



ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК

G06T 19/00 (2020.02); *G06F 15/00* (2020.02); *H04N 13/00* (2020.02)

(21)(22) Заявка: 2019138692, 29.11.2019

(24) Дата начала отсчета срока действия патента:
29.11.2019Дата регистрации:
04.08.2020

Приоритет(ы):

(66) Номер(а) и дата(ы) подачи ранее поданной(ых)
заявки(ок): 2019116513 28.05.2019

(45) Опубликовано: 04.08.2020 Бюл. № 22

Адрес для переписки:

129090, Москва, ул. Б.Спасская, 25, строение 3,
ООО "Юридическая фирма Городисский и
Партнеры"

(72) Автор(ы):

Алиев Кара-Али Алибулатович (RU),
Ульянов Дмитрий Владимирович (RU),
Лемпицкий Виктор Сергеевич (RU)

(73) Патентообладатель(и):

САМСУНГ ЭЛЕКТРОНИКС КО., ЛТД.
(KR)(56) Список документов, цитированных в отчете
о поиске: US 2018012411 A1, 11.01.2018. US
2012127334 A1, 24.05.2012. US 2017115488 A1,
27.04.2017. AU 2016266968 A1, 23.11.2017. RU
2008110492 A, 27.09.2009.

(54) НЕЙРОННАЯ ТОЧЕЧНАЯ ГРАФИКА

(57) Реферат:

Изобретение относится к компьютерной графике. Технический результат заключается в повышении рендеринга. Способ рендеринга изображений на дисплее, заключающийся в том, что: получают облако точек с нейронными дескрипторами D для каждой точки и параметрами C камеры для облака точек в качестве входных данных; оценивают направления точек наблюдения по входным данным с помощью программного обеспечения для оценки положения и геометрии камеры; растеризуют точки облака точек посредством алгоритма z-буфера с использованием нейронных дескрипторов, объединенных с направлениями

точек наблюдения, в качестве псевдоцветов; пропускают растеризованные точки через нейронную рендерную сеть для получения конечного изображения, обучая при этом нейронные дескрипторы каждой точки и нейронную сеть; осуществляют рендеринг с применением функции потерь конечного изображения на дисплее как эталона. Изобретение может быть реализовано в любой системе, где применяется компьютерная графика (игры, VR, AR), настольном компьютере, ноутбуке, мобильном телефоне. 2 н. и 12 з.п. ф-лы, 1 табл., 7 ил.

RU 2 729 166 C1

RU 2 729 166 C1



ФИГ. 1

RU 2729166 C1

RU 2729166 C1



FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(12) **ABSTRACT OF INVENTION**

(52) CPC
G06T 19/00 (2020.02); G06F 15/00 (2020.02); H04N 13/00 (2020.02)

(21)(22) Application: **2019138692, 29.11.2019**

(24) Effective date for property rights:
29.11.2019

Registration date:
04.08.2020

Priority:
(66) Number(s) and date(s) of filing of the earlier submitted application(s): **2019116513 28.05.2019**

(45) Date of publication: **04.08.2020** Bull. № 22

Mail address:
**129090, Moskva, ul. B.Spaskaya, 25, stroenie 3,
OOO "Yuridicheskaya firma Gorodisskij i
Partnery"**

(72) Inventor(s):

**ALIEV Kara-Ali Alibulatovich (RU),
ULYANOV Dmitry Vladimirovich (RU),
LEMPITSKY Victor Sergeevich (RU)**

(73) Proprietor(s):

SAMSUNG ELECTRONICS CO., LTD. (KR)

(54) **NEURAL DOT GRAPHIC**

(57) Abstract:

FIELD: computer graphics.

SUBSTANCE: invention relates to computer graphics. A method of rendering images on a display, comprising: obtaining a point cloud with neuron descriptors D for each point and camera parameters C for the point cloud as input data; estimating observation points directions on input data using camera position and geometry estimation software; point cloud points are rasterized by z-buffer algorithm using neuron descriptors combined with directions of observation points as pseudocolours; passing the rasterised points

through the neuron render network to obtain the final image, while training neuron descriptors of each point and the neural network; performing rendering using the end image loss function on the display as a reference. Invention can be implemented in any system, where computer graphics (games, VR, AR), desktop computer, laptop, mobile telephone are used.

EFFECT: technical result consists in improvement of rendering.

14 cl, 1 tbl, 7 dwg

RU 2 729 166 C1

RU 2 729 166 C1



ФИГ. 1

RU 2729166 C1

RU 2729166 C1

Область техники, к которой относится изобретение

Изобретение относится к компьютерной графике, виртуальной реальности, дополненной реальности, более конкретно, к способу моделирования сложных сцен путем представления геометрии сцены с использованием ее облака точек.

5 Описание известного уровня техники

Конвейер разрабатывался и совершенствовался специалистами в области исследований и применения компьютерной графики на протяжении десятилетий. При контролируемых настройках этот конвейер выдает удивительно реалистичные результаты. Однако некоторые из его этапов (а значит, и весь конвейер) остаются хрупкими, часто требуют непосредственного вмешательства разработчиков и фотограмметристов и сталкиваются с проблемами при обработке некоторых классов объектов (например, тонких объектов).

Несколько групп подходов имеют своей целью упростить весь конвейер за счет исключения некоторых его этапов. Например, методы рендеринга по изображениям [15, 27, 32, 38] направлены на получение фотореалистичных изображений путем деформации исходных, полученных с камеры изображений с использованием определенных (часто очень грубых) приближений геометрии сцены. С другой стороны, в точечной графике [16, 17, 25, 28] не используется оценка поверхностной сетки, а для моделирования геометрии используется набор точек или несоединенных дисков (сурфелей). В более современных методах глубокого рендеринга [4, 5, 18, 20, 33] физический рендеринг заменяют порождающей нейронной сетью, так что глубокая рендерная нейронная сеть может исправить некоторые ошибки конвейера моделирования.

Моделирование сцены RGBD

С момента появления Kinect сенсоры RGBD активно используются для моделирования сцены благодаря сочетанию низкой стоимости и пригодности для захвата трехмерной геометрии [7, 34]. В настоящее время имеются ошибкоустойчивые алгоритмы для одновременной локализации и построения карты на основе RGBD (SLAM) [11, 23, 41, 46]. Большинство алгоритмов регистрации (SLAM), работающих с данными RGBD, создают плотное объемное представление сцены, из которого можно выделить поверхность сцены, например, с помощью алгоритма марширующих кубов [30]. Однако такая процедура оценки поверхности ограничена разрешением воксельной решетки (сетки вокселей), и обычно приводит к потере, например, тонких деталей, которые могут присутствовать в необработанных данных RGBD.

35 Поверхностные световые поля.

После появления методов рендеринга по изображениям [32, 38] было предложено несколько методов параметризации пленоптической функции [32]. Наиболее эффективным из них является метод поверхностных световых полей [47].

При этой параметризации осуществляется плотная дискретизация пленоптической функции на поверхности сцены. Конкретно, для плотного набора элементов поверхности (параметризованных с использованием координат поверхности (u, v)) регистрируется интенсивность/цвет вдоль лучей по произвольным трехмерным углам α . Совсем недавно в работе [5] был предложен глубокий вариант этой параметризации, в котором для хранения поверхностного светового поля используется полносвязная нейронная сеть, принимающая (u, v, α) в качестве ввода. Параметры этой сети обучаются на наборе данных изображений и поверхностной сетке.

Создание изображений с помощью Сверточных нейросетей.

Примерами из числа быстро растущего набора работ, в которых для создания

фотореалистичных изображений используются нейронные сети [10], являются глубокий сплаттинг [4] и глубокие поверхностные световые поля [5]. Эти работы обычно значительно выигрывают от применения машинного обучения и обработки изображений для порождающего моделирования изображений и глубокой обработки изображений, и в частности, от использования состязательного обучения [14] и перцептуальных функций потерь [9, 21] для обучения сверточных нейросетей (ConvNets) [26] выводить изображения (в отличие от, например, их классификации).

В недавних работах была продемонстрирована способность синтезировать изображения с высоким разрешением [22] и моделировать сложные преобразования изображений [20, 45] и видео [44] с использованием глубоких сверточных сетей, обученных с применением таких функций потерь. В частности, в работе [33] продемонстрировано, как можно использовать такие попиксельные сети для замены вычислительно интенсивного рендеринга и для прямого преобразования изображений с растеризованными свойствами материала и нормальными ориентациями в фотореалистичные изображения

Глубокий рендеринг на основе изображений.

