

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
им. М. В. Ломоносова**

Факультет вычислительной математики и кибернетики

Кафедра математической физики

Дипломная работа

Моделирование отражения радиоволн от сложных полигональных моделей

Выполнил:
Ефремов Андрей Сергеевич,
502 группа
Научный руководитель:
к.ф.-м.н. Березин С. Б.
Рецензент:
к.ф.-м.н. Сазонов В.В.

Москва 2013

Введение

При использовании радиолокации интересной является задача моделирования распространения электромагнитного сигнала и определение зон, затененных объектом, а также областей, распространения переотраженного сигнала. Особенно это важно при использовании радиотехнических систем на протяженных объектах, так как возникают дополнительные ошибки измерения радиолокационных параметров, например, при стыковках космических кораблей и работе космических станций. [*Сазонов].

Однако, при попытках детального моделирования описанного выше процесса, мы неминуемо приходим к тому, что используемые компьютерные модели космических объектов обладают огромным количеством полигонов, а вычислительных ресурсов даже современных персональных компьютеров не хватает для их обработки.

Проведем простые предварительные расчеты. Если наша модель содержит 10 000 полигонов, то для того, чтобы определить, какие грани являются первичными для всей модели нам потребуется обработать порядка 10^8 полигонов (без алгоритмических оптимизаций, "влоб"). Мы же в работе будем использовать модель космического корабля "Союз"[*wiki], содержащий 152 000 полигонов. Это число внушительно, просто необходимы большие мощности. Поэтому для расчетов будем использовать вычисления на графических ускорителях и технологию CUDA.

Графические ускорители выросли из задач обработки и формирования изображения на экране компьютера постепенно переродившись в массивно-параллельные процессоры общего назначения. Сам термин GPU (Graphics Processing Unit) относительно новый и впервые был использован корпорацией Nvidia, в качестве обозначения того, что графические ускорители стали мощными программируемыми устройствами пригодными для решения более широкого класса задач, не связанных с графикой. [*боресков основы работы с cuda]

Первые графические ускорители представляли из себя простые растеризаторы, однако эту простую задачу делали быстрее универсального процессора, что и привело к распространению графических ускорителей. Основная причина этого – ускоритель мог обрабатывать хоть и простую, но зато масштабную работу – обрабатывать сразу много отдельных пикселей. [*боресков]

По мере развития функциональность увеличивалась. Фактически графические ускорители стали представлять из себя SIMD-процессоры (Single Instruction Multiple Data), то есть параллельные устройства, способные одновременно выполнять одну и ту же операцию над многими данными. Экспоненциальный рост производительности и функциональности дал развитие направлению GPGPU (General-Purpose computing on Graphics Processing Units).

Все это открывает новые возможности при реализации приложений, требующих больших объемов специфических вычислений. И если раньше ресурсоемкие задачи и можно было решить, то только с использованием суперкомпьютеров и кластеров, то теперь это представляется возможным на обычном пользовательском компьютере.

На сегодняшний день существуют несколько технологий для разработки приложений, использующих для вычислений графические ускорители: OpenCL, CUDA, ATI

Stream Technology. В нашей работе мы остановимся на технологии Nvidia CUDA и будем использовать её.

Тому причин несколько. Nvidia CUDA хорошо зарекомендовала себя при решении многих ресурсоемких задач: моделирование гидродинамики [*березин], волн цунами [*курако], ускорение вычисления нейронных сетей [*парубец].

