

Высокочастотный кэш излучения

Константин Востряков
Институт Прикладной Математики им. М.В.Келдыша
vostryakov@gin.keldysh.ru

Аннотация

Разработан новый алгоритм кэширования падающего излучения на основе репроекции. Экстраполяция падающего излучения позволяет существенно сократить количество трассировок луча и вычисления функции отражения. В отличие от других алгоритмов кэширования излучения, предлагаемый алгоритм может работать с высокочастотными функциями отражения и освещением. Алгоритм может быть использован для расчета финального сбора в методах фотонных карт и излучательности. Предлагаемый метод быстрее, чем Монте-Карло трассировка с учетом функции отражения, при этом он конкурирует с ним по точности расчета.

1. ВВЕДЕНИЕ

Постоянно возрастающие требования к сложности сцен вынуждают разрабатывать новые подходы для расчета освещения. Визуальное прототипирование новых продуктов промышленности находит все большее применение и требует не только фотореалистичного, но и физически корректного расчета освещения. Ядром любого алгоритма расчета глобальной освещенности является вычисление интеграла освещения по всем направлениям. Для получения физически корректного решения традиционно применяются методы обратной Монте-Карло трассировки лучей. К сожалению эти методы слишком медленно сходятся. Ради ускорения расчета большинство других алгоритмов накладывают множество ограничений (только ламбертовские поверхности или только плавное освещение). Но большинство реальных объектов имеют сложные ДФО, а падающее излучение неравномерно и имеет большой диапазон. Следовательно возникает потребность в создании алгоритмов, которые позволили бы использовать сцены со сложными ДФО и освещением, но были бы более эффективными, чем традиционные методы обратной Монте-Карло трассировки лучей.

Предлагаемый алгоритм кэширует функцию падающего излучения и переиспользует вычисленные значения в соседних точках поверхности, что позволяет существенно сократить количество трассировок луча и вычислений функции отражения. В отличие от других алгоритмов кэширования излучения, предлагаемый алгоритм может работать с высокочастотными данными. В сравнении с классической реализацией метода Монте-Карло алгоритм дает ускорение в несколько раз при сравнимой точности расчета.

2. ПРЕДЫДУЩИЕ МЕТОДЫ

Для ускорения методов обратной трассировки могут применяться выборка по значимости ДФО. То есть трассировка лучей с плотностью пропорционально функции отражения. Для этого необходимо знать хотя бы приближение подынтегральной функции или уметь обратить функцию распределения, что не всегда возможно выполнить. Адаптивные Монте-Карло алгоритмы могут дать ускорение сходимости, например, Metropolis [1], кэш видимости [2] или многомерная адаптивная трассировка [3], но для их работы необходима регу-

лярная начальная выборка для приближения интегрируемой функции, которая может быть достаточно большой для сложных сцен.

Алгоритмом быстрого глобального освещения [4] генерируется небольшое количество виртуальных точечных источников света, используя метод квази-случайного блуждания. Затем выполняется визуализация сцены с использованием затенения с перемежающейся выборкой виртуальных источников света. Этот алгоритм создает артефакты из-за дискретизации вторичных виртуальных источников и плохо подходит для неламбертовских ДФО и сцен со множеством преград.

Методы, которые основаны на теневых картах [5], имеют аппаратную поддержку в графических процессорах. Затенение вычисляется как доля заграждения геометрии сцены, спроецированной обратно на источник. Методы теневых карт в большинстве случаев пренебрегают физической корректностью. Тем не менее некоторые моменты реализации репроекции предлагаемого алгоритма были почерпнуты из этих методов.

Фотонные карты [6] сохраняют освещение как облако точек столкновения частиц, которые были выпущены из источников, с поверхностями сцены. Однако, оценка освещения, полученная с помощью фотонной карты, имеет низкочастотный шум, и для его устранения используется обратная трассировка по всем направлениям — финальный сбор, который занимает подавляющую часть времени расчета.

Кэш облучения впервые был предложен в работе [7], что позволило в несколько раз ускорить визуализацию, поскольку только в небольшой части точек, видимых через пиксели, облучение вычислялось трассировкой лучей по полусфере, а в остальных точках оно экстраполировалось. Для более лучшего контроля ошибки экстраполяции Krivanek [8] предложил использовать многопроходный адаптивный алгоритм, который, при добавлении новой точки в кэш, сравнивает относительное отклонение облучения в соседних точках, что уменьшает количество артефактов, но приводит к множеству чрезмерных вычислений. Кроме того, в работе [8] было предложено расширить метод Ward'a, чтобы использовать умеренно-зеркальные ДФО, а не только ламбертовские. Это потребовало хранения освещения как функции от направления на полусфере. Для этого использовались сферические гармоники, которые, однако, не применимы к остронаправленным ДФО.

