

Dynamical System Approach to Edge Detection Using Coupled FitzHugh-Nagumo Neurons

Shaobai Li, Srinandan Dasmahapatra, and Koushik Maharatna

Abstract—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. For instance, simple image processing tasks such as the detection of edges in binary and grayscale images have been performed by networks of neurons of the Fitzhugh-Nagumo type, using reaction diffusion models. However, in these studies, the one-to-one mapping of image pixels to component neurons makes the size of the network a critical factor in any such implementation. In this paper, we offer a simplified version of the reaction diffusion model employed, where we perform three steps. In our first step, we undertake a detailed study to locate this threshold using continuous Lyapunov exponents from dynamical systems theory. Further, we render the diffusion in the system to be anisotropic, with the degree of anisotropy being set by the gradients of grayscale values in each image. The final step involves a simplification of the model by eliminating the terms that couple the membrane potentials of adjacent neurons. We apply our technique to detect edges in data sets of artificially generated and real images, demonstrating performance that is as good if not better than the previous results without increasing the size of the network.

Index Terms—Excitability, Lyapunov exponent, edge detection, FitzHugh-Nagumo model, reaction-diffusion system.

I. INTRODUCTION

BIOLOGICAL neural networks carry out massively parallel, robust computation while dissipating low power, properties that have brought many neuroscientists and neuromorphic system designers under their spell. Neuroscientists Hodgkin and Huxley [1] modelled the generation and propagation of spikes in the giant axon of the squid by using nonlinear partial differential equations; these seminal neurophysiological models are analyzed, simplified and further developed with the help of dynamical systems theory (DST)[2], [3]. On the engineering end, Mahowald and Douglas [4] 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 neuronal models described by dynamical systems into corresponding integrated circuits have been thoroughly studied and a large number of silicon neurons and neuromorphic silicon chips have been built [5], [6], [7]. 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 [8]; 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 [9], have been designed, they perform image processing tasks at a lower resolution (normally 50×50 pixels) compared to digital processors. Hence, system size is a most important issue when designing a neurally inspired circuit for the purpose of image processing.

Nomura *et al.* [10], [11], [12], [13] and Kurata *et al.* [14] use a discrete reaction diffusion system (a type of neuronal network) for detecting edges in real grayscale images. Such kinds of reaction-diffusion networks are widely built using CMOS technology [15]. Within the rectangular grid network, each node is a simplified version of a Hodgkin-Huxley neuron [1], described by a Fitzhugh-Nagumo [2], [3] model, and each neuron is connected only with all four in its neighbourhood with diffusive coupling. Their algorithm used a constant threshold to detect edges (white-dark transitions) in binary images. In contrast, the threshold was chosen to vary with the gradients in grayscale images in order to detect edges in them [12]. However, in doing so, they encountered some problems. For instance, they found the appearance of wrong edges (as shown in Fig 5 (b,c)) and had to enlarge the size of the system to eliminate them and improve system performance [12]. In this paper, in the high level design procedure, we try to fix the size of the system and look for such an alternative solution to the problem using DST. First, we define the image threshold for a monostable FitzHugh-Nagumo model (the model with parameter values chosen so that only one stable fixed point exists) by evaluating the maximum Lyapunov characteristic exponent; this enables us to correctly formulate the problem introduced when extending the algorithm to grayscale images. Next, upon analysing the dynamics of the smallest network, that of two coupled FitzHugh-Nagumo models, we are led to a simplification of the network system by eliminating the coupling term in the membrane potential. Finally, we solve the problem of spurious edges using a novel method of obtaining the threshold by anisotropic diffusion. As a result, the performance of our system in detecting edges is improved for artificial images and remains comparable to Nomura’s method [13] while halving the network size.

In Section II, we provide the analysis on a single FitzHugh-Nagumo type of neuron using DST in detail, in order to present the basic mechanism of edge detection drawing upon the *excitability* of single neuron. Then, we provide a stability analysis on the whole network model based on the theory of weakly connected neural network [16]. We also provide a brief review of the edge detection algorithm proposed by Nomura *et al.* In Section III, we describe the ensuing problem using an example of detecting edges in one dimension. In Section

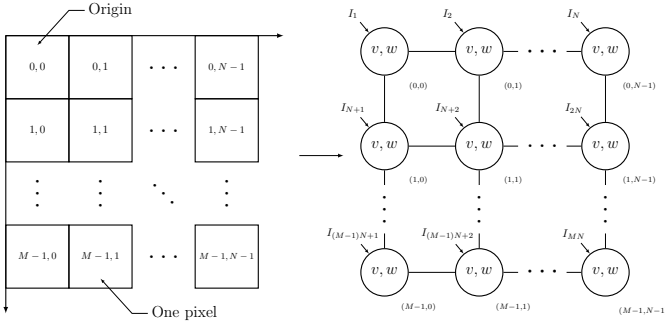


Fig. 1: One pixel to one neuron network structure

IV, we present our novel method that sets as a threshold the anisotropic diffusion of the processed image. In the last section, we compare our work with the previous algorithm by presenting the edge detection result for both artificial and real images.

II. NOMURA'S ALGORITHM OF EDGE DETECTION

This section aims to provide a brief review of the edge detection algorithm by Nomura *et al.*[17], [14], [10], [11], [12], [13]. We will discuss the mechanism underlying the detection of edges using dynamical system in two steps. First, we introduce the notion of a threshold that determines the excitable response of a single uncoupled neuron, a concept that underlies the edge detection method. Secondly, we examine the stability of the state that corresponds to the detected edges and explain the ability of the model to detect edges using Izhikevich's theory of weakly coupled neural networks (WCNN)[16].

Throughout this paper, the processed digital image is defined as a matrix of pixels. We use U to denote the image of size $M \times N$ and $U_{(m,n)}$ to denote the intensity of the pixel with the coordinate (m, n) . And the edge detection task aims to find a binary *edge map* denoted by $\mathcal{M} : U \rightarrow \{0, 1\}^{MN}$,

$$U_{(m,n)} \mapsto \mathcal{M}(U_{(m,n)}) = \begin{cases} 0, & (m, n) \in S_{\mathcal{M}} \\ 1, & (m, n) \notin S_{\mathcal{M}} \end{cases} \quad (1)$$

where $S_{\mathcal{M}}$ is the set of the coordinates of pixels along all edges in the image. We shall also use the symbol $\widetilde{\mathcal{M}}$ to denote the putative edge map obtained by edge detection algorithms that are evaluated with respect to the ground truth \mathcal{M} . As shown in Fig. 1, the neuronal system for the purpose of edge detection is a $M \times N$ rectangular grid, with one node per pixel in the image.

The neural network model can be described by the following set of ordinary differential equations:

$$\begin{cases} \dot{v}_i = f(v_i, w_i, a_i) + k_v \sum_{j \in \mathcal{P}_i} (v_j - v_i) \\ \dot{w}_i = g(v_i, w_i) + k_w \sum_{j \in \mathcal{P}_i} (w_j - w_i), \end{cases} \quad (2)$$

where i and j are the indices labelling each element in the network. v_i and w_i are the state variables respectively for the *membrane potential* and the *channel gating variable* of a

neuron. f_i and g_i describe the dynamics of a single element. a_i works as an estimated threshold of two different dynamical behaviours of a single element. The constants k_v and k_w denote the coupling strength. \mathcal{P}_i is the set consisting of the 4-neighbours of the i -th element in the network.

For the sake of convenience of expression, we use both, the index i and coordinates (m, n) to label the network element and the corresponding pixel in the processed image. There is a mapping, $\mathcal{F} : (m, n) \mapsto i$:

$$\begin{aligned} \mathcal{F} : (m, n) &\mapsto i, \quad i = n \cdot M + m + 1 \\ \mathcal{F} : i &\mapsto (m, n), \quad m = (i - 1) \bmod M \\ n &= \lceil \frac{i}{M} \rceil, \end{aligned} \quad (3)$$

where $\lceil r \rceil$ is the smallest integer larger than or equal to real number r . Hence, for $i \mapsto (m, n)$, \mathcal{P}_i is the set of coordinates $(m + 1, n)$, $(m - 1, n)$, $(m, n + 1)$, $(m, n - 1)$. For the pixels at the boundaries or corners, the zero boundary condition is imposed:

$$\begin{aligned} v(-1, n) &= v(0, n), \quad v(M, n) = v(M - 1, n), \\ v(m, -1) &= v(m, 0), \quad v(m, N - 1) = v(m, N). \end{aligned} \quad (4)$$

The initial condition of each neuron in the network equation (2) is given by the following expressions,

$$\begin{cases} v_i(0) = U^r_{(m,n)} = \xi U_{(m,n)}, \quad \xi \ll 1 \\ w_i(0) = 0 \end{cases} \quad (5)$$

where $U^r_{(m,n)}$ indicates the rescaled light intensity of the pixel (m, n) and ξ is the rescaling coefficient. Following [13], the image intensities are rescaled to lie in the interval $[0, 1/4]$. So ξ is equal to $\frac{1}{1024}$ for 8-bits grayscale images. It can be seen that each membrane potential v_i takes a rescaled pixel intensity $U^r_{(m,n)}$ as the initial condition according to the equation (5). However, w_i , which is initially set to zero, does not have a specific meaning in terms of the image description. By setting the initial condition, the system (2) will automatically evolve to the steady state after a certain period of time τ_s , and the final edge map $\widetilde{\mathcal{M}}$ will be determined by $v(\tau_s)$.

A. Binary Edge Detection Algorithm

In this section, we will mainly focus on the analysis on a single element of the network without the coupling. Each element of the network is described by a Fitzhugh-Nagumo model

$$\begin{cases} \dot{v} = f(v, w) = \frac{1}{\varepsilon} [v(1 - v)(v - a) - w] \text{ and} \\ \dot{w} = g(v, w) = v - bw, \end{cases} \quad (6)$$

where ε , a and b are all positive parameters. ε is the ratio of time scales at which the two variables evolve and b controls the slope of the nullcline $\dot{w} = 0$. We will show later that a is a threshold separating choices of initial conditions which induces two qualitatively different system trajectories for a single neuron when $\varepsilon \ll 1$.

