А.П. Остроушко, Н.В. Белоус

Харьковский национальный университет радиоэлектроники

osa@kture.kharkov.ua, belous@kture.kharkov.ua

БЫСТРЫЙ АЛГОРИТМ ВИЗУАЛИЗАЦИИ ОБЛАЧНОГО СЛОЯ ДЛЯ МЕТОДА ОБРАТНОГО ТРАССИРОВАНИЯ

Введение

Одним из требований к современным системам трёхмерной компьютерной графики является высокая реалистичность синтезируемых изображений, которая достигается за счёт увеличения детализации изображения, формирования статических и динамических теней от источников света, учёта прозрачности среды сцены, генерации различных спецэффектов. Кроме того, ключевую роль в качестве формируемых изображений играет метод синтеза. Наиболее перспективным можно назвать метод обратного трассирования лучей, обеспечивающий возможность получения исключительно высокореалистичных изображений. Главным недостатком этого метода является его высокая ресурсоёмкость. Именно поэтому разработка эффективных моделей элементов трёхмерных сцен и алгоритмов их визуализации для метода обратного трассирования является актуальной задачей.

В статье предложен быстрый алгоритм, позволяющий вести синтез изображения облачного слоя, описанного при помощи модели, предложенной в [1]. Алгоритм использует классификацию проекционных лучей для исключения из обработки лучей, не имеющих пересечения с облаком. Это позволяет значительно сократить объем вычислений и уменьшить время синтеза изображения. Возможность широкого распараллеливания вычислений позволяет использовать предложенный алгоритм в многопроцессорных, распределённых, GRID системах и кластерах.

Особо следует выделить возможность реализации алгоритма для работы на современных архитектурах GPU, благодаря малому объёму

требуемой памяти для хранения модели облака и промежуточных результатов вычислений.

Благодаря большой гибкости модели облака и высокой скорости работы алгоритма его визуализации результаты работы могут быть использованы при построении систем визуализации реального времени.

В настоящее время основным подходом при реализации метода обратного трассирования является решение системы уравнений проекционного луча, заданного в параметрической форме, и уравнения поверхности в неявном виде [2]. Классический алгоритм определения точки пересечения проекционных лучей со сферами, составляющими структуру облака, позволяет существенно сократить объем вычислений [3].

Однако, как видно из результатов, приведённых в [1], время синтеза одного кадра по-прежнему достаточно велико и это не даёт возможности использовать классический алгоритм для работы в реальном времени. Поэтому требуется разработка быстрого алгоритма для метода обратного трассирования, обеспечивающего поиск параметров пересечения проекционного луча с элементами облака.

Описание алгоритма

Классический алгоритм позволяет вести независимые расчёты для каждого проекционного луча, что даёт возможность широкого распараллеливания процесса вычислений [4]. Повысить эффективность и значительно сократить время синтеза можно за счёт использования классификации лучей [5].

Предлагается модифицировать алгоритм, приведённый в [5], для нахождения параметров пересечения проекционных лучей со сферами облака.

1. На первом шаге алгоритма осуществляется переход из системы координат (с/к) облака в с/к наблюдателя [3]. Для выполнения первого шага достаточно осуществить пересчёт координат центров сфер:

$$p_i = Mp_i^0,$$

где p_i^0 – центр i-й сферы в с/к облака; p_i – центр i-й сферы в с/к наблюдателя; M – матрица перехода из с/к облака в с/к наблюдателя.

2. На следующем шаге определяются параметры пересечения x-го β-среза с базовой сферой облака (сфера 0-го уровня). Угол между соседними β-срезами $\Delta\beta_x$ и α-срезами $\Delta\alpha_y$ не должен превышать мак-

симальную угловую погрешность системы [6]. Исходя из этого вычисляется разрешение формируемого изображения $X \times Y$.

Выполнение первого шага приводит к тому, что β-срез всегда представляет собой плоскость, проходящую через ось y с/к наблюдателя и x-й столбец пикселей формируемого изображения (рис. 1).

