Article

Chinese Science Bulletin

Geography

July 2012 Vol.57 No.19: 2442–2452
doi: 10.1007/s11434-012-5085-3

GPU-CA model for large-scale land-use change simulation

LI Dan[1], LI Xia[1*], LIU XiaoPing[1], CHEN YiMin[1], LI ShaoYing[1], LIU Kai[2], QIAO JiGang[3], ZHENG YiZhong[1], ZHANG YiHan[1]& LAO ChunHua[1]

[1] *Guangdong Key Laboratory for Urbanization and Geo-simulation, School of Geography and Planning, Sun Yat-sen University, Guangzhou 510275, China;*

[2] *Guangzhou Institute of Geography, Guangzhou 510070, China;*

[3] *Resource and Environment School, Guangdong University of Business Studies, Guangzhou 501320, China*

Received December 5, 2011; accepted January 16, 2012; published online April 24, 2012

Land-use change simulation for large-scale regions (i.e. provincial regions or countries) is very useful for many global studies. Such simulation, however, is affected by computational capability of general computers. This paper proposes a method to implement cellular automata (CA) for land use change simulation based on graphics processing units (GPUs). This method can be applied to large-scale land-use change simulations by combining the latest GPU high-performance computing technology and CA. We carried out the experiments by simulating land-use change processes at a provincial scale. This involves a lot of sophisticated techniques, such as model mapping, and computational procedure of GPU-CA model. This proposed model has been validated by land-use change simulation in Guangdong Province, China. The comparison indicates that the GPU-CA model is faster than traditional CA by 30 times. Such improvement is crucial for land-use change simulations in provincial regions and countries. The outputs of the simulation can be further used to provide information to other global change models.

graphics processing unit (GPU), cellular automata (CA), land-use change simulation, large-scale, global change

Citation: Li D, Li X, Liu X P, et al. GPU-CA model and large-scale land-use change simulation. Chin Sci Bull, 2012, 57: 2442–2452, doi: 10.1007/s11434-012-5085-3

In recent years, many scholars have been using cellular automata (CA) to simulate changes in patterns and processes, such as urban expansion, disease scattering, fire spread, population migration, economic development, desertification, and flood inundation. This has led to some remarkable advances in various geographical researches [1,2]. For example, Batty and Xie simulated urban expansion in Buffalo, New York [3]. Clarke et al. [4] used the SLEUTH model to simulate urban development in the San Francisco Bay area in the U.S. Wu and Webster [5] simulated urban land-use change in Guangzhou City, and Li and Liu [6] simulated urban expansion in the Pearl River Delta region of South China. Additionally, Li et al. [7,8] proposed the theory of Geographical Simulation and Optimization Systems (GeoSOS), which combines the theories of land-use change simulation and spatial optimization. It thus provides a powerful spatial-process analytical tool for land-use change simulation and optimization, under a complicated resource environment.

Land-use changes at larger scales (e.g. provincial, national, or even at global scale) have important impacts on large-scale environmental problems, such as global climate change, food safety, carbon recycling, and so on. Simulating and predicting large-scale land-use changes are necessary for the studies of these environmental changes.

When simulating large-scale land-use changes, selecting proper data and experimental techniques are very important to generate plausible results. Raster data are usually used in spatial simulation experiments; thus, selecting proper data precision is considerable. For low-resolution data, a single data pixel represents multiple land-use types, and the composition of these types is expressed as percentages [9]. This may affect the precision of model analysis results. However, for high-resolution data, each pixel can completely repre-

*Corresponding author (email: lixia@mail.sysu.edu.cn) sent the dominant land-use type within the involved space. Moreover, low-resolution data lose local information and non-linear features of geographic patterns; thus, the necessity arises for using high-resolution data to ensure that micro-scale information will not be omitted. Cellular automata (CA) use a bottom-up approach to explore the emerging behavior of complex systems, it is preferable to use fine-grain data in simulations. Therefore, using high-resolution data in large-scale land-use change simulation is essential.

However, large-scale land-use change simulation with high-resolution data requires a large amount of data, complicated computing processes, and very long execution time. For example, the data storage size can be hundreds of megabytes or even several gigabytes. A PC is unable to perform such simulation experiments, because of its limited computing capability and CPU-based serial computing pattern. Therefore, a parallel computing pattern is needed to improve computational capability. There have been some studies on parallel computing-based land-use change simulation, e.g., load balancing-based parallel CA simulation [10], grid computing-based CA simulation, and others. However, these methods have some shortcomings, such as high computational cost, complex configuration, and lower performance relative to computing acceleration. Thus, a new parallel computing pattern with low cost, simple configuration, and better acceleration performance is necessary, to provide better computational capability for large-scale land-use change simulation.

Therefore, a new method is proposed that combines the graphics processing unit (GPU) high-performance technique with CA. This high-performance technique uses GPUs in the computer graphics card to execute general-purpose computations. As a type of high-performance computing technique, GPU computing is a recently developed, typical parallel computing model. It is characterized by low cost, and a high degree of parallelization, programmability, and flexibility [11]. The “CPU + GPU” computing pattern is a trend representing the future development of high-performance computing techniques [12]. NVIDIA proposed the CUDA (Compute Unified Device Architecture) computing platform in 2007 [13], which provided a GPU-based general-purpose computing environment and software architecture that can be developed using C-like language. With this platform, general-purpose computing tasks can be performed with any CUDA-supported computer graphics cards, so the computational cost is relatively low. Presently, GPU-based high-performance computing is extensively applied in fields such as physical simulation [14], image processing [15], three-dimensional terrain generation, signal processing, artificial intelligence, and others. There are greatly improves on the computational efficiency comparing the original CPU-based pattern. Geographic problems are generally complex; thus, applying the GPU computing technique to geographic simulations of large-scale and/or high-resolution land-use changes would be of great importance. Some studies have focused on integration of the GPU technique with CA [16,17], but there is no application to geographical simulation.

1 GPU-CA computing model framework

The GPU-CA computing model is a CA model for simulating land-use changes combined with the GPU-based general-purpose computing technique. The model uses the high-performance computing capability of the latter technique. It improves the computational efficiency of CA simulation, especially in solving computing bottleneck problems within large-scale and/or high-resolution land-use change simulation. The GPU-CA model framework includes three aspects, namely, mapping the CA model, addressing key technical problems, and designing computing procedures. The GPU-CA land-use change model can use any mature CA model, such as logistic regression-based CA [18], artificial neural-network CA [2], decision tree-based CA [19], and others. We focus on the non-urban to urban land conversion process, thus, we use the logistic regression-based CA model and the data parallel computing pattern.

