

# Дифференцируемые процедурные модели для реконструкции 3D-моделей по одному изображению

А.Р. Гарифуллин<sup>1</sup>, Н.Д. Майоров<sup>1</sup>, В.А. Фролов<sup>1,2</sup>

<sup>1</sup> МГУ им. М.В. Ломоносова, ГСП-1, Ленинские горы, д.1, стр.52, факультет ВМК, Москва, 119234, Россия

<sup>2</sup> ИПМ им. М.В. Келдыша РАН, Миусская пл., д.4, Москва, 125047, Россия

## Аннотация

Большинство существующих методов для реконструкции 3D-моделей по одному изображению основаны на глубоком обучении с использованием неявных или воксельных представлений сцены и не способны создавать детализированные и качественные модели и текстуры, которые можно будет использовать на практике. Дифференцируемый рендеринг, с другой стороны, может использоваться для достижения высокой детализации, но требует множества изображений объекта с разных ракурсов. Мы предлагаем новый подход к 3D-реконструкции с по одному изображению, который использует входные параметры процедурного генератора в качестве представления сцены. Вместо того, чтобы оценивать положения вершин модели напрямую, мы оцениваем входные параметры процедурного генератора, минимизируя ошибку между входным изображением и реконструируемой моделью. Мы используем дифференцируемый рендеринг и дифференцируемые процедурные генераторы, что позволяет нам применять градиентные методы оптимизации и создавать высокодетализированную модель из одного изображения.

## Ключевые слова

Процедурная генерация, 3D реконструкция, дифференцируемый рендеринг.

# Differentiable Procedural Models for Single-view 3D Mesh Reconstruction

Albert Garifullin<sup>1</sup>, Nikolay Maiorov<sup>1</sup>, Vladimir Frolov<sup>1,2</sup>

<sup>1</sup> Moscow State University, GSP-1, Leninskie gory, 1, 52, CMC, Moscow, 119234, Russia

<sup>2</sup> Keldysh Institute of Applied Mathematics, Miusskaya sq., 4, Moscow, 12507, Russia

## Abstract

Most of the existing solutions for single-view 3D object reconstruction rely on deep learning techniques that use implicit or voxel representations of the scene. However, these approaches struggle to generate detailed and high-quality meshes and textures that can be directly applied in practical applications. On the other hand, differentiable rendering techniques can produce superior mesh quality, but they typically require multiple images of the object. We propose a novel approach to single-view 3D reconstruction that leverages procedural generator input parameters as a scene representation. Instead of directly estimating the vertex positions of the mesh, we estimate the input parameters of a procedural generator by minimizing the silhouette loss function between reference and rendered images. By employing differentiable rendering and differentiable procedural generators, we can optimize the loss function using gradients. This enables us to create highly detailed models from a single image.

---

ГрафиКон 2023: 33-я Международная конференция по компьютерной графике и машинному зрению, 19-21 сентября 2023 г., Институт проблем управления им. В.А. Трапезникова Российской академии наук, г. Москва, Россия

EMAIL: albgar-14@yandex.ru (А.Р. Гарифуллин) maiorov.kolyan@yandex.ru (Н.Д. Майоров); vfrolov@graphics.cs.msu.ru (В.А. Фролов)

ORCID: 0000-0003-3802-1774 (А.Р. Гарифуллин); 0009-0001-2677-881X (Н.Д. Майоров); 0000-0001-8829-9884 (В.А. Фролов)



© 2023 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

**Keywords**

Procedural generation, 3D reconstruction, differentiable rendering.

