УДК 004.932.2+004.932.72'1

И.А. Запорожченко, М.А. Григорьев, С.А. Зори
Донецкий национальный технический университет, г. Донецк
кафедра прикладной математики

АНАЛИЗ АЛГОРИТМОВ ТРАССИРОВКИ ЛУЧЕЙ ДЛЯ РЕАЛИСТИЧНОЙ ВИЗУАЛИЗАЦИИ ТРЁХМЕРНЫХ СЦЕН И СПОСОБОВ УМЕНЬШЕНИЯ ИХ ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ

Аннотация

Запорожченко И.А., Григорьев М.А., Зори С.А. ***Анализ алгоритмов трассировки лучей для реалистичной визуализации трехмерных сцен и способов уменьшения их вычислительной сложности.*** *Разработан алгоритм трассировки лучей для реалистичной визуализации трехмерных сцен. Использован метод сеточного разделения для уменьшения сложности алгоритма. Проанализированы результаты работы для различных конфигураций. Найдены оптимальные значения.*

Ключевые слова: *трёхмерная сцена, свет, луч, трассировка лучей, рендеринг, сеточное разделение.*

Постановка проблемы. В наше время компьютерная графика требует чистого, детализированного, визуально насыщенного изображения чтобы успешно взаимодействовать по определенной тематике с нужной аудиторией. Из всех существующих схем рендеринга трехмерных сцен, трассировка лучей дает наиболее высокоточный визуальный эффект.

Анализ литературы. Изучен и реализован алгоритм трассировки лучей Вайтеда [1] , для уменьшения вычислительной сложности был реализован метод сеточного разделения[2], для решения задачи определения через какие подпространства проходит луч, использовался алгоритм 3DDA - модификация алгоритма Fujimoto[3].

Цель статьи – модификация алгоритма трассировки лучей для уменьшения его вычислительно сложности и его реализация.

Постановка задачи исследования. В основу исследования будет положено понимание алгоритмов реалистичного рендеринга и поиск способа уменьшения их вычислительной сложности для моделирования динамической сцены в реальном времени.

Решение задач и результаты исследований.

Для трассировки лучей был применён алгоритм Вайтеда. Этот алгоритм можно описать следующими действиями:

1) Для каждого пикселя трассируем первичный луч в направлении V к первой видимой поверхности.
2) Для каждого пересечения трассируем вторичные лучи:

- Лучи Li в направлении источников света
- Отраженные лучи в направлении R
- Лучи преломления или проходящие лучи T

Общий вид алгоритма можно посмотреть на рисунке 1

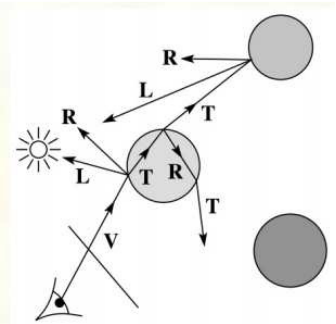


Рисунок 1 – Общий вид алгоритма Вайтеда трассировки лучей

Поиск пересечения лучей и объектов это наиболее трудоемкая задача и она занимает более 90% времени работы алгоритма. При стандартном использовании алгоритма трассировки лучей мы ищем пересечения между лучом и каждым объектом. При этом количество вычислений можно выразить формулой:

n_calculations=⟦res⟧_x*⟦res⟧_y*n_objects (1),

Где ⟦res⟧_x*⟦res⟧_y – умножение разрешения экрана, что в принципе является количеством первичных лучей,

n_objects – количество объектов на сцене.

При этом зависимость сложности от количества объектов равна O(n). Это довольно трудоемкая задача и ее можно упростить.

