

Быстрая фильтрация кубической карты

Хомяков И. В.

Аннотация—При моделировании объектов макромира необходимо учитывать оптические особенности фотоаппаратуры. Элементы сцены, которые находятся за пределами резко изображаемого пространства, становятся размытыми. Глубина этого пространства снижается при уменьшении расстояния до отображаемого объекта. Таким образом, удалённая геометрия оказывается вне зоны резко изображаемого пространства при маленьком расстоянии до объекта. В данной работе рассматриваются один из способов представления окружения и имитация глубины резкости для него.

Ключевые слова — кубическая карта, фильтрация, освещение, оптимизация, .

ВВЕДЕНИЕ

Современную компьютерную графику невозможно представить без понятия текстуры. Обычно 3D-моделирование объекта заканчивается накладыванием на него текстуры. Это даёт возможность получить реалистичную модель, сводя к минимуму затраты на излишне трудоёмкий и ресурсоёмкий процесс добавления новых полигонов. Например, при моделировании почвы не обязательно строить 3D-модели для камней, трещин и т.п., – достаточно использовать соответствующую текстуру.

Кроме того, текстуры широко применяются при моделировании фона и освещения. В этом случае было бы расточительством создавать 3D-модель для отдалённых объектов и детально их обрабатывать. Среди распространённых техник можно выделить сферические и кубические текстуры. В данной работе будет использоваться последнее представление. Среда имеет вид куба, шесть граней которого – текстуры неба и горизонта либо помещения. При моделировании объектов макромира необходимо учитывать оптические особенности фотоаппаратуры. Так на фотографиях удалённые объекты сцены выглядят размытыми.

В данной работе рассматривается задача преобразования кубической текстуры для моделирования макромира. Формально описывается задача с учётом физических особенностей фотоаппаратуры, описывается принцип кубических текстур и особенности их обработки, приведены три различных метода фильтрации текстур, рассмотрены способы их оптимизации и проведён анализ результатов.

ОСОБЕННОСТИ ФИЛЬТРАЦИИ

Объекты на фотографии, которые не попадают в так называемую глубину резко изображаемого пространства (ГРИП), претерпевают искажения. ГРИП на фотографическом изображении – это расстояние между передней и задней границами резко изображённого

пространства, измеренное вдоль оптической оси, в пределах которого объекты съёмки на снимке отображаются безусловно резко.

Чтобы описать природу искажений на фотографиях объектов, которые находятся за пределами диапазона ГРИП, удобно использовать понятие пятна рассеяния.

Пятно рассеяния (кружок рассеяния, circle of confusion, disk of confusion и др.) – искажённое изображение точки, образуемое реальной оптической системой. Возникает вследствие дифракции света на оправах компонентов оптической системы, а также вследствие остаточных aberrаций[3]. В технической документации под этим термином понимается нерезкое изображение изолированной точки, образуемое реальным съёмочным фотографическим объективом и принимаемое за норму при расчётах глубины резкости[1].

Пятно рассеяния представляет собой не что иное, как отклик системы на единичный импульс и может быть использовано для моделирования фильтра рассеивания. Несмотря на альтернативное название «disk of confusion», кружок рассеяния на практике отличается от идеального диска. Вне зоны фокуса, каждая светлая точка становится диском. С одним объективом этот диск оказывается светящимся равномерно, с другим — ярче по краям, с третьим — в центре. Часть объективов визуально приближают такие точки к камере, другие — отдаляют. Другими словами, диск с ярким центром и тёмными краями выглядит более мягким, чем равномерно освещённый или диск с яркими краями.

Однако эти ограничения зависят от конкретного объектива и не описывают общую модель. Поэтому ими можно пренебречь и далее считать, что отклик на единичный импульс представляет собой правильный, заполненный равномерно, диск.

Необходимо ввести параметр, который позволит формально характеризовать размытие. В качестве этой характеристики может выступить угол α , равный углу раствора конуса, как показано на рис. 1.

При расчёте проекции диска рассеяния на грань куба возникают сразу две проблемы. Первая заключается в том, что форма фильтра, вообще говоря, разная для всех пикселей грани. Это создаёт множество вычислительных проблем, связанных, например, с оптимизацией. Также необходимо учитывать проблему, связанную с тем, что проекция диска может падать на другие грани куба.

