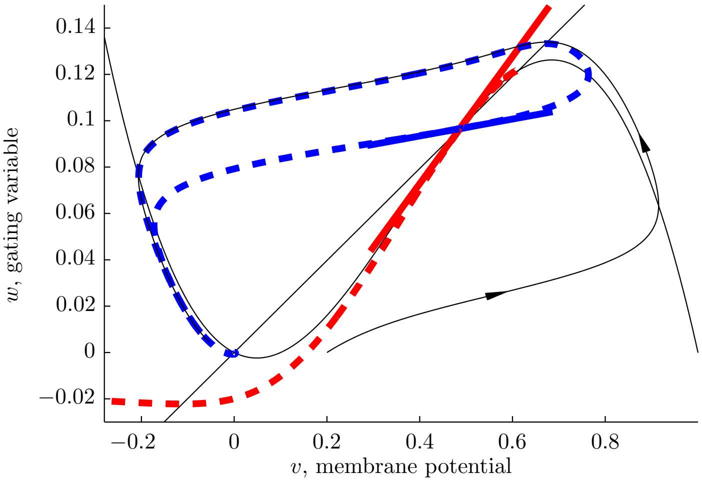
$$b = \frac{4}{(a-1)^2}$$

and the dash line (Hopf bifurcation) denotes a *traceless* Jacobian $J(v_+)$. Hence,

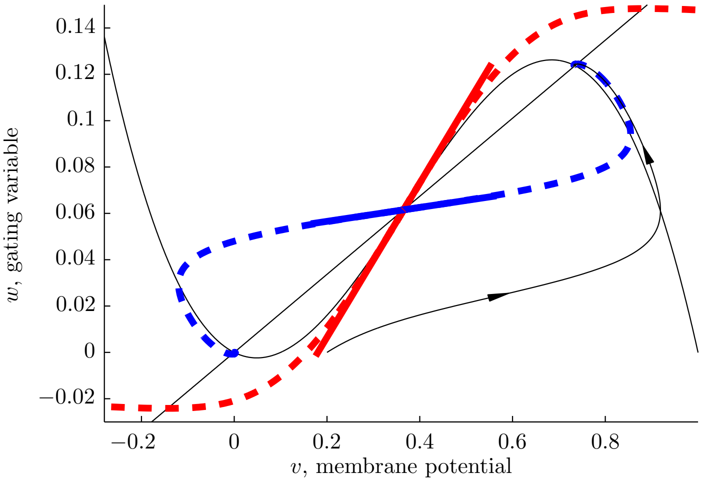
$$-\frac{1}{\epsilon}(3v_+^2 - 2(a+1)v_+ + a) - b = 0 \tag{3.12}$$

Substituting (3.10) into (3.12) gives

$$(a-1)^2 - \frac{6}{b} + (a+1)\sqrt{(a-1)^2 - \frac{4}{b}} + 2\epsilon b = 0$$



(a) Phase portrait of FitzHugh-Nagumo system with an unstable upper equilibrium



(b) Phase portrait of FitzHugh-Nagumo system with a stable upper equilibrium

Figure 3.4: Phase portrait of FitzHugh-Nagumo system with three equilibria

Figure 3.4 provides another two phase portraits of the FitzHugh-Nagumo model which have three equilibria. Specifically, the upper equilibrium in Figure 3.4(a) is unstable while the one in Figure 3.4(a) is stable. The red or blue solid line in both figures denotes the stable or unstable eigenspace of the saddle (v_-, w_-). And the red or blue dashed curves are corresponding the stable or unstable manifolds. In Figure 3.4(b), it can be clear to see that the connected two stable manifolds of the saddle (v_-, w_-) form a separatrix which divides the whole phase space $\mathbb{R}^2$ into two parts. So, such a planar model in Figure 3.4(b) is also called *bistable* system.

3.3 The Model of Edge Detection

The most basic model for the purpose of edge detection is a two component reaction-diffusion system described by a pair of parabolic partial differential equations, as follows:

$$\begin{cases} \dfrac{\partial v}{\partial t} = f(v, w, a) + D_v \nabla^2 v \\ \dfrac{\partial w}{\partial t} = g(v, w) + D_w \nabla^2 w \end{cases} \tag{3.13}$$

Again, v and w are the state variables defined in space $(\mathsf{x}, \mathsf{y}) \in \mathbb{R}^2$ and at time $t \in \mathbb{R}$. We shall follow the convention in a neural context (Ermentrout and Terman, 2010) using v and w, respectively for the *membrane potential* and the *channel gating variable* of a neuron, as previously introduced in Section 3.2. The time t is implicitly included. $f(v, w)$ and $g(v, w)$ are the reaction terms while $D_v \nabla^2 v$ and $D_w \nabla^2 w$ are the diffusion terms,where D_v and D_w are two diffusive coefficients respectively for each of the state variables and ∇^2 is the Laplacian operator,

$$\nabla^2 = \frac{\partial^2}{\partial \mathsf{x}^2} + \frac{\partial^2}{\partial \mathsf{y}^2}$$

In the application of image processing, Nomura et al. (2011a) choose the FitzHugh Nagumo model in Equation 3.3 as the reaction term in the reaction diffusion system. In their algorithm, v, indeed $v(\mathsf{x}, \mathsf{y}, t)$, is used to denote the time evolution of the spatial distribution of rescaled image intensities. Specifically, the initial condition $v(\mathsf{x}, \mathsf{y}, 0)$ takes the rescaled intensity distribution $U^r(\mathsf{x}, \mathsf{y})$ of the original image (unrescaled intensity distribution is denoted as $U(\mathsf{x}, \mathsf{y})$). However, $w(\mathsf{x}, \mathsf{y}, t)$, which is initially set as zero, does not have a specific meaning in image description.

In order to understand the principle of the edge detection algorithm based on the model in Equation 3.13, we first discretise the reaction diffusion system in Equation 3.13 according to the input of digital image. It will be seen that the reaction diffusion system after discretisation is equivalent to a neural network, where a single network element is a FitzHugh-Nagumo type of neuron and the diffusion terms play the same role as the nearest neighbour coupling. Second, we focus on the pure diffusion terms, namely a single FitzHugh-Nagumo model, to get the basic idea of edge detection mechanisms inspired by its system dynamics. Finally, we provide a review of all the algorithms proposed by Nomura and his colleagues (Nomura et al., 2003, 2008, 2011b,a; Kurata et al., 2008) followed by a brief evaluative conclusion on the latest one proposed by them.

3.3.1 From Reaction Diffusion System to Coupled Neuronal Network

For the discretisation of the reaction diffusion system in Equation 3.13, Nomura et al. (2008) originally used the Crank-Nicolson scheme (Thomas, 1995), which is a relatively advanced method to processing partial differential equations. Here, we use the most basic discretisation scheme and our main purpose of this section is to show that the discretised diffusion terms in Equation 3.13 are equivalent to the nearest neighbour couplings of the network. Because both the v and w variables will be treated similarly in the discretisation, we only take v variable as an example here. Recall the definitions of the first and second partial derivatives,

$$\begin{aligned}
\frac{\partial v}{\partial \mathsf{x}} &= \lim_{\Delta \mathsf{h}\to 0} \frac{v(\mathsf{x}+\Delta\mathsf{h},\mathsf{y}) - v(\mathsf{x},\mathsf{y})}{\Delta\mathsf{h}} \\
\frac{\partial v}{\partial \mathsf{y}} &= \lim_{\Delta \mathsf{h}\to 0} \frac{v(\mathsf{x},\mathsf{y}+\Delta\mathsf{h}) - v(\mathsf{x},\mathsf{y})}{\Delta\mathsf{h}} \\
\frac{\partial^2 v}{\partial \mathsf{x}^2} &= \lim_{\Delta \mathsf{h}\to 0} \frac{\frac{v(\mathsf{x}+\Delta\mathsf{h},\mathsf{y}) - v(\mathsf{x},\mathsf{y})}{\Delta\mathsf{h}} - \frac{v(\mathsf{x},\mathsf{y}) - v(\mathsf{x}-\Delta\mathsf{h},\mathsf{y})}{\Delta\mathsf{h}}}{\Delta\mathsf{h}} \\
\frac{\partial^2 v}{\partial \mathsf{y}^2} &= \lim_{\Delta \mathsf{h}\to 0} \frac{\frac{v(\mathsf{x},\mathsf{y}+\Delta\mathsf{h}) - v(\mathsf{x},\mathsf{y})}{\Delta\mathsf{h}} - \frac{v(\mathsf{x},\mathsf{y}) - v(\mathsf{x},\mathsf{y}-\Delta\mathsf{h})}{\Delta\mathsf{h}}}{\Delta\mathsf{h}}
\end{aligned}$$

Since the initial value of $v(\mathsf{x},\mathsf{y},t)$ is determined by the intensity distribution $U(\mathsf{x},\mathsf{y})$ of an image, we discretise the reaction diffusion system following the digital image representation provided in Figure 2.1 for the numerical solutions. Hence, we choose the finite differences in the processed image and the reaction diffusion systems to be the same finite difference, $\Delta\mathsf{h}$. So, the reaction diffusion system in Equation 3.13 will be discretised to a spatial grid of the same dimension as the processed image. We rewrite the partial derivatives in the discrete form by changing to the symbols listed below,

$$\frac{\partial^2 v}{\partial \mathsf{x}^2} \to \Delta_{\mathsf{xx}} v, \quad \frac{\partial^2 v}{\partial \mathsf{y}^2} \to \Delta_{\mathsf{yy}} v$$

$$v(\mathsf{x},\mathsf{y})|_{(\mathsf{x}=\mathsf{m}\cdot\Delta\mathsf{h},\ \mathsf{y}=\mathsf{n}\cdot\Delta\mathsf{h})} \to v_{(\mathsf{m},\mathsf{n})}$$

where, $\Delta_{\mathsf{xx}} v$ and $\Delta_{\mathsf{yy}} v$ are the second order differences of state variable v with respect to x and y respectively.

Hence, the second partial derivatives of state variable $v(\mathsf{x},\mathsf{y})$ with respect to x and y can be written in the following discrete forms,

$$\begin{aligned}
D_v\nabla^2 v(\mathsf{x},\mathsf{y}) &= D_v(\Delta_{\mathsf{xx}} v_{(\mathsf{m},\mathsf{n})} + \Delta_{\mathsf{yy}} v_{(\mathsf{m},\mathsf{n})}) \\
&= D_v(\frac{v_{(\mathsf{m}-1,\mathsf{n})} - 2v_{(\mathsf{m},\mathsf{n})} + v_{(\mathsf{m}+1,\mathsf{n})}}{\Delta\mathsf{h}^2} + \frac{v_{(\mathsf{m},\mathsf{n}-1)} - 2v_{(\mathsf{m},\mathsf{n})} + v_{(\mathsf{m},\mathsf{n}+1)}}{\Delta\mathsf{h}^2}) \\
&= \frac{D_v}{\Delta\mathsf{h}^2}((v_{(\mathsf{m}-1,\mathsf{n})} - v_{(\mathsf{m},\mathsf{n})}) + (v_{(\mathsf{m}+1,\mathsf{n})} - v_{(\mathsf{m},\mathsf{n})}) + \\
&\qquad (v_{(\mathsf{m},\mathsf{n}-1)} - v_{(\mathsf{m},\mathsf{n})}) + (v_{(\mathsf{m},\mathsf{n}+1)} - v_{(\mathsf{m},\mathsf{n})})) \\
&= k_v \sum_{(\mathsf{m}',\mathsf{n}')\in\mathcal{P}_{(\mathsf{m},\mathsf{n})}} (v_{(\mathsf{m}',\mathsf{n}')} - v_{(\mathsf{m},\mathsf{n})})
\end{aligned} \tag{3.14}$$

where, $k_v = \frac{D_v}{\Delta\mathsf{h}^2}$ is a constant and $\mathcal{P}_{(\mathsf{m},\mathsf{n})}$, as mentioned in Section 2.1, is the set of consisting of the coordinates in the neighborhood of (m,n).

$$\mathcal{P}_{(\mathsf{m},\mathsf{n})} = \{(\mathsf{m}-1,\mathsf{n}), (\mathsf{m}+1,\mathsf{n}), (\mathsf{m},\mathsf{n}-1), (\mathsf{m},\mathsf{n}+1)\}$$

Note that for the pixels at the boundaries or corners, we impose boundary condition:

$$\begin{aligned}
&v(-1,\ \mathsf{n}) = v(0,\ \mathsf{n}), \quad v(\mathsf{M},\ \mathsf{n}) = v(\mathsf{M}-1,\ \mathsf{n}), \\
&v(\mathsf{m},\ -1) = v(\mathsf{m},\ 0), \quad v(\mathsf{m},\ \mathsf{N}-1) = v(\mathsf{m},\ \mathsf{N})
\end{aligned} \tag{3.15}$$

Hence, there are no outer pixels contributing to the coupling. Specifically, the neurons at the edge of the network will have three neighbour neurons contributing to its coupling while the ones at the corner have only two. So, the coupling term (3.14) in state variable v can be further written as,

$$k_v \sum_{j\in\mathcal{P}_i} (v_j - v_i) \tag{3.16}$$

As illustrated above, the diffusion coupling in w can also be discretised following the same derivation. So the original model can be expressed as the ordinary differential equations of its single element,

$$\begin{cases}
\dot{v}_i = f(v_i, w_i, a_i) + k_v \sum_{j\in\mathcal{P}_i} (v_j - v_i) \\
\dot{w}_i = g(v_i, w_i) + k_w \sum_{j\in\mathcal{P}_i} (w_j - w_i)
\end{cases} \tag{3.17}$$

where, $k_v = \frac{D_v}{\Delta\mathsf{h}^2}$ and $k_w = \frac{D_w}{\Delta\mathsf{h}^2}$ are all the constants denoting the coupling strength of the network. In the original work done by Nomura et al. (2011a), the coupling strength does not change among the network. $\mathcal{P}_i = f_c(\mathcal{P}_{(\mathsf{m},\mathsf{n})})$ is a set consist of all the sequence numbers of the elements adjacent to the i-th element in the network.

So finally, Figure 3.5 shows the basic architecture of the neuronal network. The circles represent the basic computing elements i.e. the cells of the network. And the lines wiring the circles represent the coupling among the cells. It can clearly seen that each neuron is only connected to its 4-*neighbors* so that this kind of coupling is called the *nearest neighbour coupling*. The cell matrix consists of M rows and N columns which is of the same dimension of the input images. So it is a one-to-one structure that one cell within the network will relate to one pixel of the input image.

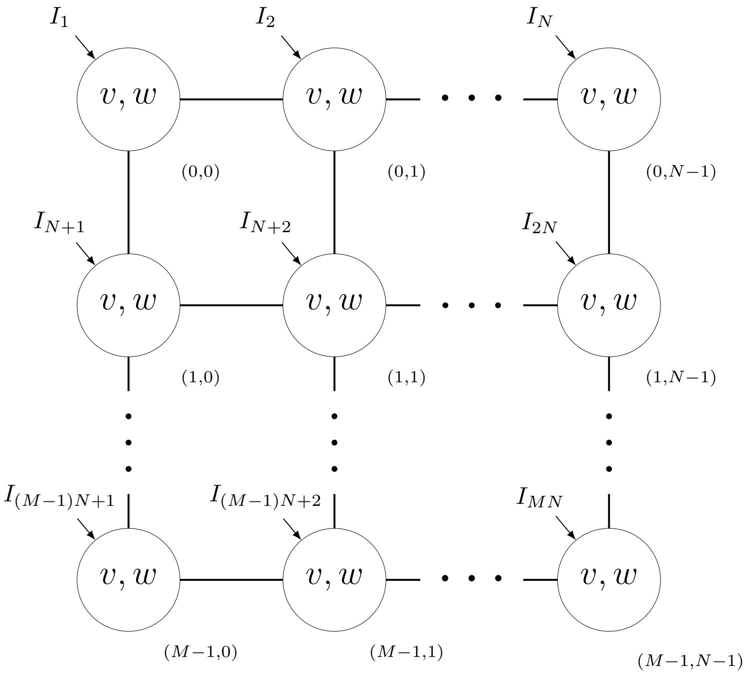


Figure 3.5: Network structure of the system

3.3.2 Excitability and Fast Slow Dynamics

The detailed dynamics of the FitzHugh-Nagumo model is previously presented in Section 3.2. Here, we focus on the neural properties of such a two-dimensional system which is claimed as the inspiration of the edge detection algorithm (Nomura et al., 2003). We recall the Equation 3.3 again as follows,

$$\begin{aligned} \dot{v} = \frac{1}{\epsilon} f(v,w) &= \frac{1}{\epsilon}(v(1-v)(v-a)-w) \\ \dot{w} = g(v,w) &= v - bw \end{aligned} \tag{3.18}$$

with the ranges of the system parameters a,b and ϵ

$$0 < a < 1, \quad b > 0, \quad \epsilon \ll 1$$

Nomura et al. (2008) claimed that the FitzHugh-Nagumo model without any diffusion terms must be a monostable system for the purpose of edge detection. So, we also need the condition below,

$$b < \frac{4}{(a-1)^2}$$

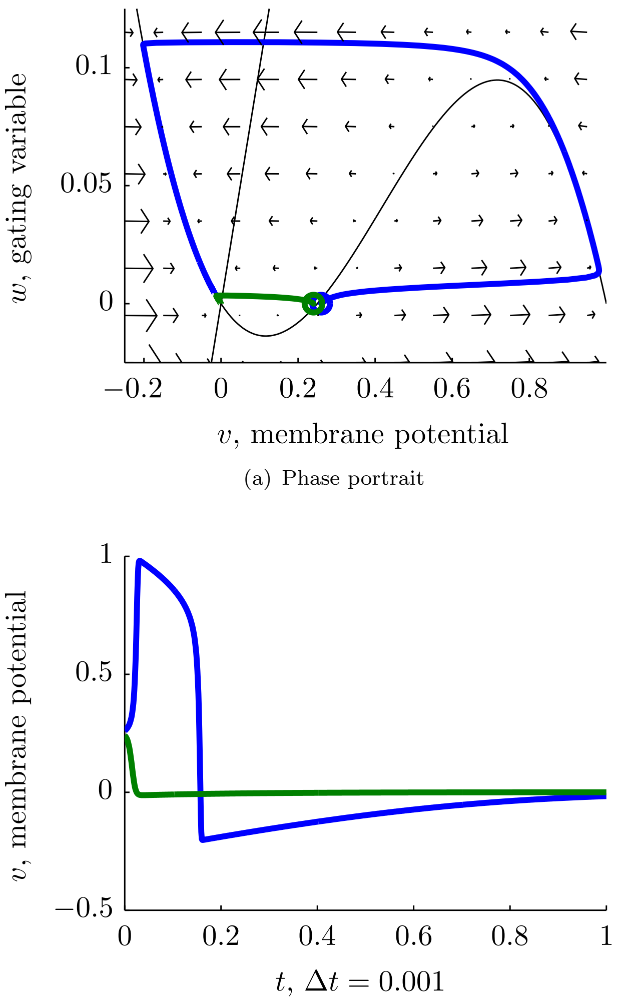
We have already presented in Section 3.2 that the origin $(0, 0)$ is always a stable equilibrium with the system parameters under these conditions. So that the solutions with any initial conditions will converge to it. In neuroscience, this origin is also called the *resting state* of a neuron. Here, as an example, we take illustrative solutions with different two initial conditions $(v_1(0), w_1(0)) = (0.24, 0)$ and $(v_2(0), w_2(0)) = (0.26, 0)$ with the parameter settings $a = 0.25$, $b = 1$ and $\epsilon = 0.001$. So, both solutions start at the v-axis and

$$v_1(0) < a, \quad v_2(0) > a$$

These two solutions are both plotted in the phase portrait and the time domain (see Figure 3.6). The solution trajectory $(v_1(t), w_1(t))$ directly goes to the equilibrium $(0, 0)$ while the other one $(v_2(t), w_2(t))$ with the initial condition $v_2(0)$ slightly greater than a goes along another way making a big loop then back to $(0, 0)$. Correspondingly, for the $v(t)$ curves in time domain as shown in Figure 3.6, $v_1(t)$ directly shrinks to zero, whereas, $v_2(t)$ goes up to nearly 1 then rapidly down to a negative value with a gradual approach to v axis, exhibiting a pulse in time. Recalling that v denotes the membrane potential of a neuron, such a sudden change of v in time which is thereby referred to as *action potential* or *spike* in a neuroscience context.

It can be clearly seen that the parameter a for each neuron works as a “threshold” to divide the solution trajectories starting on the v-axis generally into two different states, directly being back to the origin (*resting state*) or generating a spike (*excited state*), even though they will both go to the stable origin in the end. This property is often referred to as *excitability*. Although later in Chapter 5, we will show that a is not accurate as a threshold of these two states for the monostable FitzHugh-Nagumo model and simply treating this as a threshold leads to some problems, the idea of excitability forms the fundamentals of the edge detection method (Nomura et al., 2003, 2008). For a simple case of edge in processed image where two areas of different intensities U_A and U_B are adjacent, one can image that these two areas will be divided as two different states as shown in Figure 3.6. So within either of the two areas, the difference among the pixels is very small and will be extremely insensitive for the coupling terms in Equation 3.16. And thereby the coupled system will behave exactly like the uncoupled one. However, for the pixels at the boundary between two areas, the differences are huge and they will be picked out by including the diffusion terms, namely the *nearest neighbour coupling*. So, only the neurons which are located at the boundary connecting the higher or lower image intensity levels will keep a high membrane voltage level stably as time proceeds.

The constant ϵ is chosen as a small value to ensure the fast-slow dynamics. Figure 3.7 provides three phase portraits of monostable FitzHugh-Nagumo model with different values of the parameter ϵ as well as the solutions $v(t)$ and $w(t)$) in time domain. As



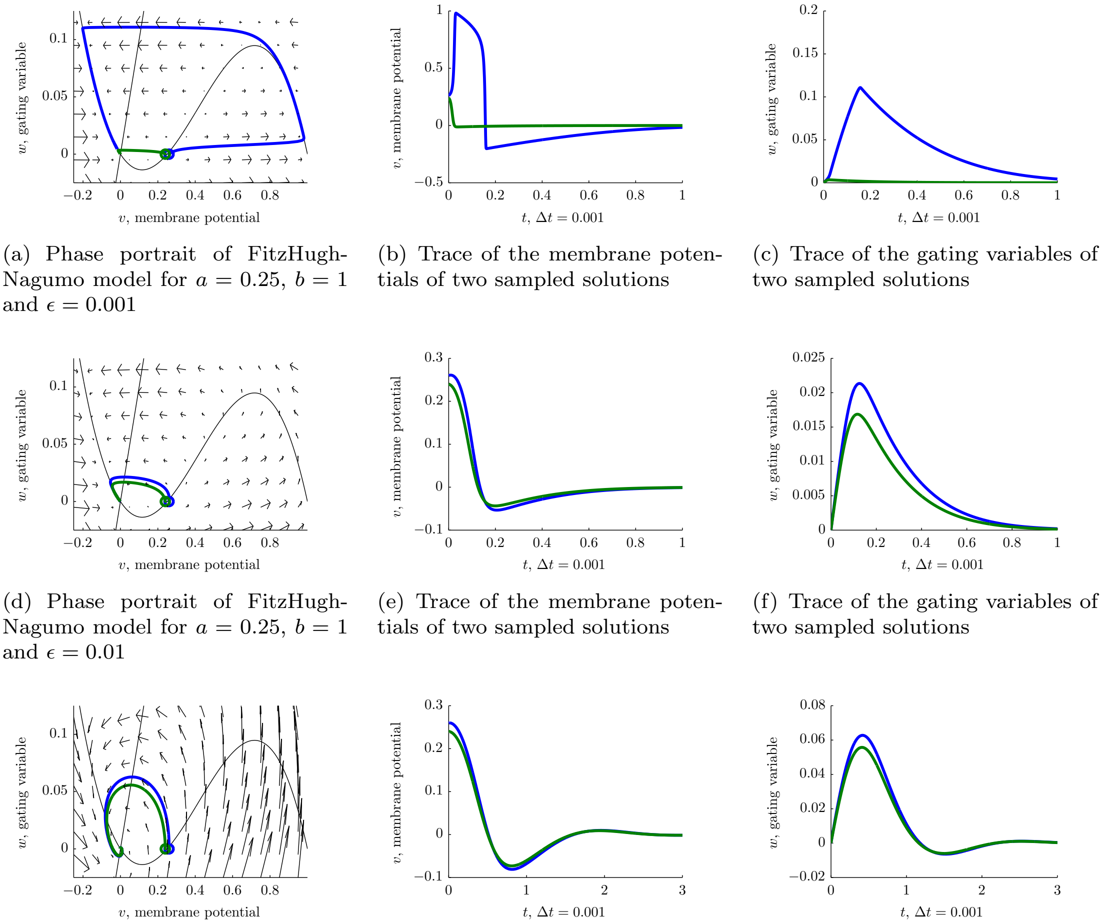
(a) Phase portrait

(b) Curves in time domain

Figure 3.6: Excitability of single FitzHugh-Nagumo model where the system parameters are fixed at $a = 0.25$, $b = 1$ and $\epsilon = 0.001$. The initial conditions of two illustrative solutions are respectively $(0.24, 0)$ and $(0.26, 0)$

shown in Figure 3.7, for $\epsilon = 0.001$ the state variables (v, w) as a point in the phase portrait will travel much slower along the w-axis, namely up or down, than along the v-axis, namely left and right; Most of the vectors in the field are nearly parallel to the v axis in the phase plot, see Figure 3.7(a). As a result, the two trajectories are clearly separated in Figure 3.7(a). However, the boundary of excited state and resting state are not clear in Figure 3.7(d) and Figure 3.7(g); the state variable (v, w) are all going back to zero towards the same direction and along the similar solution trajectories.

In order to study the firing property of a single neuron, it is generally accepted by many neuroscientists, e.g. (Ermentrout and Terman, 2010) to simply classify the spiking neurons to be *excitable* or *oscillatory*, according to whether they can fire by themselves without any input. So, we call the individual model used here as the excitable cell, because it cannot generate any spikes intrinsically, while in fact, setting the initial conditions such as Equation 3.19 for each neuron is equivalent (Izhikevich, 2006) to applying a synaptic pulse input to the variable v_i in the system described by Equation 3.17. Hence, in order to make a single neuron distinguish between different input membrane potentials, a sufficiently small value of ϵ is necessary. We will take this condition as an important prerequisite. So, in the rest of the thesis, we restrict our single model to an excitable FitzHugh-Nagumo type neuron with global mono-stability.



(a) Phase portrait of FitzHugh-Nagumo model for $a = 0.25$, $b = 1$ and $\epsilon = 0.001$

(b) Trace of the membrane potentials of two sampled solutions

(c) Trace of the gating variables of two sampled solutions

(d) Phase portrait of FitzHugh-Nagumo model for $a = 0.25$, $b = 1$ and $\epsilon = 0.01$

(e) Trace of the membrane potentials of two sampled solutions

(f) Trace of the gating variables of two sampled solutions

(g) Phase portrait of FitzHugh-Nagumo model for $a = 0.25$, $b = 1$ and $\epsilon = 0.1$

(h) Trace of the membrane potentials of two sampled solutions

(i) Trace of the gating variables of two sampled solutions

Figure 3.7: Phase portraits of FitzHugh-Nagumo model in Equation 3.3 with different values of ϵ. For the parameter settings, $a = 0.25$, $b = 1$, and ϵ is chosen as 0.001 for (a), 0.01 for (d) and 0.1 for (g). For all the values of ϵ, two types solutions starting respectively with $(0.26, 0)$ (blue trajectories) and $(0.24, 0)$ (green trajectories) are plot in the phase portraits followed by their corresponding curves in time domain. The initial conditions $(0.24, 0)$ and $(0.26, 0)$ is separated by $(a, 0) = (0.25, 0)$, while the two trajectories are widely separated only when ϵ is set very small.

3.4 Nomura Edge Detection Algorithms

This subsection aims to provide a brief review of the edge detection algorithm and its improved editions by Nomura and his colleagues (Ebihara et al., 2003; Kurata et al., 2008; Nomura et al., 2003, 2008, 2011b,a).

The discretised system in Equation 3.17 is also referred to as the *grid system* by Nomura et al. (2008, 2011a). As mentioned above, the initial values of v_i are determined by the original processed image for the edge detection. Specifically with this one neuron to one pixel structure, each membrane potential v_i of the neuron takes a rescaled pixel intensity ${U^r}_{(\mathsf{m},\mathsf{n})}$ as the initial condition. So, the initial condition of each neuron in the network is given by the following expressions,

$$\begin{cases} v_i(0) = {U^r}_{(\mathsf{m},\mathsf{n})} = \xi U_{(\mathsf{m},\mathsf{n})}, \quad \xi \ll 1 \\ w_i(0) = 0 \end{cases} \tag{3.19}$$

where ${U^r}_{(\mathsf{m},\mathsf{n})}$ and $U_{(\mathsf{m},\mathsf{n})}$ respectively indicate the rescaled and unrescaled light intensity of the pixel (m,n). ξ is the rescaling coefficient and usually $\xi = \frac{1}{1024}$ for 8-bit levels of gray scale image (Nomura et al., 2011a).

3.4.1 Binary Edge Detection Algorithm

In order to illustrate how the system detects the edges from binary images, we start with two examples of one dimensional edge detection. The discretisation in one dimensional cases is relatively easier, where $U^r(\mathsf{x})$ can be discretised as ${U^r}_i$ with the spatial $x = i{\cdot}\Delta h$. Δh is again the finite difference and i is the one-dimensional coordinate as well as the sequence number of each element. It will be shown that how a one dimensional system in Equation 3.17 detects edges from the one dimensional step functions by based on the *monostability* and *excitability* of the FitzHugh-Nagumo model with the *nearest neighbour coupling*.

