

Кодирование глобального освещения гауссианами в приложениях реального времени

К. С. Зубатов, А. С. Щербаков, В. А. Фролов, В. В. Санжаров

Московский государственный университет им. М. В. Ломоносова, Москва, Россия

Аннотация. В графике реального времени необходимо рассчитывать глобальное (вторичное) освещение – компоненту освещенности, возникающую в результате многократного переотражения света от поверхностей сцены. Вторичное освещение напрямую зависит от материалов сцены, однако его точность в графике реального времени ограничена производительностью пользовательского устройства и требованиями приложения. В данной работе исследуется применение метода гауссовых сплатов (Gaussian splatting) для хранения вторичного освещения сцены. Этот метод реконструирует материалы сцены и позволяет изменять сложность представления сцены в зависимости от требуемой точности. Предложенный метод использует этап предобработки, в ходе которого по набору изображений с диффузным вторичным освещением сцены создается множество гауссиан, аппроксимирующих световую информацию. В результате был получен метод, имеющий высокую точность при сравнении с эталонным методом трассировки путей и позволяющий регулировать точность для изменения скорости рендеринга.

Ключевые слова: глобальное освещение, вторичное освещение, гауссианы, рендеринг

Gaussians encoding of global illumination in real time applications

K. S. Zubatov, A. S. Shcherbakov, V. A. Frolov, V. V. Sanzharov

Lomonosov Moscow State University, Moscow, Russia

Abstract. In real-time graphics, it is necessary to calculate global (secondary) illumination – a component of illumination that occurs as a result of multiple reflections of light from scene surfaces. Secondary illumination directly depends on scene materials, but its accuracy in real-time graphics is limited by the performance of the user's device and application requirements. This paper investigates the use of Gaussian splatting to store secondary scene lighting. This method reconstructs scene materials and allows changing the complexity of scene representation depending on the required accuracy. The proposed method uses a preprocessing stage, during which a set of images with diffuse secondary scene lighting is used to create a set of Gaussians that approximate the light information. The result is a method that is highly accurate when compared to the reference path tracing method and allows changing the accuracy to vary the rendering speed.

Keywords: global illumination, secondary illumination, gaussians, rendering

Введение

Освещение играет ключевую роль в ощущении реалистичности сцен. Свет, взаимодействуя с поверхностями объектов, отражается, преломляется, рассеивается или поглощается в зависимости от материалов. Для первичного освещения было создано множество математических моделей, позволяющих достаточно точно описать взаимодействие света с материалом [1]. Однако вторичное (глобальное) освещение, являющееся результатом множественного переотражения света от объектов сцены, представляет собой более сложную задачу. В условиях графики реального времени вычисление глобального освещения особенно затруднено тем, что на большинстве пользовательских устройств не получится решать многомерное интегральное уравнение, описывающее взаимодействие света со сложными материалами, с достаточной скоростью. В связи с этим используются различные аппроксимации, позволяющие получить удовлетворительный результат при имеющихся у пользователя ограниченных вычислительных ресурсах. Это создает необходимость балансировать между точностью освещения и количеством потребляемых ресурсов при выборе метода расчета глобального освещения. В данной работе предлагается метод представления вторичного освещения в виде гауссиан, который позволяет регулировать точность представления материалов, используемых для создания вторичного освещения, с целью получения различной скорости рендеринга.

Обзор существующих решений

Существует множество методов аппроксимации глобального освещения, используемых в графике реального времени, отличающихся по точности, скорости работы, поддержке динамических объектов,

способу хранения информации и способности балансировать между точностью результата и скоростью работы.

Карты освещения (Light maps) [2] используют текстурный атлас для информации об освещенности объектов. Каждому текселью атласа соответствует трехмерная точка объекта, для которой вычисляется вторичное освещение на этапе предобработки. Вычисление глобального освещения во время рендеринга сводится к простой операции выборки из текстуры. Метод широко применяется на практике [3, 4] из-за высокой точности итогового освещения и минимальной вычислительной нагрузки при рендеринге, однако данный метод неприменим для сцен с динамическим освещением. Также при масштабировании требуется полный перерасчет освещения, а при высокой точности – большое количество памяти для текстур.

