

基于 GPU 的虚拟植物生长的双尺度自动机模型实现方法

董莹莹^{1,2,3}, 赵星^{3*}, 王纪华^{1,2}

(1. 浙江大学农业遥感与信息技术应用研究所, 杭州 310029; 2. 北京农业信息技术研究中心, 北京 100097;
3. 首都师范大学数学科学学院 检测成像实验室, 北京 100048)

摘要: 为提高双尺度自动机模型绘制植物图形的速度, 给出其基于图形处理器 (GPU) 的并行实现方法。该方法将拓扑结构与归一化植物器官相结合构造三维植物图形: 首先在 GPU 顶点着色器 (vertex shader) 中构造拓扑结构, 然后在 GPU 几何着色器 (geometry shader) 中实现几何造型, 最后将几何结构数据渲染到帧缓存中显示。选取植物学家给出的一株白杨进行基于 GPU 的生长模拟试验。结果显示, 随绘制顶点数的增加, 采用 GPU 进行绘制, 相对 CPU 而言, 其加速比可以从几倍扩大到十几倍以上。该方法可进一步应用于大规模自然场景绘制、虚拟农业等领域。

关键词: 可视化, 模型, 作物, 植物形态结构, 双尺度自动机模型, GPU, 几何着色器

doi: 10.3969/j.issn.1002-6819.2011.05.037

中图分类号: TP391

文献标志码: A

文章编号: 1002-6819(2011)-05-0207-06

董莹莹, 赵星, 王纪华. 基于 GPU 的虚拟植物生长的双尺度自动机模型实现方法[J]. 农业工程学报, 2011, 27(5): 207-212.

Dong Yingying, Zhao Xing, Wang Jihua. Rendering plants with dual-scale automaton model based on GPU[J]. Transactions of the CSAE, 2011, 27(5): 207-212. (in Chinese with English abstract)

0 引言

虚拟植物生长是用计算机形象直观地再现植物的生长过程。在自然景观模拟、生态学、农林学等众多学科中有着广阔的应用前景。目前用于虚拟植物生长的计算机模型有: L-系统 (L-system)^[1]、参考轴技术 (reference axis technique)^[2]、子结构模型^[3]、构件植物模型^[4]、基于模板库的三维植物形态模拟^[5]和双尺度自动机模型 (dual-scale automaton model)^[6]等。其中双尺度自动机模型用微状态和宏状态分别表示植物的叶元和生长单元, 用半马尔可夫链描述各状态之间的关系, 并通过 2 种状态的组合与循环生成拓扑结构。模型依据植物生理年龄归类生长参数, 既简化了生长参数, 又易于与实测数据结合, 且图形化表示令其形象直观、结构简洁、便于理解和使用。

双尺度自动机模型适合用来模拟真实植物的生长过程, 此类模拟是虚拟植物研究的一项重要内容。由于植物形态多样、细节繁杂, 利用计算机绘制其三维图形需要大量图元, 因此限制了真实感植物图形的渲染速度。鉴于植物生长是并行过程, 使用具有并行性的双尺度自

动机模型可以很好地模拟此过程, 且图形处理器 (graphics processing unit, GPU)^[7]适于并行计算, 因此本文选取 GPU 加速模型实现。由于渲染过程是虚拟植物生长中最耗时的部分, 而 GPU 致力于图形渲染, 于是可将其用于绘制显示。

1 模型及 GPU 可编程管线

1.1 双尺度自动机模型

双尺度自动机模型是一种植物形态发生模型^[8], 适于模拟真实植物的生长过程。该模型包含微状态和宏状态 2 种尺度状态, 其中微状态代表叶元, 宏状态代表生长单元, 宏状态由具有相同生理年龄的微状态组合而成。

建模时, 首先根据植物的生长特性确定 2 种尺度状态, 选取生理年龄归类生长参数; 然后采用图形化方法表示生长过程, 并以“微状态属性表”、“宏状态内部属性表”和“宏状态之间关系属性表”作为图形化的参数补充; 最后通过组合与循环 2 种状态生成拓扑结构, 模拟植物生长。此外, 模型还引入复合概率作为参数来描述植物生长过程存在的随机性。