1.1 Model mapping

According to the logistic regression model, the probability of a site occurring land conversion can be computed as follows:

$$P_g(S_{ij}=\text{urban})=\frac{\exp(z)}{1+\exp(z)}=\frac{1}{1+\exp(-z)}, \tag{1}$$

where $P_g(S_{ij}=\text{urban})$ refers to the conversion probability of a cell at position *ij* in cellular space when the land-use state is urban; its value is between zero and one. z describes the influence of spatial factors that affect land-use change, and is expressed as

$$z=\alpha+\sum_k b_k x_k, \tag{2}$$

where α refers to the regression constant, b_k is the coefficient of the logistic regression model, and x_k represents the value of spatial factors of the cell at position *ij*.

The probability of land conversion in a cell at position *ij* and time *t* is expressed as

$$p_{ij}^t=(1+(-\ln\gamma)^\alpha)\times\frac{1}{1+\exp(-z_{ij}^t)}\times\text{con}(s_{ij}^t)\times\Omega_{ij}^t, \tag{3}$$

where $(1+(-\ln\gamma)^\alpha)$ is a stochastic interference representing the stochastic influence on land-use change, γ is a stochastic value between zero and one, and α is an integer between 1 and 10. $\text{con}(s_{ij}^t)$ is the constraint condition for land conversion of a cell at position *ij*. Ω_{ij}^t represents the neighborhood influence on the CA model, indicating the land-use state within the neighborhood window.

Finally, the following equation determines whether land conversion will occur within a cell:

$$\begin{cases} p_{ij}^t \geqslant p_{\text{threshold}}, & \text{changed,} \\ p_{ij}^t < p_{\text{threshold}}, & \text{unchanged.} \end{cases} \tag{4}$$

If the value of p_{ij}^t is larger than or equal to the preset threshold value $p_{\text{threshold}}$ and the cell's state is non-urban, then the cell's state will be converted to urban; otherwise, there will be no conversion. One iteration is finished when computation is done for the entire cellular space. Through multiple iterations, the entire process of land-use change can be simulated.

The GPU-CA computing model maps the aforementioned computation procedure to the GPU programming and memory models. The GPU-CA model architecture is shown in Figure1.

GPU general-purpose computing can be logically divided into three levels. The smallest computing unit is the Thread. Multiple threads compose a Block (one block has one shared memory), and one or more blocks compose a Grid.

The GPU-CA computing model uses a data parallel computing pattern, which maps a thread to a cell in the cellular space, and this thread is responsible for the computation of the corresponding cell. The data, consisting of the simulation data and related space variables, are read by graphics cards from the computer mainframe (host) memory to global memory. These data are then copied to each shared memory of the blocks, where each thread reads the data according to the cell it maps. Since the CA model requires information about the neighborhood of each cell, each thread simultaneously reads data from the neighbors of the mapped cell. We adopted the 3×3 Moore neighborhood. When the data reading is complete, a computation iteration is done by the multiple thread units, and resulting data are written back to global memory. The next iteration then begins, until the termination condition is reached.

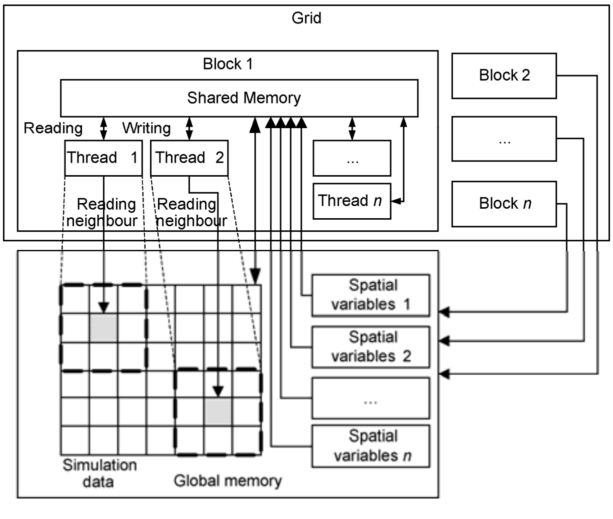

Figure 1 GPU-CA computing model architecture.

1.2 Key technical problems solved by GPU-CA

(i) Data mapping. The CA model adopts a raster data structure and is naturally similar to the GPU thread computing model. Therefore, the GPU-CA model maps raster data units to the thread units in GPU, and uses a data parallel computing pattern. Related factors in this process must be considered, such as the mode of data index mapping, number of GPU computing units, and size of the memory units. The mode of data index mapping can take the form of a one- or two-dimensional array, which links the position index information in the thread computing model with their counterparts in the raster data, thus ensuring correctness of the mapping relationship. The two mapping modes are shown in Figure 2(a) and (b), where T1, T2, $\cdots$ and Tn represent threads in a block. As shown in Figure 2, the index mapping using one-dimensional array mode is characterized by a simple relationship and directness. The index mapping using two-dimensional array mode has representation advantages but also has a more complicated relationship. Our index mapping uses the two-dimensional array mode. The number of GPU computing units and memory unit size determine the data size that can be processed by the computing function (kernel) at each execution, which is closely related to the graphics card hardware attributes. If the data size for execution is less than the overall data size, then data cannot be read into the graphics card memory once. Thus the data must be divided into blocks, for reading and computing by the kernel function multiple times.

(ii) Data competitive writing. In CA simulation, global information is generally present, e.g. the number of cells that change state after each iteration. For this type of data, each cell can alter the values of these data. As shown in Figure 3, each thread in each block can possibly write the value of global information in global memory during each iteration. In the serial computing pattern, each cell can successively

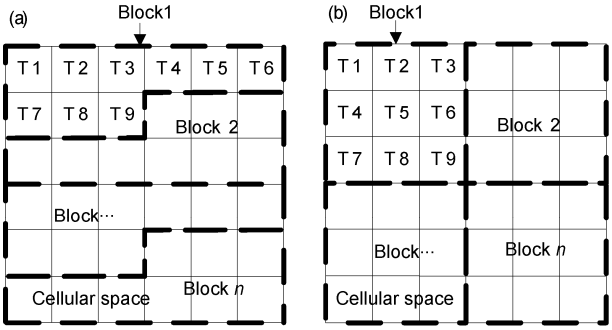

Figure 2 Diagram showing two modes of data index mapping. (a) Index mapping using one-dimensional array mode. (b) Index mapping using two-dimensional array mode.