Fig. 2 provides three phase portraits of monostable Fitzhugh-Nagumo model without couplings for different values of the parameter ε . As shown in Fig. 2a, one solution

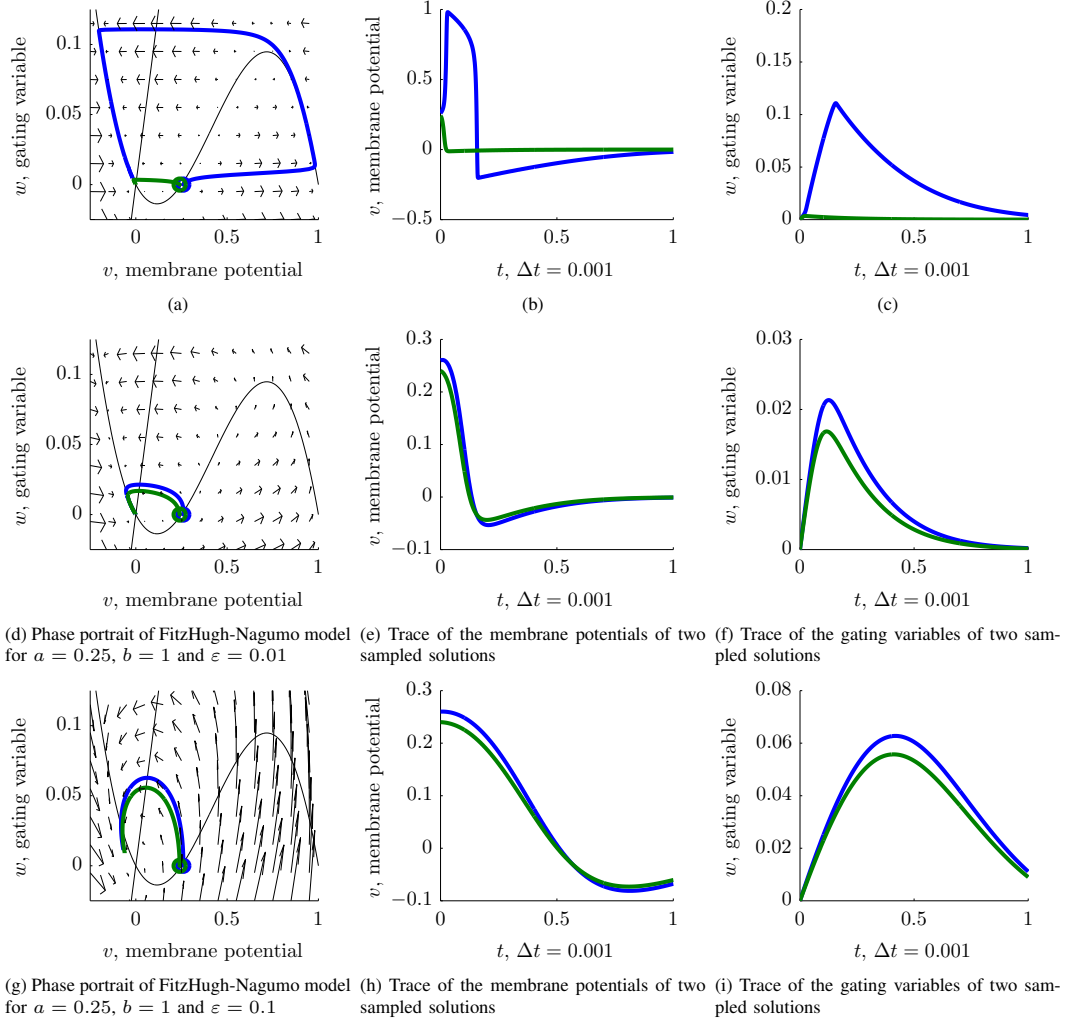


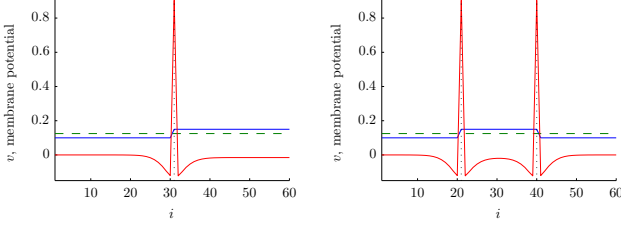
Fig. 2: Comparison of excitability of FitzHugh-Nagumo model in equation (6) with different parameter ε . The graphs in the most left column provide the vector fields and phase portraits of the FitzHugh-Nagumo model respectively with ε equal to (a) 0.001, (d) 0.01 and (g) 0.1. The values of other parameters are the same as $a = 0.25$, $b = 1$. In each phase portrait, two trajectories of solutions to the FitzHugh-Nagumo model start at two different initial values; $(v_0, w_0) = (0.24, 0)$ for the blue line and $(v_0, w_0) = (0.26, 0)$ for the green line. The trajectories separate most widely when $\varepsilon = 0.001$. The middle and right column provide both the trajectories of v and w in time domain corresponding to each phase portrait.

trajectory with the initial condition $(v(0), w(0)) = (0.24, 0)$ directly goes to the equilibrium at $(0, 0)$ while the other with $(v(0), w(0)) = (0.26, 0)$ 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 Fig. 2b, one directly shrinks to zero, whereas the other one goes up to nearly 1, and then rapidly down to a negative value followed by a gradual return to the t axis. This trajectory describes a pulse in time, and is referred to as an *action potential* or *spike* in the neuronal context. So, the parameter a for each neuron works as a threshold separating the solution trajectories into two different states even though both return to the origin: one directly goes back to the origin with a small amplitude signal (*resting state*) and the other generates a single spike (*excited state*). This property is referred to as *excitability*.

For a simple case of an edge in a processed image where two areas of different intensities are adjacent and the threshold a is set to an intermediate value between these two intensity values, one can imagine that these two areas will be divided as two different states as shown in Fig. 2a. So within either

of the two areas, there is no big difference among the pixels: Namely, $v_i(t) \approx v_j(t)$ and $w_i(t) \approx w_j(t)$ for $t > 0$, then the coupling terms $v_j - v_i$ and $w_j - w_i$ will be very small so that the coupled system will behave exactly like the uncoupled one, resulting in a state with $v(t \gg 1) \sim 0$. However, for the pixels at the boundary between two areas, the differences between the intensities of adjacent pixels are large so that the coupling cannot be neglected any longer. So, as the final result, only the neurons located at the boundary connecting the higher or lower image intensity levels have the potential to maintain a high membrane voltage level stably over time.

It is worth noting that, different from Fig. 2a, Fig. 2d and 2g present a case where both the solutions starting at $(v(0) < a, w(0) = 0)$ and $(v(0) > a, w(0) = 0)$ are going back to zero towards the same direction and along the similar trajectories. In order to make a single neuron distinguish between different input membrane potentials, a sufficiently small value of ε in equation (2) is necessary. We will take this condition as an important prerequisite. So, in the rest of the paper, we restrict our single model to an excitable FitzHugh-



(a) Edge detection result on the step function equation (7). (b) Edge detection result on the step function equation (8).

Fig. 3: Edge detection results on one dimensional step function equation (7) and (8) with constant image threshold $a = 0.125$. For other parameter settings, $\varepsilon = 0.001$, $b = 1$, $k_v = 4$ and $k_w = 20$

Nagumo type neuron (6) with global mono-stability. And the constant ε is chosen as a small value to ensure a separation of time scales over which the dynamics of the two variables v, w play out.

Fig. 3 provides two examples of using Nomura's algorithm on one dimensional step functions. One dimensional step function is the simplest case of edge profile which widely used to facilitate the analysis on edge detection algorithm (such as in [18]). It can be considered as a two dimensional edge which locally has cross-section in a certain direction. The two step functions used in Fig. 3 are provided below,

$$v_i(0) = U^r_i = \begin{cases} 0.10, & 1 \leq i \leq 30 \\ 0.15, & 31 \leq i \leq 60 \end{cases}, \quad w_i(0) = 0 \quad (7)$$

and

$$v_i(0) = U^r_i = \begin{cases} 0.10, & 1 \leq i \leq 20 \\ 0.15, & 21 \leq i \leq 40 \\ 0.10, & 41 \leq i \leq 60 \end{cases}, \quad w_i(0) = 0 \quad (8)$$

These two function is set as the initial condition to the system (2).

B. Stability of Edge Detection Result

In this section, we will explain the ability of the system to detect edges using DST. According to the classification of neural networks presented in [16], there is one group of the neural networks dealing with static input pattern referred as to *Multiple Attractor Neural Networks*. Such kind of neural networks evolve by themselves with the initial state set as the input pattern, with the state of the network converging to one of many attractors. We will show that the neural network model in equation (2) belongs to the Multiple Attractor type and the edge detection result is obtained from one of its stable equilibria (attractors).

To examine the stability of a fixed point we linearise the system dynamics around a fixed point, thus evaluating the *Jacobian*. Denoting all the state variables of the network by $\mathbf{x} = (v_1, w_1, v_2, w_2, \dots)^T \in \mathbb{R}^{2MN}$, the dynamics of whole network can be rewritten as

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^{2MN} \quad (9)$$

where $\mathbf{F}(\cdot) : \mathbb{R}^{2MN} \rightarrow \mathbb{R}^{2MN}$ is a vector function

$$\mathbf{F}(\cdot) = (f_1(\cdot), g_1(\cdot), f_2(\cdot), g_2(\cdot), \dots)^T \quad (10)$$

Each f_i and g_i is given in equation (6). Let the final edge map \mathcal{M} be mapped to the stable equilibrium $\bar{\mathbf{x}} = (\bar{v}_1, \bar{w}_1, \bar{v}_2, \bar{w}_2, \dots) \in \mathbb{R}^{2MN}$. The Jacobian of the system (9) at the equilibrium $\bar{\mathbf{x}}$ is defined as,

$$(\mathbf{J}(\bar{\mathbf{x}}))_{ij} = \left. \frac{\partial f_i(\mathbf{x})}{\partial x_j} \right|_{\mathbf{x}=\bar{\mathbf{x}}} =: (\mathbf{DF})_{ij} \quad (11)$$

Let $\mu_1(t), \mu_2(t), \dots, \mu_{2MN}(t)$ be the eigenvalues of $\mathbf{J}(\bar{\mathbf{x}})$. The proper way to examine whether $\bar{\mathbf{x}}$ is stable is to check whether $\max_i \mu_i$ is smaller than zero.

