

Новый иерархический базис для освещения на полусфере

К.А. Востряков

Институт Прикладной Математики имени М.В. Келдыша РАН,

Москва, Россия

vostryakov@gin.keldysh.ru

Аннотация

Предложен новый иерархический базис для хранения освещения на полусфере. Базис позволяет осуществлять быструю свертку и быструю проекцию на базис. Благодаря использованию квадрато-дерева разложение по базису может аппроксимировать высокочастотные функции. Эти свойства востребованы при расчете освещения трассировкой лучей от карты окружения, заданной с большим динамическим диапазоном. Свертка функций на полусфере в новом базисе на несколько порядков быстрее, чем свертка, реализованная без использования базиса. Алгоритм свертки позволяет эффективную реализацию с помощью SSE инструкций.

Keywords: quad-tree, ray tracing, hemisphere basis, HDRI environment, SSE.

1. ВВЕДЕНИЕ

Визуализация искусственных объектов в окружении реальной сцены требует вычисления освещения этого объекта излучением реальной сцены, которое может быть представлено в виде сферической фотографии с большим динамическим диапазоном. Этот метод был предложен в работе [1] и стал сегодня неотъемлемой частью многих продуктов визуализации, поскольку позволяет «оживлять» искусственные объекты в фотографии и видео. Для того чтобы искусственный объект выглядел естественно в том окружении, куда внедряется, необходимо осветить искусственный объект бесконечно удаленной картой окружения (далее КО). Математически это можно выразить так:
$$L(x, w_o) = \int_{\Omega} p(x, w, w_o) L_i(w) V(x, w) \cos q \, dw,$$

где x – точка на поверхности, w, w_o – направления падающего и отраженного лучей света, $L(x, w_o)$ – отраженное излучение, $p(x, w, w_o)$ – двулучевая функция отражения (далее ДФО), если объект однородный, то ДФО не зависит от x , т.е. $p(x, w, w_o) = p(w, w_o)$, $L_i(w)$ – падающее излучение от КО, бесконечно удаленного источника света не зависящего от x , $V(x, w) \rightarrow \{0,1\}$ – функция видимости, q – угол между нормалью к поверхности и направлением w , dw – дифференциальный телесный угол, а Ω – это полусфера. Существует множество методов решения этого уравнения для КО. Их можно разделить на две группы. Первые ориентированы на получение высоко реалистичных изображений и относительно медленны, как например [2]. Вторые

ориентированы на работу в реальном времени и либо менее точные, либо используют предварительные вычисления, а значит, менее гибкие.

В этой статье предложен новый иерархический базис для хранения освещения на полусфере. Разложение по новому базису может эффективно аппроксимировать высокочастотных функций на полусфере, что не могут некоторые другие базисы. Базис позволяет ускорить вычисление уравнения визуализации на несколько порядков, а также объединить вместе такие методы оптимизации моделирования освещения, как SSE трассировка лучей и кэширование излучения.

1.1 Визуализация в реальном времени

Для визуализации в реальном времени освещения от КО были предложены методы, которые использовали предварительные вычисления [3-7]. Проекция подынтегральных множителей в некоторый базис позволяет сократить расходы по памяти и быстро осуществить интегрирование (свертку). Предложено множество разных базисов для функций на сфере (полусфере): сферические гармоники [3, 4], вейвлеты [5], гауссианы [6], кусочно-постоянные функции [7]. В общем виде базис представляет собой разложение по ортонормальным функциям $\{\Psi_i\}$. Так функцию $L(w)$, заданную на Ω , можно представить как

$$L(w) \approx \sum_i L_i \Psi_i(w), \quad \text{где } L_i = \int_{\Omega} L(w) \Psi_i(w) dw.$$

Тогда, если спроецировать еще одну функцию на тот же базис: $T_i = \int_{\Omega} T(w) \Psi_i(w) dw$, то уравнение визуализации

можно представить как $\int_{\Omega} L(w) T(w) dw \approx \sum_i L_i T_i$, где

$$C_{ij} = \int_{\Omega} \Psi_i(w) \Psi_j(w) dw \quad \text{и} \quad C_{ij} = d_{ij} \quad \text{символ Кронекера.}$$

1.2 Визуализация КО трассировкой лучей.

Для получения высокорелистичных изображений традиционно используется трассировка лучей. Основной проблемой здесь является генерация выборки. Плотность выборки по возможности должна быть пропорциональна подынтегральному выражению, но генерировать выборку пропорционально произведению трех функций (ДФО, КО, видимость) достаточно сложно. В работе [8] из КО выделяются и заменяются параллельными источниками света наиболее яркие участки (например, солнце), в итоге КО модифицируется в относительно гладкую функцию. Выборку, учитывающую распределение света в карте окружения, предложили также в [9], где выборка для КО строится с

помощью метода релаксации точек. КО заменяется на несколько сотен источников света, что порождает ступенчатые тени и пропуск бликов от малых источников света. В работе [9] использовали перемежающуюся выборку из нескольких выборок по КО, чтобы устранить недостаток [8]. Оба метода требовали достаточно дорогие предварительные вычисления для построения выборки (десятки секунд). [10] позже предложил такую же эффективную, но простую, и требующую минимального времени предварительной обработки, процедуру генерации выборки, используя метод обращения кусочно-постоянного распределение КО.