Модели отражения

Радиоволна, как и любая другая электромагнитная волна, отражается от препятствий, причем препятствием является любая неоднородность электрических или магнитных параметров среды, то есть объект отражает электромагнитную энергию, в случае если проводимость, диэлектрическая или магнитная проницаемость отличается от соответствующих параметров среды. Так как от поверхности в результате процесса отражения отходит электромагнитная энергия, то можно считать, что отражатель сам является источником (вторичным) электромагнитного излучения. Это явление можно также представить себе следующим образом: при попадании волны на поверхность предмета, она в нем порождает вынужденные колебания зарядов, которые синхронны с колебаниями падающей волны. Колебания зарядов создают токи смещения и токи проводимости, и сами становятся источниками излучения. Каждый такой элементарный ток в рамках достаточного малого точечного объема можно считать источником новой сферической волны. Общий их результат таких элементарных токов (в соответствии с принципом Гюйгенса-Френеля [*wiki]) суммируется с различными фазовыми соотношениями и может принимать разные значения, а падающая волна отражается во всех направлениях. Значит, энергия переизлучается в различных направлениях неравномерно. [*лекции по радиолокации]

Радиоволны и световые волны имеют одну и ту же физическую электромагнитную природу, поэтому многие модели поведения света известные из оптики (кстати как и способы получения фотореалистических изображений компьютерной геометрии) применимы в нашем случае. Рассмотрим основные модели отражения радиоволн: зеркальное, диффузное и резонансное. Подробнее информацию об этом можно найти в [*лекции по радиолокации].

Зеркальное отражение

Условие возникновения: линейные размеры отражающей поверхности много больше длины волны ($l \gg \lambda$), а сама поверхность является достаточно гладкой ($h \ll \lambda$), где l – наименьший линейный размер цели; λ – длина волны; h – высота неровностей поверхности.

Свойства зеркального отражения известны еще с давних времен и являются следствиями применения принципа Ферма к отражающей поверхности [*wiki]: - луч отраженный, луч падающей, а также нормаль отражающей поверхности, проведенная к точке падения, лежат в одной плоскости; - угол падения равен углу отражения.

«тут картинка зеркального отражения»

Интенсивность отражённого света характеризуется коэффициентом отражения и зависит от соотношения показателей преломления сред, от угла падения и поляризации падающего пучка лучей. Количественно эту зависимость выражают формулы Френеля [*wiki].

Диффузное отражение

Условие возникновения: линейные размеры отражающей поверхности много больше длины волны ($l \gg \lambda$), а неровности поверхности имеют порядок длины волны (или больше) и расположены хаотично, то есть поверхность является шероховатой (матовой) ($h \geq \lambda$).

При диффузном отражении энергия рассеивается во всех направлениях. Для матовых поверхностей применимы законы Ламберта [*wiki]: для потока излучения, падающего нормально к матовой поверхности, мощность вторичного излучения под углом γ к нормали пропорциональна $\cos\gamma$.

$$P = P_0 \cdot \cos\gamma \quad (1)$$

«<Тут много других картинок про диффузное отражение>

Если поток излучения падает под углом α к поверхности, то

$$P = P_0 \cdot \cos\alpha \cdot \cos\gamma \quad (2)$$

где P_0 – мощность, которую принимал бы приемник, если бы облучение шло с "зенита то есть с увеличением α уменьшается перехватываемый поверхностью падающий поток, отчего уменьшается освещенность и яркость.

Условия зеркальности и диффузно отражающих поверхностей являются расплывчатыми. Здесь [*лекции по радиолокации] дается строгая формулировка и вывод критерия различия зеркальных и матовых поверхностей. Скажем лишь, что тип поверхностей зависит сразу от трех факторов: высота неровностей поверхности h , длина волны λ и угла падения ϕ .

Закон Ламберта – закон идеального рассеивания света, то есть это некоторая модель, которой можно приблизиться, но которой не существует в естественных условиях. В реальной жизни поверхности обычно дают отражение, представляемое смесью диффузной и зеркальной составляющих.

«очередной рисунок можно из Википедии с одним максимум ДФО – смесь диффузн и зеркальн»

Кроме того, на практике, часто встречаются материалы имеющие несколько максимумов под разными углами и дающие в соответствующих направлениях из-за этого блеск (glossy materials).

«<glossy-materials>

Все вышеперечисленные свойства можно смоделировать воспользовавшись функцией, характеризующей степень отражения в данном направлении. Такая функция называется ДФО – двулучевая функцией отражения (BRDF – Bidirectional Reflection Distribution Function). Так для диффузных моделей ДФО равна константе.

