



ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

**Жданов Д.Д., Ершов С.В.,
Волобой А.Г.**

Метод подавления
стохастического шума
изображения,
сгенерированного Монте-
Карло трассировкой лучей,
сохраняющий мелкие
детали

Рекомендуемая форма библиографической ссылки: Жданов Д.Д., Ершов С.В., Волобой А.Г. Метод подавления стохастического шума изображения, сгенерированного Монте-Карло трассировкой лучей, сохраняющий мелкие детали // Препринты ИПМ им. М.В.Келдыша. 2018. № 194. 15 с. doi:[10.20948/prepr-2018-194](https://doi.org/10.20948/prepr-2018-194)
URL: <http://library.keldysh.ru/preprint.asp?id=2018-194>

**Ордена Ленина
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В. Келдыша
Российской академии наук**

Д.Д. Жданов, С.В. Ершов, А.Г. Волобой

**Метод подавления стохастического
шума изображения, сгенерированного
Монте-Карло трассировкой лучей,
сохраняющий мелкие детали**

Москва — 2018

Жданов Д.Д., Ершов С.В., Волобой А.Г.

Метод подавления стохастического шума изображения, сгенерированного Монте-Карло трассировкой лучей, сохраняющий мелкие детали

Методы Монте-Карло трассировки лучей позволяют получать физически корректные изображения виртуальных сцен, воспроизводя большое количество оптических эффектов. Однако получаемое изображение содержит шум, присущий стохастическим алгоритмам. Предложен оригинальный метод подавления шума в этих изображениях. Мелкомасштабные детали изображения не искажаются, т.к. они берутся из вспомогательного, опорного изображения, построенного детерминистической трассировкой. Опорное изображение позволяет вычленивать компоненту вторичного освещения, которая оказывается обычно плавной функцией и допускает эффективную фильтрацию. После умножения её результата на вспомогательное детерминистическое изображение получается резкая картинка с сохранением мелких деталей и с сильно сниженным шумом.

Ключевые слова: реалистичная визуализация, трассировка лучей, метод Монте-Карло, подавление шума, фильтрация

Dmitry Dmitrievich Zhdanov, Sergey Valentinovich Ershov, Alexey Gennadievich Voloboy

Method of denoising for the image generated by Monte Carlo ray tracing preserving small details

Monte-Carlo methods of ray tracing allow generating physically correct images of virtual scenes, reproducing a large number of optical effects. However, the resulting image contains noise inherent in stochastic algorithms. An original method of denoising in these images is proposed. Small-scale details of the image are not distorted because they are taken from an auxiliary, reference image constructed by a deterministic ray tracing. The reference image allows you to extract the secondary lighting component which is usually a smooth function and allows for efficient filtering. After multiplying its result by an auxiliary deterministic image, a sharp picture is obtained with preservation of small details and with greatly reduced noise.

Key words: realistic rendering, ray tracing, Monte-Carlo ray tracing, filtering, denoising

1. Введение

Изображения, синтезированные методом двунаправленной стохастической трассировки лучей, как правило, содержат значительный шум, для устранения которого требуется значительное время вычислений. В зависимости от метода вычислений, используемого для построения изображения, шум может иметь различный вид: и «однородная» сравнительно небольшая случайная вариация яркости, и одиночные яркие точки, и кластеры, состоящие из десятков точек. В любом случае мелкие или неконтрастные элементы изображения могут полностью теряться на уровне шума.

Одним из способов борьбы с шумом является фильтрация изображения, однако даже билатеральная фильтрация может размыть мелкие детали изображения. Поэтому поиску «оптимального» фильтра, который удаляет шум и не разрушает мелкую структуру синтезированного изображения, посвящено большое количество современных исследований.

Фильтрация сигнала применялась для снижения шума давно, но эффективность ее была невысока. Шум подавляется за счет усреднения, вычисляющего значение в точке фильтрованной функции в виде суммы по многим точкам исходной функции. Поэтому для «сильного» сглаживания надо использовать много точек, и для традиционных фильтров это означало усреднение по большому «окну», приводящему к подавлению высокочастотных регулярных составляющих сигнала. Хорошее описание идей и основ методов фильтрации было дано в [1]–[3]. В традиционных линейных фильтрах, основанных на свёртке сигнала с ядром, более или менее однородным в пределах «окна усреднения», требования сохранения деталей входного сигнала и усреднения по большой области были взаимоисключающими. Известные способы преодоления этого — использование разложения по вейвлетам, лапласовской пирамиды (уровней детализации) и подобные [4].