Таким образом, уравнение β-среза имеет вид

$$x\cos(\beta_x) - z\sin(\beta_x) = 0, \qquad (1)$$

где $x \in \{1,\ldots,X\}$ – номер столбца в формируемом изображении; β_x – угол между плоскостью x-го β-среза и осью z.

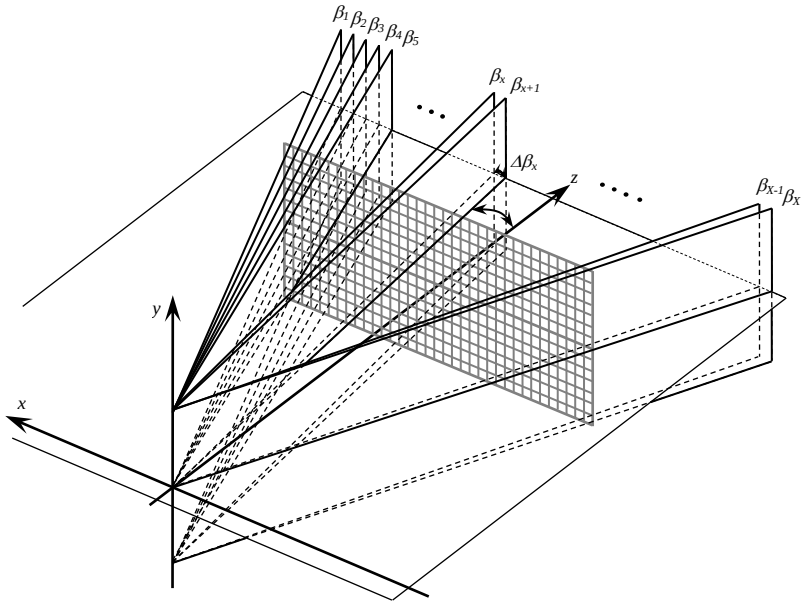

Рис. 1. Формирование β-срезов для классификации лучей

Для наличия пересечения между плоскостью и сферой расстояние d_i от плоскости β-среза до центра i-й сферы не должно превышать радиуса r_i этой сферы:

$$d_i \le r_i. \qquad (2)$$

Расстояние от точки $p(x_0, y_0, z_0)$ до плоскости $Ax + By + Cz + D = 0$ определяется соотношением

$$d = \frac{|Ax_0 + By_0 + Cz_0 + D|}{\sqrt{A^2 + B^2 + C^2}}, \qquad (3)$$

Из уравнения (1) видно, что знаменатель в соотношении (3) для любого β-среза всегда равен 1. Поэтому для проверки неравенства (2) достаточно в уравнение (1) подставить координаты центра сферы, полученные на первом шаге алгоритма. Этот процесс может выполняться независимо для каждого β-среза.

Если пересечения сферы с β-срезом нет, то алгоритм для этого β-среза завершается. Если пересечение есть, то создается два списка [7] для хранения номеров сфер и в первый из них заносится 0 – номер базовой сферы.

3. Для каждого β-среза запускается итерационный процесс, где на каждом уровне итерации формируется список сфер, пересекающих данный β-срез. Из первого списка извлекается номер сферы n^k, с которой было пересечение, и вычисляются номера сфер следующего уровня n_j^{k+1}:

$$n_j^{k+1} = m\, n^k + j; \quad j \in \{1, \ldots, m\},$$

где $m = 5$ – число сфер, на которое разбивается сфера предыдущего уровня детализации; k – номер итерации.

Для каждой n_j^{k+1} сферы осуществляется проверка условия (2) и если оно выполняется, то номер этой сферы заносится во второй список. Так продолжается до тех пор, пока не будет исчерпан первый список. После этого списки меняются местами, номер итерации увеличивается на 1. Достижение уровня $k \le K$ с требуемым уровнем детализации приводит к переходу на следующий шаг алгоритма.

4. Для каждого β-среза, у которого список, полученный на предыдущем шаге, не пустой, осуществляется расчёт диапазона α-срезов (рис. 2). Для этого определяются параметры пересечения плоскости и сферы: координаты центра окружности и значение радиуса.

