УДК 378:004
DOI: 10.25686/978-5-8158-2474-4-2025-83-93

Кроссплатформенный дифференцируемый рендеринг функций расстояний

А. С. Будак[1], А. Р. Гарифуллин[2], В. А. Галактионов[2], В. А. Фролов[1]
[1]Институт перспективных исследований проблем искусственного интеллекта
и интеллектуальных систем МГУ им. М.В. Ломоносова, Москва, Россия
[2]Институт прикладной математики им. М.В. Келдыша РАН, Москва, Россия

Аннотация. В данной работе предлагаются пути улучшения предложенного в 2024 году метода дифференцируемого рендеринга функций расстояния со знаком, а также его кроссплатформенной реализации. Кроссплатформенность даёт возможность использовать широкий спектр графических ускорителей, обеспечивая применимость в различных аппаратных конфигурациях. Для базового метода предлагаются две модификации. Во-первых, используемый метод поиска пересечения луча и функции расстояния, sphere tracing, заменяется адаптированными под дифференцируемый рендеринг альтернативными подходами: методом Ньютона и аналитическим методом. Во-вторых, этап расчёта производных по текстурным и геометрическим параметрам сцены разбивается на два: для внутреннего и граничного интегралов. Это позволяет снизить число выборок метода Монте-Карло для расчёта градиентов по текстуре и разделить вычисление на два шейдера. В результате получен метод втрое быстрее базового, но сохраняющий его точность.

***Ключевые слова*:** компьютерная графика, дифференцируемый рендеринг, трассировка лучей, кроссплатформенность

Cross-platform differentiable signed distance function rendering

A. S. Budak[1], A. R. Garifullin[2], V. A. Galaktionov[2], V. A. Frolov[1]
[1]IAI Moscow State University, Moscow, Russia
[2]Keldysh Institute of Applied Mathematics, Moscow, Russia

Abstract. This work presents improvements to the differentiable signed distance function rendering method proposed in 2024, as well as its cross-platform implementation. Cross-platform nature makes possible the use of a wide range of GPUs, ensuring applicability in various hardware configurations. Two modifications are proposed for the base method. First, the method used to find the intersection of a ray and a distance function, sphere tracing, is replaced by alternative approaches adapted for differentiable rendering: Newton's method and an analytical method. Secondly, the gradient calculation stage with respect to both texture and geometric parameters of the scene is split into two, for internal and boundary integrals. This allows us to reduce the number of Monte Carlo samples for internal gradient estimation and perform calculations in two separate shaders. This results in a method three times faster than the base approach, while retaining its accuracy.

***Keywords*:** computer graphics, differentiable rendering, ray tracing, cross-platform

Введение

Обратный рендеринг – задача восстановления параметров сцены по одному или нескольким изображениям – является важным направлением в компьютерном зрении и графике. В последнее десятилетие, на волне активного внедрения методов машинного обучения, данное направление стремительно развивается в виде дифференцируемого рендеринга, применяющего градиентные методы оптимизации для решения задачи обратного рендеринга. Функции расстояния со знаком (signed distance function, SDF) – вид неявного представления поверхности объекта. Эта функция для точки пространства возвращает расстояние от данной точки до поверхности.

Код подавляющего большинства открытых методов дифференцируемого рендеринга накладывает ограничения на поддерживаемое аппаратное обеспечение – вычисления возможно проводить либо на центральном процессоре, что значительно замедляет процесс оптимизации, либо с использованием технологии CUDA, требующей для своей работы графические карты NVidia. При этом задача дифференцируемого рендеринга с учётом геометрических параметров в сотни раз вычислительно затратнее, чем прямой рендеринг, так как для оптимизации требуется провести большое число итераций, и на каждой, помимо отрисовки, нужно обновить параметры сцены. В связи с этим необходимо, чтобы программное решение не было привязано к определённому оборудованию, а производительность оставалась одним из первых приоритетов, наряду с точностью реконструкции.

Обзор существующих работ

1. Дифференцируемый рендеринг

Все подходы, появившиеся с момента выхода первой статьи по дифференцируемому рендерингу [1], можно объединить в две группы, в зависимости от типа представления, используемого при оптимизации: две основные группы направлений представляют сцену объёмами или поверхностями. К группе объёмных представлений относятся такие методы, как нейронные поля освещённости (NeRF) [2], а также метод Gaussian splatting [3]. Методы, использующие поверхностные представления, характеризуются двумя основными направлениями. Первый – нейронный рендеринг функции расстояния [4, 5, 6], значения которой хранятся в виде векторов признаков для нейросети. Ранние методы [7, 4] использовали одну нейронную сеть для представления всей сцены, что сказывалось на скорости оптимизации. Эта проблема решалась разными способами: разбиением одной сети на несколько, каждая из которых представляла часть сцены [5, 8]; также модифицировался сам алгоритм рендеринга для существующих методов [9]. Второе направление – не-нейронные, аналитические методы, оптимизируют объекты, заданные мешем или функцией расстояния со знаком. Нас интересуют поверхностные методы, в частности аналитические, чтобы восстановленный объект можно было напрямую использовать в физически обоснованном рендеринге.

2. Методы аналитического рендеринга поверхностей

В 2018 году появился метод edge sampling [10], впервые позволивший получить аналитические производные по параметрам геометрии, в частности меша, в физически обоснованном рендеринге. В этих условиях цвет пикселя представляет собой интеграл. При дифференцировании по параметрам, задающим форму объекта, подынтегральное выражение терпит разрывы, положение которых зависит от этих параметров. Можно совершить переход к интегралу по области непрерывности подынтегральной функции – тогда от параметра будет зависеть область интегрирования. Для оценки производной этого интеграла необходимо учесть интеграл по разрывам, представляющим собой внешние и внутренние границы объекта. Этот интеграл называется граничным. Edge sampling требовал явного семплирования граничного интеграла, что для меша является нетривиальной задачей. Из-за этого отрисовка одного изображения занимала на порядок больше времени по сравнению с обычным рендерингом. При этом сам процесс оптимизации требует проведения сотен итераций, на каждой из которых нужно отрисовать одно или несколько изображений сцены с разных ракурсов. На практике это приводит к ощутимым затратам времени.