在上述属性表中, “微状态属性表”描述了每种微状态的属性, 其他 2 个属性表则补充描述了宏状态内部各微状态之间的转移矩阵以及宏状态之间的转移矩阵。通过各属性表, 模型将植物形态的拓扑结构、几何结构及生长参数等简洁条理地结合在一起, 完成了植物生长过程的模拟。

1.2 GPU 可编程管线

从 2003 年起, NVIDIA 和 ATI 等公司先后在 GPU 固定图形处理管线基础上, 增加了顶点级和像素级可编程功能, 分别对应 GPU 图形处理管线中的顶点着色器

收稿日期: 2010-07-19 修订日期: 2010-09-20

基金项目: 国家“863”计划项目 (2008AA10Z218SQ, 2007AA10Z235547); 国家自然科学基金资助项目 (60532080); 北京市科技新星资助项目 (2005B49)

作者简介: 董莹莹 (1984—), 女, 山东淄博人, 博士生, 主要从事农业遥感与信息技术, 计算机图形学的研究。杭州 浙江大学农业遥感与信息技术应用研究所, 310029。Email: cici5201@163.com

*通信作者: 赵星 (1974—), 男, 山西太原人, 副研究员, 主要从事计算机图形学, CT 理论与应用技术的研究。北京 首都师范大学数学科学学院 检测成像实验室, 100048。Email: zhaoxing_1999@yahoo.com

(vertex shader)^[9]和片元着色器(fragment shader)^[9],形成了可编程图形处理管线。其中顶点着色器的主要功能是处理顶点数据,例如生成纹理坐标^[10]、计算光照^[10]等;片元着色器的主要功能是处理几何图形数据,例如添加纹理^[10]等。2006年 Shader Model 4.0 (SM4.0)中引入了几何着色器,它位于顶点着色器和片元着色器之间,如图1所示。几何着色器的主要功能是根据顶点数据动态生成、销毁几何图形数据^[11]。

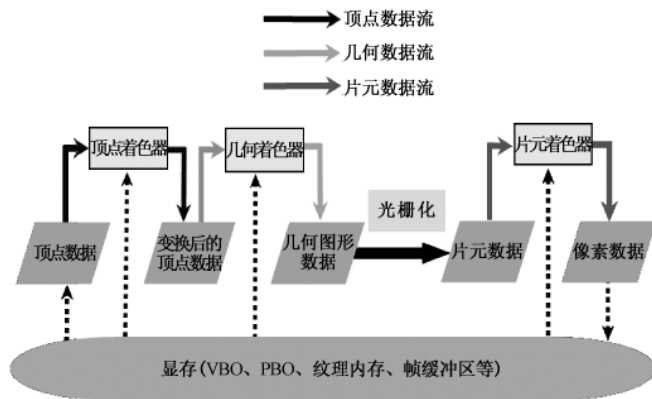


图1 GPU 可编程管线示意图

Fig.1 Schematic diagram of the GPU pipeline

2 基于 GPU 实现的双尺度自动机模型

植物形态的多样性和复杂性限制了计算机对其进行虚拟生长的速度,因此如何快速计算模拟和绘制三维植物图形成为虚拟植物研究的重要问题。考虑到植物生长过程的并行性,以及双尺度自动机模型的并行模式,本文基于 GPU 并行处理器快速实现了模型对于植物生长过程的模拟。

2.1 实现方法

本文给出的模型基于 GPU 的实现方法将拓扑结构与归一化植物器官^[6,12]相结合构造三维植物图形。首先在 GPU 顶点着色器中依据模型构造拓扑结构数据;然后在 GPU 几何着色器中构造并利用拓扑结构数据和几何参数变换归一化植物器官,实现几何造型;最后将几何结构数据渲染到帧缓存中显示。

