

Ускорение расчёта вторичного освещения с помощью фильтрации в пространстве экрана и уточнения на основе информации о близлежащей геометрии

Груздев А.М, Фролов В.А, Игнатенко А.В.

Московский государственный университет им. Ломоносова
факультет вычислительной математики и кибернетики, Москва, Россия
aleks-gruzdev92@yandex.ru, vfrolov@graphics.cs.msu.ru, ignatenko@graphics.cs.msu.ru

Аннотация.

В данной работе рассмотрен алгоритм, основанный на аппроксимации точного расчета глобального освещения. Алгоритм работает в пространстве экрана и основан на идее переиспользования информации о вторичной освещенности в соседних пикселях изображения. Программная реализация использует трассировку лучей в поле расстояний до поверхности и позволяет рассчитывать глобальное освещение интерактивно на современных графических процессорах (2-5 FPS, GTX 570).

Ключевые слова: GPU, трассировка лучей, глобальное освещение.

1. ВВЕДЕНИЕ

Одна из основных задач компьютерной графики – синтез изображений, неотличимых для наблюдателя от фотоснимков реального мира. Реалистичное изображение можно получить, вычислив интеграл освещенности.

$$I(\varphi_r, \theta_r) = \iint_{\varphi_i, \theta_i} L(\varphi_i, \theta_i) R(\varphi_i, \theta_i, \varphi_r, \theta_r) d\varphi_i d\theta_i$$

Рисунок 1. Интеграл освещенности I представляет из себя двойной интеграл по полусфере. φ и θ задают направление в сферических координатах, L – количество входящего излучения, R – двулучевая функция отражения [Nicodemus 1965].

Эту задачу решают методы глобального освещения [Ritshel et al. 2011] [Kaplanian et al. 2010] [Jensen et al. 2002] [Radax 2008]. Обладая, как правило, высокой вычислительной стоимостью данные алгоритмы позволяют синтезировать высококачественные изображения.

В силу свойства определенного интеграла, мы можем раскладывать функции R и L (рис.1) на несколько компонент, вычислять для них интеграл отдельно, а результат получать как линейную комбинацию полученных значений.

В данной работе мы сосредоточимся на ускорении только непрямого (то есть претерпевшего одно и более переотражений) диффузного освещения. Под термином 'непрямое диффузное освещение' (или 'вторичное диффузное освещение') мы будем понимать компоненту функции L , обусловленную только Ламбертовскими [Basri et al. 2000] переотражениями света.

1.1. ОБЗОР РАБОТ ПО СМЕЖНОЙ ТЕМАТИКЕ

Эффективный подход расчета глобального освещения, предложен Wang в работе [Wang et al. 2009]. Этот метод использует фотонные карты. Основное направление данной работы - использование кэша освещенности, фотонных карт и финального сбора [Jensen et al. 2002] для быстрого расчёта диффузного вторичного освещения. Прямое освещение в работе [Wang et al. 2009] рассчитывается с помощью трассировки лучей с учётом резких теней от точечных источников света. Недостаток подхода предложенного в [Wang et al. 2009] – построение

kd-деревя в реальном времени на графическом процессоре, что требует дополнительных затрат памяти. Память является одним из важнейших ресурсов для современных программ рендеринга. Это особенно верно для GPU в связи с тем, что памяти на графических процессорах, как правило, меньше.

Аналогичный подход используется в [Fabianovski and Dingliana 2009], но без кэша освещенности и финального сбора. Прямое освещение рассчитывается трассировкой лучей, а вторичное – с помощью фотонных карт. Этот алгоритм подходит для каустиков, но даёт зашумлённые результаты для вторичного освещения.

McGuire и Luebke [McGuire and Luebke 2009] скомбинировали растеризацию для первичного освещения с выполняемой на центральном процессоре трассировкой фотонов и растеризацией сплатов для расчёта вторичного освещения на графической карте.

Другой подход к расчёту глобального освещения рассмотрен в [Ritshel et al. 2009]. Этот метод основан на вычислении переноса вторичного освещения не с помощью трассировки лучей в сцене, а за счёт вычислений в экранном пространстве. Такое упрощение расчетов является большим плюсом данного метода, но он имеет и серьёзные недостатки, а именно: только одно переотражение, локальное вычисление вторичного освещения.

