

Визуализация распространения света при физически корректном моделировании освещенности

Б.Х. Барладян¹, Е.Д. Бирюков², Л.З. Шапиро³, А.Г. Волобой⁴

Институт прикладной математики им. М.В.Келдыша РАН, Москва, Россия

¹ ORCID: 0000-0002-2391-2067, bbarladian@gmail.com

² ORCID: 0000-0003-4297-6813, birukov@gin.keldysh.ru

³ ORCID: 0000-0002-6350-851X, pls@gin.keldysh.ru

⁴ ORCID: 0000-0003-1252-8294, voloboy@gin.keldysh.ru

Аннотация

Работа посвящена вопросам визуализации траекторий лучей света при физически корректном моделировании его распространения в исследуемой сцене методом стохастической трассировки лучей. Предложено сохранять лучи, получаемые в процессе трассировки, в файл, который можно рассматривать как трехмерную карту лучей. Разработаны многопоточные алгоритмы как создания таких карт лучей, так и их последующего анализа. Реализация предложенных алгоритмов на многоядерных компьютерах показала их высокую эффективность. Особое внимание уделено вопросам интеграции разработанных алгоритмов с системой автоматизированного проектирования САПР.

Ключевые слова: реалистичные изображения, глобальное освещение, оптическое моделирование, интерактивный анализ сцен, трехмерные карты лучей, стохастическая трассировка лучей, визуализация траекторий лучей.

Введение

Использование физически корректных методов расчета освещенности, моделирующих распространение света и его взаимодействие с объектами сцены, находит все большее применение в различных областях науки и индустрии. При классическом использовании этих методов для генерации реалистичных изображений [1] или для проектирования различных оптических приборов и освещающих устройств [2] результаты оптического моделирования представляются, как правило, в виде графиков, таблиц или изображений распределения таких световых характеристик как яркость, освещенность или интенсивность на виртуальных приемниках излучения. Такая форма достаточна, когда нас интересует только результат моделирования. Однако в некоторых случаях нам также необходимо знать, каким образом он был получен, или что оказало

наибольшее влияние на полученный результат. Например, при моделировании распределения яркости на поверхности клавиш телефона, разработчику необходимо понимать каким образом свет от источников света проходит по всей системе подсветки и выходит через верхнюю грань клавиши (Рис. 1). Другой актуальный пример – это анализ паразитного рассеянного света в линзовом объективе. Разработчику необходимо понимать, какая поверхность объектива и какой его линзы создает блик или засветку на изображении. Для получения исчерпывающей информации о распространении света в оптической системе удобно визуализировать траектории промоделированных лучей. Визуальное представление траекторий световых лучей в оптической системе полезно и для разработчика программного обеспечения, как средство отладки и оптимизации алгоритмов [3].

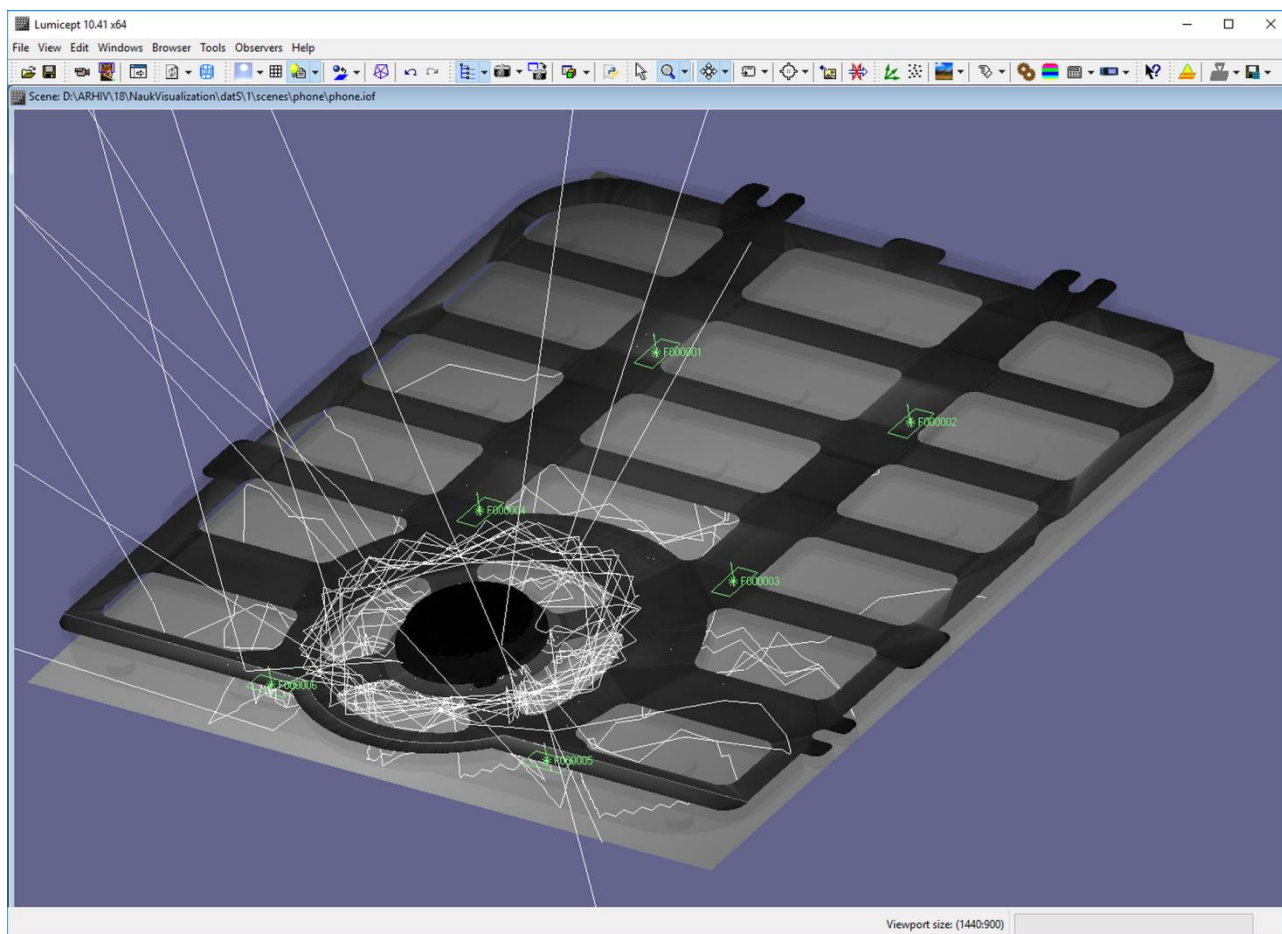


Рис. 1. Лучи, попавшие на центральную кнопку

В настоящее время генерация реалистичных изображений, основанная на физически корректном моделировании освещенности, находит применение во всех новых областях. Например, перспективным является применение физически корректных синтезированных изображений и видео последовательностей в системах обучения искусственного интеллекта [4-6]. Корректность получаемых изображений является принципиально важной во всех этих применениях. В противном случае цель не будет достигнута, и интеллектуальная система будет обучена неправильно. Однако эта физическая корректность может быть нарушена из-за ошибок в программе моделирования или ошибок пользователя при описании сцен, геометрии объектов, их оптических свойств. Визуализации распространения световых лучей является одним из наиболее эффективных средств для обнаружения и анализа такого рода ошибок [3], [7].

В силу всех этих причин визуализация траекторий световых лучей фактически стало базовой функциональностью современных систем оптического моделирования. Изначально визуальное представление траекторий лучей было реализовано в системах синтеза реалистичных изображений и оптического моделирования, разработанных в ИПМ им. М.В. Келдыша РАН [7-9], в конце 90х годов. Однако использование этих систем для решения сложных задач проектирования современных оптических приборов выявило определенные недостатки реализованных алгоритмов визуализации лучей, таких как:

- замедление трассировки лучей;
- невозможность сохранения лучей, полученных при трассировке, для последующего детального анализа;
- отсутствие визуализации траекторий лучей только для части сцены – для выбранных источников света, геометрических объектов и виртуальных измерительных приборов;

- отсутствие интеграции с промышленными системами автоматизированного проектирования (САПР).

Замедление процесса трассировки лучей была связана с используемой технологией визуализации, поскольку одним из требований к предыдущей реализации была последовательная визуализация трассируемых сегментов луча один за другим. В такой технологии алгоритм мог использовать только один поток для вычислений. Использование OpenGL для визуализации сегментов в таком режиме также неэффективно. В результате при моделировании на типичном для настоящего времени компьютере Intel Core i7-4770 3.4GHz 32GB, скорость прямой стохастической трассировки лучей для типичной сцены составляет ~1.2 миллиона лучей в секунду, в то время как визуализация трассируемых лучей происходит со скоростью всего ~1300 лучей в секунду. Добавлять сохранение лучей при такой скорости трассировки в существующий алгоритм было бессмысленно. Устранение указанных ограничений фактически требовало перепроектирования всего модуля визуализации лучей. Основными требованиями при разработке новой системы являлось эффективное использование многоядерных компьютеров как для трассировки лучей, так и для анализа результатов моделирования, а также интеграция разрабатываемой подсистемы с САПР CATIA.

