

На правах рукописи

Востряков Константин Анатольевич

**Когерентные алгоритмы синтеза реалистичных  
изображений**

Специальность 05.13.11 — математическое и программное обеспечение  
вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ

диссертации на соискание ученой степени  
кандидата физико-математических наук

Москва – 2009

Работа выполнена в Учреждении Российской академии наук Институте  
прикладной математики имени М.В. Келдыша РАН.

Научный руководитель: доктор физико-математических наук  
Галактионов Владимир Александрович

Официальные оппоненты: доктор физико-математических наук,  
профессор  
Клименко Станислав Владимирович

кандидат физико-математических наук  
Игнатенко Алексей Викторович

Ведущая организация: Нижегородский государственный уни-  
верситет имени Н.И. Лобачевского

Защита состоится «\_\_\_\_\_» \_\_\_\_\_ 2009 г. в \_\_\_\_\_ часов на заседании дис-  
сертационного совета Д 002.024.01 при Учреждении Российской академии на-  
ук Институте прикладной математики имени М.В. Келдыша РАН, располо-  
женном по адресу: 125047, Москва, Миусская пл., 4

С диссертацией можно ознакомиться в библиотеке Института.

Автореферат разослан «\_\_\_\_\_» \_\_\_\_\_ 2009 г.

Ученый секретарь  
диссертационного совета,  
доктор физико-математических наук

Т.А. Полилова

## Общая характеристика работы

### Актуальность работы

Методы синтеза реалистичных изображений, основанные на моделировании распространения света, являются эффективным средством повышения производительности труда в самых разных областях, прежде всего в авиационной и автомобильной промышленности, архитектурном проектировании, создании внешнего вида будущих промышленных товаров.



а) Кабина самолета



б) Освещение стадиона



в) Автомобильная краска

Рис. 1. Моделирование распространения света в виртуальных сценах

В кабине самолета (рис. 1а) определяется читаемость приборов, расположение бликов на стеклах, теней при включенном внутреннем освещении и при разных положениях солнца. Для архитектурных сцен, например стадиона (рис. 1б), важно оценить соответствие освещенности стандартам, равномерность освещенности поля осветительными приборами. Прежде чем разработать физический прототип нового оптически сложного материала, такого как современная автомобильная краска (рис. 1в), эффективнее сначала создать компьютерную модель, увидеть как автомобиль будет выглядеть при различном освещении и окружении, оценить внешний вид краски до запуска в производство.

### Технология расчета освещения и синтеза реалистичных изображений

Для синтеза реалистичных изображений применяется метод трассировки лучей. Основная идея метода состоит в том, что лучи трассируются из вирту-

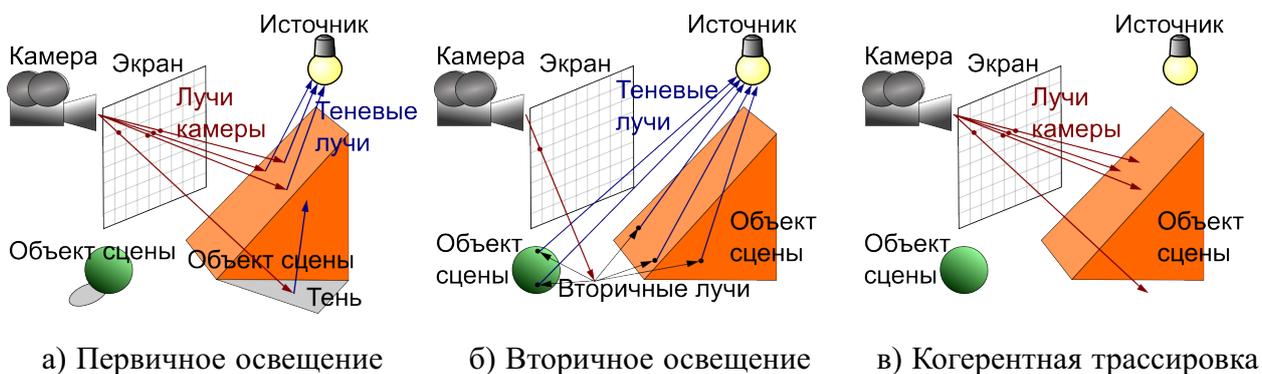


Рис. 2. Основная идея метода синтеза изображений трассировкой лучей

альной камеры сквозь пиксели экрана до ближайшего пересечения с объектами сцены (рис. 2а). В точках пересечения для расчета освещения от источников света трассируются теневые лучи. От зеркальных и прозрачных объектов лучи могут отражаться и преломляться. Таким образом формируется изображение видимое виртуальной камерой.

Освещение, непосредственно идущее от источников света, называется первичным освещением. Свет источников, отраженный от поверхностей сцены, называется вторичным освещением. Для учета этого вторичного освещения из точки, где мы его оцениваем, необходимо выполнить трассировку лучей по всем направлениям (рис. 2б). В ближайших точках пересечения лучей с поверхностями сцены необходимо снова оценить первичное освещение. То есть вычислить освещение от источников света и определить их видимость трассировкой теневых лучей. Таким образом, можно учесть дважды и более переотраженный свет от источников, попадающий в камеру.