В 2019 и 2020 годах появились два связанных между собой метода, которые избавлялись от необходимости явного семплирования границ объекта при оценке производных. Первый подход, репараметризация [11], предлагал замену переменных, которая локально учитывала разрывы, что позволяло уйти от оценки граничного интеграла вовсе. Однако используемая оценка вносила дополнительные смещение и разброс. Метод 2020 года, warped area sampling (WAS) [12], строго показал, что граничный интеграл в формуле производной возникает согласно транспортной теореме Рейнольдса, и предложил продлить область определения граничного интеграла на пространство с помощью теоремы Остроградского-Гаусса. Для этого перехода строится специальное векторное поле. Этот метод даёт несмещённую оценку, и авторами было показано, как из построенного векторного поля можно получить репараметризацию, не вносящую смещение.

Три подхода к дифференцируемому рендерингу, описанные выше, использовали меши в практической реализации. Однако меши, с точки зрения задачи реконструкции поверхности, имеют ряд фундаментальных проблем, которые делают их использование нежелательным. Их недостатки хорошо описаны и проиллюстрированы в работе 2021 года [13]. Фундаментальной проблемой данного представления является невозможность менять топологию объекта (род поверхности), также во время оптимизации могут возникать необратимые самопересечения и неоптимальные распределения примитивов, из-за чего в местах, требующих высокой детализации, не будет достаточного числа примитивов для их представления. Модификации, представленные в работе, частично решают две последние проблемы, но самопересечения всё ещё возможны, и ограничение на топологию мешает свободному применению мешей для оптимизации. В связи с этим дальнейшие работы этой группы методов используют функции расстояния со знаком.

В 2022 году репараметризацию впервые применили для функций расстояния со знаком [14]: сравнивались два подхода – прямое применение метода WAS к функциям расстояния, а также адаптированную под SDF версию. В этой работе было предложено использовать алгоритм редистансинга. Редистансинг – задача пересчёта значений сетки, чтобы её вершины содержали корректные расстояния до поверхности. Для решения этой задачи применяются методы, вычисляющие на сетке уравнения в частных производных [15], например уравнение Эйконала [16]. В дифференцируемом рендеринге пересчёт нужен, поскольку после шага оптимизации и обновления расстояний в узлах сетки она, строго говоря, уже не является SDF. Наконец, в работе 2024 года [17] предлагается переход от граничного интеграла к объёмному без построения векторного поля. Метод использует свойства функций расстояния и оценивает граничный интеграл *интегралом по релаксированной границе* – тонкой линии вокруг силуэта объекта. Использование SDF позволяет находить точки, принадлежащие этой области, во время трассировки лучей: значение функции расстояния в них будет меньше порогового, определяемого гиперпараметром. Этот метод продемонстрировал лучшие результаты в сравнении с [14], поэтому был выбран в качестве базового. Авторами была опубликована реализация метода на Github [18].

С другой стороны, на методы дифференцируемого рендеринга можно смотреть с точки зрения типа восстанавливаемых параметров. Сложной задачей является одновременное восстановление материала и поверхности объекта [19]. Рассмотренные в этом подразделе методы, в том числе базовый, могут использоваться для её решения. Однако в данной работе преследовалась цель восстановления только поверхности объекта. Тем не менее, предложенный метод не вносит никаких ограничений в сравнении с базовым ни в задачу восстановления поверхности, ни в задачу одновременного восстановления поверхности и материала.

3. Мотивация построения нового метода

В [14, 17] SDF задана на регулярной сетке, и для поиска пересечения луча и функции расстояния применяется метод sphere tracing [20]. Этот итеративный метод, предложенный в 1995 году, вычисляет значение расстояния в точке на луче, а затем делает шаг на это расстояние, пока либо не пересечёт поверхность, либо не выйдет за пределы сцены. Sphere tracing является наиболее распространённым методом для поиска пересечения луча и SDF, однако существуют альтернативы. В [21] были предложены два метода, основанные на следующем факте: внутри вокселя функция расстояния вдоль луча представима полиномом третьей степени (при использовании трилинейной интерполяции). Искать нули этого полинома можно либо явно вычисляя вещественные корни – этот метод авторы назвали *аналитическим*, – либо используя итерационный метод, в работе применяется метод Ньютона. По результатам сравнений, проведённых в [22], метод Ньютона и аналитический метод показали более высокую скорость в сравнении со sphere tracing в прямом рендеринге. Вероятно, они также способны повысить производительность дифференцируемого рендеринга. Помимо этого, в алгоритм базового метода можно внести изменения, которые ускорят семплирование.

Предложенный метод

Метод, предлагаемый в данной статье, во-первых, адаптирует алгоритмы поиска пересечения луча и функции расстояния из [21] под дифференцируемый рендеринг, а во-вторых, для базового метода реализует явное семплирование границы. Разделение расчёта внутреннего и граничного интеграла позволит задать разное число семплов для их оценки и, в частности, уменьшить это число для внутреннего. Подробнее про них рассказано в подразделе, посвященном явному семплированию. На рисунке 1 показана схема одной итерации базового алгоритма. Сперва выбирается ракурс, генерируются лучи с использованием параметров соответствующей камеры. Затем методом sphere tracing проводится рендеринг сцены: поиск пересечения лучей и поверхности объекта, а также поиск точек релаксированной границы; оцениваются внутренний и граничный интеграл. Наконец, считается функция потерь между референсным и полученным изображением, происходит обратное распространение ошибки. Красным цветом на рисунке 1 показаны предложенные модификации. Метод Ньютона [21] заменяет sphere tracing [20] в качестве метода поиска пересечения на этапе прямого рендеринга; на этапе явного семплирования идёт поиск точек для оценки граничного интеграла.

