

УДК 629.4.016.12

С.И. Юрков

## ОСОБЕННОСТИ ТРЕХМЕРНОГО КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ РАСТЕНИЙ

**1. Актуальность задачи трехмерного моделирования растений.** Трехмерные модели растений применяются в интерактивных компьютерных программах, тренажерах, а также при создании спецэффектов для фильмов.

Проблема создания изображений растений состоит из задач по созданию и улучшению:

1. Моделей роста и построения геометрической структуры растений (в т.ч. различных моделей взаимодействия растений, модели тропностей [21]).

2. Моделей распределения листьев на ветках и стеблях [22].

3. Алгоритмов визуализации (в т.ч. способов ускорения визуализации, увеличения реалистичности, построения теней и освещения).

Существующие модели построения делятся на две группы: ориентированные на биологическую структурную реалистичность (с использованием Л-систем [1] и их модификаций) и ориентированные на скорость построения и отображения (эвристические методы).

**1.1. Л-системы.** Методы, которые основаны на применении Л-систем, получили наиболее широкое развитие в работах группы под руководством Прусинкевича.

ДОЛ-система (детерминированная Л-система, правила контекстно-независимы) – упорядоченная тройка  $\langle V, w, P \rangle$ , где  $V$  – алфавит системы,  $w$  – аксиома,  $P$  – набор правил. Основное отличие – параллельный вывод, когда правила применяются не последовательно, а параллельно.

В статье Прусинкевича, «Графическое применение Л-систем» [1], показывается возможность использования Л-систем для построения трехмерных моделей растений в два шага. Первый –

---

© Юрков С.И., 2009

построение слова с использованием Д0Л-системы, второй – графическая интерпретация полученного слова с использованием конечного автомата магазинного типа, где буквы слова являются командами автомату, «черепашке», которая выполняет построение модели. Состояние «черепашки» – это позиция и ориентация в пространстве. Примеры команд: повернуть на  $Z$  градусов, пройти вперед на расстояние  $D$ . Другие примеры использования Л-систем содержатся в [2, 3].

В статье [4] показана возможность использования Л-систем для симуляции передачи сигналов, от корня и к корню растения, аналогично тому, как это происходит в действительности.

В книге «The Algorithmic Beauty of Plants» [5] авторы доказывают возможность использования стохастических, контекстно-зависимых, параметрических, временных Л-систем. Обновленная версия этого же материала появляется позже в [6, 7]. Вводится понятие суб-Л-системы [8].

0Л-система (стохастическая Л-система, правила контекстно-независимы) – упорядоченная четверка  $\langle V, w, P, p_i \rangle$ , где  $V, w$  и  $P$  определены так же, как и для Д0Л-систем, а функция  $p_i : P \rightarrow (0; 1]$  отображает набор правил в набор вероятностей применения правил. Сумма вероятностей для правил с одинаковым символом предшественником равна 1.

К контекстно-зависимым, или  $(k, l)$ -Л-системам, относятся системы с правилами вывода вида  $wl\langle a \rangle wr \rightarrow x$ , где буква  $a$  (именуемая строгим предшественником) может породить слово  $x$  только тогда, когда слева от  $a$  находится слово  $wl$ , а справа –  $wr$ .  $k$  и  $l$  определяют длину левого и правого слова соответственно.

Параметрические Л-системы образуются за счет добавления в правила условий и параметров. Теперь каждому символу алфавита могут быть назначены параметры. Например, правило  $A(t) : t > 5 \rightarrow B(t + 1)CD(\sqrt{t}, t - 2)$  может быть применено к слову  $A(16)$ , поскольку  $16 > 5$ . Результатом будет слово:  $B(17)CD(4, 14)$ .

Обзор некоторых моделей морфогенеза приводится в статье [9]. Из рассмотренных моделей следует отметить воксельный автомат, пространственно ориентированную модель, которая дает впечатляющие результаты при моделировании корней дерева.

В статье «Synthetic Topiary» [10] дается обзор существующих методов моделирования растений с учетом эндогенных механизмов. Предлагается расширение Л-систем, с тем, чтобы с их помощью можно было моделировать взаимодействие между растением и окружающей средой. Развитие этой идеи изложено в [11].

В [12] показана методология построения Л-системы по данным наблюдения за растением на примере *Fraxinus pennsylvanica* (green ash).

В программе, по которой подготовлена статья [13], впервые в работах группы выполнено создание изображения экосистемы. Выдвинуты базовые идеи по увеличению быстродействия: использование экземпляров и уровней детализации.

В [14] показана возможность изменения некоторых параметров модели без полной ее перестройки.

Практическая проверка теорий группы проводится в программах `srfg` и `vlab`, которые описаны в [15, 16].