В последние годы также наблюдается активное сближение рендеринга на основе изображений с глубоким обучением. В ряде работ объединяется деформирование ранее существующих фотографий и использование нейросетей для объединения деформированных изображений и/или для последующей обработки результата деформации. Эту деформацию можно оценить посредством стереосопоставления [12]. Также возможна оценка полей деформации по одному входному изображению и низкоразмерному параметру, задающему определенное движение из низкопараметрического семейства [13, 49]. В других работах деформацию осуществляют с использованием геометрии грубой сетки, которую можно получить с помощью мультиракурсного стерео [18, 43] или слияния объемных изображений RGBD [31].

С другой стороны, в некоторых методах вместо явной деформации используется некоторая форма оценки и параметризации пленоптической функции с помощью нейросетей. Как отмечалось выше, в работе [5] предложена параметризованная сетью глубокая версия поверхностных световых полей. В работе [40] применяется обучение нейронной параметризации пленоптической функции в виде низкоразмерных дескрипторов, расположенных в узлах регулярной сетки вокселей, и функции рендеринга, которая превращает репроекцию таких дескрипторов в новый вид RGB-изображения.

Наиболее близким аналогом является работа [42], в которой предлагается обучать нейронные текстуры кодированию точечной пленоптической функции в разных точках поверхности вместе с рендерной сверточной нейросетью.

Сущность изобретения

В настоящем изобретении нейронные дескрипторы элементов поверхности обучаются совместно с рендерной сетью (нейросетью). В этом методе используется точечное представление геометрии и тем самым устраняется необходимость в оценке поверхности и построении полигональной поверхности.

Предлагаемый метод напрямую выигрывает от наличия ошибкоустойчивых надежных алгоритмов RGBD SLAM/регистрации, однако он основан не на объемном моделировании сцены, а на использовании в качестве геометрической модели облака точек, собранного из необработанных сканов RGBD.

Существенное значение для настоящего изобретения имеют методы, в которых успешно применяются глубокие сверточные нейросети для задач устранения пропусков изображений [19, 29, 48]. Было предложено несколько модификаций сверточной

архитектуры, способных обрабатывать и заполнять дыры, и в изобретении используются стробированные (gated) сверточные слои из [48].

В данном изобретении авторы следуют парадигме точечной графики, так как они представляют геометрию сцены с помощью ее облака точек. Однако авторы не применяют явную оценку ориентации поверхности, или подходящих радиусов диска, или, фактически, даже цвета. Вместо этого в качестве примитива моделирования сохраняется трехмерная точка, и все локальные параметры поверхности (как фотометрические, так и геометрические) кодируются в нейронных дескрипторах, которые обучаются на данных.

Предложен способ рендеринга изображений на дисплее, заключающийся в том, что: получают облако точек с нейронными дескрипторами D для каждой точки и параметрами C камеры для облака точек в качестве входных данных; оценивают направления точек наблюдения по входным данным с помощью программного обеспечения для оценки положения геометрии камеры; обучают нейронные дескрипторы каждой точки и нейронную сеть; получают функцию потерь согласно обучению нейронной сети и дескрипторов; растеризуют точки облака точек посредством алгоритма z -буфера с использованием нейронных дескрипторов, объединенных с направлениями точек наблюдения, в качестве псевдоцветов; при этом пропускают растеризованные точки через обученную нейронную рендерную сеть для получения конечного изображения; осуществляют рендеринг с применением функции потерь конечного изображения на дисплее как эталона. При этом на этапе оценки оценивают положение и геометрию камеры с помощью программного обеспечения для камеры, такого как Agisoft Metashape или COLMAP или Open3D. Используются стандартные портативные сенсоры RGB-D для захвата необработанных данных, которые затем обрабатывают упомянутым программным обеспечением для камеры. На этапе растеризации сначала растеризуют каждую точку в квадрат с длиной стороны, обратно пропорциональной глубине точки относительно камеры, причем нейронная рендерная сеть обеспечивает процесс рендеринга, выполняемый OpenGL, без сглаживания; применяют алгоритм Z -буфера для наложения этих квадратов друг на друга с использованием их глубины относительно камеры; создают необработанное изображение каналов посредством итерации по всем наборам отпечатков и заполнения всех пикселей; используют предварительно обученную рендерную сеть с обучаемыми параметрами для преобразования необработанного изображения каналов в трехканальное изображение RGB. Облако точек получают с помощью алгоритмов, реализованных в различных приложениях, как открытых, так и коммерческих, выбранных из группы COLMAP или Agisoft Metashape. Облако точек является представлением геометрии сцены. Нейронная рендерная сеть использует глубокую сверточную нейронную сеть для создания фотореалистичных рендеров с новых точек наблюдения. Сверточную нейронную сеть используют таким образом, что выходное значение цвета в пикселе зависит от множества нейронных дескрипторов и множества точек, проецируемых в окрестности этого пикселя. нейронные дескрипторы описывают как геометрические, так и фотометрические свойства данных. В некоторых вариантах осуществления дескрипторы являются локальными дескрипторами, причем локальные дескрипторы обучаются непосредственно на данных, и это обучение происходит в координации с обучением рендерной сети. Камера представляет собой портативную RGBD-камеру. В других вариантах, облака точек реконструируют из простых потоков RGB или посредством стереосогласования.

Также предложен машиночитаемый носитель, на котором хранятся выполняемые компьютером инструкции для реализации предложенного способа.

КРАТКОЕ ОПИСАНИЕ ЧЕРТЕЖЕЙ

Описанные выше и/или другие аспекты станут более очевидными из описания примерных вариантов осуществления со ссылкой на прилагаемые чертежи.

5 Фиг. 1 (a) иллюстрирует облако точек, построенное по зарегистрированным сканам RGBD, (b) - обучение нейронных дескрипторов для каждой точки, (c) - предложенная рендерная нейросеть, которая преобразует дескрипторы растеризованных точек в реалистичные изображения.

Фиг. 2 - схематическое представление предлагаемого изобретения.

Фиг. 3 - результаты сравнения на наборе данных "Студия".

10 Фиг. 4 - результаты сравнения на наборе данных "Жилая комната" (из документа [7]) в том же формате, что и на фиг. 3.

Фиг. 5 - результаты сравнения на наборе данных "Растение" в том же формате, что и на фиг. 3.

15 Фиг. 6 - результаты сравнения на наборе данных "Ботинок" в том же формате, что и на фиг. 3.

Фиг. 7 иллюстрирует, что данную систему можно использовать для ускорения рендеринга синтетических сцен.

ОПИСАНИЕ ВАРИАНТОВ ОСУЩЕСТВЛЕНИЯ

20 Предложен новый точечный метод для моделирования сложных сцен изображений. В этом методе в качестве геометрического представления сцены используется необработанное облако точек, и каждая точка дополняется обучаемым нейронным дескриптором, в котором кодируется локальная геометрия и внешний вид сцены.

Предложенный метод объединяет несколько черт компьютерной графики, компьютерного зрения и глубокого обучения. Настоящее изобретение позволяет достичь 25 высокой реалистичности рендеринга при наличии не полностью восстановленной геометрии сцены, обеспечивает простоту и корректность моделирования сцены. Реалистичность рендеринга сцены улучшается в ситуациях с не идеально смоделированной геометрией сцены. Предложенное решение может храниться в памяти устройства или на любом подходящем носителе информации и может быть реализовано 30 в любой системе, где используется компьютерная графика (игры, VR, AR), настольном компьютере, ноутбуке, мобильном телефоне.

Предложен новый точечный метод для моделирования сложных сцен. Как и в классических точечных методах, в качестве примитивов моделирования (сурфелей) 35 предлагается использовать трехмерные точки. Каждая точка в предлагаемом методе связана с локальным дескриптором, содержащим информацию о локальной геометрии и внешнем виде сцены. Каждый дескриптор может содержать, помимо прочего, информацию о цвете точки. Вектор дескриптора можно назвать "псевдоцветом", хотя кроме цвета в нем может присутствовать любая информация. Рендерная сеть, которая преобразует растеризованные точки в реалистичные виды, принимая обученные 40 дескрипторы, обучается параллельно с этими дескрипторами.

Процесс обучения осуществляется с использованием набора данных для облаков точек и набора изображений для каждого облака точек. Облако точек получают из стороннего программного обеспечения (через видео). Каждая точка описывается тремя 45 координатами. Для каждой точки в этом облаке обучается дескриптор (8-мерный вектор), который, пройдя через нейронную сеть, превращается, например, в цвет RGB. Затем эти дескрипторы сохраняются и могут использоваться для визуализации картинки с другого ракурса, отличного от захвата видео, на котором обучались дескрипторы.

На практике дескриптор может быть M-мерным вектором, исходно этот вектор

"пустой" и в процессе обучения он заполняется информацией о локальной геометрии и/или внешнем виде сцены для каждой точки в данном облаке.