### Синтез реалистичных изображений. Проблемы и аппаратные средства

При синтезе реалистичных изображений существует две основные проблемы: недостаточная скорость расчета и недостаточная реалистичность. С каждым годом сложность сцен растет. Увеличивается количество используемых объектов, усложняются расчеты, связанные с учетом критических свойств ма-

териалов поверхностей, источников света, применяются все более сложные модели распространения света. Все это требует большего расчетного времени. Причем увеличение вычислительной мощности компьютеров не успевает полностью компенсировать растущую потребность, поэтому используется любая возможность для ускорения расчетов.

Современный процессор позволяет применять ОКМД инструкции (одна команда — много данных). Другими словами, над несколькими потоками данных выполняются одинаковые команды. Необходимым условием применения ОКМД является когерентность. Несколько потоков данных должны обрабатываться одинаково. Многие алгоритмы синтеза изображений могут быть адаптированы для такой обработки. Например, лучи, выпущенные через соседние пиксели, с большой вероятностью попадут в одни и те же объекты (рис. 2в). К сожалению, такая когерентность не всегда получается автоматически. Необходимо адаптировать или разработать когерентные алгоритмы синтеза изображений. Данная работа посвящена разработке таких алгоритмов.

### **Цель диссертационной работы**

Исходя из потребностей промышленности в моделировании освещения и синтезе реалистичных изображений, требований работы системы на массовых компьютерах, и недостаточности имеющихся скоростей расчета, возникают следующие цели и задачи.

Целью работы является исследование и разработка когерентных алгоритмов и программных средств для ускорения расчетов освещенности и синтеза реалистичных изображений.

Основные задачи работы:

- разработка и программная реализация эффективных алгоритмов когерентной трассировки лучей, когерентного моделирования источников света и оптических свойств материалов поверхностей;

- разработка когерентных версий базовых алгоритмов синтеза реалистичных изображений: алгоритмов устранения ступенчатости изображения и расчета вторичного освещения;
- создание программного комплекса синтеза изображений с использованием ОКМД инструкций.

### **Научная новизна**

Впервые разработан адаптивный алгоритм устранения ступенчатости изображения, представленный в главе 3, для когерентной четырехлучевой трассировки лучей, что позволяет использовать ОКМД инструкции процессора для ускорения расчетов.

Предложен новый алгоритм расчета вторичной освещенности, представленный в главе 4. В отличие от существующих решений алгоритм может работать с высокочастотными функциями отражения и освещения, а также может быть эффективно реализован на многопроцессорных системах и процессорах с ОКМД инструкциями. Разработанный алгоритм дает ускорение вычислений в несколько раз в сравнении с методом обратной трассировки Монте-Карло при сравнимой точности расчета. А при использовании вместе с Монте-Карло трассировкой дает несмещенное решение, в несколько раз ускоряя сходимость.

### **Практическая значимость**

Когерентные алгоритмы синтеза изображений эффективно реализуют преимущество ОКМД инструкций массовых процессоров. Для большинства сцен, используемых в промышленной и архитектурной визуализации, удалось достичь сокращения общего времени расчетов от 2.3 до 3.2 раз, отдельные этапы расчета освещения были ускорены более чем 4 раза. Такое ускорение расчета существенным образом повышает производительность труда при работе с системой визуализации.

Результаты работы были внедрены в индустриальный программный комплекс Inspirer, который находит широкое применение на промышленных предприятиях, в научных и учебных организациях России и других стран.

### **Апробация работы**

Результаты работы докладывались и обсуждались на:

- 16-й, 17-й, 18-й и 19-й международных конференциях по компьютерной графике и машинному зрению «Graphicon»;
- семинаре по компьютерной графике и мультимедиа под руководством Ю.М. Баяковского (ф-т ВМиК МГУ), Россия, Москва, 2009;
- семинаре отделения «Программирование» ИПМ им. М.В. Келдыша РАН, Россия, Москва, 2009.

### **Публикации**

По результатам работы имеются десять публикаций, включая одну статью в рецензируемом научном журнале из списка ВАК [1], 6 статей в трудах международных научных конференций [2–7] и 3 статьи в сборниках трудов всероссийских научных конференций [8] и научно-практических семинаров [9, 10].

### **Содержание работы**

**Во введении** рассказывается об актуальности и проблематике расчета освещения и синтеза реалистичных изображений. Приведены основные принципы используемых алгоритмов и спектр существующих аппаратных средств. Ставятся цели и задачи диссертационной работы. Описаны практическая значимость и научная новизна.

**В первой главе** диссертации представлен обзор технологий ускорения трассировки лучей. Трассировка лучей является основой большинства алгоритмов моделирования освещения, ориентированных на физическую корректность.