«<картинка из лекция 9 игнатенко>

Именно эту функцию мы и будем использовать в нашей работе. Таким образом достигается некоторая универсальность – наша программа сможет моделировать не только какой-то один конкретный вид отражения, а появляется возможность это задать в достаточно независимом виде.

Резонансное отражение

Список моделей отражения был бы не полон если не упомянуть резонансное отражение. Как уже было отмечено, падающая волна создает вынужденные колебания свободных или связанных зарядов в поверхности предмета. Тело, способное переизлучать электромагнитную волну, обладает собственной частотой колебаний частиц, несущих электрический заряд. Причем, если частота колебаний падающей волны совпадает с собственной частотой колебаний в поверхности, то имеем не что иное как явление резонансного отражения. В этом случае появляется ярко выраженная направленность вторичного излучения. Подробнее об этом явлении можно прочитать в [*лекции по радиолокации]

Математическая постановка задачи

Предположим, у нас есть ненаправленная антенна, представляемая точечным источником (расстояние до поверхности значительно больше размеров антенны) с излучающей мощностью P_0 . Тогда вся энергия распределяется равномерно по всей поверхности некоторой сферы радиуса R , площадь поверхности которой равна $4\pi R^2$. Таким образом, источник электромагнитного излучения создает плотность потока мощности

$$J = \frac{P_0}{4\pi R^2} \quad (3)$$

Следовательно, каждая точка поверхности $M(x, y, z)$, которая располагается в зоне прямой видимости, получает от источника излучения в секунду энергию мощности на единицу площади равную

$$dP_{in} = \langle \vec{J}(x, y, z) \cdot d\vec{\sigma} \rangle = J(x, y, z) \cdot d\sigma \cdot \cos\phi, \quad (4)$$

где $d\sigma$ – окрестность точки $M(x, y, z)$, а ϕ – угол между вектором $\vec{J}(M)$ и нормалью к поверхности $d\sigma$, направленной в противоположную сторону от источника излучения.

Значит, учитывая (4), первично-освещенная поверхность получает в секунду энергию мощности, равную

$$P_{in} = \int_S \langle \vec{J}(x, y, z) \cdot d\vec{\sigma} \rangle, \quad (5)$$

где S – первично-освещенная поверхность.

Часть энергии попадаемые в точки первично-освещенных поверхностей поглощаются, часть отражается. Точки, отражающие электромагнитные волны ведут себя как вторичные источники электромагнитного излучения, а значит вышепроделанные умозаключения к ним тоже применимы.

После многократного отражения и переотражения получаем, что для каждой точки $M(x, y, z)$ справедливо

$$J_{out}(x, y, z, \vec{\omega}) = \int_{\Omega} J_{in}(x, y, z, \vec{\theta}) \cdot f(x, y, z, \vec{\omega}, \vec{\theta}) \cdot \langle -\vec{\theta} \cdot \vec{n} \rangle \cdot d\theta, \quad (6)$$

где J_{out} – поток из точки (x, y, z) в направлении вектора $\vec{\omega}$, а J_{in} – поток входящего в точку (x, y, z) излучения из направления вектора $\vec{\theta}$. В данном уравнении интеграл взят по Ω – совокупности входящих направлений в точку $M(x, y, z)$, θ – телесный угол, порожденный вектором $\vec{\theta}$. Способ отражения от поверхности, а также соотношение отраженной энергии и поглощенной задано в общем случае двулучевой функцией отражения $f(x, y, z, \vec{\omega}, \vec{\theta})$.

«<Картинка из истории развития глобального освещения>

Уравнение (6) является частным случаем более общего уравнения, называемого уравнением рендеринга [wiki]. Существует несколько подходов к его решению. Рассмотрим их.