Сферические гармоники [5] хранят информацию об освещении в виде позиции гармоники и набора коэффициентов ортонормированных сферических функций, которыми кодируется освещение, входящее под определенным направлением в гармонику с заданной позицией. Во время рендеринга глобальное освещение объекта вычисляется с помощью интерполяции освещения ближайших гармоник. Метод позволяет компактно хранить информацию о диффузном освещении и поддерживает динамические объекты, однако требует большого количества памяти для хранения спекулярного освещения, не поддерживает динамические источники света, а также имеет низкую точность в закрытых сценах.

Метод излучательности (Radiosity) [6] для каждой пары полигонов сцены вычисляет (на этапе предобработки) долю света (форм-фактор), которая попадает с одного полигона на другой, формируя матрицу форм-факторов $N \times N$, где N – число полигонов в сцене (ввиду геометрической сложности большинства сцен обычно используют упрощенное представление сцены). Во время рендеринга вычисление вторичного освещения полигонов сводится к умножению матрицы форм-факторов на вектор их первичной освещенности. Метод является физически корректным и вычислительно простым, отчего нашел широкое применение в графике реального времени [7], однако имеет квадратичную сложность от числа полигонов N по количеству арифметических операций и по памяти.

Кэш освещенности (Irradiance Cache) [8] хранит информацию об освещении в виде трехмерной сетки, элементы которой содержат сферические гармоники. Также каждой сферической гармонике соответствует октаэдральная текстура глубины, определяющая зону действия её излучения. Вычисление освещения заключается в интерполяции излучения от ближайших гармоник для заданной точки сцены.

Метод трассировки вокселей конусами (Voxel Cone Tracing) [9] использует упрощенное представление сцены в виде разреженной воксельной сетки. В каждом кадре источники освещения обновляют информацию о входящем излучении для видимых вокселей, а в конце кадра вычисляется итоговое вторичное освещение при помощи трассировки конусами (т.е. упрощенной трассировки лучей) по воксельной сетке. Метод позволяет вычислять множественные переотражения света для динамических сцен, но из-за высокой вычислительной нагрузки и нагрузки на память обычно ограничиваются вычислением одного переотражения. Также для алгоритма характерны «протечки» света сквозь тонкие объекты сцены (особенно при малом разбиении сцены на воксели).

Метод объемов распространения света (Light Propagation Volumes) [10] вычисляет глобальное освещение сцены с помощью численного решения дифференциального уравнения на трехмерной сетке. В сетке хранятся сферические гармоники, кодирующие входящее со всех направлений излучение. Начальное приближение задается виртуальными источниками света [11]. Для расчета затенения при вычислении светопереноса используется дополнительная трехмерная сетка с приближенной вокселизацией поверхностей сцены. Данный метод применим для полностью динамических сцен, так как вычисляет распространение света по сетке каждого кадра, однако требует большого количества памяти для хранения регулярных трехмерных сеток (для решения этой проблемы разработана модификация с вложенными сетками [12]), имеет низкую точность освещения и подвержен «протечкам» света из-за недостаточной дискретизации сцены.

У всех перечисленных методов возникают проблемы с масштабированием, когда в зависимости от имеющихся вычислительных ресурсов необходимо либо уменьшить точность освещения для экономии

ресурсов, либо получить более точное освещение в сложных участках сцены при наличии свободных ресурсов.

- Методы, использующие трехмерные сетки (трассировка вокселей конусами, объемы распространения света, кэш освещенности), при изменении размера вокселей либо теряют информацию в областях со сложным освещением, либо перерасходуют память для хранения освещения в однородных областях сцены.
- Для карт освещения проблема аналогична методам с трехмерными сетками – с ростом размера текселя снижается точность освещения, а при малых размерах текселя множество текселей содержит одинаковую информацию для участков сцены с однородным освещением. На практике проблема частично решается применением блочного сжатия текстур. Сжатие изображений (например, в формате png) не представляется возможным из-за длительного декодирования на gpu.
- Метод излучательности имеет квадратичную сложность по памяти и вычислениям от числа полигонов сцены, что затрудняет его использование для сцен с высокой детализацией. Использование сильно упрощенной сцены для расчета форм-факторов приведет к потере точности или некорректному переотражению света.
- Изменение числа сферических гармоник требует их ручной расстановки или использования эвристических алгоритмов. Также потребуется дополнительная информация, ограничивающая зону влияния гармоник, для предотвращения просвечивания света сквозь геометрию сцены.