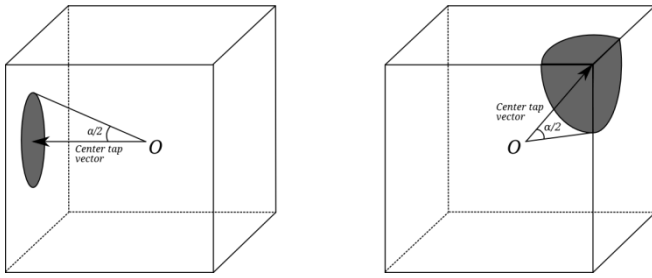


Рис 1: Иллюстрация того, что проекция диска может искажаться и выходить за пределы одной грани

АЛГОРИТМЫ ФИЛЬТРАЦИИ КУБИЧЕСКИХ КАРТ

Фильтрация по угловой мере [1]

В 2005 году исследовательская группа из ATI 3D Application Research Group в составе Торстена Шеурмана, Джона Исидоро и Джейсона Митчела (Thorsten P. Scheuermann, John R. Isidoro and Jason L. Mitchell) предложила метод фильтрации кубических карт на основе углового объёма [1]. Стандартные фильтры обрабатывали каждую грань независимо друг от друга. Новый метод позволил решить проблему передачи и обработки информации через рёбра. Кроме того новое решение не требовало специфичных особенностей от аппаратуры.

Предложенный метод был включён в утилиту под названием CubeMapGen. Данный подход полностью удовлетворяет требованиям модели и результаты его работы можно рассматривать как эталонные.

Имитация размытия

Компьютерная графика относится к тем сферам, где иногда стоит пренебречь реалистичностью ради производительности или других ресурсов. Яркими примерами могут послужить модель освещения Фонга, формат сжатия изображения JPEG и всевозможные видео-кодеки. При значительных улучшениях в остальных свойствах, можно адекватно пренебречь тем, что почти не заметит конечный пользователь. Следующий метод позволяет значительно быстрее фильтровать кубическую карту, при этом не создавая в центрах граней куба ощущения «переразмытия».

Суть предлагаемой фильтрации следующая: создаётся копия кубической текстуры, которая затем обрабатывается стационарным низкочастотным фильтром (в данном случае диск-фильтр). Результирующая текстура вычисляется путём комбинации исходной и размытой текстур следующим образом: результирующий тексель есть взвешенная сумма двух исходных.

$$p_{x,y}^r = c_{x,y} p_{x,y}^f + (1 - c_{x,y}) p_{x,y}^s,$$

$$c_{x,y} \in [0, 1].$$

Наименее размытой будет центральная часть текстуры каждой грани, углы, соответственно, – наиболее размыты. В текущей реализации размер куба был равен 1. В этом случае коэффициент $c_{x,y}$ вычисляется по следующей формуле:

$$c_{x,y} = \frac{1}{\sqrt{3}} \sqrt{0,5^2 + (x - 0,5)^2 + (y - 0,5)^2}$$

Блочный метод

Mip-текстурирование и интегральные изображения предназначены для быстрого определения влияния всех текселей на текущий пиксель на синтезируемом изображении [2]. Обе техники используются в интерактивных приложениях компьютерной графики и для них допускаются неточности в подсчёте. А именно, стороны всех просчитанных заранее блоков параллельны основным осям на изображении. При аппроксимации проекции пикселя на текстуру используется «жадный» подход: блок выбирается таким образом, чтобы вся проекция входила в него. Такая стратегия определяется желанием избавиться от эффекта алиасинга.

Mip-текстурирование и интегральные изображения удобны для оптимизации анизотропных фильтров по той причине, что избавляют от необходимости перебора всех текселей, влияющих на пиксель, чтобы получить итоговый результат. То есть до непосредственно фильтрации текстуры строятся специальные структуры данных, позволяющие быстро получить общую комбинированную информацию со многих текселей. Идея блочного метода заключается в разбиении каждой грани кубической карты на фиксированное число квадратных блоков и подсчёта среднего цвета в каждом из них. Далее во время процесса свёртки, определять, включать или исключать целый блок из суммы или обрабатывать его отдельно, имеющимся в [1] методом. Три основных отношения блока к ядру приведены на рис. 2.