Для выполнения новых требований нами были разработаны новые алгоритмы, эффективно использующие многоядерные процессоры. Новый модуль визуализации распространения света была реализована в базовой автономной системе Lumicert и в аналогич-

ной по набору функциональностей подсистеме, интегрированной в CATIA. Первые результаты докладывались на конференциях [10] и [11].

Выбор объектов для моделирования

Оптическое моделирование в САПР CATIA имеет определенную специфику, связанную с представлением геометрических объектов в сцене. Сцена в документе системы CATIA содержит большое количество геометрических объектов и источников света, не все из которых представляют интерес для пользователя в данном моделировании и визуализации лучей. Это может быть связано с желанием пользователя исключить из моделирования вспомогательные объекты или, наоборот, включить в данное моделирование только часть объектов сцены, формирующих определенный световой пучок, или использовать только ту часть сцены, где по предположению пользователя возникают определенные проблемы. В силу этих причин соответствующее моделирование в системе CATIA начинается с выбора объектов для моделирования.

Выбор объектов для моделирования — это фактически создание новой сцены из объектов уже существующей. Соответствующие диалоги для подсистемы, интегрированной в CATIA, показаны на рис 2. В левой части рис. 2 показан диалог, в котором отображаются объекты, выбранные для моделирования. Сами объекты могут быть выбраны либо непосредственно в окне визуализации сцены, либо в окне представления сцены в виде дерева объектов, показанной в правой части рис. 2.

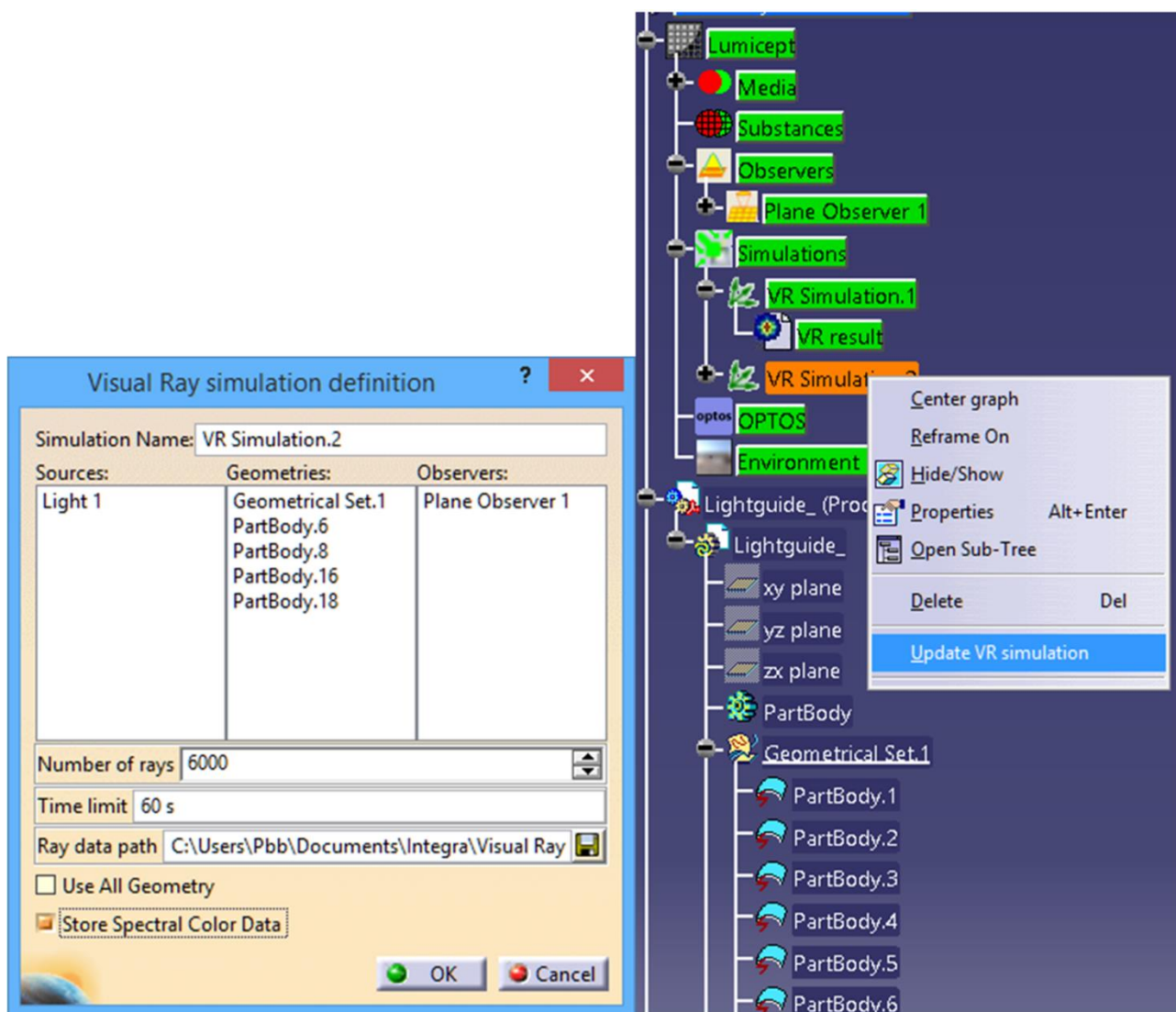


Рис. 2. Объекты, используемые при визуализации лучей, и дерево объектов сцены.

В приведенном диалоге также возможно указать сохранение данных в спектральном виде. Там же пользователь устанавливает путь для сохранения результатов моделирования, количество сохраненных лучей и временное ограничение для моделирования. После закрытия диалога заказанное моделирование отображается в дереве сцены и может быть запущено, как показано на правой части рис. 2, с помощью выпадающего диалога по нажатию правой клавиши мыши поверх соответствующего элемента в дереве сцены. Пользователь может создать неограниченное число таких моделирований для разных целей. Диалог, приведенный на рис. 2, был реализован, естественно, с помощью соответствующих средств CATIA для создания пользовательского интерфейса.

Непосредственно моделирование и расчет световых лучей выполняется в специальном модуле оптического моделирования I2 Server, построенного на базе базовой системы синтеза реалистичных изображений и оптического моделирования Lumiccept. Для представления и анализа взаимодействия с виртуальными измерительными приборами используется подсистема LumiVue [14]. Взаимодействие системы CATIA и модуля I2 Server, представлено на схеме, приведенной в верхней части на рис. 3.

CATIA формирует описание сцены в бинарном формате системы Lumiccept, сохраняет его на диске и посылает I2 Server команду VR_SCENE_LOAD (загрузи сцену), а за тем команду VR_CALCULATE (Монте-Карло трассировка лучей с сохранением их в файл для визуализации). Дополнительные

данные (путь на описание сцены, путь на файл для сохранения лучей, ограничение по времени, количество лучей и т.д.) передается при этом с использованием общей разделяемой памяти (shared memory). Схемы выполнения различных команд в I2 Server близки друг к другу. Мы здесь опишем подробно схему выполнения команды *VR_CALCULATE*, как наиболее сложную.

Выполнение команды
VR_CALCULATE:

1. CATIA сбрасывает события **ev1** и **ev2**, посылает сообщение (Windows message) *VR_CALCULATE* и ждет взведения события **ev1**.
2. I2_Server, получив сообщение *VR_CALCULATE*, извлекает из общей памяти путь на файл с описанием сцены, загружает ее, помещает в общую память дескриптор своего окна, взводит событие **ev1** и ждет взведения события **ev2**.

3. CATIA извлекает из общей памяти дескриптор окна, сбрасывает событие **ev1** и взводит событие **ev2**.
4. I2_Server, дождавшись взведения события **ev2**, выполняет стохастическую (Монте-Карло) трассировку и после завершения взводит событие **ev1**. Во время моделирования индикатор показывает степень выполнения команды: отношение числа протрассированных лучей к числу заказанных, выраженное в процентах. Результаты моделирования I2_Server записывает на диск в своем формате.
5. CATIA, ожидая взведения события **ev1**, каждые 100 миллисекунд выполняет перерисовку своего окна, делает окно I2_Server активным и самым верхним. Дождавшись взведения события **ev1**, CATIA читает с диска обновленную сцену и обновляет в соответствии с ней данные в своем документе и дереве сцены.