Описание представления сцены гауссианами

Для хранения глобального освещения предложенный метод использует представление сцены в виде гауссиан (Gaussian splatting) [13]. Гауссиана g описывается пятеркой $g = (\mu, s, r, h, o)$:

- $\mu \in \mathbb{R}^3$ – координаты центра гауссианы в мировом пространстве;
 - $s \in \mathbb{R}_+^3$ – коэффициенты масштаба гауссианы по осям x, y, z ;
 - $r = (r_0, r_i, r_j, r_k)^T \in \mathbb{R}^4$ – коэффициенты кватерниона, описывающего поворот гауссианы в мировом пространстве;
 - $h \in \mathbb{R}^{(n+1)^2 \times 3}$ – коэффициенты сферических гармоник порядка n ;
 - $o \in [0, 1]$ – коэффициент прозрачности гауссианы,
- и функцией $G(x)$ вида:

$$G(x) = \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right).$$

Матрица ковариации Σ в функции (1) задается через матрицы поворота R и масштаба S :

$$\Sigma = RSS^T R^T,$$

$$R = \begin{pmatrix} 1 - 2(r_j^2 + r_k^2) & 2(r_i r_j - r_k r_0) & 2(r_i r_k + r_j r_0) \\ 2(r_i r_j + r_k r_0) & 1 - 2(r_i^2 + r_k^2) & 2(r_j r_k - r_i r_0) \\ 2(r_i r_k - r_j r_0) & 2(r_j r_k + r_i r_0) & 1 - 2(r_i^2 + r_j^2) \end{pmatrix},$$

$$S = Es,$$

где E обозначает единичную матрицу.

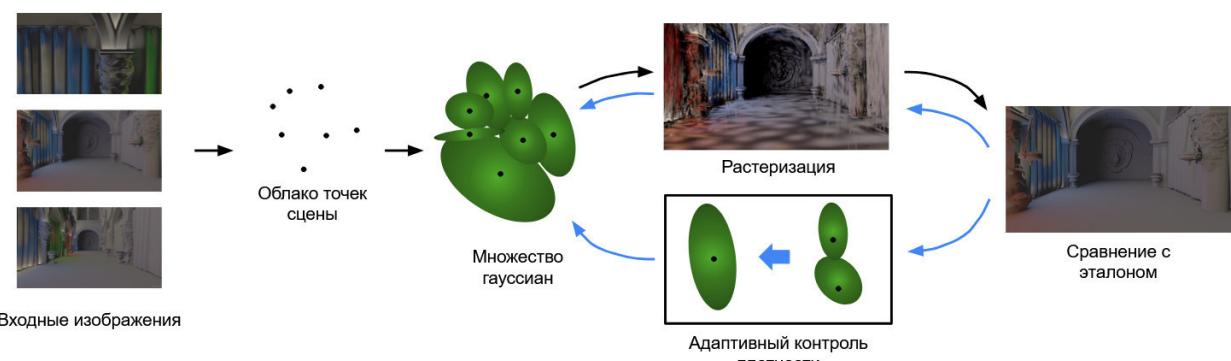


Рисунок 1. Схема построения гауссиан по изображениям сцены

Построение множества гауссиан (рис. 1) для трехмерной сцены по набору изображений заключается:

- в создании облака точек X по входным изображениям;
- задании начального множества гауссиан. Центрами гауссиан считаются элементы облака точек $x \in X$, а начальные размеры каждой гауссианы определяются на основе усредненного расстояния до трех ближайших точек из облака X ;
- оптимизации параметров гауссиан (рис. 2) с помощью итеративного процесса растеризации гауссиан и сравнения с входными изображениями. Для оптимизации размера, позиции, цвета и прозрачности используется стохастический градиентный спуск;
- адаптивном контроле плотности, регулирующем количество гауссиан через разбиение больших или слияние слишком малых гауссиан (выполняется один раз в несколько итераций).