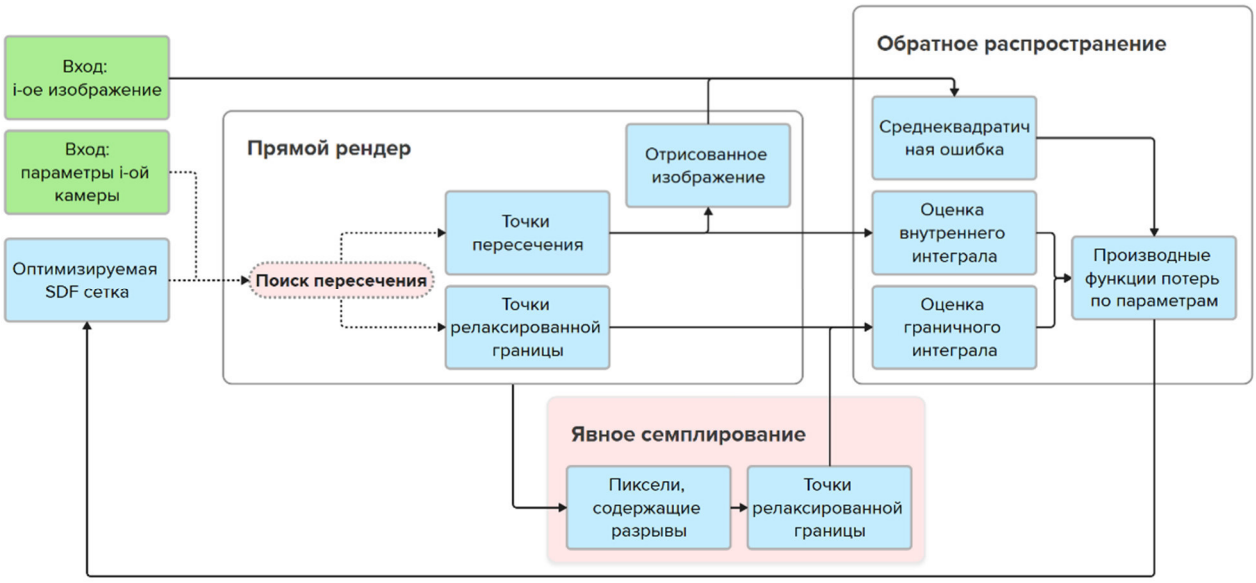


Рисунок 1. Схема одной итерации базового и предложенного методов. Зелёным цветом обозначены входные данные; синим – внутренние промежуточные. Стрелками показаны зависимости в их расчёте. Красным цветом выделены предложенные модификации

1. Явное семплирование

$$I = \int_{\Omega} f(\omega;\theta)\, d\omega\,. \tag{1}$$

В физически обоснованном рендеринге функция расчёта цвета пикселя представляет собой интеграл (1), где θ – параметры сцены. При попытке дифференцирования по параметрам, задающим форму объекта, подынтегральная функция f терпит разрывы, и положение этих разрывов зависит от параметров, по которым проводится дифференцирование. Поэтому в [12] исходный интеграл разбивается на сумму интегралов по областям непрерывности подынтегрального выражения D_i (2), при этом область интегрирования полученных слагаемых будет зависеть от параметров сцены. Согласно обобщению формулы Лейбница для дифференцирования под знаком интеграла, названному транспортной теоремой Рейнольдса, в этом случае производная интеграла будет равна

$$\frac{\partial I}{\partial \theta} = \sum_i \int_{D_i} \frac{\partial f(\omega;\theta)}{\partial \theta}\, d\omega + \sum_i \oint_{\partial D_i} f(\omega;\theta)\langle \partial_\theta \omega, \hat{\mathbf{n}}\rangle\, d\omega\,, \tag{2}$$

где ∂D_i – граница области D_i, $\hat{\mathbf{n}}$ – нормаль, направленная наружу. Первое слагаемое – это *внутренний* интеграл, второе – *граничный*. Идея базового метода [17] заключается в приближении граничного интеграла интегралом по релаксированной границе A:

$$\oint_{\partial D} f(\omega;\theta)\langle \partial_\theta \omega, \hat{\mathbf{n}}\rangle d\omega \;\approx\; I_A = \int_A \frac{1}{l(\omega)} f(\omega;\theta)\langle \partial_\theta \omega, \hat{\mathbf{n}}\rangle\, d\omega\,, \tag{3}$$
$$l(\omega) = \frac{\varepsilon}{\|x - x^*\|}.$$

Здесь $l(\omega)$ – ширина релаксированной границы, наблюдаемая из точки x, откуда пускаются лучи, x^* – точка релаксированной границы на луче с направлением ω, ε – гиперпараметр, пороговое значение для SDF.

В базовом методе внутренний и граничный интегралы оцениваются одновременно, с использованием одних и тех же семплов; для корректного восстановления поверхности авторы выбрали 64 семпла на пиксель. Однако для восстановления текстуры объекта можно использовать меньшее значение. Идея явного семплирования заключается в переносе расчёта граничного интеграла в отдельный шейдер (рис. 2). В отличие от основного, в нём не происходит обновление параметров цвета, поэтому отсутствует

сопутствующая логика. Благодаря разделению, новый шейдер позволяет сделать больше семплов для оценки граничного интеграла, уменьшив число семплов для основного шейдера. Если в оригинальной статье использовалось 64 семпла на пиксель, то в предложенном методе для основного шейдера это число снижается до 16. Для граничных пикселей выбрано 128 семплов.

