Collective behavior of large-scale neural networks with GPU acceleration

Jingyi Qu¹ · Rubin Wang²

Received: 7 December 2016 / Revised: 8 June 2017 / Accepted: 16 June 2017 / Published online: 30 June 2017
© Springer Science+Business Media B.V. 2017

Abstract In this paper, the collective behaviors of a small-world neuronal network motivated by the anatomy of a mammalian cortex based on both Izhikevich model and Rulkov model are studied. The Izhikevich model can not only reproduce the rich behaviors of biological neurons but also has only two equations and one nonlinear term. Rulkov model is in the form of difference equations that generate a sequence of membrane potential samples in discrete moments of time to improve computational efficiency. These two models are suitable for the construction of large scale neural networks. By varying some key parameters, such as the connection probability and the number of nearest neighbor of each node, the coupled neurons will exhibit types of temporal and spatial characteristics. It is demonstrated that the implementation of GPU can achieve more and more acceleration than CPU with the increasing of neuron number and iterations. These two small-world network models and GPU acceleration give us a new opportunity to reproduce the real biological network containing a large number of neurons.

Keywords Large-scale neural network · Small-world · GPU acceleration · Spatio-temporal characteristics

Introduction

To study the functionality of a brain system, modeling of network consisting of thousands of neurons is required. In any study of network dynamics, there are three crucial issues which are: (1) what model describes spiking dynamics of each neuron; (2) how the neurons are connected; (3) how to accelerate the computation is another important issue especially for a large-scale network (Haken 2002; Nomura et al. 2003).

Firstly, as we develop such a brain model consisting of spiking neurons, the model for a single neuron must be computationally simple and capable of producing rich firing patterns exhibited by real biological neurons. The widely used models of spiking and bursting neurons can be expressed in the form of ordinary differential equations such as Hodgkin–Huxley model (Hodgkin and Huxley 1952), integrate-and-fire (Stein 1967), FitzHugh–Nagumo model (FitzHugh 1961), Hindmarsh–Rose model (Rose and Hindmarsh 1989), Morris–Lecar model (Morris and Lecar 1981), spiking model by Izhikevich (2003, 2004) and so on. Hodgkin–Huxley model is one of the most realistic approaches to simulate neuronal behaviors, which parameters are biophysically meaningful and can produce rich firing patterns (Hodgkin and Huxley 1952). However, only a handful of neurons can be simulated in real time since it is computationally prohibitive with four differential equations and tens of parameters. In contrast, using integrate-and-fire model is computationally effective, but the model is too simple to produce rich spiking and bursting firings exhibited by cortical neurons (Stein 1967). So, Izhikevich et al. proposed a simple spiking model that is as biologically plausible as the Hodgkin–Huxley model and as computationally efficient as integrate-and-fire model (Izhikevich 2003, 2004). Depending on only two

✉ Jingyi Qu
jyqu@cauc.edu.cn

¹ Tianjin Key Laboratory for Advanced Signal Processing, Civil Aviation University of China, Tianjin 300300, China

² Institute for Cognitive Neurodynamics, School of Science, East China University of Science and Technology, Shanghai 200237, China

differential equations and four parameters, Izhikevich model can reproduce spiking and bursting behavior of known types of cortical neurons.

However, all models above are based on ordinary differential equation (ODE). To describe the dynamics of firing patterns generated by a neuron, such models should contain variables of multiple time scales ranging from less than one millisecond to hundreds of milliseconds to describe slow intrinsic and synaptic processes. Such a diversity of time scales limits the speed of the numerical simulation. So, it is a good way to improve computational efficiency by using a dynamical system written in the form of difference equations that generate a sequence of membrane potential samples in discrete moments of time. Attractive features of this Rulkov model include the simplicity of the equation, the ability to describe a broad range of firing patterns found in biological neurons (Rulkov 2002; Rulkov et al. 2004; Rulkov and Bazhenov 2008).