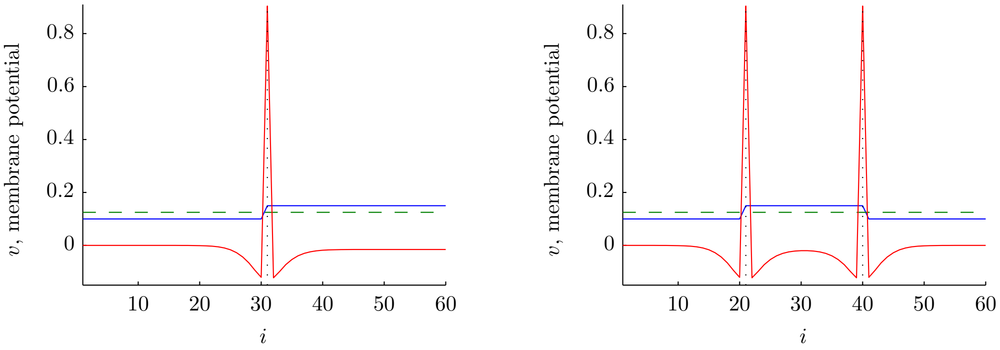
The first step function is provided as below,

$$v(0) = {U^r}_i = \begin{cases} 0.10, & 1 \leqslant i \leqslant 30 \\ 0.15, & 31 \leqslant i \leqslant 60 \end{cases}, \quad w_i(0) = 0 \tag{3.20}$$

And the second step function is provided as below,

$$v(0) = {U^r}_i = \begin{cases} 0.10, & 1 \leqslant i \leqslant 20 \\ 0.15, & 21 \leqslant i \leqslant 40 \\ 0.10, & 41 \leqslant i \leqslant 60 \end{cases}, \quad w_i(0) = 0 \tag{3.21}$$

The value of the parameter a is set as 0.125 between the two intensity levels. For other parameter settings, $\epsilon = 0.001$, $b = 1$, $k_v = 4$ and $k_w = 20$.



(a) Edge detection result on the step function in Equation 3.20

(b) Edge detection result on the step function in Equation 3.21

Figure 3.8: Edge detection results on one dimensional step function in Equation 3.20 and Equation 3.21 with constant image threshold. For other parameter settings, $\epsilon = 0.001$, $b = 1$, $k_v = 4$ and $k_w = 20$

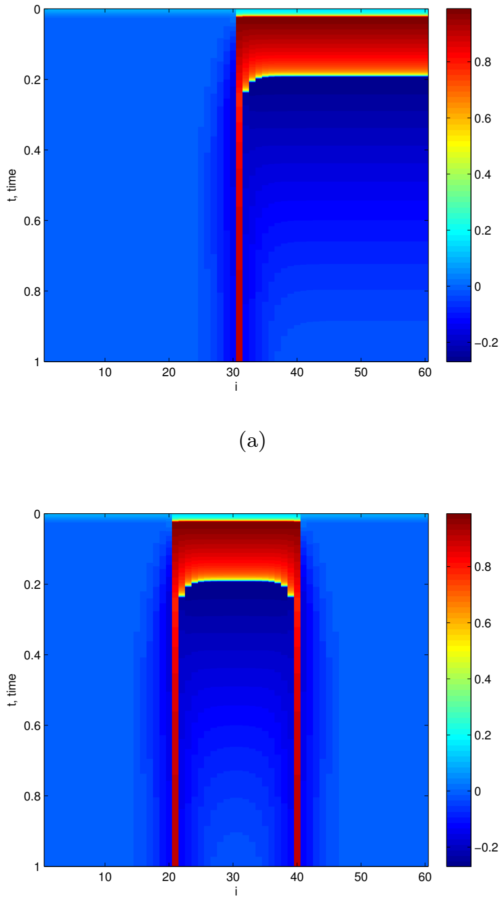
As mentioned above, for simple FitzHugh-Nagumo model without coupling in Equation 3.3, all the solutions will converge to the origin in the end. And if the coupling is included, all the neurons except the ones at the edges will behave similarly to the uncoupled case because the coupling is small. And each neuron at the edge point x_0 will maintain as a fixed high value in v forming a spatial pulse, namely $v(x_0) > v(x), x \neq x_0$. The results are provided in Figure 3.8. Hence, by setting parameter a to be a constant between the higher and lower image intensity levels, the system in Equation 3.17 is able to detect the edges in a binary image. And the final edge map $\mathcal{M}$ will be defined via a simple thresholding as follows,

$$\mathcal{M}_{(\mathsf{m},\mathsf{n})} = \begin{cases} 1, & v_i(t \geqslant \tau_s) > 0.5 \\ 0, & v_i(t \geqslant \tau_s) \leqslant 0.5 \end{cases} \tag{3.22}$$

where τ_s is the steady state time, for the time $t > \tau_s$, the solution $(v(t), w(t))$ reaches the final steady state. In order to illustrate the final result of edge detection is in a steady state, we plot the whole time evolution of $v(t, \mathsf{x})$ in Figure 3.9(a) and Figure 3.9(b) withe different initial conditions given in Equation 3.20 and Equation 3.21. It can be seen from both the figures in Figure 3.9, that the system state $v(t, \mathsf{x})$ does not change anymore when approximately $t > 0.21$. So, the value of τ_s can be chosen as 0.21.

3.4.2 Gray Level Image Edge Detection Algorithm

The network system in Equation 3.17 using a constant value of a as the threshold is only available for the binary image. In order to extend the application of edge detection



(a)

(b)

Figure 3.9: Time evolution of $v(t, \mathsf{x})$, where $\mathsf{x} = i\Delta\mathsf{h}$, $i \in [1, 60]$, $t \in [0, 1]$. Steady state pattern can found at $t = 1$ where two edge positions are detected

to gray level images, the threshold is required to be adjusted according to the processed image. So, Nomura et al. (2008, 2011b) proposed a method which chooses a blurred rescaled image intensity distribution as the threshold. Specifically, it is obtained via another diffusion equation,

$$\frac{\partial\theta}{\partial t} = D\nabla^2\theta \tag{3.23}$$

with the initial condition,

$$\theta(\mathsf{x}, \mathsf{y}, 0) = U^r(\mathsf{x}, \mathsf{y})$$

where, $\theta(\mathsf{x}, \mathsf{y}, t)$ denotes the diffused image intensity distribution and D is the continuous diffusion coefficient. Through the same discretisation with the same boundary condition provided in Section 3.3.1, we get

$$\dot{\theta}_i = d \sum_{j \in \mathcal{P}_i} (\theta_j - \theta_i), \quad \theta_i(0) = U^r{}_{(\mathsf{m},\mathsf{n})} \tag{3.24}$$

where, $d = \frac{D_\theta}{\Delta \mathsf{h}^2}$ is a constant. The diffusion adopted here is isotropic as reflected in the constant diffusion coefficient d. And blurring the original image U^r with such an isotropic diffusion is equivalent to filtering U^r with a Gaussian filter (Black et al., 1998) which is a solution to Equation 3.23 of the form

$$\theta(\mathsf{x}, \mathsf{y}, t) = \frac{1}{4\pi Dt} e^{-(\mathsf{x}^2+\mathsf{y}^2)/4Dt}. \tag{3.25}$$

So by choosing a appropriate constant of the time τ, the parameter a_i is set as $\theta_{(\mathsf{m},\mathsf{n})}(\tau)$ which is one "frame" of the time evolving solution $\theta_{(\mathsf{m},\mathsf{n})}(t)$ to the Equation 3.24.

However, the result of simply setting a variable threshold is not ideal for the edge detection of the gray level images; for a single edge position in the original image, they would get a correct spatial pulse denoting it but also a wrong one close to the correct one. The detailed formulation of this problem will be introduced later in Chapter 5. In order to eliminate the false pulses, Nomura modified the original system in Equation 3.17 as follows,

$$\begin{cases} \dot{v_i}^0 = f(v_i^0, w_i^0, a_i^0) + k_v \sum\limits_{j \in \mathcal{P}_i} (v_j^0 - v_i^0) + \dot{v_i}^1 \mathcal{H}(-\dot{v_i}^1) \\ \dot{w_i}^0 = g(v_i^0, w_i^0) + k_w \sum\limits_{j \in \mathcal{P}_i} (w_j^0 - w_i^0) \\ \dot{v_i}^1 = f(v_i^1, w_i^1, a_i^1) + k_v \sum\limits_{j \in \mathcal{P}_i} (v_j^1 - v_i^1) \\ \dot{w_i}^1 = g(v_i^1, w_i^1) + k_w \sum\limits_{j \in \mathcal{P}_i} (w_j^1 - w_i^1) \end{cases} \tag{3.26}$$

where $(v_i^0, w_i^0, v_i^1, w_i^1)$ are the dynamical variables. $\mathcal{H}(x)$ is the Heaviside step function, which gives 1 when $x \geq 0$ and gives 0 when $x < 0$. Basically, the new model in Equation 3.26 doubles the size of the original model in Equation 3.17. The superscripts 0 and 1 in each variable determines which subsystem the variable belongs to. Nomura et al. (2011b,a) claim that the term $\dot{v_i}^1 \mathcal{H}(-\dot{v_i}^1)$ in Equation 3.26 works for the elimination of the wrong spatial pulses, if the diffused intensity distribution a_i^1 is more blurred than a_i^0. More details are available in (Nomura et al., 2011b).

3.4.3 Improved Gray Level Image Edge Detection Algorithm

Although the algorithm introduced above for the edge detection on gray level images works in most cases, it still fails in some specific cases where darker intensity areas are surrounded by the brighter backgrounds. Nomura et al. (2011b) identified this problem and provided an improved approach which merged both the edge maps obtained from the original image intensity distribution $U_{\mathsf{m},\mathsf{n}}$ and its black-and-white inversion $\bar{U}_{\mathsf{m},\mathsf{n}}$ to obtain the final result. For an 8-bit image, the inverted image intensity distribution, is

$$\bar{U}_{\mathsf{m},\mathsf{n}} = 255 - U_{\mathsf{m},\mathsf{n}}.$$

Their recent results in (Nomura et al., 2011a) demonstrate the improvement in edge detection with this method.

3.4.4 Conclusion

Hence, the edge detection algorithm proposed by Nomura and his colleagues (Ebihara et al., 2003; Kurata et al., 2008; Nomura et al., 2003, 2008, 2011b,a) for the 8-bit gray level images can be concluded as in Algorithm 1

Algorithm 1 Nomura's Method

```
1: Rescale the image intensity distribution $U^r{}_{\mathsf{m,n}} = \xi U_{(\mathsf{m,n})}$ with $\xi = 1/1024$ so that
   $0 \leqslant U^r{}_{(\mathsf{m,n})} \leqslant \frac{1}{4}$.
2: Solve the equation $\dot{\theta^0}_i = d^0 \sum_{j\in\mathcal{P}_i}(\theta^0{}_j - \theta^0{}_i)$ with the initial condition $\theta^0{}_i(0) = U^r{}_i$
   to get a blurred version of rescaled image with a stopping time $\tau$ to obtain the
   threshold value $a_i^0 = \theta^0(\tau)$.
3: Solve the equation $\dot{\theta^1}_i = d^1 \sum_{j\in\mathcal{P}_i}(\theta^1{}_j - \theta^1{}_i)$ by using a larger constant $d^1 > d^0$ with
   the same initial condition $\theta^0{}_i(0) = U^r{}_i$ to get the second threshold value $a_i^1 = \theta^1(\tau)$.
4: Solve the equation below,
   $$\begin{cases} \dot{v_i}^0 = f(v_i^0, w_i^0, a_i^0) + k_v \sum\limits_{j\in\mathcal{P}_i} (v_j^0 - v_i^0) + \dot{v_i}^1 \mathcal{H}(-\dot{v_i}^1) \\ \dot{w_i}^0 = g(v_i^0, w_i^0) + k_w \sum\limits_{j\in\mathcal{P}_i} (w_j^0 - w_i^0) \\ \dot{v_i}^1 = f(v_i^1, w_i^1, a_i^1) + k_v \sum\limits_{j\in\mathcal{P}_i} (v_j^1 - v_i^1) \\ \dot{w_i}^1 = g(v_i^1, w_i^1) + k_w \sum\limits_{j\in\mathcal{P}_i} (w_j^1 - w_i^1) \end{cases}$$
   with obtained thresholds $a_i^0$ and $a_i^1$ plus the initial conditions $(v_i(0), w_i(0)) = (U^r{}_i, 0)$
   and the zero boundary conditions to get the steady state solution in $v^0$, which is
   $v_i^0(\tau_s)$.
5: Obtain the first putative edge map $\mathcal{M}_1$ via a simple thresholding $v^0_{(\mathsf{m,n})}(\tau_s)$ as below,
   $$\mathcal{M}_1(\mathsf{m,n}) = \begin{cases} 1, & v_i^0(\tau_s) > 0.5 \\ 0, & v_i^0(\tau_s) \leqslant 0.5 \end{cases}$$
6: Repeat the steps 1 to 5 above to get a second putative edge map $\mathcal{M}_2$ from the
   black-and-white inversion of the original image $\bar{U}_{(\mathsf{m,n})}$
7: Get the final edge map $\mathcal{M}$ by merging the two putative ones, $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2$
```

The detailed parameter setting and the edge detection results of both artificial and real images by this algorithm will be provided in Chapter 5 to be compared with our proposed edge detection method of this thesis.

Chapter 4

Theory of Weakly Coupled Neural Network

The discussion provided in Section 3.3.2, focusing on the excitability of a single neuron, provides some intuition for how edge detection methods work, using the network model in Equation 3.17. However, it is not enough as a good theoretical explanation of the mechanism of edge detection of such a network of coupled neurons. Originally, Nomura et al. (2003, 2008) ascribe the success in detecting the edges of their algorithm to two features of the grid system. One is the thresholding effect of the reaction part which is chosen to be the mono-stable FitzHugh-Nagumo model (FitzHugh, 1961; Nagumo et al., 1962). This is clearly shown in the previous chapter and we will focus on how to quantify the excitability of FitzHugh-Nagumo model in the next chapter. The other is the strong inhibitory constraint they set to the model that the diffusion of inhibitor should be much faster than that of activator. They claim that it is a necessary condition to ensure the stability of the final edge detection result. Because the condition is also similar to the famous Turing condition, they adopt Turing instability as the theoretical basis of their algorithm, which refers to loss of the stability of the unique attractor for pure reaction dynamics in the presence of diffusion.

However, the later work done by Kurata et al. (2008) differentiates the stable edge results obtained from Nomura's algorithm from the Turing patterns. They point out that the final edge detection results are neither the Turing patterns nor another example of propagation failures. In order to understand the model and simplify the problem, they focus on the simplest network structure which has only two excitable elements. Through both numerical simulation and analytical method, they conclude that the trivial state (the origin) is always stable no matter whether the diffusion term is included or not and the edge results are the newly appearing stable state.

In this chapter, we present the same results to that of (Kurata et al., 2008) and further analytically prove that the origin as stable equilibrium in the uncoupled system will remain stable when the diffusion terms are included, for any large size of the system ($\mathsf{M} \times \mathsf{N}$). We link the ability of the system to edge detection to the dynamical theory of weakly coupled neural network (Hoppensteadt and Izhikevich, 1997). The coupled neuronal network has a large size due its intrinsic pixel-neuron structure. In order to study such a model of very high dimension, it is natural and reasonable to think about building the link between uncoupled single neuron and the coupled ones and further the relationship between a small sub-graph of coupled neurons to the whole network. In fact, the simple linear structure of the nearest neighbour coupling makes the idea feasible. In Section 4.1, we mainly focus on proving that the origin remains stable with different settings of the coupling. In Section 4, we provide several examples to show that the mono-stable uncoupled neurons becomes a multiple stable network and those high dimensional stable equilibria, that appear only when coupling is included, correspond to the putative edge detection results. Specifically, we demonstrate the effect of coupling term to the networks and link the network model in Equation 3.17 to the theory of WCNN (Weakly-Coupled Neuron Network) (Hoppensteadt and Izhikevich, 1997).

4.1 Stability of Coupled Networks at Origin

In Section 3.2, we checked that the model of a single neuron is mono-stable under the given condition of system parameters. In this section, we will further prove that the coupled network will keep the same stability at the origin when the coupling is included. Namely, the system is always stable within some certain ranges of parameters no matter the neurons are coupled or not. First, we will analyse the structure of the whole network where the Jacobian of network connectivity can be expressed as a negative *Laplacian* matrix. Second, we will prove that this negative *Laplacian* matrix is always semi-negative definite based on the specific nearest coupling structure of the system by using the theory of Schur's complement. Specifically, we start with a chain structure and then, extend the conclusion to the grid structure of the whole network.

4.1.1 Coupling Network and Laplacian Matrix

Consider the dynamics of the whole system consisting of $\mathsf{M} \times \mathsf{N}$ neurons,

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}) \tag{4.1}$$

where $\mathbf{X} = (v_1, w_1, v_2, w_2, \dots)^T \in \mathbb{R}^{\mathsf{2MN} \times \mathsf{2MN}}$. $\mathbf{F}(\cdot) : \mathbb{R}^{\mathsf{2MN} \times \mathsf{2MN}} \to \mathbb{R}^{\mathsf{2MN} \times \mathsf{2MN}}$ is a vectorised function including the couplings. Figure 4.1 illustrates the structure of the whole neural network. Each black node represents a FitzHugh-Nagumo type of neuron. Each solid lines represent the couplings within a row while each dash represents those crossover the rows. So, each neuron is only connected to its nearby neurons forming

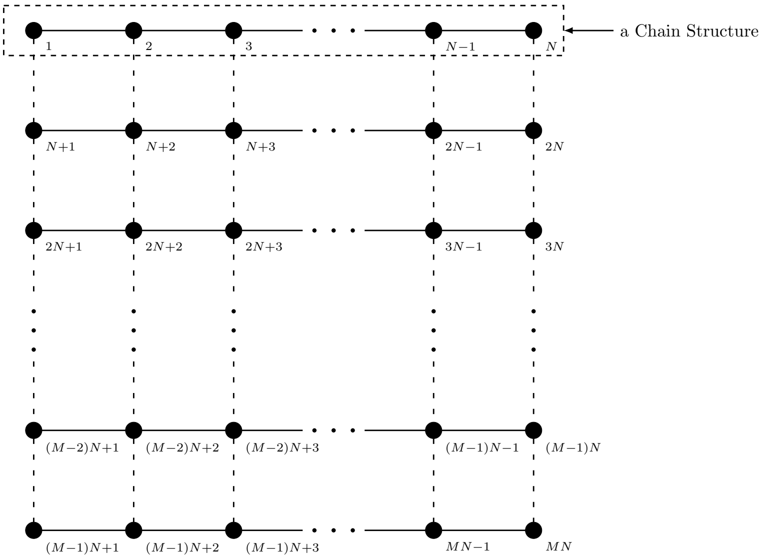


Figure 4.1: The network described in Equation 3.17 is a grid structure consisting of excitable elements (black nodes) and nearest couplings (solid or dashed lines). Each row of elements connected with the solid lines in the network can been seen as a chain. And the whole network can been seen as a chain of chains connected with the dashes.

a grid structure as whole. We will show later that this way of separately treating the couplings as two groups will facilitate understanding of the network structure.

The Jacobian $\mathbf{J}$ of the whole model in (4.1) can be expressed as below,

$$\mathbf{J} = \mathbf{DF}(\mathbf{X}) = \mathbf{J}_u(\mathbf{X}) + \mathbf{G} \tag{4.2}$$

where

$$\mathbf{J}_u = J_1 \oplus J_2 \oplus \dots J_{\mathsf{MN}} = \begin{pmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_{\mathsf{MN}} \end{pmatrix} \in \mathbb{R}^{\mathsf{2MN \times 2MN}} \tag{4.3}$$

is the Jacobian of the uncoupled network. The Jacobian J_i of a single uncoupled neuron is provided in Equation 3.5 which we rewrite as following,

$$J = \begin{pmatrix} -\frac{3}{\epsilon}v^2 + \frac{2(a+1)}{\epsilon}v - \frac{a}{\epsilon} & -\frac{1}{\epsilon} \\ 1 & -b \end{pmatrix}$$

$\mathbf{J}_u(0)$ can be easily proved to be negative definite since each $J_i(0)$ is negative definite. $\mathbf{G}$ is the Jacobian of all the coupling terms. Because the network is linearly coupled, namely the coupling terms are linearly dependent on the state variable X, $\mathbf{G}$ is a constant negative Laplacian matrix.

In graph theory, the Laplacian matrix is used as a representation of a graph such as the network structure studied here in Figure 4.1. And it is also used to model the structure of linear coupling in neural networks (Yu and Slotine, 2009). Generally, the Laplacian matrix is proved to be semi-positive definite in many other books such as (Fan, 1997). Here, we focus on the specific structure of our network model and provide an alternative proof. So, it will be finally shown that $\mathbf{J}(0) = \mathbf{J}_u(0) + \mathbf{G}$ is always negative definite, namely, the system is stable at the origin no matter whether the coupling is included or not.

Proof. In details, the coupling matrix $\mathbf{G}$ can be expressed as the sum of two parts,

$$\mathbf{G} = \mathbf{G}_c + \mathbf{G}_d$$

$\mathbf{G}_c$ represents the part of Jacobian contributed by the couplings within the rows of the network which can be expressed as below,

$$\mathbf{G}_c = I^{(\mathsf{M})} \otimes A^{(\mathsf{N})} \otimes K \tag{4.4}$$

And $\mathbf{G}_d$ represents the Jacobian contributed by the couplings crossover the rows,

$$\mathbf{G}_d = A^{(\mathsf{M})} \otimes I^{(\mathsf{N})} \otimes K \tag{4.5}$$

where $\otimes$ indicates the *Kronecker product.* For example, let $L = \left(\begin{smallmatrix} l_{11} & l_{12} \\ l_{21} & l_{22} \end{smallmatrix}\right)$ and $M = \left(\begin{smallmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{smallmatrix}\right)$,

$$L \otimes M = \begin{pmatrix} l_{11}M & l_{12}M \\ l_{21}M & l_{22}M \end{pmatrix} = \begin{pmatrix} l_{11}m_{11} & l_{11}m_{12} & l_{12}m_{11} & l_{12}m_{12} \\ l_{11}m_{21} & l_{11}m_{22} & l_{12}m_{21} & l_{12}m_{22} \\ l_{21}m_{11} & l_{21}m_{12} & l_{22}m_{11} & l_{22}m_{12} \\ l_{21}m_{21} & l_{21}m_{22} & l_{22}m_{21} & l_{22}m_{22} \end{pmatrix}$$

$K = \left(\begin{smallmatrix} k_v & \\ & k_w \end{smallmatrix}\right)$ is the matrix of coupling strength. $I^{(\mathsf{N})} \in \mathbb{R}^{\mathsf{N}\times\mathsf{N}}$ is the identical matrix. $A^{(\mathsf{N})}$ is a symmetric matrix of $\mathsf{N} \times \mathsf{N}$ size as expressed below,

$$A^{(\mathsf{N})} = \begin{pmatrix} -1 & 1 & & \\ 1 & -2 & & \\ & & \ddots & \\ & & & -1 \end{pmatrix} \in \mathbb{R}^{\mathsf{N}\times\mathsf{N}}, \tag{4.6}$$

Since both K and I are diagonal matrices, whether the matrices $\mathbf{G}_c$ and $\mathbf{G}_d$ are negative definite is determined by A.

A $(p+q) \times (p+q)$ nonsingular square matrix M can be partitioned as a 2×2 block matrix as shown below,

$$M = \left(\begin{array}{c|c} M_1 & M_2 \\ \hline M_3 & M_4 \end{array}\right)$$

where, M_1 is a $p \times p$ matrix while M_4 is a $q \times q$ matrix. So, M_2 is a $p \times q$ matrix and M_3 is a $q \times p$ matrix. If M_4 is nonsingular, the matrix, the *Schur's Complement* of M

with respect to M_4 (Boyd and Vandenberghe, 2004) is defined as

$$M/M_4 = M_1 - M_2 M_4^{-1} M_3. \tag{4.7}$$

Moreover, *Schur's Complement Lemma* Boyd and Vandenberghe (2004) states that, if M is symmetric, i.e., $M_1^T = M_1$, $M_4^T = M_4$ and $M_2^T = M_3$, then the property that M is negative semi-definite is equivalent to that the Schur's complement in Equation 4.7 is negative semi-definite.

Note that $A^{(\mathsf{N})}$ can be partitioned as below,

$$A^{(\mathsf{N})} = \left(\begin{array}{c|c} A_1 & A_2 \\ \hline A_2^T & A_3 \end{array}\right) = \left(\begin{array}{cccc|c} -1 & 1 & \cdots & 0 & 0 \\ 1 & -2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -2 & 1 \\ \hline 0 & 0 & \cdots & 1 & -1 \end{array}\right) \in \mathbb{R}^{\mathsf{N}\times\mathsf{N}}$$

Since $A_3 = -1 < 0$, the Schur's complement of $A^{(\mathsf{N})}$ with respect to A_3 is $A_1 - A_2 A_3^{-1} A_2^T$, which is indeed $A^{(\mathsf{N}-1)}$. According to Schur's complement lemma, if $A^{(\mathsf{N})}$ is negative semi-definite then $A^{(\mathsf{N}-1)}$ will also be negative semi-definite, and vice versa. Keep applying this property recursively until $\mathsf{N} = 2$, we obtain

$$A^{(2)} = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}$$

$A^{(2)}$ is a semi-negative definite matrix with two eigenvalues $\lambda_1 = -2$ and $\lambda_2 = 0$. Hence, $A^{(\mathsf{N})}$ is also semi-negative definite. Finally, both G_c and G_d are semi-negative definite.

$$\mathbf{G}_c \leqslant 0, \quad \mathbf{G}_d \leqslant 0$$

Finally,

$$\mathbf{J}(0) = \mathbf{J}_u(0) + \mathbf{G}_c + \mathbf{G}_d < 0,$$

since sum of semi-negative definite matrices and negative definite matrix is negative definite. □

Hence, we have proved that the stability of the origin does not change when the coupling is included. So, *Turing instability* which requires the loss of stability of the unique equilibrium of uncoupled system is not applicable here. Generally, Turing instability is a sufficient condition for generating self-organising patterns in the network. We will show that even though the coupling terms does not change the stability of the origin, instead, it makes the original monostable uncoupled system become multi-stable and the final edge detection result can seen as another stable equilibrium $\overline{\mathbf{X}} \in \mathbb{R}^{2\mathsf{MN}}$ which is chosen according to the initial image U^r.

4.2 Attractor Network

The behaviour or property of a network system without coupling is equivalent to the single FitzHugh-Nagumo model. Thereby, if the nearest coupling is not included, i.e. $k_v = 0$ and $k_w = 0$, the whole system with the parameter settings in Equation 3.9 has a unique equilibrium which is the high dimensional origin ($0 \in \mathbb{R}^{2\mathsf{MN}}$). It has already been proved that this high dimensional origin is always stable so that any non-zero initial conditions will finally shrink to zero. However, in the simulation of edge detection on images, we clearly see the initial conditions translated from processed images evolve to final stable edge maps. Hence, we believe that the couplings bring bifurcations to the whole system and new stable equilibrium other than the origin appears as the expected result of edge detection.

We will present several simulations on the networks of a few neurons in order to support our belief. More specifically, we will track the equilibrium with continuum in coupling strength in order to justify the bifurcation types. Finally, we link our discussion on the property of the network to the theory of weakly coupled neuron network.

4.2.1 Edge Map: High Dimensional Attractor after Bifurcations

We start with the simplest network case that consists of only two neurons ($M = 1$,$N = 2$). The equations of such system can be expressed as below,

$$\begin{cases} \dot{v_1} = \dfrac{1}{\epsilon}(v_1(1-v_1)(v_1-a_1)-w_1)+k_v(v_2-v_1) \\ \dot{w_1} = v_1 - bw_1 + k_w(w_2-w_1) \\ \dot{v_2} = \dfrac{1}{\epsilon}(v_2(1-v_2)(v_2-a_2)-w_2)+k_v(v_1-v_2) \\ \dot{w_2} = v_2 - bw_2 + k_w(w_1-w_2) \end{cases} \tag{4.8}$$

where $\mathbf{X} = (v_1, w_1, v_2, w_2)^T \in \mathbb{R}^4$. We will use $\overline{\mathbf{X}}$ to indicate the equilibrium in $\mathbb{R}^4$ of the system. And we determine the stabilities of $\overline{\mathbf{X}}$ according to the Jacobian of the system in Equation 4.8 which can be expressed as below,

$$\mathbf{J} = \begin{pmatrix} -\epsilon^{-1}(3v_1^2-2(a_1+1)v_1+a_1)-k_v & \epsilon^{-1} & k_v & 0 \\ 1 & -b-k_w & 0 & k_w \\ k_v & 0 & -\epsilon^{-1}(3v_2^2-2(a_2+1)v_2+a_2)-k_v & \epsilon^{-1} \\ 0 & k_w & 1 & -b-k_w \end{pmatrix} \tag{4.9}$$

and correspondingly, $\mathbf{G} \in \mathbb{R}^{4\times 4}$ is expressed as below,

$$\mathbf{G} = \begin{pmatrix} -k_v & 0 & k_v & 0 \\ 0 & -k_w & 0 & k_w \\ k_v & 0 & -k_v & 0 \\ 0 & k_w & 1 & -k_w \end{pmatrix}$$

We set the system with $a_1 = 0.1$, $a_2 = 0.2$, $b_1 = b_2 = 4$ to ensure the condition in Equation 3.9 so that the uncoupled network ($k_v = 0$,$k_w = 0$) has only one equilibrium at the origin. To examine the stability, we evaluate the Jacobian at the origin and find the largest eigenvalue,

$$\max_i(\mathrm{Re}(\lambda_i|_{\overline{\mathbf{X}}=0})) = -9.2423 \tag{4.10}$$

When the coupling is included ($k_v = 1$,$k_w = 5$), we evaluate all the equilibria of system by seeking the real solutions to $\mathbf{F}(\mathbf{X}) = 0$ and list them in Table 4.1, together with the corresponding eigenvalues in Table 4.2.

$\overline{\mathbf{X}}$	$\overline{\mathbf{X}}_1$	$\overline{\mathbf{X}}_2$	$\overline{\mathbf{X}}_3$	$\overline{\mathbf{X}}_4$
v_1	-0.1055	0.7853	-0.1450	0.3081
w_1	0.0245	0.1146	0.0415	0.0440
v_2	0.4646	-0.1298	0.7261	-0.0619
w_2	0.0653	0.0493	0.1038	0.0176

Table 4.1: List of all the high dimensional equilibria $\overline{\mathbf{X}} = (v_1, w_1, v_2, w_2)^T$ for the coupled two neurons in Equation 4.8. For the parameter settings, $a_1 = 0.1$, $a_2 = 0.2$, $b_1 = b_2 = 4$. For the settings of coupling strength, $k_v = 1$ and $k_w = 5$.

$\mu\|_{\overline{\mathbf{X}}}$	$\overline{\mathbf{X}} = \overline{\mathbf{X}}_1$	$\overline{\mathbf{X}} = \overline{\mathbf{X}}_2$	$\overline{\mathbf{X}} = \overline{\mathbf{X}}_3$	$\overline{\mathbf{X}} = \overline{\mathbf{X}}_4$
λ_1	-3.6362	-5.6116	-4.8088	2.8869
λ_2	2.6280	-2.1870	-0.1036	-3.5807
λ_3	-0.1459	-0.1756	$-0.2493 + 0.2727i$	-0.0281
λ_4	-0.0255	-0.0702	$-0.2493 - 0.2727i$	-0.1469

Table 4.2: List of eigenvalues $\mu|_{\overline{\mathbf{X}}} = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)^T$ of Jacobian in Equation 4.9 for all the equilibria in Table 4.1. Specifically, $\overline{\mathbf{X}}_2$ and $\overline{\mathbf{X}}_3$ are stable equilibria, because all the real parts of their eigenvalues of Jacobian are negative.