В курсе [6] описывается использование репроекции вместе с кэшем излучения. Точки пересечения лучей и поверхностей сцены, которые были вычислены ранее и записаны в кэш, проецируются на полусферу в новой точке поверхности, в которой необходимо вычислить освещение. Репроекция медленная процедура, кроме того, необходимо использовать z-буфер и что-то делать с пустотами, которые останутся после репроекции. В данной работе сделана попытка устранить недостатки алгоритма репроекции при использовании кэша излучения.

В работе [9] освещение было разделено на ближнее и дальнее. Поскольку дальнее освещение изменяется достаточно плавно, то кэш излучения гораздо более редкий, чем для полного осве-

щения. Для ближнего освещения применялся метод излучательности без учета функции видимости. Ближнее освещение вычисляется очень быстро, но игнорирование видимости дает физически некорректное решение. В предлагаемом алгоритме используется идея разделения на ближнее и дальнее освещение, но в более корректном виде.

3. ВЫСОКОЧАСТОТНЫЙ КЭШ ИЗЛУЧЕНИЯ

За основу нового алгоритма был выбран кэш излучения, поскольку он обладает хорошей масштабируемостью по сложности сцен и точности моделирования. К сожалению он, как и другие алгоритмы, имеет недостатки, которые необходимо устранить. Во-первых, трудно обеспечить заданный уровень ошибки. На практике это преодолевается гораздо большим числом точек в кэше, чем это могло бы быть, и ручным подбором параметров. Один из главных недостатков предыдущего алгоритма кэширования излучения — чрезмерная плотность кэша в зонах, где освещение меняется быстро или ДФО остронаправленная, необходимость многопроходности для точного контроля ошибки. Во-вторых, для того чтобы можно было использовать высокочастотное освещение и ДФО, а также корректно обрабатывать функцию видимости, для базиса нельзя использовать сферические гармоники. Они не могут точно представлять разрывную функцию видимости. В-третьих, использование градиентов для высокочастотного кэша излучения нецелесообразно. Для более точной экстраполяции была выбрана репроекция, которая является более медленной, но позволяет более корректно обрабатывать видимость.

3.1 АЛГОРИТМ ВЕРХНЕГО УРОВНЯ

Предлагается новый алгоритм, основанный на кэше излучения и репроекции. Вначале через все пиксели экрана трассируются лучи. Таким образом находим ближайшие точки пересечения со сценой и сохраняем их в массив. Кроме самой точки, также вычисляются и сохраняются нормали к поверхности и указатели на ДФО. Трассировкой лучей вычисляем первичное освещение от точечных и параллельных источников света. В конце проходим по массиву пикселей и запрашиваем кэш излучения, если в кэше находится подходящая точка для экстраполяции, то экстраполируем (осуществляем репроекцию и заполняем брешу), если нет, то трассируем лучи по полусфере и создаем запись в кэше (сохраняем текстуру падающего излучения и информацию для репроекции). Для удобства хранения падающего излучения в виде квадратной текстуры было выбрано отображение из квадрата на полусферу [10].

3.2 СОЗДАНИЕ ЗАПИСИ КЭША

На вход процедуры, которая вычисляет освещение, подаются координаты точки поверхности и нормаль к поверхности. На выходе имеем текстуру падающего излучения и массив п-элементов, после чего они могут быть записаны в кэш. П-элемент — это сокращение от поверхностный элемент (аналог термина *surfel* в английском языке, сокращение от *surface element*). Под этим термином понимается то, что в точке на поверхности имеется некоторая плоская дифференциальная площадка, параллельная поверхности в данной точке.

Вначале трассируем $n \times n$ лучей по полусфере (типичное значение 32×32 или 64×64), находим точки столкновения с поверхностями сцены. Далее вычисляем падающее излучение, которое идет вдоль лучей, сохраняем в текстуру. Вычисленные точки разделяем на слои по глубине. Каждая точка становится п-элементом. Сортировка выполняется по K слоям, что требует $K \cdot O(n \times n)$ времени. Слои выбираются по логарифмической шкале, поскольку ближние п-элементы нужно отсорти-

ровать точнее. Внутри слоя объединяем соседние п-элементы имеющие близкую нормаль. Для полученных п-элементов находим угловые точки в uv пространстве (двумерном пространстве полусферы). В конце сохраняем все п-элементы и текстуру падающего излучения в кэш.