Также достаточно давно известен «двойной» (bilateral) фильтр. Он нелинеен, т.е. ядро усреднения зависит от сигнала: оно исчезает там, где входная функция значительно отклоняется от своего значения в центре фильтра. В результате он хорошо сохраняет «края» изображений, практически не размывая их. Однако низкоконтрастную текстуру наподобие листвы дерева он полностью размывает. В последнее время основная идея этого фильтра была возрождена и развита. Основная идея его развития заключается в том, что фильтр расширяется некоторым критерием, определяющим влияние каждой точки области на результат фильтрации. Например, исходно фильтр исключал из усреднения точки с сильно отличным уровнем сигнала. Кроме того, можно дополнить этот критерий сравнением градиента или какой-то «сопутствующей» сигналу информации. Например, при фильтрации изображений, полученных трассировкой лучей, это может быть поле нормалей [1]. Однако использование в критерии как самого сигнала, так и тем более его градиента работает плохо при высоком уровне шума, который доминирует в отклонении. Для преодоления этого можно заменить локальное сравнение на сравнение усредненного (по «предвари-

тельному фильтру») сигнала или на сравнение гистограмм [1]. И, наконец, можно сравнивать не точечные значения, а пространственные распределения в некоторых окрестностях точек. То есть при фильтрации зашумленного изображения текста будем сравнивать «окна» размером с букву. Если они различаются сильно, точки эффективно исключаются из усреднения. В идеале фильтр произведет усреднения всех пикселей буквы «а» по всем (даже далеким) аналогичным пикселям других букв «а». Поскольку усреднение пойдет по окнам с идентичным незашумленным изображением, то никаких его деталей оно не разрушит. И, напротив, если в фотографии много букв «а», то усреднение совершенно сгладит шум. Такой нелокально-нелинейный фильтр описан в [2]. Для периодических или почти периодических изображений он работает превосходно: например, если на исходной картинке показаны обои, но их рисунок совершенно неразличим из-за шума, то фильтрованная картинка проявляет их периодический рисунок с сохранением деталей, и шум почти исчезает. Для совершенно «хаотической» текстуры наподобие волос, меха или листвы результаты много хуже. Но они все еще не сравнимы с «традиционными» фильтрами.

Фильтры на основе такого нелокально-нелинейного усреднения сейчас популярны, и было предложено много путей их развития [1]. Но и они хорошо работают не во всех случаях. Кроме того, такой подход вычислительно дорог, т.к. номинально область усреднения очень велика, и для каждой точки используется вся область, хотя некоторые точки «по критерию» и получают исчезающе малый, но не нулевой вес.

Другая группа подходов связана с использованием априорной информации о сигнале. Это возможно не всегда. Но для изображений, полученных трассировкой лучей, это часто достижимо. Например, предположим, что у нас плоскость с нанесенной текстурой освещена плавно меняющимся по пространству светом (например, это газета на стенде, освещенная небом). Изображение, полученное трассировкой лучей, может быть очень сильно зашумлено, однако нам известно заранее, что оно есть произведение плавного сигнала на высокочастотный модулятор (текстуру), который может быть вычислен детерминистическим образом. Тогда мы можем отфильтровать только освещение, поскольку оно не содержит деталей и хорошо обрабатывается даже простыми и быстрыми фильтрами. Далее результат умножается на известную текстуру. В итоге детали сохраняются и шум уходит. Помимо описанного, возможны и иные способы декомпозиции изображения, например [5]. В зависимости от способа трассировки, структуры сцены и других параметров выгодной может быть различная декомпозиция. Вся эта группа методов не является чистой фильтрацией, поскольку она тесно интегрирована в процесс трассировки лучей методом Монте-Карло. До некоторой степени к ней можно отнести и мутации трасс лучей. Мутации наиболее широко применяются при моделировании глобальной освещенности методом Метрополиса [6], однако могут быть использованы и совместно со «стандартным» методом Монте-Карло.

2. Идея метода

Основная идея предложенного метода заключается в использовании опорного изображения, полученного методом детерминистической трассировки лучей с идеальным отображением всех мелких деталей, включая четкие границы объектов и пространственную модуляцию оптических свойств (как правило, текстуру). Поскольку большинство детерминистических методов не в состоянии обеспечить прецизионный расчет вторичного освещения за ограниченное время, соизмеримое со временем фильтрации, в качестве вторичного освещения используется однородное единичное освещение. В результате, полученное изображение представляет собой распределение коэффициентов яркости видимых объектов сцены.