После обучения можно осуществлять подгонку предложенной модели к новым сценам и создавать с ее помощью реалистичные виды с новых точек наблюдения.

5 Примечательно, что предлагаемая система выполняет это исключительно на основе данных, не прибегая к построению полигональной поверхности или к любой другой форме явной реконструкции поверхности, а также без выполнения явной геометрической и фотометрической оценки параметров поверхности.

Основным техническим результатом изобретения является возможность
10 реконструкции сцен изображения на основе облаков точек, которые используются в качестве геометрических представителей для "геометрической модели", и представления объекта в трехмерном пространстве с помощью набора точек или полигонов при
отсутствии информации о связности; кроме того, изобретение позволяет корректно
устранять геометрический шум и дыры глубокими рендерными сетями. Также было
15 продемонстрировано, что данная модель выигрывает от предварительного обучения на множестве сцен и что можно получить хорошие результаты с помощью универсальной рендерной сети без точной подстройки к конкретной сцене.

Как показано на фиг. 2, получив облако точек P с нейронными дескрипторами D и параметрами камеры C , оценивают направления точек наблюдения, а затем точки
20 подвергают растеризации с помощью z -буфера. Можно также использовать мягкий z -буфер, отличающийся от обычного z -буфера тем, что в нем учитывается прозрачность объектов. Однако данный метод работает намного медленнее и не подходит для рендеринга в реальном времени при использовании в качестве псевдоцветов нейронных дескрипторов, связанных с направлениями точек наблюдения. Далее эти растры
25 пропускаются через рендерную сеть для получения конечного изображения. Предложенная модель подгоняется к новой сцене (сценам) посредством оптимизации параметров рендерной сети и нейронных дескрипторов с помощью обратного распространения перцептуальной функции потерь.

Глубокая рендерная сеть обучается параллельно с дескрипторами, так что, пропустив
30 растры облака точек с новых точек наблюдения через эту сеть, можно получить новые виды данной сцены. Вводимые растры используют обученные дескрипторы в качестве псевдоцветов точек. Предложенный метод можно использовать для моделирования сложных сцен и получения их фотореалистичных видов, исключив при этом явную оценку поверхности и построение сетки. В частности, убедительные результаты были
35 получены для сцены, сканированной стандартными портативными сенсорами RGB-D, а также стандартными камерами RGB, даже если на ней присутствовали объекты, представляющие сложность для стандартного моделирования на основе сетки (различные объекты с тонкой структурой, такие как листья, веревки, велосипедные колеса и т.п.). Существует ряд стандартных сенсоров RGB-D, наиболее доступными из
40 которых являются Microsoft Kinect или Intel RealSense. Эти сенсоры используются для захвата изображений RGB и глубины. Но они не способны самостоятельно оценить направления точек наблюдения, для этого существует специальная программа KinectFusion.

В создании виртуальных моделей реальных сцен обычно задействован длительный
45 конвейер операций. Такое моделирование обычно начинается с процесса сканирования, при котором фотометрические свойства регистрируются с использованием изображений с камеры, а необработанная геометрия сцены регистрируется сканерами глубины или посредством плотного стереосопоставления. В последнем случае обычно получается

зашумленное и неполное облако точек, требующее дальнейшей обработки с применением определенных методов реконструкции поверхности и построения полигональной поверхности. При наличии полигональной поверхности в процессах текстурирования и оценки материала определяются фотометрические свойства фрагментов поверхности, которые сохраняются в виде двумерных параметризованных карт, таких как карты текстуры [3], карты рельефа [2], зависимой от вида текстуры [8], поверхностные световых полей [47]. И наконец, создание фотореалистичных видов моделируемой сцены включает в себя вычислительно сложный процесс рендеринга, такой как трассировка лучей и/или оценка переноса излучения. Получив облако точек, построенное по зарегистрированным сканам RGBD (фиг. 1a), предлагаемая система обучает нейронные дескрипторы для каждой точки (первые три измерения PCA показаны в ложном цвете на фиг. 1b) и нейронную рендерную сеть, которая преобразует дескрипторы растеризованных точек в реалистичные изображения (фиг. 1c). В необработанных облаках точек, собранных пользовательскими камерами RGBD (фиг. 1a), неизбежны разрывы в геометрии, геометрический шум и выбросы точек, например, как в сцене из набора данных ScanNet. Предложенный метод позволяет корректно решать эти проблемы и синтезировать реалистичные рендеры, несмотря на присутствие этих проблем.

В настоящем изобретении предложена система (метод), позволяющая исключить большинство этапов классического конвейера. Она объединяет идеи рендеринга по изображениям, точечную графику и нейронный рендеринг в простой метод. В этом методе в качестве представления геометрии сцены используется необработанное облако точек, что исключает необходимость в оценке поверхности и построении полигональной поверхности. Подобно другим методом нейронного рендеринга, в предлагаемом методе также используется глубокая сверточная нейросеть для создания фотореалистичных рендеров с новых точек наблюдения. Реализм такого рендеринга улучшается за счет оценки скрытых векторов (нейронных дескрипторов), которые описывают как геометрические, так и фотометрические свойства данных. Локальные дескрипторы обучаются непосредственно на данных, и это обучение происходит в координации с обучением рендерной сети (см. фиг. 1). Дескрипторы обучаются методом градиентного спуска параллельно с нейронной сетью, которая обучается интерпретировать эти дескрипторы. Обучение нейронной сети и дескрипторов вырабатывает функцию потерь, указывающую, как следует изменить нейронную сеть и дескрипторы, чтобы они вместе создавали желаемое изображение. Иными словами, перцептуальная функция поверхности принимается в качестве целевой функции, и значения параметров дескриптора определяются при нахождении экстремума целевой функции с помощью метода градиентного спуска. Тем не менее, можно использовать любой метод оптимизации.

Предложенный метод позволяет моделировать и получать рендеры сцен, снятых портативными камерами RGBD, а также простых потоков RGB. На этапе подготовки данных из последовательности rgb или rgb-d (видео или множество фотографий, снятых с разных ракурсов) восстанавливается облако точек. Этот процесс связан с оценкой направления точки наблюдения, то есть необходимо понять, под каким углом следует смотреть на данное облако точек, чтобы оно появилось на месте объекта, изображенного на фотографии.

Был выполнен ряд сравнений с абляциями и конкурирующими методами, продемонстрировавшими возможности и преимущества предложенного способа. В целом, результаты предполагают, что при мощности современных глубоких сетей простейшие трехмерные примитивы (то есть трехмерные точки) являются достаточными и наиболее

подходящими геометрическими представителями (другим термином, соответствующим "геометрическому представителю", является "геометрическая модель", способ представления объекта в трехмерном пространстве с помощью набора точек или полигонов) для нейронного рендеринга.

5 Предлагаемый метод связан с методами, основанными на поверхностных световых полях, поскольку он неявно обучает параметризации точечной пленоптической функции на поверхности сцены в нейронных дескрипторах. В отличие от поверхностных световых полей, предлагаемый метод не требует моделирования поверхности сцены. Также, в отличие от работы [5], в которой значение цвета выводится независимо в каждой
10 вершине поверхности, в предлагаемом подходе для рендеринга используется сверточная нейронная сеть, так что выходное значение цвета в пикселе зависит от нескольких нейронных дескрипторов и нескольких точек, спроецированных в окрестности этого пикселя.

Предлагаемый процесс рендеринга заключается в следующем. Предположим, что
15 дано облако точек (набор 3D точек) $P = \{p_1, p_2, \dots, p_N\}$ (позиция 1 на фиг. 2) с M-мерными нейронными дескрипторами $D = \{d_1, d_2, \dots, d_N\}$ (позиция 2 на фиг. 2), где каждое d_i представляет M-мерный нейронный дескриптор для соответствующей
20 точки облака, и необходимо получить его рендер из нового вида, охарактеризованного камерой C (включая как внешние, так и внутренние параметры). В частности, предположим, что целевое изображение имеет пиксельную сетку размера $W \times H$, и что его точка наблюдения расположена в точке C.

Рендеринг облака точек - это рисование точек в качестве пикселей на изображении.
25 Новая точка наблюдения означает точку наблюдения, отсутствующую на тренировочных изображениях.

Термин "камера" широко используется в компьютерной графике для обозначения точки наблюдения, размера изображения, фокусного расстояния и других оптических параметров реальной камеры. Для рендеринга трехмерного объекта нужны все эти
30 параметры, так же как для фотографирования нужна оптика. Целевое изображение - это изображение из последовательности тренировочных изображений, которое необходимо реконструировать с помощью предложенной нейросети и дескрипторов точек. Рендерная сеть - это сеть, которая принимает рендерное облако точек в качестве
35 ввода и пытается перерисовать его таким образом, чтобы оно выглядело как соответствующее целевое изображение.