We will focus on an example network which has four elements ($M = 1, N = 4$), so $\mathbf{x} \in \mathbb{R}^8$. The parameter setting is $a = 0.2$, $b = 3$ and $\varepsilon = 0.001$ to ensure that the uncoupled network ($k_v = 0$, $k_w = 0$) has only one equilibrium at the origin. We evaluate the Jacobian at the origin and find its largest eigenvalues.

$$\max_i \mu_i|_{\bar{\mathbf{x}}=0} = -8.2141 < 0 \quad (12)$$

Hence, the equilibrium at the origin is stable. When the coupling ($k_v = 1$, $k_w = 5$) is included, we evaluate all the fixed points of equation (9) by seeking real solutions to

$$\mathbf{F}(\mathbf{x}) = 0. \quad (13)$$

We solve these equations by using a MATLAB interface to the PHC package [19] to obtain real solutions. We find that the coupled system has 9 equilibria of which three are stable as listed below

$$\begin{aligned} \bar{\mathbf{x}}^1 &= (0, 0, 0, 0, 0, 0, 0, 0)^T \\ \bar{\mathbf{x}}^2 &= (-0.05, 0.01, -0.10, 0.03, 0.82, 0.09, -0.12, 0.04)^T \\ \bar{\mathbf{x}}^3 &= (-0.12, 0.04, 0.82, 0.09, -0.10, 0.03, -0.05, 0.01)^T \end{aligned} \quad (14)$$

It can be seen that the system has two other fixed points apart from the origin. Hence, under the appropriate parameter settings, the uncoupled network system is monostable but in the vicinity of a *bifurcation*. When the coupling is included, the original mono-stable system becomes multi-stable. By taking the rescaled input image U^r as an initial state, the whole network will converge to one of the possible high dimensional stable equilibria corresponding to the potential edge map $\tilde{\mathcal{M}}$.

C. Grayscale Image Edge Detection Algorithm

The network system (2) using a constant value of a as the threshold can only separate regions of different (0/1) intensities in binary images. In order to extend the application of edge detection to grayscale images, the threshold needs to be adjusted according to the gradients of grayscale values in the processed image. So, Nomura *et al.* [11], [12] proposed a method which chooses a blurred rescaled image intensity distribution as the threshold. Specifically, it is obtained via the diffusion equation,

$$\frac{\partial \theta}{\partial t} = D \left(\frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial y^2} \right), \quad (15)$$

whose solution is

$$\theta(x, y, t) = \frac{1}{4\pi Dt} e^{-(x^2+y^2)/4Dt}. \quad (16)$$

The discrete version of the diffusion equation can be expressed as

$$\dot{\theta}_i = d \sum_{j \in \mathcal{P}_i} (\theta_j - \theta_i), \quad \theta_i(0) = U^r_{(m,n)}, \quad (17)$$

where θ denotes the diffused image intensity distribution on the continuous variables x and y , θ_i is the corresponding discrete distribution on the index i or the coordinates (m, n) . D is the continuous diffusion coefficient and $d = D/(\delta h)^2$ is the discrete diffusion coefficient where δh is the uniform finite difference (grid spacing) for the discretization where $x = m\delta h$ and $y = n\delta h$. The diffusion adopted here is isotropic as reflected in the constant diffusion coefficient D . And blurring the original image U^r by diffusion is equivalent to applying to U^r a Gaussian filter [20], which is the solution (16) to (15). It can be seen from (16) that the extent of diffusion, i.e. how blurred the image is, is determined by the variance $2Dt$ of the Gaussian distribution. So by choosing an appropriate time constant τ and diffusion coefficient D , the parameter a_i can be set as $\theta_{(m,n)}(\tau)$ which is a multilevel threshold and more suitable for grayscale images. However, the result of simply setting a variable threshold is not ideal for the edge detection of grayscale images. For an image with a single edge, such a threshold picks up not only its corresponding pulse at the correct spatial location, but also a second edge close to it. In order to eliminate these false pulses, Nomura *et al.* [12] modified the original system by doubling the number of dynamical variables to $(v_i^0, w_i^0, v_i^1, w_i^1)$ (2) as follows:

$$\begin{cases} \dot{v}_i^0 = f(v_i^0, w_i^0, a_i^0) + k_v \sum_{j \in \mathcal{P}_i} (v_j^0 - v_i^0) + v_i^1 \mathcal{H}(-v_i^1) \\ \dot{w}_i^0 = g(v_i^0, w_i^0) + k_w \sum_{j \in \mathcal{P}_i} (w_j^0 - w_i^0) \\ \dot{v}_i^1 = f(v_i^1, w_i^1, a_i^1) + k_v \sum_{j \in \mathcal{P}_i} (v_j^1 - v_i^1) \\ \dot{w}_i^1 = g(v_i^1, w_i^1) + k_w \sum_{j \in \mathcal{P}_i} (w_j^1 - w_i^1). \end{cases} \quad (18)$$

$\mathcal{H}(x)$ is the Heaviside step function, which gives 1 when $x \geq 0$ and gives 0 when $x < 0$. The superscripts 0 and 1 in each variable identifies the subsystem associated with each pixel location i . Nomura *et al.* [12], [13] claim that the term $v_i^1 \mathcal{H}(-v_i^1)$ in equation (18) works for the elimination of the wrong spatial pulses, if the intensity distribution of a_i^1 has a larger variance than that of a_i^0 . A more detailed description of this problem of spurious edges will be presented in the next section.

Although the algorithm introduced above for the edge detection on grayscale images works in most cases, it still fails in some specific case where darker intensity areas are surrounded by the brighter backgrounds. An improved approach was proposed in [12] by doubling the image space as well: $U_{m,n} \mapsto \{U_{m,n}, \bar{U}_{m,n}\}$, where

$$\bar{U}_{m,n} = 255 - U_{m,n} \quad (19)$$

for 8-bit images. The intersections of the edge maps $\bar{\mathcal{M}}(U_{m,n}) \cap \mathcal{M}(\bar{U}_{m,n})$ obtained by the algorithm introduced above is taken to be the final edge map [12].

The most comprehensive edge detection algorithm proposed by [12], [13] for 8-bit grayscale images is summarized in Algorithm 1.

Algorithm 1 Edge detection algorithm with discrete reaction-diffusion system

- 1: Rescale the image intensity distribution $U^r_{m,n} = \xi U_{(m,n)}$ with $\xi = 1/1024$ so that $0 \leq U^r_{(m,n)} \leq \frac{1}{4}$.
- 2: Solve the equation $\dot{\theta}_i^0 = d^0 \sum_{j \in \mathcal{P}_i} (\theta_j^0 - \theta_i^0)$ with the initial condition $\theta_i^0(0) = U^r_i$ to get a blurred version of rescaled image with a stopping time τ to obtain the threshold value $a_i^0 = \theta^0(\tau)$.
- 3: Solve the equation $\dot{\theta}_i^1 = d^1 \sum_{j \in \mathcal{P}_i} (\theta_j^1 - \theta_i^1)$ by using a larger constant $d^1 > d^0$ with the same initial condition $\theta_i^1(0) = U^r_i$ to get the second threshold value $a_i^1 = \theta^1(\tau)$.
- 4: Solve the equation (18) 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 $\tilde{\mathcal{M}}_1$ via a simple thresholding $v_{(m,n)}^0(\tau_s)$ as below,

$$\tilde{\mathcal{M}}_1(m, n) = \begin{cases} 0, & v_i^0(\tau_s) > 0.5 \\ 1, & v_i^0(\tau_s) \leq 0.5 \end{cases} \quad (20)$$

- 6: Repeat the steps 1 to 5 above to get a second putative edge map $\tilde{\mathcal{M}}_2$ from the black-and-white inversion of the original image $\bar{U}_{(m,n)}$
 - 7: Get the final edge map $\tilde{\mathcal{M}}$ by merging the two putative ones, $\tilde{\mathcal{M}} = \tilde{\mathcal{M}}_1 \cap \tilde{\mathcal{M}}_2$
-

III. 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 the initial condition

$$v_i(0) = U^r_i = \begin{cases} 0.10, & 1 \leq i \leq 50 \\ 0.15, & 51 \leq i \leq 100 \\ 0.20, & 101 \leq i \leq 150 \end{cases}, \quad w_i(0) = 0. \quad (21)$$

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 $a_i = \theta(\tau)$ using the diffusion equation (17), where we choose the constant $d = 10$. With this example, we aim to extensively describe the problem of the "wrong pulses" discovered in [12]. With increasing values of the diffusion time τ , we obtain θ with increasing degrees of blurring, with the consequent results provided in Fig. 5.

Fig. 4 provides a zoomed in view of the local area around both the edge positions in the original step function (21). Because the isotropic diffusion coefficient d is constant for all the neurons and the magnitude of the two step differences in equation (21) 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 Fig. 4a and 4b.