Альтернативный подход заключается в использовании методов Монте-Карло для расчета карт вторичного освещения в опорном изображении. Несмотря на то что освещенность, вычисленная данным методом, содержит шумовую составляющую, которая может быть практически сведена к нулю за счет использования простых фильтров, ее значение, как правило, на порядок меньше, чем шум в изображении, синтезированном методом двунаправленной стохастической трассировки лучей. В данном случае распределение освещенности будет более корректным, и если его шумовая составляющая достаточно мала, то это изображение может использоваться в качестве опорного.



Рис. 1. Примеры изображений. Левое изображение получено двунаправленной стохастической трассировкой, правое изображение — детерминистической трассировкой при однородном единичном освещении.

Необходимо отметить, что для большинства сцен вариация освещенности по поверхности сцены является плавной функцией, и локально эта функция может аппроксимироваться однородным освещением. Поэтому, в отличие от быстропеременной функции яркости, освещенность поверхности может быть достаточно корректно отфильтрована. Рис. 1 демонстрирует два изображения одной сцены: левое было получено методом двунаправленной стохастической

трассировки лучей и содержит значительную шумовую составляющую. Правое же представляет собой опорное изображение, которое было получено методом детерминистической трассировки лучей с использованием однородного единичного освещения.

Следовательно, основная идея метода заключается в том, чтобы применять методы фильтрации не к высокочастотной зашумленной компоненте вторичной яркости изображения, а к компоненте псевдо-освещенности, полученной масштабированием данной яркости на компоненту яркости равномерно освещенной сцены (т.е. коэффициент яркости). Это позволяет исключить высокочастотную пространственную составляющую яркости, вызванную, например, текстурой, из процесса фильтрации и применять более простые алгоритмы для удаления шума монотонно меняющейся освещенности. По окончании процесса фильтрации происходит обратное масштабирование полученного распределения освещенности, т.е. умножение на коэффициент яркости, что дает нам отфильтрованное изображение в единицах яркости. Кроме того, для повышения качества фильтрации при синтезе детерминистического и стохастического изображений сохраняется ряд дополнительных данных о сцене и изображении в точке наблюдения, например, история детерминистической трассы луча, содержащая ориентацию нормалей, дистанции, индексы частей, которые были получены при расчете яркости равномерно освещенной сцены, ошибка вычисления яркости. Все эти данные затем используются в процессе фильтрации изображения.

3. Вычислительный алгоритм

Яркость в точке наблюдения имеет две составляющие: яркость объектов, видимых непосредственным образом, и яркость объектов, видимых через прозрачные или зеркальные поверхности. Рассмотрим первый случай, когда луч, испущенный камерой, попадает на диффузную поверхность без зеркального преобразования на своей трассе. В этом случае видимая яркость есть

$$L = t \int f(\mathbf{v}, \mathbf{u}) F(\mathbf{u}) (\mathbf{u} \cdot \mathbf{n}) d^2 \mathbf{u}, \quad (1)$$

где \mathbf{v} — направление наблюдения, \mathbf{u} — направление освещения, \mathbf{n} — направление локальной нормали в точке освещения, f — двунаправленная функция рассеивания (BDF), t — значение текстуры, модулирующее цвет в точке наблюдения, и F — освещенность в точке наблюдения. В случае равномерной единичной освещенности выражение (1) сводится к

$$L_a = t \int f(\mathbf{v}, \mathbf{u}) (\mathbf{u} \cdot \mathbf{n}) d^2 \mathbf{u}. \quad (2)$$

Как правило, отношение

$$z \equiv \frac{L}{L_a} \quad (3)$$

является медленно меняющейся функцией, которая допускает фильтрацию. Резкие изменения z возможны, когда:

- 1) освещенность меняется слишком быстро, например, в областях ярких бликов, вблизи от источника света или на границах тени. Однако в большинстве случаев *вторичная* освещенность меняется медленно и эти эффекты не оказывают существенного влияния на качество фильтрации;
- 2) нормаль быстро меняется вблизи от точки наблюдения, например, в случае рифленой поверхности.

Естественным решением будет выполнение фильтрации в областях, где функция z является гладкой, т.е. допускает линейную или параболическую аппроксимацию. Однако проблема остается для случая сильно зашумленного z , когда шум преобладает над сигналом, и отличие оригинального z от фильтрованного не может быть точно определено.