3.3 РЕПРОЕКЦИЯ

На вход алгоритма репроекции и заполнения брешей (рисунок 1) подаются координаты точки на поверхности, ее нормаль, ДФО и направление взгляда. А также ближайшая запись из кэша, в которую входит: текстура падающего излучения и массив п-элементов. На выходе имеем исходящее излучение от точки поверхности вдоль направления взгляда. Репроекция осуществляется слой за слоем, начиная от ближнего слоя и заканчивая самым дальним. При этом полноценного Z-буфера не нужно, можно обойтись лишь маской заполнения. Репроецируем все прямоугольные п-элементы в новую точку поверхности, для каждого из них получаем в uv пространстве четыре угловые точки в новой системе координат. При этом по четырем точкам строим прямоугольник, координаты угловых точек которого — это максимальные uv координаты четырех репроецированных точек. Этим мы получаем приближенную проекцию п-элемента. Для п-элемента создаются маски п-элемента и маски неизвестности. Маска п-элемента — это битовая маска, где 1 расположены в тех местах uv пространства, куда п-элемент был спроецирован. Далее следует создание битовой маски следа движения проекции п-элемента. Эта маска вычисляется как объемлющий прямоугольник uv п-элемента в точке из кэша и новой проекции п-элемента. Маска следа позволяет определить область uv пространства, где открылась неизвестная зона (или другими словами зона тени точки из кэша). Маска неизвестности получается как маска следа минус маска заполнения. Маска неизвестности запрещает заполнение более дальних слоев в области неизвестности. Объект, который может находиться в такой зоне, может перекрыть фон, видимый из новой точки. Маской неизвестности мы препятствуем тому, что перекрытый фон будет учтен неправильно.

Далее обходим с некоторым шагом k маску заполнения и трассируем лучи там, где есть брешу. На один луч приходится несколько пикселей маски. Добавляем излучение, полученное трассировкой лучей, с весом равным числу незаполненных пикселей маски деленное на число пикселей k^2 .

Если после репроекции осталось много брешей, то можно использовать другие соседние записи из кэша. Для этого надо обнулить маску неизвестности, а маску заполнения оставить неизменной. Новый алгоритм корректно поддерживает экстраполяцию освещения от нескольких записей кэша с учетом закрытия одних объектов другими, что предыдущие методы не могли обеспечить.

3.4 РЕАЛИЗАЦИЯ РЕЗУЛЬТАТЫ

Трассировка лучей и вычисление ДФО выполнялись с использованием SSE инструкций [4]. Четыре луча трассировались одновременно. В качестве ускоряющей структуры использовалось *kdtree*. Репроекция осуществлялась сразу для четырех точек одновременно, также используя SSE2 инструкции. Маски заполнения и неизвестности являются битовыми матрицами 128×128 бит. Любая строка битовой матрицы — это 128-битная переменная. Битовые операции выполняются сразу над всей строкой длиной 128 бит, используя SSE2 инструкции. Перевод п-элемента из одной системы координат в другую может быть выполнен битовыми сдвигами. Репроеция очень похожа на растеризацию. Падающее излучение от п-элемента используется как текстура, а маска п-элемента как альфа-канал

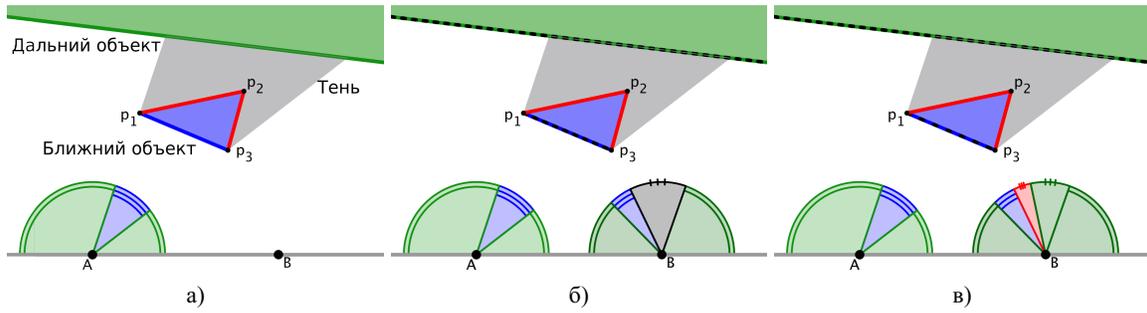


Рисунок 1: Алгоритм репроекции. а) Падающее излучение в точке А, экстраполируется в точку В б) В точке В освещение рассчитывается репроекцией значений падающего излучения, вычисленных в точке А и сохраненных в кэше. 4 значения излучения репроецируются одновременно используя SSE инструкции, учитывая перекрытие и затенения. в) После репроекции бреши и зона тени заполняются трассировкой лучей.