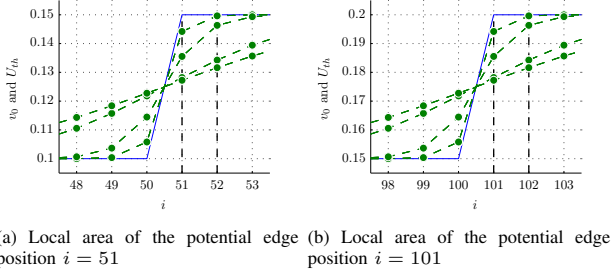


Fig. 4: Image threshold $\theta_i(\tau)$ with τ equal to 0.015, 0.030, 0.060, 0.120, 0.200, 0.300, 0.500 and 1.000, respectively corresponding to the cases provided in Fig. 5

Note that, both the diffused steps are center-symmetric. Considering the right half of the steps in Fig. 4a and 4b, 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 Fig. 5c. The mechanism and outcome is similar to that in the second example in Fig. 3b where there indeed were 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 Fig. 5d.

Hence, we can summarize that the “wrong pulse” [12] is introduced by the method of setting a as variable threshold owing to the following reasons. For a relatively small value of τ , the threshold is not sufficiently diffused, and thus there is no edge detected. And for a relatively larger value of τ , the threshold is spread out too far and a second stable edge is obtained at the wrong position.

IV. NOVEL EDGE DETECTION METHOD

In order to deal with the emergence of the wrong pulse, we introduce a novel algorithm without increasing the size of the system based on the model in equation (2). We first provide a more accurate definition of the image threshold for a monostable FitzHugh-Nagumo model (6) by evaluating the Lyapunov characteristic exponents, which enables us to correctly deal with the problem when extending the algorithm to grayscale images. Next, we further simplify the system (2) by eliminating the coupling of the membrane potential v by exploiting the intrinsic separation of time scales of the dynamics of the (v_i, w_i) variables. Finally, in order to enhance the performance of edge detection, we introduce anisotropic diffusion in equation (17) where d is also determined by the rescaled image U^r .

A. Image Threshold

In Section II-B, we drew the conclusion that the coupled network model in equation (2) has multiple stable equilibria in the high dimensional space \mathbb{R}^{2MN} and one of them corresponds to the putative edge map $\widetilde{\mathcal{M}}$. We use $\bar{x}_{\widetilde{\mathcal{M}}}$ to denote such an expected equilibrium. We denote the attraction domain

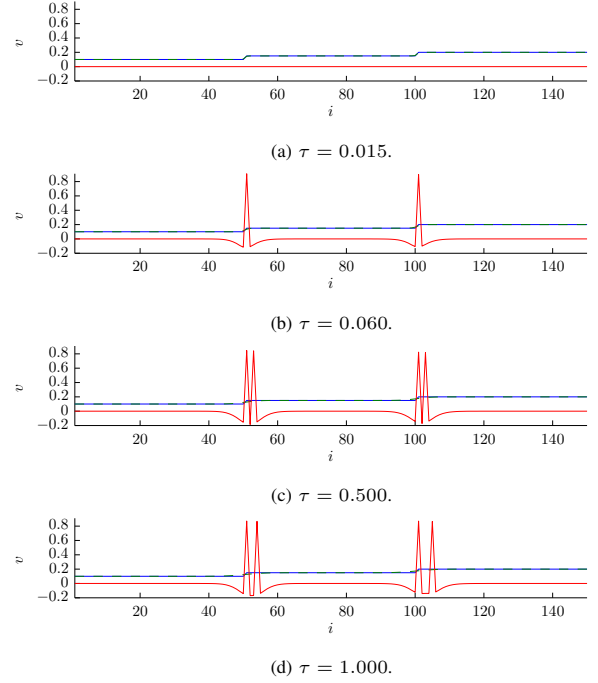


Fig. 5: Edge detection results on one dimensional step function (21) with $a_i = \theta_i(\tau; d = 10)$ which is the variable image threshold obtained from the diffusion equation (17) with different values of τ . For other parameter settings, $b = 1$, $\varepsilon = 0.001$, $k_v = 4$ and $k_w = 20$.

of a stable equilibrium \bar{x} by $\mathcal{B}(\bar{x})$ and by $\partial\mathcal{B}(\bar{x})$ its boundary. The problem of detecting the correct edges in an image using the DST approach, as in the class of models addressed in this paper, is equivalent to the following. First, we determine the existence of a stable equilibrium $\bar{x}_{\widetilde{\mathcal{M}}}$; secondly, we examine whether its attraction domain contains the initial condition given by the rescaled image, $U^r \in \mathcal{B}(\bar{x}_{\widetilde{\mathcal{M}}})$. Hence, we need to evaluate the boundaries of attraction domains $\partial\mathcal{B}(\bar{x})$ of all the stable equilibria in the model and track their changes when system parameters change.

However, it is a very difficult task to find the exact boundaries $\partial\mathcal{B}(\bar{x})$ for such a high dimensional system in equation (9). We note that Nomura *et al.* [10], [11], [12] treat the system parameter a as an estimate of the exact boundary. However, we find that the parameter a is not an accurate threshold of excitability for a single FitzHugh-Nagumo model in equation (6), and is even less likely to be a reasonable estimate of $\partial\mathcal{B}(\bar{x})$ for the coupled network.

In section II-A, we have observed that the single neuron has two categories of the initial conditions, one of which leads to the resting, the other to the excited state. However, there is no explicit boundary between these two states for the individual monostable system (6) used here, no matter how small ε , the ratio of timescales of the fast and slow variables, is. Such a monostable system has a property that its trajectories have a continuous dependence on its initial conditions [21]. Namely, for a system which has only one globally stable equilibrium \bar{x} , if $\mathbf{x}(t; \mathbf{x}_0)$ denotes the solution with the initial condition \mathbf{x}_0 , two solutions $\mathbf{x}(t; \mathbf{x}_0)$ and $\mathbf{x}(t; \mathbf{x}_0 + \delta\mathbf{x}_0)$ could be arbitrarily close, as long as their initial conditions are sufficiently close $\delta\mathbf{x}_0 \rightarrow 0$.

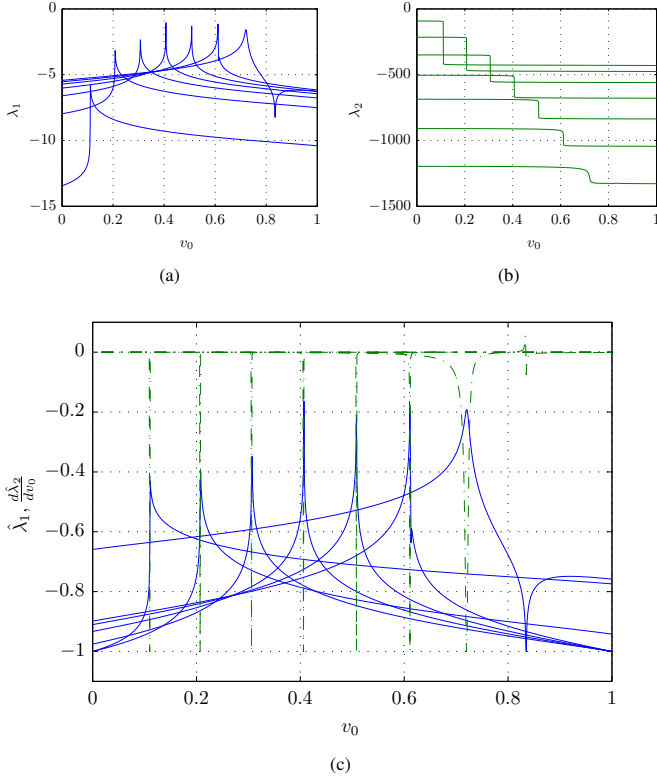


Fig. 6: Curves of Lyapunov exponents of the single FitzHugh-Nagumo model (a) $\lambda_1(v(0))$ (blue solid curves), (b) $\lambda_2(v(0))$ (green solid curves), (c) $\hat{\lambda}_1$ (blue solid curves) and $\frac{d\hat{\lambda}_2}{dv_0}$ (green solid curves). Each curve of λ_1 , λ_2 , $\hat{\lambda}_1$ or $\frac{d\hat{\lambda}_2}{dv_0}$ corresponds to a certain value of a . The extremum in each $\hat{\lambda}_1$ or $\frac{d\hat{\lambda}_2}{dv_0}$ denotes the specific initial condition v_0 where the trajectories of solution to FitzHugh-Nagumo model in equation (6) will separate most widely.

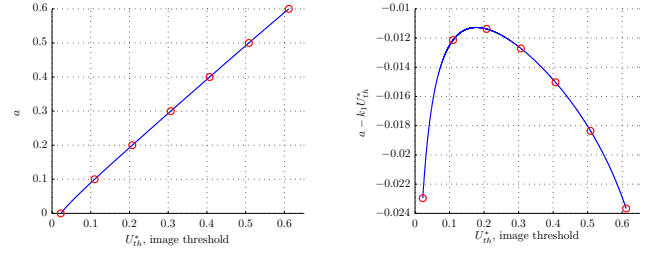
The *Lyapunov exponent* λ measures the rate of divergence of solutions $\mathbf{x}(t; \mathbf{x}_0)$ and $\mathbf{x}(t; \mathbf{x}_0 + \delta \mathbf{x}_0)$, for initial conditions differing by an infinitesimal $\delta \mathbf{x}_0$. Define $\Phi_t(\mathbf{x}_0)$ as the derivative of the trajectory $\mathbf{x}(t; \mathbf{x}_0)$ with respect to the initial condition \mathbf{x}_0 :

$$\Phi_t(\mathbf{x}_0) := \frac{\partial \mathbf{x}(t; \mathbf{x}_0)}{\partial \mathbf{x}_0} \quad (22)$$

and let $\mu'_1(t), \mu'_2(t), \dots, \mu'_{2MN}(t)$ be the eigenvalues of $\Phi_t(\mathbf{x}_0)$. The Lyapunov exponents of \mathbf{x}_0 can be defined as

$$\lambda_i := \lim_{t \rightarrow \infty} \frac{1}{t} \ln |\mu'_i(t)|, \quad i = 1, 2, \dots, 2MN \quad (23)$$

Here, we focus on the single (uncoupled) FitzHugh-Nagumo model in equation (6) and evaluate the Lyapunov exponents in order to investigate how the trajectories starting in an infinitesimal neighbourhood of the initial condition \mathbf{x}_0 diverge from each other. For the sake of simplicity, we denote its initial condition by (v_0, w_0) instead of $(v(0), w(0))$. For $\varepsilon = 0.001$, we vary the value of parameter a from 0.1 to 0.7 with step-size $\Delta a = 0.1$ to get different two dimensional FitzHugh-Nagumo systems. For each system, we evaluate the Lyapunov exponents $\lambda = (\lambda_1, \lambda_2)$ for the initial condition $w_0 = 0$, and $v_0 \in [0, 1]$ with step-size 0.001. The results are provided in Fig. 6. Detailed numerical methods for computing Lyapunov exponents λ can be found in [22], [23] and a brief introduction is provided in Appendix A.