It can be seen that all the values of λ_2 and λ_3 are negative or have negative real part. Hence, $\overline{\mathbf{X}}_2$ and $\overline{\mathbf{X}}_3$ are the newly appearing stable equilibria (attractors) other than the origin.

In the application of edge detection, only v variables take the value from the original image U^r. Here, the values of w_1 and w_2 do not have the specific meaning in final edge detection results. If we define a map, $\mathbf{v} : \mathbb{R}^{2\mathsf{MN}} \to \mathbb{R}^{\mathsf{MN}}$, so that

$$\mathbf{v}|_{\overline{\mathbf{X}}} = (v_1, v_2, v_3, \cdots)$$

then for a system of two coupled elements, the value of $\mathbf{v}$ as the expected pattern of edge detection results would only follow into one of the three cases below,

$$(v_1 \approx 1, \quad v_2 \approx 0) \quad (v_1 \approx 0, \quad v_2 \approx 1) \quad (v_1 \approx 0, \quad v_2 \approx 0)$$

where, the sign '≈' denotes 'being close to'. The first case means the left pixel is an edge (assuming that v_1 takes the image intensity of the left pixel as initial value and v_2 takes that of the right). The second case means the right pixel is an edge, and the third case means no edge. It can be clearly checked in Table 4.1 that

$$\begin{aligned}
\mathbf{v}|_{\overline{\mathbf{X}}=0} &= (0,0) \\
\mathbf{v}|_{\overline{\mathbf{X}}_2} &= (0,79-0.13) \approx (1,0) \\
\mathbf{v}|_{\overline{\mathbf{X}}_3} &= (-0.150.73) \approx (0,1)
\end{aligned}$$

The example above illustrates that the number of the equilibria becomes larger than one if a sufficiently large coupling is included. Moreover, these newly appearing equilibria could be either stable or unstable and each pattern of possibly potential edge detection results corresponds to one of those stable attractor.

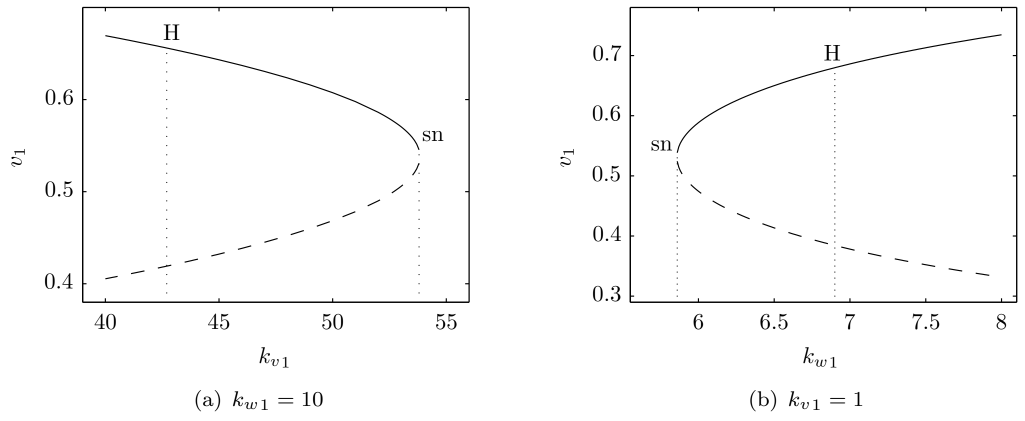
4.2.2 Effect of Coupling

Through the simulation provided in the previous section, we have got the basic idea that the monostability of uncoupled network is broken by the coupling and the system becomes multistable. And these newly appearing high dimensional equilibria are indeed closely related to the potential edge maps. This section will include couplings of different strengths to the whole network, in order to investigate their influences on the system dynamics. We will start with the uncoupled monostable network and treat both k_{vi} and k_{wi} as the bifurcation parameters. By changing both k_{vi} and k_{wi} within a relatively small range, we examine how the stability changes. Finally, we will include extremely strong couplings to the network, in order to examine how the bifurcation condition is influenced by the initial parameter settings, a_i and b_i.

Small Coupling and Bifurcation Pattern

In section 3.2, we have shown the bifurcation pattern for a single FitzHugh-Nagumo model, by tracking v_+. Here, we will see that the bifurcation pattern for two coupled neurons is similar to that for a single neuron.

The parameter settings $a_1 = a_2 = 0.1$ and $b_1 = b_2 = 2$ are the same for both examples in Figure 4.2. Moreover, the coupling strength $k_{v2} = 1$ and $k_{w2} = 5$ also remain as constants. In the first simulation, $k_{w1} = 10$ is constant and k_{v1} is treated as the bifurcation parameter swept from 40 to 55. While in the second simulation. $k_{v1} = 1$



(a) $k_{w1} = 10$

(b) $k_{v1} = 1$

Figure 4.2: Bifurcation diagram on coupling strength as bifurcation parameters

is constant and k_{w1} is treated as the bifurcation parameter swept from 5.5 to 8. The bifurcation patterns presented in Figure 4.2 are similar to that of a single neuron in Figure 3.3(b); the equilibrium undergos first a saddle node bifurcation then a Hopf bifurcation if we reduce the value of k_{v1} or increase the value of k_{w1}. Because the ratio ϵ of timescales for v and w is chosen as a very small constant, k_v is required to be set as a large value in order to make qualitative influence to the system; it shows that the bifurcation happens at least when $k_{v1} > 40$.

Consider the network in Equation 4.8 and we can rewrite expressions as below

$$\begin{cases} \epsilon\dot{v}_1 = v_1(1-v_1)(v_1-a_1) - w_1 + \epsilon k_{v1}(v_2 - v_1) \\ \dot{w}_1 = v_1 - b_1 w_1 + k_{w1}(w_2 - w_1) \end{cases} \tag{4.11}$$

with similar equations for v_2 and w_2, because of the symmetry of the coupled two neurons. It can be seen in this system of equations that the coupling through the state variable v may contribute little to the whole system dynamics compared to that through the state variable w, when $\epsilon \ll 1$. Therefore, it may be omitted from the model. In order to support this assumption, we provide another two groups of simulations illustrating the influence of k_{vi} on the two coupled neurons. For the sake of convenience, we assume $k_{v1} = k_{v2} = k_v$ and $k_{w1} = k_{w2} = k_w$ without loss of generality. We focus on the specific parameter settings $(a_1 = a_2 = 0.25, b_1 = b_2 = 4)$, and investigate both the uncoupled and the coupled cases for two different values of ϵ: $(0.001, 0.0005)$. The initial condition adopted is

$$(v_1(0), v_2(0)) = (0.26, 0.24), \quad (w_1(0), w_2(0)) = (0, 0).$$

So in this case, the expected edge detection result will be

$$\mathbf{v} = (v_1, v_2)|_{t\to\infty} \approx (1, 0)$$

The first four figures in Figure 4.3 present the simulation result for $\epsilon = 0.001$. Figure 4.3(b) and Figure 4.3(d) show that clear edge detection (in red) result is obtained when $k_v = 0$. However, when k_v increases, the edge is no longer detected and both v curves and w curves will be more and more similar to the uncoupled cases as shown in Figure 4.3(a) and Figure 4.3(c). When ϵ further decreases to 0.0005, as shown in the rest figures in Figure 4.3, the effect of changing the strength of k_v on the edge obtained is very little, as shown in Figure 4.3(f) and Figure 4.3(h), so that it reasonable to neglect it.

The simulation results in the previous sections tell us that, first, the increase of k_v is not conducive to get the edge result, and second, the effect of v coupling is limited for sufficiently small ϵ. These two conclusions provide the justification for removing the coupling term through v in Equation 3.17 to obtain a simplified model follows,

$$\begin{cases} \dot{v_i} = f(v_i, w_i, a_i) \\ \dot{w_i} = g(v_i, w_i) + k_w \sum\limits_{j \in \mathcal{P}_i} (w_j - w_i), \end{cases} \tag{4.12}$$

where, the neurons are coupled only via k_w. Again, $\mathcal{P}_i = f_c(\mathcal{P}_{(\mathsf{m},\mathsf{n})})$ is the set of nearest neighbours of the i-th element. The edge detection results provided later will illustrate that the elimination of the v coupling does not adversely affect the performance of the algorithm for the whole network system.

Extremely Large Coupling

Assume that $v = (v_1, v_2)^T$, $w = (w_1, w_2)^T$ and $k_{w1} = k_{w2} = k$. It can be clearly seen from equation Equation 4.8 that w is linearly dependent on v as,

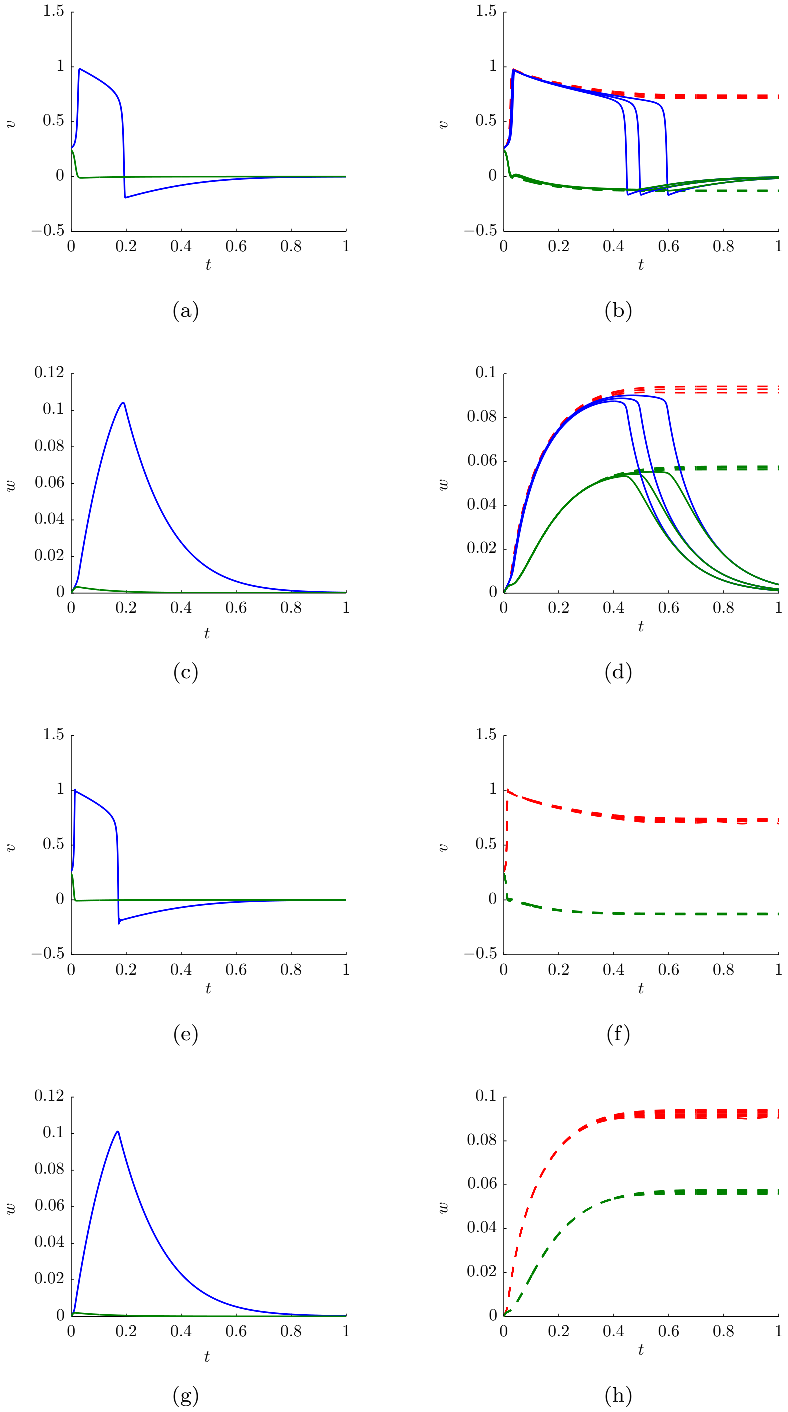
$$v + \begin{pmatrix} -b_1 - k & k \\ k & -b_2 - k \end{pmatrix} w = 0$$

So,

$$\begin{aligned} w &= -\begin{pmatrix} -b_1 - k & k \\ k & -b_2 - k \end{pmatrix}^{-1} v \\ &= \begin{pmatrix} \frac{b_2 + k}{b_1 b_2 + k(b_1 + b_2)} & \frac{k}{b_1 b_2 + k(b_1 + b_2)} \\ \frac{k}{b_1 b_2 + k(b_1 + b_2)} & \frac{b_1 + k}{b_1 b_2 + k(b_1 + b_2)} \end{pmatrix} v \end{aligned}$$

when the coupling strength k_w goes to infinity, w can be written as,

$$w = \frac{1}{b_1 + b_2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} v$$



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 4.3: The effect of v coupling on system dynamics. The first four figures (a), (b), (c) and (d) present simulation for $\epsilon = 0.001$ and the rest four figures (e), (f), (g) and (h) present that for $\epsilon = 0.0005$. Figures in the left column illustrate the simulation of the uncoupled models, while figures in the right column are that of the coupled models. In all the simulation of coupled models, $k_w = 10$ and k_v is swept from 0 to 10 with the step of 0.5. For other parameter settings, $a_1 = a_2 = a = 0.25$,$b_1 = b_2 = b = 4$.

Since $\epsilon \ll 1$, we omit the influence of k_{vi} and substitute the equation above to equation Equation 4.8 to obtain,

$$\begin{cases} (b_1+b_2)(-v_1^3+(a_1+1)v_1^2-a_1v_1)-(v_1+v_2)=0 \\ (b_1+b_2)(-v_2^3+(a_2+1)v_2^2-a_2v_2)-(v_1+v_2)=0 \end{cases}$$

Eliminating the variable v_2 obtains the equation of v_1. And the result obtained in Mathematica is presented below,

$$-(b_1+b_2)v_1p(v_1)=0$$

where $p(v_1)$ is a univariate polynomial of degree 8 whose coefficients are functions of a_1, a_2, b_1 and b_2. Obviously, this equation has a root $v_1=0$, but it is still a difficult work to find out the explicit analytical condition when it has other roots. So, we will present a numerical simulation on such coupled two neurons to see whether the coupled network always has equilibrium other than the origin.

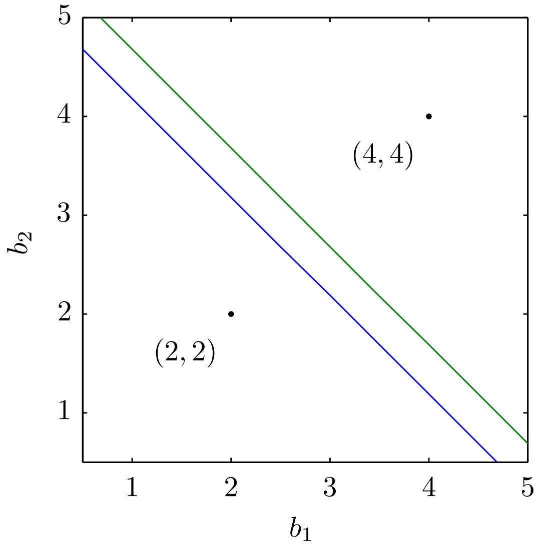
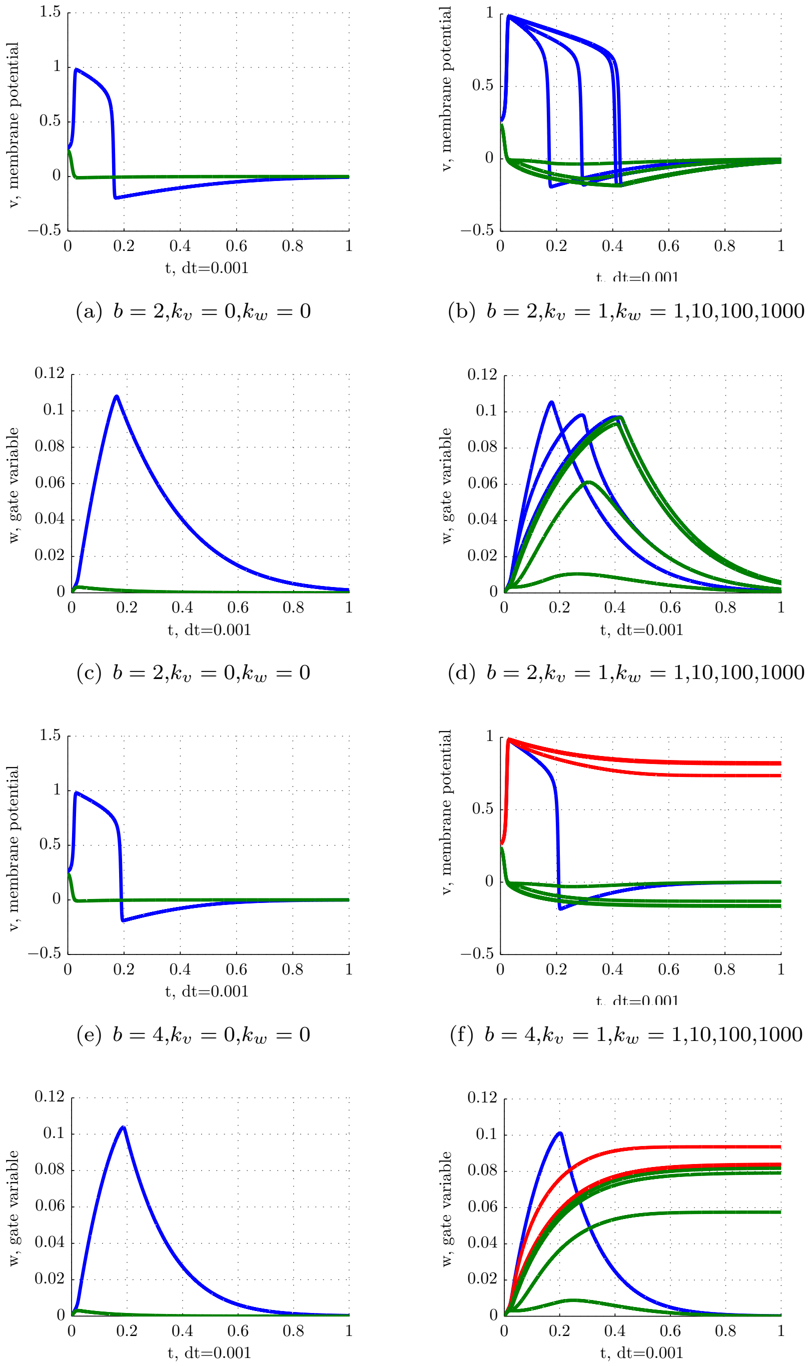


Figure 4.4: Structural stability of the uncoupled system to the extremely large coupling through the state variable w. For parameter settings, $a_1=a_2=0.25$, $\epsilon=0.001$. For coupling strength $k_{v1}=k_{v2}=1$ and $k_{w1}=k_{w2}=5\times10^3$. Two straight lines respectively indicates the boundary of saddle-node bifurcation (solid line) and Hopf bifurcation (dashed line). The boundaries exhibits the symmetry due to the same setting of a_i. Generally, the uncoupled network requires to be set near to its bifurcation points, to ensure that it undergoes the bifurcations after the coupling is included.

Two groups of time domain simulation on these two points are presented in Figure 4.5. In either group, we focus on two identical neurons and we check both uncoupled and coupled case by running the system at the same initial conditions $(v_0,w_0)=(0.24,0)$ and $(v_0,w_0)=(0.26,0)$. a is chosen to be 0.25 in both groups of simulations. However, in contrast to the first group (Figure 4.5(a)4.5(b)4.5(c)4.5(d)), b is increased from 2 to

4 in the second group (Figure 4.5(e)4.5(f)4.5(g)4.5(h)) so that the second single neuron is nearer to the bifurcation point in the uncoupled case. For coupled cases, the coupling strength k_v is chosen as a small value while k_w is increased to an extremely large one. It can be seen from results of the first group simulation ($b = 2$) that there is no new equilibrium turning up no matter how large k_w is. However, different result is obtained in the second case ($b = 4$) where a new equilibrium appears when the coupling is included. Hence, we can draw a conclusion that the parameters of the uncoupled neuron is required to be set sufficiently near to its bifurcation point in order for the coupled case having new equilibrium other than the origin.



(a) $b = 2, k_v = 0, k_w = 0$

(b) $b = 2, k_v = 1, k_w = 1,10,100,1000$

(c) $b = 2, k_v = 0, k_w = 0$

(d) $b = 2, k_v = 1, k_w = 1,10,100,1000$

(e) $b = 4, k_v = 0, k_w = 0$

(f) $b = 4, k_v = 1, k_w = 1,10,100,1000$

(g) $b = 4, k_v = 0, k_w = 0$

(h) $b = 4, k_v = 1, k_w = 1,10,100,1000$

Figure 4.5: Two groups of simulations to check the structural stability of the uncoupled system to the extremely large coupling through the state variable w. For parameter settings, $a = 0.25$, $\epsilon = 0.001$ in both simulations. However, b is chosen to be 2 in first group and 4 in second one. In the coupled systems for both simulation, k_w is swept from 10^0 to 10^3.

4.3 Discussion and Conclusion

Through the simulations, we can get some basic ideas about linking the dynamics of the network system described in Equation 3.17 to its mechanism of edge detection. Consider the dynamics of the neural network in Equation 4.1 in a more specific expression as below,

$$\dot{\mathbf{X}} = \mathbf{F}_u(\mathbf{X}) + \mathbf{C}(\mathbf{X}), \qquad \mathbf{X} \in \mathbb{R}^{2\mathsf{MN}}.$$

where, $\mathbf{F}_u(\cdot) : \mathbb{R}^{2\mathsf{MN}} \to \mathbb{R}^{2\mathsf{MN}}$ and $\mathbf{C}(\cdot) : \mathbb{R}^{2\mathsf{MN}} \to \mathbb{R}^{2\mathsf{MN}}$ are vectorised functions which respectively represent the uncoupled dynamics and the coupling. Again, we use $\overline{\mathbf{X}}$ to denote the equilibria obtained from $\dot{\mathbf{X}} = 0$. We have shown the system has only one equilibrium for the uncoupled dynamics $\mathbf{F}(\mathbf{X}) = 0$ with the appropriate parameter settings. And if the coupling is included, the whole system will have more equilibria $\overline{\mathbf{X}}$. We elaborate that one of the equilibria $\overline{\mathbf{X}}$ which is likely to be non-hyperbolic can be mapped to the expected edge map $\mathcal{M}$. We use $\overline{\mathbf{X}}_{\mathcal{M}}$ to denote such an equilibrium. By tracking $\overline{\mathbf{X}}_{\mathcal{M}}$, these equilibria, we find out that the behaviour of system experience two bifurcations, firstly saddle-node bifurcation and secondly Hopf bifurcation. Moreover, the stability of these equilibria are assuring if they are beyond the Hopf bifurcation. We point out that this pattern is similar to that of single neuron. Thereby, the ability of the network to pick up edges as the steady final results indeed inherits from a single FitzHugh-Nagumo model. Most importantly, it is clearly shown that certain conditions should be satisfied to ensure the Non-hyperbolicity of the network system, that the uncoupled system is required to be near the bifurcation point.

Thus, the problem of detecting the correct edges in an image by using the class of dynamical models such as in Equation 3.17 addressed in this thesis, is equivalent to the following two steps. First, we determine the existence of a stable equilibrium $\overline{\mathbf{X}}_{\mathcal{M}}$. Second, we examine whether its attraction domain contains the initial condition given by the rescaled image,

$$U^r \in \mathcal{B}(\overline{\mathbf{X}}_{\mathcal{M}}).$$

Based on these observations, we reasonably link the network system described in Equation 3.17 to theory of WCNN. The theory mainly claims that WCNNs requires to be non-hyperbolic to exhibit non-trivial behaviours. From a brain dynamics point of view, these non-trivial behaviours are always considered valuable in performing computational tasks on given input patterns. In other words, if all the neurons in a weakly coupled network are locally hyperbolic, then such a network is proved to have similar local behaviour to the uncoupled network and even the corresponding linearised system. In our case, the simulations previously provided show that the uncoupled neuron is required to be near enough to its bifurcation point to get a corresponding bifurcation after being coupled. So, the coupling works as a disturbance to the uncoupled neurons which are nonhyperbolic, namely structurally unstable.

Based on this theory, Hoppensteadt and Izhikevich (1997) introduce the classification of a general neural network. They claim that most of the networks of coupled neurons can be classified as one of the following three types, if assume the input pattern is a stationary image, or changing slowly dependent on the time.

MA type (Multiple Attractor Neural Network) The input pattern is given as an initial condition of the network, and the network converges to one of possibly many choices.

GAS type (Globally Asymptotically Stable Neural Network) The key pattern is given as a parameter that controls the location of a unique limit cycle or the shape of a unique equilibrium.

NH types (Nonhyperbolic Neural Network) The input pattern is given as bifurcation parameter that perturbs a nonhyperbolic equilibrium.

The simulations provided above clearly shows that the network system in Equation 3.17 belongs to the MA type of neural network. However, it is slightly different from a general MA type network. According to Hoppensteadt and Izhikevich (1997), in order for a MA type network to perform an association process, each attractor of the network is associated to a prior memorised input pattern. A typical example of an MA type network is the Hopfield network (Hopfield, 1982). But, none of the stable equilibria of the system in Equation 3.17 corresponds to a prior memory. Either the algorithms proposed by Nomura et al. (2011a,b) or the ones proposed in this thesis work for general purpose edge detection. There is neither process of remembering nor requirement of doing this. In order for the system capable of processing all kinds of patterns, it should be possible for the system to converge to any of the attractors. We denote the attraction domain of a specific attractor $\overline{\mathbf{X}}$ by $\mathcal{B}(\overline{\mathbf{X}})$ and the corresponding boundary by $\partial\mathcal{B}(\overline{\mathbf{X}})$. In other words, finding the "exact image threshold" is equivalent to evaluating the boundary of attraction domain $\partial\mathcal{B}(\overline{\mathbf{X}})$ of the stable equilibrium representing the detected edge map $\mathcal{M}$.

For the binary edge detection algorithm, the system parameter settings are constant for all the neurons, which makes both the stable equilibrium $\overline{\mathbf{X}}$ and the boundary $\partial\mathcal{B}(\overline{\mathbf{X}})$ irrelevant to the initial conditions once the system parameter settings are determined. So, the initial images can only converge to a very limited number of stable equilibria. This is also the reason why constant a is only suitable for binary images. By introducing the variable a, the system becomes suitable for grayscale images. However, the parameter settings of the whole system are now influenced by the initial conditions set by the images and so are the boundaries of attraction domains $\partial\mathcal{B}(\overline{\mathbf{X}})$ which are the "exact image threshold". So it becomes a very difficult problem to find such "exact image threshold" for such high dimensional system of Equation 4.3 and how it changes according to the initial condition given by the image U. We note that Nomura *et al.* Nomura et al. (2003)Nomura et al. (2008)Nomura et al. (2011b) treat the system parameter a as an estimate for the “exact image threshold”, inspired by the excitability of the single neuron. However, the parameter a is not even accurate as a threshold of excitability of the uncoupled single FitzHugh-Nagumo type of neuron. So, the method of setting a according to the diffused image indeed controls the attraction domain of the attractors, so that the initial condition can automatically choose its destination as correctly as possible. We address this issue in the next chapter.

Chapter 5

Quantification of Excitability Using Continuous Lyapunov Exponents

The chapter focuses on the solution to the "wrong edge" problem found in edge detection result when extending Nomura's models from binary image application to grayscale one. Previously in Section 4.3, we have indicated that the initial condition $\mathbf{X}_0$, namely the rescaled input image U^r, would automatically evolve towards to a specific attractor $\overline{\mathbf{X}}$ representing the edge detection result, once it is located in the attraction domain $\mathcal{B}(\overline{\mathbf{X}})$. So basically, we ascribe the appearance of the wrong pulse or edge to that the boundary of attraction main $\partial\mathcal{B}(\overline{\mathbf{X}})$ is not correctly found. Firstly, we provide an example of one-dimensional edge detection in order to clearly illustrative the problem. Secondly, we quantitatively analyse the excitability of a single neuron by using *Lyapunov exponent* and find that the problem is caused by the inaccurate definition of the image threshold. Hence, we redefine the image threshold according the quantitative analysis. Thirdly, we propose our edge detection method according to the new image threshold and adopt the anisotropic diffusion to further inhibit the generation of the wrong pulse. Finally, we compare the results obtained from Nomura's algorithm and our method.

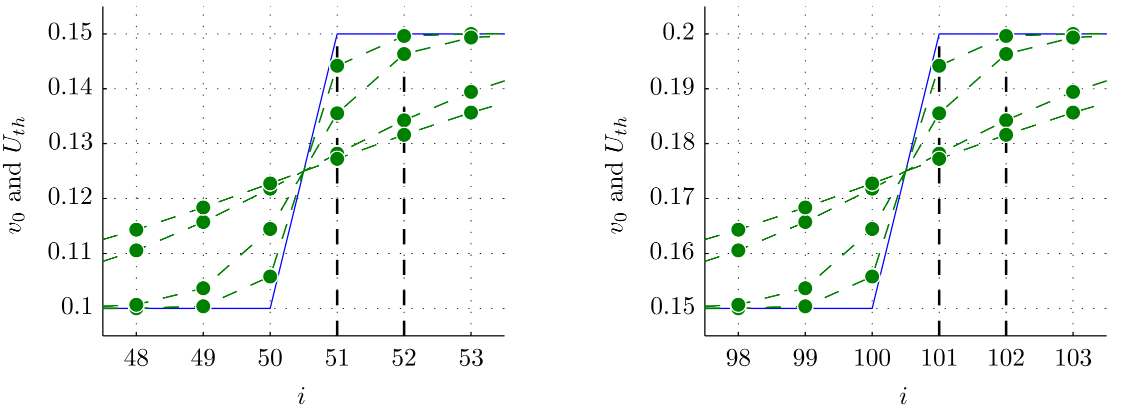
5.1 Problem Caused by Variable Threshold

This section will present the problem introduced when the threshold is set by diffusing the grayscale values of U^r. We mainly focus on an example of a 2-step image that sets initial condition

$$v_i(0) = {U^r}_i = \begin{cases} 0.10, & 1 \leqslant i \leqslant 50 \\ 0.15, & 51 \leqslant i \leqslant 100 \\ 0.20, & 101 \leqslant i \leqslant 150 \end{cases}, \quad w_i(0) = 0. \tag{5.1}$$

For such a 2-step function, a constant a is no longer sufficient for detecting both edges. So, according to Nomura's algorithm, we evaluate the parameter $a_i = \theta(\tau)$ according to Equation 3.24, where we choose the constant $d = 10$. With this example, we aim to extensively present the problem of the "wrong pulses" discovered in (Nomura et al., 2011b). With increasing values of the stopping time τ of the diffusion, we obtain θ with increasing degrees of blurring, with the consequent results provided in Figure 5.2.