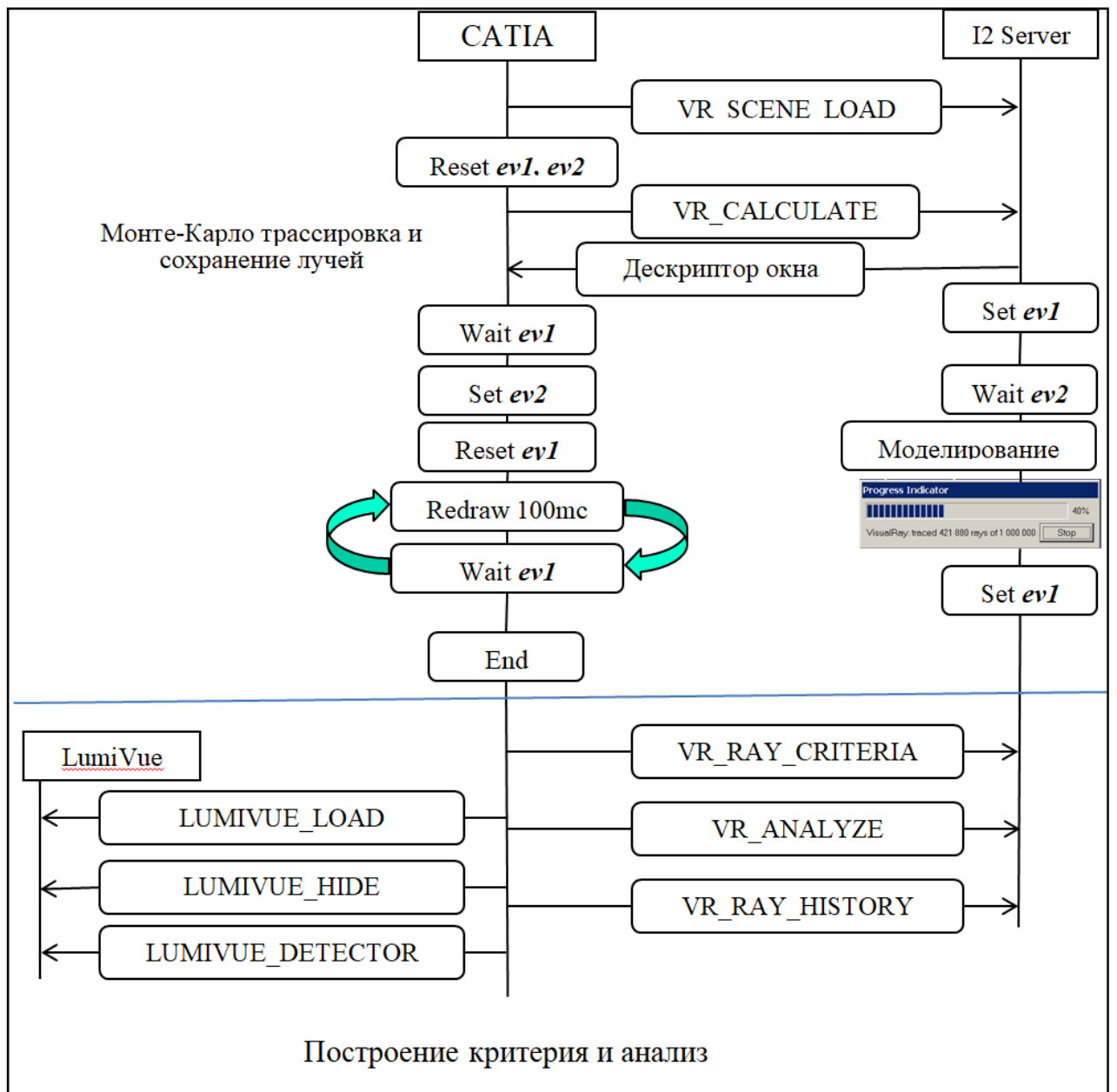


Рис. 3. Схема взаимодействия CATIA с I2 Server и LumiVue при визуализации пространства света.

Моделирование и сохранение лучей

Алгоритм сохранения лучей, полученных при трассировке, для одной порции лучей представлен на рис. 4. Фактически это та же Монте-Карло трассировка лучей, которая используется при вычислении глобальной освещенности. Лучи, построенные при Монте-Карло трассировке, будут сохраняться, если установлен соответствующий ключ моделирования. Фактически вычислительное ядро обеспечивает одно-

временное вычисление глобальной освещенности, сохраняя результаты в виде карт освещенности или значения освещенности на виртуальных измерительных приборах, и сохранение лучей. Значения освещенности на виртуальных измерительных приборах получается в результате регистрации лучей, попадающих на них. Она производится только для виртуальных приборов, специфицированных в диалоге на рис. 2. Результаты регистрации лучей сохраняются в отдельных файлах для последующего анализа при визуализации лучей. При многопоточной трассировке лучей ал-

горитм, приведенный на рис. 4, работает практически независимо в каждом вычислительном потоке для каждой вычисленной порции лучей. Единственный модуль в этом алгоритме, который нуждается в синхронизации, это запись порций лучей на диск, поскольку все лучи записываются в один файл. Эта синхронизация обеспечивается с помощью обычной критической секции

(CRITICAL_SECTION). Понятно, что алгоритм будет оставаться эффективным до тех пор, пока запись порций на диск будет производиться быстрее, чем вычислительные потоки будут их производить. Таким образом, критическим пунктом в этом формально простом алгоритме является эффективное сжатие порций лучей, полученных при Монте-Карло трассировке.

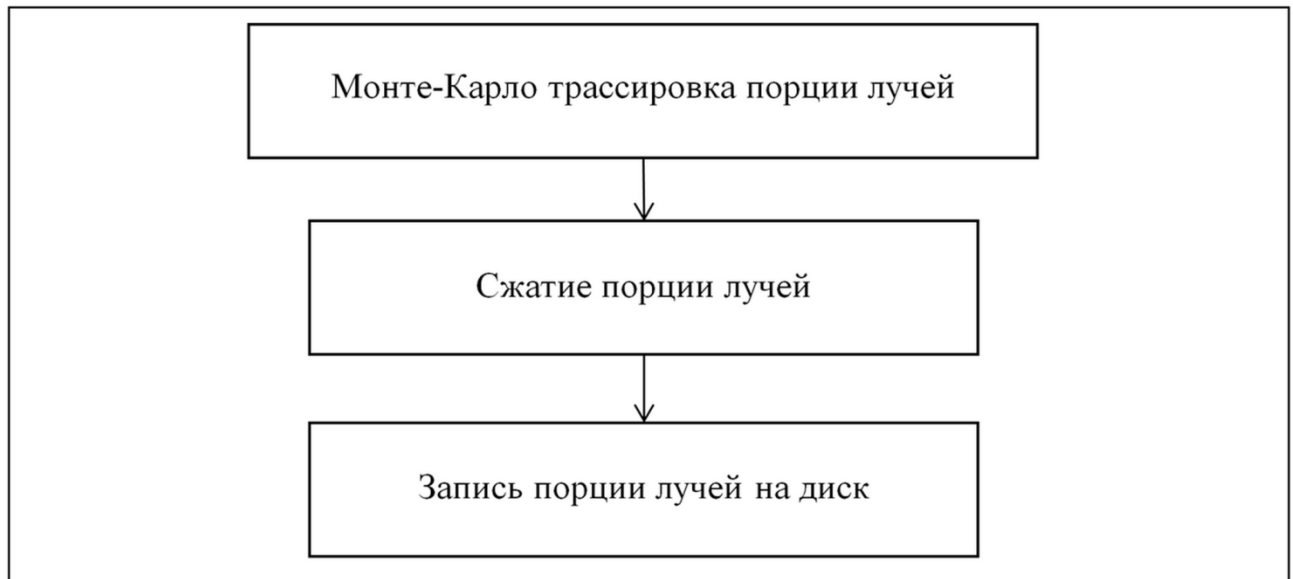


Рис. 4. Алгоритм сохранения лучей при Монте-Карло трассировке.

Таким образом, нам необходимо минимизировать объем записываемой информации, но при этом он должен быть достаточен для последующего анализа. Для каждого сегмента луча записывается следующая информация:

1. индекс источника света (short);
2. индекс объекта (short);
3. индекс треугольника (int), он используется для определения среды распространения света и свойств поверхности с обеих сторон поверхности;
4. координаты начальной точки для первого (от источника света) сегмента или конечной точки предыдущего сегмента для остальных сегментов (float[3]);
5. нормаль к поверхности в начальной точке сегмента (short[3]);
6. тип события в конечной точке сегмента (int);
7. индекс первого дескриптора взаимодействия сегмента с виртуальным

измерительным прибором (int), имеет значение -1 если таких событий не было;

8. количество дескрипторов относящихся к данному сегменту (WORD).

Всего используется 36 байт для записи одного сегмента луча. Для каждого луча записывается на один сегмент больше, чем их фактически существует. Последний сегмент используется для записи координат конечной точки предыдущего сегмента, а также типа события и направления в случае, когда луч покидает сцену. Для описания типа события в конечной точке сегмента используется int, поскольку в нем может быть записано несколько событий, и для каждого события используется отдельный бит.

Так как моделирование может происходить как в RGB, так и в спектральном пространстве, то длина массива, используемого для сохранения цвета сегмента луча, зависит от цветового простран-

ства, в котором происходит моделирование. Поэтому сохранение цвета сегмента луча происходит в специальном массиве элементов типа short. Для цвета луча в нашей Монте-Карло трассировке всегда используются нормированные значения, то есть сумма компонент цвета равна единице, поэтому точности, обеспечиваемой типом short ($1.0 / 65535 = \sim 1.5e-5$), вполне достаточно для нашей задачи визуализации лучей. Индекс первого элемента цвета в массиве цветов для данного сегмента определяется естественным способом как произведение индекса сегмента на количество величин, определяющих цвет. Это три для моделирования в пространстве RGB, и количество длин волн при спектральном моделировании.