Для упрощения разделим пространство на подпространства. То есть разделим все пространство на небольшие кубы. Для трассировки луча будем рассматривать только те подпространства, через которые проходит луч. Вид сверху данного алгоритма схематически изображен на рисунке 2

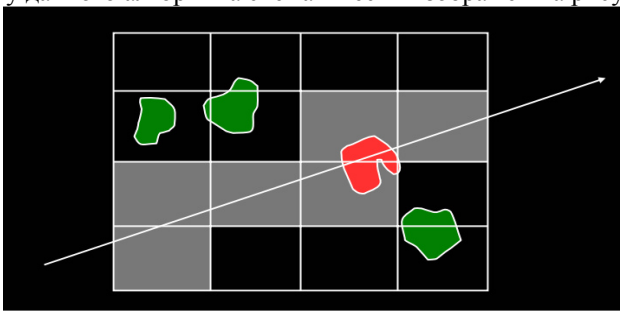


Рисунок 2 – Разделение пространства на подпространства вид сверху.

Для того, чтобы определить через какие подпространства проходит луч, использовался алгоритм 3DDA - модификация алгоритма Fujimoto.

Для реализации алгоритма будем хранить, какие объекты входят подпространство. При просчете также будем идти по всем подпространствам до первого пересечения с объектом, остальные нас не будут интересовать.

Данный алгоритм достаточно прост в реализации и достаточно эффективен, однако имеет недостаток. Если объекты собираются в одной области, то использование сеточного разделения не эффективно. (см. рис. 3).

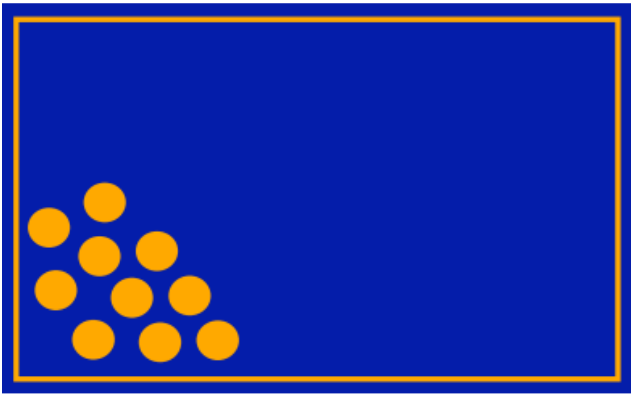


Рисунок 3 – Расположение объектов, при котором сеточное разделение не выгодно

Результаты:

Таблица 1 – Увеличение эффективности работы в зависимости от количества сфер и количества кубиков.

Сферы / Кубы	5	10	20	50	100
2	0.72	1.02	1.36	2.08	2.71
4	0.75	1.13	1.68	3.20	4.33
8	0.76	1.20	1.92	3.76	5.25
16	0.78	1.20	1.93	3.76	4.99
32	0.74	1.12	1.69	3.17	3.82

Можно заметить что данный метод становится более эффективным при большом количестве сфер. Также определено, что оптимальное количество кубов рвняется 8.