Обзор методов решения уравнения рендеринга

Алгоритм «бросания лучей» ray-casting

Фактически, данный метод учитывает только первично-освещенные точки поверхности. Из точки выпускается луч по направлению к источнику. Если этот луч имеет пересечения с каким-либо фрагментом поверхности, то значит, что точка, из которой был выпущен луч, находится в тени. В ином случае, рассчитываем долю энергии получаемую точкой. [*история рендеринга]

«красивая картинка»

Алгоритм visibility ray-tracing и трассировка фотонов

Предыдущий алгоритм учитывал только первично-освещенные точки. Повысив количество

Метод Монте-Карло (на базе трассировки лучей)

Все пространство разбивается равномерной сеткой. Количество узлов в трехмерном пространстве при этом равно N^3 . Узлы, по которым ведется интегрирование, выбираются случайно. Соответственно, при увеличении числа узлов интеграл, вычисленный методом Монте-Карло, приближается к точному значению. Но при недостаточном количестве узлов получаем шумы.

Алгоритм излучательности (radiosity)

Рассмотрим частный случай уравнения рендеринга для диффузных материалов. Двухлучевая функция отражения в этом случае представляет собой константу и уравнение сводится к тривиальному.

$$J_{out}(x, y, z, \vec{\omega}) = J_{emit}(x, y, z, \vec{\omega}) + \int_{\Omega} J_{in}(x, y, z, \vec{\theta}) \cdot f(x, y, z, \vec{\omega}, \vec{\theta}) \cdot \langle -\vec{\theta} \cdot \vec{n} \rangle \cdot d\theta, \quad (7)$$

где J_{emit} – излученная мощность точкой. Как мы уже упоминали, для диффузных материалов, значения ДФО не зависят ни от входящих, ни от исходящих направлений

$$f(x, y, z, \vec{\omega}, \vec{\theta}) = \frac{\eta(x, y, z)}{\pi} \quad (8)$$

$$\pi \cdot J_{out} = \pi \cdot J_{emit} + \eta \cdot \int_{\Omega} J_{in}(x, y, z, \vec{\theta}) \cdot \langle -\vec{\theta} \cdot \vec{n} \rangle \cdot d\theta \quad (9)$$

Далее, преобразуя интеграл по телесному углу в интеграл по всем точкам поверхности сцены, а также разбивая поверхность на непересекающиеся полигоны с

постоянной излучательностью, в конечном итоге, получаем следующую систему линейных уравнений

$$B_i = E_i + \eta_i \cdot \sum_j F_{ij} B_j, \quad (10)$$

где B_i – результирующая излучательность полигона, E_i – излучаемая энергия полигоном, η_i – коэффициент отражения, F_{ij} – форм-фактор.

Подробнее с преобразованиями и самой реализации метода можно ознакомиться здесь [*излучательность артикл 219 рейтресинг]

Минус данного метода в том, что он позволяет моделировать, в принципе, конкретный тип материала – диффузный. В реальной жизни, как уже упоминалось, это скорее исключение, чем правило.

Метод распределенной трассировки лучей

Алгоритмы

Составим некоторый алгоритм решения нашей задачи. Напомним, что никаких оптимизаций в алгоритме не проводилось, то есть расчеты велись "влоб". Так как это никоим образом не мешает анализу и сравнению производительности процессора и графического ускорителя, а также даст некоторую верхнюю практическую оценку вычисления задачи.

Пересечение отрезка и многоугольника

Задача определения пересечения отрезка и многоугольника разбивается на следующие этапы:

1. Определить точку пересечения M плоскости многоугольника и прямой, содержащей отрезок.
2. Находится ли точка пересечения M внутри многоугольника.
3. Находится ли точка пересечения M внутри отрезка.

Первичные грани

1. Берем полигон модели α .
2. Определяем её геометрический центр – точку P .
3. Для всех граней модели проверяем, пересекает ли её отрезок PL (кроме исходного полигона α , центром которого является P), где L – источник электромагнитного излучения.
4. В случае отсутствия таких граней, помечаем исходный полигон α как первичный и рассчитываем какую энергию в секунду получает наша поверхность

$$P_{in} = \frac{P_0 S \cos \phi}{4\pi R^2}, \quad (11)$$

где P_0 – мощность исходного источника излучения L , ϕ – угол между вектором \vec{LP} и нормалью \vec{n} , направленной в сторону L , расстояние $R = PL$, а S – площадь полигона.

