



ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 186 за 2018 г.



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

Никерова О.А., [Фролов В.А.](#),  
[Волобой А.Г.](#)

Фильтрация результатов  
Монте-Карло трассировки  
лучей по нескольким  
изображениям

**Рекомендуемая форма библиографической ссылки:** Никерова О.А., Фролов В.А., Волобой А.Г. Фильтрация результатов Монте-Карло трассировки лучей по нескольким изображениям // Препринты ИПМ им. М.В.Келдыша. 2018. № 186. 13 с. doi:[10.20948/prepr-2018-186](https://doi.org/10.20948/prepr-2018-186)  
URL: <http://library.keldysh.ru/preprint.asp?id=2018-186>

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

**О.А. Никерова, В.А. Фролов, А.Г. Волобой**

**Фильтрация результатов  
Монте-Карло трассировки лучей  
по нескольким изображениям**

**Москва — 2018**

*Никерова О.А., Фролов В.А., Волобой А.Г.*

**Фильтрация результатов Монте-Карло трассировки лучей  
по нескольким изображениям**

В данной работе предлагается метод фильтрации с использованием нескольких входных изображений, полученных Монте-Карло трассировкой лучей при небольших изменениях положения виртуальной камеры внутри сцены. Отдельно рассмотрен предельный случай при отсутствии сдвига камеры, фактически вырождающийся в распределение независимых Монте-Карло выборок по нескольким изображениям. Данный подход не ограничивает способ генерирования Монте-Карло выборок и позволяет использовать преимущества таких методов, как трассировка лучей на GPU и квази Монте-Карло. В отличие от существующих аналогов, в предлагаемом методе учитывается разброс значений Монте-Карло выборок для каждого виртуального суб-пикселя, что предоставляет возможность для улучшенной борьбы с выбросами и сглаживания границ при сохранении деталей на изображении.

**Ключевые слова:** реалистичная визуализация, Монте-Карло трассировка лучей, фильтрация шума, фильтрация изображений, трассировка лучей на GPU

*Olga Aleksandrovna Nikerova, Vladimir Aleksandrovich Frolov, Alexey Gennadievich Voloboy*

**Filtering of the results of Monte Carlo ray tracing using multiple images**

This paper presents a method of filtering using several input images obtained by Monte Carlo ray tracing with small changes in the position of the virtual camera inside the scene. The limiting case with the absence of a camera shift, which actually degenerates into a distribution of independent samples to several Monte Carlo images, is considered separately. This approach does not impose any restrictions on the method of generating Monte Carlo samples and allows to take advantage of methods such as ray tracing on GPU and quasi-Monte Carlo. Unlike existing analogues, the proposed method takes into account the spread of Monte Carlo samples' values for each virtual sub-pixel, which provides an opportunity for handling outliers and smoothing of aliasing effects while preserving details in the image.

**Key words:** realistic rendering, Monte-Carlo ray tracing, noise reduction, image filtering, GPU ray tracing

## 1. Введение

Методы Монте-Карло трассировки лучей позволяют получать физически корректные изображения виртуальных 3D сцен, воспроизводя большое количество оптических эффектов. Однако для получения высококачественного результата требуется провести трассировку очень большого количества лучей, что приводит к долгому ожиданию итогового результата для пользователя. Если снизить количество трассируемых лучей, можно добиться быстрого получения шумного результата, который затем можно отфильтровать для получения сглаженного фотореалистичного изображения. Методы, использующие данную особенность, позволяют существенно снизить временные затраты на визуализацию при некоторой потере качества результата. Таким образом, поставленной задачей фактически является удаление шума на промежуточном изображении с оптимальным балансом между визуальной гладкостью результата и сохранением границ объектов и оптических эффектов. Для создания исходного шумного изображения подходят методы и прямой, и обратной, и двунаправленной трассировки.