Множество работ было посвящено тому, как генерировать точки выборки на нескольких множителях подынтегрального выражения. Так [11] предложил метод генерации выборки комбинируя распределение КО и распределение ДФО, что существенно улучшило оценку, если либо КО, либо ДФО обладают пиками. Далее были предложены методы [12-13] генерации выборки согласно распределению произведения КО и ДФО, что еще больше улучшило оценку. Однако эти методы давали хорошую сходимость только там, где функция видимости была 1, в тех областях, где КО была затенена, оставался большой шум. Функция видимости принимается в расчет в [14], где вначале генерировались точки с использованием метода [15], а потом создавалась большая выборка, используя цепь Маркова, что позволило учесть все три множителя подынтегрального выражения.

1.3 Когерентная трассировка лучей

Исследования последних лет в области трассировки лучей позволили поднять ее производительность в несколько раз. Сначала использование SIMD (как правило, Intel SSE) инструкций современных процессоров в [16, 17], а потом и значительные алгоритмические усовершенствования [18-19] позволили достичь высокой производительности трассировки лучей. В результате трассировщики лучей десятилетней давности могут проигрывать в десять раз по производительности современным. Поэтому узким местом производительности при визуализации трехмерных сцен становится вычисление ДФО [18]. Отдельно этой проблеме посвящена работа [20].

Однако новые алгоритмы налагают требование когерентности, т.е. для соседних лучей направление и начало не должно сильно отличаться. В противном случае эффективность новых алгоритмов может быть сведена на нет. Это требование приводит к трудностям при реализации многих «старых» алгоритмов визуализации глобального освещения, которые используют трассировку лучей, генерируя выборку методами Монте-Карло. Естественно, что такая выборка некогерентная. Тоже можно сказать и об освещении от КО, где для получения высоко реалистичных изображений традиционно использовалась Монте-Карло трассировка лучей. Поэтому прямое использование методов генерации выборки из предыдущего раздела для когерентной трассировки лучей затруднено.

1.4 Кэш излучения

Впервые кэш облучения был предложен в работе [21], что позволило в несколько раз ускорить визуализацию, поскольку только в небольшой части пикселей облучение вычислялось трассировкой лучей по полусфере, а в остальных пикселях оно экстраполировалось. Попытаемся вычислить значение

излучения в некоторой точке. Для этого ищем в кэше уже вычисленные подходящие значения для экстраполяции. Если экстраполяция возможна, то выполняем ее и переходим к следующей точке. Если нет, то вычисляем освещение, трассируя лучи по всей полусфере, и записываем вычисленное значение в кэш. Для более эффективного поиска в кэше используется октантное дерево. При движении вглубь точка приписывается к тому узлу дерева, где впервые диаметр узла становится меньше диаметра влияния вычисленной точки, который рассчитывается как аппроксимация градиента освещения.

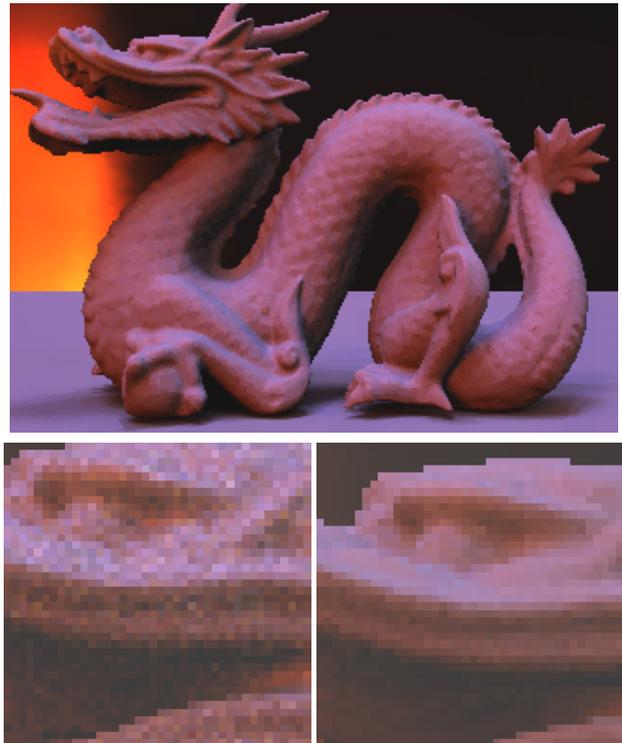


Рис. 1 Изображение, полученное с помощью кэша излучения (сверху). Увеличенное сравнение Монте-Карло интегрирования (внизу слева) и кэша излучения (внизу справа). Затрачено одинаковое вычислительное время.

