GPU Accelerated Plant Growth Modeling and Visualization

Xing ZHAO[1], Jingjing HU[2], Mengzhen KANG[3]

[1]School of Mathematical Sciences, Capital Normal University, Beijing 100048, China;
[2]School of Software, Beijing Institute of Technology, Beijing, 100081, China;
[3]Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China

zhaoxing_1999@yahoo.com, hujingjing@bit.edu.cn, mzkang@liama.ia.ac.cn

Abstract

We present a GPU-accelerated algorithm for plant growth modeling and visualization. In this algorithm, plant topological structures and geometrical structures are represented separately. Plant topological structures are generated by dual-scale automaton, and geometrical structures are dynamically constructed in parallel by the geometry shader of Graphics Processing Unit (GPU). This scheme greatly reduces the amount of transferred data between main memory and GPU memory, and speeds up plant growth modeling and rendering without compromising image quality. An application for cotton growth modeling and visualization is given in detail to confirm the advantages of the proposed algorithm.

1. Introduction

The modeling and visualization of plant structure and development has many potential applications in horticulture, agriculture, and forestry [1, 2]. In recent years, the use of different techniques has resulted in remarkable outcomes in the modeling and visualization of plants, such as graftals, L-systems, particle systems, fractals, specific branching patterns, IFS, GROGRA, multiscale tree graph model, Xfrog, and AMAP models. Most of them only seek to generate visually correct shapes of plants for computer graphics applications, which can be used for the 3D scenery synthesis of computer animations and games, computer-aided landscape and garden designs.

For the agronomic applications, a plant model Greenlab is developed to simulate interactions between plant structure and function [3]. Dual-scale automaton model is a part of Greenlab for simulating plant organogenesis [4], which can dynamically generate all the successional topological structures of growth process by simulating plant meristems developments.

Plant growth modeling and visualization becomes more and more important for many applications. However, it is still a computationally expensive work, because of the plant geometries with fine geometric details containing a large number of polygons, especially for large plant ecosystems. Fortunately the processing capability of consumer Graphics Processing Unit (GPU) is increasing dramatically in recent years. Several GPU-based plant rendering algorithms had been reported recently [5, 6]. Due to their increasing programmability, modern GPUs are capable of accelerating not only image rendering, but also many other applications, such as plant growth modeling.

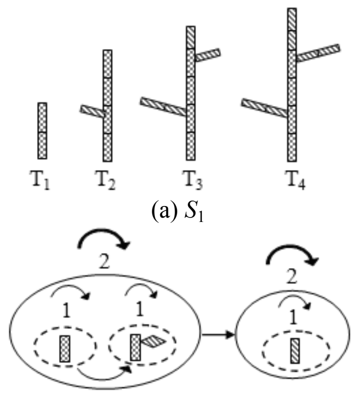
In this paper, a GPU-accelerated plant growth modeling and visualization algorithm is proposed. This algorithm separates the plant topology from the plant geometry. Plant topology is generated by dual-scale automaton model, and plant geometry is dynamically constructed in parallel by exploiting the recently introduced geometry shader of GPU. Thus the amount of transferred data between main memory and GPU memory is decreased, and plant growth modeling and rendering is speeded up. This algorithm can be employed for the high quality real-time dynamic modeling and visualization of plant growth.

2. Brief introduction of the dual-scale automaton model

Dual-scale automaton model includes two kinds of state, microstate and macrostate, respectively corresponding to metamer and growth unit. The developmental process from one state to another state is processed as a Semi-Markov chain, represented by a graph for easier understanding. The different states are distinguished by their “physiological age”, represented by a different colour for clarity. Each state can also be

978-0-7695-3988-1/10 $26.00 © 2010 IEEE
DOI 10.1109/PMA.2009.43

attached a "*state attributes table*" to record its special geometrical and morphological parameters.



(b)

Figure 1: A simple structure S_1 and the corresponding automaton: (a) from 1st cycle to 4th cycle, the resulting structure S_1 at each cycle, (b) dual-scale automaton.