В первом разделе описаны существующие технологии ускорения трассировки лучей. Трассировка лучей является самой затратной частью синтеза реалистичных изображений. В зависимости от сцены и применяемых алгоритмов она может отнимать 60–90% общего времени визуализации. Следовательно, первоочередным объектом для ускорения является именно трассировка лучей. В этом разделе описаны сильные и слабые стороны различных подходов к ускорению трассировки лучей, обоснован выбор ОКМД инструкций для реализации системы расчета освещения и синтеза реалистичных изображений.

В втором разделе рассказывается об особенностях программной реализации с использованием ОКМД инструкций SSE (Streaming SIMD Extensions, потоковые ОКМД расширения). ОКМД инструкции SSE позволяют задействовать восемь 128-битных регистров, в каждом из которых помещается четыре числа с плавающей точкой. Над всеми числами в регистре, как правило, выполняются одинаковые операции. Если алгоритм содержит условные переходы, то обе ветви алгоритма обрабатываются последовательно. С помощью битовой маски результат склеивается. Проиллюстрируем это на простом примере (рис. 3). Алгоритм вычисления функции  $y$  от  $x$  зависит от знака  $x$ , то есть зависит от входных данных. Для вычисления функции необходимо создать битовую маску  $s$  0, где  $x \leq 0$  и 1, где  $x > 0$ . Для четырех аргументов вычисляется  $x + 1$ , и для четырех аргументов  $2x$ . С помощью маски результат совмещается. Использование маски для реализации ветвлений — это программный прием, поэтому скрытые маской вычислители работают вхолостую (показаны серым на рис. 3).

Для программирования SSE можно использовать как ассемблер, так и инт-

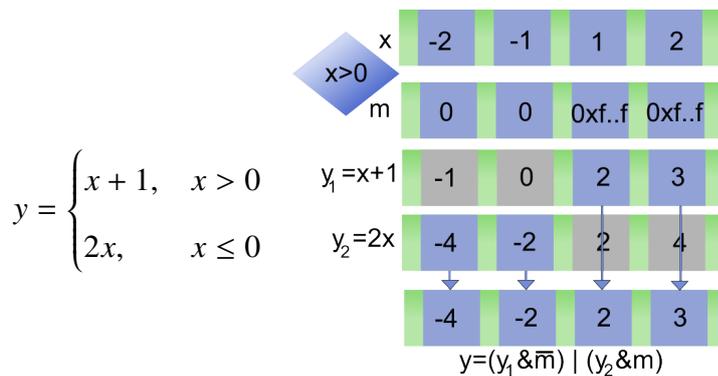


Рис. 3. Пример ОКМД реализации вычисления функции  $y(x)$ , которая зависит от входных данных

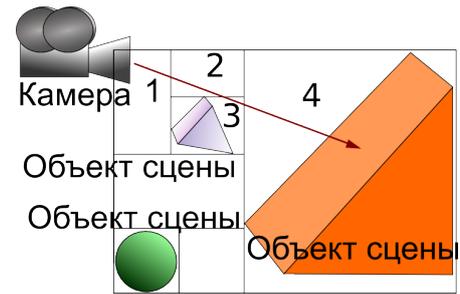


Рис. 4. Бинарное дерево

ринсики (intrinsics). Синтаксически интринсики — это функции языка C, которые воспринимаются компилятором специальным образом. Большинство интринсиков транслируются в одну команду ассемблера. Функции принимают 128 битные переменные. Компилятор сам распределяет какие переменные поместить на регистры, а какие в памяти. На интринсиках программировать намного проще, чем на ассемблере. Операция сложения четырех чисел с плавающей точкой будет выглядеть следующим образом:

```
_m128 _mm_add_ps(_m128 a, _m128 b);
```

Кроме того, для облегчения программирования и сопровождения можно использовать перегрузку арифметических операторов языка C++. Для этого был создан класс, который имеет встраиваемые (inline) перегруженные операторы работы с динамической памятью, доступа к элементам, арифметическим и логическим операциям. Для нескольких критических участков кода применялись интринсики, во всех остальных случаях использовались классы обертки.

В третьем разделе рассматривается алгоритм трассировки луча и структуры данных для его ускорения. Для того чтобы сократить количество проверок на пересечения луча с объектами сцены, использовалась ускоряющая структура данных kd-дерево (рис. 4). Это бинарное пространственное разбиение.

Ортогональность секущих плоскостей дерева значительно упрощает проверку пересечения луча и секущей плоскости. Листовым узлам дерева приписаны геометрические примитивы (треугольники), которые описывают поверхности объектов. Далее в разделе рассмотрены принципы построения и обхода kd-дерева.

В четвертом разделе рассказывается о когерентной (четырёхлучевой) SSE трассировке лучей. Четыре луча трассируются одновременно, используя SSE команды. Однако они могут проходить различные ветви дерева, поэтому необходима временная блокировка нескольких лучей из четверки. Для этого используется маска активных лучей. При достижении листового узла дерева выполняется пересечение четырех лучей с треугольниками. Часть лучей из четверки может пересечь треугольник, а часть нет. Здесь снова пользуемся приемом маскирования.