1.2. Кэш освещенности

Кэширование освещенности снижает общую стоимость расчета вторичного освещения выполнением выборки по полусфере (или финального сбора) только в отмеченных точках сцены, кэшируя результаты и используя несколько раз сохранённые значения вторичного освещения за счёт интерполяции между точками кэша. Это было предложено в [Ward et al. 1988]. Алгоритм может быть кратко представлен в следующем виде:

if возможна интерполяция **then**

 используй сохранённые значения для интерполяции;

else

 вычисли новое значение;

 сохрани в кэше;

end if

Количество точек кэша освещенности обычно на 1-2 порядка меньше количества пикселей, таким образом, кэш довольно эффективен и может значительно ускорить весь рендеринг.

Кэширование освещения на графическом процессоре подробно описано в [Kvivanek et al. 2008] и [Gauton et al. 2005]. Основная их идея – заменить финальный сбор растеризацией сплатов. Для конкретной записи кэша возможно определить, какие видимые точки вносят вклад в расщепление записи на плоскости изображения. Используя аппаратно ускоренную растеризацию для семплирования по полусфере над каждой записью и так же для расщепления записей в специальном буфере, возможно добиться высокой производительности.

Метод "irradiance hints" представленный в [Papaioannou 2011] является стабильным (для анимации) и обладает высокой скоростью при вычислении диффузного глобального освещения. Метод основан на сеточном кэшировании освещения с отражающими теневыми

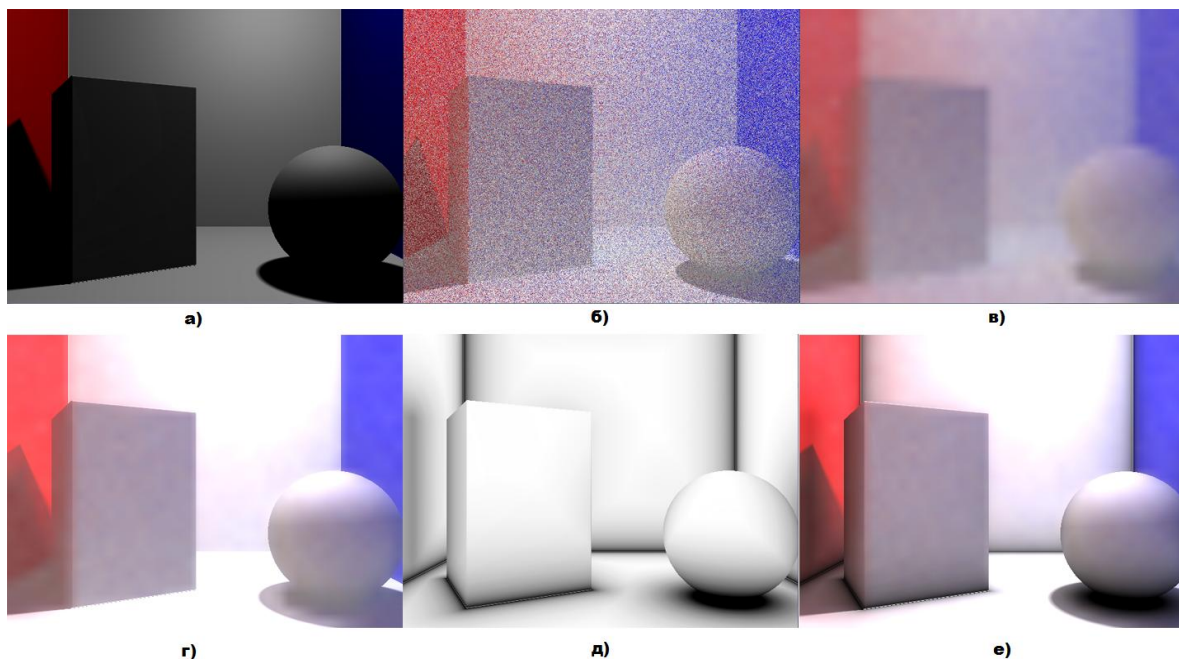


Рисунок 2. Работа алгоритма по шагам. а) – первичное освещение по Фонгу. б) – грубый расчёт освещённости по 4 случайным лучам. в) – отфильтрованное при помощи размытия вторичное освещение. г) - изображение 'а' с наложенным вторичным освещением д) - ambient occlusion е) – результирующее изображение.

картами и может обрабатывать множественные переотражения света.

В [Arıkan et al. 2005] впервые предложен метод разделяющий освещение на ближнее и дальнее. Он использует кэш вторичного освещения и представление вторичного освещения с помощью сферических гармоник. Кэш позволяет уменьшить количество точек, в которых необходимы расчёты вторичного освещения; представление освещения с помощью гармоник значительно упрощает расчет интеграла освещённости. Данный метод даёт хороший результат, но не подходит для интерактивного рендеринга из-за слишком долгого времени работы.