Figure 5.1 provides a zoomed in view of the local area around both the potential edge positions in the original step function in Equation 5.1. Because the isotropic diffusion d in Equation 3.24 is constant for all the neurons and the magnitudes of the two steps are the same, the diffused steps in $\theta(\tau; d)$ around both the edge positions have the same shapes for each value of τ as shown in Figure 5.1(a) and Figure 5.1(b).



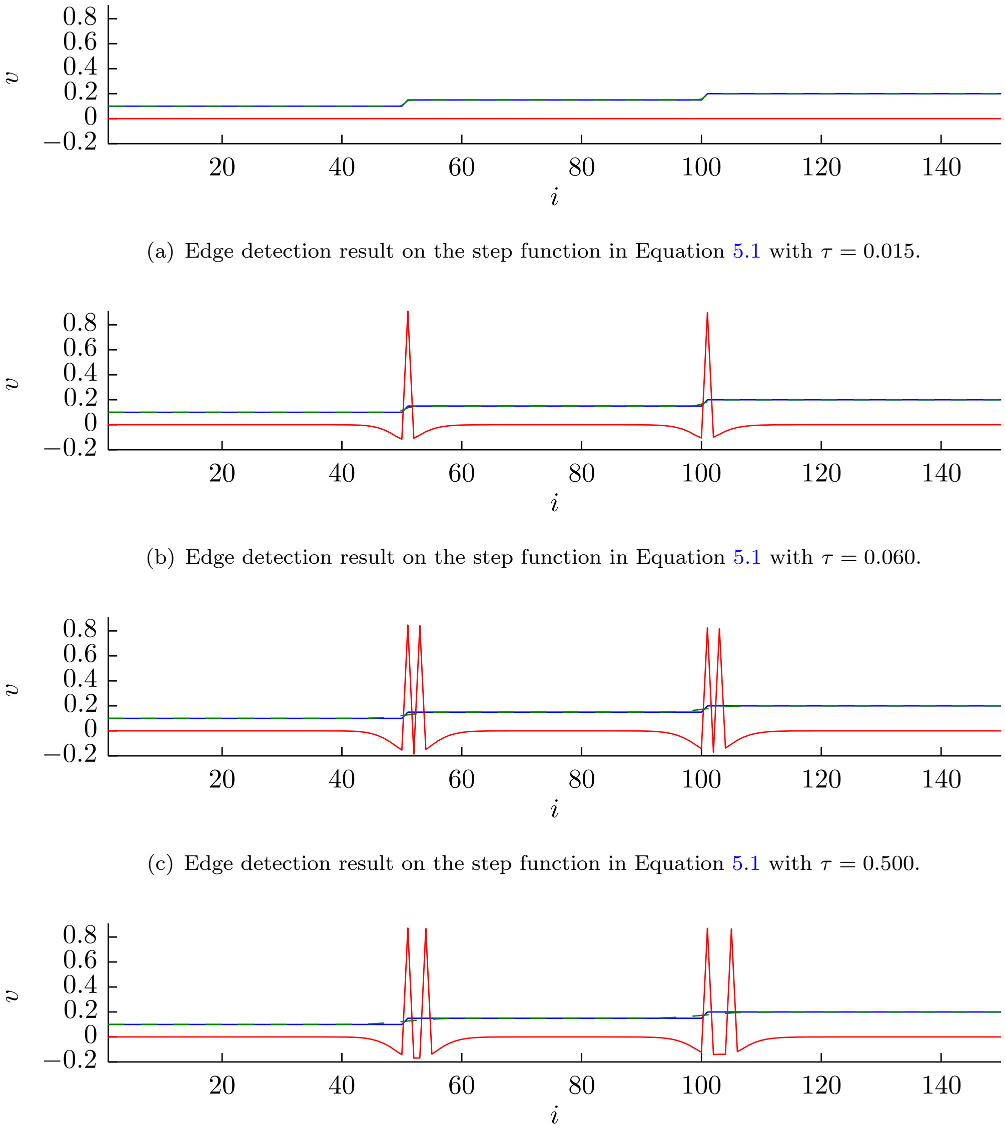
(a) Local area of the potential edge position $i = 51$

(b) Local area of the potential edge position $i = 101$

Figure 5.1: Image threshold $\theta_i(\tau)$ with τ equal to 0.015, 0.030, 0.060, 0.120, 0.200, 0.300, 0.500 and 1.000, respectively corresponding to the cases provided in Fig. 5.2

Note that, within a small local area around the edge position as shown in either Figure 5.1(a) or Figure 5.1(b), the diffused step is center-symmetric. Considering the right half of the step, the intensity level closer to the edge position becomes more distant from the original step level after diffusion, i.e. the more $\theta_i(\tau)$ decreases. Because a_i is directly set as θ_i, the neuron closer to the edge position is more likely to be excited when the step is diffused. Hence estimate that for large τ, there will be more than one neuron in an excited state, located not just at, but near the edge. Hence a wrong pulse will be produced as shown in Figure 5.2(c). The mechanism and outcome are similar to those in the second example in Figure 3.8 where there indeed are two edges to be detected. Further, the more diffused $\theta(\tau)$ is, the greater is the distance between the two pulses around either step, as shown in Figure 5.2(d).

Hence, we can summarize that the "wrong pulse" problem (Nomura et al., 2011b) is introduced by the method of setting a as variable threshold owing to the following reasons. For too small a value of τ, such as the case where $\tau = 0.015$ in Figure 5.2(a), the threshold is not sufficiently diffused, and thus there is no edge detected. And for too large a value of τ, such as the case where $\tau = 0.5$ in Figure 5.2(c), the threshold is spread out too far and a second stable edge is obtained at the wrong positions.



(a) Edge detection result on the step function in Equation 5.1 with $\tau = 0.015$.

(b) Edge detection result on the step function in Equation 5.1 with $\tau = 0.060$.

(c) Edge detection result on the step function in Equation 5.1 with $\tau = 0.500$.

(d) Edge detection result on the step function in Equation 5.1 with $\tau = 1.000$.

Figure 5.2: Edge detection results on one dimensional step function in Equation 5.1 with $a_i = \theta_i(\tau; d = 10)$ which is the variable image threshold obtained from Equation 3.24 with different values of τ. For other parameter settings, $b = 1$, $\epsilon = 0.001$ $k_v = 4$ and $k_w = 20$.

5.2 Proposed Edge Detection Method by Using Anisotropic Diffusion

In order to deal with the emergence of the wrong pulse, we proposed an alternative algorithm without increasing size of the system based on the model in Equation 3.17. In details, we provide a more accurate definition of the image threshold for a monostable FitzHugh-Nagumo model in Equation 3.3 by evaluating the Lyapunov characteristic exponents, which enables us to correctly deal with the problem when extending the algorithm to gray level images. Finally, in order to enhance the performance of edge detection, we introduce the anisotropic diffusion to Equation 3.24 where d is also determined by the rescaled image U^r.

5.2.1 Image Threshold: Boundary of Attraction Domain

As previously discussed in Section 4.3, the initial condition $\overline{\mathbf{X}}_0 = U^r$ requires to be located within the attraction domain of the correct attractor $\mathcal{B}(\overline{\mathbf{X}}_{\mathcal{M}})$. Hence, we need to evaluate the boundaries of attraction domain $\partial\mathcal{B}(\overline{\mathbf{X}})$ in the model in Equation 4.3 and track their changes when system parameters change.

However, it is a very difficult task to find the exact boundaries $\partial\mathcal{B}(\overline{\mathbf{X}})$ for such a high dimensional system in Equation 4.3. We note that Nomura *et al.* Nomura et al. (2003)Nomura et al. (2008)Nomura et al. (2011b) treat the system parameter a as an estimate for the exact boundary, which they called “image threshold”, inspired by the excitability of the single neuron. However, we will show later that the parameter a is not an accurate threshold of excitability for a single FitzHugh-Nagumo model in Equation 3.3., and is even less likely to be a reasonable estimate of $\partial\mathcal{B}(\overline{\mathbf{X}})$ for the coupled network.

Indeed, we find it also a difficult work to evaluate the threshold of excitability for a single FitzHugh-Nagumo model. By the term “threshold”, we often refer to a boundary between two states, or a critical value beyond which a qualitative change appears in a system property. Such as what we have observed in Section 3.3.2, a single neuron has two categories of the initial conditions, one of which leas to the resting, the other to the excited state. However, there is no explicit boundary between these two states for the individual monostable system used here, no matter how small ϵ , the ratio of timescales of the fast and slow variables, is, because such a monostable system has a property that its trajectories have a *continuous dependence on its initial conditions* (Palis and de Melo, 1982). Take the FitzHugh-Nagumo model in Equation 3.3 as an example. We use x to denote the vector of state variables $(v, w)^T$ and use $x(t; x_0)$ to denote the solution with the initial condition x_0. Continuous dependence of trajectories $x(t; x_0)$ on its initial conditions x_0 indicates that, for a system which has only one globally stable equilibrium $\overline{x}$, two solutions $x(t; x_0)$ and $x(t; x_0 + \delta x_0)$ could be arbitrarily close, as long as their initial conditions are sufficiently close $\delta x_0 \to 0$. In order to deal with the contradiction, we require to quantify the excitability of a single FitzHugh-Nagumo model in Equation 3.3 by using the *Lyapunov exponent*.

The *Lyapunov exponent* (LE) λ measures the rate of divergence of solutions $x(t; x_0)$ and $x(t, x_0 + \delta x_0)$, for initial conditions x_0 differing by an infinitesimal δx_0. Define $\Phi_t(x_0)$ as the derivative of the trajectory $x(t; x_0)$ with respect to the initial condition x_0:

$$\Phi_t(x_0) := \frac{\partial x(t; x_0)}{\partial x_0}$$

and let $m_1(t), \ldots, m_n(t)$ be the eigenvalues of $\Phi_t(x_0)$. The LEs of x_0 can be defined as

$$\lambda_i := \lim_{t\to\infty} \frac{1}{t} \ln |m_i(t)|, \qquad i = 1, \ldots, n \tag{5.2}$$

5.2.2 Computation of Lyapunov Exponents

In this section, we focus on the single FitzHugh-Nagumo model in Equation 3.3 and evaluate the LEs in order to investigate how the trajectories starting in an infinitesimal neighbourhood of x_0 diverge from each other. We start with a brief introduction of the numerical method of computing LEs used in this thesis. The more detailed method and relevant proofs can be found in some articles and books (Parker and Chua, 1989; Sandri, 1996). We Consider the single FitzHugh-Nagumo system as follows,

$$\dot{x} = f(x), \qquad x \in \mathbb{R}^2 \tag{5.3}$$

The direct way of computing all the Lyapunov exponents $\lambda_i(x_0)$ of this system is to follow the definition in Equation 5.2. In order to evaluate the derivative of the trajectory $\Phi_t(x_0)$ with respect to the initial condition x_0, we need to integrate the Jacobian matrix $D_x f(x) = \frac{\partial f(x)}{\partial x}$ along the solution $x(t; x_0)$ to the system in Equation 5.3 with the initial condition x_0. In order words, to calculate $\Phi_t(x_0)$, we need to solve the combined system as follows,

$$\begin{Bmatrix} \dot{x} \\ \dot{\Phi} \end{Bmatrix} = \begin{Bmatrix} f(x) \\ D_x f(x) \cdot \Phi \end{Bmatrix}, \qquad \begin{Bmatrix} x(t_0) \\ \Phi(t_0) \end{Bmatrix} = \begin{Bmatrix} x_0 \\ I \end{Bmatrix} \tag{5.4}$$

where the first row is the original system in Equation 5.3 and the second row is the linearized equation called *variational equation*. The variational equation can be obtained by differentiating both the sides of Equation 5.3 with respect to x_0 (Parker and Chua, 1989). However, this direct approach is problematic. For large value of t (Parker and Chua, 1989; Sandri, 1996), the columns of $\Phi_t(x)$ will asymptotically align with the eigenvector corresponding to the largest eigenvalue of $D_x f(x)$. So, both the matrix $\Phi_t(x)$ and its eigenvalues $m_i(t)$ will be ill-conditioned especially for the fast-slow dynamical system such as the one in Equation 3.3. Indeed, if applying the direct approach here, one of the eigenvalues m_1 of $\Phi_t(x_0)$ will fast shrink to zero.

So, instead of the definition in Equation 5.2, we are looking at the definition of Lyapunov exponents of order n such as introduced in (Sandri, 1996),

$$\lambda^n(x_0) = \lim_{t\to\infty} \frac{1}{t} \ln[\text{Vol}^n(\Phi_t(x_0))] \tag{5.5}$$

where Vol^n is the n-dimensional volume. And it is shown in (Oseledec, 1968) that

$$\lambda^n = \sum_{i=1}^{n} \lambda_i \tag{5.6}$$

So, based on this relationship, an alternative approach of computing all the λ_i can be introduced. Assume $u_1, \cdots, u_n$ are n column vectors of the matrix $\Phi_t(x_0)$. Integrate the combined system in Equation 5.4 for a relatively small period ΔT from $\Phi_0 = (u_1^{(0)}, \cdots, u_n^{(0)}) = I$ to obtain $\Phi_{\Delta T} = (u_1^{(1)}, \cdots, u_n^{(1)})$. Get the orthonormal $\hat{\Phi}_{\Delta T} = (\hat{u}_1^{(1)}, \cdots, \hat{u}_n^{(1)})$ by using the Gram-Schmidt method as follows,

$$\begin{aligned} p_1^{(1)} &= u_1^{(1)}, \qquad \hat{u}_1^{(1)} = p_1^{(1)}/\|p_1^{(1)}\| \\ p_2^{(1)} &= u_2^{(1)} - \langle u_2^{(1)}, \hat{u}_1^{(1)}\rangle \hat{u}_1^{(1)}, \qquad \hat{u}_2^{(1)} = p_2^{(1)}/\|p_2^{(1)}\| \\ &\cdots \\ p_n^{(1)} &= u_n^{(1)} - \sum_{i=1}^{n-1} \langle u_2^{(1)}, \hat{u}_i^{(1)}\rangle \hat{u}_i^{(1)}, \qquad \hat{u}_n^{(1)} = p_n^{(1)}/\|p_n^{(1)}\| \end{aligned} \tag{5.7}$$

The n-dimensional volume $\text{Vol}^n(\Phi_{\Delta T}(x_0))$ is

$$\text{Vol}^n(\Phi_{\Delta T}) \approx \|p_1^{(1)}\| \cdots \|p_n^{(1)}\|$$

Again, we continue to integrate the combined system from $\hat{\Phi}_{\Delta T}$ of the orthonormalised column vectors for the next time interval ΔT to get $\Phi_{2\Delta T}$. By repeating this integration and orthonormalisation procedure K times, during the k-th step, the increase factor of the volume is

$$\frac{\text{Vol}^n(\Phi_{k\Delta T})}{\text{Vol}^n(\Phi_{(k-1)\Delta T})} \approx \|p_1^{(k)}\| \cdots \|p_n^{(k)}\|$$

Therefore, the n-order Lyapunov exponents λ^n can be obtained as follows,

$$\lambda^n(x_0) = \lim_{K\to\infty} \frac{1}{K\Delta T} \sum_{k=1}^{K} \ln(\|p_1^{(k)}\| \cdots \|p_n^{(k)}\|)$$

And according to the relation provided in Equation 5.6, we can obtain each the Lyapunov exponent λ_i as follows,

$$\lambda_i(x_0) = \frac{1}{K\Delta T} \sum_{k=1}^{K} \ln(\|p_i^{(k)}\|). \tag{5.8}$$

Either too small or too large a value of ΔT will lead to the numerical inaccuracies in computing λ_i. Parker and Chua (1989) suggest using a ΔT that is ten or twenty times the natural period the system. In this thesis, Equation 5.4 is solved with the constant time step $dt = 0.001$. So, we choose T to be 0.01. Moreover, for the system with a globally stable equilibrium $\overline{x}$ such as Equation 3.3, if $\mu_1, \cdots, \mu_n$ are the eigenvalues of the Jacobian at $\overline{x}$, we can obtain

$$\lambda_i(\overline{x}) = \mathrm{Re}(\mu_i), \qquad i = 1, \cdots, n \tag{5.9}$$

As Lyapunov exponent λ_i is defined as the average rate of the diversion or contraction of the nearby trajectories as $t \to \infty$. So, theoretically for nearly all the x_0 Parker and Chua (1989) in the basin $\partial\mathcal{B}(\overline{x})$ of $\overline{x}$, $\lambda_i(x_0) = \lambda_i(\overline{x})$. In other words, any transient progress can be ignored if K is set as a very large integer. In order to maintain the influence of the transient process, we choose $K = 200$ in this thesis.

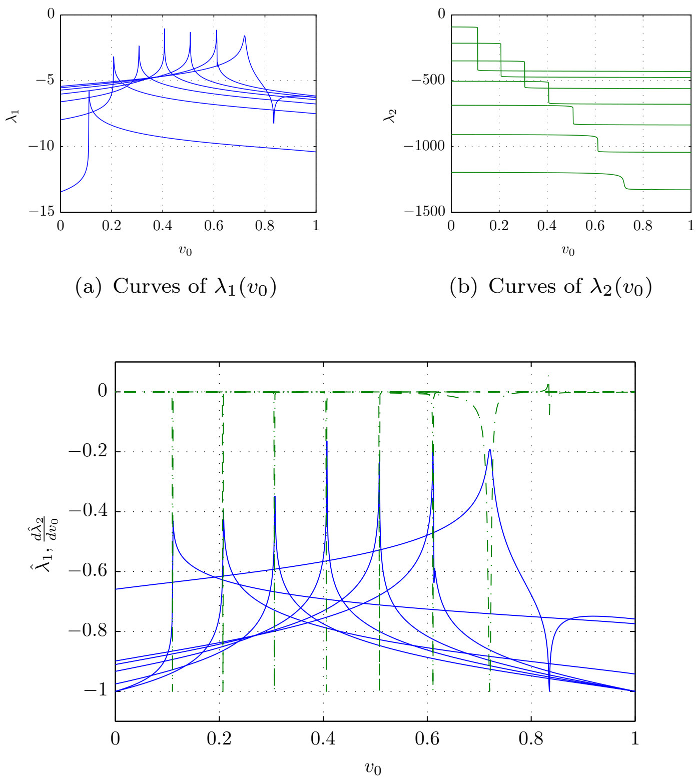
For the detailed parameter settings in simulation, we vary the value of parameter a from 0.1 to 0.7 with step-size $\Delta a = 0.1$ to get different planar systems. For each system, we evaluate the LEs $\lambda = (\lambda_1, \lambda_2)$ for the initial condition $w_0 = 0$, and $v_0 \in [0, 1]$ with step-size 0.001. Again, the ratio of timescales of fast slow variables, ϵ is kept as 0.001. The results are provided in Figure 5.3.

Due to the global stable node at the origin, all the solutions starting from x_0 will finally converge with the others starting from $x_0 + \delta x_0$. So, it can be seen that all the LEs $\lambda(v_0)$ are negative. However, there is a distinct cusp in each curve of $\lambda_1(x)$ in Figure 5.3(a) and a steep drop in each curve of $\lambda_2(x)$, indicating that from the vicinity of specific initial values, the solution trajectories maximally diverge before they finally approach to the stable node as shown in Figure 3.7(a). In order to clearly show the position of initial conditions where the dynamical system maximally changes its behaviour, we plot the other two curves $\hat{\lambda}_1$ and $\frac{d\hat{\lambda}_2}{dv_0}$ together for each system of different a in Figure 5.3(c), where

$$\hat{\lambda}_i = \frac{\lambda_i}{\max\limits_{v_0'}|\lambda_i(v_0')|}, \qquad i = 1, 2$$

The drop in each curve of $\lambda_2(v_0)$ corresponds to the cusp in $\frac{d\hat{\lambda}_2}{dv_0}$, and their positions coincide with those of the cusps in $\hat{\lambda}_1$ as shown in Figure 5.3(c). This identifies the specific initial conditions for which the trajectories are most widely separated for a single neuron as shown in Figure 3.7(a). So, for each system with different settings of parameter a, we define the threshold of excitability θ^* to be the v_0 value at which the Lyapunov exponent λ_1 reaches its peak.

Figure 5.4(a) provides the detailed relationship between a and θ^*. The curve of the function $a(\theta^*)$ is nearly a straight line. We choose the interval $\theta^* \in (0.1, 0.3)$ where the change of curve of the function $a(\theta^*)$ is relatively flat. And we do the linear fitting to obtain a continuous linear function $a : \theta^* \mapsto a$ by least squares method provided in MATLAB. The interpolated linear function is obtained as follows,



(a) Curves of $\lambda_1(v_0)$

(b) Curves of $\lambda_2(v_0)$

(c) Curves of $\hat{\lambda}_1$ and $\frac{d\hat{\lambda}_2}{dv_0}$ versus v_0

Figure 5.3: Lyapunov exponents of the single FitzHugh-Nagumo model with different values of parameter a from 0.1 to 0.7 with step-size $\Delta a = 0.1$. b is constantly chosen as 1 so that all the systems are monostable. Distinct cusps and steep drops are respectively found in the curves of $\lambda_1(x, a)$ (in (a)) and $\lambda_2(x, a)$ (in (b)). And in each pair of λ_1 and λ_2 having the same value of a, these rapid changes coincide at the same horizontal position, as shown in (c), denoting the threshold of excitation (spiking).

$$a = c_1 \cdot \theta^* + c_2, \qquad c_1 \approx 1.02,\ c_2 \approx -0.01 \tag{5.10}$$

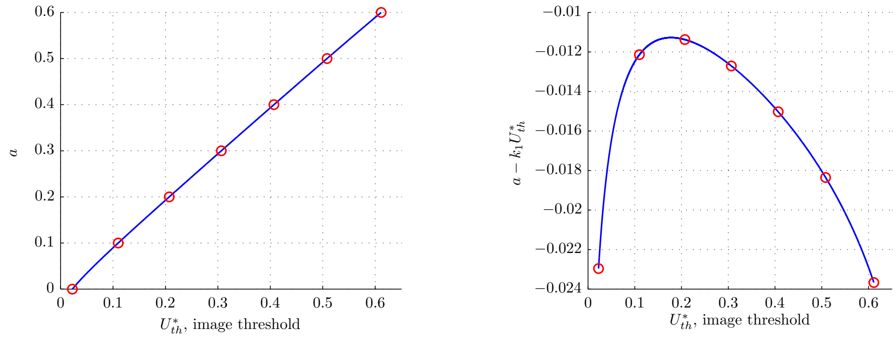
Correspondingly, we will rescale the original image intensity U to the interval $U^r \in (0.1\ 0.3)$ with the expression provided below,

$$U^r = 0.1 + 0.2 \cdot \frac{U}{L} \tag{5.11}$$

where L is the maximum value of image intensities; $L = 255$ for the 8-bit image. It can be seen from Figure 5.4(b) that the real threshold of excitability θ^* is larger than the parameter a. In the application of edge detection, we assign θ^* with the diffused version of processed image $\theta(\tau)$

$$\theta_i^* = \theta_i(\tau). \tag{5.12}$$

So, if simply assume $a = \theta^*$, the input image U^r is required to be sufficiently diffused



(a) Curve of system parameter a versus image threshold θ^*

(b) Curve of $a - c_1\theta^*$ versus image threshold θ^*

Figure 5.4: The quasi-linear relationship between system parameter a and image threshold θ^*.

in order to detect as more edges as possible. Naturally, the wrong edges will be more likely to emerge as discussed in Section 5.1. Later in this thesis, we will use θ to denote both the rescaled image and the threshold of excitability for a single neuron, due to the relation presented in Equation 5.12.

5.2.3 Anisotropic Diffusion

In Section 5.1, the problem of generating wrong pulses with the variable image threshold (Nomura et al., 2011b,a) has been stated. We recall the two reasons which cause the problem. First, the parameter a_i is inaccurately treated as the image threshold θ. Second, the extent of the diffusion is isotropically controlled for all the edges by the same constant d and stopping time τ. Due to the inaccurate settings of parameter a_i, Nomura et al. (2003, 2008, 2011b) adopt a sufficiently diffused image threshold $\theta(\tau)$ in order to detect as many edges as possible. However, if the sufficiently diffused $\theta(\tau)$ is indiscriminately applied for all the edges in image, there may be more neurons in the excited state in addition to the ones corresponding to the correct edge position. So, the wrong pulses will turn up as shown in Figure 5.2. To solve this problem, a new model in Equation 3.26 was introduced by Nomura et al. (2008, 2011b), containing two sub-systems per pixel in Equation 3.17, with the extra copy (v_i^1, w_i^1) is given slightly different dynamics in order to inhibit the generation of wrong pulses of the original system (v_i^0, w_i^0).

We attempt, instead, to keep the size of the system fixed and look for an alternative solution. Specifically, we first use LEs λ to measure the separation between two solutions for nearby initial values and thereby evaluate a more accurate relation between the parameter a and the threshold of excitability θ as in Equation 5.10. Then, since it is the sufficiently diffused image threshold $\theta(\tau)$ that leads to the emergence of a second wrong pulse as discussed previously, the most direct way to inhibit the generation of the wrong pulses is to use a less blurred image (smaller diffusion constant d if the stopping time τ is fixed) as the threshold so that it is more likely to only excite the neuron at the correct edge position while inhibiting the ones around it. In other words, we require different diffusion speeds for different regions of processed image. It is worth pointing out that Perona and Malik (1990) provide a case of using an anisotropic diffusion in edge detection. Their work shows that controlling the extent of diffusion differently according to the magnitude of the gradient of processed image helps improve the performance of processing local details. Following their work, we propose a model of anisotropic diffusion by modifying the diffusion constant in the original discrete diffusion equation in Equation 3.24 as follows:

$$\begin{aligned} \dot{\theta}_i &= d_i \sum_{j \in \mathcal{P}_i} (\theta_j - \theta_i) \\ d_i &= \widetilde{d} \cdot \mathcal{H}(\frac{||\nabla U^r{}_i||}{\max\limits_i(||\nabla U^r{}_i||)} - \eta) \end{aligned} \tag{5.13}$$

where, $\widetilde{d}$ is a small constant. $\mathcal{H}(\cdot)$ is the Heaviside step function. η is a constant threshold. And $||\nabla U^r{}_i||$ is the magnitude of the gradient for $U^r{}_i$ which can be discretised as follows,

$$\begin{aligned} &||\nabla U^r{}_i|| = ||(\frac{\partial U^r{}_i}{\partial x}, \frac{\partial U^r{}_i}{\partial y})|| = \\ &\frac{\sqrt{(U^r{}_{(\mathsf{m+1,n})} - U^r{}_{(\mathsf{m-1,n})})^2 + (U^r{}_{(\mathsf{m,n+1})} - U^r{}_{(\mathsf{m,n-1})})^2}}{2\Delta h} \end{aligned} \tag{5.14}$$

with the same boundary condition in Equation 3.15.

As a result, d_i is no longer constant, but takes on two values $\{0, \widetilde{d}\}$. If the normalized gradient magnitude $\frac{||\nabla U^r{}_i||}{\max(||\nabla U^r{}_i||)}$ is greater than the threshold η, d_i is chosen as $\widetilde{d}$. Otherwise for smaller magnitude of gradients, it is set to 0. In this way, we can inhibit the diffusion of the state of the neurons which do not correspond to regions of large gradients such as edges. By setting the appropriate value of η, the anisotropic diffusion will also contribute to solve the problem of noise sensitivity introduced by adopting the accurate threshold.

Finally, the novel edge detection algorithm using anisotropic diffusion for 8-bit grayscale images can be summarised as in Algorithm 2, for ease of comparison to Nomura's method in Algorithm 1.

Algorithm 2 Binary Diffusion Coefficient

```
Rescale the image intensity distribution $U^r{}_{\mathsf{m,n}} = \xi_1 U_{(\mathsf{m,n})} + \xi_2$ with $\xi_1 = 1/1275$ and $\xi_2 = 0.1$ so that $0.1 \leqslant U^r{}_{(\mathsf{m,n})} \leqslant 0.3$.
Solve the equation below get an anisotropically diffused version of rescaled image $\theta_i(\tau)$ with a stopping time $\tau$,
    $$\dot{\theta}_i = d_i \sum_{j \in \mathcal{P}_i} (\theta_j - \theta_i)$$
    $$d_i = \widetilde{d} \cdot \mathcal{H}(\frac{||\nabla U^r{}_i||}{\max_i(||\nabla U^r{}_i||)} - \eta)$$
    $$||\nabla U^r{}_i|| = ||(\frac{\partial U^r{}_i}{\partial x}, \frac{\partial U^r{}_i}{\partial y})||$$
Evaluate the system parameter $a_i = c_1\theta_i(\tau) + c_2$, where $c_1 \approx 1.02$ and $c_2 \approx -0.01$.
Solve the new model equation below,
    $$\begin{cases} \dot{v_i} = f(v_i, w_i, a_i) \\ \dot{w_i} = g(v_i, w_i) + k_w \sum_{j \in \mathcal{P}_i} (w_j - w_i) \end{cases}$$
    with the initial condition $(v_i(0), w_i(0)) = (U^r{}_i, 0)$ and the zero boundary conditions to get the steady state solution in $v$, which is $v_i(\tau_s)$.
Threshold $v_{(\mathsf{m,n})}(\tau_s)$ to get the final binary edge map $\mathcal{M}_{(\mathsf{m,n})}$.
    $$\mathcal{M}(\mathsf{m,n}) = \begin{cases} 1, & v_i(\tau_s) > 0.5 \\ 0, & v_i(\tau_s) \leqslant 0.5 \end{cases}$$
```

5.3 Results

This section applies Algorithm 2 proposed in the previous section to detect the edges in the test images, which are previously introduced in Section 2.3, including both artificial and real ones. We also apply the same test images to Nomura's method (Nomura et al., 2011b) in Algorithm 1 so that we can provide a comparison between these two algorithms. Specifically for testing the noise immunity of the algorithm, we present a robustness test in comparison to the results obtained from purely gradient magnitude map and Canny detector.