Используя когерентную трассировку лучей, на различных сценах удалось достичь ускорения трассировки лучей от 2.6 до 3.5 раз. Ускорение было достигнуто за счет использования SSE инструкций, которые могут дать ускорение до 4 раз в случае полной когерентности, тщательного подбора алгоритмов и структур данных. Достигнутая скорость визуализации не является интерактивной, но уже приближается к ней.

**Во второй главе** диссертации приведены когерентные версии алгоритмов моделирования источников света и оптических свойств материалов поверхностей, которые являются неотъемлемой частью любой системы расчета освещения и синтеза реалистичных изображений. Поскольку когерентная трассировка лучей дает ускорение в 2.6–3.5 раза, то другие части алгоритма визуализации становятся узким местом производительности. Их также необходимо ускорить.

В первом разделе описывается моделирование оптических свойств материалов поверхностей. Большинство объектов, встречающихся в повседневной

жизни, таких как автомобильные краски, дерево, пластик и ткани, обладают сложными оптическими свойствами, которые не могут быть описаны простыми эвристическими моделями. Для их учета в разработанной системе используются двунаправленные функции отражения (ДФО). ДФО описывают закон, по которому поверхность взаимодействует с попадающим на нее светом, и, по сути, представляют собой угловое распределение рассеянной поверхностью энергии света (рис. 5). ДФО являются результатом либо измерений образцов материала на специальной установке, либо моделирования по заданной микроструктуре.

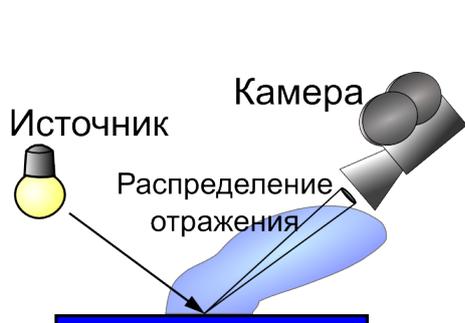


Рис. 5. Сечение ДФО

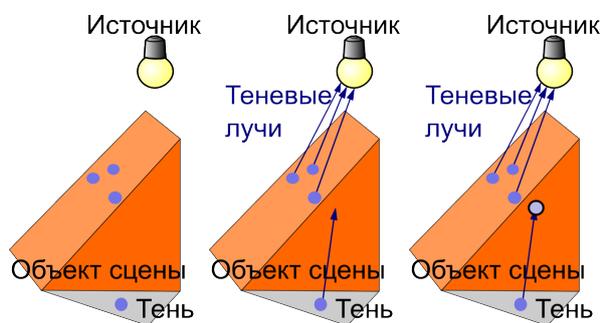


Рис. 6. Когерентное моделирование источника света

ДФО параметризуются на основе углового описания направлений освещения и наблюдения. Поскольку ДФО могут иметь резкие изменения, то они табулированы нерегулярно. Опишем алгоритм вычисления ДФО. Прежде всего вычисляется переход из глобальной системы координат падающего и отраженного лучей (в которой осуществляется их трассировка) в локальную угловую. Векторы направления лучей пересчитываются в углы системы координат ДФО. Для этого необходимо использовать обратные тригонометрические функции, которые требуют много вычислительного времени. Из-за неравномерности сетки для нахождения ячейки, в которой будет осуществляться интерполяция, применяется бинарный поиск. Внутри ячейки значения ДФО интерполируются для данных направлений лучей.

Для того чтобы эффективно использовать SSE инструкции, вычисление ДФО должно производиться для четырех точек поверхности одновременно. Были разработаны алгоритмы вычисления обратных тригонометрических функций для четырех аргументов, когерентного бинарного поиска четырех значений одновременно и интерполяции внутри ячейки.

В случае использования SSE команд бинарный поиск значительно усложняется. Если при делении пополам все искомые точки оказались с одной стороны от сечения, то, как обычно, идем в эту ветвь поиска. Если нет, то последовательно обходим обе ветви, маскируя битовой маской неактивные точки. В зависимости от размерности сетки ДФО когерентный бинарный поиск в среднем дает ускорение около 2.2 раза, что меньше 4, поскольку алгоритм содержит множество ветвлений, снижающих эффективность ОКМД команд. Для получения окончательного результата необходимо произвести линейную интерполяцию табличных значений. Удалось достичь ускорения при моделировании ДФО с использованием SSE инструкций в 3.2–3.5 раз в зависимости от размера сетки по каждому из измерений.

Во втором разделе рассказывается об использовании SSE инструкций для моделирования источников света. Под этим понимается процесс вычисления падающей интенсивности света в данной точке поверхности. Когерентное SSE моделирование источников света состоит из нескольких этапов. Вначале для четырех точек поверхности генерируются четыре точки на источнике. Далее для четырех лучей между этими точками рассчитывается яркость света, идущего от источника. После этого остается рассчитать затенение (видимость) точек трассировкой четырех лучей. Если лучи столкнулись с каким-либо объектом между этими точками и источником света, то яркость луча обнуляется (рис. 6). Определение видимости может быть реализовано с помощью описанной выше процедуры SSE трассировки лучей.