текстуры.



Рисунок 2: Сравнение качества изображений для выборки пропорционально ДФО 64×64 лучей (первая строка), алгоритма репроекции (вторая строка) и алгоритма репроекции с последующей выборкой по значимости еще 64×64 лучей (третья строка)

Для тестирования метода было выбрано изображение с большим динамическим диапазоном (БДД), которое задавало освещение на сфере. БДД освещения требует большой выборки для интегрирования. За образец для сравнения был выбран метод интегрирования освещенности Монте-Карло с генерацией 64×64 лучей с плотностью пропорционально ДФО, поскольку лишь предлагаемый метод (среди всех методов кэширования) конкурирует с ним по качеству на таких сценах. Метод Монте-Карло сравнивается с предложенным методом репроекции падающего излучения, где кэшируется также 64×64 лучей. Кроме того, после репроекции возможно построить кумулятивную кусочно-постоянную функцию распределения, используя которую можно трассировать лучи более значимым образом и получить несмещенную оценку. Репроекция с последующей выборкой по значимости еще 64×64 лучей также была протестирована.

Все тесты были выполнены для разрешения экрана 512×512

пикселей, один луч на пиксель. Тесты запускались на компьютере: Intel Core2 Quad Q9300 2.5 GHz, используя все 4 ядра процессора. Во всех случаях для уменьшения дисперсии использовалась стратификация базовых случайных величин, которые равномерно распределены в интервале $(0, 1)$. Для Монте-Карло трассировки пропорционально ДФО и при выборке по значимости SSE2 инструкции не использовались.

На рисунке 2 показано сравнение изображений при использовании всех трех методов. Классический контрольный метод Монте-Карло дает шум, который наиболее заметен на сцене shadow, она имеет наиболее остронаправленную зеркальную функцию отражения. При этом детерминизм метода превращает шум в смещение с повторяющимся шаблоном, что визуально менее приемлемо. Но можно использовать метод репроекции вместе с Монте-Карло трассировкой, что дает несмещенную оценку. При этом точность значительно повышается, а шум в освещении снижается. Качество изображение намного выше простого метода Монте-Карло, в котором лучи трассируются с плотностью пропорциональной ДФО. Как видно из таблицы 1 метод репроекции в несколько раз быстрее метода Монте-Карло с выборкой пропорционально ДФО, а репроекции с последующей выборкой по значимости показывает лишь немногим большее время расчета, чем базовый метод Монте-Карло. Также можно заметить, что при использовании метода репроекции точность расчета сравнима с методом Монте-Карло (в одной из сцен точность алгоритма репроекции оказывается немного лучше, в остальных немного хуже), при этом скорость в несколько раз больше. Если же после репроекции применяется выборка по значимости, то точность расчета значительно превосходит метод Монте-Карло.

Таблица 1: Время визуализации в секундах и через черту относительная ошибка $L_1 = \frac{\int |L - \langle L \rangle|}{\int L}$. Сравнение выборки пропорционально ДФО 64×64 лучей, алгоритма репроекции 64×64 лучей и алгоритма репроекции с последующей выборкой по значимости еще 64×64 лучей

Сцена	Кол-во треу-в	Монте-Карло ~ ДФО	Репроекция	Репроекция + Монте-Карло
shadow	110	950.9 / 4.7%	44.0 / 3.4%	961.4 / 1.9%
dragon	871k	310.3 / 2.6%	51.7 / 3.2%	362.9 / 1.8%
armadillo	346k	348.8 / 2.5%	58.1 / 3.3%	411.3 / 1.8%

На основании таблицы 1 точность можно приблизительно считать сравнимой с точностью алгоритма Монте-Карло. При этом можно говорить, что смещенная версия алгоритма репроекции дает ускорение приблизительно от 6 до 21.6 раза. Поскольку метод Монте-Карло сходится как $1/\sqrt{N}$, где

N — это количество лучей, то 2 кратное увеличение точности означает 4 кратное ускорение расчета. Используя таблицу 1 можно оценить ускорение, которое дает метод репроекции в несмещенной версии: от 1.92 до 6.1 раз (таблица 2).

