

# Фильтрация изображений, полученных трассировкой путей методами Монте-Карло

Никерова О.А., ИПМ им. М.В. Келдыша РАН  
olgerem@gmail.com

Фролов В.А., ИПМ им. М.В. Келдыша РАН  
vova@frolov.pp.ru

## Аннотация

Данная работа посвящена фильтрации изображений, полученных Монте-Карло трассировкой путей. Высокая вычислительная сложность данного класса методов не позволяет оперативно получать итоговый результат визуализации. Фильтрация промежуточного результата работы трассировки путей дает возможность ускорить процесс визуализации. В данной работе рассмотрены примеры существующих методов фильтрации результатов трассировки путей, а также подробно рассмотрены некоторые перспективные варианты развития этих методов.

## 1 Введение

Методы трассировки путей Монте-Карло позволяют получать физически корректные изображения 3D сцен, воспроизводя большое количество оптических эффектов. Однако для получения высококачественного результата требуется провести трассировку очень большого количества путей, что приводит к долгому ожиданию итогового результата для пользователя. Если снизить количество трассируемых путей, 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 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 путей.

## 2 Обзор существующих методов

Один из распространенных подходов для удаления шума на изображении – использование фильтров с ядрами разного масштаба для разных пикселей. В применении к фильтрации результатов трассировки подобный подход часто сопровождается адаптивным распределением новых Монте-Карло выборок. На первом шаге вычисляется результат с маленьким количеством выборок на пиксель. Затем алгоритмы итеративно вычисляют, в итоговых значениях каких регионов изображения нет уверенности, и рассчитывают новые выборки для полученных пикселей. Как правило, подобные регионы приходится на границы объектов и оптических эффектов, поскольку значения выборок для подобных пикселей противоречивы. Например, в [Overbeck et al. 2009] предложен алгоритм, использующий вейвлет-преобразования для определения регионов с высокой дисперсией. На этапе адаптивного распределения выборок регионы выбираются так, чтобы минимизировать оценку дисперсии для всех масштабов фильтров. Затем, на этапе реконструкции, оставшаяся ошибка вычитается из коэффициентов вейвлет-преобразований для получения сглаженного изображения.

Похожая идея используется и в статье [Rousselle et al. 2011]. В качестве фильтрации используется фильтр Гаусса, а для адаптивного распределения выборок используется дисперсия пикселей, рассчитанная по соответствующим пикселям выборкам. Поскольку оба метода значительно опираются на рассчитанную дисперсию значений пикселей, 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 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 [Xu and Pattanaik 2005] 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 [Buades et al. 2005], для определения веса сравниваются не значения самих пикселей, а значения пикселей в их заданной окрестности.

В [Rousselle et al. 2012] используется совмещение подходов с адаптивным распределением выборок и использованием фильтра non-local means. Для фильтрации предлагается использовать 3 шага: адаптивная выборка, фильтрация, оценка остаточной ошибки. В каждой последующей итерации полученная оценка используется для адаптивного распределения выборок. Интересен подход, использованный авторами для оценки ошибки: на каждой итерации 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 [Bauszat et al. 2011] предложен метод, использующий дополнительную информацию о геометрии сцены. В данном подходе для каждого пикселя сохраняется информация о нормали к поверхности и глубине (расстоянии от камеры до объекта) для луча, проходящего через центр каждого пикселя. Это упрощение позволяет ускорить расчеты, однако также негативно влияет на качество итоговой визуализации границ объектов и мелких объектов сцены. Расчеты проводятся отдельно для прямого и непрямого освещений. Подход не слишком хорошо ра-

ботает для сложных оптических эффектов, таких как каустики, поскольку ядро сглаживающего фильтра в большой степени зависит от исходной геометрии сцены, не учитывающей многие нюансы результирующего изображения.

В качестве дополнительной информации может использоваться не только информация о сцене, но также информация о способе трассировки. Авторы статьи [Sen and Darabi 2012] отмечают, что зачастую шум на изображении появляется в регионах, где значения полученных освещенностей находятся в зависимости от параметров трассировки. Речь идет о шуме, который появляется в результате расхождения в результатах трассировки для соседних выборок. 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.