1.3. Выводы

Все рассмотренные работы предлагают эффективные методы расчета глобального освещения. Тем не менее, одни из них имеют недостатки (например, [Arıkan et al. 2005] в скорости), другие применимы только для некоторых частных случаев (например, [Radax 2008] только для сцен из диффузных материалов). Задачей нашей работы не является предложить некоторый новый метод, который будет лучше рассмотренных, а разработать метод ускорения расчета вторичного освещения, который может быть применён для уже некоторого существующего алгоритма (например, [Arıkan et al. 2005]) и позволит ускорить его работу, но, при этом, существенно не потеряв качество синтезируемого изображения.

2. ПРЕДЛОЖЕННЫЙ МЕТОД

Аналогично статье [Arıkan et al. 2005] в предложенном алгоритме вторичное освещение разделяется на 2 компоненты – ближнее и дальнее освещение. На первом шаге нами вычисляется компонента, представляющая дальнее освещение.

Следуя идее кэширования освещённости, наш алгоритм основывается на том, что вторичное освещение, обусловленное диффузными переотражениями света в пространстве экрана меняется с небольшой скоростью и при переходе между соседними пикселями информация о

вторичном освещении может быть переиспользована. Однако в областях изображения с высокой геометрической детализацией, как известно [Jensen et al. 2002], эффективность переиспользования снижается и известный алгоритм кэширования освещённости [Krivanek et al. 2008] вычисляет вторичную освещённость избыточное количество раз.

В противовес кэшу освещённости, в областях высокой геометрической детализации наш алгоритм уточняет вторичное освещение на основе информации о локальной форме поверхности при помощи аппроксимации “Ambient Occlusion” [Langer et al. 2000]. Предложенный алгоритм можно разделить на три основных шага:

1. Приближенное вычисление дальнего освещения

На данном этапе для каждого пиксела производится обратная трассировка луча, находится пересечение со сценой и производится грубый расчёт интеграла освещённости методом Монте-Карло с небольшим числом сэмплов (4-16). Полученная информация записывается в текстуру экранного размера для последующего использования (рис 2.б). В текстуру сохраняется суммарное падающее освещение, без учета цвета материала текущей точки (т.е. падающее освещение на данном этапе не домножается на цвет материала).

В отдельные текстуры сохраняется информация о позициях, материалах и нормалях геометрии (т.е. строится G-буффер [Ferko et al. 2012]).

2. Фильтрация дальнего освещения

Так как мы использовали небольшое число Монте-Карло сэмплов, информация о вторичном освещении получается сильно зашумлённая и для получения приемлемого для глаза результата ее необходимо фильтровать. В данной работе в качестве фильтра использовалось построение цепочки мип-мап уровней с последующим размытием (рис. 2.в). Этот алгоритм был выбран нами вследствие высокой скорости работы и простоты реализации. Однако мы полагаем, что использование более сложной фильтрации (например, подавление шума на GPU, описанное в [Kharlamov et al. 2007]) в данном случае разумно и принесет более качественные результаты.

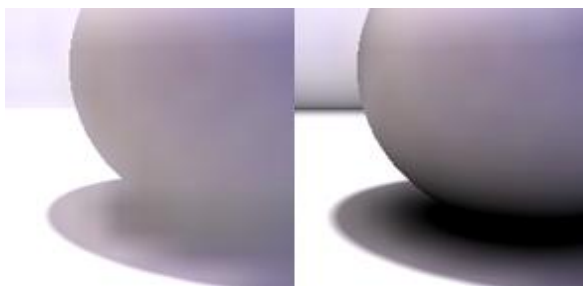
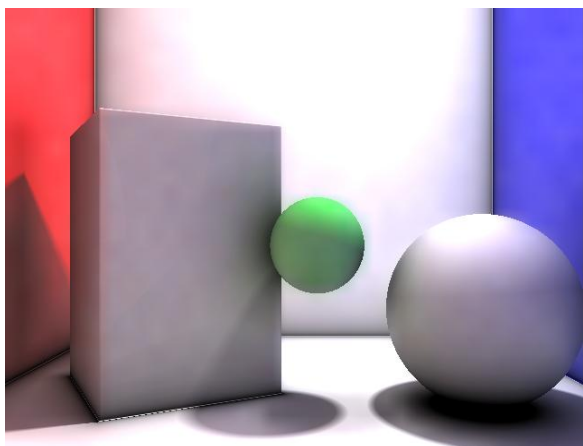


