

МЕТОД ОТЛОЖЕННОГО ЗАТЕНЕНИЯ ТРЁХМЕРНЫХ СЦЕН, ИСПОЛЬЗУЮЩИЙ БУФЕР ВИДИМОСТИ

Красковский П. Н., Серебряная Л. В.

Кафедра программного обеспечения информационных технологий
Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: kraskovskij@bsuir.by

В данной работе рассмотрены основные подходы к затенению трёхмерных сцен. Представлен метод отложенного затенения, использующий буфер видимости. Проанализированы достоинства и недостатки рассмотренных методов, а также их эффективность.

ВВЕДЕНИЕ

С увеличением мощности компьютерного оборудования стало возможным использование физически достоверных методов освещения в программах трёхмерной визуализации и видеоиграх. Такие методы требуют больших вычислительных ресурсов для вывода изображения на современные дисплеи, имеющие высокое разрешение.

При использовании наивных методов затенения в процессе визуализации трёхмерной сцены освещение для каждого пикселя может быть пересчитано несколько раз, что приводит к уменьшению скорости генерации изображения. Падение производительности особенно заметно при большом количестве источников света.

Чтобы уменьшить количество вычислений, необходимо использовать методы отложенного затенения, при которых расчёт освещения происходит после определения атрибутов всех видимых фрагментов сцены.

I. МЕТОД ПРЯМОГО ЗАТЕНЕНИЯ

Простым способом визуализации трёхмерных сцен является метод прямого затенения, при котором все расчёты освещения происходят во время заполнения Z-буфера.

Z-буфер представляет собой двумерный массив данных, который хранит глубину нарисованных фрагментов сцены. Графически Z-буфер можно представить в виде полупрозрачного изображения, каждый пиксель которого имеет определённый оттенок серого (см. рис. 1).

Перед рисованием очередного фрагмента необходимо сравнить значение глубины этого фрагмента со значением глубины нарисованного в той же позиции фрагмента в Z-буфере. Если значение глубины нового фрагмента меньше значения глубины в Z-буфере (фрагмент расположен ближе к наблюдателю), необходимо заменить значение глубины в Z-буфере на значение глубины нового фрагмента. В противном случае новый фрагмент не рисуется [1].

Очевидно, что количество замен значений в Z-буфере, а также количество вычислений освещения зависит от порядка рисования фрагмен-

тов. Если первыми рисуются самые близкие к наблюдателю фрагменты, количество вычислений будет наименьшим, так как дальние фрагменты не пройдут проверку Z-буфера и не будут нарисованы. Наихудший случай – когда фрагменты рисуются в обратном порядке от самых дальних к самым близким. В таком случае количество вычислений освещения будет наибольшим.

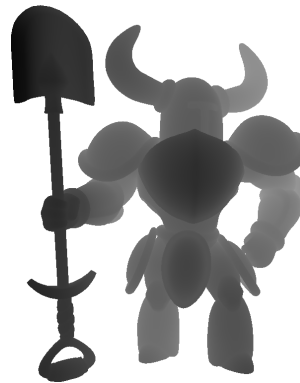


Рис. 1 – Графическое представление Z-буфера

II. МЕТОД ОТЛОЖЕННОГО ЗАТЕНЕНИЯ

Чтобы избавиться от избыточных вычислений освещения, свойственных методу прямого затенения, необходимо использовать метод отложенного затенения [2]. Основная идея данного метода заключается в том, чтобы перенести все тяжёлые вычисления освещения на более поздний этап.

Визуализация трёхмерной сцены методом отложенного затенения состоит из двух шагов. На первом шаге определяются атрибуты всех видимых фрагментов сцены, которые хранятся в виде коллекции текстур, называемой G-буфером. Данный буфер содержит информацию о позициях, нормалях, цветах и других характеристиках визуализируемых объектов (см. рис. 2). В отличие от вычислений освещения формирование G-буфера не требует больших вычислительных затрат.

На втором шаге метода выполняются вычисления освещения с использованием данных, записанных в G-буфер. При этом вычисления

производятся только для тех фрагментов, которые видны наблюдателю. Все невидимые фрагменты отбраковываются на предыдущем шаге во время заполнения Z-буфера.