## 2. Обзор существующих методов

Методы фильтрации результатов Монте-Карло трассировки, основывающиеся на шумных данных, можно условно разделить на два направления [1]. Первое – фильтрации множества Монте-Карло выборок как многомерного сигнала. Второе – фильтрация изображений как двумерного сигнала с использованием дополнительной информации о сцене. Второе направление обычно приводит к более сильным (в плане фильтрующей способности), легко масштабируемым и быстрым алгоритмам (работа [2]). Однако, работая на уровне пикселей, эти подходы не могут восстановить мелкие детали на изображении, т.к. нужная информация полностью отсутствует во входных данных. Подобная информация учитывается при работе на суб-пиксельном уровне, как в методах из первой группы (работы [3], [4], [5]). Однако данные методы более вычислительно-ёмкие.

Фильтрация результатов трассировки на основе Монте-Карло выборок часто сопровождается адаптивным распределением новых выборок. В данном классе методов на первом шаге вычисляется результат с малым количеством выборок на пиксель. Затем алгоритмы итеративно фильтруют изображение и вычисляют, в итоговых значениях каких пикселей нет уверенности, и рассчитывают для них новые выборки. Например, в [3] предложен алгоритм, использующий вейвлет-преобразования разных масштабов для фильтрации и определения регионов с высокой дисперсией. В аналогичном подходе [4] в качестве фильтрации используется фильтр Гаусса. Оба метода значительно опираются на рассчитанную дисперсию значений пикселей, поэтому погрешность в ее расчете из-за малого количества выборок на начальных итерациях алгоритма может сказаться на плохо освещенных участках сцены.

Также на границах объектов могут возникать неточности из-за необходимости применять фильтры разного масштаба для разных поверхностей, представленных одним пикселем.

Чаще всего адаптивное распределение выборок улучшает результаты на границах объектов и оптических эффектах в изображениях. Подобного эффекта можно достичь и с помощью адаптации методов фильтрации, успешно применяемых для обработки шумных фотографий, где ядро фильтра рассчитывается индивидуально для каждого пикселя. Например, в статье [6] предлагается применить к результатам трассировки билатеральную фильтрацию, в которой вес соседних пикселей зависит от степени схожести цвета этих пикселей с центральным. В подходе non-local means, предложенном в [7], для определения веса сравниваются не значения самих пикселей, а значения пикселей в их заданной окрестности.

В [5] используется совмещение подходов с адаптивным распределением выборок и использованием фильтра non-local means. Интересен подход, использованный авторами для оценки ошибки на каждой итерации адаптивного распределения: вычисляются и сравниваются два изображения, каждое из которых строится на основе половины выборок. Несмотря на улучшения в качестве по сравнению с предыдущими подходами, в данном подходе сохраняются артефакты на пикселях, содержащих мелкие или перекрывающиеся объекты или эффекты.

Помимо шумных изображений, в качестве исходных данных для фильтрации может быть использована дополнительная информация, которую легко можно получить в процессе трассировки. В подходе [2] для каждого пикселя сохраняется информация о нормали к поверхности и глубине (расстоянии от камеры до объекта) для луча, проходящего через центр каждого пикселя. Подход не слишком хорошо работает для оптических эффектов, поскольку ядро сглаживающего фильтра в большой степени зависит от исходной геометрии сцены, не учитывающей нюансы освещения.

В качестве дополнительной информации может использоваться также информация о способе трассировки. Авторы статьи [8] отмечают, что зачастую шум на изображении появляется в результате противоречий в результатах трассировки для соседних выборок, т.е. в регионах, где значения полученных освещенностей находятся в зависимости от случайных параметров трассировки. Входные и выходные данные для каждого пикселя рассматриваются как наборы случайных величин, и вычисляется статистическая зависимость между ними. Для сглаживания изображения авторы применяют билатеральный фильтр, и вычисленные шумные пиксели используются в ядре фильтра с меньшим весом.