Как показано на фиг. 2, облако точек представляет собой набор трехмерных точек, связанных с нейронными дескрипторами (M-мерными векторами) и направлениями
40 точек наблюдения. В данном контексте, под направлением точки наблюдения подразумевается вектор, указывающий из положения камеры в какую-то точку в облаке. Камера определяет, с какого направления следует смотреть на облако точек, чтобы оно совпало с соответствующим целевым изображением. Получив облако точек и камеру, точки подвергаются растеризации (или рендерингу) с помощью z-буфера. Z-буфер оставляет только самые передние точки. Затем это изображение поступает в нейронную
45 сеть, которая преобразует данный ввод в RGB-изображение, выглядящее как эталон (целевое изображение). Функция потерь - это функция, которая измеряет подобие между изображением из нейронной сети и эталоном.

Процесс рендеринга начинается с растеризации каждой точки p_i в квадрат с длиной стороны, обратно пропорциональной глубине точки относительно камеры C. Рендеринг

осуществляют с помощью OpenGL без сглаживания, так что размеры каждого квадрата по существу округляются до ближайших целых. Алгоритм Z-буфера (фиг. 2) применяется для наложения этих квадратов друг на друга с использованием их глубины относительно камеры. Пусть $f_i(C)$ обозначает набор "отпечатков" точки s_i , полученный в результате

такого рендеринга, то есть набор пикселей, которые заняты растром i -го квадрата после z-буфера. Затем создается необработанное $(M+3)$ -канальное изображение $S(P, D, C)$ путем итерации по всем наборам отпечатков $f_i(C)$ и заполнения всех пикселей из $s_i(C)$

значениями d_i (первые M каналов), как показано на фиг. 2 (растеризованные точки). Последние три канала устанавливаются на координаты нормализованного вектора

направления точек наблюдения $v_i = \frac{p_0 - p_i}{\|p_0 - p_i\|}$. Таким образом, пиксели (x, y)

необработанного изображения заполняются следующим образом:

$$\forall (x, y) \in s_i : S(P, D, C)[x, y] = \{d_i; v_i\}, \quad (1)$$

где $\{d_i, v_i\}$ обозначает конкатенацию, а $[x, y]$ - векторную запись необработанного изображения, соответствующую пикселю (x, y) . Конкатенация информации о локальной поверхности, закодированной в d_i , с направлением точки наблюдения v_i позволяет предлагаемой системе смоделировать фотометрические эффекты, зависящие от вида, а также заполнить дыры в сети облака точек с учетом ориентации поверхности относительно вектора направления точки наблюдения. Пиксели, не покрытые отпечатком, получают специальное значение дескриптора $d_0 \in \mathbb{R}^M$ (которое также обучается для конкретной сцены), а их размеры направления точки наблюдения устанавливаются на нуль.

И, наконец, применяется предварительно обученная рендерная сеть. На стадии предобучения берут 52 скана из Scannet (<http://www.scan-net.org/>) и обучают сеть осуществлять рендеринг этих сцен. На этом этапе сеть обучается интерпретировать дескрипторы точек, которые обучаются вместе с сетью.

Обучение на новой сцене - берут предварительно обученную сеть и обучают ее дальше на новой сцене, которую сеть не "видела" ранее. Использование предварительно обученной сети повышает качество рендеринга.

\mathcal{R}_θ с обучаемыми параметрами θ используют для преобразования необработанного $(M+3)$ -канального изображения $S(P, D, C)$ в трехканальное RGB-изображение I :

$$I(P, D, C, \theta) = \mathcal{R}_\theta(S(P, D, C)). \quad (2)$$

В предлагаемом случае рендерная сеть имеет обычную сверточную архитектуру U-Net [36] со стробированными свертками [48].

Процесс обучения в предлагаемой системе.

Нейронную сеть вместе с дескрипторами точек обучают по двум причинам: во-первых, требуется нейронная сеть, способная интерпретировать дескрипторы точек. Во-вторых, требуется обучить дескрипторы для конкретной сцены или объекта, чтобы осуществлять ее рендеринг с помощью нейронной сети.

Предполагается, что во время обучения имеется K тренировочных сцен. Для k -й сцены дано облако точек p^k , а также набор из L_k тренировочных эталонных RGB

изображений $\mathbf{I}^k = \{I^{k,1}, I^{k,2}, \dots, I^{k,L_k}\}$ с известными параметрами
камеры $\{C^{k,1}, C^{k,2}, \dots, C^{k,L_k}\}$. Тогда цель \mathcal{L} обучения равна
5 несоответствию между рендерным и эталонным RGB изображением:

$$\mathcal{L}(\theta, \mathbf{D}^1, \mathbf{D}^2, \dots, \mathbf{D}^K) = \sum_{k=1}^K \sum_{l=1}^{L_k} \Delta \left(\mathcal{R}_\theta \left(S(\mathbf{P}^k, \mathbf{D}^k, C^{k,l}) \right), I^{k,l} \right), \quad (3)$$

где \mathbf{D}^k - набор нейронных дескрипторов для облака точек k-й сцены, а Δ -
несоответствие между двумя изображениями (эталонным и рендерным). В предложенной
реализации используется перцептуальная функция потерь [9, 21], которая вычисляет
15 несоответствие между активациями предварительно обученной сети VGG [39].

Обучение выполняется путем оптимизации потери (уравнение (3)) как на параметрах
 θ рендерной сети, так и на нейронных дескрипторах $\{\mathbf{D}^1, \mathbf{D}^2, \dots, \mathbf{D}^K\}$ точек
в тренировочном наборе сцен. Следовательно, в предлагаемом методе нейронные
20 дескрипторы обучаются непосредственно на данных. Оптимизация осуществляется по
алгоритму ADAM [24]. При этом нейронные дескрипторы обновляются путем обратного
распространения через (1) производных потерь относительно $S(\mathbf{P}, \mathbf{D}, \mathbf{C})$ на d_i .

Моделирование новых сцен.

После выполнения обучения (3) предложенная система может смоделировать новую
25 сцену, имея ее облако точек и набор видов RGB, зарегистрированных с этим облаком
точек. Например, в случае сцены, просканированной камерой RGBD,
зарегистрированные виды RGBD могут обеспечивать как облако точек, так и
изображения RGB.

Для новой сцены, имея облако точек \mathbf{P}' и набор изображений
30 $\mathbf{I}' = \{I'^1, I'^2, \dots, I'^{L'}\}$ с параметрами камеры $\{C'^1, C'^2, \dots, C'^{L'}\}$,
нейронные дескрипторы $\mathbf{D}' = \{d'_1, d'_2, \dots, d'_{N'}\}$ новой сцены обучаются, с
сохранением при этом фиксированных параметров θ , путем оптимизации цели \mathcal{L}' :
35

$$\mathcal{L}'(\mathbf{D}') = \sum_{l=1}^{L'} \Delta \left(\mathcal{R}_\theta \left(S(\mathbf{P}', \mathbf{D}', C'^l) \right), I'^l \right). \quad (4)$$

40 Благодаря совместному использованию параметров θ рендеринга в обучающей сцене
и новой сцене предложенная система получает способность к лучшему обобщению,
что приводит к лучшему синтезу нового вида.

В качестве альтернативы, вместо того, чтобы сохранять параметры θ рендерной
сети фиксированными, предлагается подстраивать их к новой сцене, используя в качестве
45 инициализаций предварительно обученные значения. При такой тонкой подстройке
наблюдается некоторое улучшение качества рендеринга новых видов для некоторых
сцен. Однако в практических системах может быть желательным сохранять
совместимость рендерной сети для множества сцен (т.е. иметь универсальную рендерную

сеть).

Детали эксперимента.

Эта модель основана на популярной архитектуре U-Net, содержащей четыре блока понижающей и повышающей дискретизации, слои повышающей дискретизации по максимуму, слои повышающей дискретизации по среднему значению и транспонированные свертки со слоями билинейной повышающей дискретизации. Было замечено, что стробированные свертки повышают производительность модели на разреженных входных данных, поэтому в предлагаемой модели нормальные свертки заменяются стробированными свертками. Поскольку в качестве предлагаемой рендерной сети используется U-Net и обучение множества признаков точек осуществляется отдельно, оказалось, что можно использовать облегченную сеть с меньшим количеством параметров. Предлагаемая модель имеет в четыре раза меньше каналов в каждом сверточном слое, чем в исходной архитектуре, что дает 1,96 млн. параметров. Это позволяет осуществить рендеринг изображения 1296×968 на GeForce RTX 2080 Ti в режиме реального времени за 50 мс.