3.1. Выбор радиуса фильтра

В предлагаемом алгоритме для фильтрации всегда выбирается прямоугольная область с центром в заданной точке (x_0, y_0) , и основная задача заключается в том, чтобы найти ее оптимальный размер. Поиск размера области фильтрации начинается с размера 5×5 точек и идет в сторону ее увеличения.

Увеличение размера области выполняется до тех пор, пока не срабатывает один из критериев остановки. Во-первых, это критерий качества, который говорит о том, что достигнутый размер области достаточен для того, чтобы снизить уровень шума до требуемого уровня. Во-вторых, это достижение максимально допустимого размера области фильтрации. В-третьих, это критерий сильной вариации яркости в исследуемой точке, например, граница тени. Этот критерий срабатывает, когда отклонение значений фильтруемой освещенности в центральной точке, полученное аппроксимацией по области фильтрации и четырем ее подобластям (данный алгоритм аппроксимации описан в разделе 3.2), превышает допустимый уровень. Необходимо отметить, что сильная вариация нормалей, вариация видимых частей и разрывы в видимой геометрии сцены не являются причиной прекращения увеличения области фильтрации. Вместо этого точки, которые не соответствуют критерию фильтрации центральной точки, исключаются из процесса аппроксимации освещенности.

Если ни один из критериев остановки увеличения радиуса фильтрации не срабатывает, мы увеличиваем размер на одну точку до тех пор, пока один из критериев остановки не срабатывает.

3.2. Построение локальной аппроксимации

Область фильтрации есть прямоугольник размером $R(x_0, y_0)$ с центром в точке (x_0, y_0) .

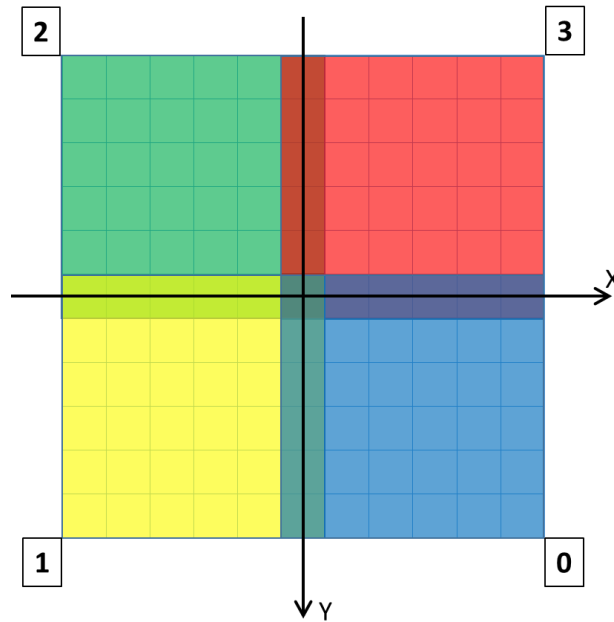


Рис. 2. Область фильтрации

Как во всей области фильтра, так и в четырех его четвертях мы вычисляем плоскость наилучшей аппроксимации $Z = ax + by + c$:

$$\sum_{x,y} w_a(x,y) (ax + by + c - z(x,y))^2 = \min!$$

где (x,y) — координаты точки (пикселя), а w_a является весом, подобным весу билатерального фильтра:

$$w_a(x,y) \equiv e^{-k \frac{|z(x,y) - \langle z \rangle(x,y)|}{\langle z \rangle(x,y)}} e^{-\sigma(x,y)} e^{-\frac{(x-x_0)^2 + (y-y_0)^2}{R(x_0,y_0)^2}},$$

$$\langle z \rangle(x,y) \equiv \frac{1}{N} \sum_{x,y} z(x,y).$$

Здесь $\sigma(x,y)$ — оценка относительного среднеквадратичного отклонения фильтруемой величины.

В аппроксимации участвуют только «хорошие» точки, т.е. те, в которых шум не превышает некоторый допустимый уровень.

Обозначив значение аппроксимирующей плоскости в точке (x,y) как Z , а значение в m -ой четверти аппроксимирующей плоскости как Z_m , мы вычисляем значение аппроксимирующей функции следующим образом:

$$\frac{Z(x,y) + w_m Z_m(x,y)}{1 + w_m},$$

где m — индекс четверти, содержащей точку (x,y) , а вес w_m выбирается как

$$w_m = \max \left(\frac{Z(x,y)}{Z_m(x,y)}, \frac{Z_m(x,y)}{Z(x,y)} \right)^8.$$

3.3. Две стадии усреднения и финальное значение в пикселе

Большинство существующих подходов основываются на том, что для вычисления значения отфильтрованной функции в точке (x_0, y_0) мы выбираем некоторую область вокруг этой точки, находим гладкую аппроксимирующую функцию внутри выбранной области, а её значение в точке (x_0, y_0) и составляет отфильтрованный сигнал.