Secondly, how the neurons are connected is another important issue to study network dynamics. Most of the existing work assumes that the coupling configuration is completely regular (Qu et al. 2012a; Heagy et al. 1994), while a few studies address the issue of synchronization in randomly networks (Gade 1996; Manrubia and Mikhailov 1999). However, many biological networks are neither completely regular nor completely random. To interpolate between these two extremes, Watts and Strogatz introduced the interesting concept of small-world networks (Watts and Strogatz 1998). In this small-world network, most neurons of the brain are coupled locally while less neurons are coupled through sparse long-range connections linking physically distant units. Convincing evidences have been presented to prove that small-world networks provide a powerful and versatile tool, leading us towards understanding the structure and function of the human brain (Yu et al. 2008; Bassett and Bullmore 2006; Volman et al. 2005; Hasegawa 2005). Hence, much attention has been devoted to studying the dynamics of small-world neuronal networks (Gong et al. 2006; Wei and Luo 2007; Perc 2007; Perc and Gosak 2008; Sun et al. 2010; Qu et al. 2014).

Thirdly, for a large-scale neural network, how to accelerate the speed of simulation is also one of the key issues. Nowadays, compared with CPU (Central Processing Unit) computing, GPU (Graphics Processing Unit) calculation has become a powerful solution to accelerate general-purpose scientific and engineering applications, in which serial processing is performed by multiple cores of a CPU while compute-intensive portions are implemented by thousands of cores of a GPU (Mei et al. 2014; Bernabe et al. 2013). Since GPUs consist of thousands of smaller and more efficient cores designed for parallel computing, much more acceleration will be achieved by implementing the code on a GPU platform (Bader et al. 2011; Raghav et al. 2015).

In the present paper, in order to simulate the collective behavior of cortical neural networks, large scale small-world networks based on both Izhikevich and Rulkov models are proposed. It is shown that by tuning connection probability and the number of nearest neighbor in small-world topology, the collective dynamics of neuronal activity of both models will be affected greatly. For a large-scale network, solving higher order nonlinear equations is the most time-consuming calculation, which has been moved from CPU to GPU to achieve great acceleration.

Network models

Izhikevich network models

For each single neuron, a simple Izhikevich model is adopted in this paper, which only consists of two ordinary differential equations (Izhikevich 2003, 2004).

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I, \quad (1)$$

$$\frac{du}{dt} = a(bv - u), \quad (2)$$

with the auxiliary after-spike resetting

$$\text{if } v \geq 30 \text{ mV}, v \leftarrow c, u \leftarrow u + d \quad (3)$$

where v represents the membrane potential of the neuron, u represents a membrane recovery variable and it provides negative feedback to v . Synaptic currents or dc-currents are delivered via the variable I . After the spike reaches its apex (+30 mV), the membrane voltage and the recovery variable are reset according to the (3). There are a, b, c, d four parameters in this simple model. Various setting of the parameters results in various intrinsic firing patterns including excitatory and inhibitory cortical cells in intracellular recordings. Neocortical neurons in the mammalian brain can be classified into several types according to spiking patterns. All excitatory cortical cells are divided into RS (regular spiking), CH (chattering) neurons while FS (fast spiking) is typical behavior of inhibitory neurons. Figure 1 gives three most typical types of neurons to different values of the parameters.

To study the network geometry, a pulse-coupled small-world neuronal network is simulated in this paper. The network is described as follow equation.

$$\frac{dv_i}{dt} = 0.04v_i^2 + 5v_i + 140 - u_i + I_{ext,i} + I_{syn,i}, \quad (4)$$

$$\frac{du_i}{dt} = a_i(b_i v_i - u_i), \quad (5)$$

$$I_{ext,i} = I_0 + D\xi_i, \quad (6)$$

Fig. 1 Typical types of neurons to different values of the parameters for Izhichevich model. RS and CH are typical cortical excitatory neurons. FS is cortical inhibitory neuron. In this simulation, time step is 1 ms and time length is 500 ms. **a** Regular spiking (RS), the