Авторы метода [9] используют в качестве дополнительной информации изображение, полученное при помощи виртуальной вспышки. Для создания подобного изображения в сцене помещается дополнительный источник освещения близко к камере, эмулируя вспышку от фотоаппарата. Для

ускорения построения изображения со вспышкой авторы переиспользуют лучи, с помощью которых было получено оригинальное шумное изображение сцены. Полученное изображение потенциально содержит информацию об отражательных и преломляющих способностях объектов вдобавок к данным о текстурах и геометрии сцены. Для реконструкции итогового изображения в статье используется фильтр non-local means, изображение с виртуальной вспышкой служит для определения весов пикселей. Поскольку описанный подход сглаживал бы сложные оптические эффекты (например, каустики), в данной работе сглаживание производится только внутри множеств гомогенных пикселей. Гомогенными считаются те пиксели, у которых схожи средние значения Монте-Карло выборки. Несмотря на достоинства использования изображения с виртуальной вспышкой, при недостаточном количестве выборок оно не всегда является надежным источником информации. Границы на изображении могут оказаться нечеткими, вдобавок из-за переиспользования лучей данный артефакт будет присутствовать также и на шумном изображении, усугубляя ошибку при фильтрации.

В работе [10] был использован гибридный подход, совмещающий в себе фильтрацию и кэш освещенности. Метод достигает хорошей фильтрующей способности и сохраняет текстуры за счёт разделения освещения на компоненты, а также фильтрации освещения, отделённого от текстур. Однако данный метод накладывает ограничения на способ построения изображения (необходимо считать отдельные компоненты освещения) и может ошибаться на границах объектов.

В настоящее время для многих задач обработки изображений активно используются методы машинного обучения, и область фильтрации результатов Монте-Карло трассировки также получила дополнительное развитие с использованием подобных методов. Подобные подходы применяются как для автоматического подбора весов для заранее заданных ядер фильтрации (работа [11], фильтрация по выборкам), так и для полноценного восстановления конечного изображения из исходных данных «напрямую». В работе [12] использована нейронная сеть с архитектурой автокодировщик с добавочными рекуррентными связями и пропусками промежуточных слоев (англ. skip connections). В качестве входных данных используются Монте-Карло выборки (1 выборка на пиксель), нормали, глубина и текстура соответствующих объектов. Подобная архитектура позволяет обрабатывать шумные данные со скоростью, приближенной к режиму реального времени, и с удовлетворительным для человеческого взгляда результатом. Однако использование лишь одной выборки на пиксель не позволяет добиться высокого качества деталей. При увеличении количества выборок на пиксель метод уступает аналогам на некоторых сценах.

### 3. Предпосылки к выбору метода

Фильтрация только на основе одного шумного изображения приводит к артефактам на границах объектов, поскольку цвета на границах геометрических объектов вычисляются из значений противоречащих друг другу выборок, соответствующих разным поверхностям [10]. При использовании дополнительной информации о сцене задачу усложняет тот факт, что на эти пиксели приходится более одного значения нормали и глубины. В результате в окрестности подобных пикселей не оказывается соседей с высоким весом, и их значения не могут быть отфильтрованы. Сохранение же детальной информации обо всех выборках труднореализуемо для рендер-систем на GPU, поскольку выборки, как правило, генерируются быстрее, чем фильтр может их обработать. Отдельно следует отметить, что при адаптивном распределении выборок существенно усложняется программная реализация учёта вклада выборок на GPU.

Вместо полной информации о выборках было решено использовать несколько изображений, полученных Монте-Карло трассировкой независимо друг от друга. Для сглаживания на границах объектов каждое из изображений получено при небольшом сдвиге камеры вбок относительно начального положения. Таким образом достигается эффект, схожий с сохранением информации о выборках. Фактически за счет смещения в каждом отдельном изображении сохраняется суб-пиксельная информация о смежных пикселях объектов.