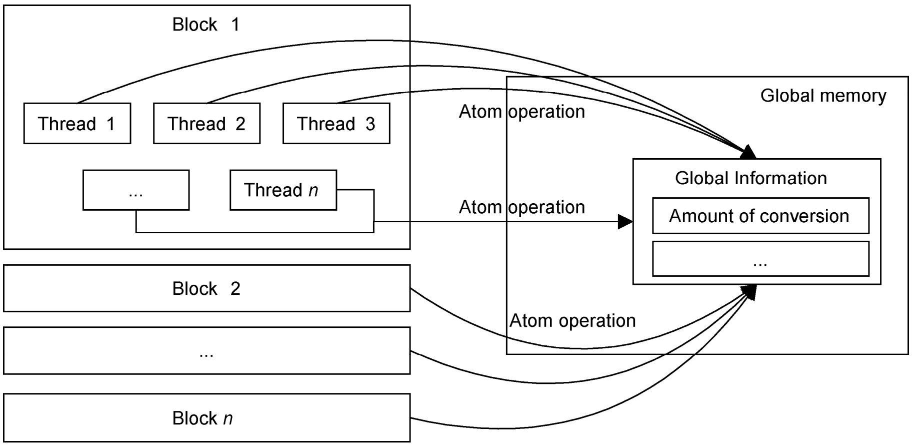

Figure 3 Phenomenon of competitive data writing solved in GPU-CA.

alter the value and will not cause conflict between cells. However, in the parallel computing pattern, if each computing unit simultaneously performs data alteration, a problem with competitive data writing results. This means that the correctness of the global information cannot be guaranteed. Therefore, the atom operation function in the CUDA environment is introduced to avoid this competitive data writing. The atom operation function ensures the correctness of a value while multiple threads are simultaneously written to the same address by the function's computing mechanism. The atom operation function thereby effectively avoids the competitive data writing problem, solving related problems that may arise in large-scale land-use change simulation.

1.3 GPU-CA computing procedure

The procedures of GPU general-purpose computing generally include initialization of the computing environment, data transfer, parallel computing using the kernel function, and exiting the environment. The pseudo code for the GPU-CA model is:

```
1. Defining sizes of the grid and block used in computing, and corresponding storage space sizes.
2. Starting the CUDA environment.
3. Reading data into the memory of the host side.
4. At the GPU side, allocating the graphic memory and copying data into it from the host side.
5. Using the kernel function to calculate the $Pg$ value of the entire cellular space.
6. Using the kernel function to perform multiple iteration computation.
    While (if the termination conditions of the simulation are not reached)
      6.1 For each block, reading the land-use type and $Pg$ values of corresponding cells from global memory into shared memory, according to the indexes.
      6.2 For each thread, reading 3×3 neighborhood information of the corresponding cell from global memory, according to the indexes.
      6.3 For each thread, computing the land conversion probability using the logistic regression-based CA formulas.
          If (conditions of conversion into urban are met)
              Converting land-use type of the corresponding cell into urban land
          End if
      6.4 For each thread, if the land-use type of the corresponding cell is converted to urban land, then the atom operation function is used to update the number of total converted cells in this iteration.
      6.5 For each block, copying the updated land-use type value of the corresponding cells from shared memory to global memory.
    End while
7. Computing the accuracy of the simulated result.
8. Copying simulated result from the GPU graphic memory to the host memory, and outputting the result to files.
9. Releasing storage spaces at the GPU and host sides, and exiting the CUDA environment.
```

2 Case application and analysis

The study area is in Guangdong Province, China, which has a land area of 179800 km^2. The land use classification data of Guangdong Province from 2000, 2005, and 2006 were acquired, as well as related space variable data. The land-use classification data are raster data files in ArcGIS ASCII format, with 80 m spatial resolution and the raster size is 9792×7376. The single data document has a storage size of approximately 500 MB. There are six land-use types, namely, farmland, forest, grassland, water area, construction land, and unused land. The space variables include distance to city center for each raster cell, and distances to the railway, highway, and roadway. Based on the logistic regression-based CA model, the land-use classification and space variable data from 2000 and 2005 were used to determine the rules of conversion, which define the conversion of nonconstruction to construction land in Guangdong Province. Subsequently, the land use data of 2005 was used as initial data to simulate the land conversion process in the period 2005–2006. After the simulation results were produced, they were then compared with the 2006 land-use classification data using two accuracy assessment approaches to validate the simulation result, these are the point-to-point and naked-eye approaches [1]. The general CA provided by GeoSOS software (GeoSOS, http://www.geosimulation.cn/) was used to simulate the same land-use change process in this province, for analyzing the degree of improvement in computational efficiency by GPU-CA. Finally, the construction land conversion process in Guangdong Province during 2010 and 2015 was predicted. Figure 4 shows the GPU-CA simulation results.

To compare with the general CA based on the CPU serial computing pattern, we adopted a powerful hardware platform configuration. This platform consists of 2× Intel Xeon E5620 2.5 GHz CPU, 24 GB mainframe memory, Windows 7 64-bit professional edition operating system, and Tesla 1060C graphics card manufactured by NVIDIA Inc., which has 4GB graphic memory and1.30 GHz core frequency. The GPU computing environment is CUDA 3.2, and the development environment is Microsoft Visual C++ 2008.

Figure 4(a)–(c) describes land use in Guangdong Province in 2000, 2005, and 2006, respectively. Figure 4(d) describes simulation results for 2006, using land use transition patterns extracted from the data of 2000 and 2005. Figure 4(e) and (f) shows predicted results for 2005–2010 and 2005–2015, respectively.

By using a point-to-point comparison method which commonly used in image consistency evaluation, a confusion matrix is generated to test the simulation accuracy. The results indicate that the overall accuracy of GPU-CA simulation is 82.9%, indicating that the GPU-CA model is very effective and can be applied to large-scale land-use change simulations.

Based on experimental data and the simulated and predicted results, we evaluated the percentages of construction land areas to total land area for each municipal administrative region, and the percentage of construction land expansion for each city in Guangdong Province for three periods. These are 2000–2006, 2000–2010, and 2000–2015 (Table 1). Each municipal administrative region in the province showed an expansion of construction land. Percentages of construction land area to total land area over the entire province were 4.57% in 2000 and 7.39% in 2006; they are predicted to be 7.70% in 2010 and 7.89% in 2015. The percentage of construction land expansion was 61.76% in 2000–2006, and is predicted to be 68.60% in 2010 and 72.55% in 2015. We conclude that a large area of land is converted to construction land in Guangdong Province, and the tight situation for land resources will continue in the future. However, the expansion ratio is different between municipal administrative regions.