(a) Curve of system parameter a versus image threshold θ (b) Curve of $a - k_1\theta$ versus image threshold θ

Fig. 7: The quasi-linear relationship between system parameter a and image threshold θ .

Due to the global stable node at the origin, all the solutions starting from \mathbf{x}_0 will finally converge with the others starting from $\mathbf{x}_0 + \delta \mathbf{x}_0$. So, it can be seen that all the Lyapunov exponents $\lambda(v_0)$ are negative. However, there is a distinct cusp in each curve of $\lambda_1(v_0)$ in Fig. 6a and a steep drop in each curve of $\lambda_2(v_0)$, indicating that from the vicinity of specific initial values, the solution trajectories maximally diverge before they finally approach the stable node as shown in Fig. 2a. 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 Fig. 6c, where

$$\hat{\lambda}_i(v_0) = \frac{\lambda_i(v_0)}{\max_{v'_0}(|\lambda_i(v'_0)|)}, \quad i = 1, 2 \quad (24)$$

The drop in each curve of $\lambda_2(v_0)$ corresponds to the cusp in $\frac{d\hat{\lambda}_2}{dv_0}$, and their locations coincide with those of the cusps in $\hat{\lambda}_1$ as shown in Fig. 6c. This identifies the specific the initial conditions for which the trajectories are most widely separated for a single neuron as shown in Fig. 2a. So, for each system with different settings of parameter a , we define the the image threshold θ to be the v_0 value at which the Lyapunov exponent λ_1 reaches its peak.

Fig. 7a 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

$$a = k_1 \cdot \theta + k_2, \quad k_1 \approx 1.02, \quad k_2 \approx -0.01. \quad (25)$$

Correspondingly, we will rescale the original image intensity U to the interval $U^r \in (0.1 \ 0.3)$ thus:

$$U^r = 0.1 + 0.2 \cdot \frac{U}{L}, \quad (26)$$

where L is the range of image intensities ($L = 255$ for an 8-bit image). It can be seen from Fig. 7b that the real image threshold θ is larger than the parameter a . In Nomura's algorithm, θ is the diffused version of processed image. So, if simply assume $a_i = \theta_i$ in the whole network, the input image U^r is required to be sufficiently diffused in order to detect as

more edges as possible. Naturally, the wrong edges will be more likely to emerge as illustrated in Section III.

B. Model Simplification

Consider the network in equation (9) consisting of only two coupled neurons, namely $\mathbf{x} = (v_1, w_1, v_2, w_2)^T \in \mathbb{R}^4$. Due to symmetry, we rewrite the model as

$$\begin{cases} \varepsilon \dot{v}_1 = v_1(1 - v_1)(v_1 - a) - w_1 + \varepsilon k_v(v_2 - v_1) \\ \dot{w}_1 = v_1 - bw_1 + k_w(w_2 - w_1), \end{cases} \quad (27)$$

with similar equations for v_2 and w_2 . Since we treat the ratio ε of timescales for v, w to be very small, $\varepsilon \ll 1$, the v coupling $k_v(v_2 - v_1)$ may contribute little to the whole system compared to the w coupling $k_w(w_2 - w_1)$ and may thereby be omitted from the model. In order to support this assumption, we illustrate the influence of the coupling strength k_v by simulating the dynamics of such a network consisting of two coupled neurons. We focus on the specific parameter settings ($a = 0.25$ $b = 4$), and investigate both the uncoupled ($k_v = 0$ $k_w = 0$) and the coupled case ($k_v \neq 0$ $k_w \neq 0$) for two different values of ε : $\varepsilon = 0.001$ and $\varepsilon = 0.0005$. The initial condition adopted is $v_1 = 0.26$, $v_2 = 0.24$ and $w_1 = w_2 = 0$. So in this case, the expected edge detection result will be $v_1(t \rightarrow \infty) \approx 1$ and $v_2(t \rightarrow \infty) \approx 0$.

The four figures in the first row of Fig. 8 present the simulation result for $\varepsilon = 0.001$. Fig. 8b and 8d show that clear edge detection 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. When ε further decrease to 0.0005, as shown in the second row of Fig. 8, the effect of changing the v coupling on the edge obtained is very little, as shown in Fig. 8f and 8h, so that it is reasonable to neglect it.

The simulation results 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 provides the justification for removing the coupling term through v in (2) to obtain a simplified model

$$\begin{cases} \dot{v}_i = f(v_i, w_i, a_i) \\ \dot{w}_i = g(v_i, w_i) + k_w \sum_{j \in \mathcal{P}_i} (w_j - w_i), \end{cases} \quad (28)$$

where the neurone are coupled only via k_w . Again, $\mathcal{P}_i = f_c(\mathcal{P}_{(m,n)})$ 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.

C. Anisotropic Diffusion

In section III, the problem of generating wrong pulses with the variable image threshold [12], [13] 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.*

[10], [11], [12] 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 locations. So, the wrong pulses will turn up as shown in Fig. 5. To solve this problem, a new model was introduced [11], [12] (18) containing two sub-systems per pixel in equation (2), 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 lyapunov exponents 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 image threshold θ as in equation (25). Secondly, 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. And moreover, inspired by the work of Perona and Malik [24], where diffusion is anisotropically controlled by the magnitude of the gradient, we propose a model of anisotropic diffusion by modifying the diffusion constant in the original discrete equation (17) as follows:

$$\begin{aligned} \dot{\theta}_i &= d_i \sum_{j \in \mathcal{P}_i} (\theta_j - \theta_i) \\ d_i &= \tilde{d} \cdot \mathcal{H}\left(\frac{\|\nabla U^r_i\|}{\max_i(\|\nabla U^r_i\|)} - \eta\right), \end{aligned} \quad (29)$$

where \tilde{d} is a small constant, $\mathcal{H}(\cdot)$ is the Heaviside step function and η is a constant threshold. $\|\nabla U^r_i\|$ is the magnitude of the gradient for U^r_i which can be discretized as

$$\begin{aligned} \|\nabla U^r_i\| &= \left\| \left(\frac{\partial U^r_i}{\partial x}, \frac{\partial U^r_i}{\partial y} \right) \right\| = \\ &= \frac{\sqrt{(U^r_{(m+1,n)} - U^r_{(m-1,n)})^2 + (U^r_{(m,n+1)} - U^r_{(m,n-1)})^2}}{2\Delta h}, \end{aligned} \quad (30)$$

with the same boundary condition as in equation (4).

As a result, d_i is no longer constant, but takes on 2 values $\{0, \tilde{d}\}$. If the length of the normalized gradient $\frac{\|\nabla U^r_i\|}{\max_i(\|\nabla U^r_i\|)}$ is greater than the threshold η , d_i is chosen as \tilde{d} . Otherwise for smaller 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 solving the problem of noise sensitivity introduced by adopting an accurate threshold. For noise-free real images, η is taken to be in the range $[0.01, 0.1]$; here, $\eta = 0$ for artificial images and 0.05 for real photographs.

Finally, the novel edge detection algorithm proposed in this paper for the 8-bit grayscale images can be summarised as

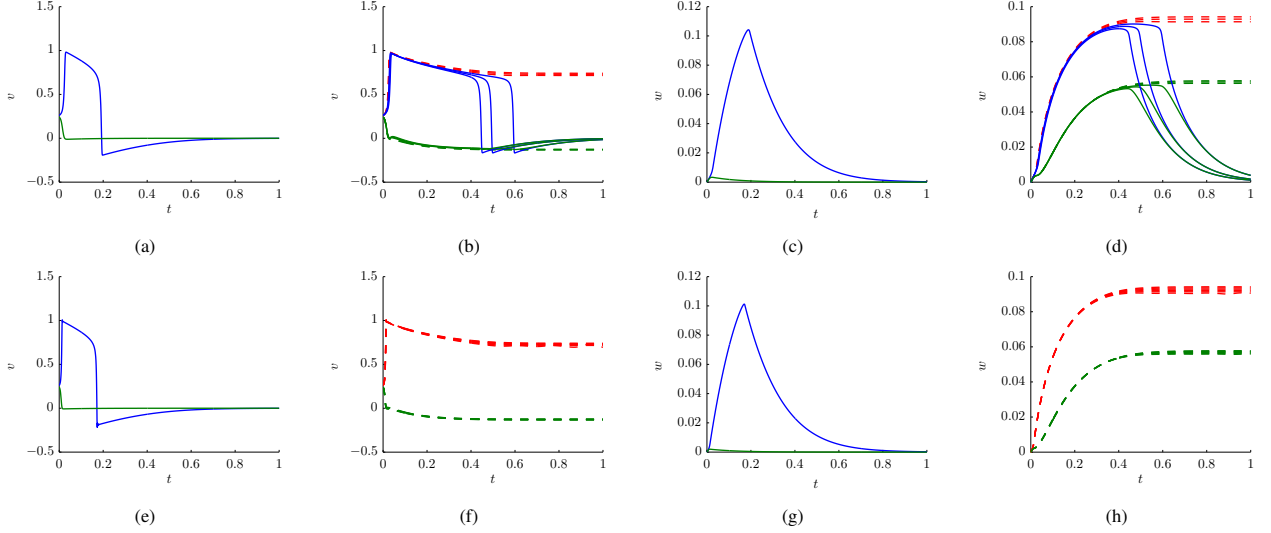


Fig. 8: The first row provide the curves of $v(t)$ and $w(t)$ respectively without coupling and with coupling for $\varepsilon = 0.001$ and in the second row $\varepsilon = 0.0005$. (a), (c), (e) and (g) are the simulation of the uncoupled models, while (b), (d), (f) and (h) are that of the coupled models. 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, $a = 0.25$, $b = 4$.

in Algorithm 2, for ease of comparison to the Nomura *et al* algorithm.