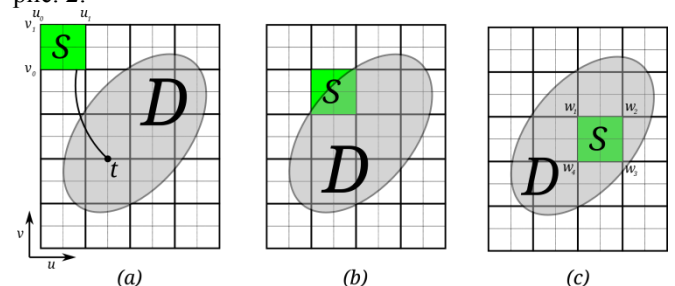


Рис 2: Три основных отношений блока к ядру. (a) - не пересекаются, (b) - пересекаются частично, (c) - блок полностью лежит в ядре

Геометрическая постановка задачи

Одним из ключевых элементов алгоритма является определение отношения квадратного блока к ядру фильтра в текущем текселе. Основная проблема здесь кроется в том, что метрики на поверхности сферы, вписанной в куб, представляющий панораму, и на грани куба не совпадают. С этим свойством и связана изменчивость формы ядра свёртки в зависимости от положения текселя.

Перечислим возможные отношения блока S и ядра D .

- $H_0: S \cap D = \emptyset$
- $H_1: S \cap D \neq \emptyset$ и $S \not\subseteq D$
- $H_2: S \subseteq D$

Проекцией t на множество S называется ближайший элемент $s \in S$:

$$Pr_S t = \operatorname{argmin}_{s \in S} (\rho(t, s))$$

Обозначим за \vec{t} вектор из начала координат по направлению к точке t , центру ядра фильтра.

Случай H_0 выполняется тогда и только тогда, когда $\cos(\vec{t}, \vec{h}) < \cos(\alpha/2)$, где $h = Pr_S t$. То есть проекция центра ядра фильтра на блок не входит в ядро.

H_2 выполняется в тех случаях, если все крайние точки множества S содержатся в ядре D . Это следует из выпуклости S и D и свойств проективных преобразований, которые сохраняют свойства выпуклости.

H_1 , соответственно, выполняется во всех остальных случаях.

Чтобы найти проекцию произвольной точки на грани кубической карты на блок, предлагается отдельно найти проекции u_0 и v_0 этой точки к основным осям с центром в точке c , направление которых задано векторами \vec{a} и \vec{b} . Тогда $Pr_S t = c + u_0 \vec{a} + v_0 \vec{b}$, где коэффициенты u_0 и v_0 определяются следующим образом:

$$u_* = \frac{(\vec{c}, \vec{t})(\vec{c}, \vec{a}) - (\vec{a}, \vec{t})}{(\vec{a}, \vec{t})(\vec{c}, \vec{a}) - (\vec{c}, \vec{t})}, \quad v_* = \frac{(\vec{c}, \vec{t})(\vec{c}, \vec{b}) - (\vec{b}, \vec{t})}{(\vec{b}, \vec{t})(\vec{c}, \vec{b}) - (\vec{c}, \vec{t})}$$

$$u_0 = \max\{\min\{u_*, u_{max}\}, u_{min}\},$$

$$v_0 = \max\{\min\{v_*, v_{max}\}, v_{min}\},$$

где $\|\vec{a}\| = \|\vec{b}\| = 1$, коэффициенты u_{min} , v_{min} , u_{max} и v_{max} фактически определяют блок S :

$$S = \{(u, v) : u_{min} \leq u < u_{max}, v_{min} \leq v < v_{max}\}.$$

Вектора $\vec{c}, \vec{t} \in R^3$ направлены из центра куба к $c, t \in P$ соответственно. Общая схема приведена на рис. 3.