Центр окружности определяется путем поворота центра сферы на угол $-\beta_x$ вокруг оси y:

$$x_i^{\beta_x} = z_i\, cos(-\beta_x) - x_i\, sin(-\beta_x) = z_i\, cos(\beta_x) + x_i\, sin(\beta_x);$$
$$y_i^{\beta_x} = y_i,$$

здесь x_i, y_i, z_i – координаты центра i-й сферы в с/к наблюдателя; $x_i^{\beta_x}, y_i^{\beta_x}$ – координаты центра окружности на плоскости β-среза.

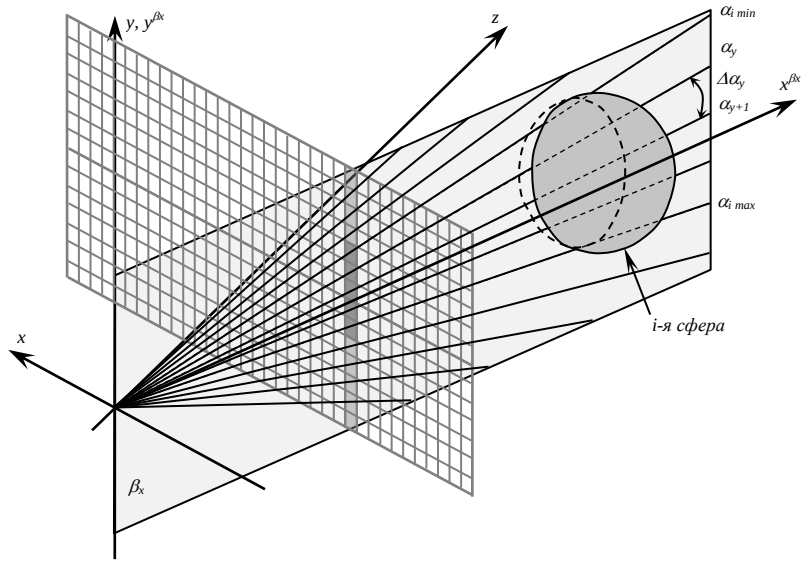

Рис. 2. Классификация по α-срезам

Для вычисления значения радиуса окружности на плоскости β-среза используем параметры, полученные при проверке условия (2):

$$r_i^{\beta_x} = \sqrt{r_i^2 - d_i^2}\ .$$

Если выполняется условие

$$x_i^{\beta_x} - r_i^{\beta_x} < 0\,,$$

то это означает, что сфера находится позади наблюдателя и должна быть исключена из дальнейшей обработки.

В противном случае определяется диапазон изменения α-срезов, пересекающих окружность:

$$\alpha_{i_{min}} = \frac{x_i^{\beta_x} y_i^{\beta_x} + r_i^{\beta_x}\sqrt{A}}{\left(x_i^{\beta_x}\right)^2 - \left(r_i^{\beta_x}\right)^2};$$

$$\alpha_{i_{max}} = \frac{x_i^{\beta_x} y_i^{\beta_x} - r_i^{\beta_x}\sqrt{A}}{\left(x_i^{\beta_x}\right)^2 - \left(r_i^{\beta_x}\right)^2},$$

где $A = \left(x_i^{\beta_x}\right)^2 + \left(y_i^{\beta_x}\right)^2 - \left(r_i^{\beta_x}\right)^2$.

Если $A < 0$, то это означает, что наблюдатель находится внутри сферы, и диапазон α-срезов в этом случае находится в пределах от α_1 до α_Y, то есть необходимо обрабатывать все лучи для данного β-среза. После этого определяем рабочий диапазон α-срезов как результат пересечения двух диапазонов $\{\alpha_{i_{min}}, \alpha_{i_{max}}\}$ и $\{\alpha_1, \alpha_Y\}$:

$$\{\alpha_{i_{min}}', \alpha_{i_{max}}'\} = \{\alpha_{i_{min}}, \alpha_{i_{max}}\} \cap \{\alpha_1, \alpha_Y\}. \qquad (4)$$

5. Для каждого проекционного луча создаётся список, куда будут заноситься параметры пересечений луча со сферами облака: значение расстояния от наблюдателя до точек пересечения l_1 и l_2 (координаты *x*, *y* и *z* однозначно восстанавливаются по параметрам проекционного луча и расстоянию *l*), прозрачность и направление нормали. Расстояние и прозрачность необходимы для учёта взаимного положения сфер облака и других объектов сцены при вычислении цвета пикселя.