Интересный подход предложен в статье [Moon et al. 2013]. Авторы предлагают использовать 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 больше информации, чем упомянутые ранее данные о нормалях и глубинах, поскольку также сохраняет информацию об отражательных и преломляющих способностях объектов. 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 некоторые оптические эффекты (например, каустики), в данной работе сглаживание производится только внутри однородных пикселей. Однородными считаются те пиксели, у которых схожи средние значения Монте-Карло выборок.

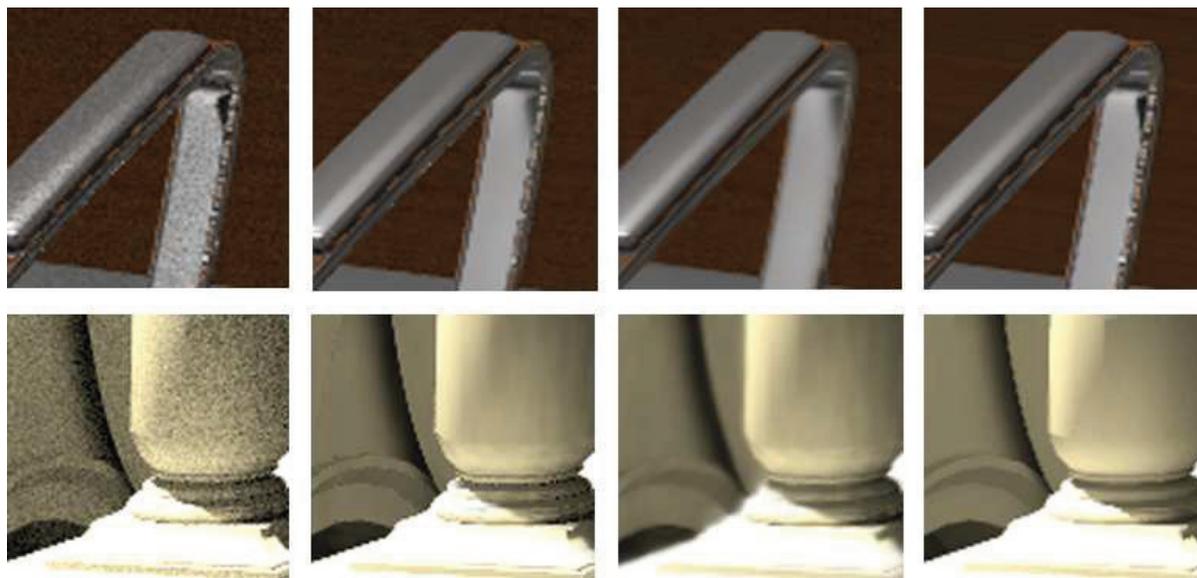


Рис. 1. Слева направо: исходное шумное изображение, результат применения билатерального фильтра (радиус 7), результат применения фильтра non-local means (радиус 7, радиус окрестности 2) и эталон.

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.

### 3 Описание исследованных методов

В рамках исследования методов фильтрации было решено совместить фильтрацию со сложным ядром и использование дополнительной информации. В качестве фильтров были выбраны билатеральный и non-local means. В качестве дополнительной информации были использованы карта глубин, карта нормалей и карта текстур объектов сцены. В подходе не были использованы значения отдельных Монте-Карло выборок, только итоговые значения пикселей изображения. При попадании нескольких поверхностей на один пиксель использовались средние значения для нормалей, глубин и текстур.

При вычислении весов пикселей для билатеральной и non-local means фильтраций 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

[Sen and Darabi 2012]. Расчет веса расстояния между пикселями:

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

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:

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

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.

При расчетах использовались следующие значения параметров:  $z_{max}$ ,  $n_{max} = 0.1$ , для текстур  $k = 0.002$ , для шумного изображения  $k = 1.0$ ,  $\sigma = (r * 2 + 1) / 4$ , где  $r$  – радиус фильтра.