Аналогичным образом для описания взаимодействия луча с виртуальными измерительными приборами используется массив соответствующих дескрипторов. Этот дескриптор включает в себя

1. индекс виртуального измерительного прибора, с которым было взаимодействие на данном сегменте луча (short);
2. индекс ячейки виртуального измерительного прибора, где было зарегистрировано взаимодействие (WORD[2]);
3. косинус угла между лучом и направлением виртуального измерительного прибора (short);
4. координаты точки пересечения луча с виртуальным измерительным прибором (или проекция на него точки пересечения с поверхностью) в относительных координатах виртуального детектора (WORD[2]).

Непосредственно сжатие в нашем алгоритме реализовано с помощью библиотеки сжатия общего назначения zlib [12]. Каждый из массивов – массив сегментов лучей, массив событий, регистрируемых на виртуальных измерительных приборах, и массив значений цветовых компонент, сжимаются отдельно.

При сохранении в файл для каждой порции записываются:

1. флаг сжатия (для целей отладки сохранена возможность записи не сжатой порции);
2. сжатый массив сегментов и его длина;
3. сжатый массив событий, регистрируемых на виртуальных измерительных приборах, и его длина (если существует);
4. количество каналов цвета, использованных при моделировании – 3 для RGB моделирования и количество длин волн при спектральном моделировании; возможно продолжить вычисления и сохранить лучи в тот же файл для новых порций лучей, сменив цветовое пространство;
5. сжатый массив значений цветовых компонент и его длина.

Этой информации достаточно для восстановления всех лучей, полученных с помощью Монте-Карло трассировки лучей, и для их анализа. Анализ позволяет выбор по заданному критерию, визуализацию выбранных лучей в окне вместе с геометрией сцены, используя естественные или искусственные цвета, вывод всех параметров данного сегмента в диалоговом окне по соответствующему запросу пользователя. Реализация описанной процедуры позволило обеспечить трассировку и сохранение лучей со скоростью ~ 396720 лучей в секунду вместо ~ 1266 лучей в секунду для старого алгоритма (Intel Core i7-4770 3.4GHz 32GB, 4 ядра, 8 потоков). То есть, при сохранении лучей было получено ускорение ~ 300 раз.

Анализ сохраненных лучей

Получаемый в результате моделирования файл по существу является трехмерной картой лучей. Эти трехмерные карты лучей используются для быстрого анализа световых характеристик на приемнике излучения с варьируемыми параметрами, для изучения особенностей распространения света в сцене и т.д. Одним из наиболее важных примеров использования этих карт является визуализация распространения свето-

вых лучей при проектировании сложных оптических систем [13]. Для изучения деталей распространения света и получения различных статистических характеристик в некоторых практических случаях используются трехмерные карты очень большого размера. Они могут содержать десятки миллионов лучей и сотни миллионов сегментов вместе с большим объемом информации об оптических событиях, имевших место вдоль траектории луча. Размеры файлов для хранения этих карт могут достигать нескольких гигабайт.

При визуализации распространения световых лучей типичной задачей, критически важной с точки зрения времени обработки, является отбор для последующей визуализации траекторий лучей, удовлетворяющих заданному критерию. Критерии при этом отборе используются самые разнообразные и достаточно сложные. Можно отметить следующие типичные события, происходящие вдоль траектории луча, которые могут представлять интерес при исследовании оптической системы [7]:

- луч был выпущен заданным источником света;
- луч пересекся или не пересекся с заданной гранью (треугольником) объекта;
- луч пересекся с заданной частью геометрического объекта оптической системы и претерпел заданное оптическое преобразование;
- было зарегистрировано взаимодействие луча с заданным виртуальным измерительным прибором;
- луч пересекся с поверхностью, обладающей заданными оптическими свойствами, и претерпел заданное оптическое преобразование;
- луч претерпел заданное оптическое преобразование (зеркальное или диффузное отражение, поглощение и т.д.) на одном из объектов оптической системы;
- луч был зарегистрирован в заданной области виртуального измерительного прибора.

В общем случае из этих элементарных событий строится логическое выражение, представленное в виде дерева событий, изображенного на рис. 6, которое и является критерием отбора лучей для визуализации. Выражение строится с помощью операций логического пересечения (\cap), объединения (\cup) и логического отрицания (\neg). Для каждого заданного оптического преобразования можно выбрать, какие именно преобразования имели место и сколько раз.

Для комфортной работы пользователя необходимо минимизировать время реакции системы при любых изменениях критерия отбора лучей для визуализации. Желательно обеспечить реакцию системы в реальном времени, когда это возможно. Для трехмерных карт лучей очень большого размера, содержащихся в гигабайтных файлах, желательно, чтобы время реакции системы не превышало хотя бы нескольких минут. Поскольку практически все используемые в настоящее время компьютеры являются многоядерными, то представляется целесообразным распараллелить процесс анализа трехмерных карт лучей.

Подсистема анализа лучей, полученных с помощью Монте-Карло трассировки, включает в себя следующие компоненты:

1. пользовательский интерфейс, обеспечивающий построение критерия выбора лучей и контроль параметров визуализации лучей;
2. чтение файла с трехмерной картой лучей и выбор из них лучей, удовлетворяющих построенному критерию;
3. визуализацию траекторий выбранных лучей и пользовательский интерфейс для представления информации о выбранных лучах – координат начала и конца сегментов, цвет сегмента (RGB и спектральный), имя среды распространения света для данного сегмента и ее основные параметры и т.д.

Пользовательский интерфейс

Пользовательский интерфейс подсистемы визуализации лучей приведен на рис. 5. В левой части приведен интерфейс, реализованный в системе CATIA, а в правой части – в системе Lumicert. Некоторая разница в интерфейсе обусловлена как использованием разных библиотек (RADE в CATIA и QT в Lumicert), так и использованием разных объектов. В системе CATIA можно ис-

пользовать грани объектов, в том числе и криволинейные, в критерии отбора лучей, в то время как в системе Lumicert такое понятие отсутствует. В системе CATIA имеется также возможность использования в качестве критерия пересечения лучом области детектора виртуального измерительного прибора, которая в системе Lumicert в настоящее время не реализована.

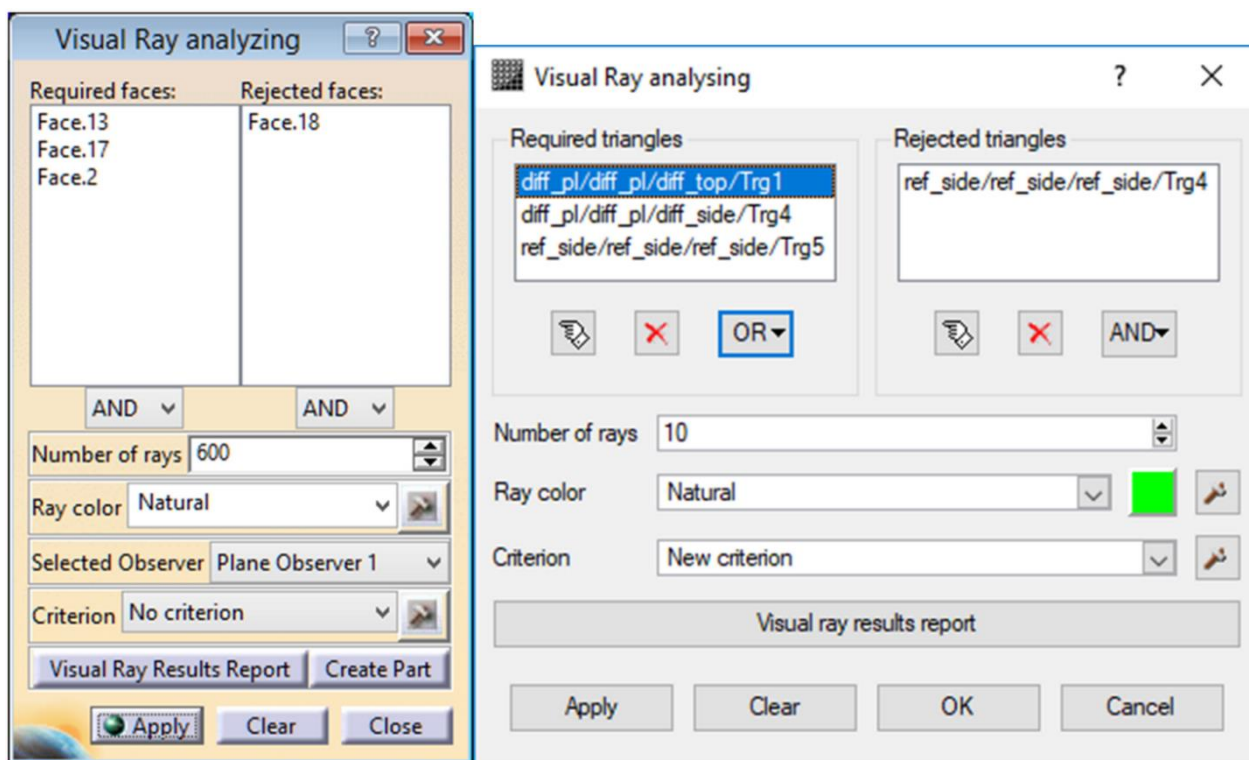


Рис. 5. Пользовательский интерфейс подсистемы визуализации лучей.