parameters are set as $a = 0.02, b = 0.2, c = -65 \text{ mV}, d = 8$. **b** chattering (CH), the parameters are set as $a = 0.02, b = 0.2, c = -50 \text{ mV}, d = 2$. **c** fast spiking (FS), the parameters are set as $a = 0.1, b = 0.2, c = -65 \text{ mV}, d = 2$

$$I_{\text{syn},i} = \sum_j W_{ij}(v_j - v_i), \quad (7)$$

where $I_{\text{ext},i}$ is the external current to the neuron. Each neuron in the network receives a noisy thalamic input. I_0 is the intensity of the dc current. D is the intensity of the noise. ξ is a random process without time correction and the random variables are identically and uniformly distributed in $[-1, 1]$. $I_{\text{syn},i}$ accounts for the synaptic current received by the neuron. The synaptic connection weights between the neurons are given by W_{ij} . The firing of j th neuron instantaneously changes variable v_i by W_{ij} .

Figure 2 gives the three typical examples of network topologies. The parameter p determines the probability of rewiring a link, whereby $p = 0$ constitutes a regular graph (as Fig. 2a), while $p = 1$ results in a random network (as Fig. 2c). For $0 < p < 1$, as exemplified in Fig. 2b, the resulting network may have small-world properties in that the normalized characteristic path length between distant units is small comparable with that of a random network, while the normalized clustering coefficient is still large comparable with that of a regular nearest-neighbor graph. In this paper, a small-world network connectivity patterns

is studied. Regular graph and random graph are excluded since it has been investigated in our previous work (Qu et al. 2012a; Qu and Wang 2012b).

For convenience, we construct a neural network consisting of 1000 neurons. Based on the anatomy of a mammalian cortex, the ratio of excitatory to inhibitory neurons is chosen as 4:1. The numbers of excitatory neurons are from 1 to 800 while inhibitory ones are from 801 to 1000. RS cells is used to model all excitatory neurons and FS cells to model all inhibitory neurons. In the model, each excitatory neuron has $(a_i, b_i) = (0.02, 0.2)$. To make different neurons have different dynamics, the excitatory neurons are assigned $(c_i, d_i) = (-65, 8) + (15, -6)r_i^2$, where r_i is a random variable uniformly distributed on the interval $[0, 1]$. So, $r_i = 0$ corresponds to a regular spiking (RS) cell and $r_i = 1$ corresponds to a chattering (CH) cell. We use $r^2 = 1$ to bias the distribution towards RS cells. Each inhibitory cell has $(a_i, b_i) = (0.02, 0.25) + (0.08, -0.05)r_i$ and $(c_i, d_i) = (-65, 2)$. During the simulation, all equations are numerically integrated using the Euler method. The time step is 0.5 ms.

Fig. 2 Examples of considered network topologies. Only 20 vertices are displayed in each panel. **a** Regular ring characterized by $p = 0$ with periodic boundary conditions. Each vertex is connected to its

$k = 4$ nearest neighbors. **b** Realization of WS small-world topology via random rewiring of a certain fraction p of links, $p = 0.1$. **c** Realization of randomly connected network, $p = 1$

Rulkov network models

In order to model the dynamics of large-scale networks, another computationally efficient Rulkov model is also adopted, which can be written in the form

$$x_{n+1} = f_x(x_n, y_n + \beta_n) \quad (8)$$

$$y_{n+1} = y_n - \mu(x_n + 1) + \mu\sigma + \mu\sigma_n \quad (9)$$

where x_n is the fast and y_n is the slow dynamic variables. Slow time evolution of y_n is achieved by using small values of the parameter $0 < \mu \ll 1$. Parameters of α and σ control the dynamics and they are set to mimic the behavior of a particular type of neuron. Terms σ_n and β_n describe external influences applied to the map, including external dc bias current and synaptic inputs.

The nonlinear function $f_x(x_n, y_n + \beta_n)$ is given by

$$f_x(x_n, y_n + \beta_n) = \begin{cases} \frac{\alpha}{1 - x_n} + y_n + \beta_n, & x_n \leq 0 \\ \alpha + y_n + \beta_n, & 0 < x_n < \alpha + y_n + \beta_n \\ -1, & x_n \geq \alpha + y_n + \beta_n \end{cases} \quad (10)$$

where α is a control parameter of the model.