Figure 1 shows a simple dual-scale automaton for illustrating this model. The automaton includes only two macrostates. The first macrostate consists of two microstates where the first one is a metamer that cannot branched, the second one bears a lateral bud that will produce a new metamer. The second macrostate includes only one microstate, which is unable to put forth a branch. The repetition time of each state is indicated by real line arrows in Figure 1.

In the example the complete structure is edified in 4 growth cycles (T1 to T4). When the first growth cycle elapses, the macrostate1, i.e. the first growth unit, is built up, which includes two metamers with only one axillary bud on its last microstate. During cycle 2, the macrostate1 is repeated and the second growth unit is built up. At the same time, the axillary bud of the first growth unit produces a lateral growth unit whose physiological age is older than the main stem's physiological age. The process goes through the 4 growth cycle in conformity with the automaton and the complete structure (S_1) is thus built up.

3. GPU-accelerated algorithm for plant growth modeling and visualization

In this algorithm, we separate plant topology from plant geometry for making the data management and implementation easier. Plant topologies at different growth stages are dynamically produced by employing the dual-scale automaton model. As for plant geometry, we notice the shape of each kind of plant organ (e.g. internode, leaf, flower, fruit, etc.) does not vary significantly during one growth stage for most of plants. Hence we select several typical shapes in different growth stages to model the growth of each kind of plant organ, and represent organ geometries by mesh-based models. We normalize the surface of each organ instance to one, and then compute the specific organs at certain growth stages by affine transform according to their geometrical parameters. This method greatly reduces the demand of memory. Besides, the mesh-based organ models can easily compute shadows or different illumination effects.

For accelerating plant growth modeling and visualization, plant geometries are dynamically constructed in parallel by applying the geometry shader of GPU. The geometry shader is a shader program model recently introduced in shader model 4.0. Different from vertex shader and pixel shader, it can generate new graphics primitives. We directly store the normalized organ mesh models in GPU memory, minimizing the communication costs between CPU and GPU, and producing great acceleration in plant simulation. Moreover, we further improve the rate of data transfers by using the powerful Vertex Buffer Object (VBO) technique.

The workflow architecture of our algorithm is outlined in Figure 2, and the detail of the algorithm is as follows:

Step1: In CPU application, the plant topological structure at current growth cycle is calculated based on dual-scale automaton and its associated state attributes tables. The generated plant topological structure is composed of metamers. Each metamer has its special geometrical and morphological attributes parameters, such as *kind*, *position*, *size*, *orientation*, *color*, and *normal*. Then the topological structure is encoded and saved into a vertex buffer object (*VBO1*), and transferred to the video memory in graphics card. The geometrical shape of each kind of plant organ is simulated by a normalized mesh model. The organ mesh models are saved as textures and transferred to the geometry shader of GPU for subsequent plant modeling and visualization.

Step2: In GPU vertex shader, each vertex of the plant topological structure is processed in parallel, whose 3D position and shading effects can be transformed to add special effects to the plant in a 3D environment, such as branches swaying in the wind.

Step3: In GPU geometry shader, each primitive of plant topological structure outputted by vertex shader is treated as a metamer. According to its parameter "*kind*", the GPU geometry shader can judge which kind of organ it is, and replace the primitive by the corresponding normalized organ mesh model. Then the selected normalized organ mesh is translated, scaled and rotated according to its *position*, *size* and *orientation* parameters. After each metamer is modeled in parallel, the whole plant geometry is constructed.

Step4: In GPU rasterizer, the plant geometry is converted into a raster image made of pixel, and in the subsequent GPU pixel shader, the color and other attributes of each pixel is computed to add 3D shading and lighting effects to pixels in plant image.

Step5: At the end of the GPU pipeline, the generated plant image is saved into frame buffer, and the simulation of the plant at this growth cycle is finished. Subsequently, the plant topological structure at next growth cycle is computed and saved into another vertex buffer object (*VBO2*). The above procedures are repeated with Ping-Pong technique by using *VBO1* and *VBO2* alternately, until the whole growth process of this plant is modeled and visualized.