Разработанная система поддерживает несколько типов источников света.

Их моделирование имеет небольшие отличия. Для того чтобы рассчитать освещение от неточечных источников света, необходимо проинтегрировать освещенность по протяженности источника несколькими лучами. От протяженных источников получаются естественные тени с мягкими границами — полутенями. Существуют точечные источники света с гониодиаграммами. Интенсивность такого источника представляется двумерной нерегулярной сеткой, чем очень похожа на ДФО. Фактически, оба алгоритма используют несколько общих функций.

Для большинства источников света ускорение превысило 4 раза. Помогла аппроксимация некоторых функций. Например, SSE инструкции позволяют вычислять обратную величину и квадратный корень за меньшее время с точностью близкой к полной 32 битной точности с плавающей точкой. Применение приближенных функций дает достаточную точность и визуально неотличимо.

**В третьей главе** диссертации представлен новый когерентный алгоритм устранения лестничного эффекта синтезированных изображений.

В первом разделе описана проблема ступенчатости изображений, проведен обзор существующих решений и обоснована потребность в разработке нового алгоритма. При синтезе изображений методом трассировки лучей оцениваемая функция яркости неизвестна и определяется путем расчета освещенности видимой сцены и трассировки лучей из камеры через пиксели экрана. Если трассировать один луч на пиксел экрана, то получаемое изображение будет ступенчатым. Для того чтобы устранить ступенчатость, необходимо выполнить несколько действий. Во-первых, выполнить трассировку лучей через экран с плотностью в несколько раз больше, чем частота раstra, то есть несколько лучей на пиксел. Во-вторых, восстановить функцию яркости на всем экране интерполяцией. В-третьих, отфильтровать частоты, больше которых растр не может обеспечить. Ступенчатость проявляется только в тех

местах, где значение яркости резко меняется. Обычно такие пиксели содержат либо геометрические границы объектов, либо границы тени от источников света, поэтому часто применяются адаптивные схемы устранения ступенчатости.

При трассировке лучей с использованием SSE каждый раз, когда обнаруживается резкое изменение яркости, необходимо наиболее эффективно трассировать 4 новых луча, а не пытаться трассировать индивидуальные лучи. Поскольку ни один из опубликованных алгоритмов не удовлетворял этому требованию, то был разработан новый алгоритм, который позволил использовать когерентную трассировку вместе с адаптивной схемой устранения лестничного эффекта.

Во втором разделе главы 3 приведено описание нового алгоритма устранения ступенчатости изображения. Алгоритм делает предположение о том, что существуют два вида резких изменений (далее просто разрывов) функции яркости: вертикальные и горизонтальные. Это не значит, что алгоритм плохо определяет другие виды разрывов. Однако эти частые разрывы алгоритм пытается обработать наиболее эффективно. Предлагаемый адаптивный алгоритм имеет три уровня адаптации. Следующий уровень последовательно применяется в случае обнаружения резкого изменения яркости на предыдущем уровне.

Вначале для текущего блока выполняется растеризация в графическом процессоре. Это может избавить от трассировки множества дополнительных лучей для определения пикселей с геометрическими разрывами. На первом уровне используется регулярная выборка. Лучи трассируются четверками через углы пикселей (рис. 7). Алгоритм определяет вид разрыва: горизонтальный или вертикальный. Горизонтальный означает, что существует высокий градиент между точками А и В или точками С и D. Если разрыв определяется неоднозначно, то алгоритм просто выбирает горизонтальный разрыв. Такое решение не скажется на корректности, а скажется лишь на производительности. Если

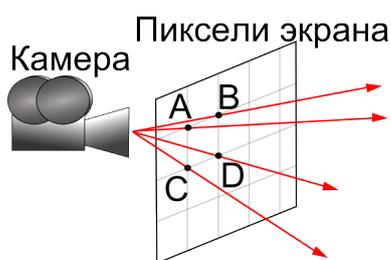


Рис. 7. Начальная 4-лучевая трассировка через углы пикселей

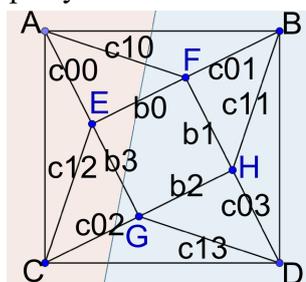


Рис. 9. Три четверки сравнений (b0–b3, c00–c03, c10–c13) для определения зон третьего уровня