В методах Монте-Карло, описанных выше, освещение в каждой точке поверхности, видимой через пиксель экрана, вычисляется независимо, что порождает высокочастотный шум (рис. 1). При использовании кэша облучения освещение вычисляется не в каждом пикселе, а только в тех, где его нельзя проэкстраполировать по уже вычисленным значениям. Экстраполяция вносит низкочастотное смещение. Поскольку вычисленное значение будет использовано для нескольких пикселей, следовательно, и его погрешность будет влиять на большее число пикселей. Поэтому для вычисления одного значения освещения от полусферы нужно использовать больше лучей в сравнении с методом Монте-Карло, хотя общее количество лучей с использованием кэша, возможно, будет меньше. Поэтому при использовании кэша лучи более когерентны, что может открыть нам возможность использовать когерентную трассировку лучей.

В работе [22] было предложено расширить метод Ward'a, чтобы использовать любые ДФО, а не только Ламбертовские. Это потребовало хранения освещения как функции от направления на полусфере. Для этого Krivanek использовал сферические гармоники. Кроме сжатия данных этот базис позволил быстро осуществлять свертку. Сферические гармоники были выбраны, поскольку проекция на базис выполняется быстро, а свертка представляет собой скалярное произведение векторов коэффициентов. Это позволило применять не только ДФО Ламберта, но и умеренно направленно-диффузные ДФО. Поскольку сферическими гармониками можно представить только низкочастотные функции, то метод не применим к остронаправленным ДФО. Погрешности представления ДФО в этом базисе заметны даже на умеренно направленно-диффузных ДФО при острых углах. Кэш излучения был взят как отправная точка проведенного исследования, но из-за ограничений сферических гармоник был предложен новый базис, который и рассматривается в этой статье.

2. ПРЕДЛАГАЕМЫЙ АЛГОРИТМ

При разработке нового базиса для расчета освещения от КО учитывалось, что он должен поддерживать высокочастотные ДФО и не создавать видимых артефактов визуализации. Для высококачественной визуализации сферические гармоники, использованные в работе [22], подходят плохо, поскольку, как уже упоминалось, не могут представлять высокочастотные функции. Кроме того, для ускорения использовалась когерентная SSE трассировка лучей. Чтобы интегрирование не стало узким местом производительности оно должно осуществляться в базисе. Поэтому при разработке нового базиса учитывались следующие требования в порядке убывания приоритета:

- скорость свертки должна быть быстрой;
- ДФО могут быть высокочастотными (т.е. иметь острый пик около направления зеркального отражения);
- КО не проецируется заранее на базис, а вычисляется непосредственно во время выполнения основного алгоритма, что позволяет использовать алгоритм для расчета освещения не только от КО, но и от других объектов сцены («размытое» зеркальное отражение);
- проекция на базис должна быть быстрой;
- реализация на центральном процессоре с возможностью применения SSE инструкций.

2.1 Отображение

Для того чтобы хранить излучение, падающее с полусферы, в линейной памяти нужно иметь отображения с полусферы на плоскость и обратно. Существует множество отображений с полусферы на плоскость, многие из которых отображают полусферу в круг, но на круге трудно построить эффективное иерархическое разбиение. Кубические карты также трудно назвать удобным отображением. Поэтому нам нужно было найти отображение из полусферы в квадрат, где существует такое простое и эффективное разбиение как quadro-дерево.

Также отображение должно сохранять площадь, т.е. если для отображения T справедливо, что $A=T(B)$, то $S(A) = kS(B)$ для любых A, B , где k – константа не зависящая от A, B . Было выбрано отображение [23]. Оно непрерывно, сохраняет

площадь, имеет небольшое искажение (рис. 2). Единственный минус этого отображения в том, что оно не дифференцируемо на диагоналях квадрата. Это свойство было бы полезно при экстраполяции.

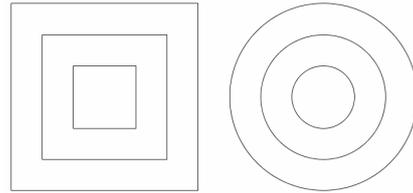


Рис. 2 Отображение из квадрата в круг. При отображении квадраты переходят в окружности.

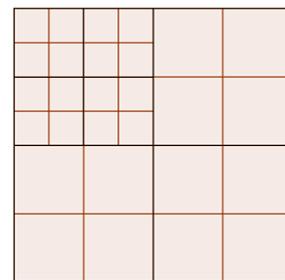


Рис. 3 Квадро-дерево, в каждом узле которого находится текстура из четырех пикселей.