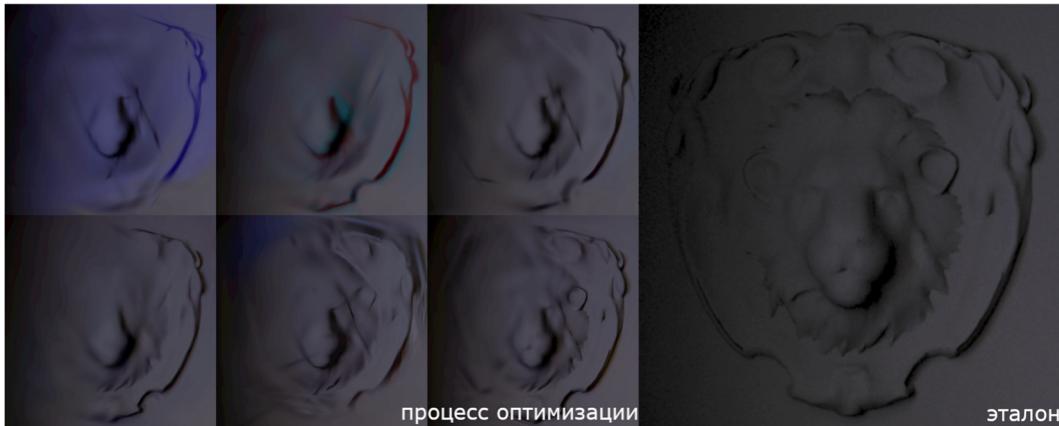


Рисунок 2. Процесс оптимизации сцены. Представление в виде гауссиан сравнивается с эталоном

Для растеризации полученного множества необходимо:

1) выполнить предобработку гауссиан с использованием матриц перехода в пространство камеры и перспективной проекции. В результате данного этапа будут отсечены гауссианы (рис. 3), не попавшие в усеченную пирамиду видимости, а для видимых гауссиан будут вычислены цвет C_i , глубина d_i , двумерная матрица ковариации Σ_i и прозрачность α_i ;

2) отсортировать видимые гауссианы по глубине;

3) растеризовать видимые гауссианы в порядке возрастания глубины. Для этого в каждом пикселе задается переменная $C := (0, 0, 0)$, которая будет хранить итоговый цвет, и переменная $T := 1$, описывающая вклад текущей гауссианы в итоговый цвет. Гауссианы, двумерные проекции которых пересекают данный пиксель, обрабатываются в порядке возрастания глубины d_i . Для проекции гауссианы с центром в точке экрана μ_i и пикселя с координатами p вычисляется прозрачность $\alpha_0 = \alpha_i \exp((\mu_i - p)\Sigma_i(\mu_i - p)^T)$. Если прозрачность в пикселе больше $1/255$, то обновляются итоговый цвет $C := C + C_i T \alpha_0$ и вклад следующей гауссианы $T := T(1 - \alpha_0)$. Если новое значение T близко к нулю, то обработка гауссиан прекращается.

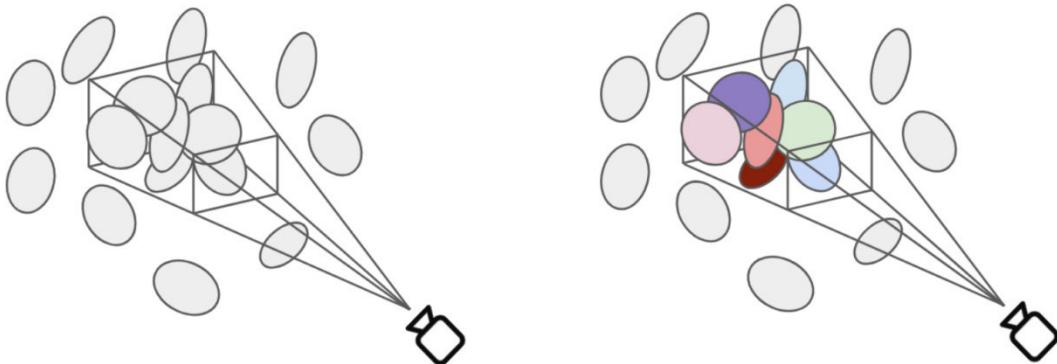


Рисунок 3. Отсечение невидимых гауссиан (серые). Далее будут обработаны и отрисованы только видимые (цветные) гауссианы

Предложенный метод

1. Предобработка сцены

Предложенный метод использует этап предобработки для создания множества гауссиан по набору изображений сцены. Однако, в отличие от оригинального метода, который в основном используется для реконструкции сцен по наборам фотографий или видеозаписям, в данной работе будет реконструироваться только информация о вторичном освещении. Вычисление вторичного освещения, как и в картах освещения [2], основывается на использовании стохастической трассировки путей.