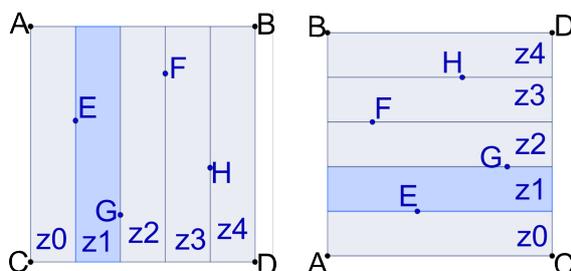


Рис. 8. Второй уровень адаптации, z0–z4 — зоны третьего уровня

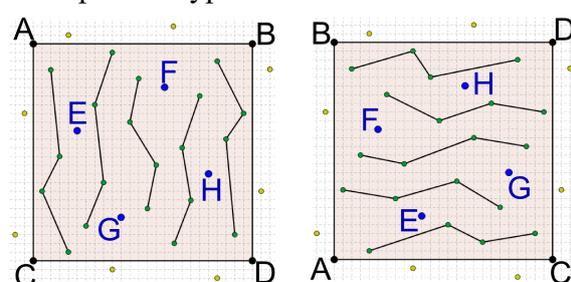
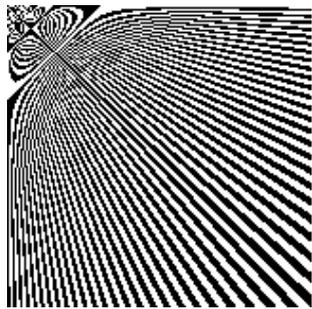


Рис. 10. Зигзаги (четверки) лучей 3-го уровня. Каждый обстреливает одну из зон

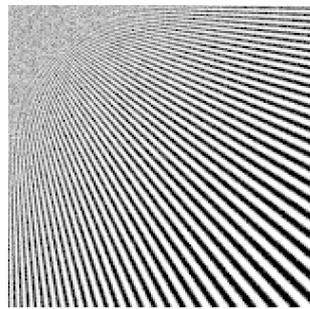
алгоритм обнаружил вертикальный разрыв, то точки ABCD поворачиваются на 90 градусов против часовой стрелки (рис. 8). Этот поворот обеспечивает дальнейшую линейность алгоритма независимо от вида разрыва. Если разрыв не обнаружен, то цвет данного пикселя определяется как среднее его угловых значений, иначе алгоритм переходит ко второму уровню адаптации для данного пикселя.

Четверка лучей трассируется через точки E, F, G, H как показано на рис. 8. Для определения разрывов между точками первого и второго уровней используем 3 четверки сравнений: b0–b3, c00–c03, c10–c13, показанных на рис. 9. Каждая четверка сравнений выполняется с помощью SSE команд над r, g, b компонентами цвета. Каждое SSE сравнение дает четырехбитную маску, которая используется как индекс в таблице определения разрывов. Значениями этой таблицы являются флаги, которые определяют зоны разрывов: z0, z1, z2, z3, z4 (рис. 8). Полученные флаги показывают зоны разрывов, где необходимо дополнительно трассировать лучи.

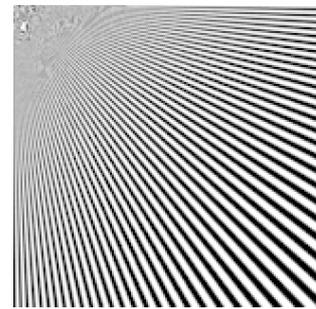
На третьем уровне через каждую разрывную зону трассируется четверка лучей зигзагом (рис. 10). Каждый зигзаг покрывает одну из разрывных зон. Максимально выстреливается 5 зигзагов на пиксел. Заметьте, что точки каждого зигзага расположены с шагом всего в 4% стороны пикселя вдоль предполагаемого высокого градиента разрыва (рис. 10). Расположение всех точек детерминировано, поэтому можно предварительно вычислить для каждой из них вес, с которым ее цвет будет просуммирован для получения цвета пикселя.



а) Один луч на пиксел.



б) Классическая реализация устранения ступенчатости.



в) Новый адаптивный когерентный алгоритм.

Рис. 11. Сложный синтетический тест. Линии сходятся в точку

Особенность устранения ступенчатости изображения состоит в том, что не существует идеального решения, с которым бы можно было сравниться. Также нет метрики, которая бы достаточно адекватно отражала человеческое восприятие ступенчатости. Поэтому за образец для сравнения по качеству была выбрана классическая реализация алгоритма устранения ступенчатости изображения. На рис. 11 показан сложный синтетический тест, где линии сходятся в точку. На рис. 11а изображение построено трассировкой 1 луча на пиксел, при этом имеем огромный муар и ступенчатость. За контрольный по качеству алгоритм устранения ступенчатости взята классическая реализация — через каждый пиксел трассируются 25 случайных лучей (рис. 11б). Муар и ступенчатость значительно уменьшены. Но из-за отсутствия адаптивности, этот алгоритм работает в 25 раз медленнее. Качество устранения ступенчатости

предложенного адаптивного метода (рис. 11в) сопоставимо с качеством при трассировке 25 случайных лучей на пиксел (рис. 11б). Тесты показали, что в типичных сценах, при использовании предложенного адаптивного алгоритма и среднем количестве лучей на пиксел 1.5–2, может быть достигнуто качество, сравнимое с 25 лучами на пиксел регулярной выборки.

