

РАЗРАБОТКА АЛГОРИТМОВ НЕПРЕДВЗЯТОГО ЗДРЕНДЕРИНГА

Федоров А.Р.¹, Федоров П.А.¹

¹Национальный исследовательский университет, Зеленоград, Москва, Россия (124498, Москва, Зеленоград, проезд 4806, д.5, МИЭТ), e-mail: kyawzawye85@gmail.com

Современные технологии моделирования внедрены непредвзятого ЗДрендеринга. Данная статья направлена на разработку алгоритмов рендеринга, обеспечивающие непосредственное получение результирующего изображения 3D сцены. Алгоритмы представлены в разделе по принципу – “от простого к сложному”, или “итеративно”, что упрощает понимание, и самое главное – их дальнейшую реализацию выбранной автором методике “итеративной разработки программного обеспечения”. Предложены методики повышения производительности рендеринга (пока без ускоряющей структуры): генерация отраженных/преломленных лучей с использованием выборки по значимости BRDF (BRDFimportancesampling); поддержка прямого/косвенного освещения, а также качества результирующего изображения: методика lightsamling для “мягких” теней.

Ключевые слова: непредвзятый ЗДрендеринг, моделирование, свет, алгоритм, изображение, трассировки путей.

DEVELOPMENT OF ALGORITHMS FOR IMPARTIALITY 3D RENDERING

Fedorov A.R.¹, Fedorov P.A.¹

¹National Research University, Zelenograd, Moscow, Russia (124498, Moscow, Zelenograd, passage 4806, House.5, MIET), e-mail: kyawzawye85@gmail.com

Modern simulation technology introduced impartial 3D rendering. This article is aimed to the development of rendering algorithms that provide direct feedback to the resulting image 3D scene. Algorithms are presented in the section on the principle - "from simple to complex," or "iterative", which simplifies understanding and most importantly - their further implementation selected by author technique “iterative software development”. Proposed methods for improvement of rendering performance (yet without accelerating structure): generation of reflected / refracted rays using importance sampling BRDF (BRDFimportancesampling); support for direct / indirect lighting, as well as the quality of the resulting images: technique lightsamlingdlya "soft" shadows.

Keywords: unbiased 3D rendering, modeling light, the algorithm, the image path tracing.

Компьютерная графика всегда решала и продолжает решать проблемы, связанные не только с индустрией развлечений, но и создавать детские обучающие программы, повышать эффективность работы людей творческих профессий (архитекторов и дизайнеров), и что самое главное – помогать людям, не имеющим по физическим причинам полноценной вовлеченности в окружающий их мир. Именно поэтому, исследования в данной области всегда будут актуальными.

Любое графическое представление объектов 3D сцены в связи с ограниченностью вычислительных ресурсов является приближенным решением специального интегрального уравнения – “уравнения рендеринга”. Среди всего множества алгоритмов, выделяют так называемые “непредвзятые”, копирующие физическую модель распространения света из реального мира, и, таким образом, снижающие погрешность в результирующем изображении – позволяя добиться “фотореалистичности”.

Главной проблемой “непредвзятости” является необходимость в действительно больших объемах вычислений, а значит и невозможность обработки алгоритмов

непредвзятого 3Dрендеринга в режиме реального времени. Такой режим просто необходим в случае интерактивных приложений, основанных на непрерывном, “интерактивном” взаимодействии с пользователем (военные симуляторы, программы обучения вождению и многие др.).

Таким образом, объектом исследования данной работы выступают алгоритмы непредвзятого 3Dрендеринга как единственные, позволяющие получить максимально реалистичное или “фотореалистичное” изображение в компьютерной графике.

Простейший алгоритм по методу “бросания лучей (raycasting)”

Как было отмечено в ходе анализа современного состояния области непредвзятого 3Dрендеринга – все алгоритмы рендеринга являются всего лишь некоторым приближением, или аппроксимацией, главного уравнения рендеринга[1]:

$$Lout(p, w) = Lemitted(p, w) + \int_{\Omega} BRDF(p, w, w') Lin(p, w') \cos(n, w') dw', \quad (1)$$

где: $Lemitted(p, w)$ – количество энергии света, излучаемого из точки p в направлении вектора w ; $BRDF(p, w, w')$ – двунаправленная функция отражательной способности поверхности, которая показывает количество энергии, передаваемой в ходе отражения света в точке p из w' в направлении w (зависит от свойств материала поверхности); $Lin(p, w')$ – количество энергии света, пришедшего в точку p с направления w' ; $\cos(n, w')$ – косинус угла между нормалью поверхности в точке p и “входящим” направлением w' .