Artificial Images

The basic structure of artificial image has been introduced in Section 2.3. Moreover, in order to test the applicability of the two algorithms on different ranges of original image intensities, we also apply them on a lighter and a darker version of the original artificial image as respectively shown in Figure 5.5(c) and Figure 5.5(d). And both are of the same size [303 × 404] as the original artificial image.

In order to evaluate the edge detection performance, the edge images obtained with the two algorithms are compared with the ground-truth data of edges as shown in Figure 5.5(b). Specifically in order to quantify the performance of edge detection of algorithms for artificial images, we will adopt the following four measures, the true positive tp, the percentage of the true positive tp_r, the false positive fp and the percentage of false positive fp_r.

We present the accuracy of edges detected using Algorithm 1 (Nomura's method) in Figure 5.6(a), Figure 5.6(c) and Figure 5.6(e), which are to be contrasted with those obtained by Algorithm 2 in Figure 5.6(b), Figure 5.6(d) and Figure 5.6(f). Table 5.1 shows the comparison with the ground-truth edge information in Figure 2.2(b). Because both the algorithms using the class of dynamical networks choose the higher of the two intensity levels that change appreciably across an edge as the location of edge, that is what we identify. So, following Nomura et al. (2011b), we also allow for an error of one pixel shift when calculating the true positive and the false positive.

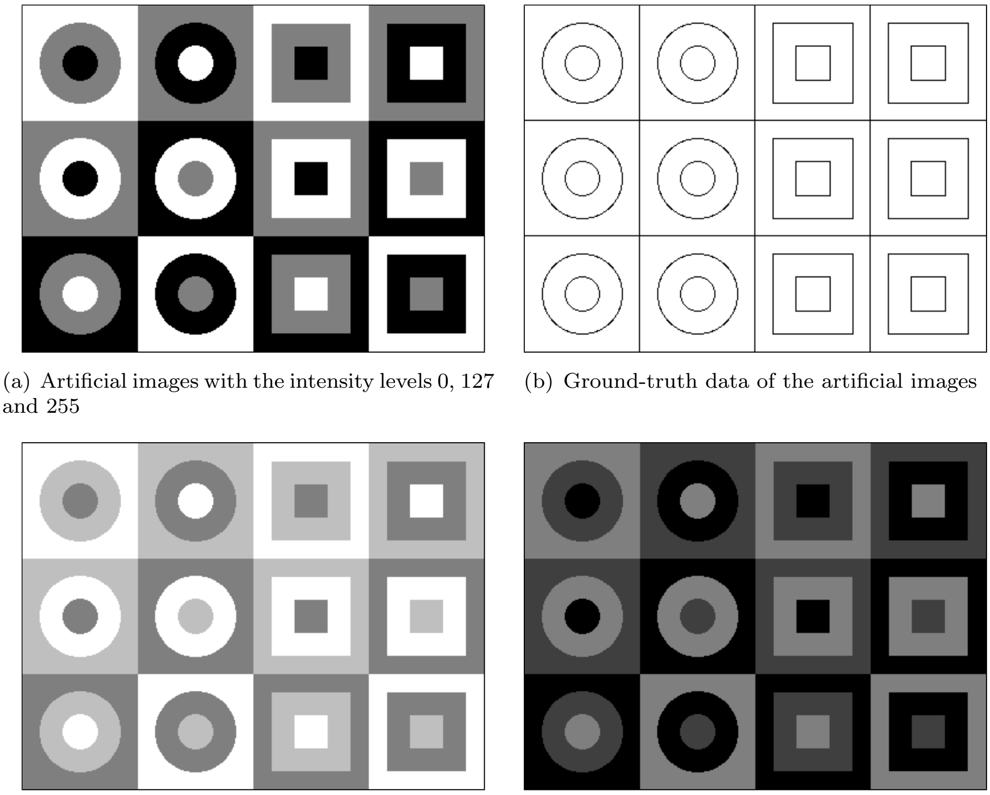
	Fig. 5.5(a)		Fig. 5.5(c)		Fig. 5.5(d)	
	Algo#1	Algo#2	Algo#1	Algo#2	Algo#1	Algo#2
tp (No. of Pixels)	11340	5758	11230	5758	11057	5758
tp_r (%)	97.08%	98.37%	97.15%	98.37%	97.03%	98.37%
fp (No. of Pixels)	1878	0	2080	0	2039	0
fp_r (%)	1.61%	0%	1.79%	0%	1.75%	0%

Table 5.1: Quantitative evaluations of edge detection algorithms

It is clear that the results on edges detected by Algorithm 2 improve upon those detected using Algorithm 1 (Nomura et al., 2011b). Specifically, the rate of true positive tp_r is increased nearly 1.2% and the rate of false positive decrease to 0%. It is also worth noting that the true positives obtained with Algorithm 1 number nearly twice as those from Algorithm 2. The reason is that the Nomura algorithm merges the two edge maps obtained respectively from the original and the inverted images and we allowed one pixel error when calculating the these measures so that most of the true positive pixels in one edge map would be the neighbour of the ones from the other map. As a result, the edges obtained with the Algorithm 1 will normally be as twice thick as the ones with our algorithm.

Since it is generally accepted (Canny, 1986) that thinner edges are better than thick ones, this is another indicator of how Algorithm 2 improves upon Algorithm 1.

We would also like to make a remark concerning the apparent lack of symmetry in Figure 5.6(a), Figure 5.6(c) and Figure 5.6(e). Although each square of the artificial image is symmetrical, we test the algorithms by applying the whole image [303 × 404] consisting of 12 squares. This tableau of 12 squares is no longer symmetric with respect to spatial reflections and rotations, which explains the asymmetry of the erroneous edge patterns in Figure 5.6(a), Figure 5.6(c) and Figure 5.6(e).



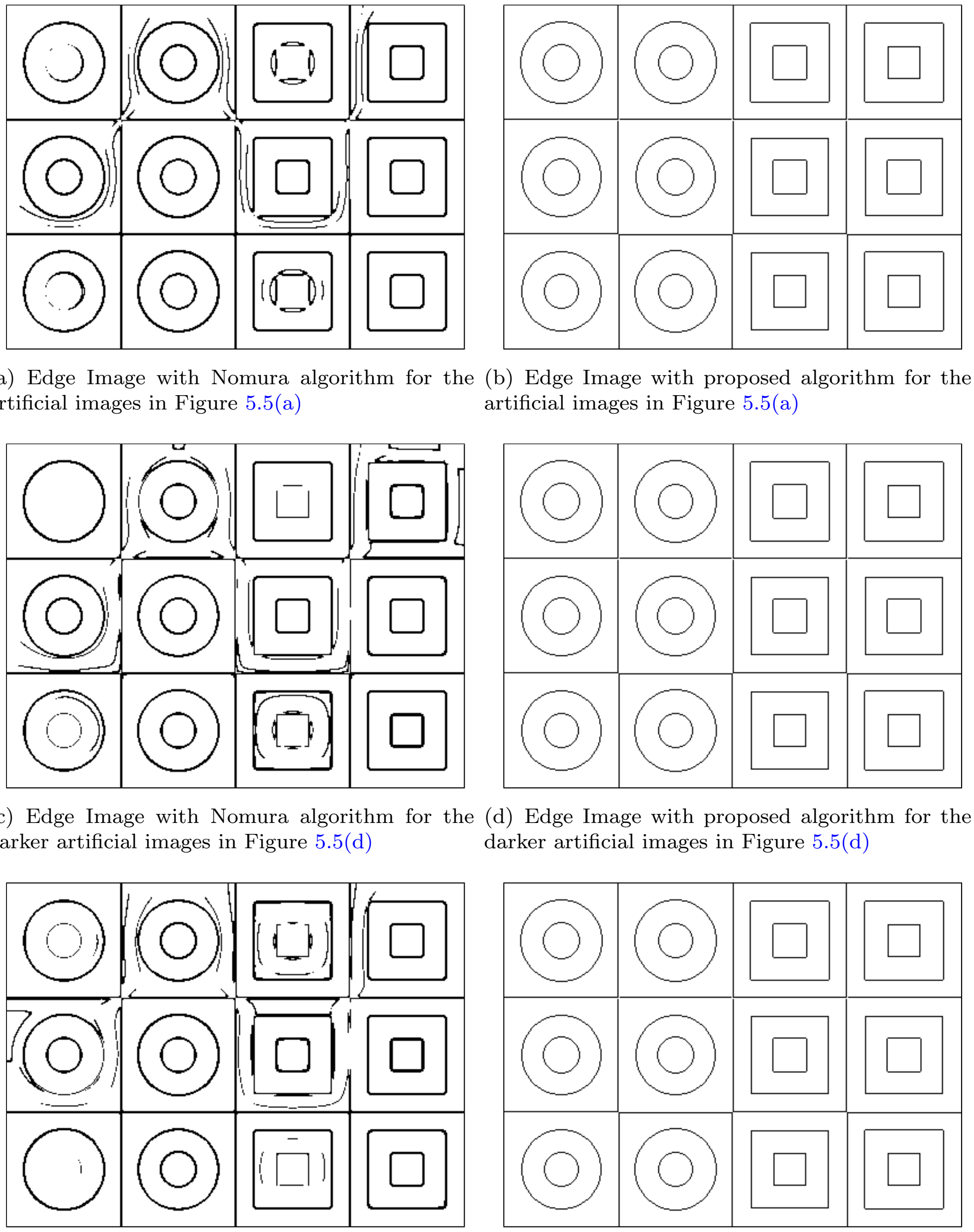
(a) Artificial images with the intensity levels 0, 127 and 255

(b) Ground-truth data of the artificial images

(c) Artificial images with the lighter intensity levels 127, 191 and 255

(d) Artificial images with the darker intensity levels 0, 63 and 127

Figure 5.5: Artificial images and the corresponding ground truth of the edge data.



(a) Edge Image with Nomura algorithm for the artificial images in Figure 5.5(a)

(b) Edge Image with proposed algorithm for the artificial images in Figure 5.5(a)

(c) Edge Image with Nomura algorithm for the darker artificial images in Figure 5.5(d)

(d) Edge Image with proposed algorithm for the darker artificial images in Figure 5.5(d)

(e) Edge Image with Nomura algorithm for the darker artificial images in Figure 5.5(c)

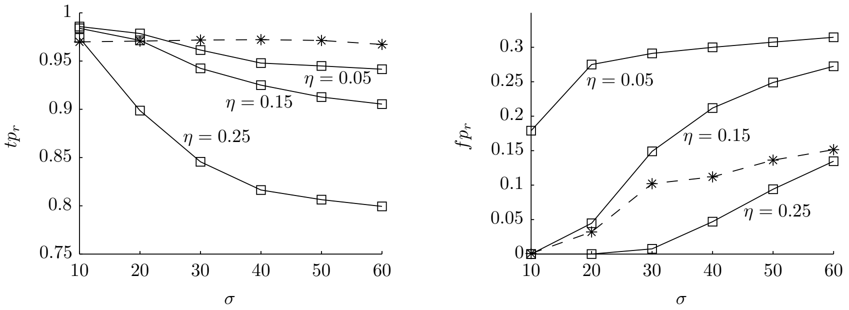
(f) Edge Image with proposed algorithm for the darker artificial images in Figure 5.5(c)

Figure 5.6: Edge detection results for artificial images. For the Nomura method, the system parameters are set as $b = 1$, $\epsilon = 0.001$, $k_v = 4$ and $k_w = 20$, and the two diffusive constants $d^0 = 40$ and $d^1 = 200$ in Equation 3.24. Both the diffusion stop times are $\tau = 1.0$. For the proposed method, the system parameters are set as $b = 3.5$, $\epsilon = 0.001$ and $k_w = 5$. The constant $K = 10$ and the threshold $\eta = 0.0$ in Equation 5.13. The steady state time $\tau_s = 1.0$ in the model equations for both algorithm.

Noise Robustness Test

In order to illustrate the usefulness of Algorithm 2, we run a noise robustness test based on the same set of artificial images. As a result, we provide a model comparison to Canny algorithm (Canny, 1986) and the method using only thresholded magnitude of local gradients, in terms of true positive rate and false positive rate as the standard deviation of the noise in the images is increased.

The intensity levels of the noise-free artificial image are 63, 127 and 191, to which we add white noise with a standard deviation σ in the range from 10 to 60. For each noisy image, the value of threshold η in Equation 5.13 is respectively set as $0.05, 0.15, 0.25$. All the measures are obtained by averaging over 50 simulations and the results are provided in Figure 5.7 and Figure 5.8.

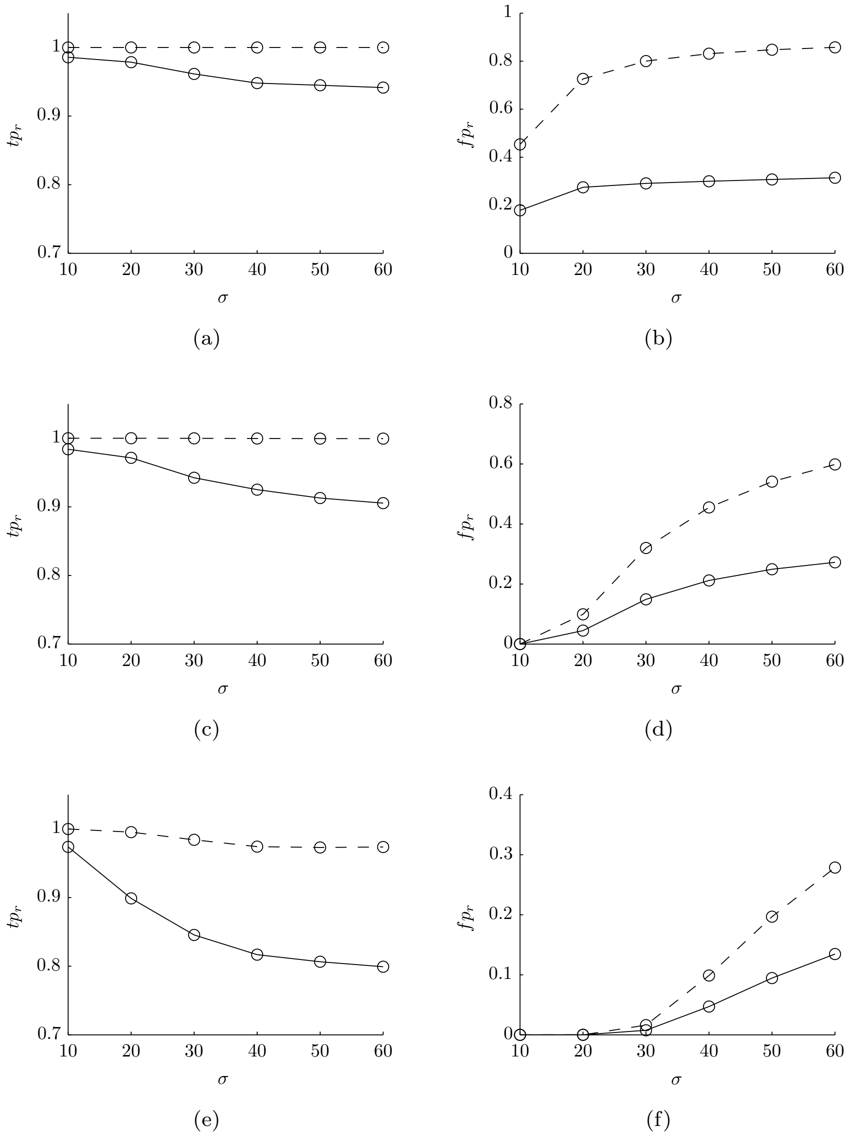


(a) (b)

Figure 5.7: Robustness test of Algorithm 2 in comparison to Canny detector. The value of measures are taken as the average of 50 simulations.

It can be seen from Figure 5.7 that, for Canny detector, the average value of tp_r remains almost constant at around 0.965, while the average value of fp_r increases with increasing σ. For our method, although the tp_r curves start at relatively higher value compared to those of the Canny detector when $\sigma = 10$, they decrease upon increasing η. Specifically, when $\eta = 0.25$ and $\sigma = 60$, the average value of tp_r for Algorithm 2 decreases to 80%. The fp_r curves decrease appreciably when the larger value of η is chosen. Specifically, when $\eta = 0.25$, we obtain the lower fp_r by Algorithm 2 than that of Canny detector.

The procedure of denoising in Algorithm 2 is the binarisation on the diffusion parameter $\widetilde{d}$, which is determined by the thresholded magnitude of gradient as shown in Equation 5.13. In fact, the thresholded magnitude of gradient can also been seen as a denoised edge map from the processed image. In order to compare the results obtained by Algorithm 2 and merely the thresholded magnitude of gradient, we run the noise robustness test on the same sets of noisy images using these two methods and present the resultant measures in Figure 5.8. It can be seen from Figure 5.8(b),Figure 5.8(d) and Figure 5.8(f) that all the curves of fp_r reduce upon increasing η. Specifically, the results obtained by Algorithm 2 always contains much fewer false positives than that



(a) (b)

(c) (d)

(e) (f)

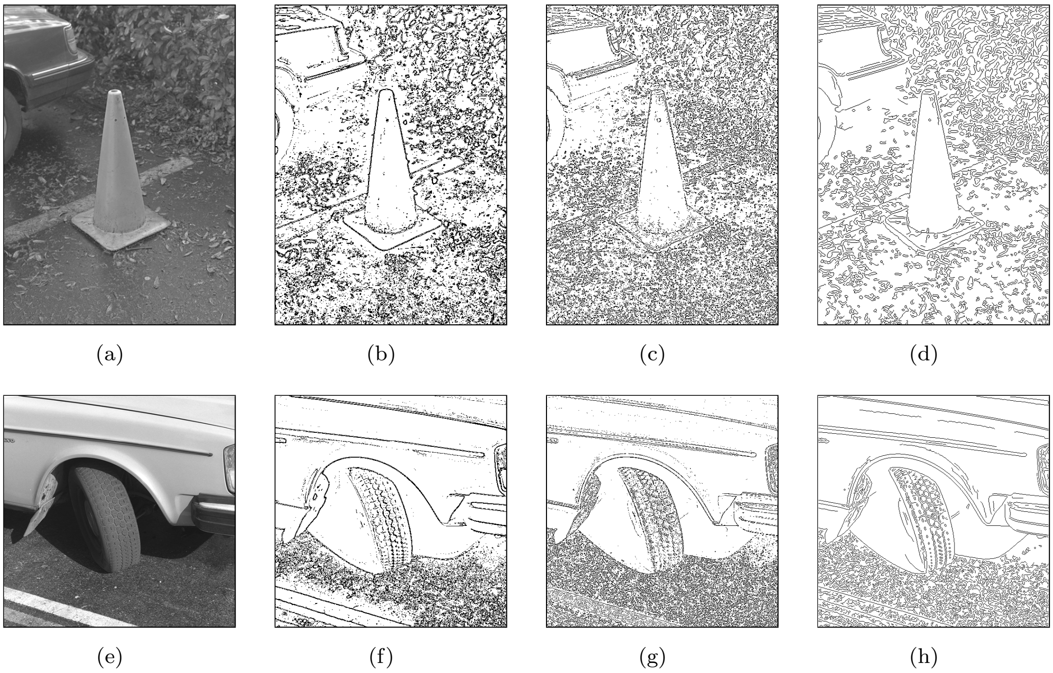
Figure 5.8: Robustness test of Algorithm 2 in comparison to the method using pure local gradient magnitude with different values of η in Equation 5.13. In each figure, the dashed line stands for the method using local gradient magnitude and the solid line stands for Algorithm 2. $\eta = 0.05$ for 5.8(a) and 5.8(b). $\eta = 0.15$ for 5.8(c) and 5.8(d). $\eta = 0.25$ for 5.8(e) and 5.8(f). The false positives become much fewer when value of η increases for both algorithms. Algorithm 2 has largely reduced fp_r curves with slight loss of tp_r. The value of measures are taken as the average of 50 simulations.

of thresholded magnitude of local gradient, with the same value of η. The high tp_r in Figure 5.8(a),Figure 5.8(c) and Figure 5.8(e) is not very convincing due to the large number of false positives. In a conclusion, according to these results, we can state that the performance of Algorithm 2 is better than Algorithm 1 and comparable with that of the Canny detector.

Real Images

Furthermore, we test Algorithm 2 for real grayscale images in comparison with Algorithm 1 (Nomura's method) and Canny detector . In Figure 5.9 and Figure 5.10, we present the results respectively obtained by Algorithm 1, Algorithm 2 and Canny detector along a row of the table with the original image as the left-most entry. All the real images are previously introduced in Section 2.3 and they were also used by Nomura et al. (2011a,b).

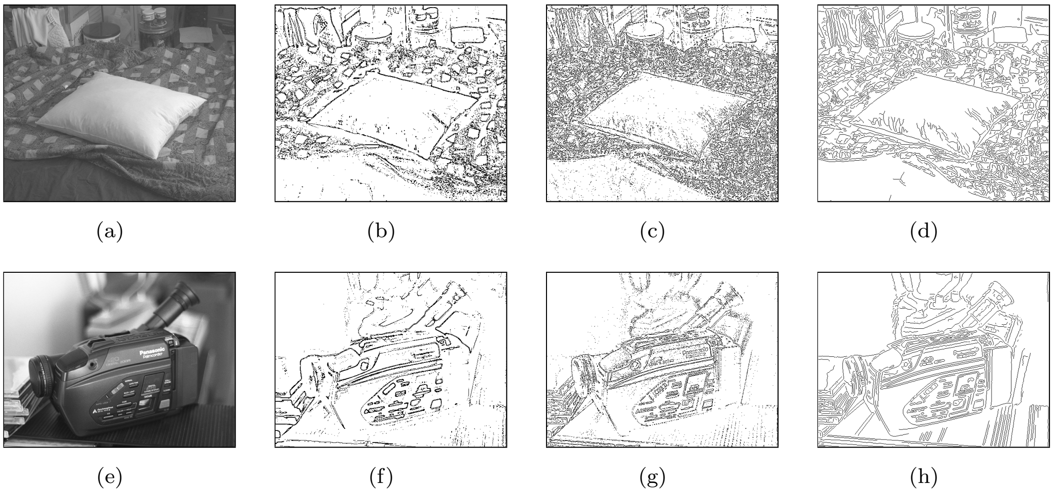
We find that Algorithm 2 is able to pick up more details than Algorithm 1 as shown in Figure 5.10(g) compared with Figure 5.10(f) and these details can also be clearly found in the result given by Canny detector as shown in Figure 5.10(h). We attribute this success to the application of a more accurate relation between the threshold of excitability θ and the system parameter a as in Equation 5.10. However, this increased accuracy makes it relatively vulnerable to background noise in real images, especially illustrated by the detection results for the ground in Figure 5.9(a) and Figure 5.9(e). Here, we choose the threshold for anisotropic diffusion uniformly as 0.05 for real images and its value should be appropriately chosen for each real image. The Canny detector shows the best balance between the accuracy and the noise immunity among the three methods. In this chapter, we aim to present the basic edge detection network in Equation 3.17 which has the comparable edge detection ability (the balance between the noise immunity and the accuracy) to Nomura's method (Nomura et al., 2011a) without doubling the network size. Since the main motivation of this chapter is to explore how dynamical theory may be used to tailor the design phase of a silicon circuit, this halving of network size compared with (Nomura et al., 2011a,b) would be of great benefit to that goal.



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

(e) (f) (g) (h)

Figure 5.9: Edge detection results for real images, (a) Traffic Cone [604×437]. (e) and Tire [512×512]. (b) and (f) provide the edge detection results obtained by adopting the Nomura method (Nomura et al., 2011a), in which the system parameters are $b = 1$, $\epsilon = 0.001$, $k_v = 4$ and $k_w = 20$, and the two diffusive constants in Equation 3.24 for the diffusive a_i^0 and a_i^1 are respectively set as $d^0 = 40$ and $d^1 = 200$. Both the diffusion stop times are $\tau = 1.0$ and the steady state time $\tau_s = 1.0$ for the model equation Equation 3.26. (c) and (g) provide the edge detection results with the proposed method in Algorithm 2 using anisotropic diffusion, in which the system parameters are $b = 3.5$, $\epsilon = 0.001$, $k_v = 0$ and $k_w = 5$. The constant $\widetilde{d} = 10$ and the threshold $\eta = 0.05$ in Equation 5.13 for all the images. And the anisotropic diffusion stopping time is $\tau = 1.0$. The steady state time $\tau_s = 1.0$ for the model in Equation 4.12. ((d)) and ((h)) are the results obtained by the Canny's algorithm where the thresholds are automatically determined by the program provided in MATLAB.



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

(e) (f) (g) (h)

Figure 5.10: Edge detection results for real images, (a) Pillow [468 × 552]. (e) Video camera [435 × 577]. (b) and ((f) provide the edge detection results obtained by adopting the Nomura method (Nomura et al., 2011a) ,in which the system parameters are $b = 1$, $\epsilon = 0.001$, $k_v = 4$ and $k_w = 20$, and the two diffusive constants in Equation 3.24 for the diffusive a_i^0 and a_i^1 are respectively set as $d^0 = 40$ and $d^1 = 200$. Both the diffusion stop times are $\tau = 1.0$ and the steady state time $\tau_s = 1.0$ for the model equation Equation 3.26. ((c)) and ((g)) provide the edge detection results with the proposed method in Algorithm 2 using anisotropic diffusion, in which the system parameters are $b = 3.5$, $\epsilon = 0.001$, $k_v = 0$ and $k_w = 5$. The constant $\widetilde{d} = 10$ and the threshold $\eta = 0.05$ in Equation 5.13 for all the images. And the anisotropic diffusion stopping time is $\tau = 1.0$. The steady state time $\tau_s = 1.0$ for the model in Equation 4.12. (d) and ((h)) are the results obtained by the Canny's algorithm where the thresholds are automatically determined by the program provided in MATLAB.

Chapter 6

Network Stability

The original description (Nomura et al., 2011b) on the system in Equation 3.17 points out that, by appropriately setting its initial conditions based on the images of interest, the system will self-organize spatio-temporal patterns and quickly exhibit stationary spatial pulses as an edge detection result. In contrast, many edge detection methods using other reaction diffusion systems, as mentioned in (Nomura et al., 2011b), are unable to maintain a stationary output. Thereby, additional steps to compute stopping times are necessary for those algorithms. Hence, Nomura et al. (2011a) claimed the network stability as a main contribution as it facilitates the complexity of the design. However, their work is still somehow incomplete. Firstly, recall that in Nomura's algorithm, the parameter a_i is determined by the diffused image θ_i based on the relation below,

$$a_i = \theta_i(\tau).$$

$\theta(\tau)$ is one frame of the time-varying solution to the diffusion equation in Equation 3.23 which we rewrite as following,

$$\dot{\theta}_i = d \sum_{j \in \mathcal{P}_i} (\theta_j - \theta_i)$$

As illustrated in both Chapter 3 and Chapter 5, τ determines the extent of diffusion of an input image, once the value of d is fixed. Thereby, in Algorithm 1 as well as in Algorithm 2, although no stopping time is required for the vector of state variables (v_i, w_i), a time τ still needs to be computed in order to stop the diffusion in θ. Secondly, the previous work done by Nomura et al. (2011b) failed to elaborate the relation between the system parameters and the stability of the output.

Hence, this goal of chapter is firstly to find out an alternative way to stop the diffusion process in Equation 3.23 instead of computing τ, and secondly, to discuss how to set the system parameters a_i, b_i and coupling strength k_i based on the stability analysis on the network system. Section 6.1 introduces a mechanism to the original diffusion so that the dynamics can be automatically stopped. Specifically, the idea originates from the OU process (OrnsteinUhlenbeck process) (Uhlenbeck and Ornstein, 1930), which is a mathematical model to describe the Brownian motion under the influence of friction. In contrast to the Wiener process which corresponds to the pure diffusion, OU process includes a mean-reverting tendency so that it is possible for a Brown particle to move back to the central location (starting point), admitting a stationary probability distribution. Section 6.2 focuses on the analysis on the condition of Hopf bifurcation among the network with the weak couplings. Section 6.3 provides detailed examples to how the parameter settings of the uncoupled neurons are related to the coupling strength, in order for the attractors representing expected edge patterns to be stable. Finally, based on all the analysis, Section 6.4 presents a edge detection method using anisotropic coupling strength k_i which is controlled by the local gradient magnitude. Edge detection results for both artificial and real images are provided. Moreover, the robustness test is also carried out in comparison to Algorithm 2.

6.1 Self-Stopping Diffusion

In Chapter 5, we have shown that how edges are detected among the multiple intensity levels by using a diffused version of the processed image $\theta(\tau)$. Recall that in both the previous algorithms, $\theta(\tau)$ is obtained by solving a continuous diffusion equation in Equation 3.23. Figure 6.1 presents an illustrative example in which diffusion is considered only in one dimension, namely, a chain of elements. Generally, diffusion smooths step changes of intensity levels in an image. In Chapter 5, we have illustrated that the wrong edge pulses will turn up when a sufficiently diffused image $\theta(\tau)$ is used. Hence, we do not want the diffusion process to run indefinitely. As discussed in Section 3.4.2 that, we can choose a stopping time τ to control how smooth the result is, if the diffusion

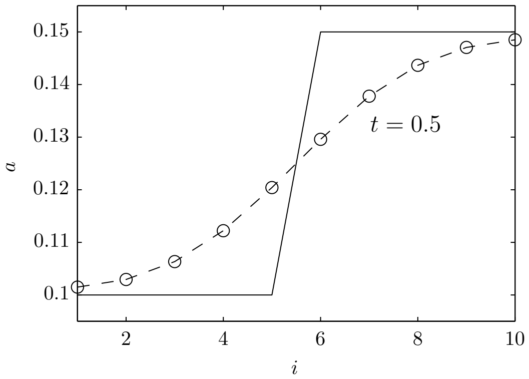


Figure 6.1: The shape of the diffusion of a step as the solution to Equation 3.23 when $\tau = 0.5$. The step function set is $\theta_{i0}(i \geqslant 6) = 0.15$ and $\theta_{i0}(i < 5) = 0.1$. All the solutions $\theta_i(\tau)$ will be smoothed versions of the step. Specifically, the larger τ is, the more smooth the step is.

coefficient D in Equation 3.23 is always constant. Indeed, the stopping time is also set in Algorithm 1 by Nomura et al. (2011a,b) to stop the diffusion, although they emphasizes the contribution of their method that no stopping time is required for the network model in Equation 3.17. In our case of application to a hardware implementation, the computation for the solution and the additional stopping time increases the complexity of the off-chip design. Hence, we prefer an alternative method with which the diffusion process can be stopped dynamically by itself. In this section, we will proposed a method that evolves the diffusion process to have an appropriate asymptotically stable solution.