Важно отметить, что того же эффекта можно было бы достичь, просто увеличив разрешение изображения или синтезировав его по кускам. Первый способ трудно применим для больших разрешений из-за огромных объемов необходимой памяти. Второй способ позволяет обойтись тем же количеством памяти, что и при рендере оригинального изображения. Однако, во-первых, он не позволит пользователю прогрессивно видеть всё изображение во время расчёта. Во-вторых, этот способ сделал бы невозможным использование квазислучайных положений с низкой невязкой [13] для вычисления положения суб-пикселей, улучшающих сглаживание на границах объектов по сравнению с положением суб-пикселей по регулярной сетке. Наконец, рендер изображения по кускам исключает возможность использования метода MLT [14] для расчёта, поскольку этот алгоритм работает сразу для всего изображения.

### 4. Предложенный метод

В исследованном методе путем Монте-Карло трассировки генерируется несколько изображений,



*Рис. 1.*  
Иллюстрация трёхмерного окна фильтра. Разным цветом обозначены различные изображения

полученных при небольших сдвигах виртуальной камеры внутри сцены. Для вычисления оптимального сдвига относительно исходного положения был использован алгоритм Хаммерсли [13]. На основе указанного алгоритма вычисляются квазислучайные смещения для Монте-Карло лучей. Использование полученного смещения при генерации выборок для отдельных входных изображений эквивалентно сдвигу виртуальной камеры. Для каждого такого изображения также сохраняется дополнительная информация: карта глубин, карта нормалей и карта текстур объектов сцены.

К полученным шумным изображениям в широком динамическом диапазоне (HDR) был применен билатеральный фильтр с трёхмерным окном, затрагивающий суб-пиксели из других шумных изображений (подробнее в разделе «Билатеральная фильтрация»). Такой подход позволяет автоматически разделить все суб-пиксели на похожие (по нормалям, глубине, текстуре и цвету) множества.

Однако применение билатеральной фильтрации только к одному результирующему изображению не позволяет добиться гладкого эффекта на границах объектов (см. рис. 2 и рис. 3) и оставляет несглаженными отдельные выбросы. Проблема может быть устранена при использовании усреднения по нескольким отфильтрованным изображениям. В данной работе усреднение производилось попиксельно по 4 и 16 исходным отфильтрованным изображениям в узком динамическом диапазоне (LDR) с равным вкладом каждого из изображений в составляющие RGB цвета.

Поскольку итоговый фильтр работает над суб-пикселями с большим количеством усреднённых выборок, отсутствует необходимость хранения всех изображений в памяти GPU. Нет необходимости хранить их целиком и в обычной оперативной памяти. Все рассчитанные изображения сохранялись на диск, после чего фильтр имеет возможность загружать изображения по частям с небольшим перекрытием. При этом пользователь в процессе расчёта может видеть как просто усреднённое изображение, так и прогрессивно улучшаемое отфильтрованное изображение (сначала по 1 входному изображению, затем по 2 и т.д.).



*Рис. 2.* Слева направо: одно из исходных шумных изображений, применение билатеральной фильтрации, усредненные шумные изображения, усредненные отфильтрованные изображения со смещением, эталон



## 5. Билатеральная фильтрация

При вычислении весов пикселей для билатеральной фильтрации были использованы нормализованные веса, основанные на расстоянии между пикселями ( $\mu_d$ ), отличиях в цвете для текстур и шумного изображения ( $\mu_c$ ), отличиях в нормалях ( $\mu_n$ ) и глубине ( $\mu_z$ ). Для расчета весов расстояния и разницы в цвете использовались формулы из [8]. Расчет веса расстояния между пикселями:

$$\mu_d = e^{-\frac{1}{2 \cdot \sigma^2 \cdot \Delta d}},$$

где  $\Delta d$  – евклидово расстояние между пикселями,  $\sigma$  – настраиваемый параметр.