Предлагаемый подход строится по аналогичному принципу, т.е. выбирается некоторая область вокруг центральной точки (x_0, y_0) и находится гладкая аппроксимирующая функция z (в данном случае построенная по пяти плоскостям) внутри этой области. Однако затем аппроксимирующая функция применяется ко *всем* точкам внутри области и полученные значения *добавляются* в результат фильтрации (с весом). Таким образом, каждая точка результирующего отфильтрованного изображения является взвешенной суммой из всех областей фильтрации, которые захватывали данную точку.

В большинстве случаев предложенный подход является предпочтительным, поскольку он позволяет уменьшить ошибку аппроксимации. Например, если мы возьмем функцию $f = x^2$, то линейная аппроксимация внутри каждой ячейки размера Δ имеет точность порядка $\Delta^2/12$. Следовательно, вычислив значение отфильтрованной функции в центральной точке, мы получим отклонение порядка $\Delta^2/12$. Между тем, это же значение, усредненное по всем точкам области, включая центральную, будет вычислено более корректно. При этом, если усреднение идет с постоянным весом, то искажение *полностью* исчезает, а в случае непостоянного веса оно значительно меньше, чем $\Delta^2/12$.

Хотя вышеизложенное и не может служить доказательством того, что предложенный подход предпочтителен во всех случаях, он был выбран в качестве основы для реализации фильтра изображений. Для каждой точки изображения (x_0, y_0) мы находим область фильтрации $R(x_0, y_0)$, расположенную вокруг нее, и вычисляем аппроксимирующую функцию $Z(x, y)$ для всех точек в этой области, а затем добавляем $w(x, y)Z(x, y)$ к результирующему отфильтрованному распределению.

Вес $w(x, y)$ может варьироваться от нуля до единицы. Кроме перечисленных выше составляющих веса, определяемых отклонением координат точки (x, y) от центральной точки (x_0, y_0) , отклонением яркости в точке (x, y) от яркости в центральной точке (x_0, y_0) и ошибкой вычисления яркости, к значению освещенности в точке (x, y) применяется ряд следующих составляющих веса:

1. Составляющая, определяемая отклонением нормали в точке (x, y) от нормали в точке (x_0, y_0) :

$$w_n(x, y) = \begin{cases} (\vec{n}(x, y) \cdot \vec{n}(x_0, y_0))^4, & (\vec{n}(x, y) \cdot \vec{n}(x_0, y_0)) > 0.5 \\ 0, & (\vec{n}(x, y) \cdot \vec{n}(x_0, y_0)) \leq 0.5 \end{cases}$$

Данная весовая составляющая позволяет не только четко отчерчивать грани объектов, но и хорошо аппроксимировать зубчатый профиль объекта, показанный на рис. 3.

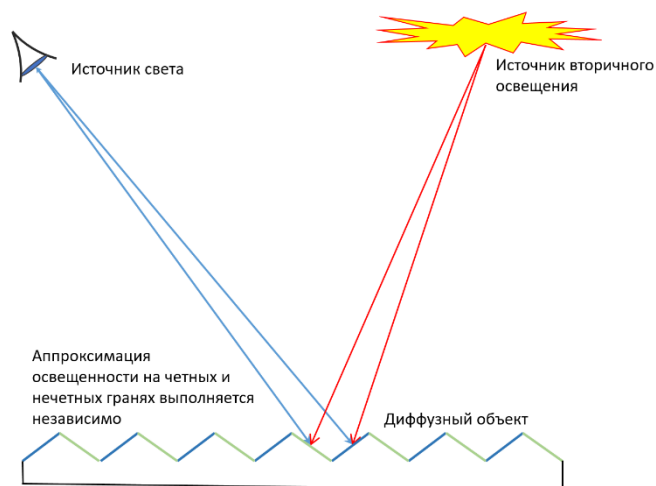


Рис. 3. Аппроксимация зубчатого профиля

Освещенность каждой из четных граней зубца аппроксимируется по освещенности четных граней (т.е. половине области); аналогичным образом аппроксимируется освещенность нечетных граней.