This Rulkov model is also capable of generating various types of spiking and bursting activity. Typical regimes of temporal behavior of the model are shown in Fig. 3

To study the network geometry, a WS small world scale-free network is simulated in this paper. The spatiotemporal evolution of the studied network is governed by the following equations.

$$x_{i,n+1} = f(x_{i,n}, y_{i,n} + \beta_{i,n}) \quad (11)$$

$$y_{i,n+1} = y_{i,n} - \mu(x_{i,n} + 1) + \mu\sigma_i + \mu\sigma_{i,n} \quad (12)$$

$$\beta_{i,n} = g_c \varepsilon_{ji} \beta^e (x_{j,n-\tau} - x_{i,n}) \quad (13)$$

$$\sigma_{i,n} = g_c \varepsilon_{ji} \sigma^e (x_{j,n-\tau} - x_{i,n}), \quad (14)$$

where index i specifies the cell, $i = 1, 2, \dots, N$. x_i is the membrane potential of the i th neuron. σ_i is the parameter that defines the dynamics of uncoupled cell. The coupling between the cells $\beta_{i,n}$ and $\sigma_{i,n}$ is provided by the current flowing from one cell to the other. g_c is the parameter characterizing the strength of the coupling. The coefficients β^e and σ^e can adjust the coupling for the fast and slow processes in the cells, respectively. In the numerical simulations, the values of coefficients are set as $\beta^e = 1, \sigma^e = 1$. Each node corresponds to one neuron, whose dynamics is governed by Rulkov map, as described above. If neuron i is coupled to neuron j , $\varepsilon_{ij} = 1$ and $\varepsilon_{ij} = 0$ otherwise.

Similar with Izhikevich network above, the numbers of excitatory neurons are from 1 to 800 while inhibitory ones are from 801 to 1000. To make different neurons have different dynamics, the excitatory neurons are assigned $(\alpha_i, \sigma_i, \beta_i) = (4.0, 0.01, -3.0) + (0.5, 0.1, -0.25)r_i$, where r_i is a random variable uniformly distributed on the interval $[0, 1]$. So, $r_i = 0$ corresponds to a regular spiking cells and $r_i = 1$ corresponds to a bursting cell. Each inhibitory cell has $(\alpha_i, \beta_i) = (4.0, -3.0)$, only $\sigma_i = 0.1 + 0.01r_i$.

Collective behavior of Izhikevich small-world network

The effects of two important parameters of WS small-world network are investigated, respectively. One parameter is the connection probability p and the other one is the number of nearest neighbor of each node k . Setting various values of p and k will lead to the change of topology of the network.

Firstly, the effect of connection probability p is simulated. As tuning p , the other parameters are set as constants. If two neurons i, j are connected, $W(i, j) = 40$, otherwise, $W(i, j) = 0$. The half of the number of nearest neighbor of

Fig. 3 Typical types of neurons to different values of the parameters for Rulkov model. In this simulation, time step is 0.5 ms and time length is 2000 ms. **a** Regular spiking (RS), the parameters are set as $\alpha = 4.0, \sigma = 0.01, \beta = -3.0, \mu = 0.001$. **b** Chattering (CH), the

parameters are set as $\alpha = 6.0, \sigma = -0.1, \beta = -4, \mu = 0.001$. **c** Fast spiking (FS), the parameters are set as $\alpha = 4.0, \sigma = 0.1, \beta = -3.0, \mu = 0.001$

each node $k = 2$. The external input $I_0 = 0$, the noise strength for excitatory and inhibitory neurons are set as $D_{exc} = 5$ and $D_{inh} = 2$. Figure 4a shows the small-world topology via random rewiring of a small fraction $p = 0.02$ and Fig. 4b gives its corresponding neuron spiking result. It is demonstrated that if $p = 0.02$, all the neurons are in almost synchronous state, neither randomly spiking nor completely synchronization. Figure 4c gives the small-world topology via random rewiring of a bigger fraction $p = 0.2$ and Fig. 4d shows its corresponding neuron spiking result. It is shown that all the neurons are in complete synchronization state in case of $p = 0.2$ and maintains the interval of spiking.