Рисунок 3. Уточнение освещения с помощью Ambient Occlusion

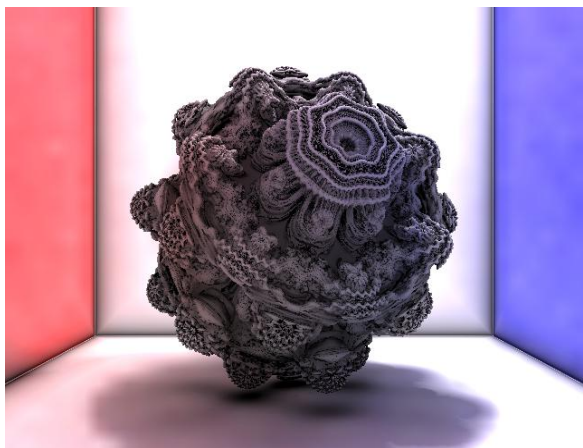
Мы также полагаем, что использование других методов, ускоряющих вычисление вторичной освещенности (например, кэш освещенности в пространстве экрана, LPV [Karlunyan et al. 2010], instant radiosity [Kinkelin et al. 2008]) для вычисления дальнего освещения возможно вместо описанных шагов Монте-Карло сэмплирования с последующей фильтрацией. Однако в данной работе мы не исследовали этот вопрос.

3. Синтез итогового изображения

На финальном шаге информация о позициях, нормалях и материалах считывается их G-буфера. Первичное освещение вычисляется по модели Фонга. Вторичное освещение вычисляется (с учетом цвета материала) при помощи аппроксимации "Ambient Occlusion" (рис. 2.д) с тем отличием, что так называемый "фоновый" цвет, или компонента ambient, берется равной не белому цвету (как полагается в классической реализации "Ambient Occlusion"), а цвету, сохраненному в текстуре дальнего



(a)



(б)

Рисунок 4. Тестовые сцены

освещения (рис 2.в). Уточнение освещения с помощью Ambient Occlusion позволяет избавиться от артефактов в виде ступенчатости после работы первых двух шагов (рис. 3)

3. ОЦЕНКА СТОИМОСТИ АЛГОРИТМА

Стоимость алгоритма складывается из:

1. Стоимости расчета вторичного освещения
2. Стоимости фильтрации текстуры вторичного освещения
3. Стоимости вычисления Ambient Occlusion

Выбирая различные алгоритмы для каждого шага, возможен выигрыш либо в скорости работы, либо в точности результата.

4. РЕАЛИЗАЦИЯ И РЕЗУЛЬТАТЫ

В программной реализации трассировка лучей производилась на основе функции расстояния [Quilez 2008] и геометрии, таким образом, задавалась аналитически. Данный вид трассировки лучей позволяет быстро и точно вычислять "Ambient Occlusion", однако не является оптимальным для простых примитивов, с которыми легко вычислять пересечение аналитически. Используемая трассировка лучей на основе функций расстояния, тем не менее, не ограничивает общность представленного в работе метода. Например, возможна реализация алгоритма для полигональной геометрии с использованием алгоритма расчёта Ambient Occlusion на основе буфера глубины, описанного в [Bavoil et al. 2008].

Программная реализация предложенного алгоритма выполнена на OpenGL4.0 и языке GLSL. Она протестирована на тестовых сценах (рис. 4). На сцене (рис 4. а) была достигнута скорость 2-5 FPS и на (рис. 4 б) – 1-3 FPS (в зависимости от минимальной длины шага, заданной для трассировки лучей на основе функций расстояния) на графическом процессоре NVIDIA GTX570.

5. ЛИТЕРАТУРА

- [1] [Arikan et al. 2005] Arikan O., Forsyth D. A. and O'Brein J. F. 2005. *Fast and Detailed Approximate Global Illumination by Irradiance Decomposition*. In Proceedings of ACM SIGGRAPH 2005, 1108-1114.
- [2] [Basri et al. 2000] Basri R. And Jacobs D. 2000, *Lambertian reflectance and linear subspaces*. Tech. Rep. MCS00-21, 2000-172R, Waizmann Institute of Science, NEC Research Institute
- [3] [Bavoil et al. 2008] Bavoil L. and Sainz M. 2008, *Screen Space Ambient Occlusion*, NVIDIA Corporation
- [4] [Fabianovski and Dingliana 2009] Fabianovski, B. and Dingliana, J., *Interactive Global Photon Mapping*, Proc. of the EUROGRAPHICS Conference, 2009
- [5] [Ferko et al. 2012] Ferko M. and Valient M. 2012, *Real-time Lightning Effects using Deferred Shadin*. Proceedings of CESC 2012
- [6] [Gauton et al. 2005] Pascal Gautron, Jaroslav Křivánek, Kadi Bouatouch, and Sumanta Pattanaik. *Radiance cache splatting: A GPU-friendly global illumination algorithm*. In *Proceedings of Eurographics Symposium on Rendering*, June 2005.
- [7] [Jensen et al. 2002] Jensen, H. W., Suykens F., Christensen Per H., Kato T. A *Practical Guide to Global Illumination using Photon Mapping*. SIGGRAPH 2002 Course Note #43. ACM, July 2002. (San Antonio, USA, July 21-26).