Для учета разницы в текстурах объектов сцены и полученного при трассировке шумного изображения было использовано сравнение цветов пикселей в цветовом пространстве RGB:

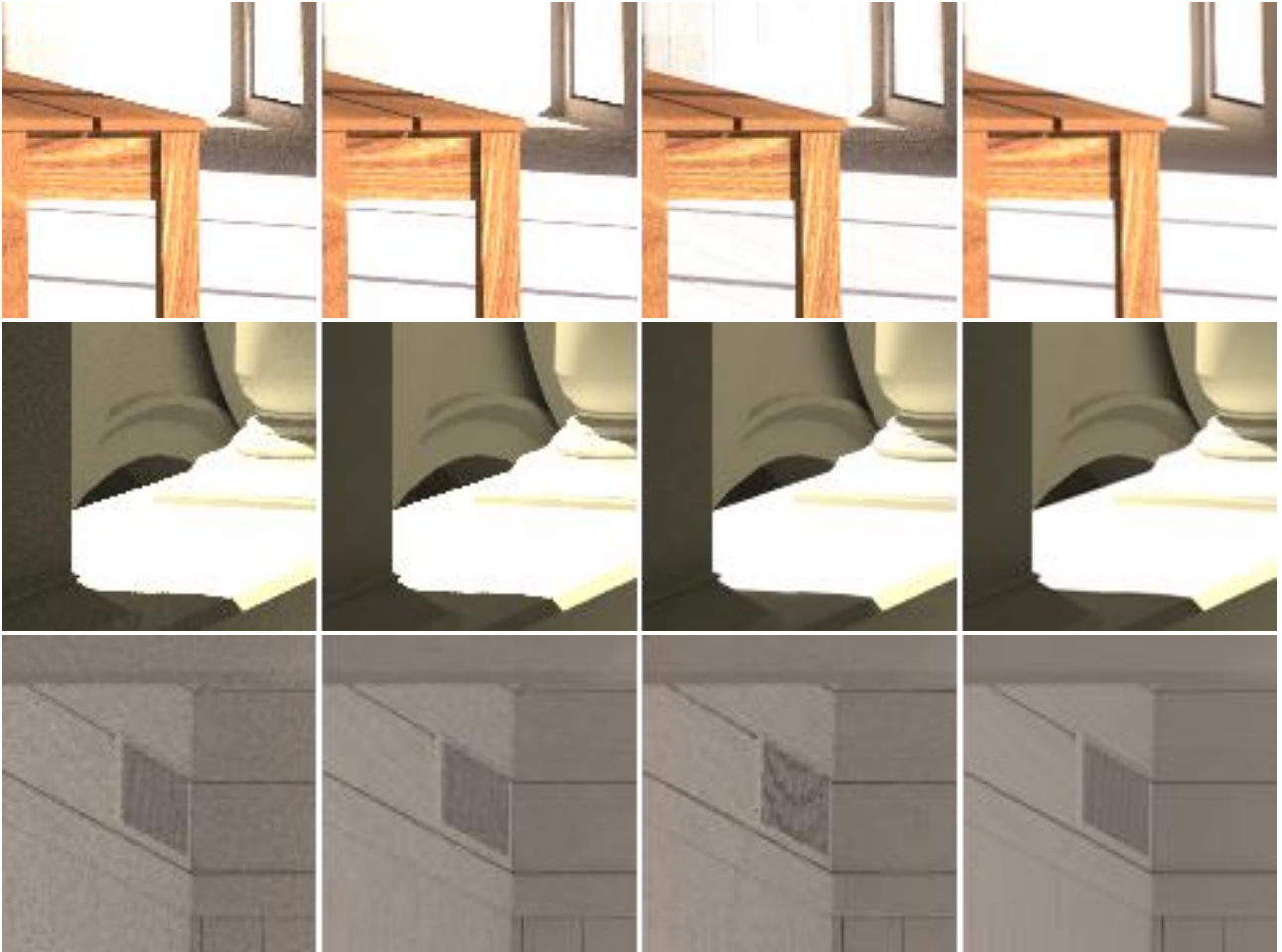
$$\mu_c = e^{-\frac{1}{2 \cdot k^2 \cdot \Delta c}},$$

где  $\Delta c$  – суммарная квадратичная разница цветовых каналов, а  $k$  – настраиваемый параметр.