Secondly, the effect of the nearest neighbor number of each node k is discussed. Small-world topologies and corresponding neuron spiking results with different values of k are shown in Fig. 5. Other parameters are set as connection probability p equals 0.2. If two neurons i, j are connected, $W(i, j) = 20$, otherwise, $W(i, j) = 0$. The external input $I_0 = 0$, the noise strength for excitatory and inhibitory neurons are set as $D_{exc} = 5$ and $D_{inh} = 2$, respectively. If $k = 2$, 1000 neurons randomly spike (Fig. 5b). As Fig. 5d, the value of k continuously increases

to eight, all the neurons spike with the same rhythm (time interval is about 130 ms) and reach the complete synchronization. So, as the number of nearest neighbor of each node increases, the network will become more and more synchronous.

Collective behavior of Rulkov small-world network

The corresponding simulation results based on Rulkov small-world network are given in this section. As tuning connection probability p , the other parameters are set as constants. If two neurons i, j are connected, $\varepsilon_{ij} = 1$ and $\varepsilon_{ij} = 0$ otherwise. The coupling strength $g_c = 0.01$. The number of nearest neighbor of each node $k = 2$. Figure 6a, b shows the small-world topology via random rewiring of a small fraction and its corresponding spatiotemporal behaviors. It is demonstrated that if $p = 0.02$, all the neurons are in almost synchronous state, neither randomly spiking nor completely synchronization. Figure 6c, d gives the small-world topology via random rewiring of a bigger fraction $p = 0.2$ and its corresponding neuron spiking result. It is

Fig. 4 Characteristic of a WS small-world network for different connection probability p based on Izhichevich model. The connection probability p are set as **a**, **b** 0.02, **c**, **d** 0.2. If two neurons i, j are connected, $W_{i,j} = 40$, otherwise, $W_{i,j} = 0$. Other system parameters are: $k = 2$, $I_0 = 0$, $D_{exc} = 5$ and $D_{inh} = 2$. The time length is 1000 mS and the neuron number is 1000

Fig. 5 Characteristic of a WS small-world network for the number of nearest neighbor of each node k based on Izhichevich model. The number of nearest neighbor of each node k are set as **a**, **b** 2, **c**, **d** 8. If two neurons i, j are connected, $W(i, j) = 20$, otherwise, $W(i, j) = 0$. Other system parameters are: $p = 0.2$, $I_0 = 0$, $D_{exc} = 5$ and $D_{inh} = 2$. The time length are 1000 mS and the neuron number is 1000

shown that as p increases, the synchronization status of all the neurons increase and maintain the interval of spiking. It is demonstrated that the connection probability will affect the collective behavior of the network.

How the number of nearest neighbor of each node effects the synchronization is further discussed. Small-world topologies and corresponding neuron spiking results with different values of k are shown in Fig. 7. Other parameters are set as constant values. If $k = 2$, 1000 neurons are in almost synchronous state, neither randomly spiking nor completely synchronization (Fig. 7a, b). As Fig. 7b, the value of continuously increases to four, all the neurons spike with the same rhythm (time interval is about 200 ms) and reach the complete synchronization. So, as the number of nearest neighbor of each node increases, the network will become more and more synchronous.

Acceleration using GPU

In GPU computing, the CPU and GPU are treated as host and device, respectively. As shown in Fig. 8, these two processors own private memories named as host and device

memories, respectively. Direct host memory access by the GPU is forbidden, and so is direct device memory access for the CPU. CPU and GPU are connected by PCIE bus. The CPU can copy data to or back from the GPU when the GPU is executing specific arithmetic tasks. Three kinds of parallel processing can be implemented in GPU computing, which are grid, block and thread. A thread block is a batch of threads that can cooperate with each other by synchronizing their execution and efficiently sharing data through a low latency shared memory. The threads from two different blocks cannot cooperate.