Другими словами, количество энергии света, попадающего в глаз наблюдателя из любой точки сцены, складывается из двух факторов: энергии излучаемой из этой точки, а также – отраженной. Последняя зависит от BRDF материала поверхности, на которой лежит точка, а также направления поступающего в нее света (вклад вносят все лучи, попадающие в полусферу с центром в этой точке).

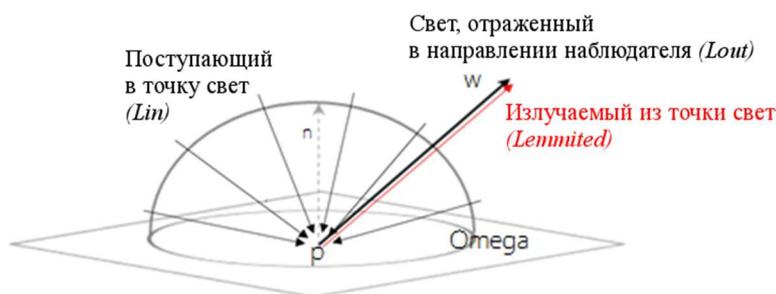


Рисунок 1. Визуализация главного уравнения рендеринга

Таким образом, решение главного уравнения рендеринга согласно методу “бросания лучей (raycasting)” при допущении, что все объекты сцены излучают свет, может быть записано, как:

$$Lout(p, w) = Lemitted(p, w), \quad (2)$$

Видно, что в расчете (2) участвует лишь левое слагаемое суммы (1), что делает метод бросания лучей действительно простым, но сильно “предвзятым”. Вкратце, данный метод заключается в эмиссии лучей из глаза наблюдателя в направлении сцены таким образом, что каждый луч проходит через один пиксель экрана наблюдателя. Пересекаясь с объектами сцены, лучи добавляют их цвета в соответствующие пиксели (согласно допущению, сделанному ранее об эмиссии света всеми объектами сцены). Этот процесс показан на рис. 2.

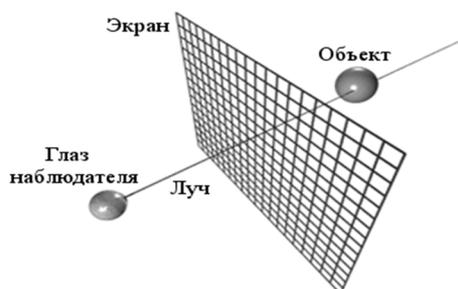


Рисунок 2. Метод “бросания лучей (raycasting)”

Тогда, алгоритм трассировки лучей по методу “бросания лучей” может быть представлен на рис. 3. Из схемы ясно, что операции над конкретными пикселями экрана наблюдателя, вообще говоря – одинаковы, но что более важно – независимы. Принимая во внимание одно из главных условий задачи диссертации, заключенное в использовании мультипараллельной архитектуры (GPU), как платформы для реализации и верификации разработанного алгоритма: представленный алгоритм легко “параллелится” на уровне инструкций отдельными пикселями (или на “пиксельном уровне”). А значит, каждый поток такой системы в конечном итоге трассирует один из лучей, эмитированных из глаза наблюдателя. Стоит также отметить, что подобный уровень параллелизма инструкций применим к более сложным алгоритмам трассировки лучей (и путей), рассматриваемым в последующих разделах.

Кроме того, на рис. 3 видно одно из самых “проблемных” мест алгоритма с точки зрения его производительности – это поиск пересечения луча и сцены. Подобная схема – без использования ускоряющей структуры – требует проверки пересечения луча с каждым из полигонов сцены, поэтому общая вычислительная сложность алгоритма складывается из сложности теста на пересечение, умноженного на количество полигонов сцены (очевидно, в современных реалиях, где только объекты сцены содержат в себе уже миллионы полигонов, такой подход крайне невыгоден).

Алгоритм трассировки путей по методу Монте Карло

Используя метод Монте Карло для нахождения приближенного значения интеграла из (1), можно получить более непредвзятое и “реалистичное” решение главного уравнения рендеринга (2):

$$Lout(p, w) \approx Lemitted(p, w) + \frac{1}{2\pi N} \sum_{i=0}^N BRDF(p, w, w_i) Lin_i(p, w_i) \cos(n, w_i), \quad (3)$$

где направления лучей w_i под знаком суммы – равномерно распределены по полусфере с центром в точкер.