Для учета особенностей геометрии сцены и сохранения границ объектов учитывались разница в нормалях и глубине объектов. В обоих случаях при слишком большой разнице значений (больше порогов  $z_{max}$  и  $n_{max}$  для глубины и нормалей соответственно) вес принимался равным 0. Веса рассчитывались по следующим формулам:

$$\mu_z = \sqrt{1 - \frac{\Delta z}{z_{max}}}, \mu_n = \sqrt{1 - \frac{\Delta n}{n_{max}}},$$

где  $\Delta z$  – абсолютная разница глубин для пикселей,  $\Delta n$  – евклидово расстояние между векторами нормалей.

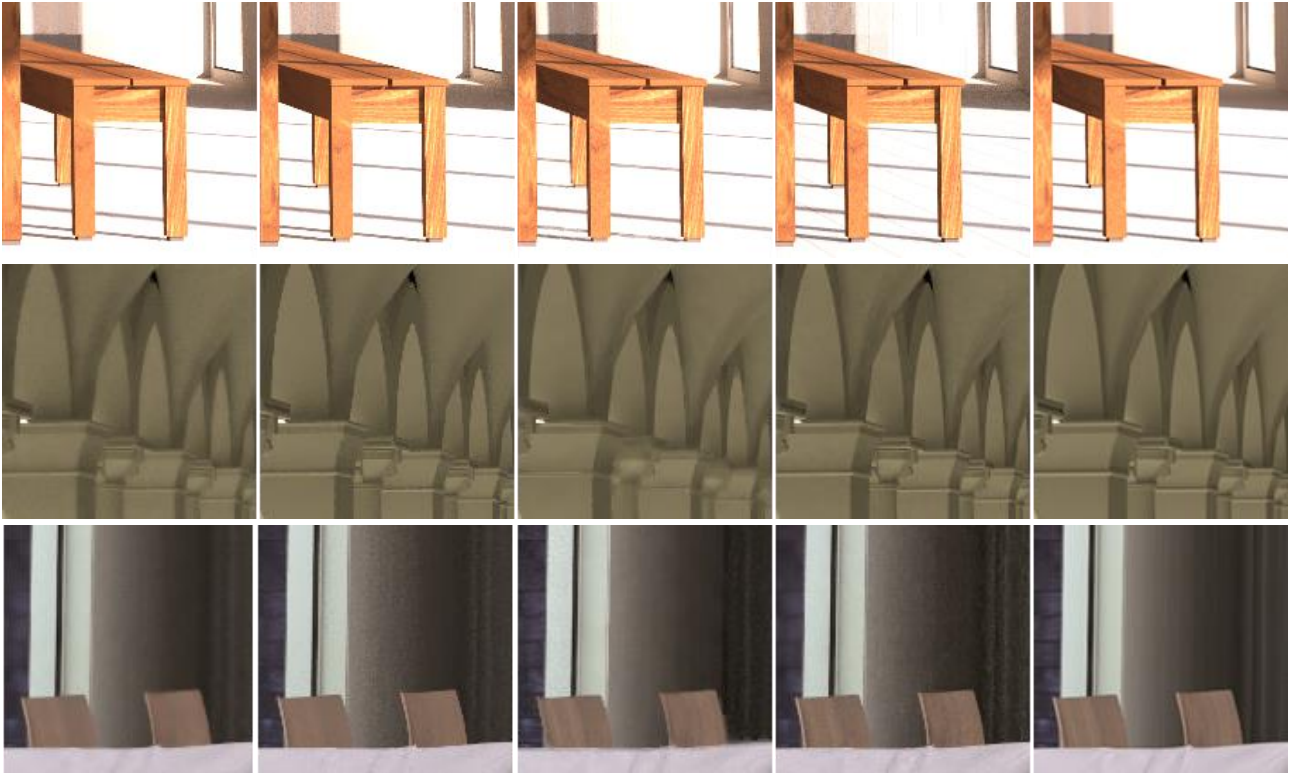


*Рис. 3.* Слева направо: усредненное шумное изображение, фильтрация без смещения камеры, фильтрация со смещением камеры, эталон