V. RESULTS

In this section, we evaluate the performance of our novel algorithm with that of Nomura *et al* [12] on artificial and natural images.

A. Results on Artificial Images

The 404×303 artificial images used to execute the edge-detection tasks to evaluate the performance of our novel method compared to that in [12] are shown in Fig. 9a. These images are composed of twelve small images divided in two groups, circles and squares. Each image has only three intensity levels, 0, 127 and 255 so that either group of circles or squares has all permutations of these three colors. 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, but of the same size as, the artificial images of Fig. 9a as shown in Fig. 9c and Fig. 9d.

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 Fig. 9b. Both the artificial image and its ground-truth data are graphed by the programs in MATLAB. Specifically, in order to quantify the performance of edge detection of algorithms for artificial images, we will adopt the following four measures: the number of true positives tp , the percentage of true positives tp_r , the false negatives fn and the percentage of false negatives fn_r .

We present the accuracy of edges detected using Nomura *et al*'s algorithm in Fig. 10a, 10b and 10c, which are to be contrasted with those obtained by our method in Fig. 10d, 10e and 10f. Table I shows the comparison with the ground-truth edge information in Fig. 9b. Because the algorithm using the network 2 chooses the higher of the two intensity levels that change appreciably across an edge as the location of the edge,

Algorithm 2 Edge detection algorithm with coupled FitzHugh-Nagumo Model using anisotropically diffusive image threshold

- 1: Rescale the image intensity distribution $U^r_{m,n} = \xi_1 U_{(m,n)} + \xi_2$ with $\xi_1 = 1/1275$ and $\xi_2 = 0.1$ so that $0.1 \leq U^r_{(m,n)} \leq 0.3$.
- 2: Solve the equation below get an anisotropically diffused version of rescaled image $\theta_i(\tau)$ with a stopping time τ ,

$$\begin{aligned} \dot{\theta}_i &= d_i \sum_{j \in \mathcal{P}_i} (\theta_j - \theta_i) \\ d_i &= \tilde{d} \cdot \mathcal{H} \left(\frac{\|\nabla U^r_i\|}{\max_i (\|\nabla U^r_i\|)} - \eta \right) \\ \|\nabla U^r_i\| &= \left\| \left(\frac{\partial U^r_i}{\partial x}, \frac{\partial U^r_i}{\partial y} \right) \right\| \end{aligned} \quad (31)$$

- 3: Evaluate the system parameter $a = k_1 \theta + k_2$, where $k_1 \approx 1.02$ and $k_2 \approx -0.01$.
- 4: Solve the new model equation below,

$$\begin{cases} \dot{v}_i = f(v_i, w_i, a_i) \\ \dot{w}_i = g(v_i, w_i) + k_w \sum_{j \in \mathcal{P}_i} (w_j - w_i) \end{cases} \quad (32)$$

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 $v_i(\tau_s)$.

- 5: Threshold $v_{(m,n)}(\tau_s)$ to get the final binary edge map $\mathcal{M}_{(m,n)}$.

$$\widetilde{\mathcal{M}}(m, n) = \begin{cases} 0, & v_i(\tau_s) > 0.5 \\ 1, & v_i(\tau_s) \leq 0.5 \end{cases} \quad (33)$$

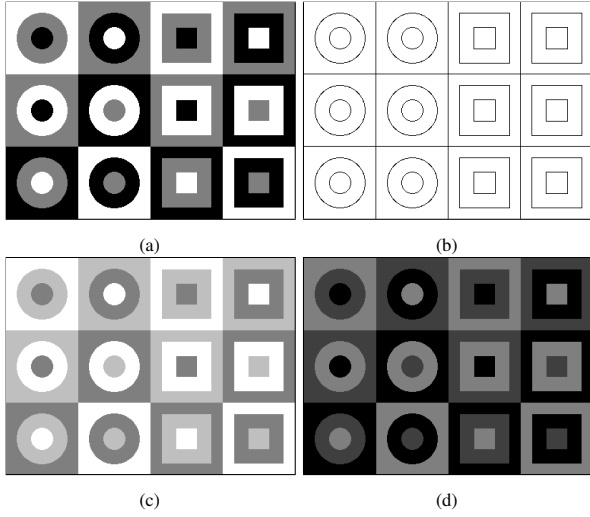


Fig. 9: Artificial images with different combinations of intensity levels (a) (0, 127, 255), (c) (127, 191, 255), (d) (0, 63, 127) and (b) the ground truth of the edge data. (c) and (d) are respectively set relatively lighter and darker in order to test the algorithm in different background intensity levels.

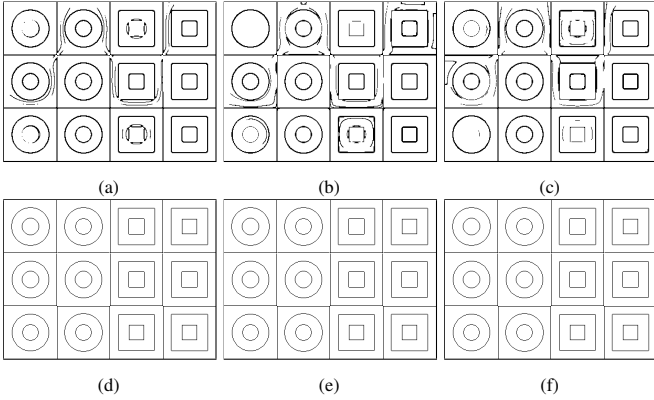


Fig. 10: Edge detection results for artificial images. (a), (b) and (c) are respectively obtained from Fig. 9a, 9d and 9c by Nomura's method. While (d), (e) and (f) are respectively obtained from Fig. 9a, 9d and 9c by the proposed method. For Nomura's method, the system parameters are set as $b = 1$, $\varepsilon = 0.001$, $k_v = 4$ and $k_w = 20$, and the two diffusive constants $d^0 = 40$ and $d^1 = 200$ in equation (17). Both the diffusion stop times are $\tau = 1.0$. For the proposed method, the system parameters are set as $b = 3.5$, $\varepsilon = 0.001$ and $k_w = 5$. The constant $\tilde{d} = 10$ and the threshold $\eta = 0.0$ in equation (29). The steady state time $\tau_s = 1.0$ for both algorithm.

that is what we identify. Also, following Nomura et al. [12], we also allow for an error of one pixel shift when calculating the true positives and the false negatives.

It is clear that the results on edges detected by our algorithm improve upon those detected using [12]. Specifically, the rate of true positive tp_r is increased by nearly 17% and the rate of false negative is decreased by nearly 44%. It is also worth noting that the true positives obtained with the Nomura algorithm number nearly twice as those from our algorithm. The reason is that the Nomura algorithm merges the two edge maps obtained from the original and the inverted images; we allowed one pixel error when calculating these measures, so that most of the true positive pixels in one edge map would be the neighbours of those from the map from the inverted image. As a result, the edges obtained with the Nomura method will normally be as twice thick as the ones with our algorithm. Since it is generally accepted [18] that thinner edges are better

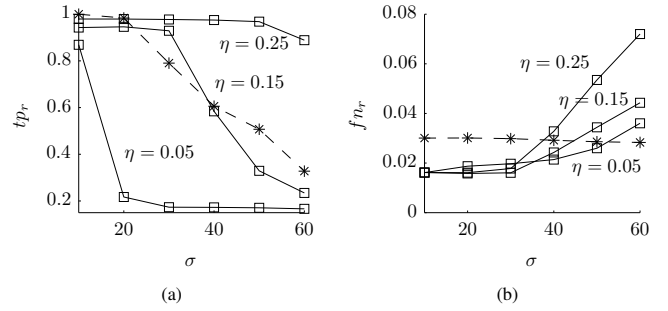


Fig. 11: Robustness test in comparison with canny detectors. (a) true positive rate, (b) false negative rate. The solid lines with square markers are the result obtained by the proposed method and the dashed lines with asterisk markers are the result by Canny algorithm.

than the thick ones this is another indicator of how our method improves upon [12].

We would also like to make a remark concerning the apparent lack of symmetry in the Figures 10a, 10b and Fig. 10c. Although each square of the artificial image is symmetrical, we test the algorithms by applying the whole image $[404 \times 303]$ 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 Fig. 10a, Fig. 10b and Fig. 10c.

B. Robustness Test

In order to illustrate the usefulness of our novel algorithms, 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 [18], an algorithm commonly accepted as a benchmark in the image processing community, in terms of true positive rate and false negative rate as the standard deviation of the noise in the images is increased.

The intensity levels of the artificial image are 63, 127 and 191, to which we add white noise with a standard deviation σ in the range 10 to 60. For each noisy image, the value of threshold η in equation (29) is respectively set at three different values – 0.05, 0.15 and 0.25. All the measures are obtained by averaging over 100 simulations and the results are provided in Fig. 11.

It can be seen from Fig. 11 that, for the Canny detector, tp_r decreases with increasing σ , while fn_r remains constant at 0.03. For our method, the average value of each tp_r increases when η becomes larger. However, although the fn_r curves start at relatively lower values compared to those of the Canny detector when $\sigma = 10$, they increase rapidly upon increasing σ . Specifically at $\sigma = 60$, the larger η is chosen, the larger fn_r becomes. According to these results, we can state that the performance of the proposed algorithm is better than the original method and comparable with that of the Canny algorithm.