**1. Введение**

Индустрия видеоигр и компьютерной графики в кино показывает быстрый и стабильный рост на протяжении много лет. Для наполнения виртуальных миров требуется все больше разнообразных объектов, при этом растут требования к их качеству, детализации и реалистичности. Создание моделей вручную требует больших трудозатрат и более предпочтительным является получение 3D моделей реально существующих объектов. Хотя современные технологии сканирования и 3D-реконструкции на основе облака точек позволяют создавать высококачественные модели, они часто требуют сложного оборудования и значительного количества времени. Кроме того, может оказаться невозможным отсканировать объект или получить достаточное количество его изображений с разных ракурсов. Для этих случаев разрабатываются методы 3D-реконструкции по одному изображению. Методы глубокого обучения, такие как [1] и [2], добились значительного прогресса в решении этой задачи, но качество реконструированных моделей часто недостаточно для их использования на практике. Обычно методы глубокого обучения не учитывают важный аспект человеческого восприятия: то, что человек в первую очередь распознает структуру объекта, а не только его форму. Процедурная генерация - это альтернативный метод получения 3D-моделей. Процедурные генераторы обычно представляют собой набор правил, которые устанавливают связь между входными параметрами и 3D-объектами определенного класса. Таким образом, процедурный генератор также, как и человек, работает с объектами на уровне их структуры. В этой статье мы предлагаем новый подход, который использует процедурную генерацию для реконструкции 3D-моделей и текстур по одному или нескольким изображениям. Вместо того, чтобы реконструировать позиции вершин модели напрямую, мы находим входные параметры процедурного генератора, которые соответствуют схожей с той, что представлена на входном изображении. В такой постановке задача сводится к минимизации функции потерь с последующим обратным распространением ошибки для нахождения минимума градиентным методом. Наш подход значительно снижает размерность оптимизируемой функции и гарантирует получение модели с корректной структурой, т.к. ее обеспечивает процедурный генератор. Также, поскольку результатом работы алгоритма является набор входных параметров процедурного генератора, сохраняется возможность дальнейшей модификации 3D на уровне структуры, а не отдельных вершин.

**2. Обзор существующих методов****2.1. Восстановление 3D-модели по одному изображению**

Восстановление 3D-модели по одному изображению остается очень сложной задачей, в первую очередь потому, что эта задача является существенно некорректной. На одном изображении видна лишь часть объекта, а для восстановления невидимых частей необходимо использовать дополнительные знания о структуре восстанавливаемого объекта. По этой причине методы машинного обучения являются самой распространенной группой методов для решения этой задачи. Таким образом, методы, основанные на обучении, стали доминирующими в этой области. Эти методы используют различные представления сцены, такие как полигональные сетки (меш) [3] [4] [5], воксельные сетки [6 – 7], облака точек [8 – 9] или неявные функции [10]. Среди этих методов реконструкция мешей является наиболее актуальной для нашей работы. Большинство методов реконструкции меша по одному изображению используют архитектуру энкодер-декодер, в которой энкодер извлекает признаки из входного изображения, а декодер деформирует исходный меш (например, сферу) до целевой формы. Важно отметить, что эти методы обучаются и оцениваются на одних и тех же категориях объектов и их применение к другим категориям требует повторного обучения и наличия достаточно большой выборки 3D-моделей определенного типа. Недавняя работа [11] показала, что такие подходы к 3D-

реконструкции в первую очередь выполняют распознавание типа объекта, а не его реконструкцию и повышение качества результата их работы весьма затруднительно.

Есть также несколько работ, реализующих восстановление 3D-модели по одному изображению без ограничения на тип объекта [12] [2], что, однако, приводит к более низкому качеству моделей тех классов, которые не присутствовали в обучающей выборке.

## 2.2. Дифференцируемый рендеринг

Физически корректный дифференцируемый рендеринг [13] [14] - это активная область исследований, тесно связанная с высокоточной реконструкцией 3D-моделей и материалов. Дифференцируемый рендеринг позволяет вычислять градиент функции потерь по параметрам сцены, который может быть использован для минимизации функции потерь при градиентном спуске. Выбор представления сцены также важно при реконструкции на основе дифференциального рендеринга. Представление на основе полигональных сеток на данный момент является самым распространенным и хорошо изученным, и для этой задачи были предложены специальная регуляризация [15] и модификация для оптимизатора Adam [16]. В недавней работе [17] были использованы результаты этих и других исследований, что позволило авторам добиться очень высокого качества реконструкции при относительно небольшом количестве входных изображений. Были также изучены другие представления сцен. В работе [18] был предложен алгоритм рендеринга дифференцируемой функции расстояния со знаком и его использование для 3D реконструкции. Дифференцируемый рендеринг также может быть объединен с глубоким обучением [2], чтобы обеспечить контроль качества изображения и регуляризацию во время обучения.