**1.2. Методы моделирования.** В [17] изложен процесс моделирования клена. Показывается метод моделирования сочленений веток. Данные о производительности метода не приведены.

**1.3. Методы визуализации.** В [19] исследованы характеристики прозрачности, отражения и другие свойства листьев с целью построения алгоритма их реалистичной визуализации. В [20] предлагается алгоритм непрерывного изменения детализации, наиболее ценная часть которого – построение «нечестных» изображений растений на горизонте.

**1.4. Ускорение визуализации.** В [18] предложен алгоритм визуализации лесных массивов, основанный на предварительной оценке затрат на визуализацию каждой модели. Благодаря такой оценке удается поддерживать приемлемое количество кадров в секунду независимо от количества растений в кадре.

Проведенный анализ литературы показал, что проблема повышения эффективности трехмерного моделирования растений является многогранной и актуальной. В представленной статье предложен новый метод увеличения реалистичности визуализации трехмерных изображений растений без снижения эффективности.

**2. Ускоренная визуализация большого количества деревьев без billboard-эффекта.** Проблема отображения листьев дерева состоит в

том, что в настоящее время не существует таких ресурсов графического процессора, которые позволят отображать высокодетализированные полигональные сетки (ПС) в большом количестве в реальном времени.

Чтобы обойти это ограничение применяются спрайты (sprites) – плоские четырехугольники со специальной текстурой, которые всегда расположены перпендикулярно линии взгляда [23, 24, 25, 26]. При их использовании становится заметна регулярность. Для устранения этого недостатка применяют усовершенствованные спрайты, нормаль к которым отклонена от линии взгляда на случайный, но постоянный во времени угол (рис. 1).

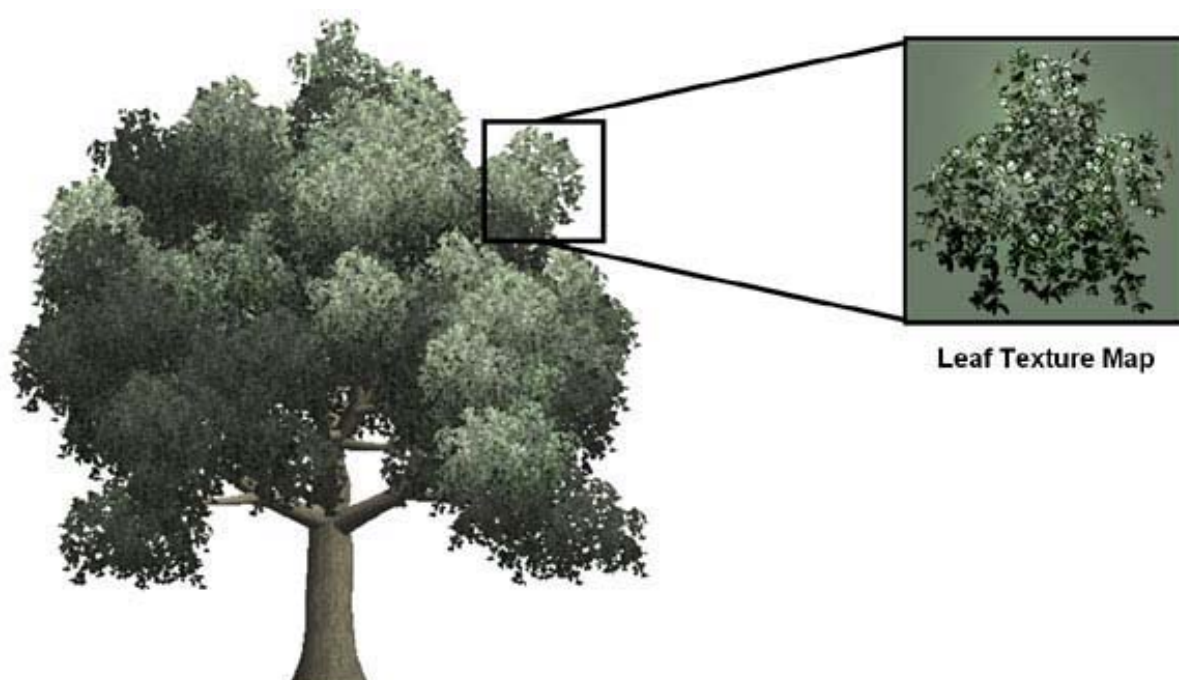


Рисунок 1 - Усовершенствованные спрайты в системе SpeedTree [27]

Недостаток вышеизложенных методов (будем называть его billboard-эффект) состоит в следующем: при «облете» дерева становится заметно, что листья отображаются при помощи спрайтов, поскольку при повороте изображение пучка листьев не изменяется.