**В четвертой главе** предложен новый когерентный алгоритм расчета вторичного освещения, основанный на кэшировании и нелинейной экстраполяции падающего излучения. В первом и втором разделах данной главы сформулирована проблема расчета вторичного освещения и проведен анализ существующих подходов к ее решению.

Свет источников, отраженный от поверхностей сцены, называется вторичным освещением. Для учета вторичного освещения необходимо выполнить трассировку лучей по всем направлениям. Метод называется обратной Монте-Карло трассировкой лучей. К сожалению, методы обратной Монте-Карло трассировки сходятся достаточно медленно, поэтому для получения точного решения необходимо огромное количество лучей. Для ускорения расчета большинство существующих алгоритмов накладывают ограничения на ДФО и освещение. Например, поддерживаются только ламбертовские ДФО или только плавное рассеянное вторичное освещение, что достаточно обременительно. Большинство реальных объектов имеют ДФО сложной формы, а падающее излучение неравномерно и имеет большой диапазон. В таких условиях остается мало альтернатив методам Монте-Карло трассировки лучей. Поэтому 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.

В третьем разделе четвертой главы представлен новый алгоритм когерентного расчета вторичного освещения — высокочастотный кэш излучения. Пред-

лагаемый алгоритм кэширует функцию падающего излучения и переиспользует вычисленные значения в соседних точках поверхности, что позволяет существенно сократить количество трассировок луча и вычислений функции отражения.

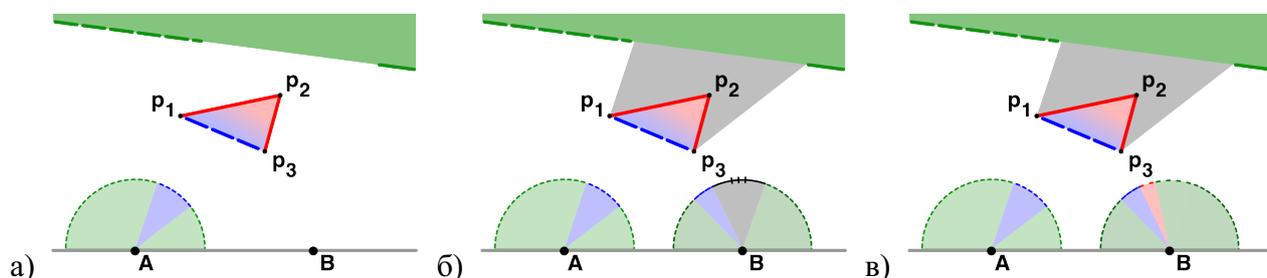


Рис. 12. Схема работы алгоритма репроекции.

Новый алгоритм репроекции работает следующим образом. На рис. 12а показано как из точки А лучи трассируются по всем направлениям. Из точки А виден дальний объект (зеленые сектора) и ближний объект (синий сектор). В точках столкновения лучей с поверхностями, вычисляем значения падающего излучения. Серым показана область неизвестности (тени), которую не видно из точки А.

Вычисленные точки разделяются на слои по глубине. Каждая точка становится п-элементом (поверхностным элементом). Под этим термином понимается то, что в точке на поверхности имеется некоторая плоская дифференциальная площадка, параллельная поверхности в данной точке. Внутри слоя соседние п-элементы, имеющие близкую нормаль, объединяются. Для полученных п-элементов вычисляются их угловые точки в системе координат полусферы. В конце все п-элементы и текстура падающего излучения сохраняются в кэше.

В точке В (рис. 12б), куда необходимо экстраполировать освещение, выполняется репроекция вычисленных в точке А и сохраненных в кэше значений падающего излучения (показаны пунктиром). Четыре значения излучения

репроецируются одновременно, используя SSE. При репроекции учитывается перекрытие и затенение. 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. Кроме того, для каждого п-элемента создается битовая маска неизвестности (тени). Она запрещает заполнение более дальних слоев в области неизвестности. Маски заполнения и неизвестности являются битовыми матрицами  $128 \times 128$  бит. Любая строка битовой матрицы — это 128 битная SSE переменная. Таким образом, битовые логические операции применяются для 128 битов одновременно.

Из-за дискретности точек после репроекции остаются бреши (рис. 12в). Бреши и зона тени заполняются трассировкой лучей. Если бы затенение не учитывалось, то обратная сторона ближнего объекта (красный сектор) была бы учтена неправильно.