Recall the discrete diffusion equation in Equation 3.24 which we rewritten as

$$\dot{\theta}_i = d\sum_{j\in\mathcal{P}_i}(\theta_j - \theta_i), \quad \theta_i(0) = U^r{}_{(\mathsf{m},\mathsf{n})}$$

where $d = D/\Delta\mathsf{h}^2$ is a constant which is proportional to D and $\mathcal{P}_i$ is a set consisting of the coordinates of all the neighbour pixels of the ith pixel. Here,

$$\mathcal{P}_i = \{i-1, i+1\}$$

The discrete diffusion equation illustrates that, at a time t, the rate of changing in θ at the specific position i is locally determined by the sum of the difference between θ_i and its neighbourhood θ_{i-1} and θ_{i+1}. The rate of changing in θ_i will be zero when there is no difference between the nearby intensity levels. Therefore, the derivatives $\dot{\theta}_i$ for $i > 6$ or $i < 5$ is zero at the beginning. So, as clearly shown in Figure 6.1, the values of θ_i near to the step edge changes much larger than those far from the step edge. If the diffusion continue to run as τ goes to infinity, then final state of θ will be a constant value.

Hence, we need to add an another term to the discrete diffusion equation in Equation 3.24 in order to draw the diffused intensity θ_i back to the initial place, once it starts to deviate. Basically, the restriction of state variable θ_i to its initial value θ_{i0} can be expressed by the following dynamics,

$$\frac{d\theta_i}{dt} = -(\theta_i - \theta_{i0}). \tag{6.1}$$

For this equation, the derivative $\dot{\theta}_i$ is determined by the deviation of $\theta_i(t)$ from the initial value θ_{i0}. The further the deviation is, the faster it is going back. Finally, any state of θ_i will finally converge to the fixed value θ_{i0}. By adding 6.1 to the diffusion equation in Equation 3.23, we obtain,

$$\frac{d\theta}{dt} = D\nabla^2\theta_i - \widetilde{D}(\theta - \theta(0)) \tag{6.2}$$

where $\widetilde{D}$ is a non-zero constant and if both sides of the equation is divided by $\widetilde{D}$, we obtain

$$\frac{d\theta}{d(\widetilde{D}t)} = \frac{D}{\widetilde{D}}\nabla^2\theta - (\theta - \theta(0)) \tag{6.3}$$

Let $\widetilde{t} = \widetilde{D}t$ and $\xi = \frac{D}{\widetilde{D}}$. The equation becomes

$$\partial_t \theta = \xi \nabla^2 \theta - (\theta - \theta(0)) \tag{6.4}$$

The equation above illustrate that the state variable θ_i will be drew back to the starting point by the new added term $\theta - \theta(0)$ once there is difference between the current state $\theta_i(t)$ and the initial condition θ_{i0}. Hence, the diffusion process somehow still remember the initial state. Assume the initial condition θ_0 is a step expressed by the Heaviside function $A\mathcal{H}(x - x_0)$. Solving the partial differential equation in Equation 6.4 gives

$$\theta(x) = C_1 e^{\frac{x}{\sqrt{\xi}}} + C_2 e^{-\frac{x}{\sqrt{\xi}}} - \frac{1}{2} A e^{-\frac{x+x_0}{\sqrt{\xi}}} \left(e^{\frac{x}{\sqrt{\xi}}} - e^{\frac{x_0}{\sqrt{\xi}}} \right)^2 \mathcal{H}(x - x_0) \tag{6.5}$$

By setting the following two boundary conditions,

$$\theta(x \to \infty) = A, \quad \theta(x \to -\infty) = 0$$

we evaluate the constants in Equation 6.5,

$$C_1 = \frac{1}{2} e^{-\frac{x_0}{\sqrt{\xi}}}, \quad C_2 = 0 \tag{6.6}$$

So, the solution $\theta(x)$ to Equation 6.4 will be,

$$\theta(x) = \frac{1}{2} e^{\frac{x-x_0}{\sqrt{\xi}}} + A\mathcal{H}(x - x_0)(1 - \cosh\left(\frac{x - x_0}{\sqrt{\xi}}\right)) \tag{6.7}$$

The discrete form of Equation 6.4 can be expressed as below,

$$\dot{\theta}_i = \xi \sum_{j \in \mathcal{P}_i} (\theta_j - \theta_i) - (\theta_i - \theta_{i0}) \tag{6.8}$$

Again we run the equation with the same step function and the result chosen as the steady state in Equation 6.8 for each value of ξ is provided as below,

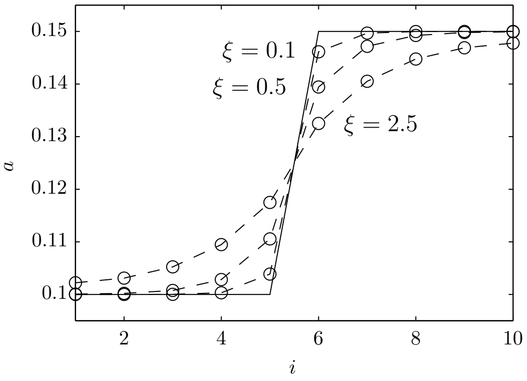


Figure 6.2: Control of the shape of the diffusion with the parameter ξ in Equation 6.8. When ξ is small the result is near to the original step.

It can be clearly figured out that when ξ is large, the diffusion part in Equation 6.8 dominates and the resultant solution will be sufficiently smoothed as the diffusion process. While ξ is small, the result will be more near to the original step.

6.1.1 Modified Model Equations

We previously eliminate the coupling through the state variable v ($k_v = 0$) due to the very small value of ϵ. Here, based on the analysis provided above, we find an alternative way of stopping the diffusion with a controllable parameter ξ. So, we proposed a modified model for edge detection as below,

$$\begin{cases} \epsilon \dot{v}_i = v_i(v_i - a_i)(1 - v_i) - w_i \\ \dot{w}_i = v_i - b_i w_i + k_i \sum\limits_{j \in \mathcal{P}_i} (w_j - w_i) \\ \dot{\theta}_i = \gamma(\xi \sum\limits_{j \in \mathcal{P}_i} (\theta_j - \theta_i) - (\theta_i - \theta_{i0})) \end{cases} \qquad a_i(t) = c_1 \theta_i(t) + c_2 \tag{6.9}$$

It can be seen that updates in θ_i are independent of v_i and w_i. γ is a large constant,

$$\gamma \ll \epsilon^{-1}$$

One can image that when γ is large enough, the variable θ_i reaches its steady state very quickly so that the equation can be independently treated. For the sake of convenience in computer simulation, the third equation in Equation 6.9 will still be processed separately from the first two in the following examples. The simulations on edge detection on both artificial and real images are provided later in Section 6.4.

6.2 Stability of Multi-Attractor Type Network

According to the theory of WCNN, we know that the uncoupled neuron is required to be set near to the bifurcation point in order for nonhyperbolicity of high dimensional equilibrium $\overline{\mathbf{X}}$ of the coupled network. Through the analysis provided in the previous chapter, we find that the behaviours of the network system will undergo two bifurcations when couplings are included, if the system parameters a_i and b_i are properly set. Assume that we start with a relatively small value in k_{wi} and gradually increase it, the system will first experience a saddle-node bifurcation and then a Hopf bifurcation. Newly appearing high dimensional equilibria become attractors only when k_i continues to increase until they go beyond the Hopf bifurcation. So if ϵ is fixed and takes an infinitesimal value, stability of the attractors are influenced by all these parameters a_i, b_i and k_i. As we know, a_i is determined by the diffused image θ which is controlled by the variance diffusion parameter ξ.

According to the WCNN theory, each single neuron in the network before including the coupling requires to be set near to the Hopf bifurcation point. Once both a_i and b_i are set, the stability of the attractors is only determined by the coupling strength. Hence, the boundary, beyond which a specific high dimensional equilibrium representing an edge detection result becomes an attractor, can be expressed as

$$\beta(\mathbf{k}) = 0 \tag{6.10}$$

where $\beta : \mathbb{R}^{2\mathsf{MN}} \to \mathbb{R}$ and $\mathbf{k} = (k_1, k_2, ..., k_{2\mathsf{MN}})$. The goal of this section is to estimate this boundary in order to appropriately set the coupling strength. As mentioned in the previous chapter, the coupled network has many attractors. In order to pick out the correct edge detection result, the previous methods only control the boundary of attraction domain by diffusing the image and assigning the resultant value to parameter a_i. If stability of the attractors can be ideally controlled by the coupling strength k_i, we can reduce the number of the spurious attractors by unstabilising them. In other words, the chance for the initial condition to converge to the correct attractor will increase. This can be achieved by designing a pattern of k_i according to the magnitude of the local gradient of the input image.

6.2.1 Analysis on Jacobian: Hopf Bifurcation in Coupled Network

Denote the Jacobian of the whole network as $\mathbf{J} \in \mathbb{R}^{2\mathsf{MN}\times 2\mathsf{MN}}$. If the coupling is not included, then $\mathbf{J}$ is a block-diagonal matrix,

$$\mathbf{J} = \mathbf{J}_u = \bigoplus_i J_i, \tag{6.11}$$

where J_i is the Jacobian of a single neuron in Equation 3.5 which we rewrite as following,

$$J_i = \begin{pmatrix} -\epsilon^{-1}(3v_i^2 - 2(a_i+1)v_i + a_i) & -\epsilon^{-1} \\ 1 & -b_i \end{pmatrix}$$

J_i can be considered as a 2×2 sub-matrix in the diagonal of $\mathbf{J}$. The relation below is satisfied,

$$J_i U_i = U_i \Lambda_i \tag{6.12}$$

where U_i is the matrix with columns given by eigenvectors of J_i and Λ_i is the matrix whose diagonal elements are eigenvalues of J_i. This relation also holds for $\mathbf{J}$,

$$\mathbf{J}\mathbf{U} = \mathbf{U}\boldsymbol{\Lambda} \tag{6.13}$$

The eigenvalues of $\mathbf{J}$ can be found as the roots of characteristic equation of $\mathbf{J}$

$$\det(\mathbf{J} - \mu I) = \prod_i \det(J_i - \mu I) = 0 \tag{6.14}$$

It can be clearly seen that any eigenvalue λ_k of J_i must satisfy the equation (6.14). Hence, the eigenvalue matrix of whole uncoupled network $\mathbf{\Lambda}$ can be expressed as below,

$$\mathbf{\Lambda} = \bigoplus_i \Lambda_i \tag{6.15}$$

Consequently,

$$\mathbf{U} = \bigoplus_i U_i \tag{6.16}$$

Assume that

$$J_i = \begin{pmatrix} J_{k,k} & J_{k,k+1} \\ J_{k+1,k} & J_{k+1,k+1} \end{pmatrix}, \quad k = 2i - 1$$

we know that the trace of a matrix equals to the sum of all its eigenvalues. Hence,

$$J_{k,k} + J_{k+1,k+1} = \lambda_k + \lambda_{k+1} \tag{6.17}$$

If the ith neuron is undergoing a Hopf bifurcation by increasing the parameter b_i to its bifurcation point, then the eigenvalues λ_k and λ_{k+1} are a pair of purely imaginary conjugated eigenvalues. Correspondingly, J_i is traceless, namely

$$J_{k,k} + J_{k+1,k+1} = 0.$$

Theorem 6.1. *Assume there is no repeated eigenvalue λ for the uncoupled Jacobian $\mathbf{J}_u$, where*

$$\mathbf{J}_u = \bigoplus_i J_i,$$

and

$$J_i U_i = U_i \begin{pmatrix} \lambda_k & \\ & \lambda_{k+1} \end{pmatrix}, \quad k = 2i - 1. \tag{6.18}$$

Consider the change in the ith 2×2 diagonal block of $\mathbf{J}_u$ as the ΔJ_i,

$$\Delta J_i = \begin{pmatrix} \Delta J_{k,k} & \Delta J_{k,k+1} \\ \Delta J_{k+1,k} & \Delta J_{k+1,k+1} \end{pmatrix}, \quad k = 2i - 1$$

and with the corresponding changes of eigenvalues λ_k and λ_{k+1}, then the relation

$$\Delta\lambda_k + \Delta\lambda_{k+1} = \Delta J_{k,k} + \Delta J_{k+1,k+1} + \mathcal{O}(\mathbf{k}^2)$$

is satisfied for small value of coupling $\mathbf{k}$.

Proof. Let the perturbed Jacobian $\mathbf{J}$ be given as a function of coupling strength $\mathbf{k}$. Assume all the matrices in Equation 6.13 are differentiable in a neighbourhood of k_i and the derivative with respect to k_i is denoted by a prime. We have

$$\mathbf{J}^{'}\mathbf{U} - \mathbf{U}\mathbf{\Lambda}^{'} = -\mathbf{J}\mathbf{U}^{'} + \mathbf{U}^{'}\mathbf{\Lambda} \tag{6.19}$$

As there is no repeated eigenvalue, we can assume that column eigenvector in the derivative matrix $\mathbf{U}'$ can be expressed as the linear combination of the eigenvectors in $\mathbf{U}$,

$$\mathbf{U}^{'} = \mathbf{U}C \tag{6.20}$$

Substitute (6.20) into (6.19) and both sides of equation premultiplying by the matrix $\mathbf{P}^{*} = \mathbf{U}^{-1}$ give

$$\mathbf{P}^{*}\mathbf{J}^{'}\mathbf{U} - \mathbf{\Lambda}^{'} = -\mathbf{\Lambda}C + C\mathbf{\Lambda} \tag{6.21}$$

The equation above has the following form if considered element by element,

$$\mathbf{p}_k^{*}\mathbf{J}^{'}\mathbf{u}_l - \delta_{kl}\lambda_l^{'} = (\lambda_l - \lambda_k)c_{kl} \tag{6.22}$$

where $\mathbf{p}_k^{*}$ and $\mathbf{u}_k$ are respectively the kth left- and right-eigenvector (here, $k = 2i - 1$). λ_k is the kth element on the diagonal of Λ, c_{kl} is the entry (k, l) of matrix C and $\delta_{kl} = 1$ if $k = l$ and 0 otherwise. So, we have

$$\lambda_k^{'} = \mathbf{p}_k^{*}\mathbf{J}^{'}\mathbf{u}_k, \quad k = 1, \ldots, 2\mathsf{MN}, \tag{6.23}$$

So, for the ith block of sub-matrix J_i in $\mathbf{J}$, we investigate the how the corresponding eigenvalues λ_k and λ_{k+1} (here, $k = 2i - 1$) change when the coupling is included,

$$\Delta\lambda_k = \mathbf{p}_k^{*}\Delta\mathbf{J}\mathbf{u}_k + \mathcal{O}(\mathbf{k}^2) \tag{6.24}$$

We have already illustrated the structure of $\Delta\mathbf{J}$ in Section 4.1.1,

$$\Delta\mathbf{J} = \mathbf{G} = \mathbf{G}_c + \mathbf{G}_d \tag{6.25}$$

where the expressions of $\mathbf{G}_c$ and $\mathbf{G}_d$ are respectively provided in Equation 4.4 and Equation 4.5. However, we are not using isotropic constant k now, so based on Equation 4.4 and Equation 4.5, the expression of $\Delta\mathbf{J}$ can be rewritten as following,

$$\Delta\mathbf{J} = K(I^{(M)} \otimes A^{N} + A^{(M)} \otimes I^{N}) \tag{6.26}$$

where $\mathrm{diag}(K) = (k_{v1}, k_{w1}, k_{v2}, k_{w2}, \ldots, k_{v\mathsf{MN}}, k_{w\mathsf{MN}})$.

Assume that $P_i^* = U_i^{-1}$, according to Equation 6.16, $\mathbf{P}$ can be expressed as

$$\mathbf{P} = \bigoplus_i P_i \tag{6.27}$$

Hence, in the column vectors $\mathbf{p}_k$ and $\mathbf{p}_{k+1}$, only the elements $p_{k,k}$, $p_{k+1,k}$, $p_{k,k+1}$ and $p_{k+1,k+1}$ are non-zero. And it is also the same situation for $\mathbf{u}_k$ and $\mathbf{u}_{k+1}$. Thereby,

$$\Delta\lambda_l = \sum_{r,s\in\{k,k+1\}} (p_{r,l}\Delta J_{r,s}u_{s,l}) + \mathcal{O}(\mathbf{k}^2), \quad l = k, k+1 = 2i-1, 2i \tag{6.28}$$

Considering the structure of coupling that $\Delta J_{k+1,k}$ and $\Delta J_{k,k+1}$ are both zeros,

$$\begin{aligned}\Delta\lambda_k + \Delta\lambda_{k+1} =&(p_{k,k}u_{k,k} + p_{k,k+1}u_{k,k+1})\Delta J_{k,k}+\\ &(p_{k+1,k}u_{k+1,k} + p_{k+1,k+1}u_{k+1,k+1})\Delta J_{k+1,k+1} + \mathcal{O}(\mathbf{k}^2)\end{aligned} \tag{6.29}$$

Due to $\mathbf{UP}^* = I$, namely,

$$\begin{aligned}p_{k,k}u_{k,k} + p_{k,k+1}u_{k,k+1} = 1\\ p_{k+1,k}u_{k+1,k} + p_{k+1,k+1}u_{k+1,k+1} = 1\end{aligned} \tag{6.30}$$

Hence,

$$\Delta\lambda_k + \Delta\lambda_{k+1} = \Delta J_{k,k} + \Delta J_{k+1,k+1} + \mathcal{O}(\mathbf{k}^2) \tag{6.31}$$

□

So, it is clear to see that the relation described in Equation 6.17 still holds for small couplings. Thereby, if one of the equilibria of the coupled network system becomes non-hyperbolic by undergoing a Hopf bifurcation, then correspondingly, the block of sub-matrix in $\mathbf{J}$ is traceless as expressed below,

$$J_{k,k} + \Delta J_{k,k} + J_{k+1,k+1} + \Delta J_{k+1,k+1} = 0 \tag{6.32}$$

Recalling that Jacobian of a FitzHugh-Nagumo model is expressed as,

$$J_i = \begin{pmatrix} -\epsilon^{-1}h_i & -\epsilon^{-1} \\ 1 & -b_i \end{pmatrix},$$

where

$$h_i = h(v_i, a_i) = 3v_i^2 - 2(a_i+1)v_i + a_i.$$

And since we set $k_v = 0$ in the previous chapter, we can define $k_i = k_{wi}$ and consequently,

$$\Delta J_{k,k} = 0, \quad \Delta J_{k+1,k+1} = -K_i$$

where K_i equals to $2k_i$, $3k_i$ and $4k_i$, respectively when the ith neuron is located at the corner, the side or in the center area of an image. So, the relation described in the equation below

$$-\epsilon^{-1}h_{v_i,a_i} - b_i - K_i = 0 \tag{6.33}$$

is used to determine the Hopf bifurcation condition for the coupled network.

Since Izhikevich (2006) presented an example using normal form to estimate the state variable after the saddle-node bifurcation, we attempt to estimate the relation between v_i and k_i through the same way assuming all the neurons are exactly at the saddle-node bifurcation points before the coupling is included. We start with the single FitzHugh-Nagumo model by treating b as the bifurcation parameter. However, the accuracy of resultant estimate is not sufficient for the application. Some relevant results are presented in Appendix A.

6.3 Parameter Settings for Uncoupled Neurons

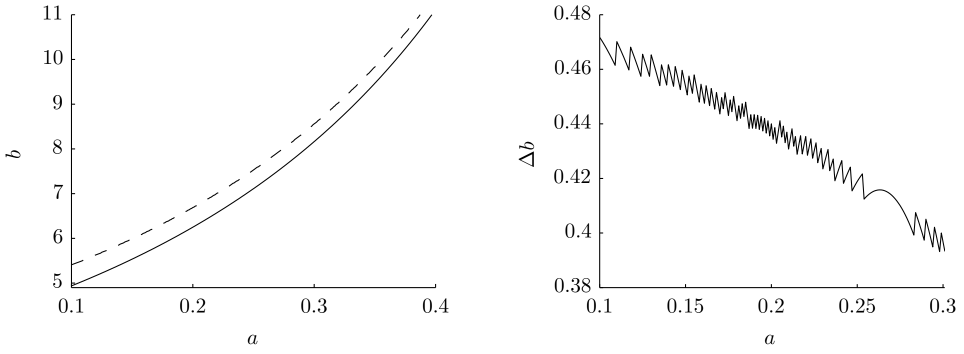
The estimation in Equation 6.33 provides the equivalent condition for Hopf bifurcation in coupled network instead of evaluating the detailed Jacobian $\mathbf{J}$ and eigenvalues $\mathbf{\Lambda}$. Now, it remains to determine the value of parameter b_i. The coupling strength k_i is closely related to the distance of an uncoupled neuron to its bifurcation point. One can image that if b_i is set relatively small, then the uncoupled neurons are relatively farther from its bifurcation point. Namely, the network will be less susceptible to be nonhyperbolic. In this case, we require that the coupling strength k_i be increased. However, the condition in 6.33 will no longer be effective if k_i becomes too large. So, in this thesis, we claim that the uncoupled network is sufficiently close to its bifurcation, if the two conditions that the coupling strength k_i is weak and the new fixed points of is an attractor are both satisfied.

According to the bifurcation pattern provided in Section 4.2.2, a nonhyperbolic equilibrium undergoes a saddle node bifurcation prior to a Hopf bifurcation. Figure 6.3(a) illustrates the bifurcation structures of a FitzHugh-Nagumo model We use b_{SN} and b_H to denote respectively the Saddle-Node and the Hopf bifurcation point. Recall that the expression of the saddle-node bifurcation curve for the single FitzHugh-Nagumo model is

$$b_{SN} = \frac{4}{(a-1)^2} \tag{6.34}$$

It can be clearly seen that curve of b_H is always higher than b_{SN} and difference between these two curves can be approximated by a linear fitting expressed as below,

$$\Delta b = -0.3a + 0.495 \tag{6.35}$$



(a)

(b)

Figure 6.3: Bifurcation structure of a single FitzHugh-Nagumo model. In (a), the solid curve denotes the Saddle-Node bifurcation point for a single neuron and the dashed curve denotes the Hopf bifurcation point. For fixed value of a, the value of b for Hopf bifurcation is always larger that for Saddle-Node bifurcation. And (b) presents the curve of the difference between the b values for these two different bifurcation points.

Hence,

$$b_H = b_{SN} - 0.3a + 0.495 \tag{6.36}$$

Therefore, it is natural to consider the condition of saddle-node bifurcation as a good starting point for $k_i = 0$.

We will provide a simulation on four coupled neurons in a chain in order to check that such a setting makes uncoupled neurons sufficiently close to Hopf bifurcation. Consider a network consisting of four neurons, where $\mathbf{X} \in \mathbb{R}^8$. Define five potential attractors $\overline{\mathbf{X}}_i$,where $i = 1, 2, ..., 5$ as below,

$$\begin{aligned}
\mathbf{v}|_{\overline{\mathbf{X}}_1} &\approx [1, 0, 0, 0]^T \\
\mathbf{v}|_{\overline{\mathbf{X}}_2} &\approx [0, 1, 0, 0]^T \\
\mathbf{v}|_{\overline{\mathbf{X}}_3} &\approx [0, 0, 1, 0]^T \\
\mathbf{v}|_{\overline{\mathbf{X}}_4} &\approx [1, 0, 1, 0]^T \\
\mathbf{v}|_{\overline{\mathbf{X}}_5} &\approx [0, 1, 1, 0]^T
\end{aligned}$$

and define

$$\mathbf{k} = [k_1, k_2, k_3, k_4]^T$$

It can be clearly seen that, in fact, $\mathbf{v}|_{\overline{\mathbf{X}}_4}$ is the combination of $\mathbf{v}|_{\overline{\mathbf{X}}_1}$ and $\mathbf{v}|_{\overline{\mathbf{X}}_3}$, while $\mathbf{v}|_{\overline{\mathbf{X}}_5}$ is the combination of $\mathbf{v}|_{\overline{\mathbf{X}}_2}$ and $\mathbf{v}|_{\overline{\mathbf{X}}_3}$.

For all the four neurons, $a_i = 0.1$ and b_i is set as b_{SN} when coupling is not included. So, $b_i = 4.9383$. We carefully tuned the coupling strength and find out all these three equilibria $\overline{\mathbf{X}}_1$, $\overline{\mathbf{X}}_2$ and $\overline{\mathbf{X}}_3$ have a pair of purely imaginary eigenvalues when

$$\mathbf{k} = [0.47157,\ 0.23815,\ 0.23455,\ 0.1]^T.$$

Namely they are all undergoing Hopf bifurcations. The values of the equilibria as roots to the equation $\mathbf{F}(\mathbf{X}) = 0$ are presented in Table 6.1. At this time, the eigenvalues of $\overline{\mathbf{X}}_4$ and $\overline{\mathbf{X}}_5$,

$$\lambda|_{\overline{\mathbf{X}}_4} = \begin{bmatrix} -173.05 \\ 1.19 + 30.92i \\ 1.19 - 30.92i \\ 0.59 + 31.05i \\ 0.59 - 31.05i \\ -11.38 \\ -14.71 \\ -108.45 \end{bmatrix}, \quad \lambda|_{\overline{\mathbf{X}}_5} = \begin{bmatrix} -171.46 \\ 54.66 \\ 52.89 \\ -11.44 \\ 11.08 \\ 11.91 \\ -108.24 \\ -14.73 \end{bmatrix}$$

If we slightly increase the coupling strength to

$$\mathbf{k} = [0.48150,\ 0.23828,\ 0.23699,\ 0.1]^T \tag{6.37}$$

then $\overline{\mathbf{X}}_4$ is right at the bifurcation point and it has two pair of purely imaginary eigenvalues. At this time, both $\overline{\mathbf{X}}_1$ and $\overline{\mathbf{X}}_3$ cross the boundary of stability while $\overline{\mathbf{X}}_2$ is still undergoing Hopf bifurcation. And the eigenvalues of $\overline{\mathbf{X}}_5$ are

$$\lambda|_{\overline{\mathbf{X}}_5} = \begin{bmatrix} -172.97 \\ 54.76 \\ 51.72 \\ -11.39 \\ 11.10 \\ 12.21 \\ -108.24 \\ -14.73 \end{bmatrix}$$

Again the values of the equilibria are presented in Table 6.2. Finally, in order to make $\overline{\mathbf{X}}_5$ go beyond the Hopf bifurcation point, we increase the coupling strength to

$$\mathbf{k} = [0.54,\ 0.48259,\ 0.47759,\ 0.1]^T$$

Now, the eigenvalues of $\overline{\mathbf{X}}_5$ are

$$\lambda|_{\overline{\mathbf{X}}_5} = \begin{bmatrix} -182.69 \\ 0.00 + 0.3109i \\ 0.00 - 0.3109i \\ -1.61 + 0.3131i \\ -1.61 - 0.3131i \\ -11.13 \\ -108.45 \\ -14.71 \end{bmatrix}$$

$\overline{\mathbf{X}}$	v_1	v_2	v_3	v_4
$\overline{\mathbf{X}}_1$	0.6818	-0.0182	-0.0003	0.0000
$\overline{\mathbf{X}}_2$	-0.0340	0.6818	-0.0180	-0.0001
$\overline{\mathbf{X}}_3$	-0.0007	-0.0182	0.6818	-0.0081
$\overline{\mathbf{X}}_4$	0.6805	-0.0343	0.6812	-0.0081
$\overline{\mathbf{X}}_5$	-0.0337	0.6448	0.6455	-0.0081

Table 6.1: $k_1 = 0.47157$, $k_2 = 0.23815$, $k_3 = 0.23455$, $k_4 = 0.1$

$\overline{\mathbf{X}}$	v_1	v_2	v_3	v_4
$\overline{\mathbf{X}}_1$	0.6831	-0.0182	-0.0003	0.0000
$\overline{\mathbf{X}}_2$	-0.0346	0.6818	-0.0181	-0.0001
$\overline{\mathbf{X}}_3$	-0.0007	-0.0182	0.6824	-0.0081
$\overline{\mathbf{X}}_4$	0.6818	-0.0343	0.6818	-0.0081
$\overline{\mathbf{X}}_5$	-0.0343	0.6447	0.6460	-0.0081

Table 6.2: $k_1 = 0.48150$, $k_2 = 0.23828$, $k_3 = 0.23699$, $k_4 = 0.1$

$\overline{\mathbf{X}}$	v_1	v_2	v_3	v_4
$\overline{\mathbf{X}}_1$	0.6889	-0.0336	-0.0014	0.0000
$\overline{\mathbf{X}}_2$	-0.0376	0.7289	-0.0328	-0.0003
$\overline{\mathbf{X}}_3$	-0.0016	-0.0331	0.7295	-0.0080
$\overline{\mathbf{X}}_4$	0.6860	-0.0604	0.7277	-0.0080
$\overline{\mathbf{X}}_5$	-0.0382	0.6816	0.6832	-0.0081

Table 6.3: $k_1 = 0.54$, $k_2 = 0.48259$, $k_3 = 0.47759$, $k_4 = 0.1$

The simulation above provides a typical example illustrating the relation among the stabilities of the attractors in a MA type network. From an application point of view, although some potential attractors, such as $\overline{\mathbf{X}}_4$ and $\overline{\mathbf{X}}_5$, can be considered as the combination of others, such as $\overline{\mathbf{X}}_1$, $\overline{\mathbf{X}}_2$ and $\overline{\mathbf{X}}_3$, the condition for non-hyperbolicity, or more specifically stability, of the latter is generally not applicable to the former. Because of the nearest coupling, we can expect that the influence is only locally strong and becomes less obvious when distances among neurons increase. So, we can see that the bifurcation condition of $\overline{\mathbf{X}}_1$ and $\overline{\mathbf{X}}_3$, in terms of $\mathbf{k}$ is almost equal to that of $\overline{\mathbf{X}}_4$. It can be explained that the edge pulses in the shape of $\overline{\mathbf{X}}_4$ is not quite near to each other. Whereas in $\overline{\mathbf{X}}_5$, the two pulse are adjacent so that we require a much stronger coupling to reach the bifurcation point.

However, we expect that bifurcation condition of attractor consisting of neighbouring pulses become close to the combinatory conditions of each single one, when the uncoupled neurons becomes further close to the bifurcation point. Considering the extreme case where all the uncoupled neurons are right at their own Hopf bifurcation points, any infinitesimal coupling will lead both single pulse attractor and the corresponding combinatory pulse attractor to become stable.