Чтобы продемонстрировать универсальность данного метода, его оценивали на нескольких типах сцен. Представляет интерес захват реальных сцен с помощью недорогих пользовательских устройств. Поэтому рассматривались два типа захвата. Во-первых, рассматривались потоки RGBD из набора данных ScanNet для сцен в помещении, отсканированных сенсором RGBD со структурной подсветкой. Во-вторых, рассматривался поток видео RGB, снятый смартфоном. И наконец, была продемонстрирована применимость предлагаемого метода для моделирования фотометрически сложных синтетических сцен на примере его выполнения на стандартной тестовой сцене из программного пакета Blender.

Для сцен ScanNet использовались предоставленные зарегистрированные данные, полученные с набором данных BundleFusion. Использовалась геометрия сетки, вычисленная BundleFusion в соответствующих базовых версиях. Имея зарегистрированные данные, получили облака точек путем соединения трехмерных точек из всех кадров RGBD и использования объемной субдискретизации (с шагом сетки 1 см), в результате чего полученные облака точек содержали несколько миллионов точек на одну сцену.

При оценке использовались две сцены ScanNet - "Студия" (сцена 0), содержащая 5578 кадров, и "Жилая комната" (сцена 24), содержащая 3300 кадров. В каждом случае каждый сотый кадр в траектории использовался для проверки правильности. Затем были удалены кадры в пределах 20 временных шагов от каждого из этих проверочных кадров из подгоночного набора, а оставшиеся 3303 и 2007 кадров соответственно использовались для подгонки (тонкой настройки) и оценки дескриптора. Предварительное обучение заключалось в выполнении сетью рендеринга 52 сцен (предварительно обработанных аналогичным образом), которые не содержали сцены "Студия" и "Жилая комната".

Для сцен, снятых смартфоном, авторы изобретения использовали коммерческий пакет Agisoft Metashape (<https://www.agisoft.com/>, найдено 20.05.2019), один из лучших пакетов для моделирования/реконструкции сцены. Agisoft Metashape обеспечивает регистрацию, создание облака точек и построение сетки посредством запатентованных методов создания структуры и движения и многовидового стерео представления. Оценивались две сцены: "Ботинок" и "Растение". Сцена "Растение" содержала 2727 кадров, снятых с интервалами 250 мс, из которых каждый 50-й кадр помещался в набор проверки правильности, а 10 кадров вокруг этих кадров отбрасывались, остальные

кадры использовались в качестве подгоночного набора. Сцена "Ботинок" была взята преднамеренно как содержащая очень малое число изображений - 100 кадров, снятых с интервалами 250 мс, которые были перемешаны, и 10 кадров было отобрано для проверки.

5 Сравнилось несколько методов оценки сцен. Большинство из этих методов используют рендерную сеть как в предлагаемом способе, которая принимает промежуточное представление и затем обучается выводить конечное изображение RGB. Если не указано иное, для всех методов использовалась описанная выше сеть с параметрами 1,96М.

10 Адаптированный вариант. В этом варианте предложенной системы рендерную сеть и пространство дескрипторов предварительно обучают на 52 сценах ScanNet. Затем обучаются нейронные дескрипторы и выполняется точная подстройка (адаптация) рендерной сети по подгоночной части оценочной сцены. Такая тонкая подстройка сходится через 30 эпох (от 8 минут до 1,5 часов на 4x NVIDIA Tesla V-100 в зависимости от размера сцены).

15 Универсальный вариант. В этом варианте выполняется то же самое, что и выше. Однако рендерная сеть не подвергается тонкой подстройке к оценочной схеме и остается фиксированной, а обучаются нейронные дескрипторы точек. Сохранение рендерной сети "универсальной", то есть неадаптированной к конкретной сцене, может быть более
20 целесообразным во многих сценариях. Такое обучение сходится через 20 эпох (от 5 минут до 1 часа на 4x NVIDIA Tesla V-100 в зависимости от размера сцены).

Сцена. Этот вариант не предусматривает предварительное обучение рендерной сети, а обучает ее только на оценочной сцене (ее подогнанной части) вместе с дескрипторами точек. Естественно, что этот метод более подвержен переобучению. Такое обучение
25 сходится через 50 эпох (от 12 минут до 2,5 часов на 4x NVIDIA Tesla V-100 в зависимости от размера сцены).

Полигональная поверхность+Текстура. В этой базовой версии, получив полигональную поверхность сцены с помощью BundleFusion или Metashape, обучается текстура через обратное распространение той же функции потерь, которая используется
30 в предлагаемом методе, посредством процесса построения карты текстуры. Это приводит к "классическому" представлению сцены текстурированной полигональной поверхностью.

Сетка+RenderNet. В этом варианте (аналогично, например, Lookin-Good) дополнительно обучается рендерная сеть, которая преобразует растры
35 текстурированной сетки в конечные изображения RGB. Эта рендерная сеть имеет ту же архитектуру, как и предложенная (за исключением того, что вход имеет три канала), и в обучении используется та же функция потерь, что и в изобретении.

Прямой Сетевой Рендер (Direct RenderNet). В этом варианте оценивается абляция предложенной точечной системы без нейронных дескрипторов. В данном случае
40 обучается рендерная сеть, которая строит карту облака точек, растеризованного так же, как в предлагаемом методе. Однако вместо нейронных дескрипторов используются цвет точки (взятый из исходного изображения RGBD скана/RGB изображения), трехмерная координата точки и направление точки наблюдения v_i в качестве 9D псевдоцвета. Затем рендерную сеть обучают с той же потерей, что и в предложенной
45 сети. Рендерная сеть также предварительно обучается на наборе из 52 сцен.

Direct RenderNet (медленная). Было замечено, что описанный выше вариант Direct RenderNet значительно выигрывает при использовании более производительной и более медленной рендерной сети. Поэтому оценивался вариант с рендерной сетью, имеющей

удвоенное число каналов во всех промежуточных слоях (в результате получается 4x параметров, 4x FLOP).

Авторы изобретения также приложили значительные усилия, чтобы адаптировать метод поверхностных световых полей к предложенным данным. Тем не менее, редко наблюдалось какое-либо улучшение по сравнению с вариантом "полигональная поверхность+текстура", и в среднем результаты по данным контрольной выборки были хуже. По-видимому, оценка поверхностного светового поля не подходит для случаев с грубой геометрией поверхности.

Результаты сравнения

В таблице 1 представлены количественные результаты сравнения.

Способ	Студия (RGBD поток) Потеря↓PSNR↑SSIM↑	Жилая комната (RGBD поток) Потеря↓PSNR↑SSIM↑	Ботинок (RGB видео) Потеря↓PSNR SSIM↑	Растение (RGB видео) Потеря↓PSNR↑ SSIM↑	Параметры RenderNet
15 Полигональная поверхность+текстура	551.193 18.372 0.821	678.957 15.373 0.764	378.502 20.224 0.852	482.707 18.822 0.746	- 1.96M
Полигональная текстура+RenderNet	539.210 19.180 0.840	581.367 17.536 0.819	280.242 24.062 0.910	375.322 21.846 0.817	1.96 M
Сцена (изобретение)	523.287 19.615 0.811	545.406 18.486 0.828	284.617 23.851 0.888	340.275 22.674 0.817	
*Direct RenderNet	528.705 18.185 0.819	557.513 16.840 0.793	265.722 22.552 0.890	363.445 21.888 0.801	
*Direct RenderNet (медлен)	518.310 19.814 0.850	534.648 18.834 0.840	260.530 25.936 0.920	345.232 23.334 0.837	1.96M
*Изобретение полностью	508.182 19.899 0.852	525.016 18.888 0.842	260.981 25.330 0.917	323.483 23.655 0.844	7.84M
*Изобретение универсал.	510.135 19.900 0.863	525 043 18.894 0.842	259.528 25.524 0.917	323.483 23.655 0.844	

Таблица I Результаты сравнения по показателям потери восприятия (чем ниже, тем лучше), PSNR (чем выше, тем лучше), SSIM (чем выше, тем лучше). Методы, отмеченные *, обучались на наборе контрольной выборки данных сцены. См. описания методов по тексту. В большинстве случаев варианты предложенного метода превосходят базовые версии.

Измерения для всех сравнений выполнялись на подмножествах проверки правильности, для чего сравнивались полученные и эталонные RGB-изображения. Показано значение потерь на этих подмножествах (следует отметить, что это сравнение является достоверным, поскольку в большинстве методов оптимизируются одинаковые потери на обучающем наборе). Также показано пиковое соотношение сигнал/шум (PSNR) и мера самоподобия (SSIM). Кроме того, на фиг. 3-6 показано качественное сравнение кадров набора проверки правильности, где также показано облако точек.