- [8] [Langer et al. 2000] Langer M.S. and Buelthoff H. H. 2000, *Depth discrimination from shading under diffuse lighting*, Perception. 29 (6) 649—660
- [9] [Kaplanyan et al. 2010] Kaplanyan A. and Dachsbacher C. 2010, *Cascaded Light Propagation Volumes for Real Time Indirect Illumination*, Crytek
- [10] [Kharlamov et al. 2007] Kharlamov A. and Podlozhnyuk V. 2005, *Image Denoising*, NVIDIA Corporation
- [11] [Kinkelin et al. 2008] Kinkelin M. and Liensberger C. 2008, *Instant Radiosity An Approach for Real-Time Global Illumination*, ICGA/TU Wien SS2008
- [12] [Krivánek et al. 2008] Krivánek, J., Gauthron, P., Ward, G., Jensen, H. W., Christensen, P. H., and Tabellion, E. 2008. *Practical global illumination with irradiance caching*. In ACM SIGGRAPH 2008 Classes (Los Angeles, California, August 11 - 15, 2008). SIGGRAPH '08. ACM, New York, NY, 1-20.
- [13] [McGuire and Luebke 2009] McGuire M. and Luebke D. 2009, *Hardware-Accelerated Global Illumination by Image Space Photon Mapping* Proc. of ACM SIGGRAPH/EuroGraphics High Performance Graphics 2009
- [14] [Nicodemus 1965] Nicodemus E. Fred, 1965. *Directional reflectance and emissivity of an opaque surface* (abstract). Applied Optics 4 (7): 767–775. DOI:10.1364/AO.4.000767
- [15] [Papaioannou 2011] G. Papaioannou, *Real-Time Diffuse Global Illumination Using Radiance Hint*. Presented at High Performance Graphics 2011, Vancouver, Canada, Aug. 2011.
- [16] [Quilez 2008] Quilez I., 2008. *Rendering Worlds with Two Triangles with raytracing on the GPU*. NVSCENE 08
- [17] [Radax 2008] Radax I., 2008, *Instant Radiosity for Real-Time Global Illumination*, Vienna University of Technology.
- [18] [Ritschel et al. 2011] Ritschel T., Dachsbacher C., Grosch T., and Kautz J., 2011, *The State of the Art in Interactive Global Illumination*, DOI: 10.1111/j.1467-8659.2012.02093.x
- [19] [Ritschel et al. 2009] Ritschel T., Grosch T. and Seidel H.P. 2009, *Approximating Dynamic Global Illumination in Image Space*. [18:40:38] Vladimir Frolov: Proceedings ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, Boston, MA, February 27—March 1, 2009
- [20] [Ward et al. 1988] Ward, G., Rubinstein, F., and Clear, R. 1988. *A ray tracing solution for diffuse interreflection*. In SIGGRAPH 1988, Computer Graphics Proceedings.
- [21] [Wang et al. 2009] Wang R., Zhou K., Pan, M., and Bao, H. 2009. *An efficient GPU-based approach for interactive global illumination*. ACM Trans. Graph. 28, 3 (Jul. 2009), 1-8.

Об авторах

Алексей Груздев - студент третьего курса ВМК МГУ имени М.В. Ломоносова. Его адрес: alexs-gruzdev92@yandex.ru
Владимир Фролов – аспирант ИПМ имени М.В. Келдыша. Он работает в NVIDIA инженером по развитию технологий. Его адрес: vfrolov@graphics.cs.msu.ru
Алексей Игнатенко - научный сотрудник, к.ф.-м.н ВМК МГУ имени М.В. Ломоносова. Он занимается исследованиями в области фотореалистичного 3D рендеринга. Его адрес: ignatenko@graphics.cs.msu.ru