5. Повторяем шаги 1-4 для всех полигонов поверхности.

Параллелизация выполнялась на всех уровнях: как на внешнем цикле 1-4, так и на внутреннем 3. Соотношение и размерности блоков и нитей на циклы можно менять программно.

Вторичные грани

1. Проведем расчеты первично-освещенных граней.
2. Для каждой первичной грани α выполним следующие шаги:
3. Определим центр первично-освещенной грани P .
4. Для каждого полигона модели β с геометрическим центром V , если не существует элементов моделей, пересекающих отрезок PV , будем считать его вторично-освещенным (аналогично как при определении первичных).

5. Мощность, получаемая вторично-освещенным полигоном от первично-освещенного фрагмента в этом случае рассчитываем по формуле

$$\Delta P'_{in} = \frac{P_{in} f S' \cos \phi'}{2\pi R'^2}, \quad (12)$$

где P_{in} – получаемая мощность первичной грани от источника излучения, f – двучеловечевая функция отражения ($P_{in} \cdot f = P_{out}$ в направлении проверяемой грани), S' – площадь полигона β , расстояние $R' = PV$, угол ϕ' между вектором $\vec{P}\vec{V}$ и нормалью \vec{n}' к полигону β , направленной в сторону точки P .

Выполнив весь внешний цикл 2-5 получим

$$P'_{in} = \sum_j \Delta P'_{in_j}, \quad (13)$$

то есть влияние всех первичных граней сложится.

Параллелизация выполнялась на всех уровнях: на внешнем цикле 2-5, на среднем 4 и на внутреннем 4. Соотношение и размерности блоков и нитей на циклы можно менять программно.

Если грань по итогам работы двух процедур не относилась ни к первичной, ни к вторичной, то она помечалась особым цветом.

На основе информации об освещении граней строилась сфера вокруг модели – сфера освещения. Расчет получаемой мощности полигонами сферы проводился аналогичными указанными выше способами.

Также, программа строит несколько проекций сфер на плоскость и получает изображение.

Реализация

Результаты

Результатом работы стало программное обеспечение, написанное на языке программирования C, работающее под управлением операционных систем Linux и Windows. Основные возможности программы:

1. Загружать модель из файла (Wavefront OBJ geometry format, .obj) и отображать ее в трехмерном пространстве с использованием технологии OpenGL.
2. Возможность ставить источник излучения в любую точку пространства.
3. Проводить расчеты первичных и вторично-освещенных граней как на GPU (с использованием технологии CUDA), так и на CPU.
4. Сохранять проведенные расчеты в файл (.obj). Дополнительная информация хранится в комментариях особого вида, что, фактически, не повреждает сам файл.
5. Проводить расчет сферы освещения.
6. Возможность делать проекции сферы освещения с сохранением в файл картинки (Bitmap picture, .bmp).