6. Для каждого проекционного луча, попадающего в рабочий диапазон (4), осуществляется расчёт точек пересечения с окружностью:

$$x_i^{1,2} = \frac{x_i^{\beta_x} + B y_i^{\beta_x} \mp \sqrt{(x_i^{\beta_x} + B y_i^{\beta_x})^2 - A(1 + B^2)}}{1 + B^2}, \qquad (5)$$

где $B = tan(\alpha_y)$.

Пара координат точек пресечения, являющаяся результатом вычислений соотношения (5), заносятся в отдельный выходной список, формируемый для каждого проекционного луча.

Следует отметить, что значения B и $1 + B^2$ могут быть заранее вычислены и занесены в таблицу, поскольку зависят только от углового разрешения графической системы.

7. Для каждого списка, полученного на предыдущем шаге, выполняется сортировка элементов по первой точке.

8. Обработка выходного списка заключается в определении прозрачности каждой сферы и начинается со сферы, находящейся ближе всех к наблюдателю.

Прозрачность среды внутри облака определяется метеорологической дальностью видимости (МДВ) *Sм*, которая также является входным параметром модели. МДВ определяет дальность видимости абсолютно чёрного объекта и однозначно характеризует ослабление средой излучения.

Для практических расчётов видимости используют коэффициент пропускания среды *T*, который для оптически однородного слоя среды толщиной, равной единице длины, носит название удельной прозрачности *t*. Зная удельную прозрачность и толщину слоя среды *l*, можно определить коэффициент пропускания [8]:

$$T = t^l . \qquad (6)$$

Удельная прозрачность среды и МДВ связаны соотношением

$$S_м = ln\,0{,}02 / ln\,t .$$

Таким образом, зная МДВ и рассчитав длину пути проекционного луча внутри сферы легко получить коэффициент пропускания.

Расстояния l_1 и l_2 вычисляются на основании данных, полученных в (5), по следующим соотношениям

$$l_i^1 = x_i^1\sqrt{1+B^2};\;\; l_i^2 = x_i^2\sqrt{1+B^2} .$$

Определяется длина участка, на котором проекционный луч проходит внутри сферы

$$l = l_i^2 - l_i^1$$

и осуществляется расчет базового коэффициента пропускания T_i по соотношению (6).

Далее определяется значение функции-модификатора прозрачности облака [1]:

$$f(d_i, r_i) = 1 - \frac{d_i}{r_i} .$$

Осуществляется расчет скорректированного коэффициента пропускания T_i' с учетом функции-модификатора по соотношению:

$$T_i' = T_i f(d_i, r_i) + 1 - f(d_i, r_i).$$

9. Для вычисления цвета сферы облака необходимо определить направление нормали. Для этого потребуется восстановить координаты *x*, *y* и *z* первой точки пересечения по параметрам проекционного луча:

$$\begin{aligned} x_i^1 &= l_i^1 \cos(\alpha_y)\sin(\beta_x); \\ y_i^1 &= l_i^1 \sin(\alpha_y); \\ z_i^1 &= l_i^1 \cos(\alpha_y)\cos(\beta_x). \end{aligned} \qquad (7)$$

На основании данных, полученных по соотношениям (9) рассчитывается нормаль в первой точке пересечения с i-ой сферой:

$$n_{x_i} = x_i^1 - x_i;\ n_{y_i} = y_i^1 - y_i;\ n_{z_i} = z_i^1 - z_i\,.$$

Затем направление вектора нормали модифицируется в соответствии с соотношением

$$\vec{n}' = f(u,v)\vec{n}\,, \qquad (8)$$

где $f(u,v)$ – параметрическая функция изменения нормали [1].