Отобразив с полусферы в квадрат и обратно, можно построить иерархическое разбиение – quadro-дерево (рис. 3). Для того чтобы можно было эффективно использовать SSE инструкции, в каждом узле дерева хранятся 4 пикселя. Таким образом, одной SSE инструкцией можно обработать одну компоненту (R, G или B) сразу для четырех пикселей узла дерева.

2.2 Алгоритм проекции на базис

Теперь, имея отображение из полусферы в квадрат и обратно, можно спроецировать освещение на базис. В квадрате генерируется регулярная сетка точек, эти точки отображаются на полусферу, и трассировкой лучей вычисляется падающее излучение вдоль каждого луча. Далее это излучение отображается обратно на квадрат. В итоге имеем квадратную текстуру, которая описывает освещение на полусфере. Теперь эту текстуру необходимо сжать, чтобы сэкономить вычисления и память. Для этого снизу вверх начинаем суммировать уровни, подобно построению классических MIP тар уровней, за тем исключением, что для обычных текстур мы отфильтровываем высокие частоты гауссовским или другим фильтром, а в нашем случае усредняем только дочерние пиксели, поскольку мы строим кусочно-постоянную аппроксимацию. Одновременно вычисляется верхняя погрешность кусочно-постоянного представления исходной функции в данном узле дерева по формуле:

$$node.error = \max_{child} (child.f) - \min_{child} (child.f) + \max_{child} (child.error)$$

Когда все уровни посчитаны, начинаем движение сверху вниз, обходя quadro-дерево в глубину до тех пор, пока не

встретим уровень удовлетворяющей заданной погрешности. Проходимые узлы записываем в массив и настраиваем ссылки на дочерние элементы. Полученное дерево сохраняем в памяти, а временные регулярные уровни больше не нужны.

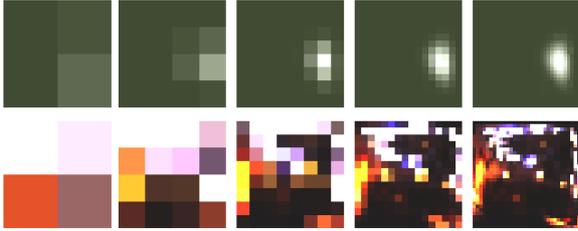


Рис. 4 Иерархическое представление функций на полусфере в новом базисе: ДФО (сверху) и карты окружения (снизу).

2.3 Свертка

Свертка двух и более функций в предлагаемом базисе осуществляется очень быстро. Поясним сначала на простом примере. Пусть требуется вычислить

интеграл $\int_{\Omega} L(w)T(w)dw$. И пусть функции L, T заданы

равномерной сеткой на Ω как $\mathbf{L} = \{L_i\}$, $\mathbf{T} = \{T_i\}$. Тогда

искомый интеграл приближенно равен $\frac{\Omega}{N} \sum_{i=1}^N L_i T_i$, или

через скалярное произведение как $\frac{\Omega}{N} \mathbf{L} \cdot \mathbf{T}$. Теперь

рассмотрим случай иерархического задания функций L, T

(рис. 4). Рассмотрим интегрирование отдельного узла quadro-дерева. Пусть требуется вычислить интеграл по k -му узлу на

области Ω_k : $\int_{\Omega_k} L(w)T(w)dw$. Тогда, если хотя бы один

из узлов листовой, то это значит, что эта функция с хорошей точностью аппроксимируется постоянной функцией на Ω_k .

Допустим, что это узел quadro-дерева для функции L . Тогда L может быть вынесена за знак интеграла, а интеграл по функции T на Ω_k равен значению в узле его quadro-дерева на той же глубине.

Для простоты изложения в этой секции предполагается, что в одном узле содержится один пиксель. Для того чтобы выполнить свертку, обходим в глубину одновременно оба quadro-дерева до тех пор, пока хотя бы один из узлов не станет листовым. В листовых узлах перемножаем значения двух функций, а также умножаем на площадь, покрываемую узлом, $L_k T_k \Omega_k$, и добавляем это значение в накопитель.

Для того чтобы применять наш базис к уравнению визуализации

$$L(x, w_0) = \int_{\Omega} p(x, w, w_0) L_i(w) V(x, w) \cos q dw \quad \text{во}$$

время трассировки лучей по полусфере получаем значения излучения уже умноженные на функцию видимости

$L_i(w) V(x, w)$. По этим значениям строим quadro-дерево.

ДФО записана во втором quadro-дереве, которое строится заранее. Множитель $\cos q$ можно включить в quadro-дерево с ДФО или с излучением.