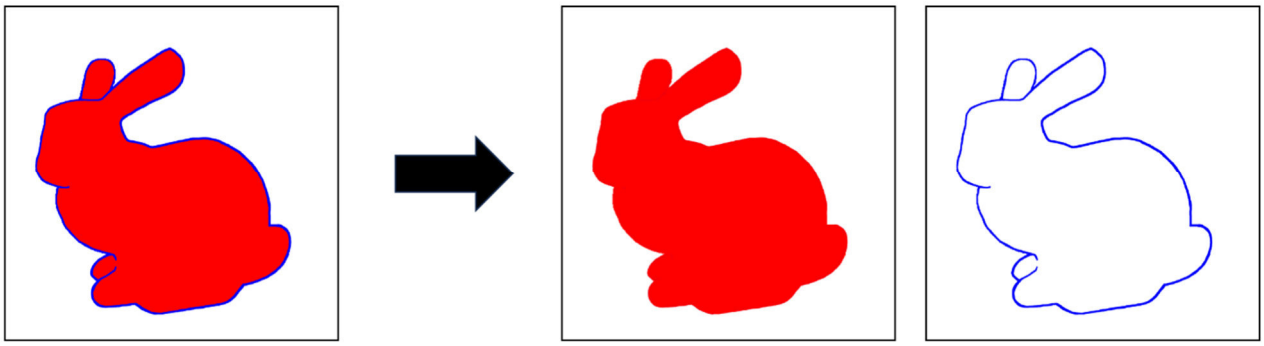


Рисунок 2. Визуализация явного семплирования. Вместо одновременного оценивания внутреннего и граничного интегралов (семплы красного и синего цвета соответственно) расчёт проводится сначала для внутреннего, затем для граничного

2. Альтернативные методы поиска пересечения

В качестве альтернативы sphere tracing добавлены метод Ньютона и аналитический метод. Эти методы обходят каждый воксель, в который попал луч, пока не будет найдено пересечение или луч не выйдет за пределы сцены. Для регулярной сетки, содержащей большое число пустых вокселей, эти методы были реализованы таким образом, чтобы во время поиска пересечения игнорировались воксели, если все его значения больше порогового значения, задающего релаксированную границу.

Также требовалось адаптировать для них поиск релаксированной границы. Точка, лежащая на луче, принадлежит релаксированной границе, если расстояние до поверхности в этой точке меньше порогового значения ε, а производная функции расстояния вдоль луча в ней равна нулю, и эта точка является точкой локального минимума (рис. 3). Оба метода основаны на том, что внутри вокселя функция расстояния вдоль луча представима в виде полинома третьей степени, для которого разными способами находятся корни. Таким образом, для поиска точек релаксированной границы в аналитическом методе и методе Ньютона нужно в каждом вокселе находить точки локального минимума полинома третьей степени, а затем проверять значение функции расстояния в них. Метод Ньютона уже находит нули производной для начального приближения, и поиск нужных значений внутри вокселя требует минимального усложнения логики алгоритма, поэтому он используется в сравнениях. Кроме того, при использовании трилинейной интерполяции нередки случаи, когда локальный минимум достигается на границе вокселей; этот случай обрабатывается отдельно.

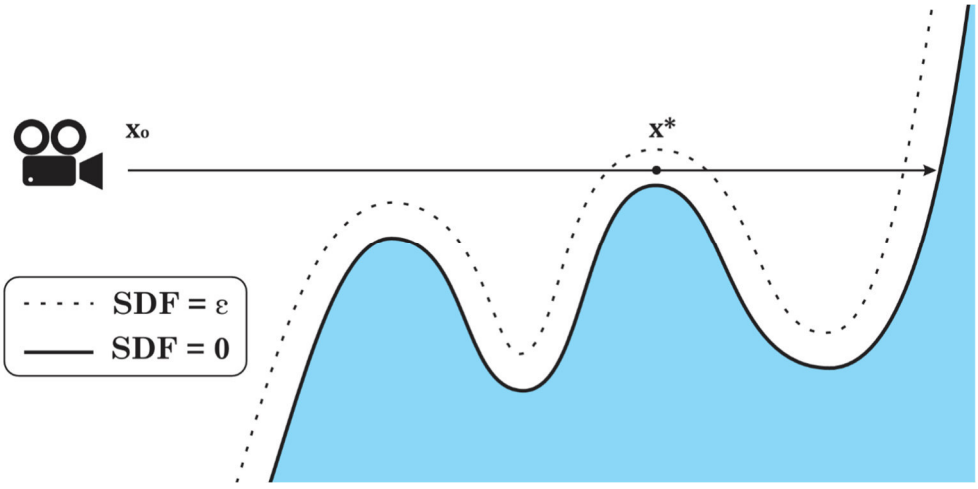


Рисунок 3. Визуализация пересечения луча, выпущенного из точки $\mathbf{x_o}$, с SDF. Точка $\mathbf{x}^*$ принадлежит релаксированной границе, так как 1) SDF вдоль луча достигает в $\mathbf{x}^*$ локального минимума, 2) $SDF(\mathbf{x}^*) < \varepsilon$

Детали реализации

В качестве инструментов разработки выбраны язык C++ и Vulkan для поддержки наиболее широкого спектра графических процессоров. Система Kernel Slicer [23] используется для переноса C++ кода на GPU. Разработка началась с реализации системы, которая может отрисовать объект, представленный функцией расстояния, методом sphere tracing, с поддержкой цвета и простой модели освещения. Следующим шагом была модификация функции рендеринга, которая, во-первых, добавляет поиск точек релаксированной границы в sphere tracing, а во-вторых, аналитически рассчитывает и распространяет производные по параметрам SDF – системы автоматического дифференцирования не используются. Для обеспечения параллелизма производные по параметрам записываются в отдельный массив. Затем, на этапе расчёта функции потерь, каждый элемент массива умножается на её производную, после чего применяется регуляризация, оптимизатор Adam и происходит обновление параметров. В конце для обновлённой сетки применяется алгоритм редистансинга для пересчёта и корректировки значений. Регуляризация имеет вид свёртки с дискретным оператором Лапласа, написана параллельная версия на CPU. Для редистансинга реализована параллельная версия метода fast sweeping [24], выполняется на CPU.