We will run another group of simulations to verify this idea. Assume that number of the neurons in the network does not change and $a_i = 0.1$ for all the neurons. Now, we increase the value of parameter b_i. By using μ to denote the constant in Equation 6.36, we obtain,

$$b_i = \frac{4}{(1-a_i)^2} - 0.3a_i + \mu \tag{6.38}$$

We attempt to choose $\mu = 0.25$. The values of all the selected equilibria obtained during the simulation for different settings of $\mathbf{k}$ are listed in Table 6.4 and 6.5. Again we start with the case where $\overline{\mathbf{X}}_1$, $\overline{\mathbf{X}}_2$ and $\overline{\mathbf{X}}_3$ are undergoing Hopf bifurcations. At this time,

$$\mathbf{k} = [0.24548,\ 0.12334,\ 0.12264,\ 0.1]^T$$

In order for $\overline{\mathbf{X}}_5$ becoming stable, we strengthen the couplings to values as below.

$$\mathbf{k} = [0.24548,\ 0.24754,\ 0.24534,\ 0.1]^T$$

It is clearly to see that these sets of coupling strength become much closer.

6.4 Anisotropic Coupling Strength

The section aims to propose an edge detection based on the analysis provided above. The basic idea is to first set the parameter b_i in Equation 6.38 with the parameter μ to control the distance of the uncoupled neurons to the Hopf bifurcation point. Then, set the coupling strength k_{wi} according to the normalised gradient magnitude,

$$k_i = \nu + \frac{||\nabla {U^r}_i||}{\max\limits_i(||\nabla {U^r}_i||)} \tag{6.39}$$

ν is a parameter to set the baseline which controls the coupling strength. In contrast to the setting of gradient magnitude in Algorithm 2, the magnitude adopted here to assign

$\overline{\mathbf{X}}$	v_1	v_2	v_3	v_4
$\overline{\mathbf{X}}_1$	0.6818	-0.0097	-0.0001	0.0000
$\overline{\mathbf{X}}_2$	-0.0187	0.6818	-0.0097	-0.0001
$\overline{\mathbf{X}}_3$	-0.0002	-0.0097	0.6818	-0.0080
$\overline{\mathbf{X}}_4$	0.6815	-0.0188	0.6817	-0.0080
$\overline{\mathbf{X}}_5$	-0.0187	0.6644	0.6645	-0.0080

Table 6.4: $k_1 = 0.24548$, $k_2 = 0.12334$, $k_3 = 0.12264$, $k_4 = 0.1$

$\overline{\mathbf{X}}$	v_1	v_2	v_3	v_4
$\overline{\mathbf{X}}_1$	0.6815	-0.0186	-0.0004	0.0000
$\overline{\mathbf{X}}_2$	-0.0187	0.7092	-0.0183	-0.0001
$\overline{\mathbf{X}}_3$	-0.0004	-0.0185	0.7090	-0.0080
$\overline{\mathbf{X}}_4$	0.6808	-0.0349	0.7085	-0.0080
$\overline{\mathbf{X}}_5$	-0.0187	0.6818	0.6818	-0.0080

Table 6.5: $k_1 = 0.24548$, $k_2 = 0.24754$, $k_3 = 0.24534$, $k_4 = 0.1$

k_i need not to be thresholded since the Hopf bifurcation points work as an intrinsic boundary for the stability. One could image that if ν is relatively small, there will be many neurons silent which however are supposed to be excited. In other words, some of the edge points in the ground truth data will be missing. And if ν is too large, more error edges will turn up in the final results. By doing so, we largely make use of the property of the model according to the theory of WCNN and largely simplify the off-chip design procedures. The algorithm can be concluded as the following Algorithm 3,

Algorithm 3 Anisotropic Coupling Strengths

```
1: Rescale the image intensity distribution ${U^r}_{\mathsf{m,n}} = U_{(\mathsf{m,n})}/1275 + 0.1$ so that $0.1 \leqslant {U^r}_{(\mathsf{m,n})} \leqslant 0.3$.
2: Solve the equation below get a diffused version of rescaled image $\theta_i(\tau_s)$
       $\dot{\theta}_i = \xi \sum_{j \in \mathcal{P}_i} (\theta_j - \theta_i) - (\theta_i - {U^r}_i)$
3: Evaluate the system parameter $a_i = 1.02\theta_i - 0.01$ and set the system parameter $b_i$ as
       $b_i = \frac{4}{(1 - a_i)^2} - 0.3a_i + \mu$
4: Set the coupling strength as
       $k_i = \nu + \frac{||\nabla {U^r}_i||}{\max\limits_i(||\nabla {U^r}_i||)}$
5: Solve the model equation below,
       $\begin{cases} \dot{v_i} = f(v_i, w_i, a_i) \\ \dot{w_i} = g(v_i, w_i) + k_i \sum_{j \in \mathcal{P}_i} (w_j - w_i) \end{cases}$
   with the initial condition $(v_i(0), w_i(0)) = ({U^r}_i, 0)$ and the zero boundary conditions to get the steady state solution in $v$, which is $v_i(\tau_s)$.
6: Threshold $v_{(\mathsf{m,n})}(\tau_s)$ to get the final binary edge map $\mathcal{M}_{(\mathsf{m,n})}$.
       $\mathcal{M}(\mathsf{m,n}) = \begin{cases} 0, & v_i(\tau_s) > 0.5 \\ 1, & v_i(\tau_s) \leq 0.5 \end{cases}$
```

6.5 Results

This section applies the algorithm proposed in the previous section to detect the edges in both the artificial and real images, in comparison to the results obtained by Algorithm 2. Specifically for testing the noise immunity of the algorithm, we present a robustness test in comparison to Algorithm 2.

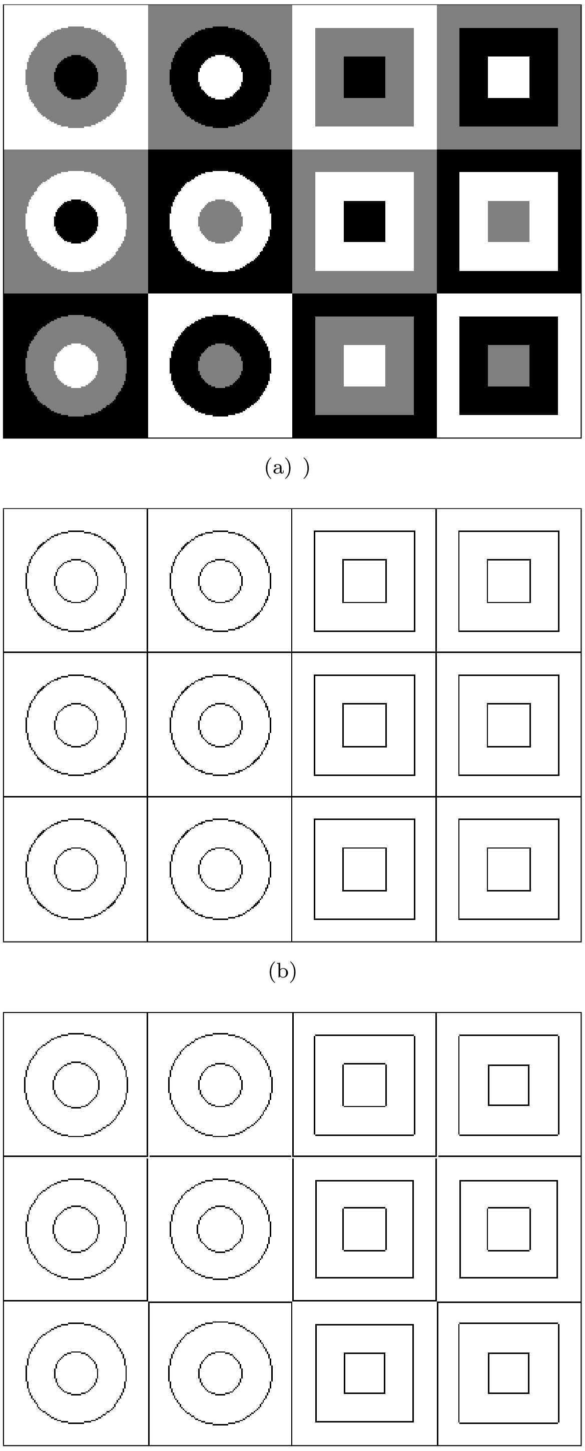
Artificial Images

The basic structure of the artificial images used to test the method in Algorithm 3 is presented in Figure 6.4(a) , which is the same as that used in Section 5.3. The darker and lighter versions of the artificial image are also applied in order to test the usefulness of the algorithm. Hence, the ground truth data also keeps the same as shown Figure 6.4(b). We use this set of three artificial images to respectively test Algorithm 2 and Algorithm 3. Moreover, in order to see the influence of self-stopping diffusion on the algorithm performance, we also test Algorithm 3 where we only replace the self-stopping diffusion with the original diffusion which has a stopping time $\tau = 1$. The results obtained by using these three methods from all the artificial images are the same as shown in Figure 6.4(c), where $tp = 5758$, $tp_r = 98.37\%$, $fp = 0$ and $fp_r = 0\%$.

Noise Robustness Test

For the noise robustness test of method in Algorithm 3, the three intensity levels of the artificial image are chosen to be 63, 127 and 191 and the white noise is added with an increasing standard deviation σ from 10 to 60 with step 10. Figure 6.6(a) presents an illustrative example of the noisy image with $\sigma = 30$. We compare the results to the previous method in Algorithm 2, which introduced in Section 5.2.3. As mentioned above, ν in Equation 6.39 is set to denoise, achieving the same function as that of the parameter η in Algorithm 2. Specifically, a smaller value of ν stands for a averagely weak coupling strength and thereby a further distance from Hopf bifurcation point. Hence, a lower rate of false positives would be expected in this case. In the simulation, ν is set in the range -0.3 to -0.05 with a step of 0.05, for each value of σ, while η is set by sweeping from 0.05 to 0.3 with a step of 0.05 for each value of σ.

Figure 6.5 presents the curves of true positive rate and false positive rate respectively obtained by Algorithm 2 and Algorithm 3. Specifically, the solid lines in Figure 6.5(a) and Figure 6.5(c) illustrate the resultant curves obtained by Algorithm 3 while the dashed lines in Figure 6.5(b) and Figure 6.5(d) illustrate those by Algorithm 2. It can clearly seen in Figure 6.5(c) that the curves of false positive rate decays upon a decreasing ν as expected. The fp_r curves of the proposed method is generally lower than that of the previous method. But the previous method can always lead to higher tp_r curves.

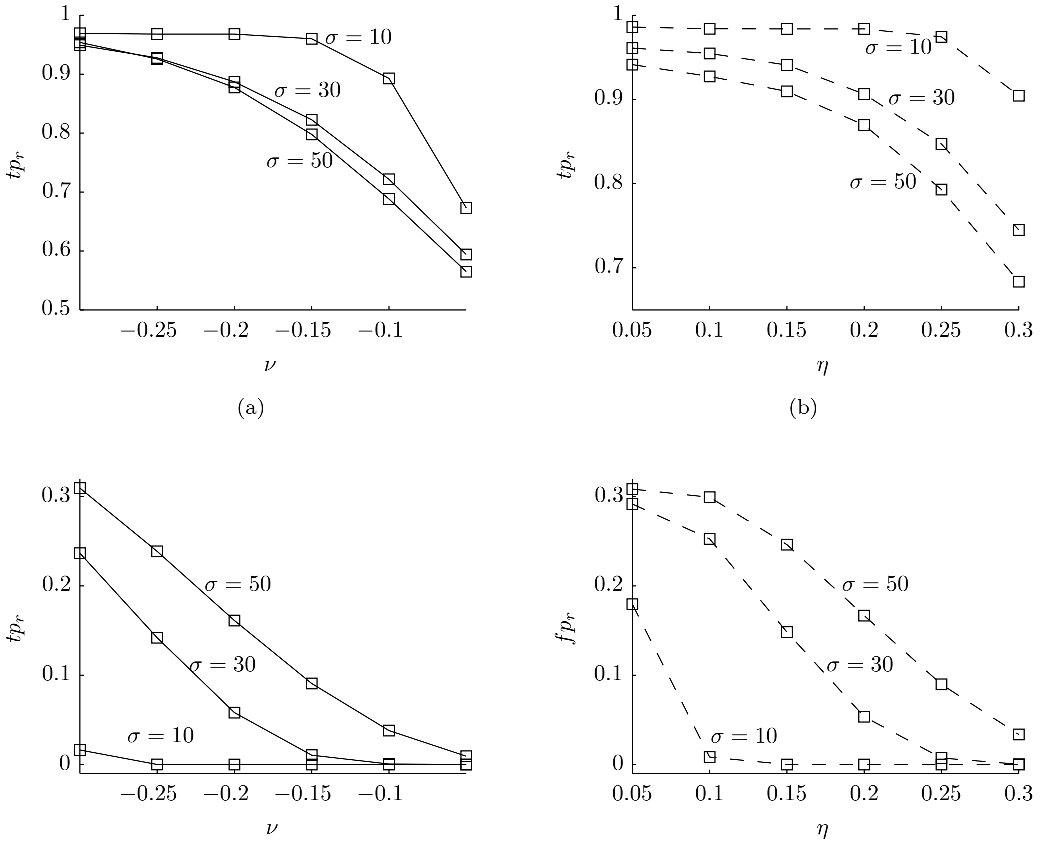


(a))

(b)

(c)

Figure 6.4: Edge map obtained by Algorithm 2 and Algorithm 3 using both self-stopping diffusion and diffusion with a stopping time. In Algorithm 2, the parameter settings are $b = 3.5$, $\epsilon = 0.001$, $\eta = 0$, $\widetilde{d} = 10$ and $\tau = 1$. For Algorithm 3, $\mu = 0.25$ and $\nu = 0$. For the self-stop diffusion, $\xi = 3$, otherwise for the original diffusion, a stopping time τ is chosen to be 1. The results are all the same as shown in (c), where $tp = 5758$, $tp_r = 98.37\%$, $fp = 0$ and $fp_r = 0\%$.

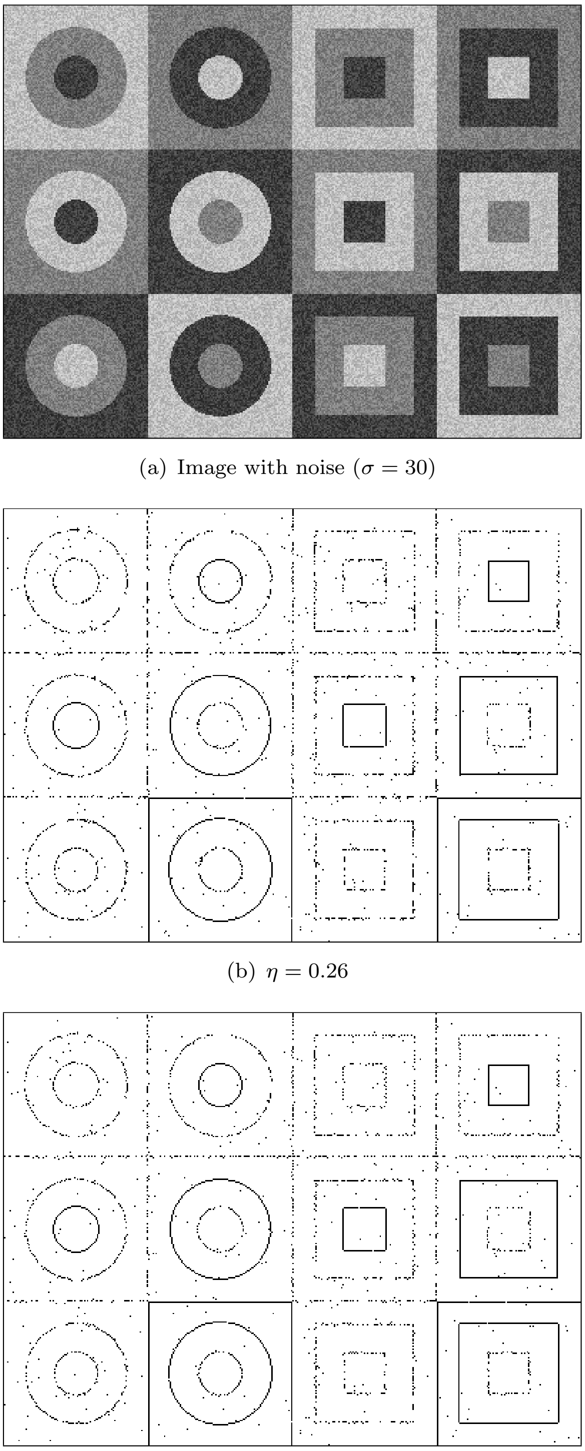


(a) (b)

(c) (d)

Figure 6.5: Robustness test for the proposed method in comparison with the previous algorithm in Section 5.2.3. Performance measures on the previous method is illustrated by the dashed lines while measures on the proposed method is illustrated by the solid lines. Generally, the results obtained by the proposed method have less false positives. However, the results obtained by the previous method have more true positive points.

Figure 6.6(b) and Figure 6.6(c) provide two illustrative edge maps which are obtained from noisy image with $\sigma = 30$ respectively by Algorithm 2 and Algorithm 3.



(a) Image with noise ($\sigma = 30$)

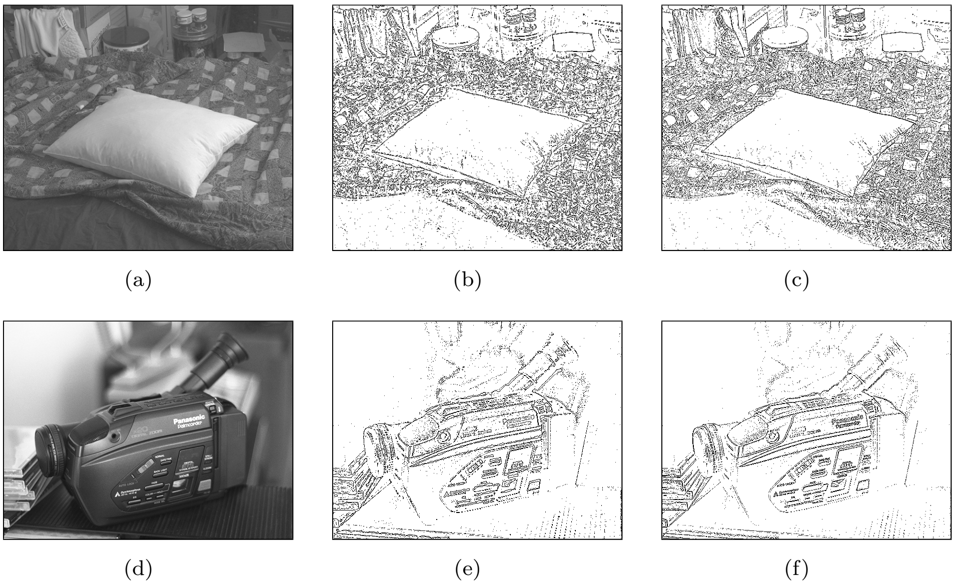
(b) $\eta = 0.26$

(c) $\nu = -0.22$

Figure 6.6: The true positive rates of these two edge maps are nearly the same, $tp_r = 0.8112$ for 6.6(b) and $tp_r = 0.8110$ for 6.6(b). The false positive rate of the later ($fp_r = 0.0025$) is lower than that of the former ($fp_r = 0.0030$).

Real Images

Figure 6.7 and Figure 6.8 provide the edge detection results from the real grayscale images obtained by Algorithm 2 and Algorithm 3. For each real image, the baseline ν in Equation 6.39 can be optimised for each value of the distance μ for uncoupled neurons from the Hopf bifurcation point. Here, we choose $\mu = 0.25$ and $\nu = -0.05$. The results obtained by both algorithms are similar. Hence, we can draw conclusion that the parameter settings in Equation 6.38 and Equation 6.39 which is designed according to the theory of WCNN is appropriate to the model in Equation 4.12. And the diffused image obtained by Equation 6.8 works correctly as that obtained from the original diffusion process which requires a stopping time.


(a) (b) (c)

(d) (e) (f)

Figure 6.7: Edge detection results for real images, (a) Pillow [468 × 552]. (d) Video camera [435×577]. (b) and (e) provide the edge detection results obtained by Algorithm 2, in which the system parameters are $b = 3.5$, $\epsilon = 0.001$, $k_v = 0$ and $k_w = 5$. The constant $\widetilde{d} = 10$ and the threshold $\eta = 0.05$ in Equation 5.13 The stopping time of the anisotropic diffusion is $\tau = 1.0$. The steady time $\tau_s = 1.0$ for all the models. (c) and (f) provide the edge detection results obtained by Algorithm 3, in which the parameter b is set according to Equation 6.38 where $\mu = 0.25$ and the coupling strength k_i is set according to Equation 6.39 where $\nu = -0.05$. The diffusive coefficient $\xi = 3$ in Equation 6.8. The steady state time $\tau_s = 1.0$ for the model.

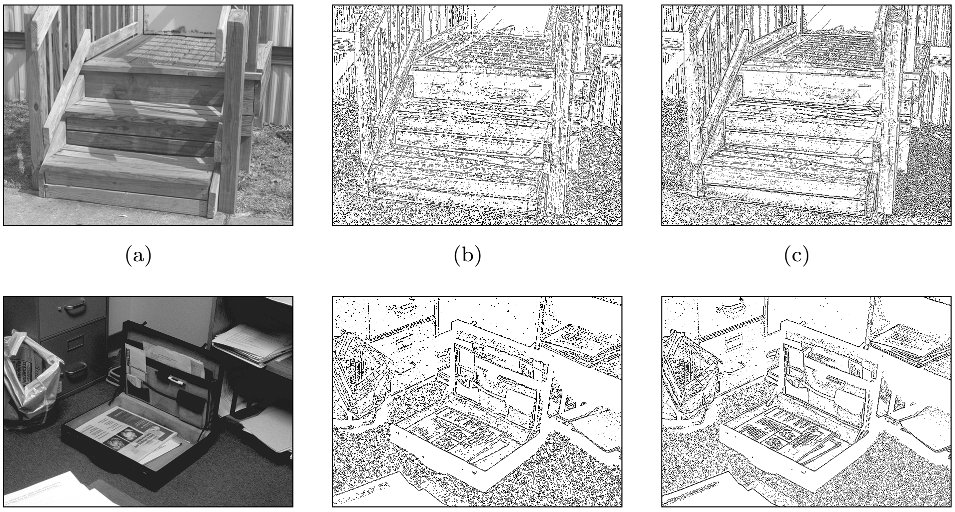


Figure 6.8: Edge detection results for real images, (a) Stairs [441 × 579]. (d) Briefcase [419×577]. (b) and (e) provide the edge detection results obtained by Algorithm 2, in which the system parameters are $b = 3.5$, $\epsilon = 0.001$, $k_v = 0$ and $k_w = 5$. The constant $\widetilde{d} = 10$ and the threshold $\eta = 0.05$ in Equation 5.13 The stopping time of the anisotropic diffusion is $\tau = 1.0$. The steady time $\tau_s = 1.0$ for all the models. (c) and (f) provide the edge detection results obtained by Algorithm 3, in which the parameter b is set according to Equation 6.38 where $\mu = 0.25$ and the coupling strength k_i is set according to Equation 6.39 where $\nu = -0.05$. The diffusive coefficient $\xi = 3$ in Equation 6.8. The steady state time $\tau_s = 1.0$ for the model.

Chapter 7

Conclusion

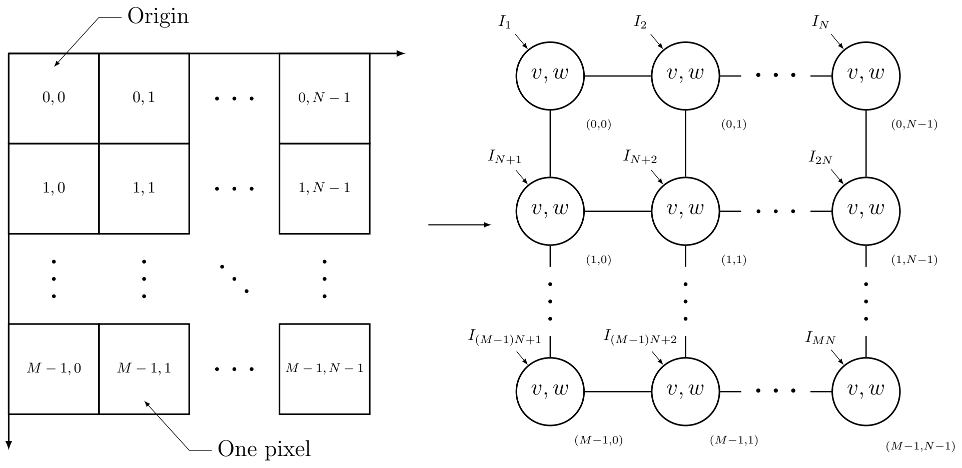


Figure 7.1: One pixel to one neuron structure of the network model

In this thesis, we have set out to refine in performance and reduce in size, a reaction diffusion system that models a physical process for edge detection (Nomura et al., 2011b) that can be fabricated in silicon. As such, the aspects of computational complexity of edge detection algorithms implemented in digital circuits are not relevant to the kinds of algorithms that we have been studying in this thesis. In particular, we have chosen to focus on edge detection to illustrate the usefulness of fixed point states of dynamical systems in addressing a computational need, and the prior work presented by Nomura et al. (2011a,b) provided a convenient reference for comparative purposes. In addition, while there are many neural network algorithms that have been proposed for tackling the task of edge detection, such as (Aizenberg et al., 1998; Suzuki et al., 2003, 2004), but they typically use them as abstract data structures that enable sophisticated, biologically inspired algorithms to be run on digital processors. We would like to emphasise this difference from many existing algorithms, even though we do not completely eschew digital software processing.

In fact, we share the computational burden between software components that rely on digital processing, which we refer to off-chip design, and hardware analog computation. The very core of the original edge detection algorithm and the proposed ones is the dynamical system described in Equation 3.17, of which the structure is clearly shown in Figure 7.1. Building such a network using the technology of integrated circuit belongs to the hardware design procedures. The design of the single element of this system in Equation 3.3 is covered by the huge topic "silicon neurons" (Mahowald and Douglas, 1991; Mizoguchi et al., 2011) and the coupling structure is also thoroughly studied under the other topic called "Cellular Neural Network" (Chua and Yang, 1988; Karahaliloğlu and Balkır, 2004). We provided an introductory review of silicon neurons and networks in Appendix B. We provide a simplified version in Equation 6.9 by eliminating the couplings of the membrane potentials from the adjacent neurons and we propose an modification on dynamics so that the diffusion process will automatically stop. By taking the rescaled image intensities as the input, such the system hardware can generate fast self-evolving edge results of binary images and grayscale images respectively with an constant image threshold (Nomura et al., 2003) or a variable one (Nomura et al., 2008, 2011b,a) embedded as the system parameters.

For evaluating the image threshold θ, the previous work (Nomura et al., 2011b,a) in Algorithm 1 does not require many computations after hardware design as they adopt the equation $a = \theta$ (which we find to be inadequate) and subject a to isotropic diffusion. The diffusion equation in Equation 3.23 needs to be solved, and the rescaling of the intensities of the original and inverted images need to be evaluated - these are the computational tasks for Algorithm 1. In addition to the rescaling of the original image, our method in Algorithm 2 requires another four steps in order to obtain the θ from the original image U^r. In particular, we need to set the constant threshold η within the correct range, evaluate the magnitude of the gradient $\|\nabla U^r\|$, set the diffusion coefficient d and finally, solve the anisotropic diffusion in Equation 5.13. All these computations belong to the off-circuit design, which will be completed by the software program before being applied to the circuit system. The major difference of our method in Algorithm 3 from Algorithm 2 is to redesign the diffusion of rescaled image on hardware and the diffusion process will automatically stop. By slightly increasing the chip size, we appreciably reduce the off-chip procedure as no diffusion equation is required to be solved. A more extensive comparison of these three algorithms is provided in next section.

7.1 Comparison on Models and Methods

The three methods of edge detection are mainly discussed and compared in this thesis, which are respectively the original method by Nomura et al. (2008) in Algorithm 1, our method using anisotropically diffused image in Algorithm 2 and the proposed one using anisotropic coupling strength in Algorithm 3.

Recall that the model of the system originally used by Nomura et al. (2011a,b) can be expressed as follows,

$$\begin{cases} \dot{v_i}^0 = f(v_i^0, w_i^0, a_i^0) + k_v \sum\limits_{j \in \mathcal{P}_i} (v_j^0 - v_i^0) + \dot{v_i}^1 \mathcal{H}(-\dot{v_i}^1) \\ \dot{w_i}^0 = g(v_i^0, w_i^0) + k_w \sum\limits_{j \in \mathcal{P}_i} (w_j^0 - w_i^0) \\ \dot{v_i}^1 = f(v_i^1, w_i^1, a_i^1) + k_v \sum\limits_{j \in \mathcal{P}_i} (v_j^1 - v_i^1) \\ \dot{w_i}^1 = g(v_i^1, w_i^1) + k_w \sum\limits_{j \in \mathcal{P}_i} (w_j^1 - w_i^1). \end{cases} \tag{7.1}$$

This system is generally of double size of the one in Equation 7.2. In contrast, both our previous method and proposed one still use the system. In fact, the expressions of the network models used in the latter two algorithms can unified as the following equations,

$$\begin{cases} \dot{v_i} = f(v_i, w_i, a_i) \\ \dot{w_i} = g(v_i, w_i) + k_i \sum\limits_{j \in \mathcal{P}_i} (w_j - w_i) \end{cases} \tag{7.2}$$

It is also worth noting that the coupling through the variable v ($k_v = 0$) is eliminated in the model for both of our methods.

All the three methods require a step to diffuse the original image in order to set the system parameter a which is considered as a threshold in the dynamics of single neuron. In Nomura's method, the diffused image $\theta_i(\tau)$ is obtained through a diffusion equation by taking the image intensities as initial condition. And in our previous method, in order for better performance, we slightly changed the diffusion equation by using non-identical diffusive coefficient d_i so that $\theta_i(\tau)$ becomes an anisotropic diffusive version of the original image. However, a stopping time τ is needed in both of these two methods. By adding a term of state variable subtracting the initial condition in the diffusion equation, the proposed method does not require the setting of stopping time any more. We use a different syntax $\theta_i(\tau_s)$ to denote diffusive image in the proposed method where τ_s is the time when state variables reach the steady state.

Nomura et al. (2011a,b) claim that doubling the size of the system benefits the reduction of influence of the “wrong pulses” problem. This problem has been illustrated and discussed in Chapter 5 in details by examples. The system parameter a is simply treated as the diffused image $\theta_i(\tau)$ in Nomura's method. And we point out that it is one of the important reasons which leads to the ”wrong pulse” problem. By carefully setting up the relationship between a and $\theta_i(\tau)$ (or $\theta_i(\tau_s)$ in the proposed method), we solve the ”wrong pulse” problem without increasing the size of the system. Moreover, according to the range where the relationship becomes the most linear, we reset range of rescaled image

$U^r \in [0.1, 0.3]$. Nevertheless, why the original range $U^r \in [0, 0.25]$ was not mentioned in the work done by Nomura et al. (2003, 2008, 2011a,b).