2.1.1 拓扑结构模拟

基于 GPU 对植物形态拓扑结构进行模拟计算时,首先通过测量和分析被模拟植物的真实形态结构,记录组成植物体的各叶元所包含植物器官的种类、数量和空间位置关系等信息;然后将上述信息分类存储到各属性表,并以二维纹理方式将各属性表传递到 GPU 中;最后在顶点着色器中根据各属性表信息并行计算各顶点属性以得到拓扑结构,并将其存储到(vertex buffer object, VBO)^[9]中。之所以使用二维纹理传递属性表数据,是由于二维纹理不仅可以一次性将 CPU 中的二维数组数据传递到 GPU 中,而且可以在 GPU 中方便地查找属性表中的数据;使用 VBO 存储拓扑结构数据,是由于此类数据需要反复地改动

和使用, VBO 不仅可以实现显示列表的优化功能,而且可以灵活地改动其数据,符合动态生成拓扑结构的需要。

由于植物芽的生长和死亡具有随机性,因此在并行计算各顶点属性时加入了复合概率参数。设 c 是芽在每个生长周期生存着的概率, b 是生存着且能长出新叶元的概率,则在 N 个生长周期内,长出 k 个新叶元的概率如式(1)^[6]所示

$$Pc(x=k) = \sum_{i=k}^{N-1} (1-c)^i C_i^k b^k (1-b)^{i-k} + c^N C_N^k b^k (1-b)^{N-k} \quad (1)$$

式中右边第一项表示从第 k 个周期到第 $N-1$ 个周期长出 k 个新叶元的概率,第二项表示直到第 N 个生长周期时才能长出 k 个新叶元的概率。其中 C_i^k 和 C_N^k 为组合数,分别表示在 i 个和 N 个周期中选取 k 个周期。在实际应用中,用 (N, b, c, S) 4 个参数来确定植物生长所遵从的概率分布曲线,其中参数 S 表示概率分布的偏移。用 GPU 模拟时,上述 4 个参数可作为一致限定变量(Uniform)^[10]传递到 GPU 中,并作为概率公式的自变量参与到 GPU 顶点着色器程序的运行中。

本方法选取一个时钟计数器来控制植物的生长节奏,开始生长时,时钟计数为 0。计数每增加 1,表示已完成对于植物一个生长周期的模拟,此时需重置各顶点属性。随着计数值的增长,各顶点属性的动态更新,可完成植物形态拓扑结构的动态模拟。

2.1.2 几何造型实现

基于 GPU 实现植物几何造型时,首先通过测量和分析被模拟植物各阶段的生长状态,提取组成植物体的各叶元所包含植物器官的生长参数等;然后以生理年龄为标准归类生长参数,并将此类数据添加到“微状态属性表”中;最后在几何着色器中构造归一化植物器官,并依据拓扑结构调用其三维图形,通过几何参数对其进行仿射变换^[13],最终实现几何造型。之所以引入“归一化植物器官”,是由于大多数植物在其生长过程中,各器官的尺寸、颜色变化明显,而形状变化不大。

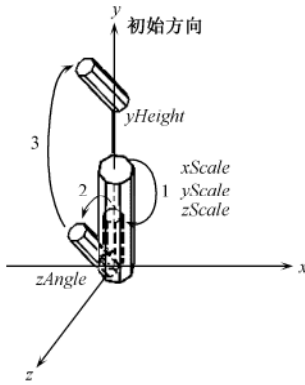
现以归一化枝条(本方法选取八棱柱模拟枝条)为例讲述植物器官三维变换,如图 2 所示,其中数字 1 代表缩放变换,数字 2 代表旋转变换,数字 3 代表平移变换。图 2 中 $xScale$ 、 $yScale$ 、 $zScale$ 为枝条相对其初始尺寸(即单位尺寸)在 x 、 y 、 z 轴的缩放比例; $zAngle$ 为枝条相对其初始方向(即 y 轴正方向)绕 z 轴的旋转角; $yHeight$ 为枝条相对其初始位置(即以原点为起点、以初始方向为正方向的位置)沿 y 轴的平移距离。