## 2.3. Альтернативные представления сцены

Представление модели в виде полигональной сетки является самым распространенным в компьютерной графике и позволяет применять эффективные алгоритмы рендеринга, а также легко экспортировать полученные в ходе реконструкции модели в другие приложения. Однако оптимизация геометрии поверхности на основе изображений может быть сложной задачей из-за невыпуклости оптимизируемой функции, особенно в сценах, содержащих мелкие детали. Использование воксельных сеток [19] [20] позволяет существенно упростить процесс поиска минимума, но значительно ухудшает восстановление мелких деталей. В качестве альтернативы, также было показано, что представления фигур в виде облака точек позволяют создавать высококачественные реконструкции сцен [21] [22]. Другим ключевым инструментом для масштабируемого представления сцен является использование нейронных сетей на основе координат, также известных как нейронные поля [23] [24] [25], которые не ограничены пределами разрешения дискретизированных сеток и обобщаются на сигналы более высокой размерности, такие как направленное излучение [23]. Нейронные поля [25] продемонстрировали способность обрабатывать сложные сцены и выдавать достаточно качественные результаты за считанные секунды. Однако, чтобы использовать полученные результаты на практике, требуется преобразовать их из этого неявного представления в полигональную сетку. Такие преобразования могут представлять серьезную проблему и их результатом становятся неоптимальные модели с большим количеством треугольников и более низким качеством изображения.

## 2.4. Процедурная генерация

Процедурная генерация контента - это хорошо известный подход к созданию разнообразных виртуальных миров. Он используется на разных уровнях, начиная от отдельных объектов и заканчивая крупномасштабными открытыми мирами и игровыми сценариями [26] [27]. Многие генераторы процедурных объектов работают как функции, которые преобразуют вектор числовых параметров в 3D-модель определенного класса, например, дерева или здания. Есть

несколько работ, посвященных обратной процедурной генерации, которая включает в себя оценку входных параметров для генератора с учетом входной модели или изображения для создания моделей, аналогичных входным [28] [29] [30]. Однако, поскольку нетривиальные процедурные генераторы являются необратимыми функциями, существующие работы обладают ограниченной способностью к обобщению и сосредоточены на очень специфических процедурных моделях.

### 3. Предложенный метод

Предлагаемый способ реализует реконструкцию объектов и текстур по одному или нескольким изображениям с использованием дифференцируемого рендеринга и дифференцируемого процедурного генератора. Он ограничен определенным классом объектов, которые способен создавать выбранный генератор. Метод состоит из трех этапов: (1) реконструкция полигональной сетки, (2) реконструкция текстуры и (3) постобработка. Реконструкция как полигональной сетки, так и текстуры основаны на минимизации функций потерь (функция потери силуэта и текстуры соответственно). Мы определяем все параметры сцены как  $\pi = (P, T, S, C)$ , где  $P$  - входные параметры процедурного генератора,  $T$  - текстура объекта (или текстуры),  $S$  - глобальные параметры сцены (например, источники света, окружающий свет), а  $C$  - параметры камер.

Функция потерь силуэта определяется следующим образом:

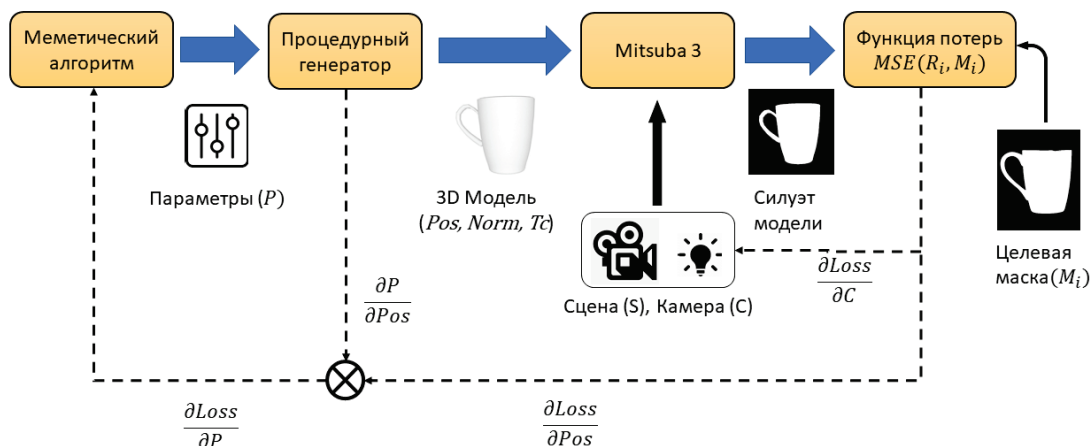
$$Loss_s(P, T, S, C) = \frac{1}{N} * \sum_{i=1}^N MSE(Render_s(Gen(P), T, S, C), M_i)$$

Функция потерь текстуры:

$$Loss_t(P, T, S, C) = \frac{1}{N} * \sum_{i=1}^N MSE(Render_t(Gen(P), T, S, C), R_i)$$

$R_i$  и  $M_i$  - эталонные изображения и маски (силуэты) для этих изображений соответственно,  $N$  - количество заданных эталонных изображений,  $Gen(P) = (pos, norm, tc)$  – полигональная сетка, созданная процедурным генератором,  $Render_s$  - это изображение сцены, отрисованное в режиме силуэта, а  $Render_t$  - это изображение сцены, отрисованное в обычном режиме, с текстурами и освещением.