Заметьте, что не включать ни в одно из деревьев множитель

$\cos q$, а лишь умножить на $\int_{\Omega_k} \cos q dw$ во время сверки k -

го узла, нельзя. Поскольку мы останавливаем прохождение вглубь quadro-дерева во время свертки, если любой из узлов листовой. Например, если узел k дерева L листовой, а для дерева T нет, то

$$\begin{aligned} \int_{\Omega_k} L(w)T(w) \cos q dw &\approx L_k \int_{\Omega_k} T(w) \cos q dw \neq \\ &\neq L_k T_k \int_{\Omega_k} \cos q dw. \end{aligned}$$

Это вынуждает нас включать $\cos q$ в одно из деревьев, что является не вполне удачным, поскольку уменьшает степень сжатия нашего кусочно-постоянного базиса, а значит, снижает и скорость. Но есть другое решение. Если отображение из квадрата на полусферу будет с плотностью $\frac{\cos q}{p}$, отображение Shirley позволяет легко этого добиться

[19], то при переходе с полусферы Ω на квадрат с координатами $(u, v) \in [0,1] \times [0,1]$ мы получим следующее выражение:

$$\int_{\Omega} L(w)T(w) \cos q dw = p \int_0^1 \int_0^1 L(u, v)T(u, v) dudv.$$

Для трассировки лучей лучи генерируются также с помощью такого отображения, что позволяет трассировать лучи по полусфере более значимым образом (в смысле расчета данного интеграла).

2.4 Вращение в базисе

Если ДФО хранить в глобальной системе координат, то она начинает всегда зависеть от x . Задание ДФО в локальной системе координат значительно сокращает затраты по памяти, поскольку ДФО не нужно хранить во всех точках, где она одинаковая. Но это требует процедуры вращения ДФО, заданной в базисе, из локальной системы координат в глобальную. Или наоборот: источник света поворачивается в локальную систему координат ДФО. Для сферических гармоник существует такая процедура [24]. Недавно и для вейвлетов Хаара был предложен метод вращения [25]. Наиболее просто вращение задается для сферических кусочно-постоянных базисных функций (СКПБФ) [7]. Вращение в этом базисе – это просто вращение базисных функций, как говорится в [7]. Но за такой простотой скрываются определенные трудности. Во-первых, после вращения базисных функций мы имеем разложение уже по другим базисным функциям, следовательно, не можем выполнять свертку с функцией, которая была задана в старом базисе, также легко как, если бы базисные функции совпадали. В [7] обошли эту трудность, используя таблицы суммирования для ДФО, они сворачивали ее с вращаемой КО, заданной в кусочно-постоянном базисе. Во-вторых, как

было замечено в [26], кусочно-постоянная функция не задается на сфере, а задается на плоскости, и после отображается на сферу. Если бы существовало «хорошее» отображение с плоскости на сферу, то вращение на сфере приводило бы к перемещению на плоскости. Но такого «хорошего» отображения не существует, поэтому квадрат, заданный на плоскости, отображенный на сферу и после поворота сферы отображенный обратно, искажается и перестает быть квадратом.

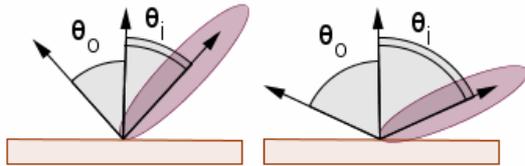


Рис. 5 При вращении выходящего вектора ДФО, отражательная способность вращается по полусфере.

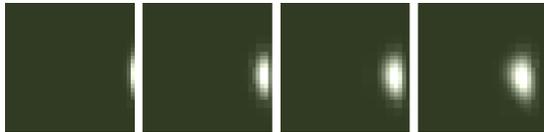


Рис. 6 Вращение ДФО в новом базисе при уменьшении угла падения.

Но, к сожалению, иерархичность предложенного базиса, которая дает ему большие преимущества в свертке, плохо подходит для вращения. Однако у ДФО есть свойство, воспользовавшись которым эти трудности можно обойти. Рассмотрим для простоты двумерную модель Фонга (рис. 5, 6). В этой модели «лепесток» рассеивания расположен симметрично вокруг направления зеркально отраженного луча. Если теперь изменить угол исходящего луча q_o , то на такой же угол изменится и направление зеркально отраженного луча, а, следовательно, и «лепесток» рассеивания. Т.е. ре-параметризация ДФО Фонга позволяет вращать отражательную способность по полусфере. Это свойство справедливо не всех ДФО, но при маленьком угле поворота оно является хорошим приближением для любых ДФО. Другими словами все ДФО слабо зависят от угла $q_i + q_o$. Это свойство также было использовано в работе [27], но для сжатого представления ДФО. Для большей точности можно ввести экстраполяционную компенсацию, коэффициент k , который можно найти из уравнения:

$$p(q_i, q_o) = k p(q_i + dq, q_o + dq) \approx k \left(p(q_i, q_o) + \frac{\partial p(q_i, q_o)}{\partial q_i} dq + \frac{\partial p(q_i, q_o)}{\partial q_o} dq \right)$$

Поскольку, по построению квадродерева ДФО, в листовых текселях значение производной мало, то компенсационный коэффициент k можно хранить в том же квадродереве. Четырехмерный случай ДФО рассматривается абсолютно аналогично. Также благодаря тому, что разбиение на кусочно-постоянные функции фиксировано (в смысле

разбиение осуществляется всегда посередине на четыре подузла, как это принято в квадродеревьях), то можно выполнять свертку произвольного числа функций точно также.