在三维欧氏空间中,任意互为相似形的 2 个物体可以通过上述三类变换组合得到的仿射变换进行转换。本文选用的归一化植物器官其上任意一点的仿射变换矩阵可如下表述(以归一化枝条为例)。

设 $Matrix$ 为枝条的仿射变换矩阵,则 $Matrix$ 可由式(2)计算

$$Matrix = MatrixTranslation \times MatrixRotation \times MatrixScale \quad (2)$$

式中,



注：数字 1 代表缩放变换，数字 2 代表旋转变换，数字 3 代表平移变换； $xScale$ 、 $yScale$ 、 $zScale$ ：枝条相对其初始尺寸（即单位尺寸）在 x 、 y 、 z 轴的缩放比例； $zAngle$ ：枝条相对其初始方向（即 y 轴正方向）绕 z 轴的旋转角； $yHeight$ ：枝条相对其初始位置（即以原点为起点、以初始方向为正方向的位置）沿 y 轴的平移距离。

图 2 归一化枝条的三维变换示意图
Fig.2 3D transform of normalized branch

$$MatrixTranslation = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & yHeight \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$MatrixRotation = \begin{bmatrix} \cos(zAngle) & -\sin(zAngle) & 0 & 0 \\ \sin(zAngle) & \cos(zAngle) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$MatrixScale = \begin{bmatrix} xScale & 0 & 0 & 0 \\ 0 & yScale & 0 & 0 \\ 0 & 0 & zScale & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

枝条上任一点坐标 $OriPosition=[x,y,z,1]$ 经上述仿射变换后的坐标 $NowPosition$ 为

$$NowPosition = OriPosition \times Matrix \quad (3)$$

用 GPU 模拟时，式 (2) 所计算的仿射变换矩阵可通过 GPU 可编程管线中的着色器程序运行实现，其所需三类变换中的数据，可作为参数传递到程序中。

归一化植物器官的引入将植物几何造型的实现过程归结为对同类相似形植物器官进行仿射变换的过程，该变换过程简洁且易于编程实现。

2.2 实现步骤

双尺度自动机模型基于 GPU 实现方法的流程见图 3，其具体步骤如下。

步骤 1：在 CPU 应用程序中，创建一个时钟计数器控制植物的生长节奏，并将其初值置 0；创建 2 个 VBO 用于植物形态拓扑结构的动态生成，分别记为 VBO0 和 VBO1，使用 VBO0 将植物形态拓扑结构的初始信息传递到 GPU 中；使用属性表数据纹理将存储属性表的二维数组传递到 GPU 中。

步骤 2：在 GPU 顶点着色器程序中，根据各属性表数据纹理计算各顶点属性，以得到拓扑结构，并将其存储在 VBO1 中；计算各顶点光照及各属性表数据纹理坐

标，以用于真实感绘制。在此，使用蒙特卡罗方法模拟 (N, b, c, S) 所确定的概率分布，生成具有随机性的拓扑结构，其中 N 、 b 、 c 、 S 作为一致限定变量传递到 GPU 中。

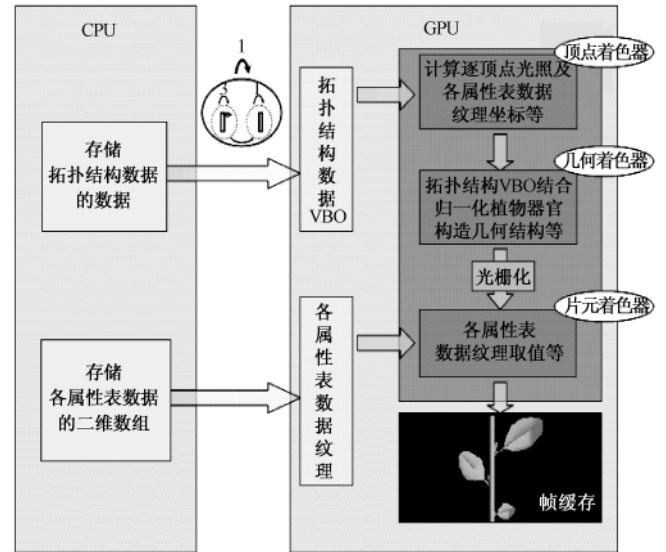


图 3 基于 GPU 的双尺度自动机模型实现方法流程图
Fig.3 Rendering plants with dual-scale automaton model based on GPU