C. Results on Real Images

Furthermore, we test the proposed algorithms for real grayscale images, in comparison with Nomura's method and Canny detector. In Fig. 12 we present the results respectively

TABLE I: Quantitative evaluations of edge detection algorithms

	Fig. 9a		Fig. 9c		Fig. 9d	
	Algorithm#1	Algorithm#2	Algorithm#1	Algorithm#2	Algorithm#1	Algorithm#2
tp (No. of Pixels)	11340	5758	11230	5758	11057	5758
tp_r (%)	85.79%	100.00%	84.37%	100.00%	84.43%	100.00%
fn (No. of Pixels)	174	97	170	97	177	97
fn_r (%)	2.92%	1.63%	2.85%	1.63%	2.97%	1.63%

obtained from [13], our method and Canny detector along a row of the table with the original image as the left-most entry. All the four real images are available on the website [25] by Heath *et al.* and were also used in the previous work [12], [13].

We find that our algorithm is able to pick up more details than the original method as shown in Fig. 12o compared with 12n and these details can also be clearly found in the result given by Canny detector as shown in Fig. 12p. We attribute this success to the application of a more accurate relation between the threshold θ and the system parameter a as in equation (25). However, this increased accuracy makes it relatively vulnerable to background noise in real images, especially illustrated by the detection results for the ground in Fig. 12a and 12e. Here, we choose the threshold η for anisotropic diffusion uniformly as 0.05 for real images and its value should be appropriately chosen for each real image. The Canny's algorithm shows the best balance between the accuracy and the noise immunity among the three methods. In this paper, we aim to present the basic edge detection network (2) which has the comparable edge detection ability (the balance between the noise immunity and the accuracy) to Nomura's method [13] without doubling the network size. Since the main motivation of this paper is to explore how DST may be used to tailor the design phase of a silicon circuit, whose task is chosen to be edge detection as an exemplar, this halving of network size compared with [13] would be of great benefit to that goal.

VI. DISCUSSION

In this paper we have set out to refine in performance and reduce in size, a reaction-diffusion system that models a physical process for edge detection [13] 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 paper. 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 of Nomura *et al.* 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 [26], [27], [28]), 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 and hardware analog computation. The very core of both the original

edge detection algorithm [13] and our improved one is the dynamical system described in equation (2), of which the structure is clearly shown in Fig. 1. Building such a network using the technology of integrated circuit belongs to the hardware design procedure. The design of the single element in equation (6) is covered by the extensively researched topic of "silicon neurons" [4], [29] and the coupling structure is also thoroughly studied by the "Cellular Neural Network" [30], [15] community. We provide a simplified version of the circuit in equation (28) by eliminating the couplings of the membrane potentials from the adjacent neurons. 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 [10] or a variable one [11], [12], [13] embedded as the system parameters.

For evaluating the image threshold θ , the previous work [12], [13] does not require much digital computation after the hardware design phase, as they adopt the equation $a = \theta$ (which we find to be inadequate) and subject a to isotropic diffusion. This diffusion equation needs to be solved, and the rescaling of the intensities of the original and inverted image needs to be evaluated – these are the computational tasks. In addition to the rescaling of the original image, our method requires another four steps in order to obtain the θ from the original image U^r and also a linear mapping from θ to a . In particular, we need to set the constant threshold η within the correct range, evaluate the magnitude of the gradient $\|\nabla U^r\|$, the diffusion coefficient d , and finally, solve the anisotropic diffusion (29). All these computations belong to the off-circuit design, which will be completed by the software program before being applied to the circuit system.

APPENDIX A

COMPUTATION OF LYAPUNOV EXPONENTS

Consider the single FitzHugh-Nagumo system as follows,

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^2 \quad (34)$$

The direct way of computing all the Lyapunov exponents $\lambda_i(\mathbf{x}_0)$ of this system is to follow the definition in equation (23). In order to evaluate the derivative of the trajectory $\Phi_t(\mathbf{x}_0)$ with respect to the initial condition \mathbf{x}_0 , we need to integrate the Jacobian matrix $\mathbf{DF}(\mathbf{x})$ along the solution $\mathbf{x}(t; \mathbf{x}_0)$ to the system (34) with the initial condition \mathbf{x}_0 . In order words, to calculate $\Phi_t(\mathbf{x}_0)$, we need to solve the combined system as follows,

$$\begin{Bmatrix} \dot{\mathbf{x}} \\ \dot{\Phi} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}(\mathbf{x}) \\ \mathbf{DF}(\mathbf{x}) \cdot \Phi \end{Bmatrix}, \quad \begin{Bmatrix} \mathbf{x}(t_0) \\ \Phi(t_0) \end{Bmatrix} = \begin{Bmatrix} \mathbf{x}_0 \\ I \end{Bmatrix} \quad (35)$$

where the first row is the original system (34) and the second row is the linearized equation called *variational equation*.

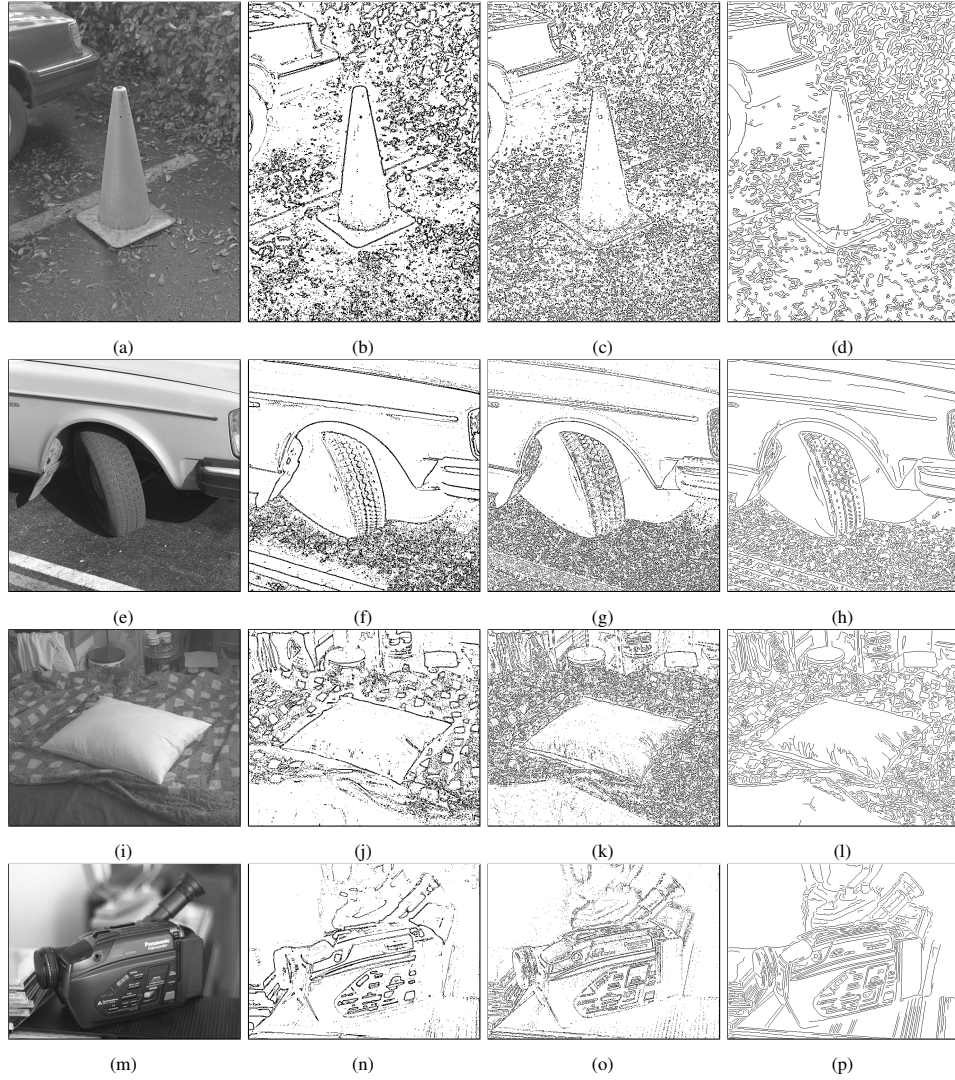


Fig. 12: Edge detection results for real images, (a) Traffic Cone [437 × 604]. (e) Tire [512 × 512] (i) Pillow [552 × 468]. (m) Videocamera [577 × 435] [25]. (b), (f), (j) and (n) provide the edge detection results obtained with the Nomura method [13], in which the system parameters are $b = 1$, $\varepsilon = 0.001$, $k_v = 4$ and $k_w = 20$, and the two diffusive constants in equation (17) 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 (18). (c), (g), (k) and (o) provide the edge detection results with the proposed method using anisotropic diffusion, in which the system parameters are $b = 3.5$, $\varepsilon = 0.001$, $k_v = 0$ and $k_w = 5$. The constant $\tilde{d} = 10$ and the threshold $\eta = 0.05$ in equation (29) 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 equation (28). (d), (h), (l) and (p) are the results obtained by the Canny's algorithm where the thresholds are automatically determined by the inclined function in MATLAB. All the real images are provided by Heath *et al* which are available on the website "Edge Detector Comparison"

The variational equation can be obtained by differentiating both the sides of equation (34) with respect to \mathbf{x}_0 [22]. However, this direct approach is problematic. For large value of t [22], [23], the columns of $\Phi_t(\mathbf{x})$ will asymptotically align with the eigenvector corresponding to the largest eigenvalue of $\mathbf{DF}(\mathbf{x})$. So, both the matrix $\Phi_t(\mathbf{x})$ and its eigenvalues $\mu'_i(t)$ will be ill-conditioned especially when ε in equation (6) is very small. Indeed, if applying the direct approach here, one of the eigenvalues μ'_1 of $\Phi_t(\mathbf{x}_0)$ will fast shrink to zero.