Дополнительный, более сложный критерий можно построить с помощью редактора критерия визуализации, первоначально реализованного в базовой системе Lumicert. В системе CATIA реализация этого диалога обеспечивается через I2 Server по схеме, приведенной на рис. 3 с помощью команды VR_RAY_CRITERIA. Пользовательский интерфейс сложного критерия отбора лучей приведен на рис. 6. Окончательно критерий объединяет все события, указанные на рисунках 5 и 6.

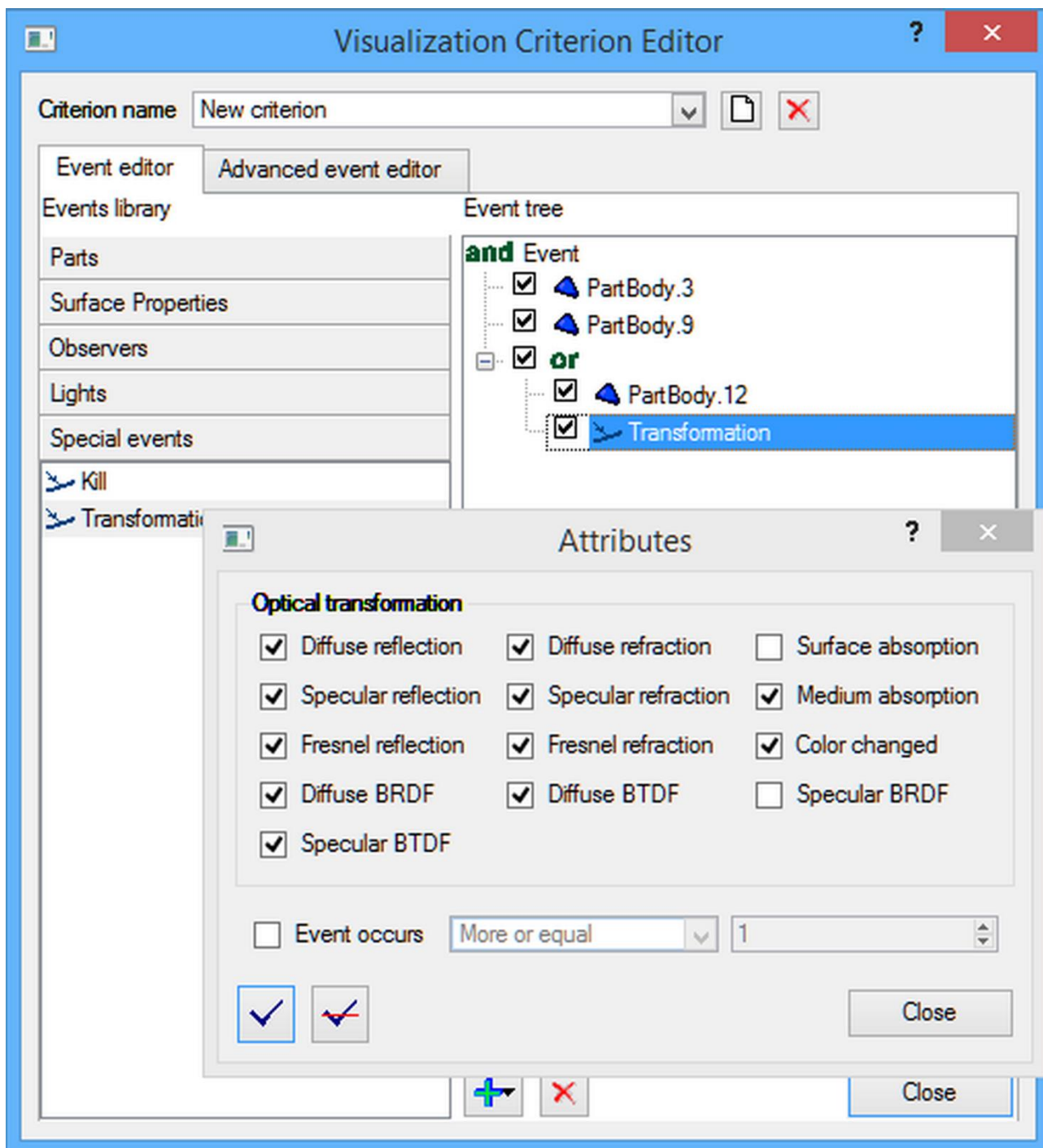


Рис. 6. Сложного критерия отбора лучей.

Если при моделировании были заданы виртуальные измерительные приборы, то результаты, полученные при регистрации лучей, могут быть использованы для задания дополнительного критерия. Для этой цели используется модуль визуального анализа LumiVue, который позволяет выделять и редактировать область на изображении виртуального прибора в виде прямоугольника или эллипса. Регистрация лучей на этой области может рассматриваться в качестве дополнительного критерия при визуализации лучей. Пример такой выделенной области приведен на рис. 7. Дополнительные возможности использования выделенной области для анализа изображения приведены в работе [13].

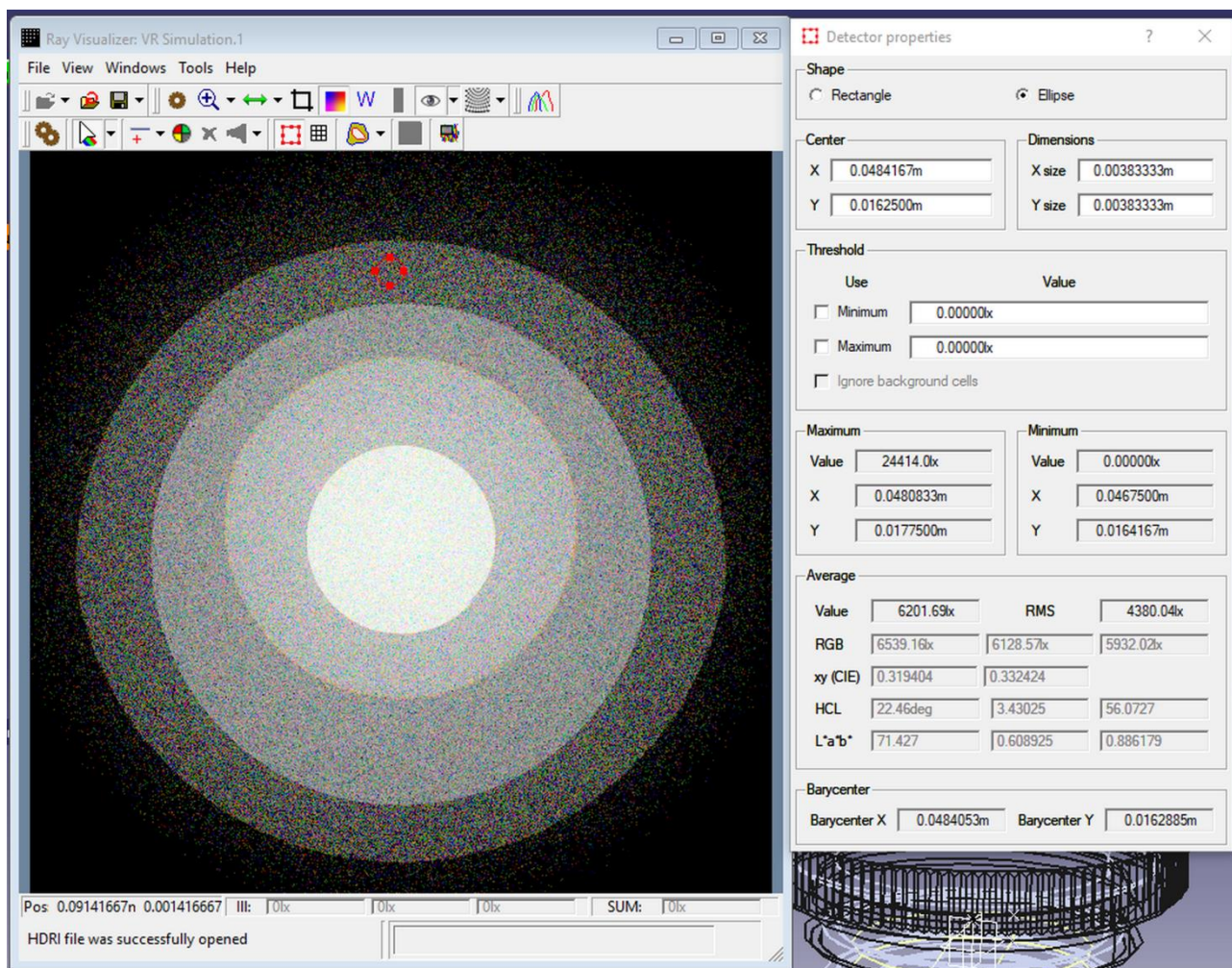


Рис. 7. Использование выделенной области в качестве критерия.