Далее предлагается модификация последнего метода: визуализация с помощью спрайтов с текстурой, которая зависит от направления взгляда. Будем создавать такую текстуру с помощью проекции изображения ПС на сферу. В дальнейшем эта сфера отображается вместо ПС с использованием маски, полученной с помощью alpha-смешивания комбинации масок, также полученных

из ПС посредством кубической проекции. При использовании предложенного метода необходимо увеличивать количество треугольников на один спрайт. Это не повлияет на степень заполнения пропускного канала и объем памяти, занимаемый данными о спрайтах, на новых видеокартах с поддержкой Shader Model 4.0. Естественно, увеличивается нагрузка на вершинный шейдер, однако, за счет оптимизации площади спрайтов, уменьшится количество alpha-тестов и, теоретически, возможно увеличение быстродействия.

Результат моделирования – текстура  $C = C_{ij}$  размера  $M \times N$  и карта глубины  $D = D_{ij}$  аналогичного размера. Модель, для которой необходимо построить  $C$  и  $D$ , задана текстурой  $C1 = C1_{ij}$  размера  $M1 \times N1$  и набором треугольников  $M = \{T_k(I_1, I_2, I_3)\}$ . Каждый треугольник – это три индекса в наборе вершин  $V = \{V'_k(x', y', z', u, v)\}$ , где  $x', y', z'$  – координаты в пространстве, а  $u$  и  $v$  – текстурные координаты.

Найдем сферу минимального объема  $S(x'_s, y'_s, z'_s, r)$ , которая бы заключала в себе все вершины  $V$  [D3DXComputeBoundingSphere в 28]. Для сокращения записи перейдем в новую систему координат, где  $S$  будет находиться в центре координат и иметь радиус равный единице. В новой системе координат:

$$V = \{V_k(x, y, z, u, v)\} = \left\{ V'_k \left( \frac{x' - x_s}{r}, \frac{y' - y_s}{r}, \frac{z' - z_s}{r} \right) \right\}$$

Для каждого пикселя  $C_{i,j}$  найдем координаты соответствующей ему точки в сферической с.к.  $\alpha$  (0;  $\pi$ ) и  $\beta$  (0;  $2\pi$ ), используя проекцию Меркатора:

$$\alpha = (2i+1) * \pi / (2M)$$

$$\beta = (2j+1) * \pi / N$$

В декартовой с.к.:

$$R_x = \sin(\alpha)\sin(\beta)$$

$$R_y = \cos(\alpha)$$

$$R_z = \sin(\alpha)\cos(\beta)$$

Проведем луч, направленный к центру сферы. Начало луча  $R_o(R_x, R_y, R_z)$ , направление луча  $R_d(-R_x, -R_y, -R_z)$ . Пройдем по всем треугольникам сетки и из точек, которые пересек луч [D3DXIntersectTri в 28], выберем ближайшую к точке  $R_o$ , используя карту глубины  $D$ . Для найденной точки известны ее

барицентрические координаты  $u_1$ ,  $u_2$  и вершины треугольника  $V_1$ ,  $V_2$ ,  $V_3$ , к которому она принадлежит. Умножив барицентрические координаты на текстурные координаты вершин треугольника, найдем текстурные координаты точки:

$$u = u_1 * u\{V_1\} + u_2 * u\{V_2\} + (1 - u_1 - u_2) * u\{V_3\}$$

$$v = u_1 * v\{V_1\} + u_2 * v\{V_2\} + (1 - u_1 - u_2) * v\{V_3\}$$

Установим цвет пикселя  $C_{ij}$  равным цвету на текстуре  $C_1$  в точке с координатами  $u$ ,  $v$ . Для приемлемого результата должно быть достаточно использования билинейной фильтрации. Если луч не пересек ни одного треугольника полигональной сетки, то установим  $\alpha$ -компоненту пикселя  $C_{ij}$  равной нулю (абсолютно прозрачный).

Алгоритмы, которые здесь не описаны, не влияют на производительность отображения в конечной программе, их реализация содержится в [28].

**3. Выводы.** Эксперименты с программой, разработанной по предложенному методу, подтвердили целесообразность ее использования для устранения billboard-эффекта.