При расчетах использовались следующие значения параметров:  $Z_{\max}$ ,  $n_{\max} = 0.1$ , для текстур  $k = 0.002$ , для шумного изображения  $k = 0.1$ ,  $\sigma = (r * 2 + 1) / 4$ , где  $r$  – радиус фильтра.

## **6. Фильтрация по нескольким изображениям при отсутствии сдвига камеры**

Использование изображений со сдвигом вызывает закономерный вопрос: не имеет ли смысл трассировать несколько изображений без сдвига, а затем оценивать ошибку и учитывать ее при расчете весов в фильтрации? В рамках данной работы был протестирован метод, использующий билатеральную фильтрацию усредненных шумных изображений. Имеющаяся информация о разбросе значений шумных пикселей была учтена в весах фильтра. Наименьший вес имели пиксели с максимальной разницей в RGB цвете между входными шумными изображениями (вычисление веса при сравнении цветов описано в разделе 3.4, использован коэффициент  $k = 0.1$ ).



*Рис. 4.* Слева направо: non-local means, Surface blur (Adobe Photoshop), метод [10], фильтрация со смещением камеры, эталон. В отличие от других методов, предложенный метод лишён артефактов ступенчатости

На крупных однородных поверхностях использование смещения не приносит принципиального выигрыша по сравнению с вариантом без смещения. Однако при использовании изображений, полученных со смещением камеры, лучше подавляется эффект ступенчатости (см. рис. 3, два верхних ряда). При этом использование смещения может порождать неточности или артефакты, связанные со скрытием или появлением мелких, узких или далеких объектов (см. рис. 3, нижний ряд).

## 7. Заключение

В результате применения метода изображение визуально становится более гладким, при этом границы объектов и большинства оптических эффектов остаются четкими (сравнение с некоторыми другими методами представлено на рис. 4). Метод оставляет мелкий высокочастотный шум за счет использования финального усреднения шумных изображений, однако подобный артефакт на глаз воспринимается лучше, чем размытие с низкочастотным шумом, заметное при использовании некоторых методов фильтрации.

В методе использовались билатеральная фильтрация на основе шумного изображения и дополнительной информации о сцене, представленной изображениями, и усреднение отфильтрованных изображений. Подобные

алгоритмы предоставляют возможности для быстрой реализации и распараллеливания вычислений.

В качестве критики к исследованному методу можно заметить, что при смещении камеры некоторые детали сцены могут скрываться или, наоборот, попадать на изображение. В результате усреднения по нескольким изображениям вклад таких пикселей остается небольшим, однако если изменения попадают на несколько входных изображений, вклад «лишних» поверхностей суммируется и может повлиять на итоговый результат (см. рис. 3, нижний ряд). Возможное решение – замена последовательности Хаммерсли на квазислучайные числа Соболя с различным начальным состоянием для каждого пиксела (артефакты должны трансформироваться в шум).

Таким образом, разработанный метод фильтрации по нескольким изображениям позволяет улучшить качество по сравнению с аналогами за счёт увеличения количества информации во входных данных. При этом применение смещения камеры позволяет устранить ступенчатость, т.к. даёт возможность работы на суб-пиксельном уровне. Тем не менее, для “ультратонких” деталей (см. рис. 3, решётки и листья в нижних рядах) смещения не позволяют воспроизводить детали с достаточной точностью для получения свободного от артефактов и/или шума изображения.