2. Составляющая, формируемая на границе перехода между объектами или на границе одного объекта. Пример границы объекта в виде ступени представлен на рис. 4.

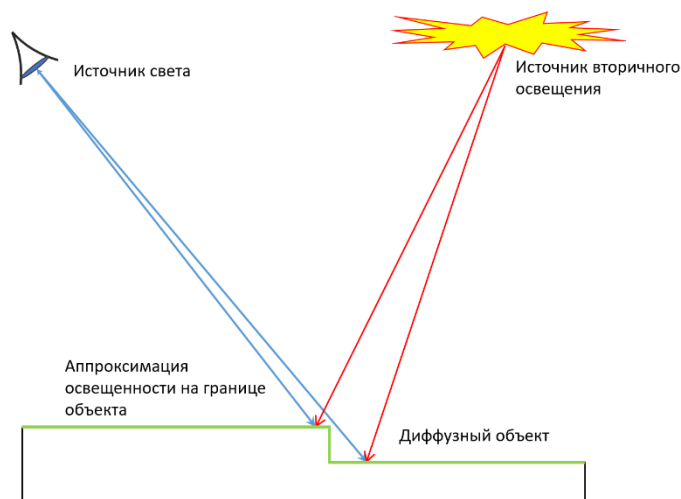


Рис. 4. Аппроксимация на границе объекта

При аппроксимации точки А область на нижней ступени имеет нулевой весовой коэффициент, т.е. исключается из процесса аппроксимации. Если весовой коэффициент оставить единичным, то ступенька визуально исчезнет, поскольку освещенность будет сглажена и однородна, а коэффициент яркости, на который перемасштабируется освещенность, для обеих частей ступени есть величина постоянная. В результа-

те область со ступенью приобретет постоянную яркость. Потому при переходе через границы объектов и на границах одного объекта устанавливается нулевой весовой коэффициент: $w_b(x, y) = 0$.

3. Составляющая, формируемая на частях различных объектов. При визуализации сцены соседние пиксели камеры могут видеть различные участки сцены. Эти участки могут принадлежать одному объекту, но иметь различные свойства или принадлежать разным объектам. Кроме того, совпадение видимых участков сцены может быть частичным. Этот эффект возникает в сценах, содержащих зеркальные элементы, например, при взгляде на частично зеркальный пол камера видит общий объект (пол) и различные объекты сцены, отраженные в нем. В данном случае весовой коэффициент варьируется от 1, когда все видимые объекты и их части в точках (x_0, y_0) и (x, y) совпадают, до 0, когда нет совпадения в точках (x_0, y_0) и (x, y) .

После обработки всех точек (x_0, y_0) мы получаем отфильтрованное изображение $z(x, y)$ и затем переводим его в единицы яркости сцены, осуществляя обратную нормировку.

$$L(x, y) = L_a(x, y)z(x, y).$$

4. Точки, видимые сквозь зеркальные поверхности

До сих пор мы предполагали, что камера видит диффузную поверхность непосредственно, без каких-либо зеркальных событий на трассе луча. Однако в большинстве случаев ситуация более сложная, например, когда мы смотрим в комнату через окно или видим отражение предметов в зеркале. В этом случае луч, идущий от камеры и попадающий на «зеркальную» поверхность, разбивается на отраженный и преломленный лучи, которые в свою очередь идут дальше и аналогичным образом разбиваются на отраженные и преломленные. Этот детерминистический процесс является альтернативой «русской рулетки», используемой в трассировке лучей методом Монте-Карло. Для предотвращения бесконечного суммирования количество актов разбиения «зеркальной» трассы луча ограничено пятью событиями.

Необходимо отметить, что оптические свойства поверхности могут быть частично «зеркальными», например, стол, покрытый лаком. В данном случае происходит не только разбиение одного луча на два, но и накопление яркости на диффузной поверхности. В результате цвет видимой точки на детерминистической трассе луча, идущего от камеры, является суммой всех цветов, накопленных от всех «зеркальных» лучей, включая цвет непосредственно видимой диффузной поверхности.

Аналогичным образом «точное» изображение, которое отличается от детерминистической трассировки лучей только параметрами освещения, также является суммой по всему бесконечному количеству «зеркальных» трасс лучей. То есть «точное» изображение и его детерминистическая контрпара имеют об-

щую часть траектории и часть, отличающуюся остаточной бесконечной суммой траекторий. Как правило, яркость, приносимая остаточной частью траектории, значительно меньше общей, и ее игнорирование в процессе фильтрации не влияет на результат.