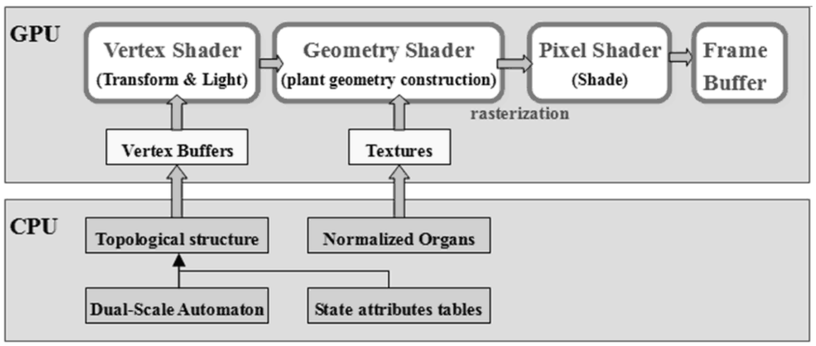


Figure 2: Workflow architecture of our algorithm.

4. Applications

We implemented our algorithm in OpenGL on a PC that has an Intel Xeon 5120 dual-core CPU with 2 GB system memory and a NVIDIA Quadro FX4600 graphics card with 768 MB of video memory.

Figure 3(c) shows a cotton topological structure presented by botanist [7]. Its main stem bears two vegetative branches at nodes 2 and 3. The other upper branches are fruiting branches. In normal conditions, cotton plant undergoes about 22 growth cycles from birth to death. The duration of each growth cycle is depending on environmental conditions.

We can use the automaton shown in Figure 3(a) to generate the cotton topological structure presented in Figure 3(c). This automaton contains four macrostates. Each macrostate consists of only one microstate. The first microstate has no lateral bud and cannot put forth a branch. The second microstate 2 has a lateral bud that can produce vegetative branches, but the lateral bud will be activated after 4 growth cycles. The lateral bud of the third microstate is used to produce fruiting branches from the main stem or vegetative branches. The fourth microstate also has a lateral bud used to simulate the development pattern of fruiting branches. The self-cycling number of each macrostate (respectively 1, 2, 18 and 1 for macrostate 1, 2, 3, and 4) and the self-cycling number of each microstate within macrostate (all of them are set to 1) are labeled in the figure. The transition relations between states are denoted by arrows. With chronological age increasing, the automaton shown in Figure 3(a) can dynamically generate all 22 successive topological structures constructed during the cotton development. Figure 3(b) shows the resulting cotton topological structures in 4th, 6th, 8th and 11th growth cycle, respectively.

We represent the internodes, flowers and bolls of cotton by one kind of mesh model, respectively, while the leaves by three kinds of mesh models for simulating the change of leaves in different growth stages. By integrating with some geometrical parameters of cotton and its organs, the automaton model can visually express cotton growth process and variability. The resulting plant images at all growth cycles make the simulation more understandable to farmer and scientists (Figure 4).

Because the normalized organ mesh models are shared by the plants at all every growth cycles, and only these organ mesh models and the plant topological structure for each growth cycle are transferred to video memory from system memory, our GPU-based algorithm will greatly reduce the amount of transferred data. In this example, all 22 successive topological structures contain altogether about 759 vertices, and the six normalized organ mesh models are composed of about 320 vertices, while the traditional cotton geometry representation is made of vertices and faces needs about 41740 vertices. That means in the case of without compromising image quality, our algorithm reduces the amount of transferred data by a factor of about 53, which depending on the complexity of organ models. Furthermore, plant geometry construction is greatly speeded up due to the parallel replacements of organs in GPU geometry shader. When we use a window size of 1280×800 pixels, the simulation of the cottons at all 22 growth cycles takes only 0.11-0.12 seconds, which meets the need for plant growth modeling and visualization in real time, even for a small population of plants (Figure 5).