步骤 3：在 GPU 几何着色器程序中，构造归一化植物器官（例如枝条、叶、花、果实）子函数，并根据拓扑结构 VBO1 和各属性表数据纹理对各归一化植物器官进行相应地仿射变换，以得到全部的植物器官，进而实现几何造型。

步骤 4：在 GPU 片元着色器程序中，根据各顶点的各属性表数据纹理坐标，对各属性表数据纹理进行取值，并将采样得到的纹理数据映射到植物的几何图形上。至此，得到了植物形态的几何结构数据，可将其写入到帧缓存中显示。

步骤 5：在 CPU 应用程序中，时钟计数增加 1，若达到时钟上限则停止，否则将 VBO1 中的数据拷贝到 VBO0 中并返回步骤 2。

在上述 5 个步骤中，随着时钟计数的增长，动态生成了三维植物图形，完成了整个植物生长过程的模拟。其中对 2 个 VBO 的操作采用了乒乓技术 (ping-pong technique)^[9]，即将每一次的渲染输出转换成为下一次运算的输入，充分提高了运算效率。

2.3 模拟试验

根据上述基于 GPU 的双尺度自动机模型实现方法，本文选取植物学家 Godin 在 1997 年给出的一株生长了 2 a 的白杨形态^[14-15]进行模拟试验，此白杨的形态结构如图 4a 所示。

利用 GPU 模拟白杨的生长过程时，首先建立双尺度自动机模型，如图 4b 所示，根据组成植物体的各叶元所包含植物器官的种类、数量、形态结构特征、空间位置和彼此之间结构关系等数据，分别构造相应的属性表，