Полное освещение сцены описывается интегральным уравнением [14]:

$$L_o(x, \omega) = L_e(x, \omega) + \int_{\Omega} f_r(x, \omega, \omega') L_i(x, \omega') (-\omega' \cdot n) d\omega', \quad (1)$$

где $L_e(x, \omega)$ – количество собственного излучения, исходящего из точки x вдоль направления ω ;

$f_r(x, \omega, \omega')$ – двулучевая функция отражательной способности;

$L_i(x, \omega')$ – количество излучения, входящего в точку x вдоль направления ω' ;

Ω – единичная полусфера для точки x и нормали n ;

точка x и направление ω определяются камерой.

Данный интеграл описывает полное освещение $L_o(x, \omega)$ сцены для точки x и направления ω . Получение вторичного освещения $L_s(x, \omega)$ обеспечивается игнорированием вклада от лучей, отразившихся один раз, и собственного излучения $L_e(x, \omega)$, исходящего из точки x вдоль направления ω , при вычислении интеграла (2). Для описания влияния луча, имеющего направление ω и попавшего в точку x , на вторичное освещение введем функцию $\mathbb{1}(x, \omega)$:

$$\mathbb{1}(x, \omega) = \begin{cases} 1, & \text{луч не попал в камеру после первого отражения} \\ 0, & \text{иначе} \end{cases}$$

Тогда $L_s(x, \omega)$ имеет вид

$$L_s(x, \omega) = \int_{\Omega} \mathbb{1}(x, \omega) f_r(x, \omega, \omega') L_i(x, \omega') (-\omega' \cdot n) d\omega'. \quad (2)$$

Для вторичного освещения рассматриваются ламбертовские материалы с $f_r(x, \omega, \omega') = \frac{C}{\pi}$, где C обозначает цвет объекта в точке x . Тогда уравнение (2) можно переписать в следующем виде:

$$L_s(x, \omega) = C \int_{\Omega} \frac{1}{\pi} \mathbb{1}(x, \omega) L_i(x, \omega') (-\omega' \cdot n) d\omega'. \quad (3)$$

Таким образом, при вычислении вторичного диффузного освещения можно вынести цвет C точки x , получив тем самым более низкочастотное изображение с входящим в точку вторичным освещением (рис. 4).



Рисунок 4. Сравнение вторичного исходящего и входящего освещения

Данный подход призван снизить количество гауссиан, необходимых для хранения глобального освещения. Его использование не имеет визуального влияния на конечный результат.

2. Масштабирование

При оптимизации параметров гауссиан на этапе предобработки для функции потерь используется следующая комбинация метрик \mathcal{L}_1 (манхэттенское расстояние) и \mathcal{L}_{D-SSIM} :

$$\mathcal{L} = 0.2\mathcal{L}_1 + 0.8\mathcal{L}_{D-SSIM}.$$

$\mathcal{L}_{D-SSIM}(x, y) = \frac{1-SSIM(x, y)}{2}$ выражает структурные отличия двух изображений через метрику SSIM.

Ограничиваая минимальное значение функции потерь, можно достичь разного количества гауссиан, необходимых для приближенного представления вторичного освещения. Таким образом, предложенный метод позволяет плавно выбирать между необходимой точностью освещения и скоростью визуализации, которая зависит от числа гауссиан.

3. Рендеринг сцены

На этапе рендеринга для получения изображения с полным освещением сцены необходимо:

1) вычислить первичное освещение сцены:

$$L_p(x, \omega) = L_e(x, \omega) + \int_{\Omega} (1 - \mathbb{1}(x, \omega)) f_r(x, \omega, \omega') L_i(x, \omega') (-\omega' \cdot n) d\omega';$$

2) растеризовать гауссианы для получения вторичного освещения $L_s(x, \omega)$ вида (3);

3) совместить первичное и вторичное освещение для получения полного освещения $L_o(x, \omega)$:

$$L_o(x, \omega) = L_p(x, \omega) + L_s(x, \omega) = L_e(x, \omega) + \int_{\Omega} f_r(x, \omega, \omega') L_i(x, \omega') (-\omega' \cdot n) d\omega'.$$

Первичное освещение вычисляется с помощью техники отложенного освещения, в результате которой сохраняется текстура цвета объектов сцены, которая будет использована для третьего пункта. Он заключается в композиции первичного освещения и вторичного входящего освещения, умноженного (используется произведение Адамара) на текстуру с цветом объектов сцены (рис. 5).