Загрузка изображения с результатом регистрации лучей на данном виртуальном измерительном приборе и показ окна LumiVue происходит по сообщению LUMIVUE_LOAD, которое посылает CATIA (рис. 3.) Путь на файл с изображением при этом передается через общую память, используемую процессами (shared memory). Результат моделирования в виде сохраненных лучей может использовать несколько виртуальных измерительных приборов. Можно выбрать любой из них или вообще отказаться от использования виртуальных измерительных приборов при анализе. В случае замены виртуального измерительного прибора сообщение LUMIVUE_LOAD посылается еще раз, но через общую память теперь передается путь на новый файл с результатом регистрации на новом измерительном приборе. В случае отказа от использования измерительного прибора CATIA по-

сылает сообщение LUMIVUE_HIDE и LumiVue скрывает свое окно. Для получения параметров выделенной области CATIA посылает сообщение LUMIVUE_DETECTOR, и LumiVue заносит все параметры выделенной области в общую память. Все остальные параметры, необходимые для построения критерия, и путь на файл с лучами уже находятся в общей памяти. CATIA теперь посылает сообщение VR_RAY_HISTORY. По этому сообщению I2 Server читает файл с лучами и начинает выбирать из них лучи, удовлетворяющие построенному критерию.

Чтение файла и выбор лучей

Чтение файла с трехмерной картой лучей и выбор из нее лучей, удовлетворяющих построенному критерию, является наиболее критичной процедурой для скорости визуализации световых

лучей. Крайне важно обеспечить для этой процедуры приемлемое для пользователя время реакции системы. Желательно, чтобы оно не превышало нескольких секунд. В общем виде алгоритм выбора лучей, удовлетворяющих построенному критерию, представлен на рис. 8.

На многоядерных компьютерах естественным подходом для ускорения обработки является параллельное использование нескольких потоков. Этот подход эффективно нами используется для прямой стохастической трассировки лучей [14]. Количество потоков выбирается, как правило, равным количеству виртуальных ядер компьютера.

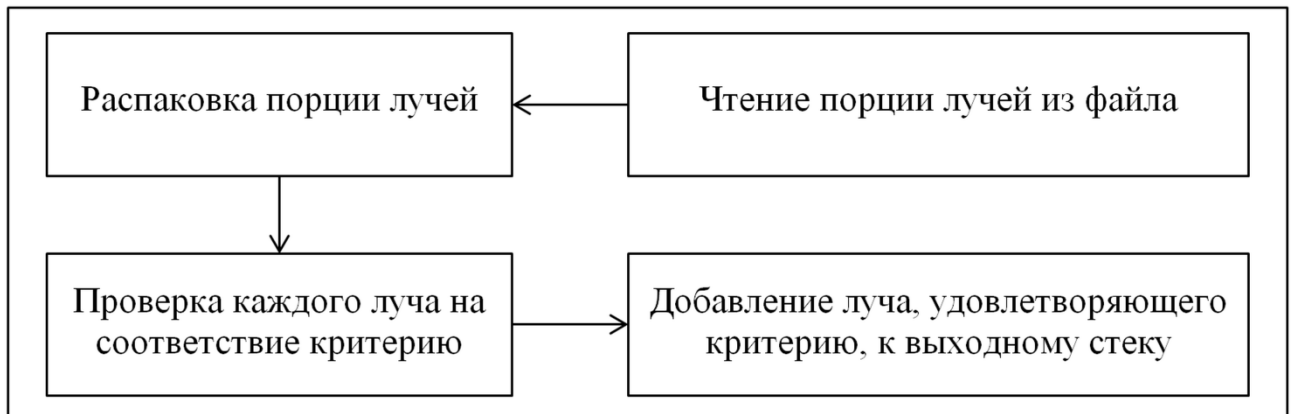


Рис. 8. Общий алгоритм выбора лучей, удовлетворяющих критерию.

Ясно, что в процедуре, приведенной на рис. 8, только распаковка порции и проверка луча на соответствие критерию могут выполняться параллельно для разных порций лучей. Чтение порций лучей из файла должно выполняться последовательно, поскольку все потоки работают с одним файлом. Добавление найденного луча, удовлетворяющего критерию, в выходной стек также не может выполняться параллельно, поскольку стек является общим для всех потоков. Таким образом, мы приходим к многопоточной схеме выбора лучей, удовлетворяющих заданному критерию, приведенной на рис. 9.

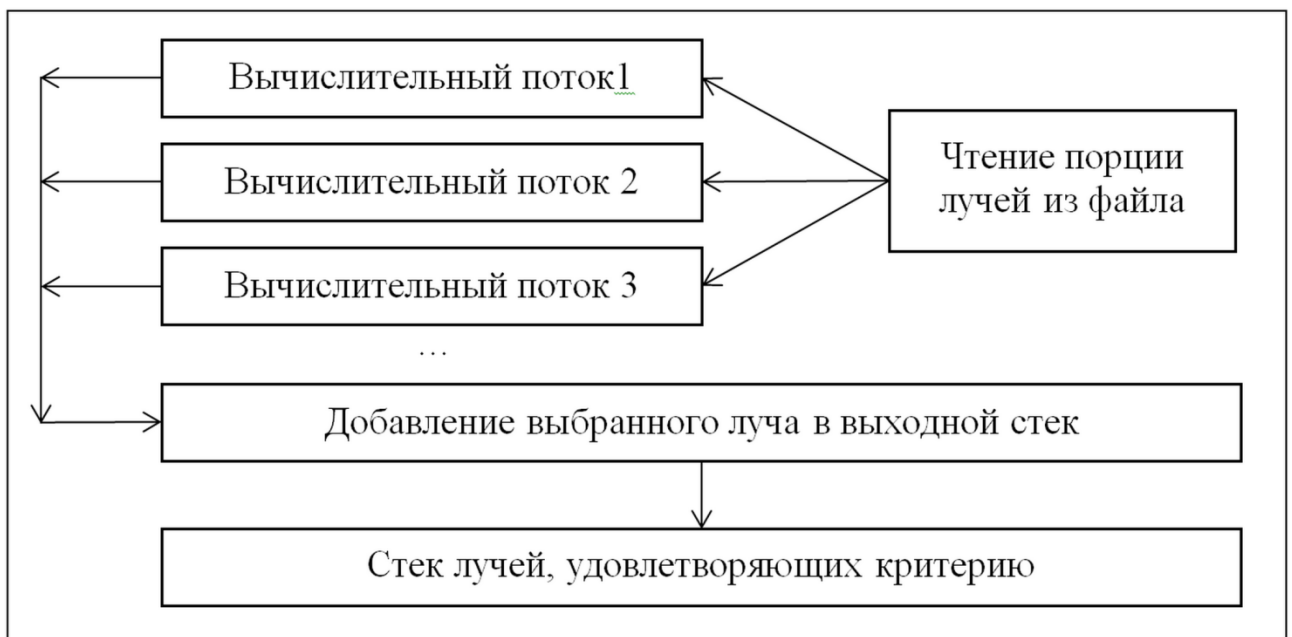


Рис. 9. Многопоточный алгоритм выбора лучей, удовлетворяющих критерию.

Процедура анализа трехмерной карты лучей стартует после того, как пользователь указал необходимые параметры критерия выбора лучей. Создаются все необходимые объекты для хранения результатов обработки данных, в частности, выходной стек лучей, открывается файл, в котором хранится карта лучей. А также создаются и запускаются вычислительные потоки для выбора лучей. Далее вся обработка данных осуществляется в вычислительных потоках, а главный поток только отражает прогресс этой обработки, указывая процент обработанных лучей.

Обработка лучей в каждом потоке осуществляется в соответствии со схемами алгоритма, приведенными на рисунках 8 и 9:

1. Считывается порция лучей из файла. Поскольку все потоки работают с одним файлом, то здесь для синхронизации потоков используется критическая секция. Поскольку чтение файла в современных компьютерах использует специальный сопроцессор ввода / вывода и кэш, то эта процедура практически не ограничивает работу потоков, которые распаковывают порции и проверяют критерии.
2. Полученная порция лучей распаковывается и осуществляется поиск лучей, удовлетворяющих заданному критерию.
3. Найденный луч, удовлетворяющий заданному критерию, добавляется в выходной стек в формате, пригодном для немедленной визуализации. Т.к. выходной стек является общим для всех вычислительных потоков, то это добавление также осуществляется с использованием критической сек-

ции. Поскольку эта процедура сводится, в основном, к добавлению в стек указателя на объект, содержащий массив сегментов, и достаточно редко требуется большое число лучей, то время ее выполнения достаточно мало и не задерживает работу основных вычислительных потоков.

Эффективность построенной процедуры можно охарактеризовать следующими цифрами. Для файла, содержащего ~50 миллионов лучей (~450 миллионов сегментов), время полной обработки трехмерной карты лучей на компьютере Intel Core(TM) i7-4770 с четырьмя физическими ядрами и восемью потоками (с использованием Hyper Threading) составляет ~40 секунд. Файл в данном случае обрабатывался полностью, поскольку заказанное число лучей, удовлетворяющих заданному критерию, так и не было достигнуто. В файле содержались спектральные данные, поэтому размер файла был ~6 Gb. В большинстве практических случаев достаточно обработать всего несколько миллионов лучей. В этом случае обработка и визуализация выбранных лучей будет осуществляться в реальном времени.