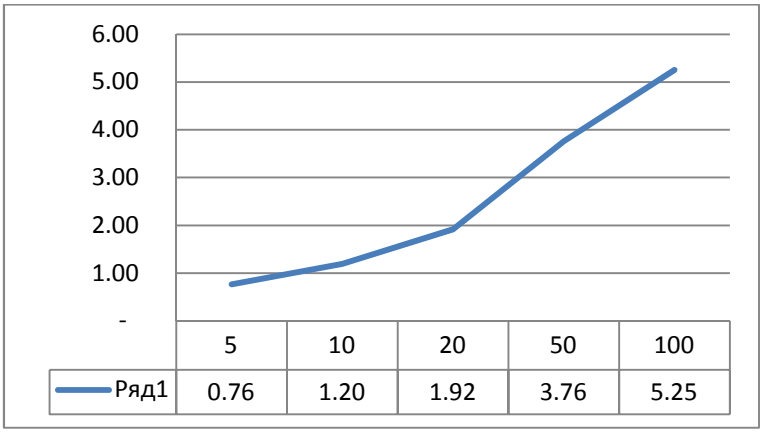
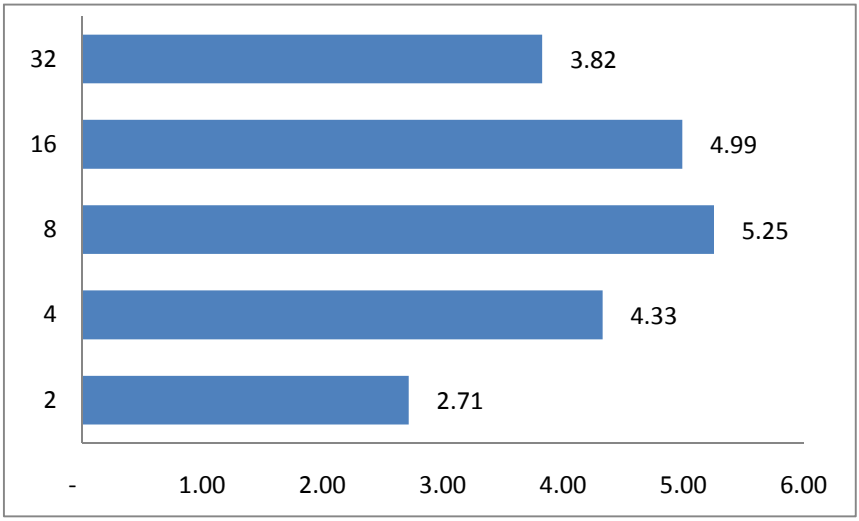


Рисунок 4 – Зависимость эффективности метода от количества сфер при оптимальном количестве кубов



.

Рисунок 5 – Зависимость эффективности метода от количества кубов для 100 сфер.

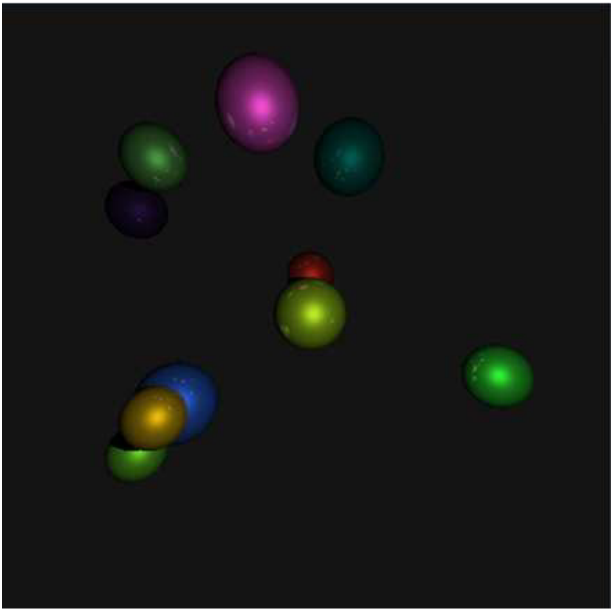


Рисунок 6 – Результат работы

Выводы. Разработан алгоритм трассировки лучей для реалистичной визуализации трехмерных сцен. Использован метод сеточного разделения для уменьшения сложности алгоритма. Проанализированы результаты работы для различных конфигураций. Данный метод становится более эффективным при большом количестве сфер. Определено, что оптимальное количество кубов равняется 8

Список литературы

1. Don Fussel. Ray tracing. [Электронный ресурс]. – Электрон. дан.: 2013 – Режим доступа: ttp://www.cs.utexas.edu/~fussel/courses/cs384g/lectures/lecture09-Ray_tracing.pdf – Яз. англ.

2. Regular grid. [Электронный ресурс]. – Электрон. дан.: 2013 – Режим доступа: http://www.ray-tracing.ru/articles182.html

3. An Efficient and Robust Ray–Box Intersection Algorithm. [Электронный ресурс]. – Электрон. дан.: 2013 – Режим доступа: http://www.cs.utah.edu/~awilliam/box/box.pdf– Яз. англ.