Такой подход позволяет достаточно близко аппроксимировать уравнение рендеринга (1) с одним лишь допущением: количество лучей попадающих в конкретную точку ограничено некоторым числом – N . Другими словами, основная идея метода состоит в ограничении бесконечного числа падающих на точку лучей, позволяя, таким образом, трассировать их за “разумное” время.

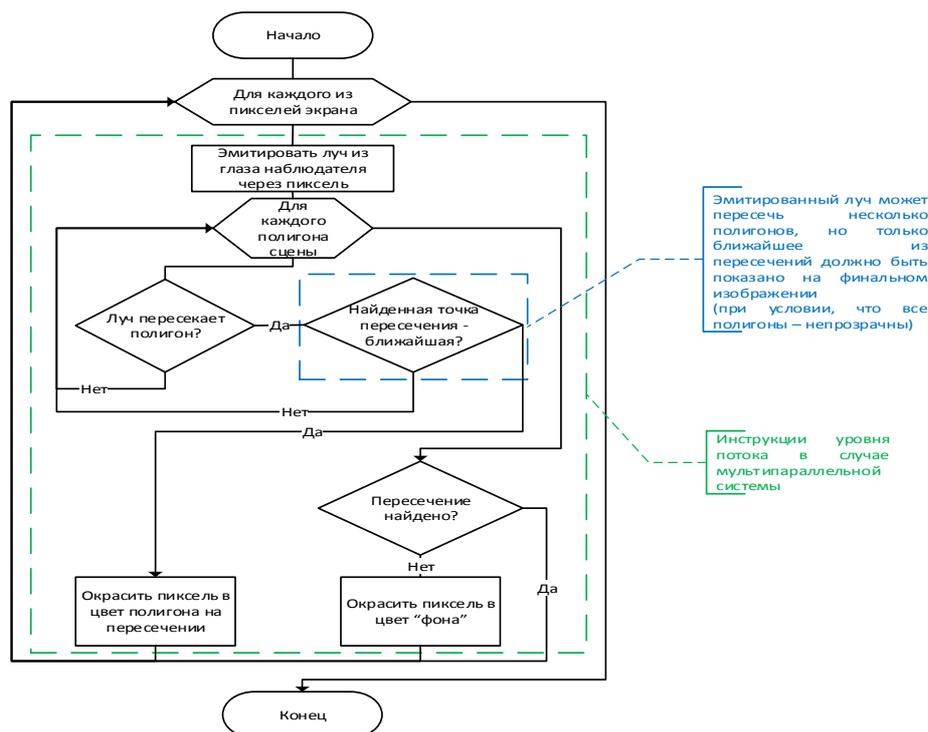


Рисунок 3. Схема простейшего алгоритма рендеринга по методу “бросания лучей”

Метод Монте Карло в решении уравнения рендеринга применяют в алгоритме под названием – “трассировка путей по методу Монте Карло”[2]. Сама по себе трассировка путей также вносит некоторые упрощения в процесс рендеринга. Лучи, подобно методу бросания лучей, испускают из глаза наблюдателя сквозь пиксели в экране. Проходя через них, лучи взаимодействуют с объектами сцены. На каждом их пересечении образуют по одному отраженному и преломленному лучу, до тех пор, пока луч не достигнет источника света. Другими словами, между глазом наблюдателя и источниками света строятся (и трассируются) пути, состоящие из лучей, образованных между пересечениями. Стоит заметить, что сам процесс протекает в “обратном” направлении, нежели в реальной физической модели распространения света. Однако, как было замечено ранее – такой подход не влияет на физику процесса, но позволяет сэкономить на ненужных вычислениях (трассировке) лучей, не попадающих в экран наблюдателя.

Таким образом, можно сказать, что алгоритм трассировки путей, использующий для решения уравнения рендеринга – уравнение (3) с $N=1$ и называют “алгоритмом трассировки путей по методу Монте Карло”. Его схема представлена на рис. 5.

Видно, что схема алгоритма построена таким образом, что поиск пересечения луча и сцены выделен в отдельную процедуру (сам алгоритм поиска пересечения тот же, что представлен на рис. 3 – в алгоритме по методу “бросания лучей”). Так, процесс разработки алгоритмов ускоряющей структуры затронет лишь эту процедуру, оставив все остальное – без изменений.