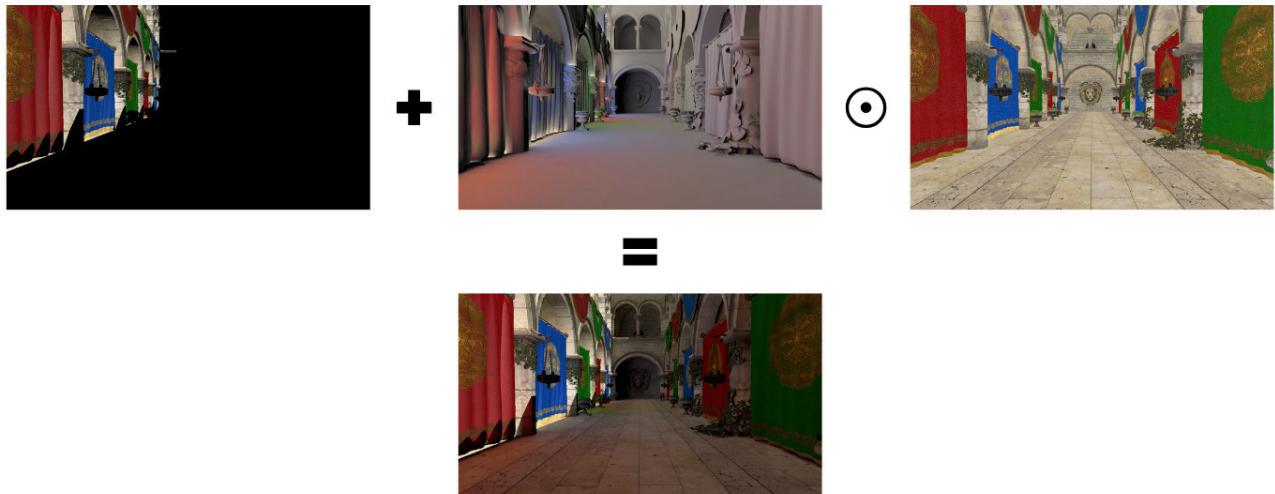


Рисунок 5. Композиция первичного освещения, вторичного освещения и буфера цвета

Результаты экспериментов

1. Сравнение с существующими решениями

Из существующих решений был выбран метод карт освещения, так как он имеет наивысшую точность среди методов реального времени. Сравнение включало анализ потребляемой памяти и времени рендеринга (рис. 6). Тестирование предложенного метода проводилось на сцене *Crytek Sponza* в разрешении 1920×1080 , использовалась видеокарта *Nvidia RTX 3060*.

Результаты показали, что предложенный метод обладает высокой точностью и потребляет меньше памяти, чем метод карт освещения. Однако растеризация гауссиан в предложенном методе требует больше вычислительных ресурсов, чем выборка из текстурного атласа в картах освещения. Для оценки точности метода было проведено сравнение с трассировкой путей, являющейся эталонным методом, по метрике SSIM (рис. 7). Метод продемонстрировал высокое сходство с эталонными изображениями.