На первом шаге мы находим  $P, C = argmin_{P,C}\{Loss_s(P, T, S, C)\}$  или  $P = argmin_P\{Loss_t(P, T, S, C)\}$ , если указаны позиции и параметры камер. На этапе восстановления текстуры мы находим  $T = argmin_T\{Loss_t(P, T, S, C)\}$  с фиксированными камерами и геометрией. Для обоих этапов мы используем итеративную стратегию оптимизации на основе градиента. Более подробная информация о процессе оптимизации приведена ниже, но ключевым моментом является то, что для этого требуются градиенты функции потерь по отношению к параметрам сцены.  $\frac{dLoss}{dT}$  и  $\frac{dLoss}{dC}$  могут быть легко получены посредством дифференцируемого рендеринга. Наша реализация использует Mitsuba 3 [31] для этой цели. Получение  $\frac{dLoss}{dP}$  сложнее. Учитывая правило дифференцирования сложной функции и тот факт, что силуэт не зависит от векторов нормалей модели и координат текстуры, мы получаем  $\frac{dLoss}{dP} = \frac{dLoss}{dpos} * \frac{dpos}{dP}$ . Частные производные функции потерь по позициям вершин модели  $\frac{dLoss}{dpos}$  также могут быть получены за счет дифференцируемого рендеринга, а якобиан  $\frac{dpos}{dP}$  получается из процедурного генератора. Весь процесс оптимизации функции потери силуэта проиллюстрирован ниже на рисунке 1.



**Рисунок 1** – Схема предложенного метода реконструкции модели. 3D модель

Модель, созданная процедурным генератором (Our Proc. Gen.) и поступает вместе с параметрами сцены в дифференцируемый рендер (Mitsuba 3). Далее вычисляется функция потерь для полученного изображения и входного изображения и производится обратное распространение ошибки и изменение параметров для следующей итерации оптимизации. Модификация параметров генератора осуществляется меметическим алгоритмом (memetic algorithm).

## 4. Дифференцируемые процедурные генераторы

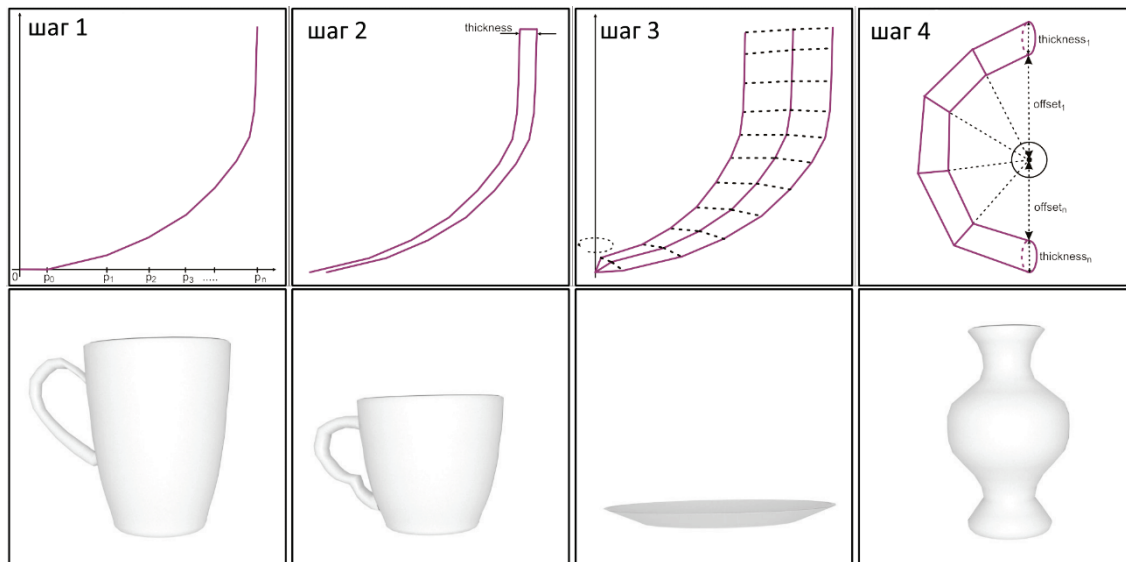
Алгоритм, описанный в предыдущем разделе, рассматривает процедурный генератор как функцию, которая может создавать модель в формате  $(pos, norm, tc)$  и якобиан  $\frac{dpos}{dP}$  из входного вектора  $P$ . На практике почти невозможно создать генератор нетривиальных объектов, который работал бы как гладкая дифференцируемая функция при таком определении. В первую очередь это связано с объектами, обладающими дискретными свойствами, такими как количество этажей в здании. Для простоты и единообразия мы по-прежнему включаем эти параметры в вектор  $P$ , но для каждого параметра  $P_k$  этого типа мы предполагаем, что  $\frac{dpos}{dP_k}$ .