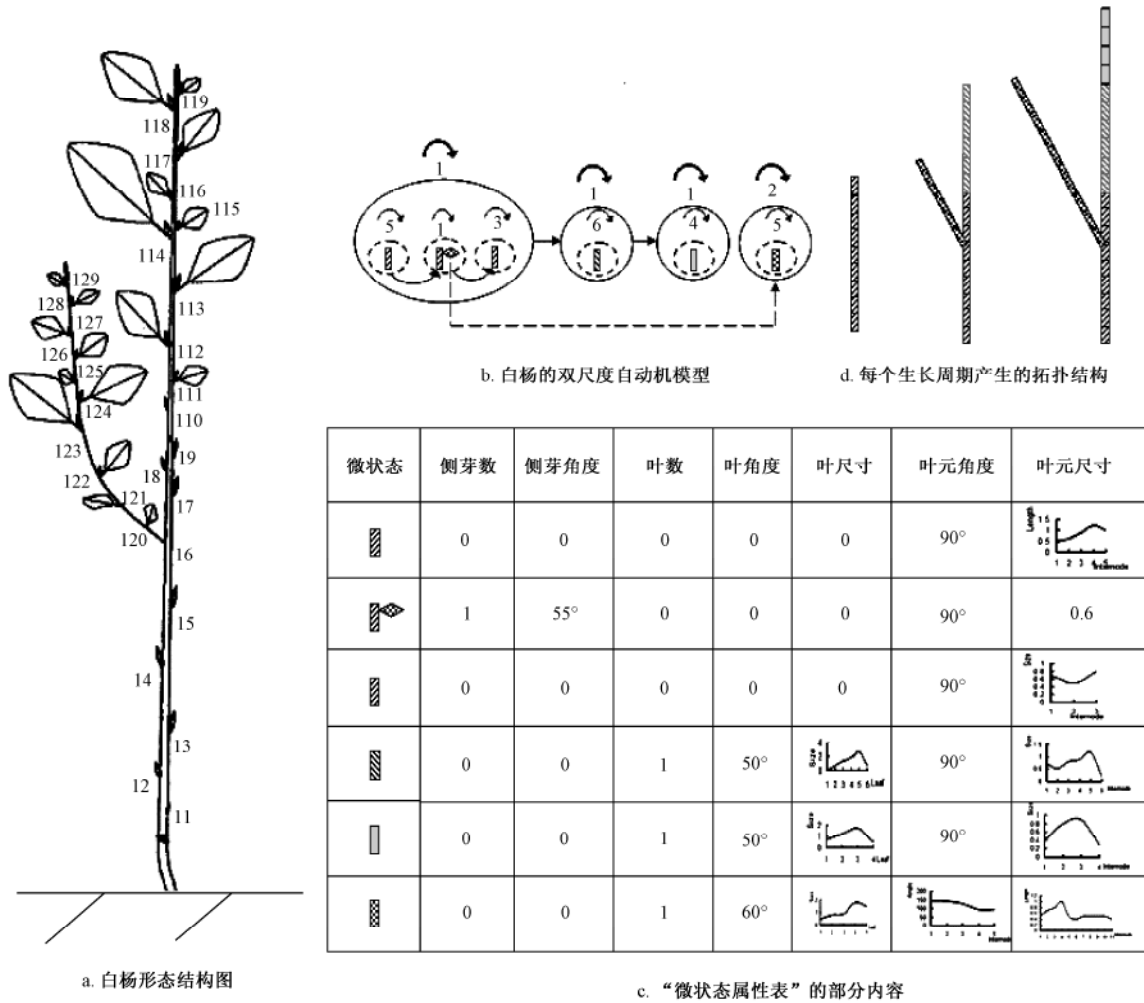


图 4 双尺度自动机模型及“微状态属性表”的部分内容
Fig.4 Dual-scale automaton model and its microstate attribute table

并以二维纹理方式将各属性表传递到 GPU 中。其中“微状态属性表”的部分内容如图 4c 所示，表中曲线为叶元生长过程中的相关概率分布曲线，由于植物在其生长过程中受到外界环境等因素的影响，因此其生长过程满足一定的概率分布；然后根据各属性表动态地生成拓扑结构 VBO，图 4d 从左到右分别为植物在第 1、2、3 个生长周期后的拓扑结构；最后使用拓扑结构 VBO 和各属性表数据纹理对归一化植物器官（如图 5 所示）进行仿射变换及组合，动态地生成三维植物图形。

直接对图形硬件进行编程控制，是为 GPU 编程设计的高级绘制语言。算法测试的计算机配置为 Intel Xeon 5120 (1.83 GHz)，4GB DDR2 内存，NVIDIA Quadro FX4600 显卡 (768 MB 显存，核心频率 500 MHz)，试验结果如图 7，其中 GPU 绘制时间包含了数据传输时间。绘制图像大小为 512 (H) × 512 (V) 像素，当顶点数目从 1024 增加到 1024² 时，GPU 绘制相对 CPU 绘制而言，其加速



图 5 归一化植物器官模拟图
Fig.5 Normalized plant organs

应用双尺度自动机模型基于 GPU 实现方法的模拟试验结果如图 6b 所示。本绘制试验采用 Visual Studio.Net 2003 作为开发平台，在 C++代码框架下结合 CG 语言实现，其中 CG 即 C for Graphics，由 NVIDIA 公司开发，

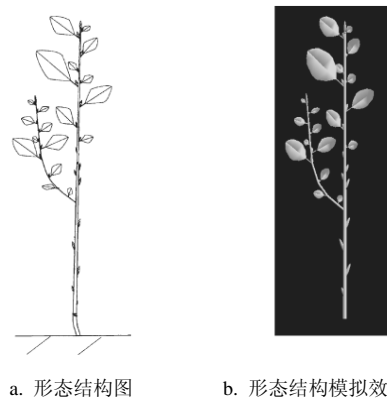


图 6 白杨的形态结构图及其模拟效果图
Fig.6 Morphological structure and its GPU-based rendering result of a poplar tree