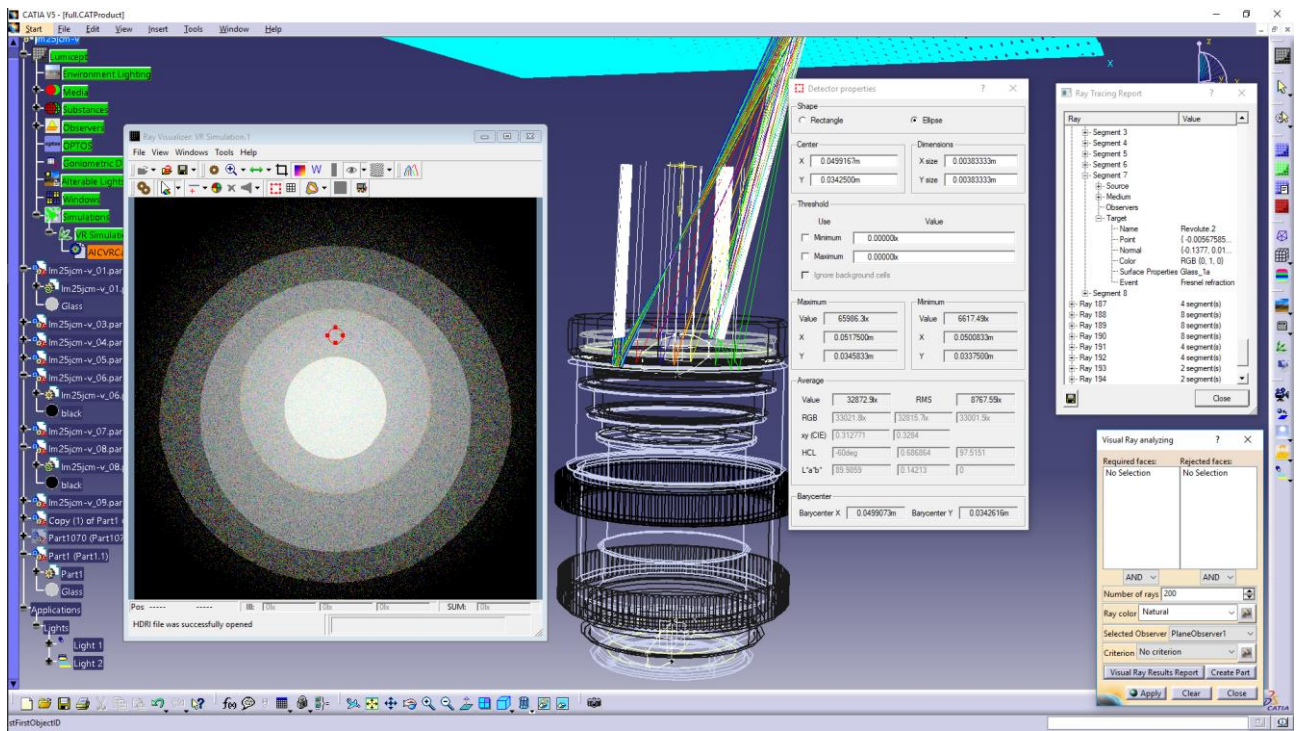
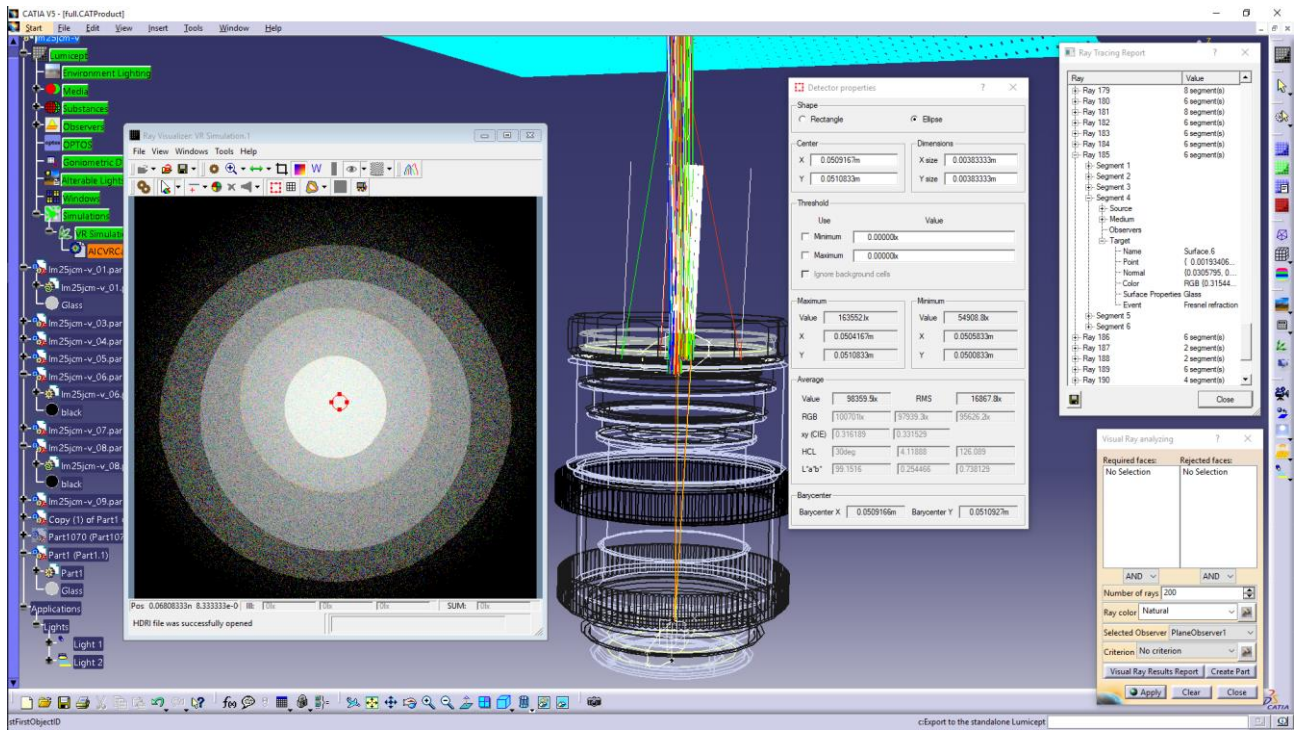
На рис. 10 приведена зависимость скорости обработки трехмерной карты лучей от количества используемых потоков для указанного выше компьютера. Видно, что предложенный алгоритм хорошо масштабируется при увеличении числа используемых процессоров. Даже использование виртуальных процессоров (добавление 5, 6, 7 и 8 потоков) дает заметное увеличение скорости обработки.



Рис. 10. Зависимость скорости обработки трехмерной карты лучей от количества используемых потоков.

Примеры визуализации выбранных лучей

Примеры визуализации хода лучей в объективе для разных областей на изображении виртуального прибора приведены на рис. 11. На объектив падает параллельный пучок белого света, направление которого отклонено от оси объектива на угол ~ 2 градуса. Материал линз обладает дисперсией – величина показателя преломления (индекса рефракции) зависит от длины волны света. Поэтому после преломления луча на границе двух сред луч дальше трассируется со значением единственной длиной волны, выбранной из набора длин волн, заданным пользователем. Длина волны (и соответственно направление преломленного луча) выбирается из заданного набора длин волн с вероятностью, пропорциональной части энергии луча, соответствующей выбираемой длине волны. По этой причине после преломления лучи на рис. 11 становятся цветными, если выбран естественный цвет для визуализации. При нажатии кнопки Visual Ray Results Report открывается диалоговое окно, приведенное в правой верхней части рисунков 11, в котором пользователь может детально анализировать все события, имевшие место вдоль траектории луча.