Рисунок 6. Сравнение предложенного метода и карт освещения по времени и памяти

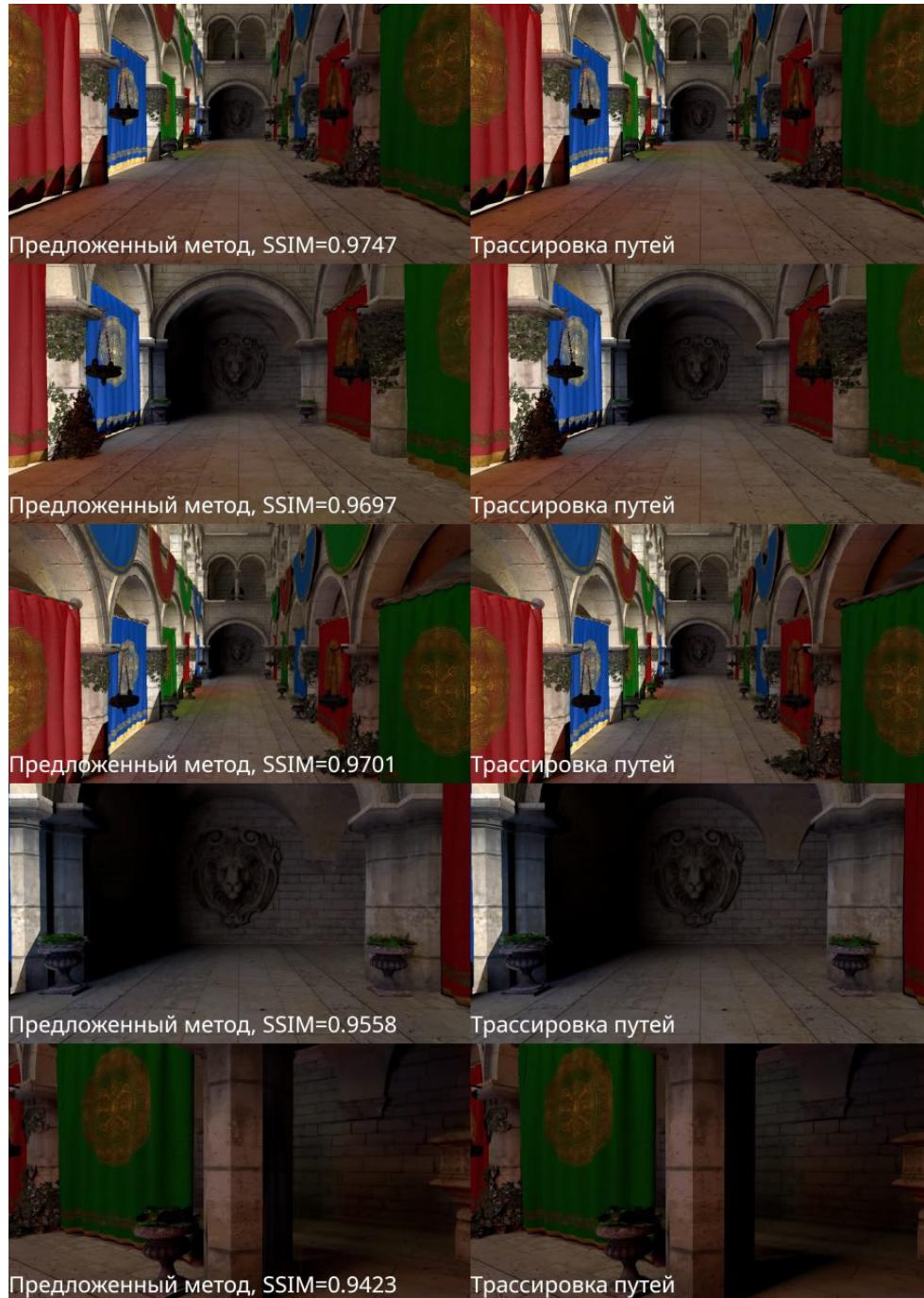


Рисунок 7. Сравнение предложенного метода и трассировки путей (эталона) по метрике SSIM

2. Исследование масштабируемости метода

Также было исследовано влияние количества используемых гауссиан на производительность и точность. Эксперименты показали (рис. 8), что возможно достичь существенного ускорения рендеринга при умеренном снижении качества визуализации, что демонстрирует масштабируемость метода под конкретные требования к скорости и качеству визуализации.