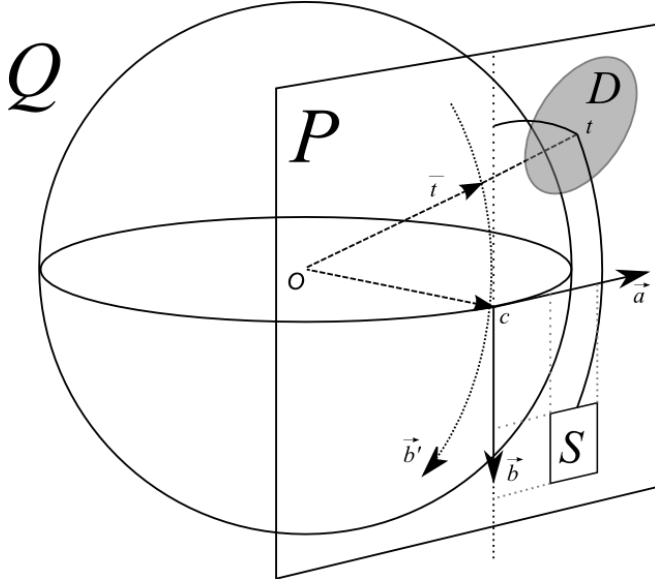


Рис 3: Общая схема вычисления проекции точки на грани куба на блок

Имея точку проекции t на блок S , и воспользовавшись условиями ситуаций H_0 , H_1 и H_2 , можем определить отношение ядра фильтра к блоку. Ситуации H_0 и H_2 позволяют избавиться от лишних вычислений. Блоки в ситуации H_1 предлагается обрабатывать аналогично представленной выше фильтрацией по угловой мере.

Дальнейшая оптимизация

Следует отметить, что размер блока напрямую влияет на актуальность использования блочного метода. На практике выяснилось, что использование блоков размером меньше 8×8 нерентабельно, поскольку все

преимущества использования заранее подсчитанных сумм не покрывают затраты на проверку ситуаций H_0 , H_1 и H_2 . Блоки слишком большого размера никогда не удовлетворяют условиям H_2 и практически всегда попадают в класс H_1 , что также ведёт к увеличению времени работы программы.

Очевидно, что блоками фиксированного размера не удаётся покрыть полностью ядро фильтра. На практике были проведены исследования для фильтров с различными значениями угла α и различных фиксированных размеров блоков. В лучших по времени результатах ситуация H_1 принималась чаще, чем $1/3$ всех случаев. Этот показатель необходимо минимизировать, поскольку обработка таких блоков является самой затратной.

Для решения этой проблемы был использован адаптивный подбор размеров блоков. Для этого задаются нижний и верхний mip -уровни. Поиск отношения H_0 , H_1 и H_2 сначала осуществляется на верхнем допустимом уровне. Затем, рекурсивно, если не достигнут нижний, блок разбивается на 4 равных части и алгоритм переходит к обработке нижележащего уровня. Если нижний порог достигнут, а отношение остаётся равным H_1 , то выполняется метод из [1].

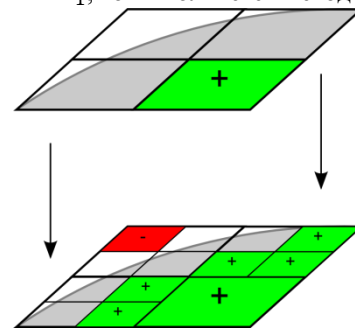


Рис 4: Адаптивный анализ блоков

Согласно результатам тестирования, адаптивный выбор размера блока увеличивал производительность в 1,5 – 2 раза на ряде примеров.

АНАЛИЗ РЕЗУЛЬТАТОВ

Приведём несколько примеров результатов работ фильтра по угловой мере, предложенного в [1] и имитации размытия. Вычисления проводились на машине с процессором Intel®Core® i7-3630QM CPU @ 2.40GHz x 8.