В общем, как количественное, так и качественное сравнение показали преимущество использования облака точек в качестве геометрического представления. Следовательно, методы "полигональная поверхность+текстура" и "полигональная поверхность+RenderNet" работают хуже, чем все методы, в которых используются облака точек. Исключением является сцена "Ботинок", где процедура создания полигональной поверхности была успешной благодаря построению достаточно хорошей полигональной поверхности. Во всех других сценах имеются части сцены, в которых процесс построения поверхности (BundleFusion или Metashape) не удался, что привело к грубым ошибкам в рендерах. Качественное сравнение выявило сбои, которые особенно заметны на тонких объектах (например, детали велосипеда на фиг. 3 или листьях растения на фиг. 5).

На фиг. 3 показаны результаты сравнения на наборе данных "Студия". Показаны текстурированная полигональная поверхность, цветное облако точек, результаты трех нейронных рендерных систем и эталон. Предлагаемая система способна успешно воспроизводить детали, которые представляют сложность для построения сетки, и меньше подвержена размытости, чем система Direct RenderNet. Слева направо, сверху вниз показаны: 1) сетка+текстура (см. таблицу 1) облако точек с цветами, полученными из BundleFusion, 3) Direct RenderNet (см. таблицу 1), сетка+RenderNet (см. таблицу 1), 5)

изобретение полностью (см. таблицу 1) и изображение, полученное от сенсора RGB ("эталон")

На фиг. 4 показаны результаты сравнения для сцены "Гостиная" из набора данных ScanNet.

5 На фиг. 6 показаны результаты сравнения для набора данных "Ботинок" в том же формате, что и на фиг. 3. В отличие от трех других наборов данных, геометрия этой сцены лучше подходила для представления посредством полигональной поверхности, и рендер на основе полигональной поверхности получился относительно хорошо. Предложенный метод и в этом случае превосходит базовый вариант Direct RenderNet.

10 Предлагаемая система на основе нейронных дескрипторов точки обычно превосходит прямую абляцию RenderNet, которая не имеет таких дескрипторов. Предлагаемые проверочные кадры не слишком далеки от подгоночного набора, и замечено, что качественное различие методов возрастает с перемещением камеры дальше от камер подгоночных наборов. Этот эффект можно наблюдать в дополнительном видео. Обычно
15 у предложенного метода качество отдельных кадров для таких положений камеры значительно выше, чем у базового варианта Direct (который подвержен размытости и потере деталей). В то же время, предположительно, это значительное улучшение качества отдельных кадров происходит за счет увеличения временного мерцания изображения.

Результаты по синтетическим данным

20 Была продемонстрирована способность предлагаемого метода моделировать синтетические сцены с чрезвычайно сложными фотометрическими свойствами (фиг. 7). На фиг. 7 показано, что эту систему можно использовать для ускорения рендеринга синтетических сцен. Здесь представлены рендеры стандартной тестовой сцены Blender с использованием предложенной системы (третий столбец). В четвертом столбце показан
25 ближайший кадр из набора данных кадров, используемых для подгонки модели. Хотя предложенная система не имеет точного совпадения с результатом рендеринга методом трассировки лучей, ей удалось воспроизвести некоторые детали в зеркальном отражении и мелкие детали текстуры, причем это выполнялось в режиме реального времени.

В данном случае использование предложенного метода может быть оправдано в
30 качестве средства ускорения рендеринга. Для этого взяли стандартную тестовую сцену Blender [2] со сложным освещением и сильно отражающим объектом в центре, произвели выборку облака точек (2,5 миллиона точек) с его поверхности и обучили нейронные дескрипторы и рендерные сети на 200 случайных видах сцены. Сравнение предлагаемых рендеров с синтетическими рендерами "эталона", полученными методом трассировки
35 лучей в Blender, показало очень близкое совпадение (фиг. 7). В то время как Blender требует около 2 минут для рендеринга одного кадра этой сцены на двух GeForce RTX 2080 Ti (при максимальной установке качества), предлагаемые рендеры были получены со скоростью 50 мс (20 кадров в секунду) на одной GeForce RTX 2080Ti. Следует
40 отметить, что при наличии хорошей полигональной поверхности для данной сцены методы нейронного рендеринга на основе сетки, вероятно, тоже хорошо справятся с этой задачей.

Таким образом, предложен нейронный точечный метод моделирования сложных сцен. Как и в классических точечных методах, в качестве примитивов моделирования
45 используются 3D-точки. Каждая из точек в предлагаемом методе связана с локальным дескриптором, содержащим информацию о локальной геометрии и внешнем виде. Рендерная сеть, которая преобразует растры точек в реалистичные виды, принимая обученные дескрипторы в качестве ввода псевдоцветов точки, обучается параллельно с самими дескрипторами.

Процесс обучения осуществляется с использованием набора данных облаков точек и изображений. После обучения предложенную модель можно подгонять к новым сценам и создавать реалистичные виды с новых точек наблюдения.

Примечательно то, что предлагаемая система выполняет все это исключительно на основе данных, не прибегая к построению сетки или к любой другой форме явной реконструкции поверхности, а также, не прибегая к явной оценке геометрических и фотометрических параметров поверхности.

Основной вклад состоит в демонстрации того, что облака точек можно успешно использовать в качестве геометрических представителей для нейронного рендеринга, а проблемы недостающей информации о связности, а также геометрических шумов и дыр можно корректно решать с помощью глубоких рендерных сетей.

Предварительное обучение на множестве сцен благоприятно влияет на данную модель и позволяет получить хорошие результаты с универсальной рендерной сетью, которая не была точно настроена для конкретной сцены.

Ограничения и улучшения. На сегодняшний день предложенная модель не может реалистично заполнить очень большие дыры в геометрии. Такую возможность, скорее всего, можно будет получить дополнительной обработкой/отрисовкой облака точек, которой потенциально можно обучить вместе с предлагаемым конвейером моделирования. Изучается производительность системы для динамических сцен, где может потребоваться введение некоторого механизма обновления для нейронных дескрипторов точек.

ЛИТЕРАТУРА

- [1] Blender Online Community. retrieved 20.05.2019. Blender - a 3D modelling and rendering package. Blender Foundation, Blender Institute, Amsterdam. <http://www.blender.org>
- [2] James F Blinn. 1978. Simulation of wrinkled surfaces. In Proc. SIGGRAPH, Vol. 12. ACM, 286-292.
- [3] James F Blinn and Martin E Newell. 1976. Texture and reflection in computer generated images. Commun. ACM 19, 10 (1976), 542-547.
- [4] Giang Bui, Truc Le, Brittany Morago, and Ye Duan. 2018. Point-based rendering enhancement via deep learning. The Visual Computer 34, 6-8 (2018), 829-841.
- [5] Anpei Chen, Minye Wu, Yingliang Zhang, Nianyi Li, Jie Lu, Shenghua Gao, and Jingyi Yu. 2018. Deep Surface Light Fields. Proceedings of the ACM on Computer Graphics and Interactive Techniques 1, 1 (2018), 14.
- [6] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. 2017. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In Proc. CVPR.
- [7] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. 2017. BundleFusion: Real-Time Globally Consistent 3D Reconstruction Using On-the-Fly Surface Reintegration. ACM Trans. Graph. 36, 3 (2017), 24:1-24:18.
- [8] Paul Debevec, Yizhou Yu, and George Borshukov. 1998. Efficient view-dependent image-based rendering with projective texture-mapping. In Rendering Techniques. Springer, 105-116.
- [9] Alexey Dosovitskiy and Thomas Brox. 2016. Generating Images with Perceptual Similarity Metrics based on Deep Networks. In Proc. NIPS. 658-666.
- [10] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. 2015. Learning to generate chairs with convolutional neural networks. In Proc. CVPR. 1538-1546.
- [11] Felix Endres, Jürgen Hess, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. 2014. 3-D mapping with an RGB-D camera. IEEE transactions on robotics 30, 1 (2014), 177-187.
- [12] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. 2016. Deepstereo: Learning

to predict new views from the world's imagery. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 5515-5524.

[13] Yaroslav Ganin, Daniil Kononenko, Diana Sungatullina, and Victor S. Lempitsky. 2016. DeepWarp: Photorealistic Image Resynthesis for Gaze Manipulation. In Proc. ECCV. 311-326.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In Proc. NIPS. 2672-2680.

[15] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. 1996. The Lumigraph. In SIGGRAPH. ACM, 43-54.