## 4 Результаты

В результате применения билатеральной фильтрации с дополнительной информацией изображение визуально становится более гладким, при этом границы объектов остаются четкими. Однако даже для простых сцен сохраняются типичные артефакты - выбросы на границах геометрических объектов и размытие оптических эффектов (см. Рис. 1).

Значения пикселей на границах геометрических объектов вычисляются из значений противоречащих друг другу выборок, соответствующих разным поверхностям. При малом количестве выборок на пиксель этот факт приводит к неизбежным артефактам, поскольку возможен сильный перевес вклада одной из поверхностей, зависящий от случайного распределения направлений первичных лучей при трассировке. Также 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 усредняются.

Для световых же эффектов, таких как тени, каустики и т.п., использования данного подхода напрямую недостаточно, поскольку геометрия и текстура объектов не предоставляют достаточно информации для хорошей фильтрации: приходится выбирать между сглаживанием и сохранением четкости изображения.

Использование более продвинутого фильтра non-local means не вполне устраняет описанные недостатки. Выбросы на границах геометрических объектов сглаживаются, однако страдает четкость итогового изображения. Заметно повышение размытия и оптических эффектов.

## 5 Выводы и дальнейшие планы

Фильтрация только на основе шумного изображения, без использования адаптивного распределения дополнительных выборок и/или полной информации о выборках и других данных оставляет артефакты на границах объектов. В дальнейшем предполагается включить в анализ субпиксельную информацию – данные о выборках и задействованных поверхностях сцены.

Использование информации о геометрии сцены помогает сохранить четкими границы объектов. В качестве альтернативы или до-

полнения изображениям с нормальми, глубиной и текстурой объектов предполагается рассмотреть изображение с виртуальной вспышкой, как в подходе [Moon et al. 2013] и сравнить результаты.

Артефакты со сложными световыми эффектами на изображении невозможно устранить лишь с помощью сохранения дополнительной геометрической информации, единственным простым и надежным источником информации о них остается шумное изображение, полученное в результате трассировки. Для сохранения четких границ у оптических эффектов понадобится применять сегментацию по цветам итоговых пикселей или по распределению выборок, как это было сделано в статье [Moon et al. 2013] в рамках вычисления гомогенных пикселей.

## Список литературы

- [Bauszat et al. 2011] Bauszat P., Eisemann M., Magnor M. 2011 *Guided image filtering for interactive high-quality global illumination*, Computer Graphics Forum 30, 4.
- [Buades et al. 2005] Buades, A., Coll, B., Morel, J. 2005. *A review of image denoising algorithms, with a new one*. SIAM Journal on Multiscale Modeling and Simulation 4, 2.
- [Moon et al. 2013] Moon B., Jun J. Y., Lee J., Kim K., Hachisuka T., Yoon S.-E. 2013 *Robust image denoising using a virtual flash image for Monte Carlo ray tracing*. Computer Graphics Forum 32, 1.
- [Overbeck et al. 2009] Overbeck R.S., Donner C., Ramamoorthi R. 2009 *Adaptive wavelet rendering*, ACM Trans. Graph. 28, 5.
- [Rousselle et al. 2011] Rousselle F., Knaus Z., Zwicker M. 2011 *Adaptive Sampling and Reconstruction using Greedy Error Minimization*, ACM Transactions on Graphics (SIGGRAPH Asia).
- [Rousselle et al. 2012] Rousselle F., Knaus C., Zwicker M. 2012 *Adaptive rendering with non-local means filtering*, ACM Trans. Graph. 31, 6.
- [Sen and Darabi 2012] Sen P., Darabi S. 2012 *On filtering the noise from the random parameters in Monte Carlo rendering*. ACM Trans. Graph. 31, 3.
- [Xu and Pattanaik 2005] Xu R., Pattanaik S. 2005 *A novel Monte Carlo noise reduction operator*. IEEE Comput. Graph. Appl. 25, 2.