Для того чтобы применить разработанный алгоритм фильтрации, двунаправленная стохастическая трассировка лучей сохраняет изображения, полученные для разных историй трассы «зеркального» луча, в отдельных слоях. Аналогичным образом детерминистическая трассировка, формирующая опорное изображение для единичного равномерного освещения, сохраняет отдельные слои изображения для соответствующих трасс «зеркального» луча.

Затем каждый слой изображения фильтруется в соответствии с алгоритмом, описанным в разделе 3. Оставшаяся часть, *превышающая* пять «зеркальных» событий на луче, не фильтруется, поскольку, во-первых, для нее нет соответствующей контрпары, а во-вторых, яркость этой компоненты, как правило, мала и ее шум не оказывает существенного влияния на качество изображения. В конечном итоге, результирующее отфильтрованное изображение есть сумма всех отфильтрованных слоев изображения и исходной остаточной части изображения.

5. Заключение

Как было показано, разработанный алгоритм достаточно хорошо справляется с удалением шумовой составляющей диффузной компоненты изображения, полученного методом двунаправленной стохастической трассировки лучей. Несмотря на то что наблюдается незначительное размытие резких перепадов яркости, границы теней и скачки освещенности, чья ширина превышает несколько пикселей, сохраняются практически без изменения контраста. Кроме того, алгоритм работает без снижения контраста с изображением границ объектов и объектов, содержащих текстуры. В большинстве случаев использование алгоритма фильтрации позволяет значительно снизить (в ряде случаев более чем на порядок) общее время, затрачиваемое на синтез изображения.

Результаты работы алгоритма представлены на рисунках 5 и 6. На рис. 5 представлена сцена Cornel Vox с текстурой на стенках. Верхняя часть рисунка представляет собой эталонное изображение, левая нижняя часть рисунка, содержащая шумовую составляющую, была получена методом двунаправленной стохастической трассировки лучей, а правая нижняя часть рисунка представляет результат фильтрации.

На рис. 6 представлена сцена с интерьером комнаты. Верхняя часть рисунка представляет собой эталонное изображение, левая нижняя часть рисунка, содержащая шумовую составляющую, была получена методом двунаправленной стохастической трассировки лучей, а правая нижняя часть рисунка представляет результат фильтрации.