Для этой работы созданы два разных процедурных генератора: для посуды и зданий. Оба были разработаны с нуля с использованием библиотеки автоматического дифференцирования CppAD [32] для получения требуемых производных. Генератор посуды (рисунок 2) является примером простого алгоритма, который имеет только один дискретный параметр. Напротив, здания имеют множество дискретных элементов в своей структуре, и генератор зданий (рисунок 3) был создан для демонстрации способности нашего метода справляться с такими задачами.

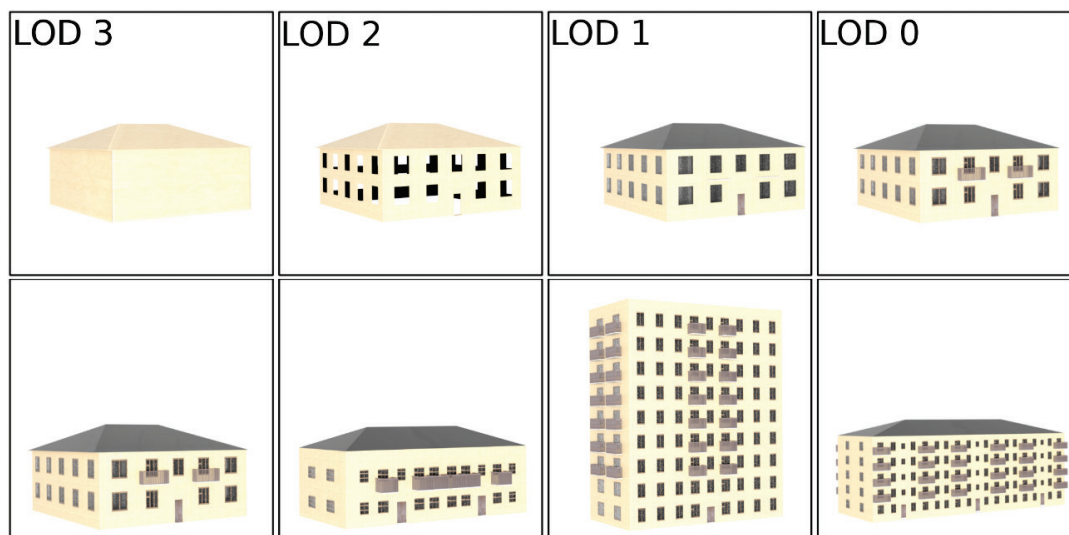
## 5. Оптимизация

Даже относительно простые процедурные модели, описанные в предыдущем разделе, оказываются чрезвычайно сложными функциями для оптимизации по двум причинам: во-первых, большого количества нелинейных внутренних зависимостей в генераторе и, во-вторых, разнородных входных параметров, некоторые из которых являются целыми числами с ограниченным набором возможных значений. По этим причинам минимизируемая функция потерь имеет множество локальных минимумов и поверхностей разрыва сложной формы и требует более продвинутых методов оптимизации, чем при прямом применении дифференцируемого рендеринга. Одна из предыдущих попыток 3D-реконструкции с помощью процедурных генераторов растений [30] использовала специальный вариант генетического

алгоритма [33] для поиска относительно хороших решений без вычисления градиента, но такой подход имел ограниченную точность и подходил только для приблизительного восстановления формы деревьев. Возможность рассчитать градиент функции потерь расширяет список доступных методов оптимизации. Однако мы все еще не можем получить производную по параметрам, которые представляют дискретные характеристики процедурной модели (например, количество этажей в здании).