Не смотря на хорошие свойства отображения Shirley, для второго аргумента ДФО использовалось параболическое отображение [28], которое вычислительно очень простое и имеет малое искажение. Свойство квадрата здесь не так важно, поскольку по второму параметру не использовалась иерархическая дискретизация. В текущей реализации задействована простая регулярная таблица.

В отличие от [22] мы не интерполируем каждый коэффициент ДФО в базе в зависимости от исходящего вектора. Вместо этого выполняются 4 свертки с узловыми квадродеревьями, и билинейная интерполяция итогового значения выполняется уже по ним, что гораздо удобнее, чем интерполировать отдельные коэффициенты, поскольку предлагаемый базис иерархический.

3. СРАВНЕНИЕ

В отличие от сферических гармоник, предлагаемый базис может представлять высокочастотные функции. Падающее освещение практически всегда разрывное, но если ДФО является низкочастотной функцией, то отраженное излучение меняется плавно. Поэтому применение сферических гармоник возможно в случае только умеренно диффузно-направленных ДФО.

В отличие от вейвлетов Хаара, предлагаемый базис хранит на верхних узлах дерева «избыточную» информацию, которая позволяет быстрее осуществлять свертку.

В отличие от СКПБФ, предлагаемый алгоритм хранит ДФО в сжатом виде, что позволяет экономить память и осуществлять свертку быстрее.

В отличие от гауссианов, предлагаемый базис имеет быстрый алгоритм проекции.

	сферические гармоник	вейвлеты Хаара	СКПБФ	гауссианы	квадродерево
сжатие	среднее	сильное	среднее	сильное	среднее
время проекции	быстрое	среднее	среднее	медленно	быстрое
время свертки	быстрое	среднее	быстрое	быстрое	быстрое
вращение	быстрое	медленное	быстрое	быстрое	быстрое
точность	низкая	высокая	высокая	высокая	высокая

Табл. 1. Сравнение нового базиса с предложенными ранее базисами.

Таким образом, предлагаемый базис обладает рядом преимуществ перед каждым из упомянутых базисов для расчета освещения от карты окружения при налагаемых требованиях. Сравнение базисов представлено в табл. 1.

4. РЕЗУЛЬТАТЫ

Для того чтобы тестировать скорость работы нового базиса, он был включен в работающую систему визуализации, использующую когерентную SSE трассировку лучей и кэш излучения. Для тестов была взята типичная ДФО: сумма

Ламбертовского ДФО и ДФО Фонга, а текстура базиса ограничена разрешением 32x32 пикселя. Измерения проводились на процессоре Intel Centrino Duo 1.6 Ghz, используя одно ядро. Ошибка кусочно-постоянного представления была установлена в 5%. Измеренные времена усреднялись по трем разным сценам и двум разным КО (рис. 8). Измерения приведены в тактах процессора и являются приблизительными, поскольку точные значения зависят от контекста использования алгоритмов.

При использовании некогерентного Монте-Карло интегрирования число трассировок луча и вычисления ДФО совпадает. При использовании некогерентной трассировки лучей время вычисления ДФО составляет 535 тактов процессора или около 30% от трассировки одного луча (ДФО Фонга включает «дорогую» функцию возведения в степень pow).

Для кэша излучения сравнивали использование регулярной текстуры 32x32 пикселя и квадродерева. Свертка двух функций заданных регулярными текстурами занимает 36197 тактов, а свертка с использованием квадродерева – 338 тактов. Что в **100 раз быстрее**. При пересчете на минимальный пиксель (делим на 32x32) означает 0.33 такта.

Точное вычисление всего одного значения ДФО составляет 535 тактов. Что дает ускорение в 1621 раз.

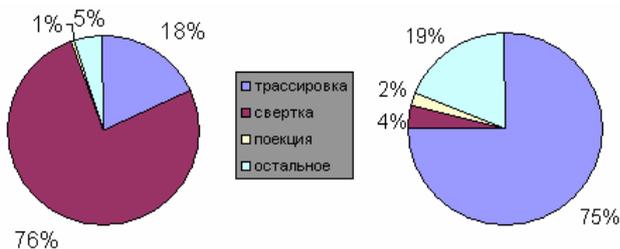


Рис. 7 Соотношение времени расчетов для разных частей алгоритма с использованием регулярной текстуры (слева) и квадродерева (справа).