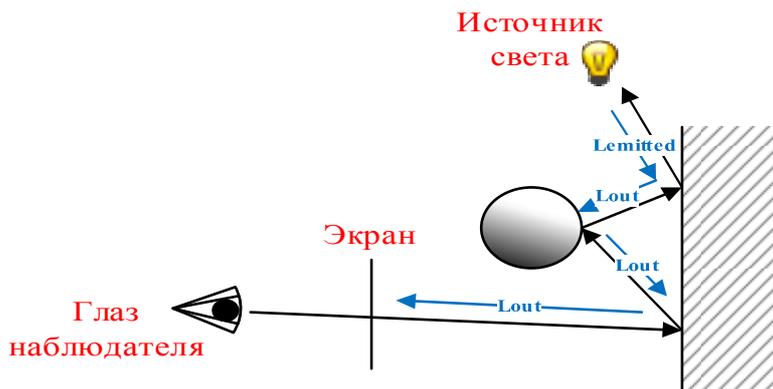


Рисунок 4. Алгоритм трассировки путей

Кроме этого, стоит заметить, что представленная схема показывает операции, выполняемые отдельным потоком (т.е. уже содержит предложенный ранее – “пиксельный уровень” параллелизма). Также, предложенный алгоритм включает в себя рекурсию – для генерации отраженного/преломленного лучей на каждом взаимодействии с объектами сцены. Поскольку генерация лучей завершается, как только путь встречает источник света – количество рекурсивных вызовов в общем случае – не ограничено. Поэтому, для завершения рекурсии используется распространенный в рендеринге метод “русской рулетки” [3]. Идея состоит в том, что исходное уравнение рендеринга (1), а в нашем случае – (3) изменяется таким образом, что в 50% случаев, в качестве решения берется только левое слагаемое (энергия эмиссии света из точки), а в оставшихся – правое (включающее рекурсию). Это вносит некоторую предвзятость в алгоритм, однако позволяет вычислять его за ограниченное время – при помощи “остановки” рекурсии с 50% вероятностью (однако, во время реализации алгоритма не стоит забывать о математической необходимости в этом случае – добавить $\frac{1}{2}$ множители к слагаемым исходного уравнения).

Улучшенный алгоритм трассировки путей по методу Монте Карло

Несмотря на все преимущества стандартного алгоритма трассировки путей по методу Монте Карло, описанного выше, он все еще достаточно сложен для использования в режиме реального времени. Предлагаемая в данном разделе методика позволяет улучшить стандартный алгоритм, сосредоточившись, во-первых – на повышении его

производительности (посредством самого алгоритма рендеринга, но не ускоряющей структуры), и во-вторых – на улучшении качества результирующего изображения.

Так, для достижения первой цели, необходимы: генерация отраженных/преломленных лучей с использованием метода выборки по значимости (BRDF importance sampling); поддержка прямого/косвенного освещения.