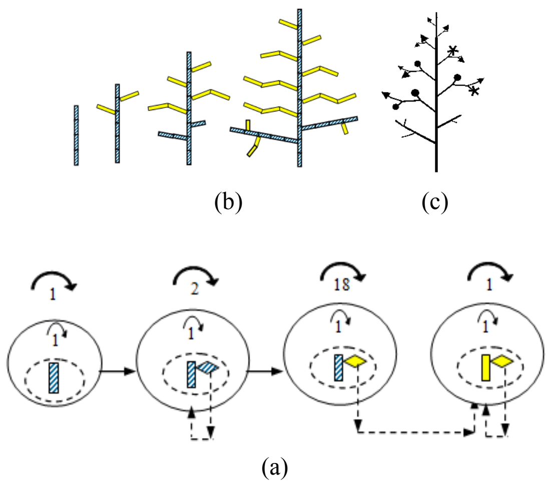


Figure 3: Generation of cotton topological structures: (a) the corresponding automaton, (b) the resulted cotton topological structures, (c) cotton topological structure presented by botanist (from [7]).

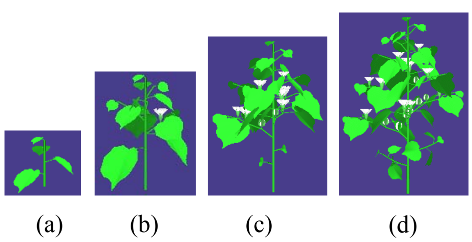


Figure 4: Simulation of cotton development. (a), (b), (c) and (d) The cotton images generated by our algorithm in 4th, 6th, 8th and 11th growth cycle, respectively.

Figure 5: Visualization of a small population of plants using our method, total 133568 vertices, about 0.32s-0.36s.

5. Conclusion

In the work, we have investigated and implemented a GPU-accelerated algorithm for plant growth modeling and visualization. The algorithm dynamically produces plant topological structures by dual-scale automaton, and geometrical structures by the geometry shader of GPU. This scheme greatly reduces the amount of transferred data between main memory and GPU memory. In this algorithm, plant organ geometries are represented by mesh-based models, which are stored directly in the graphics hardware to produce great acceleration in plant modeling and rendering. This method does not lose realism even when the camera is extremely near the plant object. Further work is in progress to apply our algorithm to virtual agriculture.

Acknowledgement

This study is supported by National High-Tech Research and Development Plan of China (863 Plan) under Grant No. 2008AA10Z218.

References

[1] Hu B-G, Zhao X., Yan H-P, De Reffye, Ph. et al. Plant Growth Modelling and Visualization -Review and Perspective. Acta Automatica Sinica. 2001, 27(4): 567-584.

[2] Guo Y. and Li B.-G. New Advances in Virtual Plant Research. Chinese Science Bulletin. 2001, 46(11), 888-894.

[4] Hu B-G, De Reffye Ph, Zhao X, Yan H.-P., Kang M.-Z.“GreenLab: A New Methodology towards Plant Functional-Structual Model-Structual Aspect”, Plant Growth Modeling and Applications (PMA03). 2003, 21-35.

[5] Zhao X, De Reffye Ph., Xiong F-L, Hu B-G, Zhan Z-G. Dual-scale automaton model for virtual plant development. The Chinese Journal of Computers. 2001, 24(6): 608-617.

[6] Teng J, Jaeger M., Hu B-G. A fast ambient occlusion method for real-time plant rendering. Journal of computer science and technology. 2007, 22 (6): 859-866.

[7] Baele X, Warzee N. Real time L-system generated trees based on modern graphics hardware. 2005 International Conference on Shape Modeling and Applications,2005: 184–193.

[8] Jallas E. Improved model-based decision support by modeling cotton variability and using evolutionary algorithms. Thesis PhD of Philosophy, Mississippi State University. USA, 1998.