So, instead of the definition in equation (23), we are looking at the definition of Lyapunov exponents of order n [23],

$$\lambda^n(\mathbf{x}_0) = \lim_{t \rightarrow \infty} \frac{1}{t} \ln[\text{Vol}^n(\Phi_t(\mathbf{x}_0))] \quad (36)$$

where Vol^n denotes the n -dimensional volume and it is shown in [31] that

$$\lambda^n = \sum_{i=1}^n \lambda_i \quad (37)$$

So, an alternative approach of computing all the λ_i can be introduced. Assume u_1, \dots, u_n are n column vectors of the matrix $\Phi_t(\mathbf{x}_0)$. Integrate the combined system (35) for a relatively small period ΔT from $\Phi_0 = (u_1^{(0)}, \dots, u_n^{(0)}) = I$ to obtain $\Phi_{\Delta T} = (u_1^{(1)}, \dots, u_n^{(1)})$. Get the orthonormal $\hat{\Phi}_{\Delta T} = (\hat{u}_1^{(1)}, \dots, \hat{u}_n^{(1)})$ by using the Gram-Schmidt method as follows,

$$\begin{aligned} p_1^{(1)} &= u_1^{(1)}, & \hat{u}_1^{(1)} &= p_1^{(1)} / \|p_1^{(1)}\| \\ p_2^{(1)} &= u_2^{(1)} - \langle u_2^{(1)}, \hat{u}_1^{(1)} \rangle \hat{u}_1^{(1)}, & \hat{u}_2^{(1)} &= p_2^{(1)} / \|p_2^{(1)}\| \\ &\dots & & \\ p_n^{(1)} &= u_n^{(1)} - \sum_{i=1}^{n-1} \langle u_n^{(1)}, \hat{u}_i^{(1)} \rangle \hat{u}_i^{(1)}, & \hat{u}_n^{(1)} &= p_n^{(1)} / \|p_n^{(1)}\| \end{aligned} \quad (38)$$

The n -dimensional volume $\text{Vol}^n(\Phi_{\Delta T}(\mathbf{x}_0))$ is

$$\text{Vol}^n(\Phi_{\Delta T}) \approx \|p_1^{(1)}\| \cdots \|p_n^{(1)}\| \quad (39)$$

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

$$\frac{\text{Vol}^n(\Phi_{k\Delta T})}{\text{Vol}^n(\Phi_{(k-1)\Delta T})} \approx \|p_1^{(k)}\| \cdots \|p_n^{(k)}\| \quad (40)$$

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

$$\lambda^n(\mathbf{x}_0) = \lim_{K \rightarrow \infty} \frac{1}{K\Delta T} \sum_{k=1}^K \ln(\|p_1^{(k)}\| \cdots \|p_n^{(k)}\|) \quad (41)$$

And according to the relation provided in equation (37), we can obtain each Lyapunov exponent λ_i as

$$\lambda_i(\mathbf{x}_0) = \frac{1}{K\Delta T} \sum_{k=1}^K \ln(\|p_i^{(k)}\|). \quad (42)$$

As mentioned in [22], either too small or too large a value of ΔT will lead to numerical inaccuracies in computing λ_i and it is suggested be ten or twenty times the natural period of the system. In this paper, the equation (35) 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 $\bar{\mathbf{x}}$ such as (6), if μ_1, \dots, μ_n are the eigenvalues of the Jacobian at $\bar{\mathbf{x}}$, we can obtain

$$\lambda_i(\bar{\mathbf{x}}) = \text{Re}(\mu_i), \quad i = 1, \dots, n \quad (43)$$

As lyapunov exponent λ_i is defined as the average rate of the diversion or contraction of the nearby trajectories as $t \rightarrow \infty$. So, theoretically for nearly all the \mathbf{x}_0 [22] lie in the basin $\partial B(\bar{\mathbf{x}})$ of $\bar{\mathbf{x}}$, $\lambda_i(\mathbf{x}_0) = \lambda_i(\bar{\mathbf{x}})$. In other words, any transient can be ignored if K is a large integer. In order to present a distinct cusp in $\lambda(\mathbf{x}_0)$, we choose $K = 200$ in this paper.

REFERENCES

- [1] A. Hodgkin and A. Huxley, "A quantitative description of membrane current and application to condition and excitation in nerve," *The Journal of Physiology*, vol. 117, pp. 500–544, 1952.
- [2] R. FitzHugh, "Impulses and physiological states in theoretical models of nerve membrane," *Biophysical Journal*, vol. 1, pp. 445–466, 1961.
- [3] J. Nagumo, S. Arimoto, and S. Yoshizawa, "An active pulse transmission line simulating nerve axon," *Proceedings of the IRE*, vol. 50, pp. 2061–2070, 1962.
- [4] M. Mahowald and R. Douglas, "A silicon neuron," *Nature*, vol. 354, pp. 515–518, 1991.
- [5] A. Basu and P. Hasler, "Nullcline-based design of a silicon neuron," *IEEE Transactions on Circuits and Systems*, vol. 7, pp. 421–432, 2010.
- [6] J. Cosp, S. Binczak, J. Madrenas, and D. Fernández, "Implementation of compact vlsi fitzhugh-nagumo neurons," in *In Proceeding of ISCAS*, 2008, 2008, pp. 2370–2373.
- [7] J. Wijekoon and P. Dudek, "Compact silicon neuron circuit with spiking and bursting behaviour," *Journal of Neural Networks*, vol. 21, pp. 524–534, 2008.
- [8] J. Arthur and K. Boahen, "Silicon-neuron design: a dynamical system approach," *IEEE Transactions on circuits and systems*, vol. 58, pp. 1034–1043, 2011.
- [9] K. Zaghloul and K. Boahen, "A silicon retina that reproduces signals in the optic nerve," *Journal of Neural Engineering*, vol. 3, pp. 257–267, 2006.
- [10] A. Nomura, M. Ichikawa, H. Miike, M. Ebihara, H. Mahara, and T. Sakurai, "Realizing visual functions with the reaction-diffusion mechanism," *Journal of the Physical Society of Japan*, vol. 72, pp. 2385–2395, 2003.
- [11] A. Nomura, M. Ichikawa, R. H. Sianipar, and H. Miike, "Edge detection with reaction-diffusion equations having a local average threshold," *Pattern Recognition and Image Analysis*, vol. 18, pp. 289–299, 2008.
- [12] A. Nomura, M. Ichikawa, K. Okada, H. Miike, and T. Sakurai, "Edge detection algorithm inspired by pattern formation processes of reaction-diffusion systems," *International Journal of Circuits, Systems and Signal Processing*, vol. 5, pp. 105–115, 2011.
- [13] A. Nomura, M. Ichikawa, K. Okada, M. H. T. Sakurai, and Y. Mizukami, "Image edge detection with discrete spaced fitzhugh-nagumo type excitable elements," in *Image Processing*, 2011, pp. 1–8.
- [14] N. Kurata, H. Kitahata, H. Mahara, A. Nomura, H. Miike, and T. Sakurai, "Stationary pattern formation in a discrete excitable system with strong inhibitory coupling," *Physical Review E*, vol. 18, pp. 289–299, 2008.
- [15] K. Karahaliloğlu and S. Balkır, "An cmos cell circuit for compact implementation of reaction-diffusion models," in *In Proceedings of Neural Networks. 2004 IEEE International Joint Conference*, 2004, pp. 2831–2836.
- [16] F. C. Hoppensteadt and E. Izhikevich, *Weakly Connected Neural Networks*. Springer, 1997.
- [17] M. Ebihara, H. Mahara, T. Sakurai, A. Nomura, A. Osa, and H. Miike, "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*, vol. 32, pp. 378–385, 2003.
- [18] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698, 1986.
- [19] J. Verschelde, "Phcpack: A general-purpose solver for polynomial systems by homotopy continuation," *ACM Transactions on Mathematics Software*, vol. 25, pp. 251–276, 1999.
- [20] M. Black, G. Sapiro, D. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Transactions on Image Processing*, vol. 7, pp. 421–432, 1998.
- [21] J. Palis and W. de Melo, *Geometric Theory of Dynamical Systems: An Introduction*. Springer-Verlag, 1982.
- [22] T. Parker and L. Chua, *Practical Numerical Algorithms for Chaotic Systems*. Springer-Verlag, 1989.
- [23] M. Sandri, "Numerical calculation of lyapunov exponents," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 78–84, 1996.
- [24] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 629–639, 1990.
- [25] M. Heath, S. Saida, A. Osa, and K. Munechika, "A robust visual method for assessing the relative performance of edge-detection algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 1338–1359, 1997.
- [26] I. Aizenberg, N. Aizenberg, and J. Vandewalle, "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, 1998, pp. 301–306.
- [27] K. Suzuki, I. Horiba, and N. Sugie, "Neural edge enhancer for supervised edge enhancement from noisy images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1582–1596, 2003.
- [28] K. Suzuki, I. Horiba, N. Sugie, and M. Nanki, "Extraction of left ventricular contours from left ventriculograms by means of a neural edge detector," *IEEE Transactions on Medical Imaging*, vol. 23, pp. 330–339, 2004.
- [29] N. Mizoguchi, Y. Nagamatsu, K. Aihara, and T. Kohno, "A two-variable silicon neuron circuit based on the izhikevich model," *Artificial Life and Robotics*, vol. 16, pp. 383–388, 2011.
- [30] L. Chua and L. Yang, "Cellular neural networks: Theory and applications," *IEEE Transaction on Circuits and Systems*, vol. 35, pp. 1257–1272, 1988.
- [31] V. Oseledec, "A multiplicative ergodic theorem: Lyapunov characteristic numbers for dynamical system," *Transactions of the Moscow Mathematical Society*, vol. 16, pp. 197–231, 1968.