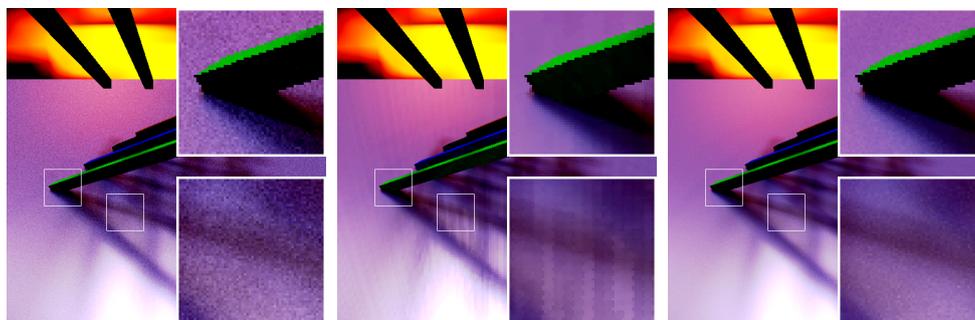


Рис. 13. Сравнение качества синтеза изображений метода Монте-Карло (слева), нового алгоритма репроекции (в середине) и нового алгоритма репроекции с последующей выборкой по значимости (справа)

Метод Монте-Карло был выбран за образец для сравнения. Он сравнивался с предложенным методом репроекции падающего излучения. После репроекции, используя полученное решение, можно продолжить трассировать лучи более значимым образом и получить несмещенную оценку, что также было протестировано.

На рис. 13 показано сравнение изображений при использовании всех трех методов. Классический контрольный метод Монте-Карло дает шум. При этом детерминизм метода превращает шум в смещение с повторяющимся шаблоном, что визуально менее приемлемо. При использовании метода репроекции вместе с Монте-Карло трассировкой точность значительно повышается, а шум в освещении снижается. Качество изображения намного выше простого метода Монте-Карло.

Смещенная версия алгоритма репроекции дает приблизительно ускорение от 3.7 до 21.6 раза. Метод репроекции в своей несмещенной версии дает ускорение в 1.7–6.1 раз. Ускорение было достигнуто как за счет использования SSE инструкций, так и за счет когерентности освещения, то есть экстраполяции. В отличие от предыдущих методов кэширования излучения, новый алгоритм репроекции точнее контролирует ошибку экстраполяции, что позволяет разбить изображение на независимые блоки и эффективно задействовать несколько потоков. Для 4 потоков удалось добиться ускорения расчета вторичного освещения в 3.8 раза.

**В заключении** представлены примеры практического использования и основные результаты работы.

Основные результаты работы состоят в следующем:

- На основе проведенных исследований существующих подходов разработаны новые когерентные алгоритмы расчета освещения и генерации изображений, эффективно использующие ОКМД инструкции массовых процессоров.
- Разработанные оригинальные реализации когерентных алгоритмов трассировки лучей, моделирования оптических свойств поверхностей и источников света, устранения ступенчатости изображения и расчета вторичного освещения позволили достичь общего ускорения расчетов в

2.3–3.2 раза при моделировании освещения и синтезе реалистичных изображений.

- На основе разработанных алгоритмов реализована программная компонента, которая привела к существенному ускорению процесса генерации изображений в индустриальном программном комплексе реалистичной визуализации и оптического моделирования Inspire.

## Список публикаций

- [1] *Б.Х. Барладян, А.Г. Волобой, К.А. Востряков, В.А. Галактионов, Л.З. Шапиро.* Применение когерентной трассировки лучей в задачах физически аккуратной визуализации // *Программирование.* — 2008. — № 5. — С. 67–80.
- [2] *К.А. Востряков.* Моделирование глобального освещения // Международная научная студенческая конференция «Студент и научно-технический прогресс» / НГУ. — Новосибирск: 2005. — С. 157–158.
- [3] *К.А. Востряков.* Ускорение финального сбора с помощью октантных текстур // Международная научная студенческая конференция «Студент и научно-технический прогресс» / НГУ. — Новосибирск: 2006. — С. 121–127.
- [4] *К.А. Востряков.* Глобальное освещение с помощью октантных текстур // Доклады 16-й Международной конференции по компьютерной графике и машинному зрению «Графикон 2006». — Новосибирск: 2006. — С. 473–476.
- [5] *К.А. Востряков, А.Г. Волобой.* Алгоритм устранения лестничного эффекта для 4-лучевой SSE трассировки лучей // 17-ая международная кон-

ференция по компьютерной графике и зрению / МГУ. — 2007. — 23-27 июня. — С. 269–272.

[6] *К.А. Востряков*. Новый иерархический базис для освещения на полусфере // Труды 18-ой международной конференции по компьютерной графике и зрению / Россия, Московский Государственный Университет. — 2008. — 23–27 июня. — С. 262–269.

[7] *К.А. Востряков*. Высокочастотный кэш излучения // 19-ая международная конференция по компьютерной графике и зрению / МГУ. — 2009. — С. 360–363.

[8] *К.А. Востряков*. Глобальное освещение стохастической трассировкой лучей // Всероссийская конференция студентов и молодых ученых «Наука. Технологии. Инновации» / НГТУ. — Новосибирск: 2004. — С. 17–18.

[9] *К.А. Востряков*. Расчет освещения от карты окружения, заданной с большим динамическим диапазоном // Материалы 11-го научно-практического семинара «Новые информационные технологии в автоматизированных системах» / М.:МГИЭМ. — 2008. — С. 24–28.

[10] *К.А. Востряков*. Нелинейная экстраполяция излучения // Новые информационные технологии в автоматизированных системах: материалы двенадцатого научно-практического семинара / М. МГИЭМ. — 2009. — С. 62–73.