[16] Markus Gross, Hanspeter Pfister, Marc Alexa, Mark Pauly, Marc Stamminger, and Matthias Zwicker. 2002. Point based computer graphics. Eurographics Assoc.

[17] Jeffrey P Grossman and William J Dally. 1998. Point sample rendering. In Rendering Techniques'98. Springer, 181-192.

[18] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel J. Brostow. 2018. Deep blending for free-viewpoint image-based rendering. ACM Trans. Graph. 37, 6 (2018), 257:1-257:15.

[19] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2017. Globally and Locally Consistent Image Completion. ACM Transactions on Graphics (Proc. of SIGGRAPH 2017) 36, 4, Article 107 (2017), 107:1-107:14 pages.

[20] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Imagenet- Image Translation with Conditional Adversarial Networks. In Proc. CVPR. 5967-5976.

[21] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In Proc. ECCV. 694-711.

[22] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2018. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In International Conference on Learning Representations.

[23] Christian Kerl, Jürgen Sturm, and Daniel Cremers. 2013. Dense visual SLAM for RGB-D cameras. In Proc. IROS. IEEE, 2100-2106.

[24] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. CoRR abs/1412.6980 (2014). arXiv:1412.6980

[25] Leif Kobbelt and Mario Botsch. 2004. A survey of point-based techniques in computer graphics. Computers & Graphics 28, 6 (2004), 801-814.

[26] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. 1989. Backpropagation applied to handwritten zip code recognition. Neural computation 1, 4 (1989), 541-551.

[27] Marc Levoy and Pat Hanrahan. 1996. Light field rendering. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. ACM, 31-42.

[28] Marc Levoy and Turner Whitted. 1985. The use of points as a display primitive. Citeseer.

[29] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. 2018. Image inpainting for irregular holes using partial convolutions. In Proceedings of the European Conference on Computer Vision (ECCV). 85-100.

[30] William E Lorensen and Harvey E Cline. 1987. Marching cubes: A high resolution 3D surface construction algorithm. In Proc. SIGGRAPH, Vol. 21. 163-169.

[31] Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien Valentin, Sameh Khamis, Philip Davidson, Anastasia Tkach, Peter Lincoln, et al. 2018. LookinGood: enhancing performance capture with real-time neural re-rendering. In SIGGRAPH Asia 2018 Technical Papers. ACM, 255.

[32] Leonard McMillan and Gary Bishop. 1995. Plenoptic modeling: an image-based rendering

system. In SIGGRAPH. ACM, 39-46.

[33] Oliver Nalbach, Elena Arabadzhiyska, Dushyant Mehta, Hans-Peter Seidel, and Tobias Ritschel. 2017. Deep Shading: Convolutional Neural Networks for Screen Space Shading. *Comput. Graph. Forum* 36, 4 (2017), 65-78.

5 [34] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew W. Fitzgibbon. 2011. KinectFusion: Real-time dense surface mapping and tracking. In ISMAR. IEEE Computer Society, 127-136.

[35] Hanspeter Pfister, Matthias Zwicker, Jeroen Van Baar, and Markus Gross. 2000. Surfels: 10 Surface elements as rendering primitives. In Proceedings of the 27th annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co., 335-342.

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks 15 for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention. Springer, 234-241.

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *CoRR abs/1505.04597* (2015). arXiv:1505.04597 <http://arxiv.org/abs/1505.04597>

[38] Steven M Seitz and Charles R Dyer. 1996. View morphing. In Proceedings of the 23rd 20 annual conference on Computer graphics and interactive techniques. ACM, 21-30.

[39] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR abs/1409.1556* (2014). arXiv:1409.1556 <http://arxiv.org/abs/1409.1556>

25 [40] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. 2019. DeepVoxels: Learning Persistent 3D Feature Embeddings. In Proc. CVPR.

[41] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. 2012. A benchmark for the evaluation of RGB-D SLAM systems. In Proc. IROS. IEEE, 573-580.

[42] Justus Thies, Michael Zollhöfer, and Matthias Nießner. 2019. Deferred Neural Rendering: 30 Image Synthesis using Neural Textures. In Proc. SIGGRAPH.

[43] J. Thies, M. Zollhöfer, C. Theobalt, M. Stamminger, and M. Nießner. 2018. IGNOR: Image-guided Neural Object Rendering. arXiv 2018 (2018).

[44] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. Video-to-Video Synthesis. In Proc. NIPS.

35 [45] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In Proc. CVPR.

[46] Thomas Whelan, Michael Kaess, Hordur Johannsson, Maurice Fallon, John J Leonard, and John McDonald. 2015. Real-time large-scale dense RGB-D SLAM with volumetric fusion. 40 *The International Journal of Robotics Research* 34, 4-5(2015), 598-626.

[47] Daniel N Wood, Daniel I Azuma, Ken Aldinger, Brian Curless, Tom Duchamp, David H Salesin, and Werner Stuetzle. 2000. Surface light fields for 3D photography. In Proc. SIGGRAPH. 287-296.

[48] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. 2018. Free- 45 Form Image Inpainting with Gated Convolution. arXiv preprint arXiv:1806.03589 (2018).

[49] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. 2016. View synthesis by appearance flow. In Proc. ECCV. 286-301.

[50] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. 2001. Surface

splatting. In Proc. SIGGRAPH. ACM, 371-378.

(57) Формула изобретения

1. Способ рендеринга изображений на дисплее, заключающийся в том, что:
 - получают облако точек с нейронными дескрипторами D для каждой точки и параметрами C камеры для облака точек в качестве входных данных;
 - оценивают направления точек наблюдения по входным данным с помощью программного обеспечения для оценки положения геометрии камеры;
 - растеризуют точки облака точек посредством алгоритма z -буфера с использованием нейронных дескрипторов, объединенных с направлениями точек наблюдения, в качестве псевдоцветов;
 - пропускают растеризованные точки через нейронную рендерную сеть для получения конечного изображения, обучая при этом нейронные дескрипторы каждой точки и нейронную сеть;
 - осуществляют рендеринг с применением функции потерь конечного изображения на дисплее как эталона.
2. Способ по п. 1, в котором на этапе оценки оценивают положение и геометрию камеры с помощью программного обеспечения для камеры, такого как Agisoft Metashape, или COLMAP, или Open3D.
3. Способ по п. 2, в котором используют стандартные портативные сенсоры RGB-D для захвата необработанных данных, которые затем обрабатывают упомянутым программным обеспечением для камеры.
4. Способ по п. 1, в котором на этапе растеризации
 - сначала растеризуют каждую точку в квадрат с длиной стороны, обратно пропорциональной глубине точки относительно камеры, причем нейронная рендерная сеть обеспечивает процесс рендеринга, выполняемый OpenGL, без сглаживания;
 - применяют алгоритм Z -буфера для наложения этих квадратов друг на друга с использованием их глубины относительно камеры;
 - создают необработанное изображение каналов посредством итерации по всем наборам отпечатков и заполнения всех пикселей;
 - используют предварительно обученную рендерную сеть с обучаемыми параметрами для преобразования необработанного изображения каналов в трехканальное изображение RGB.
5. Способ по п. 1, в котором облако точек получают с помощью алгоритмов, реализованных в различных приложениях как открытых, так и коммерческих, выбранных из группы COLMAP или Agisoft Metashape.
6. Способ по п. 1, в котором облако точек является представлением геометрии сцены.
7. Способ по п. 1, в котором нейронная рендерная сеть использует глубокую сверточную нейронную сеть для создания фотореалистичных рендеров с новых точек наблюдения.
8. Способ по п. 6, в котором сверточную нейронную сеть используют таким образом, что выходное значение цвета в пикселе зависит от множества нейронных дескрипторов и множества точек, проецируемых в окрестности этого пикселя.
9. Способ по п. 1, в котором нейронные дескрипторы описывают как геометрические, так и фотометрические свойства данных.
10. Способ по п. 8, в котором дескрипторы являются локальными дескрипторами, причем локальные дескрипторы обучаются непосредственно на данных и это обучение

происходит в координации с обучением рендерной сети.

11. Способ по п. 1, в котором камера представляет собой портативную RGBD-камеру.

12. Способ по п. 1, в котором облака точек реконструируют из простых потоков RGB.

5 13. Способ по п. 11, в котором облака точек реконструируют посредством стереосопоставления.

14. Машиночитаемый носитель, на котором хранятся выполняемые компьютером инструкции для реализации способа по пп. 1-13.