Algorithm 3 claims how to set the parameter b_i and the coupling strength k_i in the system. However, both Algorithm 1 and 2 fails to do this. Based on the theory of WCNN, the b_i is required to set near to the Hopf bifurcation in an uncoupled neuron which stands for the boundary of stability.

There is almost no steps for denoising in the Nomura's method. And this drawback is also mentioned in Nomura's work (Nomura et al., 2011b) as well as in other work (Denisov et al., 2010). In order to denoise, both of our methods use the preprocess of the image for the normalized magnitude of the gradient $\overline{||\nabla U^r{}_i||}$ as well as a constant threshold (η in previous method and ν in proposed one). In the previous method, denoising is achieved by controlling the speed diffusion in $\theta(\tau)$; the diffusive $\widetilde{d}$ is binarised according to the heaviside function $\mathcal{H}(\overline{||\nabla U^r{}_i||} - \eta)$. Differently for Algorithm 3, the mechanism of denoising is directly embedded in the parameter setting of k_i and binarisation is no longer adopted because the boundary of Hopf bifurcation point in $\mathbf{k}$ makes a similar function of thresholding.

Table 7.1: Comparison on steps among Nomura's method, previous method and proposed algorithms

	Nomura's algorithm	Previous algorithm	Proposed algorithm
Rescaling	$U^r \in [0, 0.25]$	$U^r \in [0.1, 0.3]$	$U^r \in [0.1, 0.3]$
Diffusion of Image	Solve the equations below to obtain respectively two diffused images $\theta_i^0(\tau)$ and $\theta_i^1(\tau)$ with the stopping time τ, $\dot{\theta}_i^0 = d^0 \sum_{j \in \mathcal{P}_i} (\theta_j^0 - \theta_i^0)$ $\dot{\theta}_i^1 = d^1 \sum_{j \in \mathcal{P}_i} (\theta_j^1 - \theta_i^1)$ where, $d^1 > d^0$ and the initial conditions are $\theta_i^0(0) = \theta_i^1(0) = U_i^r$.	Solve the equations below to obtain an anisotropically diffused image $\theta_i(\tau)$ with the stopping time τ, $\dot{\theta}_i = d_i \sum_{j \in \mathcal{P}_i} (\theta_j - \theta_i)$ $d_i = \widetilde{d} \cdot \mathcal{H}(\overline{\vert\vert \nabla {U^r}_i \vert\vert} - \eta)$ where, $\vert\vert \nabla {U^r}_i \vert\vert = \vert\vert (\frac{\partial {U^r}_i}{\partial x}, \frac{\partial {U^r}_i}{\partial y}) \vert\vert$ $\overline{\vert\vert \nabla {U^r}_i \vert\vert} = \frac{\vert\vert \nabla {U^r}_i \vert\vert}{\max\limits_i (\vert\vert \nabla {U^r}_i \vert\vert)}$ and the initial condition is $\theta_i(0) = U_i^r$.	Solve the equation below for the blurred image $\theta_i(\tau_s)$ with steady state time τ_s, $\dot{\theta}_i = \xi \sum_{j \in \mathcal{P}_i} (\theta_j - \theta_i) - (\theta_i - \theta_i(0))$ where, the initial condition is $\theta_i(0) = U_i^r$.
Parameter a	$a = \theta(\tau)$	$a = c_1 \theta(\tau) + c_2$	$a = c_1 \theta(\tau_s) + c_2$
Parameter b	$b = 1$	$b = 3.5$	$b_i = 4/(1 - a_i) - 0.3 * a_i$
Coupling Strength	$k_v = 4, k_w = 20$	$k_v = 0, k_w = 5$	$k_v = 0,$ ${k_w}_i = k_i = \nu + \overline{\vert\vert \nabla {U^r}_i \vert\vert}$ where, $\vert\vert \nabla {U^r}_i \vert\vert = \vert\vert (\frac{\partial {U^r}_i}{\partial x}, \frac{\partial {U^r}_i}{\partial y}) \vert\vert$ $\overline{\vert\vert \nabla {U^r}_i \vert\vert} = \frac{\vert\vert \nabla {U^r}_i \vert\vert}{\max\limits_i (\vert\vert \nabla {U^r}_i \vert\vert)}$
Solving System Equation	Solve the system in Equation 7.1 with initial condition $(v_i^0(0), w_i^0(0), v_i^1(0), w_i^1(0)) = ({U^r}_i, 0, {U^r}_i, 0)$ and zero boundary condition for $v_i^0(\tau_s)$ with steady state time τ_s	Solve the system in Equation 7.2 with initial condition $(v_i(0), w_i(0)) = ({U^r}_i, 0)$ and zero boundary condition for $v_i(\tau_s)$ with steady state time τ_s	Solve the system in Equation 7.2 with initial condition $(v_i(0), w_i(0)) = ({U^r}_i, 0)$ and zero boundary condition for $v_i(\tau_s)$ with steady state time τ_s
Edge Map	Obtain the first putative edge map $\mathcal{M}_1$ via a simple thresholding $v^0_{(\mathsf{m,n})}(\tau_s)$ as below, $\mathcal{M}_{1\mathsf{m,n}} = \begin{cases} 1, v_i^0(\tau_s) > 0.5 \\ 0, v_i^0(\tau_s) \leq 0.5 \end{cases}$ Repeat all the steps above to get a second putative edge map $\mathcal{M}_2$ from the black-and-white inversion of the original image $\bar{U}_{(\mathsf{m,n})}$ Get the final edge map $\mathcal{M}$ by merging the two putative ones, $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2$	Obtain the edge map $\mathcal{M}$ via a simple thresholding $v_{(\mathsf{m,n})}(\tau_s)$ as below, $\mathcal{M}_{\mathsf{m,n}} = \begin{cases} 1, v_i(\tau_s) > 0.5 \\ 0, v_i(\tau_s) \leq 0.5 \end{cases}$	Obtain the edge map $\mathcal{M}$ via a simple thresholding $v_{(\mathsf{m,n})}(\tau_s)$ as below, $\mathcal{M}_{\mathsf{m,n}} = \begin{cases} 1, v_i(\tau_s) > 0.5 \\ 0, v_i(\tau_s) \leq 0.5 \end{cases}$

Appendix A

Center Manifold Reduction of Planar FitzHugh-Nagumo Model

In order to find out the local dynamics at the saddle node bifurcation where the Jacobian **J** is singular, we need to focus on the *center manifold* at the corresponding bifurcation point. Due to the whole network is uncoupled, we start with the dynamics of a single neuron in Equation 3.3. Assume that $b = \frac{4}{(a-1)^2}$. So the system is at the bifurcation point and the system has a second equilibrium ($v_{sn} = \frac{a+1}{2}, w_{sn} = \frac{a+1}{2b}$) other than the origin which is called a *saddle node*. In order for the convenience of analysis, we require to locate the equilibrium at the origin with the coordinate transformation below,

$$v = v^* + v_{sn}, \quad w = w^* + w_{sn}, \quad b = b^* + b_{sn} \tag{A.1}$$

Substitute the equations above in Equation 3.3 and it is obtained,

$$\begin{aligned}
\dot{v} &= -\epsilon^{-1}((v + v_{sn})^3 - (a+1)(v + v_{sn})^2 + a(v + v_{sn}) + (w + w_{sn})) \\
\dot{w} &= (v + v_{sn}) - (b + b_{sn})(w + w_{sn}) \\
\dot{b} &= 0
\end{aligned} \tag{A.2}$$

For the sake of convenience, we keep using v and w as the state variable instead of v^* and w^*. Rewrite the equation above in the form below,

$$\dot{\mathbf{x}} = L\mathbf{x} + P(\mathbf{x}) \tag{A.3}$$

where L is the binarisation which equal to the Jacobian J at the origin,

$$L = \begin{bmatrix} (\epsilon b_{sn})^{-1} & -\epsilon^{-1} \\ 1 & -b_{sn} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & -b \end{bmatrix} \tag{A.4}$$

And $P(\mathbf{x})$ includes all the high order terms,

$$P(\mathbf{x}) = \begin{bmatrix} p(v) \\ -w_{sn}b \end{bmatrix} \tag{A.5}$$

where

$$p(v) = -\epsilon^{-1}(v^3 + \frac{a+1}{2}v^2) \tag{A.6}$$

It is clear that the determinant of L is zero and it has two eigenvalues,

$$\Lambda = \begin{bmatrix} \lambda & \\ & 0 \end{bmatrix} = \begin{bmatrix} (\epsilon b_{sn})^{-1} - b_{sn} & \\ & 0 \end{bmatrix} \tag{A.7}$$

And the two corresponding eigenvectors are,

$$\xi_1 = \begin{bmatrix} 1 \\ \epsilon b_{sn} \end{bmatrix}, \quad \xi_2 = \begin{bmatrix} 1 \\ b_{sn}^{-1} \end{bmatrix} \tag{A.8}$$

Now, we apply a transformation $\mathbf{x} = T\mathbf{y}$ in Equation A.3 where $T = [\xi_1 \ \xi_2]$ in order to diagonalize the Jacobian L,

$$\dot{\mathbf{y}} = T^{-1}LT\mathbf{y} + T^{-1}P(T\mathbf{y}) \tag{A.9}$$

where

$$T^{-1} = \begin{bmatrix} b_{sn}^{-1}c & -c \\ -\epsilon b_{sn}c & c \end{bmatrix}, \quad c = \frac{1}{b_{sn}^{-1} - \epsilon b_{sn}} = \frac{1}{\epsilon\lambda} \tag{A.10}$$

So the equation will be

$$\begin{aligned} \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} &= \left(\begin{bmatrix} (\epsilon b_{sn})^{-1} - b_{sn} & \\ & 0 \end{bmatrix} + b \begin{bmatrix} \epsilon b_{sn}c & b_{sn}^{-1}c \\ -\epsilon b_{sn}c & -b_{sn}^{-1}c \end{bmatrix} \right) \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} q_1(y_1, y_2) \\ q_2(y_1, y_2) \end{bmatrix}, \\ q_1(y_1, y_2) &= b_{sn}^{-1}cp(y_1 + y_2) + cw_{sn}b = -(\epsilon b)^{-1}c((y_1 + y_2)^3 + \frac{a+1}{2}(y_1 + y_2)^2) + cw_{sn}b \\ q_2(y_1, y_2) &= -\epsilon bcp(y_1 + y_2) - cw_{sn}b = bc((y_1 + y_2)^3 + \frac{a+1}{2}(y_1 + y_2)^2) - cw_{sn}b \end{aligned} \tag{A.11}$$

We substitute the expansions

$$h(y_2, b) = \mu_1 y_2^2 + \mu_2 b y_2 + \mu_3 b^2 + O(3), \quad h^{'}(y_2) = 2\mu_1 y_2 + \mu_2 b + O(2) \tag{A.12}$$

into the equation blow,

$$h^{'}(y_2)q_2(h(y_2), y_2) - \lambda_1 h(y_2) - p_1(h(y_2), y_2) = 0 \tag{A.13}$$

Setting the coefficients of like powers of y_2 equal to zero yields

$$\mu_1 = \frac{(a+1)c^2}{2b}, \quad \mu_2 \approx 0 \tag{A.14}$$

Thus,

$$h(y_2) = \frac{(a+1)c^2}{2b} y_2^2 + O(y_2^3) \tag{A.15}$$

Substituting this result into the equation below

$$\dot{y_2} = q(h(y_2), y_2), \tag{A.16}$$

we obtain

$$\dot{y_2} = -cw_{sn}b - \frac{c}{b_{sn}} by_2 + \frac{(a+1)b_{sn}c}{2} y_2^2 - (\frac{3c^2}{2\lambda}(a+1) + \frac{c^3 w_{sn}}{\epsilon\lambda^2}(a+1)^2) by_2^2 + O(3) \tag{A.17}$$

Appendix B

Silicon Neurons and Networks

This section aims to introduce the basic model used throughout this thesis. It is based on simplifying the Hodgkin-Huxley's complicated model. But many qualitative features of neuronal spike generation are well retained. Hence, in this brief introduction, the detailed electrophysiology of neurons are omitted to a great extent.

As illustrated above, historically the dynamics of individual neuron is often neglected when building the mathematic models of neural networks. But these intrinsic dynamics are increasingly realized to have great computational properties. Ermentrout and Terman (2010) concludes that population rhythms are generated through interaction of three network components:

- The intrinsic properties of neurons
- The synaptic properties of neural connections
- Network structure

The general model for the whole system of a neural network described in this thesis can be expressed as the form

$$\dot{\mathbf{x}}_i = \mathbf{f}_i(\mathbf{x}_i, t) + \mathbf{C}_i, \quad \mathbf{x}_i \in \mathbb{R}^{M_i} \tag{B.1}$$

where the vector $\dot{\mathbf{x}}_i$ is the state variables(also called state vector) of a single neural system i. $\mathbf{f}_i(\cdot)$ denotes the nonlinear vector functions of correspondent i-th single neural system. $\mathbf{f}_i(\cdot)$ is assumed to be continuous and sufficiently smooth. $\mathbf{C}_i$ is the coupling term of the system. M_i denotes the number of dynamical variables of the i-th single neuron. And moreover, the total number of neurons within the network is denoted by N. A network is composed of nodes and edges, with neurons identified with the nodes and synaptic connections representing edges. In detail, a neuron noted by a state vector can be described in terms of its dynamics without any coupling. For simplicity, they are briefly studied via a general two-state dynamics, $\mathbf{x} = (v, w)^T$ in this case, which can be written in the form

$$\begin{aligned} \frac{dv}{dt} &= f(v, w) \\ \frac{dw}{dt} &= \epsilon g(v, w) \end{aligned} \tag{B.2}$$

where the time t in (B.1) is implicitly included, so that this system is indeed presented by a system of ordinary differential equations. The state variable v represents the membrane potential of the cell while the state variable w is called channel gating variable. And due to it change very shortly compared to v, it is called also called a slow variable. And the parameter ϵ is a small positive parameters. So, the oscillator state as a point in the phase portrait will travel much slower along the w-axis, namely up or down, than along the v-axis, namely left and right. One of the most significant feature of this fast-slow two-dimension dynamics is that most of the state vectors in the phase plane will look parallel to the axis.

This two-variable neuron, also called two-order system, two-dimensional system, or two-state system, is a widely used model by both physiologists and modellers due to its simplicity. Many neuron models belong to this classification including the Morris-Lecar model and the models analyzed in Chapter 2 and 3 belong to this class. The dynamics of this sort of models can mainly be characterized by the nonlinearity of their nullclines. The v-nullcline of the Morris-Lecar Model defines a cubic-shaped curve and the w-nullcline defines a monotonically increasing curve. The variation of the nullclines will cause the qualitative changes in the behavior of dynamical systems, called *bifurcation* (Izhikevich, 2006).

The most fundamental bifurcations which a normal dynamic neuron, such as the Morris-Lecar model, possesses are referred to the qualitative changes on the dynamics of system states, such as, the emergence or vanishing of equilibrium and limit cycles or the variation on their stabilities (Izhikevich, 2006). Different combination of these qualitative changes lead to different neuron dynamics and excitabilities. In order to easily understand the influence of these dynamics and excitabilities of a single cell on whole system, Ermentrout and Terman (2010) classifies the cells generally as *excitable* and *oscillatory*. Chapter 2 of this thesis mainly introduce how a large network made up by all *excitable* cells completes the image processing through a system of reaction diffusion equations. And Chapter 3 will focus on the properties and applications of a network made up by all *oscillatory* neurons.

Apart from the threshold and rate-coded models created in the time of early generations of artificial neural networks, a large number of circuit models mimicking real neurons have also been proposed via CMOS process with schematic diagrams, experiment validation and even layout design. The most direct application of a silicon neuron is for the neurophysiological simulation of a large scale network which is beyond the capability of digital computers. These physically implemented silicon neurons emulate neuronal behavior instead of solving systems of differential equations. And except neuron emulation, this parallel feature of silicon neuronal cells is also believed powerful in the applications where large amount of computations is required such as in motor controls and image processing.

A series of fundamental design methodologies and elementary circuit devices of CMOS technology, has been well concluded by Mead in (Mead, 1989) in order for neural network design. Basically, the approach to derive a circuit model of dynamical neuron can be figured out directly from Hodgkin and Huxley's equations. In (Mahowald and Douglas, 1991), Mahowald and her colleague present a pioneering job of implementing a whole conductance-based Hodgkin-Huxley type of neuron via CMOS process. This model is quite accurate in representing biological neuron but consumes silicon area. Hence, the simplified versions of the neuron model such as FitzHugh Nagumo model and Morris-Lecar model are popular for circuit designers. In these models, it is very valuable that the probability variables representing the gating state of activation and inactivation is replaced by a current-like feedback gating variable to simply depolarize or hyperpolarize the membrane potential. And the resultant reduction of the system dimension thoroughly facilitate the circuit design. However, designing these types of simplified neurons commonly have to face up to two main problems. One is that all these types of reduced two dimensional neuron models intrinsically have inductors. Conventionally, coiling spiral CMOS inductor in VLSI technology is very costly in silicon area. So it has very specific application such as RF circuits of low inductances or the design both precision and bandwidth are highly required. The other problem is to deal with the complex nonlinearity of the models.

Linares-Barranco et al. (1991) implemented and fabricate an analog VLSI chip of FitzHugh Nagumo model in CMOS 2-μm process. They proposed a piecewise linear function synthesis technique to present the nonlinearity of the cubic nullcline of the model and adopted a transconductance mode of circuit structure to avoid the emergence of inductors. Then, based on the basic design, they simplified the oscillator by replacing the inductance with a hysteresis comparator to achieve a compact design. Cosp et al. (2008) also provided a design of compact VLSI FitzHugh Nagumo neuron consisting of only 17 small transistors. They consider that the active inductor of low quality factor due to the low parasitic series resistances is very suitable for FitzHugh Nagumo model as this kind of model is coincidentally found to require a series resistance.

The nonlinearity of Morris-Lecar model is relatively difficult to implement than FitzHugh-Nagumo model for its both cubic v-nullcline and sigmoid w-nullcline. Patel and DeWeerth (1997) developed a silicon neuron of similar characteristics to those of Morris-Lecar model in a 2 μm n-well process. This design utilizes the ohmic effect of transistors to implement the nonlinearity of Morris-Lecar model. It is a current mode circuit of a single neuron which consumes 22 transistors.

Although these two-state models are substantially simpler than the original conductance-based ones for the neurophysiologists, circuit modelers still find them too complicated. Hence, considerably more attention is devoted to a group of very simple models called Integration and Fire models which is written in the form (Izhikevich, 2006)

$$\dot{v} = I - v, \qquad if \quad v = V_t, \quad then \quad v \leftarrow 0, \tag{B.3}$$

where v represents the membrane potential and I is the synaptic input current. The very simple conception of neuron excitability is incorporated by this model that if the membrane potential v reaches the threshold V_t, the voltage sensitive current will immediately activate so that an action potential will be elicited. And soon after this, the membrane potential will be reset to new value below the resting state.

Mead (1989) developed a simple circuit implementation of this model, "Axon-Hillock" circuit, which consumes only 8 transistors. It incorporates the neuronal computational properties of class I excitability (Hodgkin and Huxley, 1952) (However, the threshold for spike emission is intrinsically determined) and the refractory period, but no frequency adaptation. Moreover, another obvious drawback of this circuit is the huge power consumption due to its own structure. Thereby based on that, a series of improvements have been proposed. Schultz and Jabri (1995) presented an alternative circuit of which the output spike frequency adapts the input and the spiking threshold can be externally modulated. Boahen (1998) also presented that a structure of integrate and fire models connected with current-mirror integrator in negative-feedback mode can implement spike-frequency adaptation. However, Schultz's circuit still suffer from the large power consumption and obviously Boahen's added structure increase the circuit area. With the respect to the reduction of power consumption, Schaik (2001) proposed a method to make the poor power condition improved, but not to thoroughly reduce it. And in (Culurciello et al., 2001), Culurciello and his colleagues fully discussed the method to optimize the power consumption but neglecting the model's basic functionalities. Inspired by Culurciello's method, Indiveri (2003) proposed a compact design which is successful in mimicking the neuron behaviors and reducing the power consumption. And the relative work of characterizing this circuit as well as presenting the experiment data was provided in (Rubin et al., 2004).

The main advantage of the integrate and fire neuron is obvious that it can be implemented in very compact structure, normally consisting of 8 to 20 transistors, due to the reduction to the one-state equation. However, it looses many valuable computational properties of real neurons. In spite of that, it is still worth noting that one important application of these models (Rubin et al., 2004) is to reproduce the spike-frequency adapting behavior exhibited by most pyramidal cells in neocortex and hippocampus. Similar to the integrate and fire models greatly incorporating the saddle-node bifurcation mechanisms of those neural integrators, the Andronov-Hopf bifurcation mechanisms of those neural resonators can also be presented by a simplified model called resonate and fire models. Nakada *et al.* proposed a circuit design of such a model via CMOS design with low power consumption to present various dynamical behaviors of real neurons, such as fast damped subthreshold oscillation, frequency preference (bandpass filter) and post-inhibitory rebound.

Further more, by combining both the features of integrators and resonators together, Izhikevich (2006) proposed a smart model of very simple nonlinearity to reproduce overall 20 neurocomputational properties which the cortex neurons commonly have. And he proved it to be the simplest possible model which exhibits all these neurocomputational properties. And inspired by this model, Wijekoon and Dudek (2006, 2008) built a circuit consuming only 14 transistors in 0.35 μm technology with acceptable energy efficiency.

References

Aizenberg, I., Aizenberg, N., and Vandewalle, J. (1998). Precise edge detection: representation by boolean functions, implementation on the CNN. In *Proceeding of the Fifth IEEE International Workshop on Cellular Neural Networks and their Applications, London, UK*, pages 301–306.

Arthur, J. and Boahen, K. (2011). Silicon-neuron design: a dynamical system approach. *IEEE Transactions on Circuits and Systems*, 58:1034–1043.

Basu, A. and Hasler, P. (2010). Nullcline-based design of a silicon neuron. *IEEE Transactions on Circuits and Systems*, 7:421–432.

Black, M., Sapiro, G., Marimont, D., and Heeger, D. (1998). Robust anisotropic diffusion. *IEEE Transactions on Image Processing*, 7:421–432.

Boahen, K. (1998). Communication neuronal ensembles between neuromorphic chips. In *Neuromorphic Systems Engineering*, pages 229–259.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization.* Cambridge University Press.

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

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

Chua, L. and Yang, L. (1988). Cellular neural networks: Theory and applications. *IEEE Transaction on Circuits and Systems*, 35:1257–1272.

Cosp, J., Binczak, S., Madrenas, J., and Fernández, D. (2008). Implementation of compact VLSI FitzHugh-Nagumo neurons. In *In Proceeding of ISCAS, 2008*, pages 2370–2373.

Culurciello, E., Etienne-Cummings, R., , and Boahen, K. (2001). Arbitrated address-event representation digital image sensor. *Electronics Letters*, 37:1443–1445.

Denisov, A. M., Krylov, A. S., and Medvedeva, V. Y. (2010). Edge detection using reaction-diffusion equation with variable diffusion coefficient. In *In Proceeding of The 20th International Conference on Computer Graphics and Vision, GraphiCon2010*, pages 129–132.

Ebihara, M., Mahara, H., Sakurai, T., Nomura, A., Osa, A., and Miike, H. (2003). Segmentation and edge detection of noisy image and low contrast image based on a reaction-diffusion model. *Journal of the Institute of Image Electronics Engineers of Japan*, 32:378–385.

Eckhorn, R., Reitboeck, H., Arndt, M., and Dicke, P. (1989). A neural network for feature linking via synchronous activity: results from cat visual cortex and from simulations. In *Models of Brain Function*, pages 255–272.

Ermentrout, G. and Terman, D. (2010). *Mathematical Foundation of Neuroscience.* Springer.

Fan, C. (1992, revised 1997). *Spectral Graph Theory.* American Mathematical Society.

FitzHugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal*, 1:445–466.

Gonzalez, R. and Woods, R. (2008). *Digital Image Processing.* Pearson Prentice Hall.

Heath, M., Saida, S., Osa, A., and Munechika, K. (1997). A robust visual method for assessing the relative performance of edge-detection algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:1338–1359.

Hebb, D. (1949). *The Organizations of Behaviour.* Wiley.

Heiligenberg, W. (1991). *Neural Nets in Electric Fish.* MIT Press.

Hodgkin, A. and Huxley, A. (1952). A quantitative description of membrane current and application to condition and excitation in nerve. *The Journal of Physiology*, 117:500–544.

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. In *Proceedings of National Academy of Sciences USA*, number 79, pages 2554–2558.

Hoppensteadt, F. C. and Izhikevich, E. M. (1997). *Weakly Connected Neural Networks.* Springer.

Indiveri, G. (2003). A low-power adaptive integrate-and-fire neuron circuit. In *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on*, pages 820–823.

Izhikevich, E. (2006). *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting.* The MIT Press.

Karahaliloğlu, K. and Balkır, S. (2004). An cmos cell circuit for compact implementation of reaction-diffusion models. In *In Proceedings of Neural Networks. 2004 IEEE International Joint Conference*, pages 2831–2836.

Kurata, N., Kitahata, H., Mahara, H., Nomura, A., Miike, H., and Sakurai, T. (2008). Stationary pattern formation in a discrete excitable system with strong inhibitory coupling. *Physical Review E*, 18:289–299.

Kuwabara, N. and Suga, N. (1993). Delay lines and amplitude selectivity are created in subthalamic auditory nuclei: the brachium of the inferior colliculus of the mustached bat. *Journal of Neurophysiology*, 69:1713–1724.

Li, Z. (2001). Computational design and nonlinear dynamics of a recurrent network model of the primary visual cortex. *Neural Computation*, 13:1749–1780.

Linares-Barranco, B., Sanchez-Sinencio, E., Rodriguez-Vazquez, A., and Huertas, J. (1991). A CMOS implementation of FitzHugh-Nagumo neuron model. *IEEE Journal of Solid State Circuit.*, 26:956–965.

Maass, W. (1997). Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10:1659–1671.

Maass, W. (2001). On the relevance of time in neural computation and learning. *Theoretical Computer Science*, 261:157–178.

Mahowald, M. and Douglas, R. (1991). A silicon neuron. *Nature*, 354:515–518.

McCulloch, W. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133.

Mead, C. (1989). *Analog VLSI and Neural Systems.* Addison-Wesley.

Mizoguchi, N., Nagamatsu, Y., Aihara, K., and Kohno, T. (2011). A two-variable silicon neuron circuit based on the izhikevich model. *Artificial Life and Robotics*, 16:383–388.

Nagumo, J., Arimoto, S., and Yoshizawa, S. (1962). An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50:2061–2070.

Nomura, A., Ichikawa, M., Miike, H., Ebihara, M., Mahara, H., and Sakurai, T. (2003). Realizing visual functions with the reaction-diffusion mechanism. *Journal of the Physical Society of Japan*, 72:2385–2395.

Nomura, A., Ichikawa, M., Okada, K., H, M., Sakurai, T., and Mizukami, Y. (2011a). Image edge detection with discrete spaced FitzHugh-Nagumo type excitable elements. In *Image Processing*, pages 1–8.

Nomura, A., Ichikawa, M., Okada, K., Miike, H., and Sakurai, T. (2011b). Edge detection algorithm inspired by pattern formation processes of reaction-diffusion systems. *International Journal of Circuits, Systems and Signal Processing*, 5:105–115.

Nomura, A., Ichikawa, M., Sianipar, R. H., and Miike, H. (2008). Edge detection with reaction-diffusion equations having a local average threshold. *Pattern Recognition and Image Analysis*, 18:289–299.

Oseledec, V. (1968). A multiplicative ergodic theorem: Lyapunov characteristic numbers for dynamical system. *Transactions of the Moscow Mathematical Society*, 16:197–231.

Palis, J. and de Melo, W. (1982). *Geomeric Theory of Dynamical Systems: An Introduction*. Springer-Verlag.

Parker, T. and Chua, L. (1989). *Pratical Numerical Algorithms for Chaotic Systems*. Springer-Verlag.

Patel, G. and DeWeerth, S. (1997). Analogue VLSI Morris-Lecar neuron. *Electronics Letters*, 33:997–998.

Perko, L. (1991). *Differential Equations and Dynamical Systems*. Springer-Verlag.

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

Rubin, D.-D., Chicca, E., and Indiveri, G. (2004). Firing properties of an adaptive analog VLSI neuron. In *Biologically Inspired Approaches to Advanced Information Technology First International Workshop, BioADIT 2004*, pages 189–200.

Sandri, M. (1996). Numerical calculation of lyapunov exponents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:78–84.

Schaik, V. (2001). Building blocks for electronic spiking neural networks. *Neural Networks*, 14:617–628.

Schultz, S. and Jabri, M. (1995). Analogue VLSI integrate-and-fire neuron with frequency adaptation. *Electronic Letters*, 31:1357–1358.

Suzuki, K., Horiba, I., and Sugie, N. (2003). Neural edge enhancer for supervised edge enhancement from noisy images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1582–1596.

Suzuki, K., Horiba, I., Sugie, N., and Nanki, M. (2004). Extraction of left ventricular contours from left ventriculograms by means of a neural edge detector. *IEEE Transactions on Medical Imaging*, 23:330–339.

Thomas, J. (1995). *Numerical Partial Differential Equations: Finite Difference Methods*. Springer-Verlag.

Thorpe, S. (1990). Spike arrival times: A highly efficient coding scheme for neural networks. In *Parallel processing in neural systems and computers*, pages 91–94.

Uhlenbeck, G. and Ornstein, L. (1930). On the theory of the Brownian motion. *Physical Review*, 36:823–841.

Wijekoon, J. and Dudek, P. (2006). Simple analogue VLSI circuit of a cortical neuron. *IEEE ICECS*, pages 1344–1347.

Wijekoon, J. and Dudek, P. (2008). Compact silicon neuron circuit with spiking and bursting behaviour. *Journal of Neural Networks*, 21:524–534.

Yen, S. and Finkel, L. (1998). Extraction of perceptually salient contours by striate cortical networks. *Vision Research*, 38:719–741.

Yu, G. and Slotine, J. (2009). Visual grouping by neural oscillator neural networks. *IEEE Transactions on Neural Network*, 20:1871–1884.

Zaghloul, K. and Boahen, K. (2006). A silicon retina that reproduces signals in the optic nerve. *Journal of Neural Engineering*, 3:257–267.