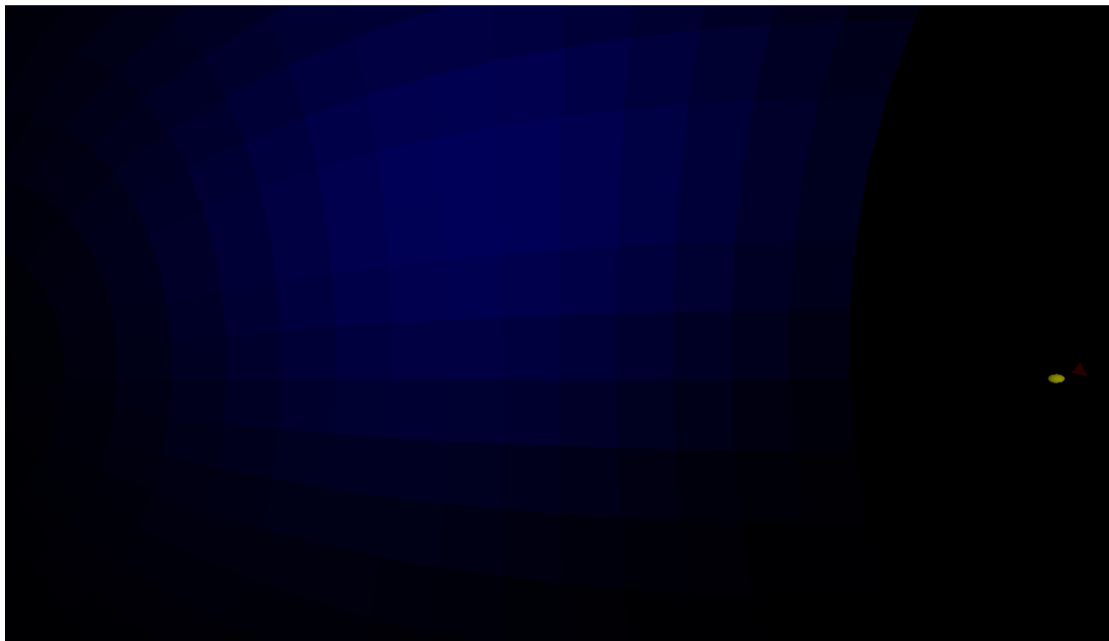
Диффузное и зеркальное

Как мы уже упомянули, в программе есть возможность задавать функцию ДФО вручную. Рассмотрим работу программы на простом примере: один треугольный полигон (красный) и один точечный источник излучения (желтая сфера).

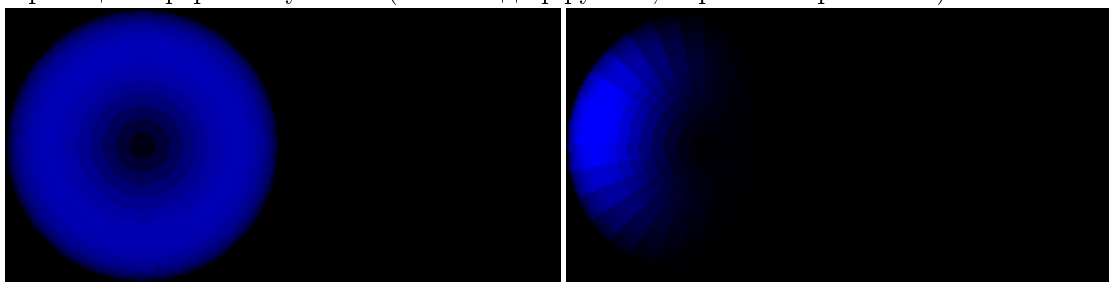
Скриншот программы для диффузного рассеивания по Ламберту:



Скриншот программы для зеркального отражения:



Проекции сферы излучения (слева – диффузное, справа – зеркальное):

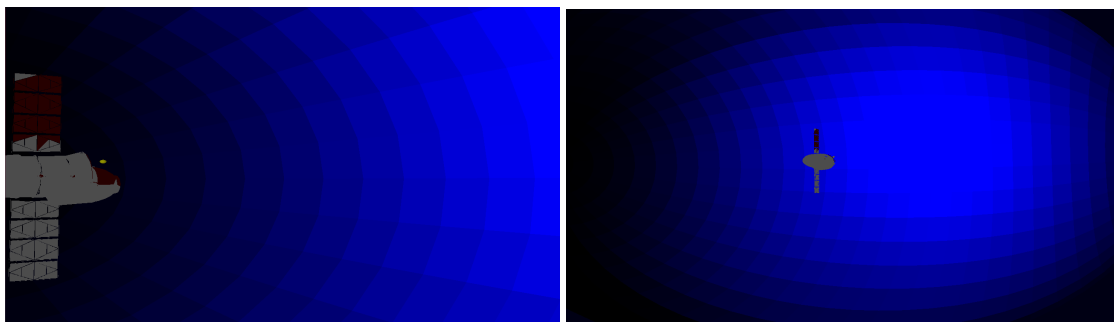


Таким образом, у нас есть возможность достаточно гибко моделировать различные материалы.