## Библиографический список

- [1] Zwicker M., Jarosz W., Lehtinen J., Moon B., Ramamoorthi R., Rousselle F., Sen P., Soler C., Yoon S.-E. Recent Advances in Adaptive Sampling and Reconstruction for Monte Carlo Rendering // 2015. Computer Graphics Forum, Wiley. 34 (2). pp.13. URL: <https://cs.dartmouth.edu/~wjarosz/publications/zwicker15star.html>.
- [2] Bauszat P., Eisemann M., Magnor M. Guided image filtering for interactive high-quality global illumination // Computer Graphics Forum. 2011. 30, 4.
- [3] Overbeck R.S., Donner C., Ramamoorthi R. Adaptive wavelet rendering // ACM Trans. Graph. 2009. 28, 5. URL: <https://cseweb.ucsd.edu/~ravir/Overbeck2009AWR.pdf>.
- [4] Rousselle F., Knaus Z., Zwicker M. Adaptive Sampling and Reconstruction using Greedy Error Minimization // ACM Transactions on Graphics (SIGGRAPH Asia). 2011. URL: <https://cs.umd.edu/~zwicker/publications/AdaptiveSamplingGreedyError-SIGA11.pdf>.
- [5] Rousselle F., Knaus C., Zwicker M. Adaptive rendering with non-local means filtering // ACM Trans. Graph. 2012. 31, 6. URL: <http://cseweb.ucsd.edu/~ravir/274/15/papers/a195-rousselle.pdf>.

- [6] Xu R., Pattanaik S. A novel Monte Carlo noise reduction operator // *IEEE Comput. Graph. Appl.* 2005. 25, 2.
- [7] Buades A., Coll B., Morel J. A review of image denoising algorithms, with a new one // *SIAM Journal on Multiscale Modeling and Simulation.* 2005. 4, 2. URL: <https://hal.archives-ouvertes.fr/hal-00271141/document>.
- [8] Sen P., Darabi S. On filtering the noise from the random parameters in Monte Carlo rendering // *ACM Trans. Graph.* 2012. 31, 3.
- [9] Moon B., Jun J. Y., Lee J., Kim K., Hachisuka T., Yoon S.-E. Robust image denoising using a virtual flash image for Monte Carlo ray tracing // *Computer Graphics Forum.* 2013. 32, 1.
- [10] Gruzdev A. M., Frolov V. A., Ignatenko A. V. Practical Approach to the fast Monte-Carlo Ray Tracing // *Programming and Computer Software.* 2015. Vol. 41. No. 5. pp. 253–257.
- [11] Kalantari N. K., Bako S., Sen P. A Machine Learning Approach for Filtering Monte Carlo Noise // *ACM Transactions on Graphics (TOG) (Proceedings of SIGGRAPH 2015).* 2015. Vol. 34. No. 4.
- [12] Chakravarty R. Alla Chaitanya, Kaplanyan A., Schied C, Salvi M., Lefohn A., Nowrouzezahrai D., Aila T. Interactive Reconstruction of Monte Carlo Image Sequences using a Recurrent Denoising Autoencoder // *SIGGRAPH.* 2017. Vol. 36. Iss. 4. No. 98.
- [13] Wong T.-T., Luk W.-S., Heng P.-A. Sampling with Hammersley and Halton Points // *Journal of Graphics Tools.* 1997. vol. 2. no. 2. pp. 9-24.
- [14] Lehtinen J., Karras T., Laine S., Aittala M., Durand F., Aila T. Gradient-Domain Metropolis Light Transport // *ACM Trans. Graph.* 2013. 32, 4.

## Оглавление

1. Введение.....	3
2. Обзор существующих методов.....	3
3. Предпосылки к выбору метода.....	6
4. Предложенный метод .....	6
5. Билатеральная фильтрация .....	8
6. Фильтрация по нескольким изображениям при отсутствии сдвига камеры ..	9
7. Заключение .....	10
Библиографический список.....	11