Эталонная и собственная реализации использовали одинаковую конфигурацию для экспериментов. Применяется модель освещения Ламберта с одним источником света. Значение гиперпараметра ε взято из оригинальной работы [17] и равно 10^{-4}. Для восстановления используются изображения размером 512x512 пикселей, полученные отрисовкой модели-референса с 16 ракурсов. Их положение соответствует ракурсам эталонной реализации [18], полученным с параметрами $\theta = 8,\ \varphi = 4$. На каждой итерации расчёт производных проводился для четырех ракурсов из 16. Оптимизация занимала 1000 итераций; каждые 200 итераций разрешение сетки увеличивалось в два раза, размер финальной сетки 512^3.

Результаты экспериментов

Эксперименты включают в себя сравнение средней скорости работы одной итерации базового и предложенного метода, а также валидацию, демонстрирующую сохранение качества реконструкции. В качестве GPU было выбрано Nvidia RTX 4090, так как эталонная реализация [18] написана с использованием CUDA. Эксперименты проводились для эталона и предложенных методов на идентичных сценах и ракурсах камер. Использовались шесть классических моделей: Bunny, Armadillo, Happy Buddha, Asian dragon – модели из Стэнфордского репозитория сканов [25]; Utah teapot – модель чайника за авторством Мартина Ньюэлла; Nefertiti – скан бюста Нефертити [26]. Модели содержат разные по степени детализированности поверхности, что подходит для валидации предлагаемого метода в сравнении со старым.

1. Сравнение скорости

В проводимом сравнении производительности и в данной работе, и в эталонной реализации [18] замерялось время работы шейдеров, так как в обеих программах логика трассировки сцены, поиска точек релаксированной границы и оценки обоих интегралов выполняется полностью на GPU. Для тестов, использующих явное семплирование границы, время означает среднее суммарное время работы основного и нового шейдеров; в остальных случаях указано среднее время выполнения основного шейдера. На рисунке 4 приведён график сравнения скорости для рассматриваемых представлений. Тестировались эталонная реализация [18], собственная реализация базового метода, а также две предложенные модификации и их комбинация. Замерялось среднее время выполнения одной итерации алгоритма оптимизации на графическом процессоре для шести моделей. Из полученных результатов сравнения можно сделать следующие выводы:

1. Собственная реализация базового метода сопоставима с эталонной в скорости.
2. Каждая из предложенных модификаций ускоряет базовый метод.
3. Итоговый метод показывает лучший результат, ускоряя базовый метод в три раза.

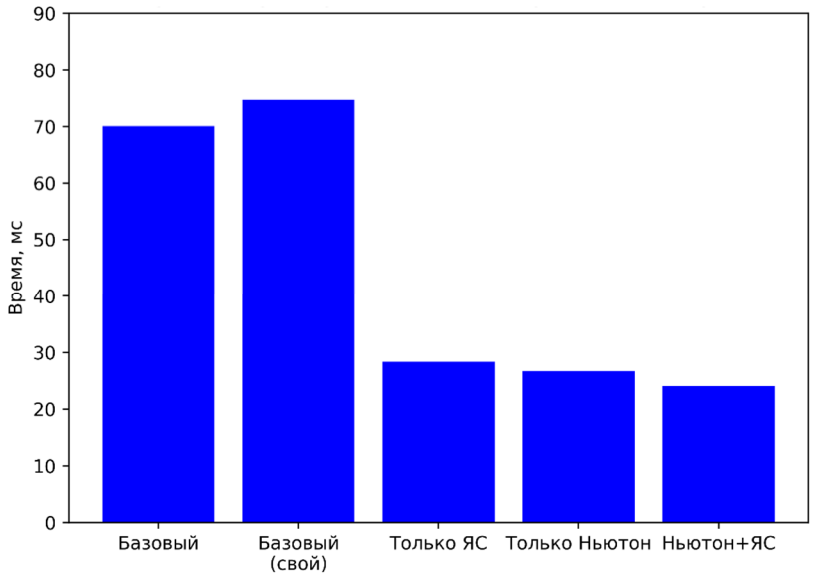
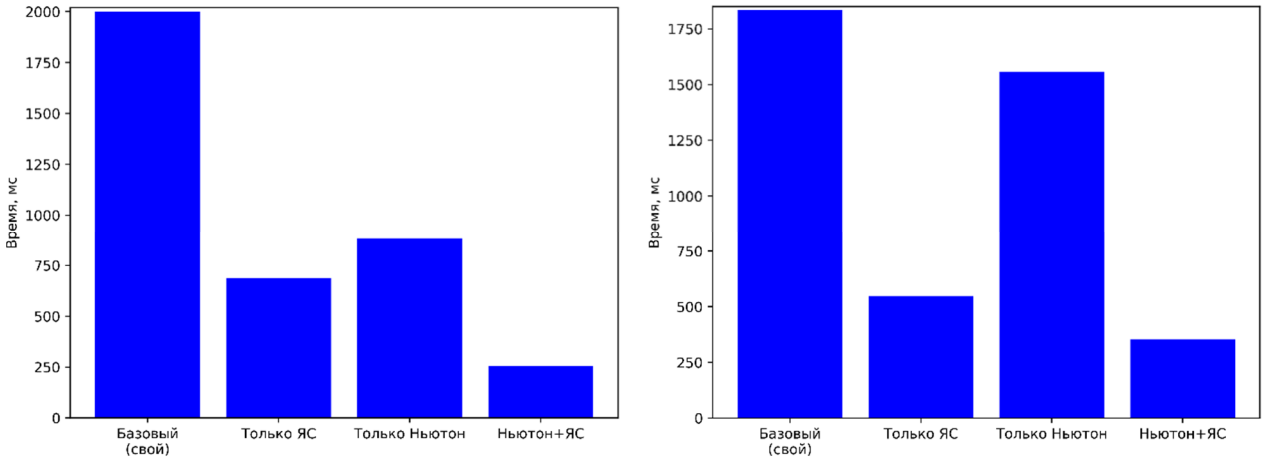


Рисунок 4. Среднее время работы одной итерации (GPU ядро)

Дополнительно проводилась оптимизация на других GPU. На рисунке 5 показано время работы на интегрированных графических ускорителях AMD Radeon Graphics (процессор AMD Ryzen 9 7950X) и Intel Iris Xe Graphics (процессор Intel Core i5-11300H). Базовый метод не включён в замеры, поскольку эталонная реализация не поддерживает GPU этих производителей.