The GPU-based implementation flow based on Izhikevich model is proposed as Fig. 9. The initial parameters a , b , c , d matrix are set on CPU firstly. Then, copy these parameters to GPU. On GPU, higher order nonlinear equations are calculated using Euler method, which is really time consuming especially for large neuron numbers. After that, copy the u and v matrix back to CPU in order to plot the spatiotemporal of the network.

In order to demonstrate the effectiveness of the proposed GPU-based implementation, both GPU- and CPU-based algorithms are implemented on two platforms, respectively. One platform is a workstation with Intel Xeon CPU

Fig. 6 Characteristics of a WS small-world network for different connection probability p based on Rulkov model. The connection probability are set as **a, b** $p = 0.02$ **c, d** $p = 0.2$, respectively. Other parameters are set as neuron number $N = 1000$, $k = 2$, $g_c = 0.01$, $\mu = 0.001$

X5690 at 3.47 GHz containing sixteen cores and with nVIDIA Quadro FX 5800 GPU owning 240 cores. The other is a work-station with Intel Xeon CPU E5-1620 at 3.6 GHz containing eight cores and with nVIDIA GeForce GTX TITAN X GPU owning 3072 cores. Its operating system is Microsoft Windows 7. In this section, only Izhikevich network model is calculated. The code is developed using Matlab, which has provided some functions to realize the parallel running on GPU. For example, GPUArray() function is used to move the parameters from CPU to GPU before calculation, and gather() function is adopted to move the simulation results back to CPU (Table 1).

Platform 1: CPU1+GPU1

Figure 10a gives the computational performance with different neuron numbers both on GPU and CPU platform, respectively. It can be seen that the neuron number 6000 is the threshold. If neuron number is smaller than 6000, the performances of CPU and GPU have little difference. CPU is even more quickly than GPU calculation. However, if the neuron number is in the range from 6,000 to 10,000, the time increases greatly from 2.5355 to 76.0402 s for CPU-

based implementation while the time increases only from 5.4338 to 30.2934 s for GPU-based implementation. As 6,000 neurons, GPU can only achieve 2.8922 s acceleration, while GPU achieves about 40 s acceleration for 10,000 neurons. It can be shown that the more neuron numbers, the more effective using GPU-based algorithm.

Figure 10b gives the computational performance with different iteration times both on GPU and CPU platform, respectively. With the increment of iteration time, the required time of both GPU-based and CPU-based algorithms increases linearly. If the iterations are 4000, the time difference between GPU and CPU is 101.5775 s. As the iteration increases to 10,000, the difference increases to 376.5042 s. For 10,000 iterations, CPU takes 502.1779 s while GPU takes 125.6737 s. GPU achieves almost four times performance improvement than CPU. It can also be concluded that the more iterations, the more effective using GPU-based algorithm.

For a more detailed investigation, the graphs of performance are plotted in the (iteration, neuron number)-parameter plane as illustrated in Fig. 11. From the maps, one obvious conclusion is that if iteration times are small and neuron number are not too much, the performance of CPU

Fig. 7 Characteristic of a WS small-world network for the number of nearest neighbor of each node k based on Rulkov model. The connection probability are set as **a**, **b** $k = 2$ **c**, **d** $k = 4$, respectively. Other parameters are set as neuron number $N = 1000$, $p = 0.1$, $g_c = 0.01$, $\mu = 0.001$

Fig. 8 Computing framework of CPU and GPU. Host is CPU part while device is GPU part

and GPU are almost same. However, with the iteration times and neuron number, the GPU achieve higher and higher performance. As 10,000 iterations and 10,000 neurons, GPU implementation is 419.312 s while CPU-based method is 958.1942, which achieves more than two times acceleration.

Platform 2: CPU2+GPU2

Figures 12 and 13 give corresponding simulation results using platform2, which have the similar results as platform1. It can be seen that for 10,000 neurons and 10,000 iterations,

GPU2 achieves 123.0913 s, which are about seven times better than CPU2. Since GPU2 is more powerful than GPU1, the performance has large improvement.