10

15

20

25

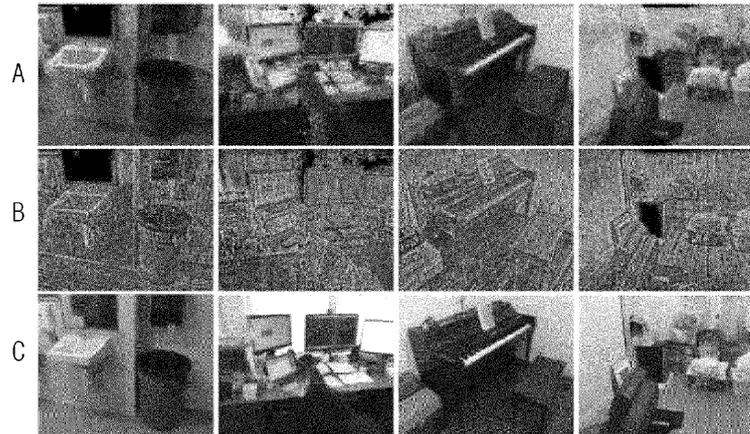
30

35

40

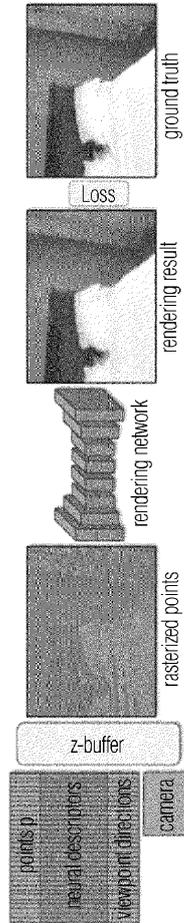
45

1/7

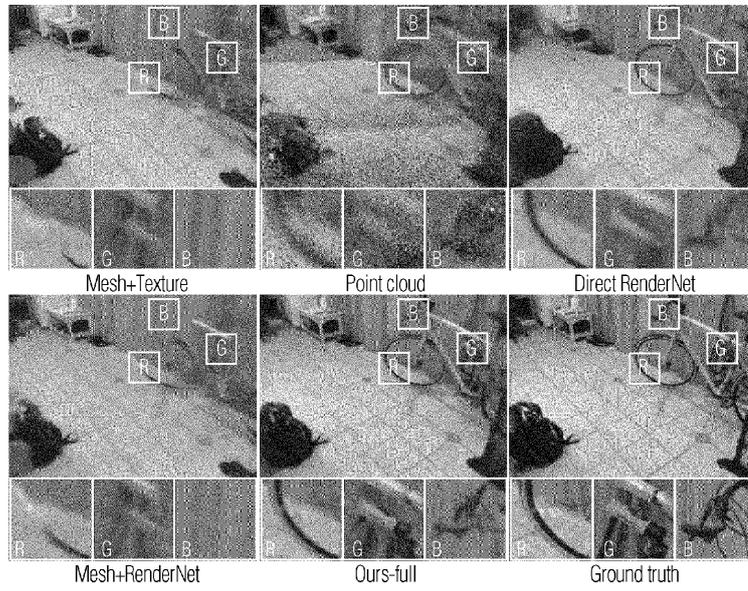


Фиг. 1

2/7

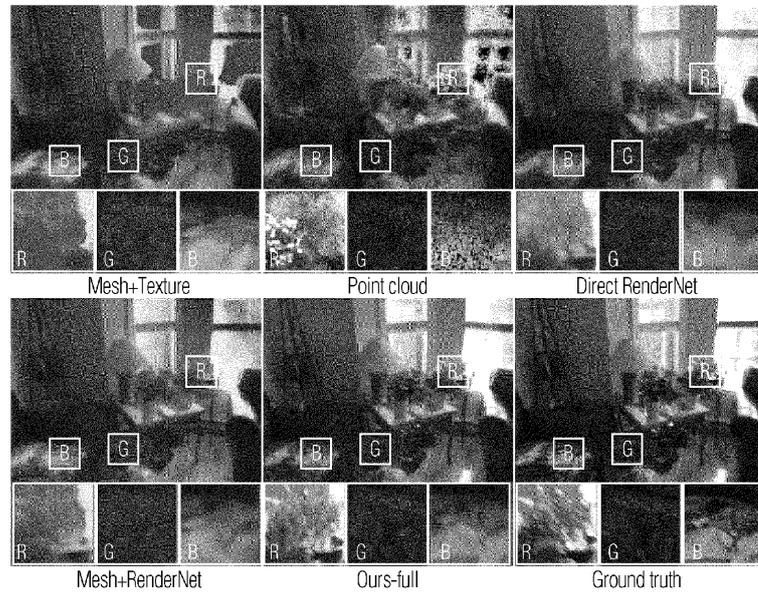


Фиг. 2



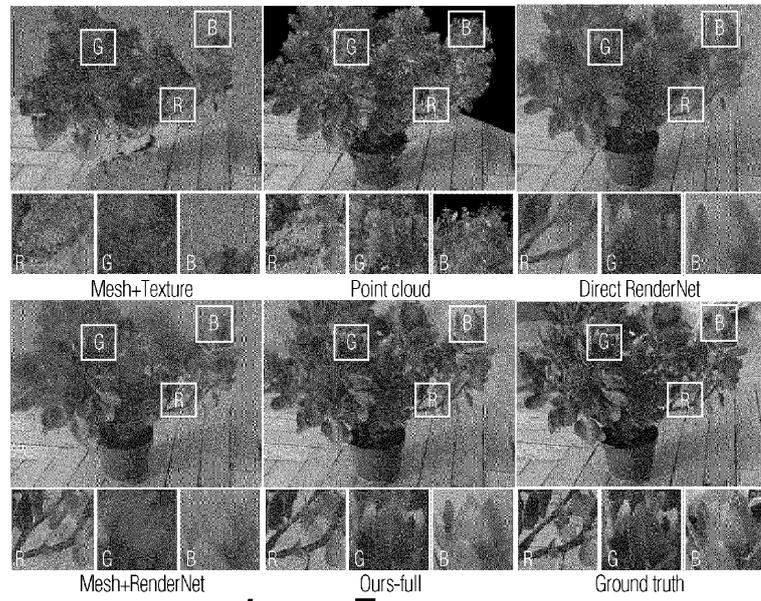
ФИГ. 3

4/7



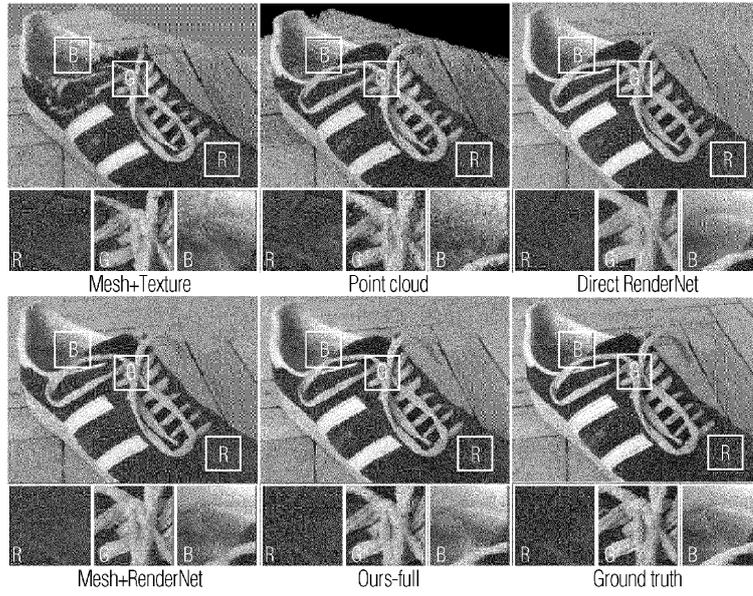
Фиг. 4

5/7



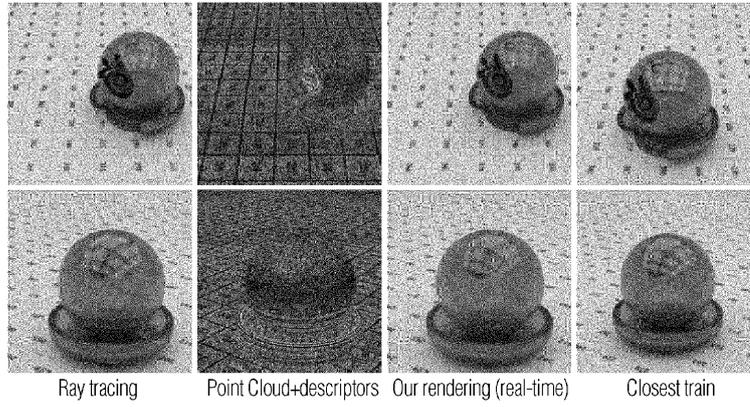
ФИГ. 5

6/7



Фиг. 6

7/7



ФИГ. 7