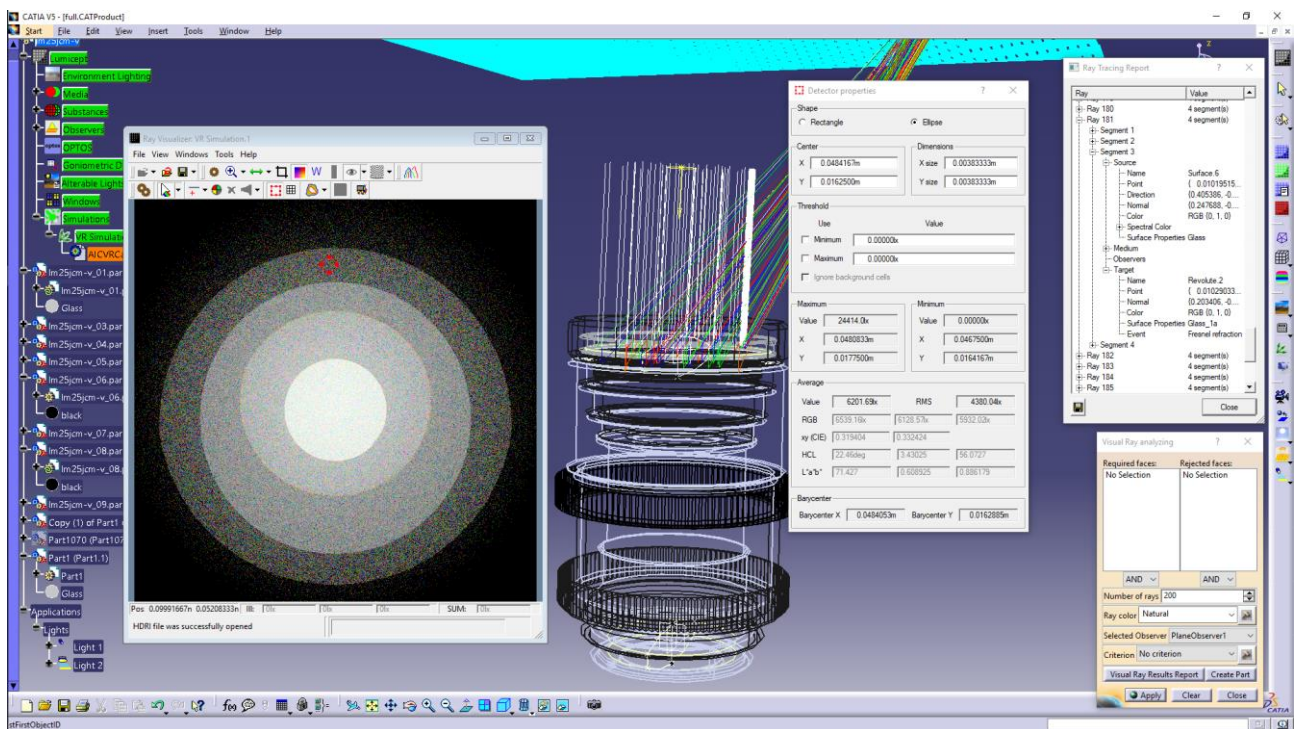
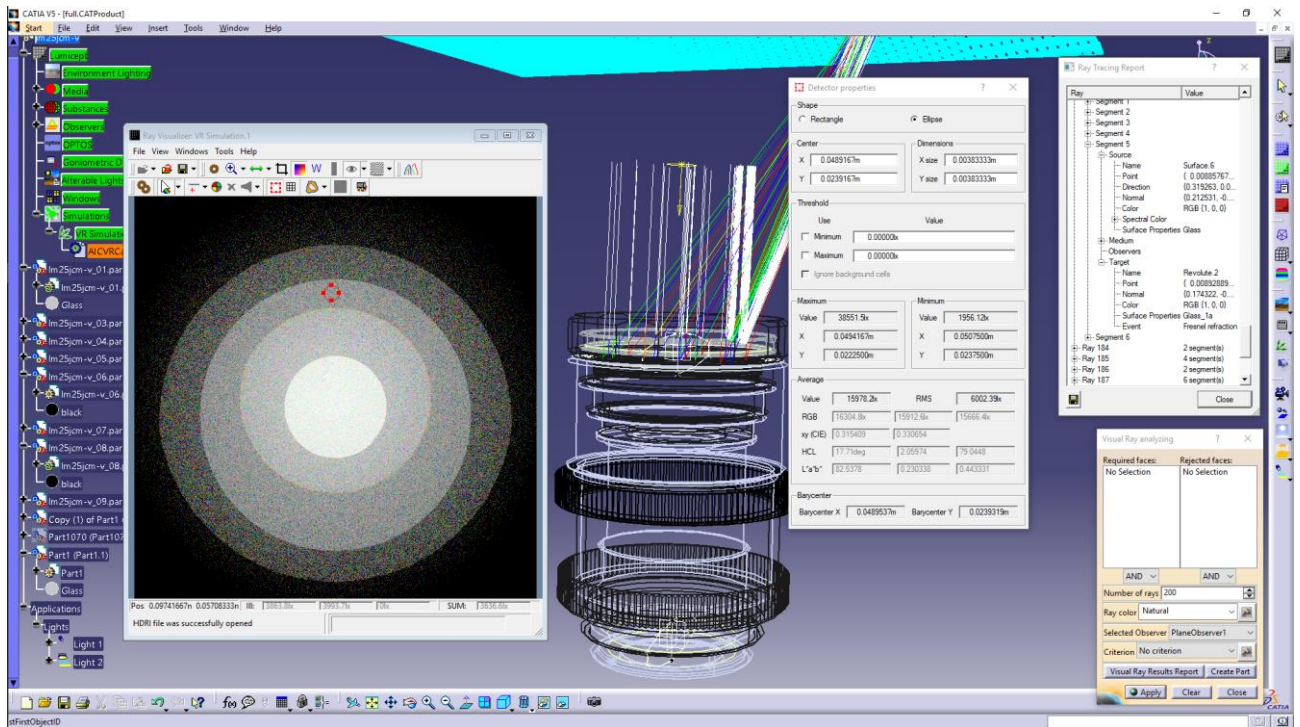


Рис. 11. Примеры визуализации выбранных лучей и отчета о событиях вдоль луча.

Заключение

Визуальное представление траекторий световых лучей стало фактически базовой функциональностью современных систем оптического моделирования. Разработанные алгоритмы позволяют эффективно использовать многоядерные компьютеры как для расчета трехмерной карты лучей, получаемой при Монте-Карло трассировке, так и

при их визуализации с использованием различных критериев для отбора лучей. Разработанные алгоритмы реализованы как в базовой системе синтеза реалистичных изображений и оптического моделирования Lumicert, разработанной в ИПМ им. М.В.Келдыша РАН, так и в соответствующей системе визуализации лучей, интегрированной в САПР CATIA.

Список литературы

- [1] Pharr M., Humphreys G., Physically Based Rendering: From Theory to Implementation. Second Edition. Morgan Kaufmann Publishers Inc., 2010.
- [2] Bogdanov N., Zhdanov D., Potemin I., Zhdanov A. Design of Ergonomic Illumination Systems for Cultural, Medical, Educational Facilities. The Educational Review, USA, 1(4), 85-90. <http://dx.doi.org/10.26855/er.2017.04.001>.
- [3] Wernert E., A unified environment for presenting, developing and analyzing graphics algorithms. Computer Graphics, т. 31, № 3, pp. 26-28, 1997.
- [4] Tsirikoglou A., Kronander J., Wrenninge M., Unger J., Procedural Modeling and Physically Based Rendering for Synthetic Data Generation in Automotive Applications. arxiv.org. 2017.
- [5] McCormac J., Handa A., Leutenegger S., Davison A., SceneNet RGB-D: 5M Photorealistic Images of Synthetic Indoor Trajectories with Ground Truth. The IEEE International Conference on Computer Vision (ICCV 2017).
- [6] Rozantsev A., Lepetit V., Fua P., On rendering synthetic images for training an object detector. Comput. Vis. Image Underst. 137, C (August 2015), 24-37. DOI: <http://dx.doi.org/10.1016/j.cviu.2014.12.006>.
- [7] Волобой А.Г., Вишняков С.М., Галактионов В.А., Жданов Д.Д., Средства визуализации распространения света в задачах проектирования и анализа оптических систем // Препринты ИПМ им. М.В. Келдыша РАН, № 54, 2007, 20 с.
- [8] Kopylov E., Dmitriev K., Light propagation visualization as a tool for 3D scene analysis in lighting design. Computers & Graphics, т. 24, № 1, pp. 31-39, 2000.
- [9] Волобой А.Г., Галактионов В.А., Жданов А.Д., Жданов Д.Д., Средства визуализации распространения световых лучей в задачах проектирования оптических систем. "Информационные технологии и вычислительные системы", № 4, с. 28-39, 2009.
- [10] Barladyan B., Shapiro L., Voloboy A., Ray maps technique for effective interrogation of results of MCRT simulation // Conference proceedings of 21-th International Conference on Computer Graphics and Vision GraphiCon-2011, Moscow State University, September 26-30, 2011, Moscow, Russia, pp. 46-49.
- [11] Б.Х. Барладян, Е.Д. Бирюков, А.Г. Волобой, Л.З. Шапиро, «Эффективный алгоритм визуализации предварительно рассчитанных лучей» // Труды 28-й Международной конференции по компьютерной графике, обработке изображений и машинному зрению, системам визуализации и виртуального окружения GraphiCon-2018, Томск, 24-27 сентября 2018 г., с. 36-39.
- [12] "zlib" библиотека сжатия общего назначения, <http://zlib.net.ru/>.
- [13] Barladian B.K., Potemin I.S., Zhdanov D.D., Voloboy A.G., Shapiro I.V., Valiev L.S., Birukov E.D., Visual analysis of the computer simulation for both imaging and non-imaging optical systems // Proc. SPIE 10021, Optical Design and Testing VII, 100210T (October 31, 2016); doi:10.1117/12.2247751.
- [14] Б.Х. Барладян, Л.З. Шапиро, Е.Ю. Денисов, А.Г. Волобой, Эффективный многопоточный алгоритм расчета глобальной освещенности // Программирование, 2017, № 4, с.3-12.