Conclusion

In this paper, the collective dynamics of a large-scale small-world neuronal network based on both Izhichevich and Rulkov models are presented in detail. The coupled neurons will exhibit types of collective behavior by tuning

Fig. 9 GPU-based implementation flow based on Izhikevich model. The higher order nonlinear equations are calculated by GPU

Table 1 Hardware configurations about GPU and CPU

Platform1	
CPU1	GPU1
Intel Xeon X5690	nVIDIA Quadro FX 5800
16 cores	240 cores
Platform2	
CPU2	GPU2
Intel Xeon E5-1620	nVIDIA GeForce GTX TITANX
8 cores	3072 cores

two key parameters. One is connection probability and the other is the number of nearest neighbor. Since it is large scale network, in order to accelerate the calculation speed, the program is implemented on GPU platform.

Four major outcomes are found based on this network model. (1) In addition, it is shown that the collective dynamics of neuronal activity are changed by tuning the connection probability in small-world topology. It is demonstrated that with increasing the connection probability, the network will display the behavior from almost synchronous to complete synchronization and maintains it. (2) Moreover, it is presented that increasing the number of nearest neighbor in small-world topology can also significantly affect the collective dynamics of neuronal activity, from randomly spiking to completely synchronization. (3) One of them is the demonstration that if iteration times are small and neuron numbers are small, the performances of CPU and GPU are almost same, both has good performance. (4) However, the more the iteration is, the more the neuron number is, the GPU can achieve higher and higher performance than CPU. These two small-world network models and GPU acceleration give us a new opportunity to reproduce the real biological network containing a large number of neurons. These methods and results provide some guidelines to understand the collective dynamics of mammalian cortex. Since the functions provided by Matlab have been well encapsulated, it is not easy for us to modify detail parameters for GPU calculation. To overcome this problem, Compute Unified Device Architecture (CUDA) programming model will be adopted in our next step work, which can provide the programmers adequate C language like APIs to better exploit the parallel power of the GPU.

Fig. 10 Computational performance on platform1 (CPU1+GPU1). **a** Plot of performance vs neuron numbers. The iteration time is 1000 mS. **b** Plot of performance vs iteration. The neuron number is 8000

Fig. 11 The plot of the performance in the (iteration, neuron number) parameter based on platform 1. **a** CPU1-based, **b** GPU1-based. The range of neuron number is from 1000 to 10,000. The iteration time is from 1000 to 10,000

Fig. 12 Computational performance on platform2 (CPU2+GPU2). **a** Plot of performance vs neuron numbers. The iteration time is 1000. **b** Plot of performance vs iteration. The neuron number is 8000

Fig. 13 The plot of the performance in the (iteration, neuron number) parameter based on platform2. **a** CPU1-based, **b** GPU1-based. The range of neuron number is from 1000 to 10,000. The iteration time is from 1000 to 10,000

Acknowledgements The work was supported by the National Natural Science Foundation of China (11402294, 11232005, 61502498).