Для вычисления значения отраженного излучения при экстраполяции по трем точкам из кэша необходимо провести $3 * 4 = 12$ сверток в базисе (минимально три точки из кэша умножить на 4 для билинейной интерполяции). Даже если считать, что $0.33 * 12 = 3.96$ тактов на один минимальный пиксель текстуры, то это, тем не менее, в 409 раз быстрее, чем вычисление точного значения.

Время проекции на базис текстуры 32x32 пикселя составляет 75307 цикла, что занимает около двух процентов от трассировки $32x32=1024$ лучей.

С использованием нового базиса свертка функций занимает немного времени по сравнению с трассировкой лучей, в среднем 4% (рис. 7). В итоге общее время визуализации уменьшилось более чем в два раза. Теперь можно сконцентрироваться над дальнейшей оптимизацией трассировки лучей, снижая стоимость как собственно трассировки, так и улучшая алгоритм экстраполяции, уменьшая количество необходимых точек. Поэтому следует ожидать, что доля свертки, в общем времени визуализации, будет выше, что еще больше повысит значимость ускорения с использованием нового базиса.

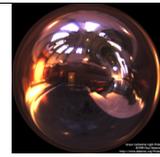
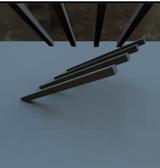
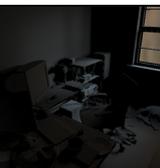
Общее время визуализации	 KO Grace Cathedral	 KO Eucalyptus Grove
Shirley shadow	 27.8 сек.	 15.4 сек.
Stanford dragon	 92.7 сек.	 79.4 сек.
Howard's room	 441 сек.	 378 сек.

Рис. 8 Время визуализации для разных сцен и карт окружения.

5. ЗАКЛЮЧЕНИЕ

Предложен иерархический базис, который позволяет представлять очень сложные, разрывные функции на полусфере. Свертка функций на полусфере в новом базисе на два порядка быстрее, чем свертка, реализованная с использованием обычных регулярных текстур. Благодаря квадродереву точность аппроксимации кусочно-постоянными функциями хорошо масштабируется под заданную точность. Базис позволяет легкую реализацию на SSE инструкциях, что и бы было сделано. Ускорение общего времени визуализации алгоритма кэширования излучения от использования нового базиса составило более двух раз.

6. БЛАГОДАРНОСТИ

Модель «Shirley shadow» была взята из набора Global illumination test scenes:

<http://www.cs.utah.edu/~bes/graphics/scenes/>.

Модель «Stanford dragon» из набора The Stanford 3D Scanning Repository:

<http://graphics.stanford.edu/data/3Dscanrep>.

Модель «Howard's room» создана Брайном Ховардом:

<http://www.bryan-howard.com/>.