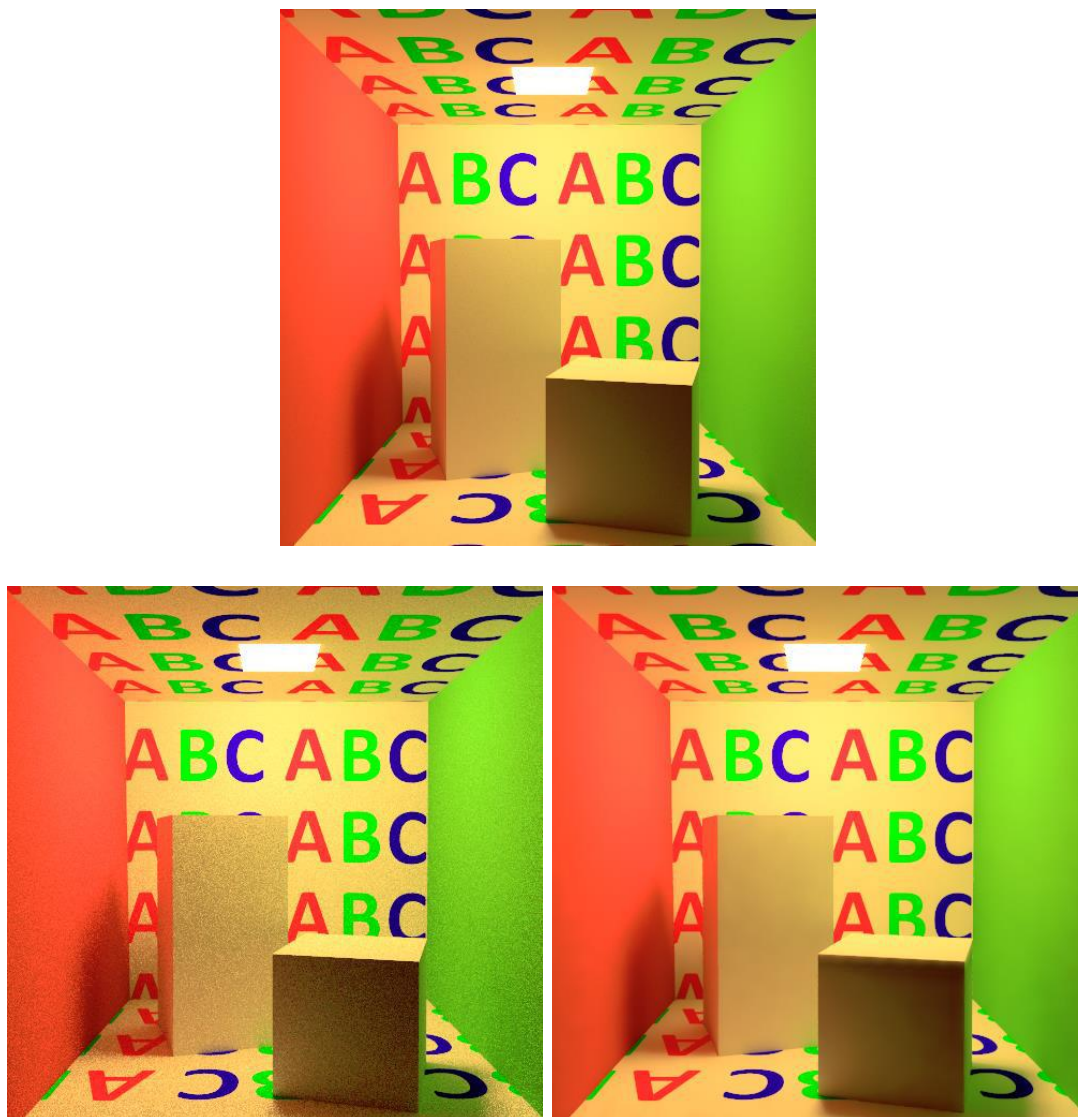


Рис. 5. Изображение сцены Cornell Box. Верхняя часть рисунка — эталонное изображение, левая нижняя часть рисунка была получена методом двунаправленной стохастической трассировки лучей, правая нижняя часть рисунка представляет результат фильтрации

Видно, что фильтр практически полностью удалил шум, не размыл границ мелких объектов и сохранил текстуру, наложенную на объекты сцены.

Основные направления развития данного алгоритма заключаются в повышении его вычислительной эффективности и возможности его адаптации к фильтрации каустической составляющей изображения.



Рис. 6. Изображение комнаты. Верхняя часть рисунка – эталонное изображение, левая нижняя часть рисунка была получена методом двунаправленной стохастической трассировки лучей, правая нижняя часть рисунка представляет результат фильтрации

6. Библиография

- [1] Bitterli B., Rousselle F., Moon B., Iglesias-Gutián J. A., Adler D., Mitchell K., Jarosz W., and Novák J. Nonlinearly Weighted First-order Regression for Denoising Monte Carlo Renderings. In: Proceedings of EGSR, 2016. Vol. 35.
- [2] Takeda H., Farsiu S., Milanfar P. Kernel regression for image processing and reconstruction. IEEE Transactions on Image Processing 16, 2 (Feb 2007), 349–366.
- [3] Buades A., Coll B., Morel J.-M. A review of image denoising algorithms, with a new one. Multiscale Modeling & Simulation 4, 2 (2005), 490–530.
- [4] Dammertz H., Sewtz D., Hanika J., Lensch H. P. A. Edge-avoiding à-tours wavelet transform for fast global illumination filtering. In Proc. Conf. on High Performance Graphics (2010), HPG '10, Eurographics Association, pp. 67–75
- [5] Mara M., McGuire M., Bitterli B., Jarosz W. An Efficient Denoising Algorithm for Global Illumination. Proceedings of HPG '17, July 28-30, 2017, <https://doi.org/10.1145/3105762.3105774>
- [6] Фролов В.А., Галактионов В.А. Компактная по памяти реализация алгоритма METROPOLIS LIGHT TRANSPORT на графических процессорах // Программирование, 2017, № 3, с. 83-92.

Оглавление

1. Введение.....	3
2. Идея метода	5
3. Вычислительный алгоритм	6
3.1. Выбор радиуса фильтра	7
3.2. Построение локальной аппроксимации	7
3.3. Две стадии усреднения и финальное значение в пикселе.....	9
4. Точки, видимые сквозь зеркальные поверхности.....	11
5. Заключение	12
6. Библиография	14