Based on the different percentage of construction land area to total land area in each municipal administrative region in 2000–2015, the regions in Guangdong Province can be divided into three types (Figure 5). Guangzhou, Shenzhen, Dongguan, Foshan, Zhongshan, and Zhuhai show percentages exceed 20%. Jiangmen, Yangjiang, Maoming, Zhanjiang, Huizhou, Shantou, Shanwei, Chaozhou, and Jieyang

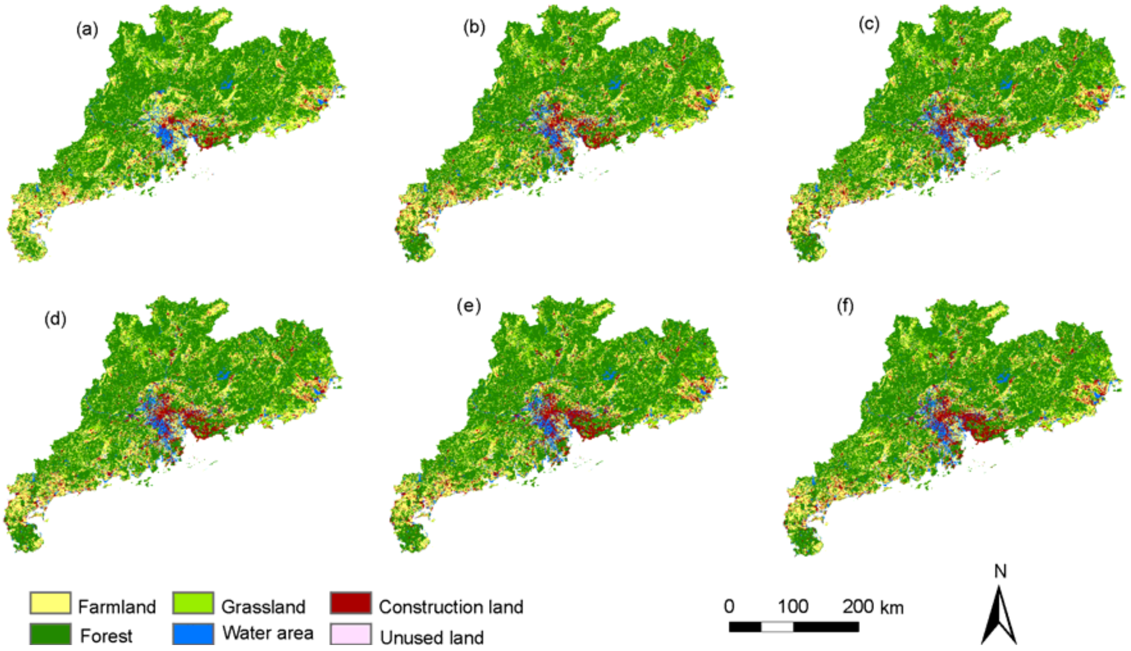

Figure 4 Diagram showing the simulated results of GPU-CA.

Table 1 Construction land area of all municipal administrative regions in Guangdong, and percentages of construction land and its expansion

Municipal administrative region	Land area (km^2)	2000 (km^2)	2005 (km^2)	2006 (km^2)	Simulation result of 2006 (km^2)	Simulation result of 2010 (km^2)	Simulation result of 2015 (km^2)	Percentage of construction land in 2006 (%)	Percentage of construction land in 2010 (%)	Percentage of construction land in 2015 (%)	Percentage of construction land expansion in 2000–2006 (%)	Percentage of construction land expansion in 2000–2010 (%)	Percentage of construction land expansion in 2000–2015 (%)
Guangzhou	7226.27	844.04	1328.59	1374.23	1373.40	1417.70	1447.62	19.02	19.62	20.03	62.82	67.97	71.51
Shenzhen	1863.31	656.09	799.53	842.96	843.62	865.18	876.98	45.24	46.43	47.07	28.48	31.87	33.67
Zhuhai	1449.95	180.24	261.00	283.33	283.48	289.26	295.94	19.54	19.95	20.41	57.19	60.48	64.19
Shantou	2152.27	361.65	390.48	411.41	410.77	418.30	427.90	19.12	19.44	19.88	13.76	15.67	18.32
Foshan	3845.65	556.83	929.53	1085.32	1085.50	1110.41	1147.26	28.22	28.87	29.83	94.91	99.42	106.04
Shaoguan	18439.42	225.71	343.51	362.85	362.75	370.75	376.33	1.97	2.01	2.04	60.76	64.26	66.73
Heyuan	15659.39	73.29	285.03	318.22	318.97	326.78	339.50	2.03	2.09	2.17	334.22	345.90	363.25
Meizhou	15903.24	164.61	435.06	461.54	461.56	479.32	491.22	2.90	3.01	3.09	180.38	191.17	198.41
Huizhou	11312.79	520.26	746.44	802.98	804.15	840.12	868.60	7.10	7.43	7.68	54.34	61.48	66.95
Shanwei	4800.22	134.78	275.97	281.63	282.04	285.70	288.22	5.87	5.95	6.00	108.95	111.97	113.84
Dongguan	2426.23	696.84	935.00	1058.54	1058.96	1252.67	1284.62	43.63	51.63	52.95	51.90	79.76	84.35
Zhongshan	1692.76	201.15	428.58	445.82	445.97	459.40	467.48	26.34	27.14	27.62	121.64	128.39	132.41
Jiangmen	9281.44	582.55	693.81	756.23	756.48	770.90	793.60	8.15	8.31	8.55	29.81	32.33	36.23
Yangjiang	7788.38	243.83	423.92	438.88	438.52	451.97	461.08	5.64	5.80	5.92	79.99	85.36	89.09
Zhanjiang	12177.76	862.58	1255.70	1286.78	1286.99	1322.14	1342.91	10.57	10.86	11.03	49.18	53.28	55.69
Maoming	11337.11	663.62	680.32	701.15	701.70	733.89	749.65	6.18	6.47	6.61	5.66	10.59	12.96
Zhaoqing	14880.81	216.28	467.16	500.59	499.48	521.11	538.30	3.36	3.50	3.62	131.46	140.95	148.90
Qingyuan	18977.04	242.02	679.04	723.35	724.09	751.76	775.98	3.81	3.96	4.09	198.89	210.62	220.63
Jieyang	3101.45	144.19	220.13	228.01	227.84	230.50	233.96	7.35	7.43	7.54	58.13	59.86	62.26
Chaozhou	5238.42	326.03	461.95	462.67	462.64	479.21	483.73	8.83	9.15	9.23	41.91	46.98	48.37
Yunfu	7788.36	207.70	264.13	283.35	283.75	286.87	294.34	3.64	3.68	3.78	36.42	38.12	41.71