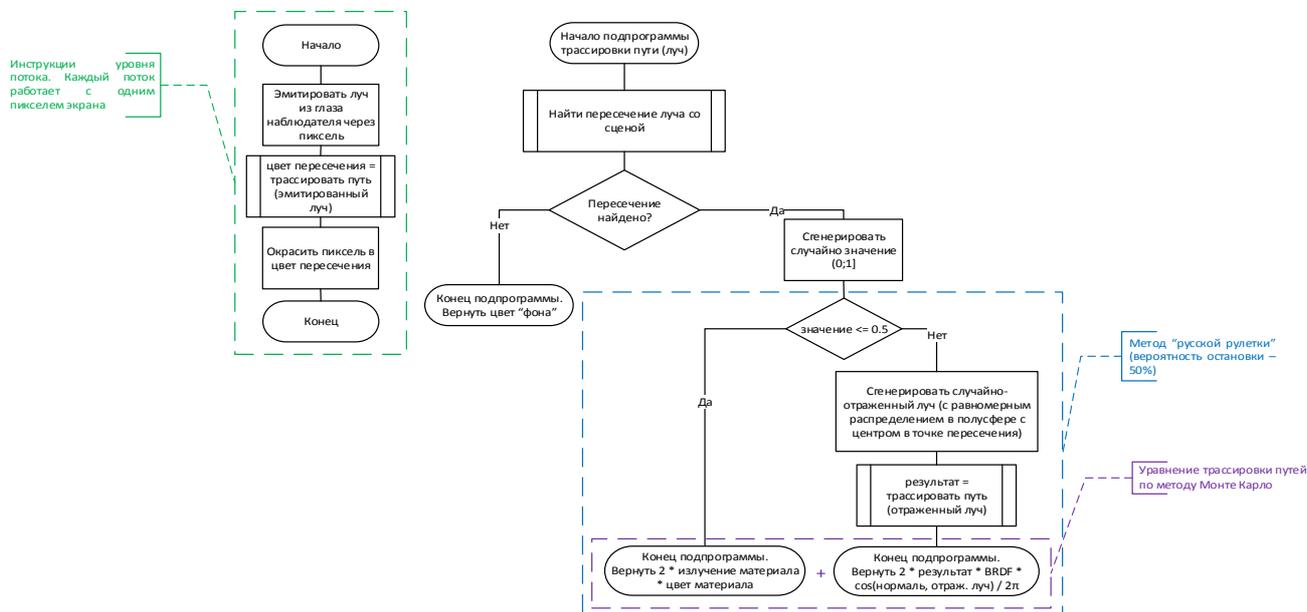


Рисунок 5. Схема алгоритма трассировки путей по методу Монте Карло

Метод выборки по значимости (“BRDF importance sampling”)

Распределение энергии света при отражении от поверхности, вообще говоря, зависит от свойств материала. Это значит, что только “важные” направления вносят вклад в количество энергии света, пришедшей в точку. Например, в случае зеркальной поверхности свет почти полностью отражается в зеркальном направлении, а значит генерировать и трассировать лучи в других направлениях – нет смысла [4,5].

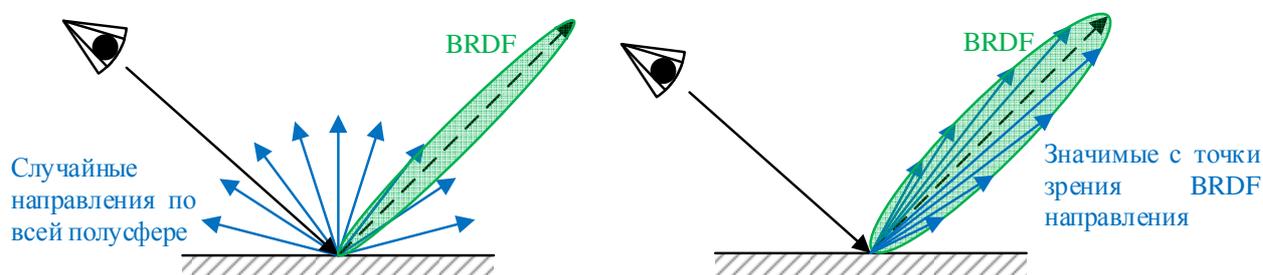


Рисунок 6. Выборка по значимости BRDF

Использование выборки по значимости в методе Монте Карло для решения уравнения рендеринга (3) может быть записано, как:

$$L_{out}(p, w) \approx L_{emitted}(p, w) + \frac{1}{2\pi N} \sum_{i=0}^N BRDF(p, w, w_i) L_{in_i}(p, w_i) \cos(n, w_i), \quad (4)$$

где $PDF(w_i)$ – функция плотности вероятности, связанная с соответствующей функцией $BRDF$. Необходимость использования делителя PDF обусловлена фактом генерации отраженного/преломленного луча согласно функции $BRDF$ (в стандартном алгоритме трассировки путей, представленном ранее лучи генерировались равномерно по полусфере, с центром в точке пересечения с объектом сцены – в этом случае коэффициент делитель, очевидно, был равен 2π). Другими словами, неравномерность выборки необходимо корректировать, чтобы интеграл в правой части уравнения (1) аппроксимировался корректно.

Поддержка прямого/косвенного освещения

Основная идея заключается в рассмотрении всего поступающего в точку света, как состоящего из двух компонент: прямой и косвенной. Первая включает в себя всю энергию, напрямую исходящую от источников света в направлении точки. Вторая же – это отраженные от других объектов лучи.

$$L_{out}(p, w) \approx L_{emitted}(p, w) + \sum_{i=0}^L BRDF(p, w, w') L_{direct}(p, w_i) \cos(n, w') + \frac{1}{N} \sum_{i=0}^N \frac{BRDF(p, w, w_i) L_{indirect}(p, w_i) \cos(n, w_i)}{PDF(w_i)}, \quad (5)$$

где:

- L – общее число источников света сцены;
- $L_{indirect}$ – включает в себя сумму второго и третьего слагаемых уравнения (без первого – $L_{emitted}$!), но для “следующей” точки в трассировке пути, которая лежит в направлении w' относительно точки p . Таким образом, видно, что 3-е слагаемое превращает формулу (5) в рекурсивную.