比从 3.8605 扩大到 12.3434, 即顶点数目增多时, CPU 时间/GPU 时间在增大, 且增长幅度随顶点数目的增多而变大。结果充分验证了模型基于 GPU 实现方法的可行性和有效性。综上可知, 该方法既能确保绘制效果, 又能大幅提高渲染速度。

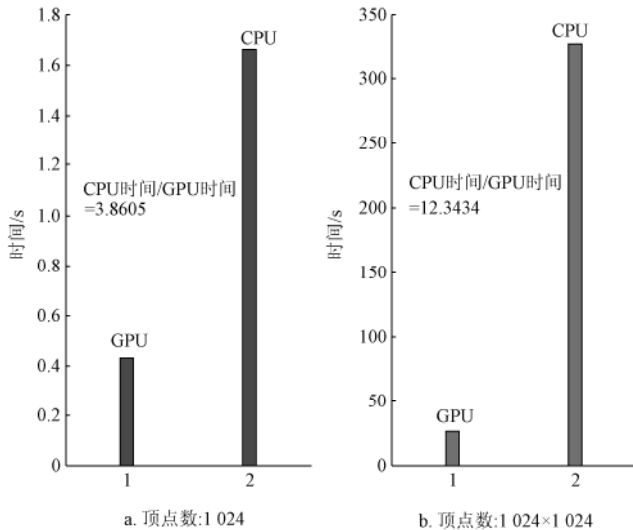


图7 GPU与CPU绘制时间柱状图

Fig.7 Rendering time of GPU and CPU

3 结论与讨论

本文采用基于 GPU 并行渲染的方式, 快速实现了双尺度自动机模型。

1) 植物生长是并行过程, 双尺度自动机模型适合模拟此过程, 现有的模型实现方法是基于 CPU 的, 为串行过程, 主要使用函数递归调用模拟植物生长, 没有体现出模型的并行模拟优势。考虑到 GPU 致力于图形渲染、适于并行计算, 本文提出了模型基于 GPU 的实现方法。该方法充分利用了模型的并行模拟原理和 GPU 的并行渲染优势, 实现了植物生长过程的并行模拟。

2) 该方法充分利用 GPU 可编程管线的数据并行处理功能及最新的几何着色器功能, 快速并行地实现了植物生长过程中耗时较多的几何造型部分, 大大提高了三维植物图形的绘制速度。

本文提出的模型基于 GPU 的并行实现, 不仅可以用于植物的生长过程模拟, 而且可以在进一步的研究中, 将该模型与植物力学模型、碰撞检测算法和植物向光性模型相结合, 探讨植物在风中摇曳、植物与障碍物之间的碰撞检测、枝条之间的躲让效应 (shyness effect)、植物的向光性生长等的 GPU 实现问题。

[参 考 文 献]

[1] Lindenmayer A. Mathematical models for cellular interaction in development[J]. Parts I and II. *Journal of Theoretical Biology*, 1968, 18(3): 280—299.
 [2] de Reffye Philippe, Edelin C, Françon J, et al. Plant models faithful to botanical structure and development[J]. *Computer*

Graphics, 1988, 22(4): 151—158.