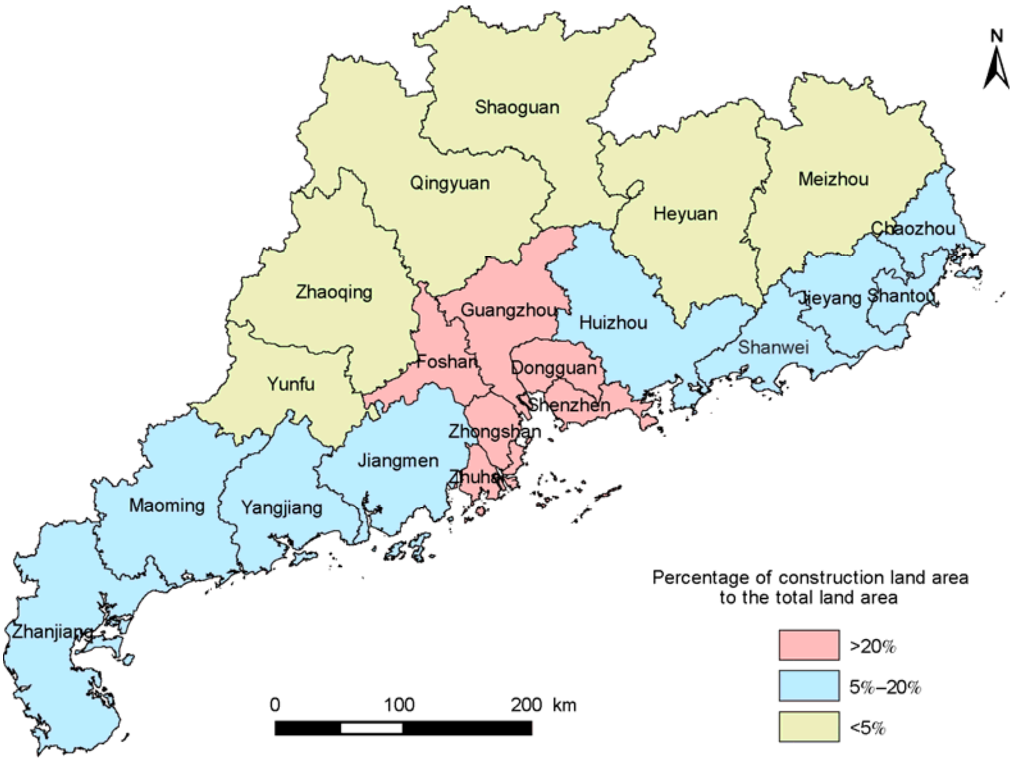

Figure 5 Diagram showing percentages of construction land area to total land area in each municipal administrative region of Guangdong Province for 2000–2015.

show percentages varying between 5%–20%, where as Shaoguang, Zhaoqing, Qingyuan, Yunfu, Heyuan, and Meizhou show less than 5%. The six cities with percentages of over 20% are all in the Pearl River Delta region. According to the rapid growth of the regional economy in 2000–2006, a larger area of land was converted to construction land. In particular, in Dongguan, the percentage was 43.63% in 2006 and will be 51.63% in 2010 and 52.95% in 2015. Thus, large areas of land will be converted to construction land in the future (Figure 6). The cities with percentages between 5%–20% are generally along coastal regions in the province. These cities benefited from an opening-up policy and location advantages, so their economic development is also rapid. Again, a large construction land area is needed to meet the requirements of economic development. The cities with percentages of less than 5% are generally in the northwestern and northeastern mountainous regions and in economically underdeveloped regions. Only small areas of construction land are required in these regions, because of constraints in economic conditions.

Based on the ratio of construction land expansion in the various regions from 2000–2015, the municipal administrative regions in Guangdong Province can also be divided into three types (Figure 7). For Zhongshan, Zhaoqing, Qingyuan, Meizhou, Heyuan, and Shanwei, expansion rates all exceeded 100%. A typical rate was 132.41% for Zhongshan, where large areas of land have been converted to construction land (Figure 8). For Guangzhou, Zhuhai, Foshan, Shaoguang, Dongguan, Huizhou, Yangjiang, Zhanjiang, and Chaozhou, expansion rates all ranged between 50%–100%. A typical rate was 66.95% for Huizhou, which belongs to the intermediate ratio of construction land expansion (Figure 9). For Shenzhen, Shantou, Jiangmen, Jieyang, and Yunfu, expansion rates were all less than 50%. A typical rate is 41.71% for Yunfu, where less land is converted to construction land (Figure 10).

In simulating multiple urban expansion types as discussed above, the GPU-CA model can precisely simulate the construction land development process for each type. This indicates that the model is very adaptable and capable of effectively simulating the pattern and process of construction land development.

3 GPU-CA model performance analysis

For the performance analysis of parallel computation, the concept of speedup is introduced [20,21]. It uses the ratio of parallel computation time to CPU serial computation time, to represent the acceleration performance of parallel computing. For analyzing GPU-CA model performance, the time for GPU-CA simulation and that for CPU computation under the same conditions, are recorded. The speedup is calculated to determine how the GPU computing improves the CA simulation efficiency. To compare acceleration performances of GPU-CA more completely, Dongguan, the