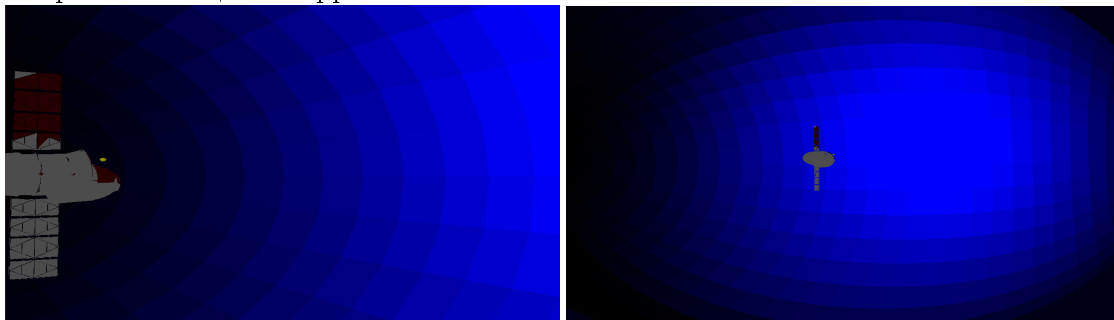
Первичные и вторичные

Рассмотрим одно положение объектов и сравним разницу между картинами в первом случае создаваемой только первично-освещенными областями и во втором – совокупностью первичных и вторичных групп полигонов.

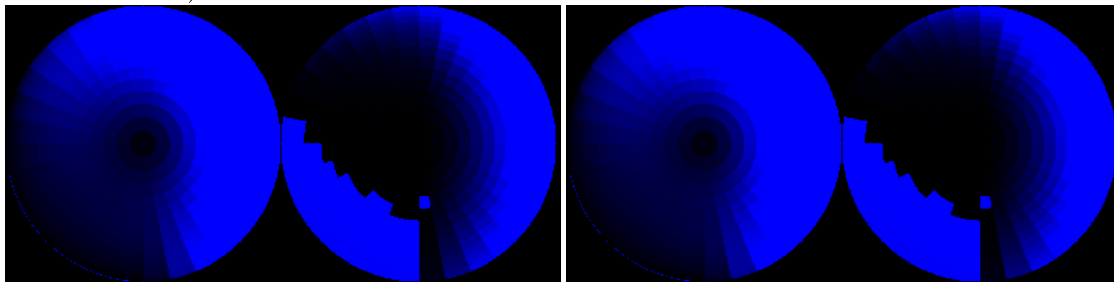
Первично-освещенные области:



Вторично-освещенные фрагменты:



Проекции сферы излучения (слева – только первичные, справа первичные и вторичные области):



Сравнение производительности

Проведем сравнение производительностей по моделированию поставленных нами задач. Помимо параллельной версии программы, запускаемой на графическом ускорителе, была написана последовательная версия программы, запускаемая на центральном процессоре.

Характеристики центрального процессора (CPU): модель Intel Core 2 Duo T6600, тактовая частота 2200 МГц.

Несмотря на то, что процессор многоядерный, программа была написана последовательно, то есть во время исполнения занято было только одно ядро.

Характеристики графического ускорителя (GPU): чипсет Nvidia GeForce GT 240M, объем 1024 mb.

Сравнение проводилось на трех сложных одинаковых полигональных моделях слонов с разной степенью детализации.

время в сек.	слон 1148	слон 10152	слон 39292
CPU	0	0	0
GPU	0	0	0
ускорение	0	0	0

Выводы

Список литературы

- [1] *N. Nikitin*, Finite-difference method for incompressible Navier–Stokes equations in arbitrary orthogonal curvilinear coordinates // J. Comput. Phys. 217 (2006) 759–781.
- [2] *Etienne Non, Roger Pierre, and Jean-Jacques Gervais*, Linear stability of the three-dimensional lid-driven cavity // Physics of Fluids 18, 084103 (2006)