**Рисунок 2** – Генератор посуды. Верхний ряд: этапы генерации. 1) Создание сплайна из вектора вертикальных смещений. 2) Преобразование, чтобы создать замкнутый сплайн с заданной толщиной. 3) Вращение. 4) (Необязательно) Создание ручки из сплайна окружности и вектора смещений. Количество точек в сплайнах может быть изменено для получения различных уровней детализации на основе одного и того же набора параметров. Нижний ряд: несколько примеров сгенерированных объектов. 1) Кружка 2) Чайная чашка 3) Тарелка 4) Ваза



**Рисунок 3** – Генератор зданий. Верхний ряд: Уровни детализации для одного и того же здания. 1) Коробка с крышей. 2) Только наружные стены. 3) Здание без мелких деталей 4) Полностью детализированное здание. Нижний ряд: несколько примеров сгенерированных зданий

Чтобы решить эту проблему, мы используем меметический алгоритм [34], который сочетает генетический алгоритм с оптимизацией на основе градиента. Мы начинаем с набора значений параметров, который берется из заранее подготовленного списка. Каждый набор параметров из этого соответствует корректной модели, которую по этому набору может создать процедурный генератор (примеры моделей из такого списка для генератора посуды показаны на рисунке 2).

Меметический алгоритм выполняет случайные мутации и рекомбинацию исходной популяции наряду с оптимизацией на основе градиента. Этот процесс требует нескольких тысяч итераций, однако он может быть выполнен на моделях с низким уровнем детализации и низким разрешением рендеринга, что снижает время выполнения одной итерации до нескольких сотен миллисекунд. Чтобы еще больше улучшить качество, мы используем решение, полученное меметическим алгоритмом, в качестве начального приближения для следующего раунда оптимизации на основе градиента с более высоким уровнем детализации и более высоким разрешением рендеринга. Для восстановления текстуры выполняется только этап оптимизации на основе градиента. Результаты на каждом шаге показаны на рисунке 4.



**Рисунок 4** – Результаты реконструкции модели после каждого шага оптимизации. Слева направо: входное изображение (reference), генетический алгоритм с разрешением рендеринга 128x128 (шаг 1), градиентный спуск с разрешением 256x256 (шаг 2) и 512x512 (шаг 3) соответственно

## 6. Результаты

В этом разделе показаны результаты 3D-реконструкции по одному и нескольким входным изображениям с использованием нашего метода. Мы использовали как реальные фотографии, так и синтезированные изображения в качестве входных данных для нашего алгоритма. На рисунке 5 показаны результаты нашей реконструкции по одному изображению на фотографиях, сделанных в неконтролируемой среде, т.е. мы не располагаем данными о позиции и характеристиках камеры, а также освещении на фотографии. Наш алгоритм позволил реконструировать модель для обеих объектов с высокой точностью. Кроме того, была реконструирована текстура, с некоторыми предположениями относительно невидимых областей. Для генератора здания мы протестировали два различных метода реконструкции. Первый, "текстурированная коробка", заключается в создании только простейшей модели внешних стен и крыши, а затем покрытии ее реконструированной текстурой. Этого может быть достаточно для представления удаленных объектов на сцене или авиасимуляторов. Второй - работа с детализированной моделью, но без реконструкции текстуры, так как разрешение входного изображения недостаточно для того, чтобы корректно реконструировать текстуры отдельных частей здания. На рисунке 6 показаны результаты обоих методов.

### 6.1. Сравнения с существующими методами

Мы сравнили результаты работы предложенного метода с существующими решениями для 3D-реконструкции, чтобы продемонстрировать, что наш подход способен обеспечить визуально лучшие результаты даже на основе одного входного изображения. Рисунок 7 демонстрирует, что процедурно сгенерированная модель, полученная нашим методом, имеет более гладкую структуру и лучше восстанавливает вогнутую форму входного объекта. Процедурная модель применяет строгие правила к структуре модели и гарантирует, что полученная модель не будет содержать дыр, неровностей и иных геометрических артефактов, если они не предусмотрены структурой модели. Рисунок 8 показывает, как увеличение количества входных изображений влияет на качество реконструкции. Наличие хотя бы двух изображений с разных сторон может существенно увеличить качество реконструкции текстуры, однако для восстановления геометрии достаточно и одного изображения.