7. ЛИТЕРАТУРА

- [1] Debevec, P. *Rendering Synthetic Objects Into Real Scenes: Bridging Traditional and Image-Based Graphics With Global Illumination and High Dynamic Range Photography*, in Cohen, M., ed., 'SIGGRAPH 98 Conference Proceedings', ACM SIGGRAPH, Addison Wesley, 1998, pp. 189--198.
- [2] А.Г. Волобой, В.А. Галактионов, Э.А. Копылов, Л.З. Шапиро, *Моделирование естественного дневного освещения, задаваемого изображением с большим динамическим диапазоном*. "Программирование", № 5, 2006, стр. 62-80.
- [3] Sloan, P., Kautz, J. and Snyder, J. *Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments*. ACM Transaction on Graphics (21:3), 2002, pp. 527--536.
- [4] Kautz, J., Sloan, P. and Snyder, J. *Fast, arbitrary BRDF shading for low-frequency lighting using spherical harmonics*. 2002.
- [5] Ng, R., Ramamoorthi, R. and Hanrahan, P. *Triple product wavelet integrals for all-frequency relighting*. ACM Transactions on Graphics (TOG) (23:3), 2004, pp. 477--487.
- [6] Green, P., Kautz, J., Matusik, W. and Durand, F. *View-dependent precomputed light transport using nonlinear Gaussian function approximations*. Symposium on Interactive 3D Graphics and Games', 2006, pp. 7--14.
- [7] Xu, K., Jia, Y., Fu, H., Hu, S. and Tai, C. *Spherical Piecewise Constant Basis Functions for All-Frequency Precomputed Radiance Transfer*. IEEE Transactions on Visualization and Computer Graphics (), 2007.
- [8] А.Г. Волобой, В.А. Галактионов, Э.А. Копылов, Л.З. Шапиро, *Расчет солнечного освещения, заданного изображением с большим динамическим диапазоном*. 16-ая Международная конференция по компьютерной графике и ее приложениям – Графикон-2006, Новосибирск, 2006, стр. 467-472.
- [9] Agarwal, S., Ramamoorthi, R., Belongie, S. and Jensen, H. *Structured importance sampling of environment maps*. ACM Transactions on Graphics (TOG) (22:3), 2003, pp. 605--612.
- [10] Kollig, T. and Keller, A. *Efficient illumination by high dynamic range images*. EGRW '03: Proceedings of the 14th Eurographics workshop on Rendering', Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2003, pp. 45--50.
- [11] Matt Pharr, G. H. *Infinite Area Light Source with Importance Sampling*. 2004. Internet publication.
- [12] Veach, E. and Guibas, L. J. *Optimally Combining Sampling Techniques for Monte Carlo Rendering*. 'Computer Graphics Proceedings, Annual Conference Series, 1995 (SIGGRAPH 95)', 1995, pp. 419--428.
- [13] Burke, D., Ghosh, A. and Heidrich, W. *Bidirectional Importance Sampling for Direct Illumination*. Proceedings of Eurographics Symposium on Rendering 2005', Eurographics, 2005.
- [14] Talbot, J., Cline, D. and Egbert, P. *Importance Resampling for Global Illumination*. Proceedings of Eurographics Symposium on Rendering 2005', Eurographics, 2005.
- [15] Ghosh, A. and Heidrich, W. *Correlated visibility sampling for direct illumination*. Vis. Comput. (22:9), 2006, pp. 693--701.
- [16] Wald, I. *Realtime Ray Tracing and Interactive Global Illumination*. PhD thesis 2004.
- [17] A. Adinets, B. Barladian, V. Galaktionov, L. Shapiro, A. Voloboy, *Physically Accurate Rendering with Coherent Ray Tracing*. Proceeding of GraphiCon'2006 - The 16-th International conference of Computer Graphics and Applications, Novosibirsk, 2006, pp. 8-15.
- [18] Reshetov, A., Soupikov, A. and Hurley, J. *Multi-level ray tracing algorithm*. ACM Transactions on Graphics (24:3), 2005, pp. 1176--1185.
- [19] Havran, V. *Heuristic Ray Shooting Algorithms*. PhD thesis. 2000.
- [20] А.В. Адинец, Б.Х. Барладян, А.Г. Волобой, В.А. Галактионов, Э.А. Копылов, Л.З. Шапиро, *Когерентная трассировка лучей для сцен, содержащих объекты со сложными светорассеивающими свойствами*. Препринт ИПМ им. М.В. Келдыша РАН № 107, 2005, http://library.keldysh.ru/prep_qf.asp.
- [21] Ward, G. J.; Rubinstein, F. M. & Clear, R. D. *A ray tracing solution for diffuse interreflection*. SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques', ACM, New York, NY, USA, pp. 85--92.
- [22] Krivánek, J. *Radiance Caching for Global Illumination Computation on Glossy Surfaces*. PhD thesis, Universitade Rennes 1 and Czech Technical University in Prague.
- [23] Shirley, P. & Chiu, K. (1997), *A Low Distortion Map Between Disk and Square*. Journal of graphics tools 2(3), 45-52.
- [24] Krivánek, J.; Kontinen, J.; Bouatouch, K.; Pattanaik, S. & Zára, J. *Fast Approximation to Spherical Harmonic Rotation*. In 'SCCG '06: Proceedings of the 22nd spring conference on Computer graphics', ACM Press, New York, NY, USA.
- [25] Wang, R.; Ng, R.; Luebke, D. & Humphreys, G. *Efficient Wavelet Rotation for EnvironmentMap Rendering*. In 'Rendering Techniques 2006: 17th Eurographics Workshop on Rendering', pp. 173--182.
- [26] Ng, R., Ramamoorthi, R. and Hanrahan, P. *All-frequency shadows using non-linear wavelet lighting approximation*. ACM Transactions on Graphics (TOG) (22:3), 2003, pp. 376--381.
- [27] Ramamoorthi, R. *Frequency Space Environment Map Rendering*. SIGGRAPH, ACM Transactions on Graphics (21:3), 2002, pp. 517--526.
- [28] Heidrich, W. & Seidel, H. *Realistic, hardware-accelerated shading and lighting*. SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques', ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, pp. 171--178.

A new hierarchical basis for hemisphere lighting

Abstract

The paper is devoted to new hierarchical basis for hemisphere lighting. The basis allows performing fast convolution and fast projection to the basis. Due to quad-tree nature it is capable to represent all-frequency hemisphere functions. It is useful for

computation of illumination by ray tracing from a HDRI environment. We achieved few order of magnitude acceleration of convolution from native implementation. The convolution algorithm allows efficient implementation via SSE instructions.

Keywords: *quad-tree, ray tracing, hemisphere basis, HDRI environment, SSE.*

About the authors

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