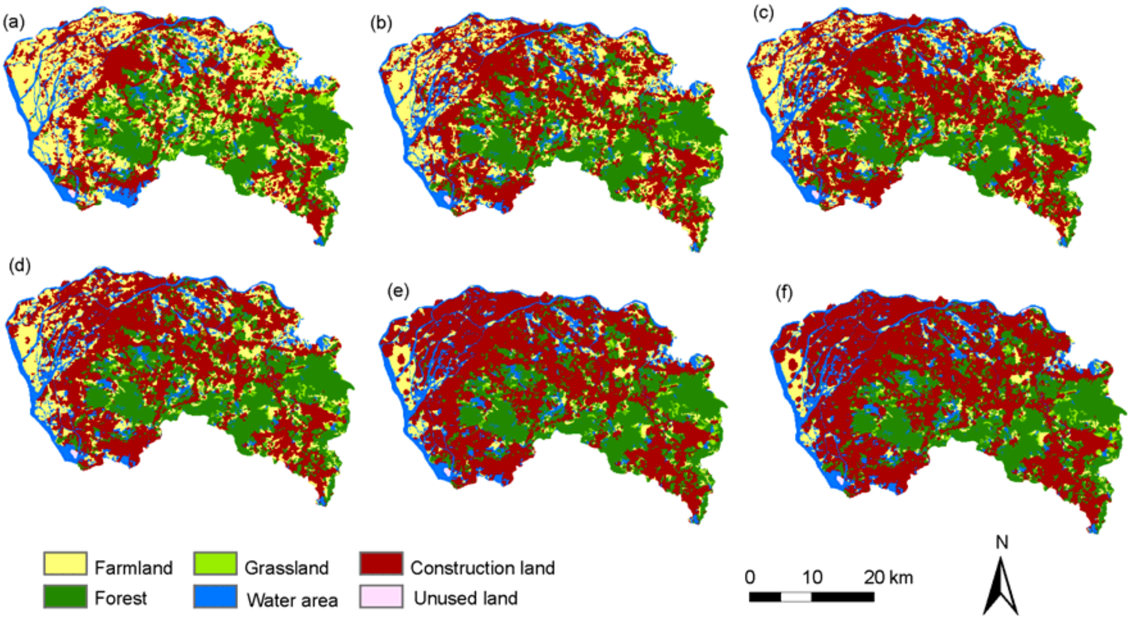

Figure 6 Diagram showing construction land expansion in Dongguan from 2000–2006, and the GPU-CA simulated and predicted results. (a)–(c) describe land use in Dongguan in 2000, 2005, and 2006, respectively, (d) describes simulated results for 2006, (e) and (f) show predicted results for 2010 and 2015, respectively.

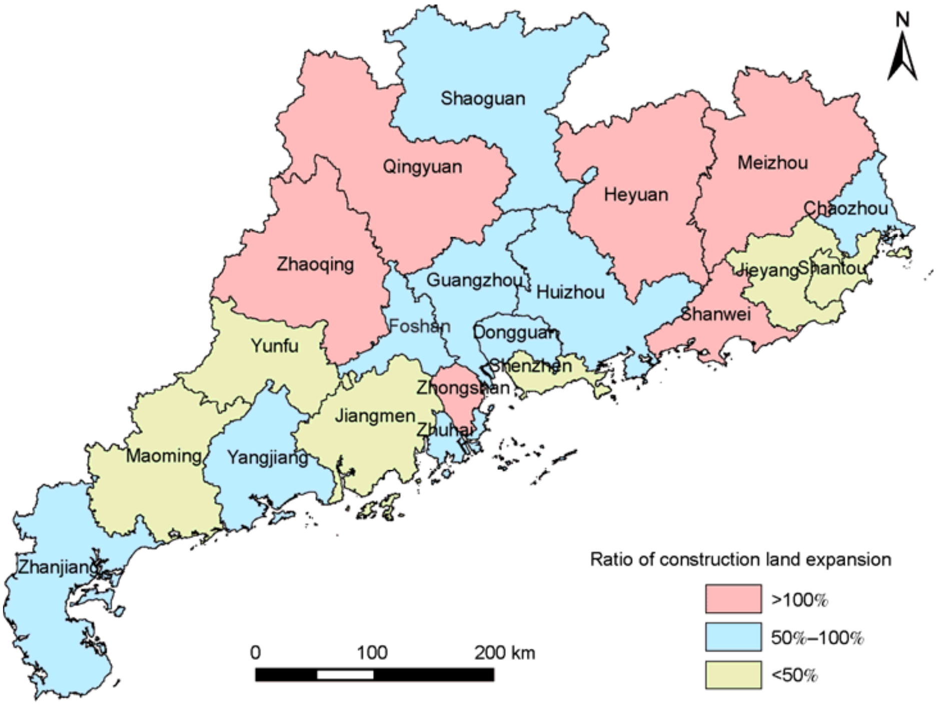

Figure 7 Diagram showing ratio of construction land expansion in each municipal administrative region of Guangdong Province from 2000–2015.

Pearl River Delta region, and Guangdong Province were selected as study zones for the simulation respectively. CPU-CA and the general CA in GeoSOS were used for the simulation computations. The computation performances of the two different computing patterns were compared, and results are shown in Table 2. To ensure that the comparison conditions were the same, there were no image refresh operations in the simulations with GeoSOS. Instead, only simulation computation was done. Moreover, data reading and output processes in the two patterns were basically the same. Therefore, only the time for CA simulation computing are compared, to investigate the difference in computation efficiency between the GeoSOS serial and GPU-CA parallel patterns.

As shown in Table 2, GPU-CA can remarkably accelerate the CA simulation, with the greatest speedup reaching 33.97. This is much greater than the execution efficiency of the CPU computing pattern, indicating that GPU-CA is very

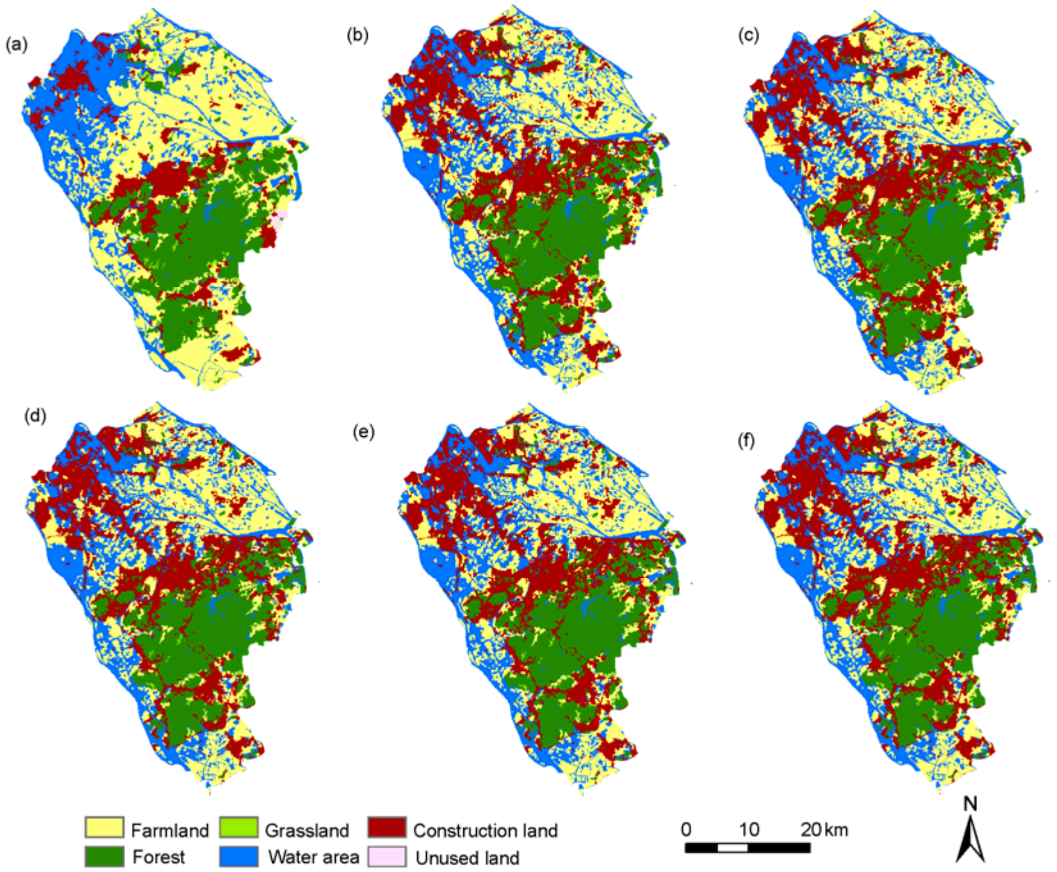

Figure 8 Diagram showing construction land expansion in Zhongshan from 2000–2006, and the GPU-CA simulated and predicted results. (a)–(c) describe land use in Zhangshan in 2000, 2005, and 2006, respectively, (d) describes simulated results for 2006, (e) and (f) show predicted results for 2010 and 2015, respectively.