**Рисунок 5** – Результаты реконструкции по одной фотографии с использованием процедурного генератора посуды. Верхний ряд: входные изображения, нижний ряд: реконструированные модели



**Рисунок 6** – Результаты реконструкции по одной фотографии с использованием процедурного генератора зданий. Слева направо: входное изображение, маска окон, результат реконструкции режиме "текстурированная коробка", результат детальной реконструкции с использованием текстур по умолчанию

## 7. Заключение

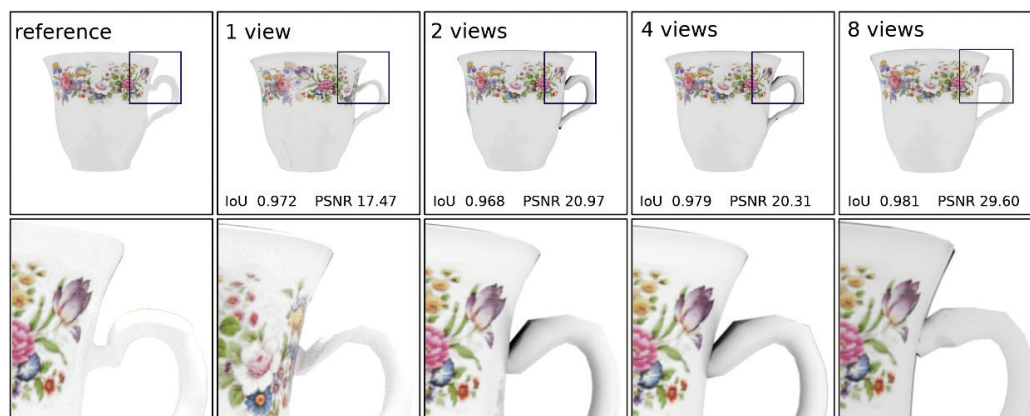
В этой работе мы представили новый подход к 3D-реконструкции по одному изображению, который оценивает входные параметры процедурного генератора для восстановления модели. Мы реализовали эффективный алгоритм поиска оптимального набора параметров на основе входного изображения. Наш метод демонстрирует лучшие результаты по сравнению с существующими подходами и создает модели с меньшим количеством артефактов. Однако наш подход имеет некоторые ограничения, прежде всего в том, что он хорошо работает только с классом объектов, которые может создать используемый процедурный генератор. Процедурные генераторы, используемые в этой работе, были созданы с нуля и ограничены в своих возможностях. В ходе будущих исследований мы планируем обобщить наш подход на более широкий класс объектов. Это требует либо интеграции существующих генераторов в наш конвейер реконструкции, либо использования генеративных нейронных сетей для обобщения процедурных генераторов или непосредственного создания моделей.





**Рисунок 7** – Сравнение результатов, полученные с использованием нашего метода, с дифференцируемой реконструкцией SDF [18], InstantNGP [25] и Pixel2Mesh [3]

Мы измерили среднее значение метрики intersection over union (IoU) между масками эталонной и реконструированной модели для 64 позиций камеры. Наш подход намного лучше, чем еконструкция с использованием Pixel2Mesh по одному изображению, и дает сопоставимые результаты с методами реконструкции по нескольким изображениям.



**Рисунок 8** – Результаты 3D-реконструкции с нашим подходом при разном количестве входных изображений.

Для реконструкции полигональной сетки достаточно одной точки обзора, и дальнейшее увеличение не приводит к значительному улучшению. Однако большее количество точек обзора позволяет выполнить более точную реконструкцию текстуры.

## 8. Благодарности

Работа поддержана некоммерческим фондом развития науки и образования «Интеллект».