Дальнейшие действия по вычислению цвета пикселя, учитывающие пространственное положение облака и других объектов сцены относительно источников света и друг друга, могут быть выполнены по стандартным алгоритмам [3, 4].

10. Результирующий цвет определяется соотношением

$$C = C_1\left(1 - T_1'\right) + \sum_{l=2}^{n}\left(C_l(1 - T_l')\prod_{k=1}^{l} T_k'\right),$$

здесь C – цвет сферы, полученный на шаге 9; n – количество сфер на проекционном луче.

Следует отметить, что при выполнении последнего шага алгоритма можно сократить объем вычислений, если при выполнении каждого очередного умножения проверять результирующую прозрачность. При выполнении условия

$$T \le 0.2 \qquad (9)$$

вычисления могут быть остановлены и остальные точка списка исключены из обработки, поскольку в соответствии с законом Вебера [9, 10] глаз перестает различать дальнейшее изменение прозрачности облака.

Практические результаты

Использование предложенного алгоритма позволило ускорить процесс вычислений приблизительно в четыре раза по сравнению с классическим алгоритмом, использующим параметрическое описание проекционных лучей. Результаты расчета на персональном компьютере с процессором Intel CoreDuo E6600 и 4 ГБ ОЗУ изображения облака с

разрешением 800x600 пикселей приведены в табл. 1. Расчет велся с использованием одного процессорного ядра.

Таблица 1. Сравнение производительности алгоритмов

Количество сфер на последнем уровне	**Время счета, мс**	
	Классический алгоритм	**Алгоритм с классификацией лучей**
15625	4750	1171
390625	9203	2422
9765625	22116	8844

Заключение

Предложенный алгоритм визуализации использует разбиение пространства по β-срезам и α-срезам, что позволяет выполнить классификацию лучей по наличию пересечений со сферами и исключить из обработки проекционные лучи не пересекающие облако. Следует отметить, что использование технологий AMD Stream или NVIDIA CUDA позволит существенно улучшить производительность, поскольку существует возможность широкого распараллеливания процесса вычислений как по β-срезам так и по α-срезам при вычислении параметров пересечения проекционного луча со сферами облака и даст возможность осуществлять обработку в реальном времени.

Библиографический список

1. *Ostroushko A., Bilous N., Bugriy A., Chagovets Ya.* Mathematical Model of the Cloud for Ray Tracing // International Journal "Information Theories and Applications", Vol. 17, Number 1, 2010
2. *Никулин Е.А.* Компьютерная геометрия и алгоритмы машинной графики. – СПб:БХВ-Петербург, 2003. – 560 с.
3. *Fransis S. Hill.* Computer Graphics Using OpenGL (c) Prentice Hall 2001.
4. *Foley J.D., van Dam A., Feiner S.K., Hughes J.F.* Computer Graphics (principles and practice) by Addison-Wesley Publishing Company, Inc. 1996. – 1175 p.
5. *Гусятин В.М.* Алгоритмы сканирования сцены в системах визуализации // Харьков, ХТУРЭ, Радиоэлектроника и информатика. – 2000. – №1.– С. 46-49.

6. *Гусятин В.М., Бугрий А.Н.* Расчет угловой погрешности спецпроцессора в системе визуализации // Автоматизированные системы управления и приборы автоматики: Сб. научн. трудов. Выпуск 112. – Харьков: ХТУРЭ, 2000. – С.4-10.
7. *Bjarne Stroustrup*. The C++ Programming Language— Addison-Wesley Pub Co; 3rd edition, 2000. – 990 p.
8. *McCartney E.J.* Optics of the Atmosphere Scattering by Molecules andParticles (Wiley, New York, 1976), 176–261
9. Атмосфера: Справочник. – Л.: Гидрометеоиздат, 1991. – 512 С.
10. *Дрофа А.С., Кацев И.Л.* Некоторые вопросы видения через облака и туманы. // Метеорология и гидрология, 1981. − № 1. − С. 101-109.