Таблица 2: Ускорение (раз) в сравнении с контрольным методом Монте-Карло метода репроекции и метода репроекции с последующей выборкой по значимости (несмещенная версия)

Сцена	Кол-во треу-в	Репроекция	Репроекция + М-К (несмещенная)
shadow	110	21.6	6.12
dragon	871k	6.0	2.09
armadillo	346k	6.0	1.92

В отличие от предыдущих методов кэширования излучения, новый алгоритм репроекции точнее контролирует ошибку экстраполяции, что позволило отказаться от многопроходности, разбив изображение на независимые блоки, которые рассчитываются параллельно в разных потоках. Удалось добиться ускорения расчета вторичного освещения в 3.83 раза при использовании 4 потоков.

3.5 ЗАКЛЮЧЕНИЕ

Разработан алгоритм кэширования и нелинейной экстраполяции падающего излучения. Экстраполяция позволяет переиспользовать вычисленные значения освещения в соседних точках поверхности, существенно сократить количество трассировок луча и вычисления функции отражения. В отличие от других алгоритмов кэширования излучения, предлагаемый алгоритм может работать с высокочастотными функциями отражения и освещением. Более точный контроль ошибки экстраполяции позволяет отказаться от многопроходности и иметь локальный кэш для каждого потока, что ведет к эффективной многопоточной реализации. Алгоритм может быть использован для расчета финального сбора в методах фотонных карт и излучательности. В сравнении с методом обратной трассировки Монте-Карло при сравнимой точности расчета алгоритм работает в несколько раз быстрее. А при использовании вместе с Монте-Карло трассировкой дает несмещенное решение, в несколько раз ускоряя сходимость.

БЛАГОДАРНОСТИ

Работа поддержана грантами РФФИ №07-01-00450 и №09-01-00472.

4. СПИСОК ЛИТЕРАТУРЫ

- [1] Abhijeet Ghosh and Wolfgang Heidrich, “Correlated visibility sampling for direct illumination,” *Vis. Comput.*, vol. 22, no. 9, pp. 693–701, 2006.
- [2] Petrik Clarberg and Tomas Akenine-Möller, “Exploiting Visibility Correlation in Direct Illumination,” *Computer Graphics Forum (Proceedings of EGSR 2008)*, vol. 27, no. 4, pp. 1125–1136, 2008.
- [3] Toshiya Hachisuka, Wojciech Jarosz, Richard Peter Weistroffer, Kevin Dale, Greg Humphreys, Matthias Zwicker, and Henrik Wann Jensen, “Multidimensional adaptive sampling and reconstruction for ray tracing,” *ACM Transactions on Graphics (Proc. of SIGGRAPH)*, vol. 27, 08/2008 2008.
- [4] Ingo Wald, *Realtime Ray Tracing and Interactive Global Illumination*, Ph.D. thesis, 2004.

- [5] Michael Schwarz and Marc Stamminger, “Bitmask soft shadows,” *Comput. Graph. Forum*, vol. 26, no. 3, pp. 515–524, 2007.
- [6] Per H. Christensen, Per H. Henrik Wann Jensen, Toskiaki Kato, and Frank Suykens, “A practical guide to global illumination using photon mapping,” in *SIGGRAPH 2002 Course Notes*. Association for Computing Machinery, Aug. 2002, ACM SIGGRAPH, Course 43.
- [7] Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear, “A ray tracing solution for diffuse interreflection,” in *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 1988, pp. 85–92, ACM.
- [8] Jaroslav Křivánek, *Radiance Caching for Global Illumination Computation on Glossy Surfaces*, Ph.d. thesis, Université de Rennes 1 and Czech Technical University in Prague, December 2005.
- [9] Okan Arıkan, David Forsyth, and James F. O’Brien, “Fast and detailed approximate global illumination by irradiance decomposition,” in *ACM SIGGRAPH 2005 Full Conference DVD-ROM*, Aug. 2005.
- [10] Peter Shirley and Kenneth Chiu, “A low distortion map between disk and square,” *journal of graphics tools*, vol. 2, no. 3, pp. 45–52, 1997.

High-frequency radiance cache

Abstract

The paper is devoted to a new radiance cache algorithm based on reprojection. By sparsely sampling, caching and reprojecting incident radiance, we significantly reduce number of traced rays and shading calculations. In opposite to previous cache methods, our method is able to handle high-frequency BRDF and lightings. The method is intended for final gathering in radiosity or photon mapping algorithms. Our method is faster than a Monte-Carlo BRDF importance sampling and able to compete with it by quality.

Keywords: radiance cache, ray tracing, global illumination.

About the author

Konstantin Vostryakov is a PhD student of the Keldysh Institute for Applied Mathematics RAS. vostryakov@gin.keldysh.ru.