- [3] Yan Hongping, de Reffye Philippe, Pan Chunhong, et al. Fast construction of plant architectural models based on substructure decomposition[J]. *Journal of Computer Science and Technology*, 2003, 18(6): 780—787.
 [4] 丁维龙, 熊范纶, 张友华. 基于构件的植物三维结构模拟模型[J]. *小型微型计算机系统*, 2004, 25(9): 1624—1627. Ding Weilong, Xiong Fanlun, Zhang Youhua. Model for 3D simulation of plant structure based on component[J]. *Mini-Microsystems*, 2004, 25(9): 1624—1627. (in Chinese with English abstract)
 [5] 丁维龙, 刘端阳, 古辉, 等. 基于模板库的三维植物形态模拟系统研究[J]. *计算机工程*, 2006, 32(9): 178—180. Ding Weilong, Liu Duanyang, Gu Hui, et al. Research of three-dimension plant-image simulation system based on templates library[J]. *Computer Engineering*, 2006, 32(9): 178—180. (in Chinese with English abstract)
 [6] 赵星, de Reffye Philippe, 熊范纶, 等. 虚拟植物生长的双尺度自动机模型[J]. *计算机学报*, 2001, 24(6): 608—615. Zhao Xing, de Reffye Philippe, Xiong Fanlun, et al. Dual-scale automaton model for virtual plant development[J]. *Chinese Journal of Computers*, 2001, 24(6): 608—615. (in Chinese with English abstract)
 [7] Fernando R. GPU Gems: Programming techniques, tips, and tricks for real-time graphics[M]. San Francisco: Addison-Wesley Professional, 2004.
 [8] 胡包钢, 赵星, 严红平, 等. 植物生长建模与可视化——回顾与展望[J]. *自动化学报*, 2001, 27(6): 816—835. Hu Baogang, Zhao Xing, Yan Hongping, et al. Plant growth modeling and visualization review and perspective[J]. *Acta Automatica Sinica*, 2001, 27(6): 816—835. (in Chinese with English abstract)
 [9] Pharr M, Fernando R. GPU Gems 2: Programming techniques for high-performance graphics and general-purpose computation[M]. San Francisco: Addison-Wesley Professional, 2005.
 [10] Rost R. OpenGL Shading Language[M]. San Francisco: Addison-Wesley Professional, 2004.
 [11] Nguyen H. GPU Gems 3[M]. San Francisco: Addison-Wesley Professional, 2007.
 [12] 康孟珍, de Reffye Philippe, 胡包钢, 等. 快速构造植物几何结构的子结构算法[J]. *中国图象图形学报*, 2004, 9(1): 79—86. Kang Mengzhen, de Reffye Philippe, Hu Baogang, et al. Fast construction of geometrical structure of plant with substructures algorithm[J]. *Journal of Image and Graphics*, 2004, 9(1): 79—86. (in Chinese with English abstract)
 [13] Shreiner D, Woo M, Neider J, et al. OpenGL Programming Guide: The official guide to learning opengl.5th Edition[M]. San Francisco: Addison-Wesley Professional, 2005.
 [14] Godin C, Costes E, Caraglio Y. Exploring plant topological structure with the AMAPmod software:an outline[J]. *Silva Fennica*, 1997, 31(3):357—368.
 [15] Godin C, Caraglio Y. A multiscale model of topological structures [J]. *Journal of Theoretical Biology*, 1998, 191(1): 1—46.

Rendering plants with dual-scale automaton model based on GPU

Dong Yingying^{1,2,3}, Zhao Xing^{3*}, Wang Jihua^{1,2}

(1. *Institute of Agricultural Remote Sensing and Information Technology, Zhejiang University, Hangzhou, 310029, China;*

2. *Beijing Research Center for Information Technology in Agriculture, Beijing, 100097, China;*

3. *Laboratory of Computer Tomography, School of Mathematical Sciences, Capital Normal University, Beijing, 100048, China)*

Abstract: In order to accelerate plants rendering of dual-scale automaton model, a method based on GPU was proposed for model realization. In this method, the 3D structure of plant was constructed by combining with topological structure and normalized plant organs. Firstly, topological structure was constructed in vertex shader, then geometrical structure was built in geometry shader, and finally geometrical structure was transferred to frame buffer for displaying. The GPU-based method was tested by rendering a poplar tree. In the experiment, as the point number multiplied, the speed-up ratio of GPU rendering relative to CPU rendering could increase several-fold and even ten times. The results not only adequately showed the effectiveness of this GPU-based rendering method, but also fully represented its remarkable improvement in computation. Moreover, rendering plants with dual-scale automaton model based on GPU can be applied to natural scenery rendering, virtual agriculture, computer-aided design etc, which involves enormous data.

Key words: visualization, model, crops, plant morphological structure, dual-scale automaton model, GPU, geometry shader