а) б)

Рисунок 5. Среднее время работы одной итерации: *а*) на AMD Radeon Graphics; *б*) Intel Iris Xe Graphics

2. Валидация

Для валидации метода проведено сравнение результатов работы базового и итогового методов на рассматриваемых моделях с исходным мешем. После окончания процесса оптимизации сравнивалось среднее значение PSNR между восстановленной моделью и референсом на ракурсах, использованных в оптимизации.

На рисунке 6 приведено сравнение качества для эталона, своей реализации базового метода и метода со всеми модификациями. Очевидно, что качество реализованных методов сопоставимо с эталонной реализацией. Предложенный метод показывает немного лучшее качество в сравнении с базовым, так как он делает больше семплов для оценки интеграла, получая более точные градиенты. На рисунке 7 показаны рендеры исходного меша и восстановленной SDF-сетки для всех рассмотренных моделей. На рисунке 8 показаны различия между парами этих рендеров для каждой модели.

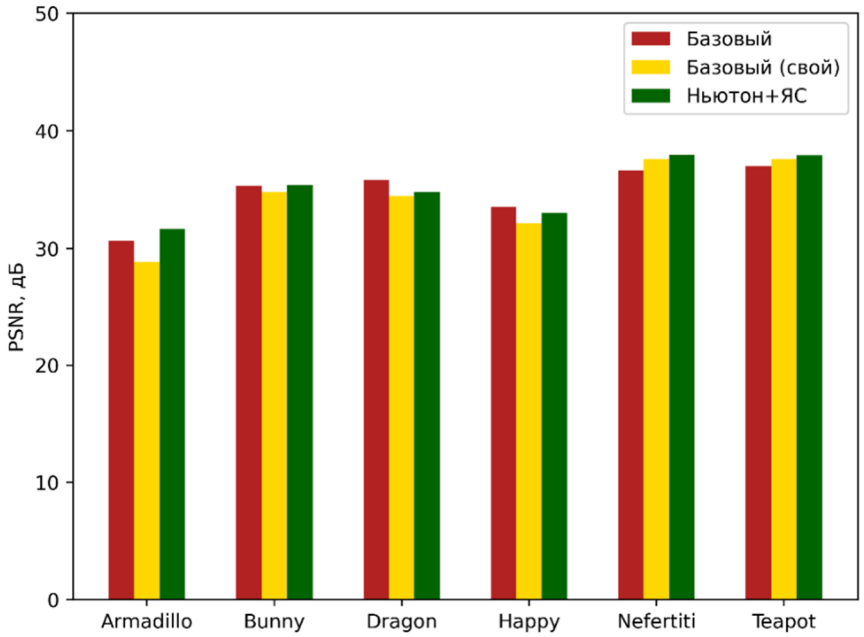


Рисунок 6. Среднее по ракурсам значение PSNR для восстановленных моделей

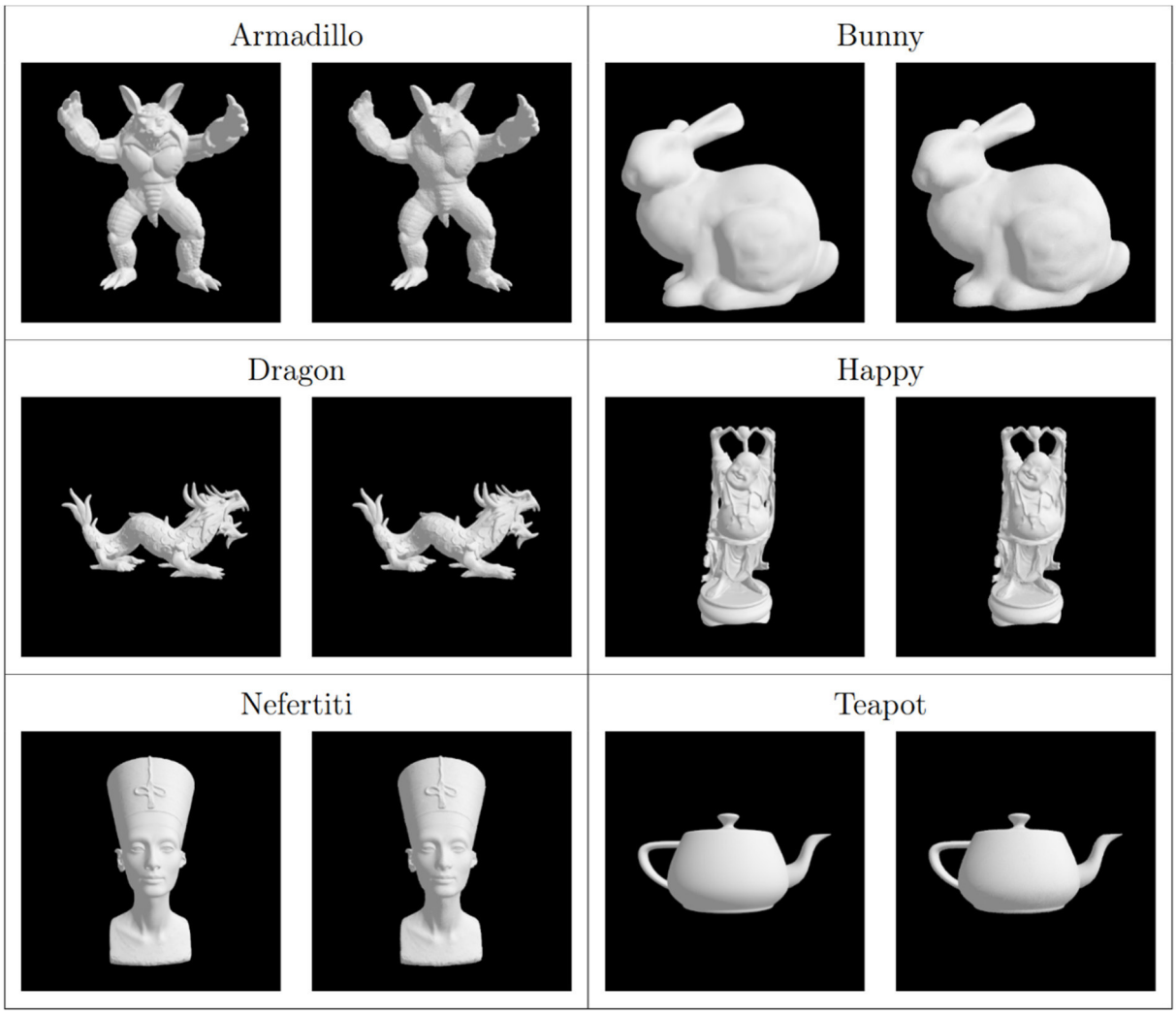


Рисунок 7. Сравнение исходного меша (*слева*) и его реконструкции (*справа*) для рассматриваемых моделей

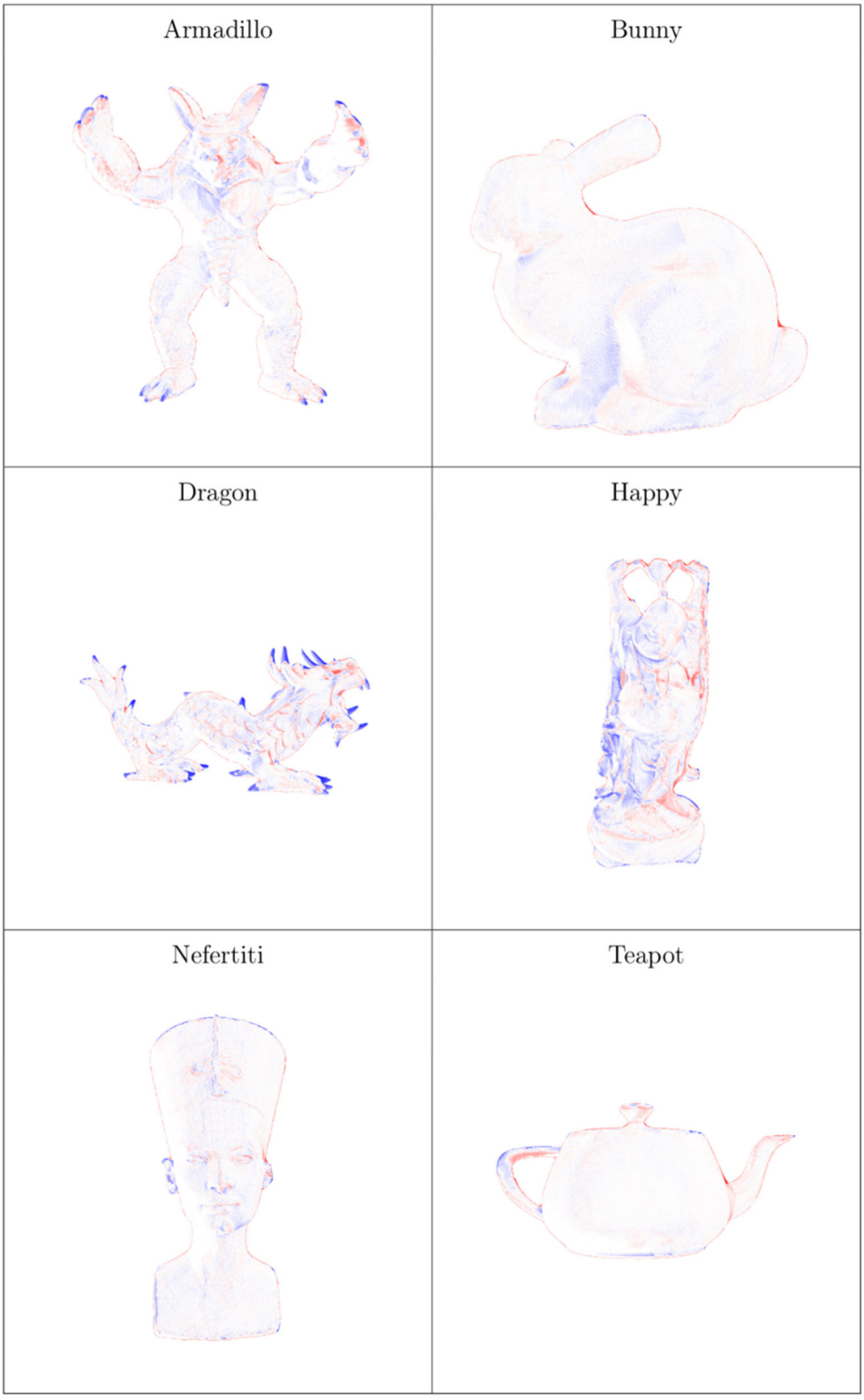


Рисунок 8. Различие между рендером восстановленной модели и исходным мешем (для лучшей видимости изменён контраст). Синим и красным цветом показаны отрицательное и положительное различия соответственно. Синие области на границах модели означают отсутствие у восстановленной модели поверхности там, где она есть у референса. Красные, наоборот, указывают на наличие поверхности там, где её нет у исходного меша

Заключение

В данной работе реализована кроссплатформенная программная система дифференцируемого рендеринга SDF, основанная на подходе с использованием интеграла по релаксированной границе, с использованием C++ и Vulkan. Реализация базового метода сравнима с авторской реализацией по производительности в задаче реконструкции геометрии. Далее были введены модификации, значительно ускоряющие процесс оптимизации без потери качества. Реализованный метод не вводит новых ограничений, но содержит ограничение, унаследованное от используемого SDF-представления: для представления мелких деталей требуются воксели маленького размера. Для SDF-сетки это достигается исключительно повышением разрешения всей сетки, что значительно увеличивает размер модели. При этом большая часть вокселей представляет пустое пространство, а значит, информация о них не требует хранения. Дальнейшая работа будет посвящена адаптации реализованного метода под разреженные представления.