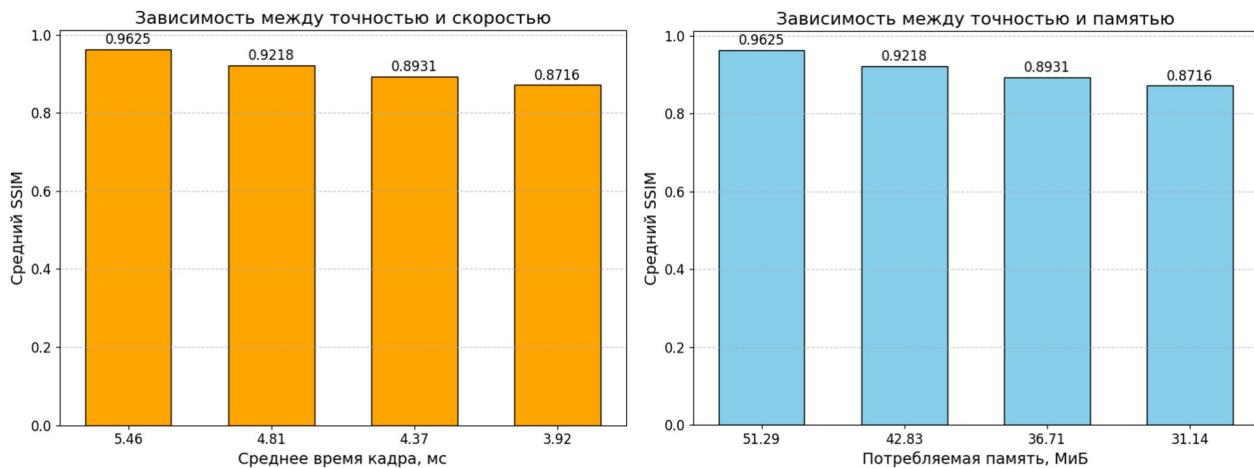


Рисунок 8. Сравнение точности и времени кадра предложенного метода при разном числе гауссиан

Выводы

В работе представлен метод хранения вторичного освещения на основе гауссиан. Подход позволяет реконструировать материалы сцены и управлять детализацией освещения, тем самым обеспечивая баланс между точностью визуализации и производительностью. Результаты исследования показали, что метод способен значительно снизить объем используемой памяти по сравнению с традиционными картами освещения при сохранении высокой точности, сопоставимой с трассировкой путей. Также показана масштабируемость метода: настройка порогового значения функции потерь позволяет балансировать между скоростью визуализации и качеством.

Список литературы

1. Pharr M., Jakob W., Physically based rendering: From theory to implementation. MIT Press, 2023.
2. Quake's lighting model: Surface caching / M. Abrash, et al. // Graphic programming black book. 2000. Vol. 2.
3. Unity Technologies. Unity User Manual, Lightmapping. URL: <https://docs.unity3d.com/2019.1/Documentation/Manual/Lightmappers.html>.
4. Epic Games. Understanding Lightmapping in Unreal Engine. URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/understanding-lightmapping-in-unreal-engine>.
5. Sloan P. P., Kautz J., Snyder J. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments // Seminal Graphics Papers: Pushing the Boundaries. 2023. Vol. 2. P. 339-348.
6. Cohen M. F., Greenberg D. P. The hemi-cube: A radiosity solution for complex environments // ACM Siggraph Computer Graphics. 1985. Vol. 19, no. 3. P. 31-40.
7. Martin S. Enlighten real-time radiosity // ACM SIGGRAPH 2011. Computer Animation Festival. 2011. P. 97-97.
8. McGuire M., et al. Real-time global illumination using precomputed light field probes // Proceedings of the 21st ACM SIGGRAPH symposium on interactive 3D graphics and games. 2017. P. 1-11.
9. Interactive indirect illumination using voxel-based cone tracing: An insight / C. Crassin, et al. // ACM SIGGRAPH 2011 Talks. 2011. P. 1-1.
10. Kaplanyan A. Light propagation volumes in cryengine 3 // ACM SIGGRAPH Courses. 2009. Vol. 7, no. 2.
11. Keller A. Instant radiosity // Proceedings of the 24th annual conference on Computer graphics and interactive techniques. 1997. P. 49-56.
12. Kaplanyan A., Dachsbacher C. Cascaded light propagation volumes for real-time indirect illumination // Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games. 2010. P. 99-107.
13. 3D Gaussian splatting for real-time radiance field rendering / B. Kerbl, et al. // ACM Trans. Graph. 2023. Vol. 42, no. 4. P. 139:1-139:14.
14. Kajiya J. T. The rendering equation // Proceedings of the 13th annual conference on Computer graphics and interactive techniques. 1986. P. 143-150.