Очевидным недостатком данного метода является повышенное потребление оперативной памяти, так как в G-буфере требуется сохранять большое количество текстур. Также данный метод не подходит для визуализации простых сцен с малым количеством объектов и источников света и может работать медленнее метода прямого затенения.

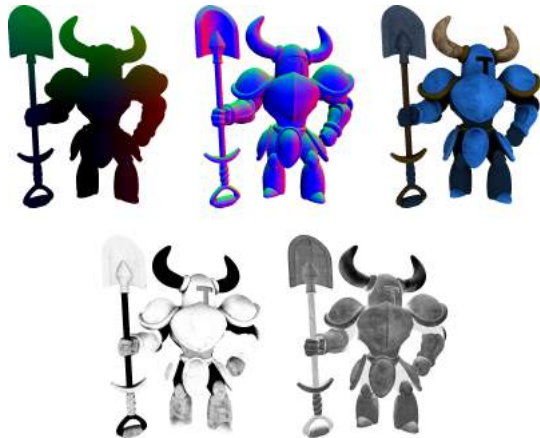


Рис. 2 – Содержимое G-буфера: позиции, нормали, альbedo, металличность, шероховатость

III. БУФЕР ВИДИМОСТИ

Потребление оперативной памяти можно уменьшить, если вместо G-буфера использовать буфер видимости, который хранит индексы объектов и треугольников, которым принадлежат видимые фрагменты [3]. Тогда на втором шаге метода по этим индексам можно получить атрибуты вершин треугольников для нахождения атрибутов соответствующих фрагментов, используя барицентрические координаты.

Данный метод позволяет существенно сократить потребление оперативной памяти, а также подходит для визуализации сцен любой сложности. Тем не менее метод не подходит для визуализации полупрозрачных объектов, так как в буфере содержится информация только о самых близких к наблюдателю фрагментах. Данная проблема свойственна всем методам отложенного затенения, но отсутствует у метода прямого затенения.

IV. АНАЛИЗ ЭФФЕКТИВНОСТИ МЕТОДОВ

На рисунках 3 и 4 представлены диаграммы сравнения скорости работы и потребления оперативной памяти рассмотренными ранее методами. Все замеры производились для трёх сцен, состоящих из одного объекта и одного источника света. В каждой сцене объекты отличались геометрической сложностью и количеством многоугольников. Все вычисления производились на процессоре без использования графических API. Изоб-

ражение выводилось на дисплей с разрешением 1366×768 .

На рисунках видно, что наилучшее соотношение скорости и потребления оперативной памяти показывает метод, использующий буфер видимости. Прирост скорости особенно заметен на объектах, имеющих сложную геометрию и большое количество многоугольников. Худшие показатели у классического метода отложенного затенения. Это обусловлено описанными ранее недостатками данного метода – замедление скорости работы при малом количестве объектов и источников света, а также повышенное потребление оперативной памяти.



Рис. 3 – Сравнение скорости работы различных методов затенения



Рис. 4 – Сравнение потребления памяти различных методов затенения

V. ВЫВОДЫ

Метод отложенного затенения, использующий буфер видимости, целесообразно использовать при визуализации трёхмерных сцен любой сложности. Если в сцене присутствуют полупрозрачные объекты, необходимо использовать дополнительные механизмы для правильного отображения этих объектов при использовании данного метода.

1. Real-Time Rendering / T. Akenine-Möller [et al.] – 4th ed. – Boca Raton : Taylor & Francis, CRC Press, 2018. – 1198 p.
2. The triangle processor and normal vector shader: a VLSI system for high performance graphics / M. Deering [et al.] – New York : Association for Computing Machinery, 1988. – P. 21–30. – (ACM SIGGRAPH Computer Graphics ; 22(4)).
3. Burns C. A. The Visibility Buffer: A Cache-Friendly Approach to Deferred Shading / C. A. Burns, W. A. Hunt. – P. 55–69. – (Journal of Computer Graphics Techniques (JCGT) ; 2(2))