Список литературы

1. M. M. Loper and M. J. Black, "OpenDR: An Approximate Differentiable Renderer", in *Computer Vision - ECCV* 2014, Zurich, Switzerland, 2014.

2. B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi and R. Ng, "NeRF: representing scenes as neural radiance fields for view synthesis", *Commun. ACM*, vol. 65, no. 1, p. 99–106, 2021.

3. B. Kerbl, G. Kopanas, T. Leimkuehler and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering", *ACM Trans. Graph*., vol. 42, no. 4, pp. 1-14, 2023.

4. P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura and W. Wang, "NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction", *Advances in Neural Information Processing Systems*, 2021.

5. T. Müller, A. Evans, C. Schied and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding", *ACM Trans. Graph*., vol. 41, no. 4, pp. 1-15, 2022.

6. Z. Li, T. Müller, A. Evans, R. H. Taylor, M. Unberath, M.-Y. Liu and C.-H. Lin, "Neuralangelo: High-Fidelity Neural Surface Reconstruction", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, 2023.

7. J. J. Park, P. Florence, J. Straub, R. Newcombe and S. Lovegrove, "DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, California, 2019.

8. Y. Wang, Q. Han, M. Habermann, K. Daniilidis, C. Theobalt and L. Liu, "NeuS2: Fast Learning of Neural Implicit Surfaces for Multi-view Reconstruction", *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Paris, 2023.

9. A. Dogaru, A. T. Ardelean, S. Ignatyev, E. Zakharov and E. Burnaev, "Sphere-Guided Training of Neural Implicit Surfaces", *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, California, 2023.

10. T.-M. Li, M. Aittala, F. Durand and J. Lehtinen, "Differentiable Monte Carlo ray tracing through edge sampling", *ACM Trans. Graph*., vol. 37, no. 6, pp. 1-11, 2018.

11. G. Loubet, N. Holzschuch and W. Jakob, "Reparameterizing discontinuous integrands for differentiable rendering", *ACM Trans. Graph*., vol. 38, no. 6, pp. 1-14, 2019.

12. S. P. Bangaru, T.-M. Li and F. Durand, "Unbiased warped-area sampling for differentiable rendering", *ACM Trans. Graph.*, vol. 39, no. 6, pp. 1-18, 2020.

13. B. Nicolet, A. Jacobson and W. Jakob, "Large steps in inverse rendering of geometry", *ACM Trans. Graph*., vol. 40, no. 6, pp. 1-13, 2021.

14. D. Vicini, S. Speierer and W. Jakob, "Differentiable signed distance function rendering", *ACM Trans. Graph*., vol. 41, no. 4, p. 18, 2022.

15. J. A. Sethian, "A Fast Marching Level Set Method for Monotonically Advancing Fronts", *National Academy of Sciences of the United States of America*, Washington, 1996.

16. H. Zhao, "A fast sweeping method for Eikonal equations", *Math. Comp*., vol. 74, pp. 603-627, 2005.

17. Z. Wang, X. Deng, Z. Zhang, W. Jakob and S. Marschner, "A Simple Approach to Differentiable Rendering of SDFs", *SIGGRAPH Asia 2024 Conference Papers*, Tokyo, 2024.

18. Z. Wang, "The official implementation for 'A Simple Approach to Differentiable Rendering of SDFs". Available: https://github.com/zichenwang01/relaxed-boundary. [Дата обращения: 07. 07. 2025].

19. F. Luan, S. Zhao, K. Bala and Z. Dong, "Unified Shape and SVBRDF Recovery using Differentiable Monte Carlo Rendering," *Computer Graphics Forum*, vol. 40, no. 4, pp. 101-113, 2021.

20. J. Hart, "Sphere Tracing: A Geometric Method for the Antialiased Ray Tracing of Implicit Surfaces", *The Visual Computer*, vol. 12, p. 527–545, 1996.

21. H. Hansson Söderlund, A. Evans and T. Akenine-Möller, "Ray Tracing of Signed Distance Function Grids", *Journal of Computer Graphics Techniques (JCGT)*, vol. 11, no. 3, pp. 94-113, 2022.

22. A. S. Budak, A. R. Garifullin, V. A. Frolov and V. A. Galaktionov, "Study of Surface Representation Methods Based on Signed Distance Functions", *Programming and Computer Software*, vol. 51, p. 131–139, 2025.

23. V. A. Frolov, V. V. Sanzharov and V. A. Galaktionov, "kernel_slicer: high-level approach on top of GPU programming API", *Ivannikov Ispras Open Conference (ISPRAS)*, Moscow, 2022.

24. M. Detrixhe, F. Gibou and C. Min, "A parallel fast sweeping method for the Eikonal equation", *Journal of Computational Physics*, vol. 237, pp. 46-55, 2013.

25. Stanford Computer Graphics Laboratory, "The Stanford 3D Scanning Repository". Available: https://graphics.stanford.edu/data/3Dscanrep/. [Дата обращения: 07. 07. 2025].

26. J. N. Nelles and N. Al-Badri, "Nefertiti scan", 2015. Available: https://nefertitihack.alloversky.com/. [Дата обращения: 07. 07. 2025].