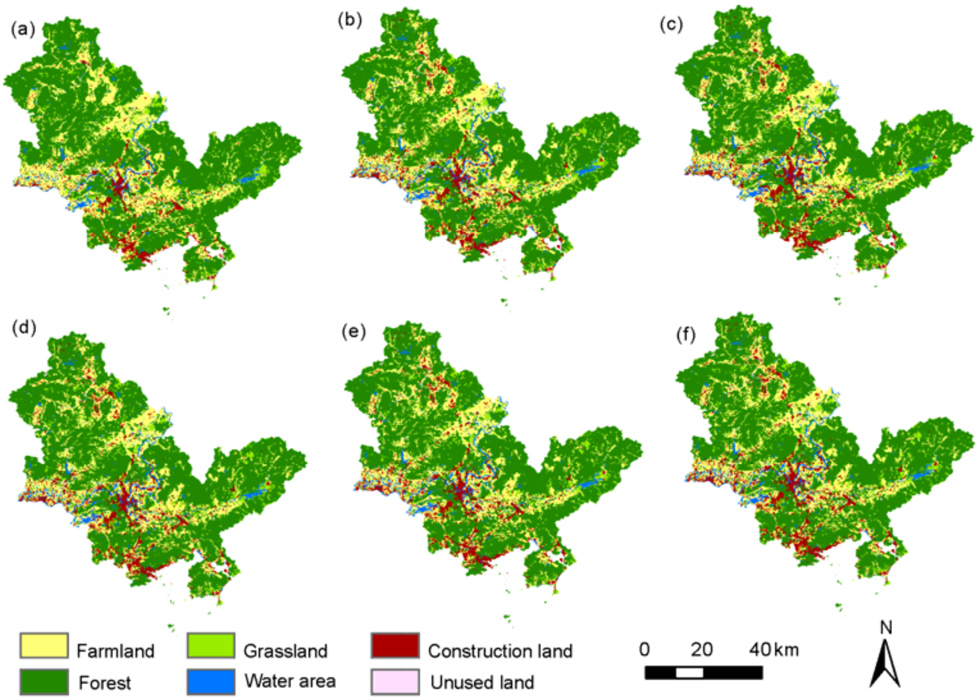

Figure 9 Diagram showing construction land expansion in Huizhou from 2000–2006, and the GPU-CA simulated and predicted results. (a)–(c) describe land use in Huizhou in 2000, 2005, and 2006, respectively, (d) describes simulated results for 2006, (e) and (f) show predicted results for 2010 and 2015, respectively.

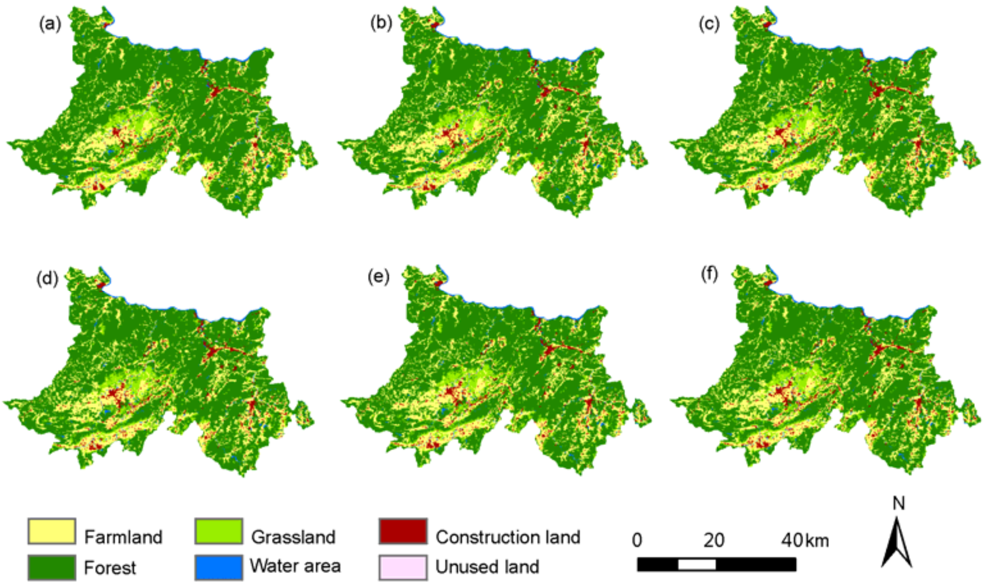

Figure 10 Diagram showing construction land expansion in Yunfu from 2000–2006, and the GPU-CA simulated and predicted results. (a)–(c) describe land use in Yunfu in 2000, 2005, and 2006, respectively, (d) describes simulated results for 2006, (e) and (f) show predicted results for 2010 and 2015, respectively.

Table 2 Table of comparison of computation time between GPU-CA and general CA

Study area	Cells count	GPU-CA (s)	General CA (s)	Speedup
Dongguan	768×544	1.059	17.700	16.71
Pearl River Delta	2912×2096	14.715	258.876	17.59
Guangdong	9792×7376	150.729	5102.147	33.97

efficient in computation.

Figure 11 shows the relationship between the GPU-CA computation scale and time speedup. It shows that time speedup increases with computation scale. This is because under computation involving small data amounts, the time required for data transfer maybe greater than the time for data computation. As the computational scale increasing, the time required for data transfer decreases in total computation time. Therefore, given a large volume of data, GPU-CA achieves a better acceleration performance. This result demonstrates that GPU-CA is more suitable for processing problems involving large amounts of data, and it’s consistent with theoretical analysis of the acceleration performance of parallel computing [20]. Further, the GPU high-performance computation is characterized by good parallelism and high computing density. Hence, we conclude that the GPU-CA computing model is very suitable for solving problems involving large-scale land-use change simulation.