## 9. СПИСОК ИСТОЧНИКОВ

- [1] PONTES J. K., KONG C., SRIDHARAN S., LUCEY S., ERIKSSON A., FOOKES C.: Image2mesh: A learning framework for single image 3d reconstruction. In *Computer Vision—ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part I 14 (2019)*, Springer, pp. 365–381.
- [2] YANG X., LIN G., ZHOU L.: Zeromesh: Zero-shot single-view 3d mesh reconstruction. arXiv preprint arXiv:2208.02676 (2022).
- [3] WANG N., ZHANG Y., LI Z., FU Y., LIU W., JIANG Y.G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV) (2018)*, pp. 52–67.
- [4] NIE Y., HAN X., GUO S., ZHENG Y., CHANG J., ZHANG J. J.: Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)*, pp. 55–64.
- [5] YE Y., TULSIANI S., GUPTA A.: Shelf-supervised mesh prediction in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)*, pp. 8843–8852.
- [6] CHOY C. B., XU D., GWAK J., CHEN K., SAVARESE S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VIII 14 (2016)*, Springer, pp. 628–644.
- [7] POPOV S., BAUSZAT P., FERRARI V.: Corenet: Coherent 3d scene reconstruction from a single rgb image. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16 (2020)*, Springer, pp. 366–383.
- [8] FAN H., SU H., GUIBAS L. J.: A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition (2017)*, pp. 605–613.
- [9] CHEN C., HAN Z., LIU Y.-S., ZWICKER M.: Unsupervised learning of fine structure generation for 3d point clouds by 2d projections matching. In *Proceedings of the IEEE/CVF international conference on computer vision (2021)*, pp. 12466–12477.
- [10] CHEN W., LING H., GAO J., SMITH E., LEHTINEN J., JACOBSON A., FIDLER S.: Learning to predict 3d objects with an interpolation-based differentiable renderer. *Advances in neural information processing systems* 32 (2019).
- [11] TATARCHENKO M., RICHTER S. R., RANFTL R., LI Z., KOLTUN V., BROX T.: What do single-view 3d reconstruction networks learn? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (2019)*, pp. 3405–3414.
- [12] ZHANG X., ZHANG Z., ZHANG C., TENENBAUM J., FREEMAN B., WU J.: Learning to reconstruct shapes from unseen classes. *Advances in neural information processing systems* 31 (2018).
- [13] LAINE S., HELLSTEN J., KARRAS T., SEOL Y., LEHTINEN J., AILA T.: Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–14.
- [14] ZHANG C., YU Z., ZHAO S.: Path-space differentiable rendering of participating media. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–15.
- [15] WICKRAMASINGHE U., FUA P., KNOTT G.: Deep active surface models. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)*, pp. 11647–11656. doi: 10.1109/CVPR46437.2021.01148.

- [16] WANG Y., SOLOMON J.: Fast quasi-harmonic weights for geometric data interpolation. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–15.
- [17] NICOLET B., JACOBSON A., JAKOB W.: Large steps in inverse rendering of geometry. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–13.
- [18] VICINI D., SPEIERER S., JAKOB W.: Differentiable signed distance function rendering. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–18.
- [19] LOMBARDI S., SIMON T., SARAGIH J., SCHWARTZ G., LEHRMANN A., SHEIKH Y.: Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751* (2019).
- [20] VICINI D., JAKOB W., KAPLANYAN A.: A non-exponential transmittance model for volumetric scene representations. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–16.
- [21] YIFAN W., SERENA F., WU S., ÖZTIRELI C., SORKINEHORNUNG O.: Differentiable surface splatting for point-based geometry processing. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 1–14.
- [22] RÜCKERT D., FRANKE L., STAMMINGER M.: Adop: Approximate differentiable one-pixel point rendering. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–14.
- [23] MILDENHALL B., SRINIVASAN P. P., TANCIK M., BARRON J. T., RAMAMOORTHI R., NG R.: Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM* 65, 1 (2021), 99–106.
- [24] FRIDOVICH-KEIL S., YU A., TANCIK M., CHEN Q., RECHT B., KANAZAWA A.: Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 5501–5510.
- [25] MÜLLER T., EVANS A., SCHIED C., KELLER A.: Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)* 41, 4 (2022), 1–15. 2, 4
- [26] HENDRIKX M., MEIJER S., VAN DER VELDEN J., IOSUP A.: Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 9, 1 (2013), 1–22.
- [27] FREIKNECHT J., EFFELSBURG W.: A survey on the procedural generation of virtual worlds. *Multimodal Technologies and Interaction* 1, 4 (2017), 27.
- [28] STAVA O., PIRK S., KRATT J., CHEN B., MÄCHER R., DEUSSEN O., BENES B.: Inverse procedural modelling of trees. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 118–131.
- [29] GUO J., JIANG H., BENES B., DEUSSEN O., ZHANG X., LISCHINSKI D., HUANG H.: Inverse procedural modeling of branching structures by inferring l-systems. *ACM Transactions on Graphics (TOG)* 39, 5 (2020), 1–13.
- [30] GARIFULLIN A., SHCHERBAKOV A., FROLOV V.: Fitting parameters for procedural plant generation.
- [31] JAKOB W., SPEIERER S., ROUSSEL N., NIMIER-DAVID M., VICINI D., ZELTNER T., NICOLET B., CRESPO M., LEROY V., ZHANG Z.: Mitsuba 3 renderer, 2022. <https://mitsuba-renderer.org>.
- [32] BELL B. M.: Cppad: a package for c++ algorithmic differentiation. *Computational Infrastructure for Operations Research* 57, 10 (2012).
- [33] MITCHELL M.: *An introduction to genetic algorithms*. MIT press, 1998.
- [34] NERI F., COTTA C.: Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation* 2 (2012), 1–14.