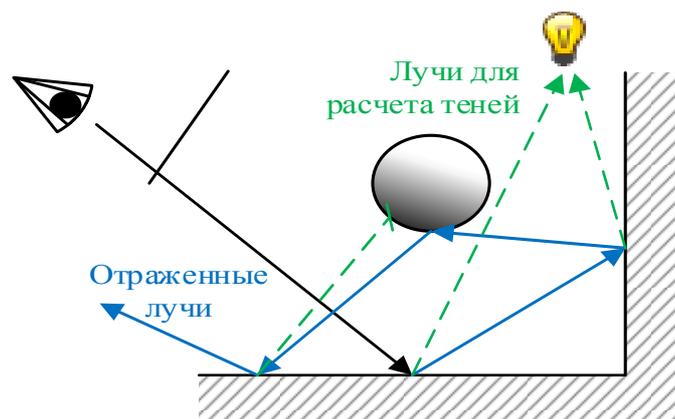


Рисунок 7. Прямое/косвенное освещение

С точки зрения алгоритма такой подход предполагает генерацию 2 типов лучей при взаимодействии (пересечении) с объектом сцены: теневого и отраженного/преломленного. Первый используется для расчета левого слагаемого уравнения (5), второй – преследует те же цели, что и раньше: трассировка пути до источника света, но теперь вносит вклад исключительно косвенной компоненты освещения. Именно поэтому слагаемое $L_{emitted}$

необходимо исключить из дальнейших расчетов косвенного освещения, о чем говорилось в описании Lindirect.

Заключение

Рассмотрены различные методы решения “главного уравнения рендеринга”, на основе которых разработаны соответствующие алгоритмы рендеринга: простейший алгоритм трассировки по методу “бросания лучей (raycasting)”; стандартная трассировка путей по методу Монте Карло. Кроме того, предложены методики повышения производительности рендеринга (пока без ускоряющей структуры): генерация отраженных/преломленных лучей с использованием выборки по значимости BRDF (BRDF importance sampling); поддержка прямого/косвенного освещения, а также качества результирующего изображения: методика lightsampling для “мягких” теней.

Список литературы

1. Фролови А. Фролов. BVH. 2007. [Сайт]. URL: <http://ray-tracing.ru/articles184.html> (дата обращения: Февраль, 2013)
2. A.A. Aleev, S.V. Ivanov, V.A. Kozlov, R.P. Kuibeda, T.V. Kulevoy, S.V. Rogozhkin, A.I., Semennikov, B.Yu. Sharkov, and A.G. Zaluzhny, International Topical Meeting on Nuclear Research Applications and Utilization of Accelerators, Vienna, Austria, 4–8 May 2009, AP/P5_07, p.47.
3. S.Marshner. Path Tracing notes.Февраль, 2012 [Электронный ресурс]. URL: <http://www.cs.cornell.edu/Courses/cs6630/2012sp/notes/07pathtr-notes.pdf> (дата обращения: Март, 2013)
4. S.Popov, I. Georgiev, R. Dimov, and P. Slusallek, "Object partitioning considered harmful: space subdivision for BVHs," in HPG '09 Proceedings of the Conference on High Performance Graphics 2009, New York, NY, USA, 2009, pp. 15-22.
5. U.Assarsson. TDA361 – Computer Graphics course. Pathtracer tutorial. 2013.[Электронный ресурс].URL: <http://www.cse.chalmers.se/edu/year/2013/course/TDA361/Pathtracer.zip> (дата обращения: Апрель, 2013)

Рецензенты:

Гагарина Л. Г., д.т.н., профессор, заведующий кафедрой «Информатика и программное обеспечение вычислительных систем» Национального исследовательского университета «МИЭТ», г.Москва.

Портнов Е.М., д.т.н., профессор кафедры «Информатика и программное обеспечение вычислительных систем», начальник научно-исследовательской лаборатории «Управляющие информационные системы» Национального исследовательского университета «МИЭТ», г.Москва.