We also compare the acceleration performances of GPU-CA with the load balancing-based parallel CA [10]. Parallel CA uses a 3×3 neighborhood to simulate land conversion in the Pearl River Delta region, with the computation time approximately 1200–1600 s and a speedup around 2. The simulation time using GPU-CA is approximately 15 s, and speedup around 18. Therefore, GPU-CA shows much improved computing performance over parallel CA. Moreover, parallel CA requires eight PCs for parallel computing. GPU-CA only requires one computer with a graphics card supporting the CUDA environment; thus, the computational cost is much lower than parallel CA. This result fully demonstrates that GPU-CA has a low computational cost and high computing capacity.

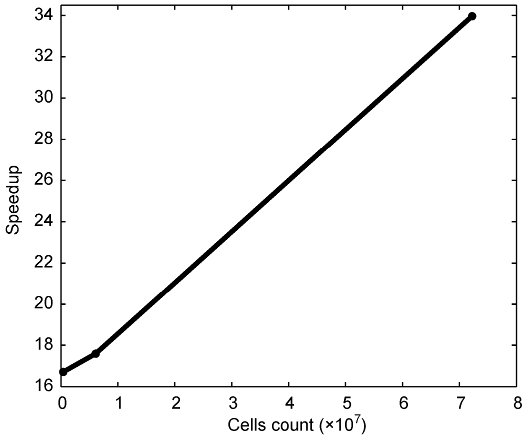

Figure 11 Diagram showing relationship between GPU-CA computing scale and time speedup.

4 Conclusions

We used the GPU general-purpose computing technology for large-scale land-use change simulation, and established the GPU-CA computing model. The proposed model not only guarantees valid simulation results, but also remarkably improves the efficiency of simulation computation. This new model helps solve the problems encountered in some simulations which have a computation with large amounts of spatial data and excessively long computation time. These problems cannot be effectively solved by the original CPU serial computing pattern. Hence, this model is a highly efficient computing tool for future large-scale geographic simulation and computation. In comparison with other parallel computing models, the GPU general-purpose model has advantages of low computational cost, high computing density, easy configuration and installation, computational speedup increasing with the increased computational scale. Thus, the model is more suitable for geographic simulation and intensive geographic computing. Guangdong Province was selected as the study area here, for large-scale land-use change simulation. The GPU-CA computing model can be conveniently applied to the entire country or even on a global scale.

We used the logistic regression-based CA model for land-use change simulation. This is suitable for simulating changes of a single land-use type, but the model cannot produce the patterns of conversion for multiple land-use types. Therefore, models for multiple land-use types, like artificial neural-network-based CA, decision tree-based CA, and others, can be used in the future to better explain and simulate the land-use change process. Moreover, GPU computation time maybe remarkably reduced by using program optimization. Alternatively, parallel computing protocols, such as OpenMP, can be used to construct a computing cluster based on multiple GPU graphics cards. It can improve computing efficiency and expand the data processing scale. This is the subject of our future research.

This work was supported by the Key National Natural Science Foundation of China (40830532), and the National Natural Science Foundation of China (40901187 and 41171308).

1 Li X, Yeh A G, Liu X P, et al. Geographical Simulation Systems: Cellular Automata and Multi-agent Systems (in Chinese). Beijing: Science Press, 2007. 250–258

2 Li X, Yeh A G. Neural-network-based cellular automata for simulating multiple land use changes using GIS. Int J Geogr Inf Sci, 2002, 16: 323–343

3 Batty M, Xie Y. From cells to cities. Environ Plan B-Plan Design, 1994, 21: 531–548

4 Clarke K C, Hoppen S, Gaydos L. A self-modifying cellular automaton of historical urbanization in the San Francisco Bay area. Environ Plan B-Plan Design, 1997, 24: 247–261

5 Wu F, Webster C J. Simulation of land development through the integration of cellular automata and multicriteria evaluation. Environ Plan B-Plan Design, 2004, 5: 103–126

6 Li X, Liu X P. Case-based cellular automaton for simulating urban development in a large complex region (in Chinese). Acta Geogr Sin, 2007, 62: 1097–1109

7 Li X, Liu X P, He J Q, et al. A geographical simulation and optimization system based on coupling strategies (in Chinese). Acta Geogr Sin, 2009, 64: 1009–1018

8 Li X, Li D, Liu X P, et al. The implementation and application of geographical simulation and optimization systems (GeoSOS) (in Chinese). Acta Sin Nat Univ Sunyatseni, 2010, 49: 1–5

9 Veldkamp A, Fresco L O. CLUE-CR: An integrated multi-scale model to simulate land use change scenarios in Costa Rica. Ecol Model, 1996, 91: 231–248

10 Li X, Yeh A G. Parallel cellular automata for large-scale urban simulation using load-balancing techniques. Int J Geogr Inf Sci, 2010, 24: 803–820

11 Wu E H, Liu Y Q. General purpose computation on GPU (in Chinese). J Comp-Aid Des Comp Gra, 2004,16: 601–612

12 Owens J D, Houston M, Luebke D, et al. GPU Computing. Proc IEEE, 2008, 96: 879–899

13 NVIDIA. NVIDIA CUDA Compute Unified Device Architecture Programming Guide Version 2.0. 2008

14 Chen F G, Ge W, Li J H. Molecular dynamics simulation of complex multiphase flow on a computer cluster with GPUs. Sci China Ser B-Chem, 2009, 52: 372–380

15 Yang J Y, Zhang Y S, Dong G J. Investigation of parallel method of RS image SAM algorithmic based on GPU (in Chinese). Sci Surv Map, 2010, 35: 9–11

16 Gobron S, Devillard F, Heit B. Retina simulation using cellular automata and GPU programming. Mach Vision Appl, 2007, 18: 331–342

17 Ferrando N, Gosalvez M A, Cerda J. Octree-based, GPU implementation of a continuous cellular automaton for the simulation of complex, evolving surfaces. Comput Phys Commun, 2011, 182: 628–640

18 Wu F. Calibration of stochastic cellular automata: The application to rural-urban land conversions. Int J Geogr Inf Sci, 2002, 16: 795–818

19 Li X. Data mining of cellular automata's transition rules. Int J Geogr Inf Sci, 2004, 18: 723–744

20 Liu D C, Wang D X, Shen M M, et al. Potential in data parallelism (in Chinese). J Soft, 1994, 5: 8–15

21 Yang G, Wang S L, Shen J L. Parallel algorithms for solving large scale nonlinear systems of equations based on Dawning1000A (in Chinese). Chin J Comp, 2002, 25: 397–402