References

- Bader DA, Kaeli DR, Kindratenko V (2011) Guest editor's introduction: special issue on high-performance computing with accelerations. *IEEE Trans Parallel Distrib Syst* 22(1):3–6
- Bassett DS, Bullmore E (2006) Small-world brain networks. *Neuroscientist* 12(6):512–523
- Bernabe S, Lopez S, Plaza A, Sarmiento R (2013) GPU implementation of an automatic target detection and classification algorithm for hyperspectral image analysis. *IEEE Geosci Remote Sens Lett* 10(2):221–225
- FitzHugh R (1961) Impulses and physiological states in theoretical models of nerve membrane. *Biophys J* 1(6):445–466
- Gade PM (1996) Synchronization of oscillators with random nonlocal connectivity. *Phys Rev E* 54:64–70
- Gong YB, Xu B, Xu Q, Yang CL, Ren TQ, Hou ZH, Xin HW (2006) Ordering spatiotemporal chaos in complex thermo sensitive neuron networks. *Phys Rev E* 73:046137
- Haken H (2002) Brain dynamics synchronization and activity patterns in pulse-coupled neural nets with delays and noise. Springer, Berlin
- Hasegawa H (2005) Synchronizations in small-world networks of spiking neurons: diffusive versus sigmoid couplings. *Phys Rev E* 72:056139
- Heagy JF, Carroll TL, Pecora LM (1994) Synchronous chaos in coupled oscillator systems. *Phys Rev E* 50(3):1874–1885
- Hodgkin AL, Huxley AF (1952) A qualitative description of membrane current and its application to conduction and excitation in nerve. *J Physiol* 117:500–544
- Izhikevich EM (2003) Simple model of spiking neurons. *IEEE Trans Neural Netw* 14(6):1569–1572
- Izhikevich EM (2004) Spike-timing dynamics of neuronal groups. *Cereb Cortex* 14(8):933–944
- Manrubia SC, Mikhailov SM (1999) Synchronization and clustering in randomly coupled chaotic dynamical networks. *Phys Rev E* 60:1579–1589
- Mei SH, He MY, Chen ZM (2014) Optimizing Hopfield neural network for spectral mixture unmixing on GPU platform. *IEEE Geosci Remote Sens Lett* 11(4):818–822
- Morris C, Lecar H (1981) Voltage oscillations in the barnacle giant muscle fiber. *Biophys J* 35:193–213
- Nomura M, Fukai T, Aoyagi T (2003) Synchrony of fast spiking interneurons interconnected by GABAergic and electrical synapses. *Neural Comput* 15:2179–2198
- Perc M (2007) Stochastic resonance on excitable small-world networks via a pacemaker. *Phys Rev E* 76:066203
- Perc M, Gosak M (2008) Pacemaker-driven stochastic resonance on diffusive and complex networks of bistable oscillators. *New J Phys* 10:053008
- Qu JY, Wang RB (2012) Dynamics of a cortical neural network based on a simple model. *Chin Phys Lett* 29(8):080503
- Qu JY, Wang RB, Du Y, Cao JT (2012) Synchronization study in ring-like and grid-like neuronal networks. *Cogn Neurodyn* 6(1):21–31
- Qu JY, Wang RB, Yan CK, Du Y (2014) Oscillations and synchrony in a cortical neural network. *Cogn Neurodyn* 8(2):157–166
- Raghav S, Marongiu A, Atienza D (2015) GPU acceleration for simulating massively parallel many-core platforms. *IEEE Trans Parallel Distrib Syst* 26(5):1336–1348
- Rose RM, Hindmarsh JL (1989) The assembly of ionic currents in a thalamic neuron. I The three-dimensional model. *Proc R Soc Lond B* 237:267–288
- Rulkov NF (2002) Modeling of spiking-bursting neural behavior using two-dimensional map. *Phys Rev E* 65:041922
- Rulkov NF, Bazhenov M (2008) Oscillations in large-scale cortical network models. *J Biol Phys* 34:279–299
- Rulkov NF, Timofeev I, Bazhenov M (2004) Oscillations in large-scale cortical networks: map-based model. *J Comput Neurosci* 17:203–223
- Stein RB (1967) Some models of neuronal variability. *Biophys J* 7(1):37–68
- Sun WG, Wang RB, Wang WX, Cao JT (2010) Analyzing inner and outer synchronization between two coupled discrete-time networks with time delays. *Cogn Neurodyn* 4(3):225–231
- Volman V, Baruchi I, Ben-Jacob E (2005) Manifestation of function-follow-form in cultured neuronal networks. *Phys Biol* 2(2):98–110
- Watts DJ, Strogatz SH (1998) Collective dynamics of 'small world' networks. *Nature* 393:440–442
- Wei DQ, Luo XS (2007) Ordering spatiotemporal chaos in discrete neural networks with small-world connections. *Europhys Lett* 78:68004
- Yu S, Huang DB, Singer W et al (2008) A small world of neuronal synchrony. *Cereb Cortex* 18(2):2891–2901