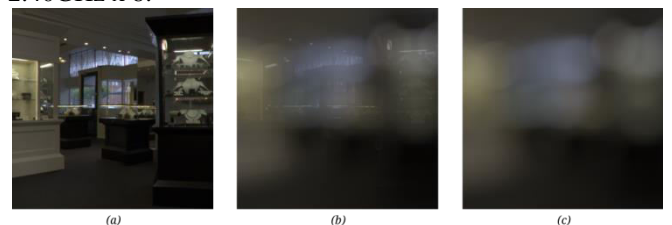


Рис 5: Результаты фильтрации. (a) - исходное изображение, (b) - имитация размытия, (c) - фильтрация по угловой мере и блочный метод

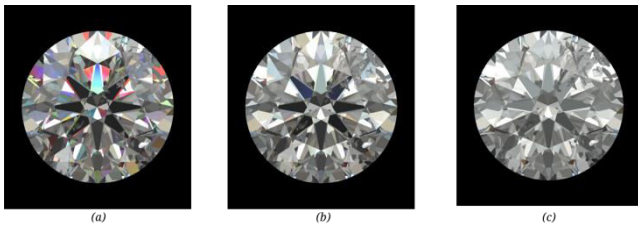


Рис 6: Синтез модели алмаза при размытии фона, который задаёт освещение сцены, низкочастотным фильтром с параметром α . (a): $\alpha=0$, (b): $\alpha=5$, (c): $\alpha=30$

Размер текстуры	α	ФУМ	БМ	ИР
512	15	14 с.	16 с.	0,5 с.
512	25	37 с.	27 с.	0,6 с.
1024	10	97 с.	69 с.	2,5 с.
1024	15	229 с.	120 с.	3 с.
1024	25	668 с.	218 с.	4 с.
2048	10	1625 с.	537 с.	17 с.
2048	15	3817 с.	982 с.	22 с.

Таблица 1: Время работы алгоритмов. Здесь приведено время, необходимое для обработки одной грани. Обозначения: ФУМ - фильтрация по угловой мере, БМ - блочный метод, ИР - имитация размытия

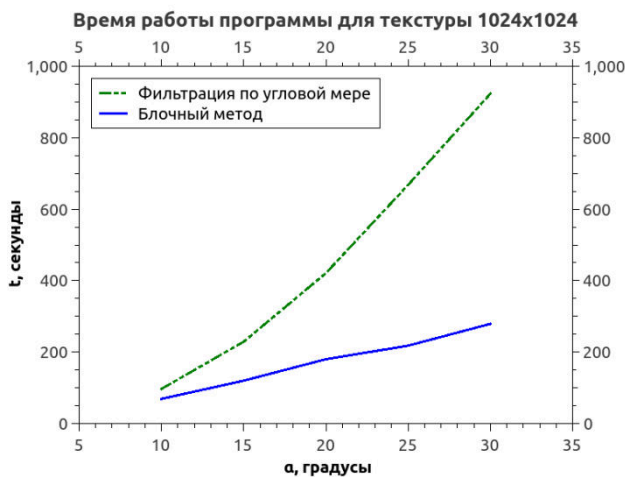


Рис 7: Сравнение фильтрации по угловой мере и блочного метода

ЗАКЛЮЧЕНИЕ

Результаты показывают, что метод, основанный на имитации размытия позволяет получить результат гораздо быстрее фильтрации по угловой мере. Однако нельзя не заметить, что ради этих показателей пришлось во многом отойти от реалистичности. Кроме того, качество размытия при имитации падает при увеличении параметра фильтра α . В этом случае контуры объектов становятся отчётливо видны. Связано

это с чрезмерным преобладанием низких частот на втором, отфильтрованном, изображении, которое сливается с исходным.

Учитывая, что требования к имитации размытия были значительно упрощены, сравнивать фильтрацию по угловой мере с ней некорректно. В то же время, блочный метод даёт реалистичные результаты, но за меньшее время. Заметные улучшения – в два раза, – достигаются уже на кубических картах размером 1024x1024.

Проблема фильтрации кубических карт остаётся открытой и в будущем планируется перенос вычислений метода фильтрации по угловой мере на графический процессор с использованием технологий CUDA. Но эта проблема может стать темой отдельного исследования.

БИБЛИОГРАФИЯ

- [1] John R. Isidoro, Jason L. Mitchell, «Angular Extent Filtering with Edge Fixup for Seamless Cubemap Filtering,» – ATI Research, 2005.
- [2] Tomas Akenine-Möller, Eric Haines, Naty Hoffman, «Real-Time Rendering,» – 3rd ed., pp. 156-174, 2008.
- [3] Яштолд-Говорко В. А. Фотосъёмка и обработка. Съёмка, формулы, термины, рецепты. Изд. 4-е, сокр. — М.: «Искусство», 1977.