## ЛИТЕРАТУРА

1. P. Prusinkiewicz: Graphical applications of L-systems. Proceedings of Graphics Interface '86 / Vision Interface '86, pp. 247-253.
2. P. Prusinkiewicz, L. Kari: Subapical bracketed L-systems. Grammars and their Application to Computer Science, volume 1073 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, 1996, pp. 550-564.
3. P. Prusinkiewicz: Modeling of spatial structure and development of plants. Scientia Horticulturae vol. 74, pp. 113-149.
4. P. Prusinkiewicz, A. Lindenmayer, J. Hanan: Developmental Models of Herbaceous Plants for Computer Imagery Purposes. Computer Graphics, Volume 22, Number 4, August 1988, pp. 141-150.
5. P. Prusinkiewicz, A. Lindenmayer: The Algorithmic Beauty of Plants. Springer-Verlag, New York, 1990.
6. Prusinkiewicz, Hammel, Hannan, Mech: Visual models of plant development. Handbook of formal languages. Springer-Verlag, 1996.
7. Prusinkiewicz, Hammel, Hannan, Mech: The Artificial Life of Plants. Artificial life for graphics, animation, and virtual reality, volume 7 of SIGGRAPH'95 Course Notes, pages 1-1-1-38. ACM Press, 1995.

8. J. Hanan: Parametric l-systems and their application to the modeling and visualization of plants. University of Regina, 1992.
9. P. Prusinkiewicz: Modeling and Visualization of Biological Structures. Proceeding of Graphics Interface'93, pages 128-137, May 1993.
10. P. Prusinkiewicz, M. James, and R. Mech: Synthetic Topiary. Proceedings of SIGGRAPH 94. ACM SIGGRAPH, pp. 351-358.
11. Radomir Mech and Przemyslaw Prusinkiewicz. Visual Models of Plants Interacting with Their Environment. Proceedings of SIGGRAPH 96 (New Orleans, Louisiana, August 4-9, 1996). In Computer Graphics Proceedings, Annual Conference Series, 1996, ACM SIGGRAPH, pp. 397-410.
12. Hammel, Prusinkiewicz, Remphrey, Davidson: Simulating the development of *Fraxinus pennsylvanica* shoots using L-systems. Proceedings of the Sixth Western Computer Graphics Symposium, pages 49-58, March 1995.
13. Oliver Deussen, Pat Hanrahan, Bernd Lintermann, Radomir Mech, Matt Pharr, and Przemyslaw Prusinkiewicz. Realistic modeling and rendering of plant ecosystems. Proceedings of SIGGRAPH'98 (Orlando, Florida, July 19-24, 1998). In Computer Graphics Proceedings, Annual Conference Series, 1998, ACM SIGGRAPH, pp. 275-286.
14. Joanna L. Power, A. J. Bernheim Brush, Przemyslaw Prusinkiewicz, and David H. Salesin: Interactive arrangement of botanical L-system models. In Proceedings of the 1999 Symposium on Interactive 3D Graphics, pp. 175-182 and 234.
15. Przemyslaw Prusinkiewicz, Jim Hanan and Radomir Mech. An L-system-based plant modeling language. In: M. Nagl, A. Schuerr and M. Muench (Eds): Applications of graph transformations with industrial relevance. Proceedings of the International workshop AGTIVE'99, Kerkrade, The Netherlands, September 1999. Lecture Notes in Computer Science 1779, Springer, Berlin, 2000, pp.395-410.
16. P. Federl and P. Prusinkiewicz: Virtual Laboratory: an Interactive Software Environment for Computer Graphics. In Proceedings of Computer Graphics International 1999, pp. 93-100.
17. Bloomenthal: Modeling the Mighty Maple. SIGGRAPH'85 Proceedings.

18. Hernandez, Benes: Robin Hood's Algorithm for Time-Critical Level of Detail. Graphicon'05 Proceedings.
19. Gladimir V. G. Baranoski Jon G. Rokne: A Simplified Model for Light Interaction with Plant Tissue. Graphicon'98 proceedings.
20. Н. А. Елыков, И. В. Белого, С.А. Кузиковский: Метод непрерывной детализации трехмерных моделей растительности. Материалы Графикон'06.
21. Tropism  
<http://en.wikipedia.org/wiki/Tropism>
22. Phyllotaxis  
<http://en.wikipedia.org/wiki/Phyllotaxis>
23. Ворсин: Визуализация лесных массивов в реальном времени.  
<http://www.gamedev.ru/articles/?id=30125>
24. Decaudin, Neyret, Cohen: Rendering Forest Scenes in Real-Time.  
<http://www.antisphere.com/Research/Forest.php>
25. Geometric modelling of plants.  
<http://www.vterrain.org/Plants/Modelling/index.html>
26. Billboards  
<http://www.opengl.org/resources/code/samples/advanced/advanced97/notes/node31.html>
27. Speed Tree Help System  
<http://www.speedtree.com/>
28. Microsoft DirectX Software Development Kit.  
<http://msdn.microsoft.com/en-us/directx/default.aspx>

Одержано 29.11.2008р.