



DOI: <https://doi.org/10.15688/mpcm.jvolsu.2023.4.3>

УДК 004.827  
ББК 16.632.0

Дата поступления статьи: 20.07.2023  
Дата принятия статьи: 01.10.2023

## **ПОСТРОЕНИЕ 3D-МОДЕЛИ ОБЪЕКТА ПО НАБОРУ ЕГО ИЗОБРАЖЕНИЙ ПРИ ПОМОЩИ НЕЙРОННОЙ СЕТИ НА БАЗЕ АЛГОРИТМА NeRF**

**Александр Юрьевич Дряба**

Аспирант кафедры компьютерных наук и экспериментальной математики,  
Волгоградский государственный университет  
casha.dryaba@mail.ru  
<https://orcid.org/0009-0002-9587-9179>  
просп. Университетский, 100, 400062 г. Волгоград, Российская Федерация

**Аннотация.** В работе представлены способы реконструирования трехмерных моделей объектов по набору плоских изображений с использованием алгоритма NeRF для получения представления объемной сцены в виде весов многослойного перцептрона. Для каждого способа прилагается оценка затрачиваемого времени. Исходя из полученных данных можно сделать вывод о возможности распознавания форм объектов из естественной обстановки в пределах 5–10 минут, при условии переноса шага обучения нейронной сети на сторону сервера.

**Ключевые слова:** 3D-реконструкция, алгоритм NeRF, карта глубин, многослойный перцептрон, объемный рендеринг.

### **Введение**

В последние годы применение нейронных сетей в области компьютерного зрения позволило получить значительные результаты как в задачах, связанных с извлечением информации из окружающей обстановки, так и для генерирования новых данных.

Одной из задач данной области является трехмерная реконструкция объекта, которая предполагает по данному набору плоских изображений некоторой сцены, содержащей целевой объект, получить трехмерную модель объекта.

Одним из способов решения такой задачи является применение специализированных устройств, способных захватывать глубину сцены [2], с последующим объединением получающихся облаков точек.

Однако можно применить другой подход, заключающийся в получении трехмерной реконструкции, основываясь исключительно на плоских изображениях объекта.

Популярным сегодня способом является получение нейронного представления трехмерной сцены, с последующим шагом реконструирования финальной формы. Примером является существующий алгоритм NeRF (Neural Radiance Fields), позволяющий получить абстрактное представление сцены в виде весов многослойного перцептрона (MLP [11]). Подобное представление позволяет генерировать новые изображения объекта с других положений камеры, которые не представлены в изначальном наборе данных [12, с. 5].

В данной работе будут рассмотрены способы реконструирования трехмерной модели объекта с использованием такого абстрактного представления, а также проведены замеры времени каждого подхода.

## 1. Сопутствующие работы

С развитием области компьютерного зрения появилось множество алгоритмов для решения задачи реконструкции трехмерной модели по изображениям. Рассмотрим далее некоторые последние работы.

В работе [10] применяется подход с использованием сверточных нейронных сетей, а также архитектуры автокодировщика [9, с. 19] для реконструкции модели человеческого лица по фотографии. Нейронная сеть в этой работе занимается бинарной классификацией, отмечая воксели, которые принадлежат модели.

В работе [1] представлен метод реконструкции модели объекта по одному или нескольким изображениям, с предварительным распознаванием класса из заранее заданного набора [1, с. 31] и применением отдельной нейронной сети для объединения нескольких воксельных сеток вместе для образования итоговой модели.

В работе [13] представлен метод, основанный на модификации алгоритма NeRF для реконструирования моделей по изображениям из Интернета. В методе предложено использование двух MLP для формы объекта и цвета соответственно, а также 3 различных подхода к выбору точек на лучах, пускаемых через сцену, с сравнением качества и скорости обучения для каждого. В представленных примерах требуется несколько часов для получения оптимальной реконструкции больших сцен.

Работа [8] посвящена методу получения 3D-формы объекта с использованием сети, запоминающей формы объектов во внутреннем хранилище для возможности реконструирования новых классов путем переиспользования информации о близких к нему категориях.

В работе [14] используется иной подход, а именно диффузная модель для получения 3D-реконструкции. Пропуская изображения через обучаемый понижающий уровень шума модуль (denoiser), на выходе получаем трехплоскостное (triplane) представление, с помощью которого можно генерировать изображения сцены с новых ракурсов камеры.

Помимо этого существует ряд работ, посвященных фундаментальным аспектам самой задачи восстановления 3D-объекта по его проекции на 2D-изображения. Перечислим такие работы.

В работе [4] доказывается теорема о существовании решения задачи вычисления параметров плоскости, содержащей треугольник, а также единственность такого решения при выполнении определенных условий.

В работе [5] предлагается два подхода к решению задачи восстановления поверхности вращения по граничным кривым ее проекции: через решение системы дифференциально-функциональных уравнений и через кусочно-коническую аппроксимацию искомой поверхности.

В работе [3] предлагается алгоритм восстановления поверхности по проекции ее треугольной сетки, при условии наличия знания об углах треугольников. Помимо этого алгоритм может быть использован для определения ориентации объектов по фотографии.

В работе [6] описываются математические основы методов решения задач определения пространственной ориентации и положения объекта по его фотографии.

Опишем далее принцип работы алгоритма NeRF.

## 2. Neural Radiance Fields

В работе [12] был представлен метод получения изображений трехмерной сцены с новых ракурсов камеры сцен при помощи оптимизации функции, отображающей множество точек пространства  $(x_i, y_i, z_i)$ , а также направлений камеры  $(\theta_i, \phi_i)$  во множество цветов  $c_i$  и соответствующих им плотностей  $\sigma_i$ .

Здесь  $\sigma_i$  означает вероятность того, что пущенный через сцену луч остановится на бесконечно малой частице в точке  $i$ . Или, иными словами, вероятность столкновения луча с объектом в данной точке.

Для получения абстрактного представления сцены необходимо обучить нейронную сеть, вычисляя для каждой точки пространства значения  $\sigma_i$  и цвета, с последующим рендерингом финального изображения и вычислением среднеквадратичной ошибки, в качестве значения функции потерь (loss). Важными параметрами здесь являются матрица, преобразующая координаты окна камеры в мировые, а также фокусное расстояние.

Итоговая модель позволяет получать изображения объекта с новых ракурсов камеры (не представленных в изначальном наборе данных), пуская луч для каждого пикселя изображения по направлению камеры  $(\theta, \phi)$ . Финальное качество зависит как от разрешения, так и от числа точек, которые берутся на луче. Два этих основных параметра напрямую влияют на время обучения MLP и на время рендеринга кадров.

Стоит отметить, что помимо позиционного кодирования (positional encoding [12, с. 412]), влияющего на качество обучаемой модели, в работе используется иерархическое семплирование (hierarchical volume sampling [12, с. 413]), которое подразумевает под собой одновременную оптимизацию второй функции, с использованием знания о положении объекта на сцене для того, чтобы брать точки на луче, находящиеся ближе к нему. Как итог, вторая функция является более точной.

Такой метод, тем не менее, увеличивает время обучения (и необходимые вычислительные ресурсы для него), так как требует оптимизации сразу двух MLP, поэтому этот шаг может быть опущен в случаях, когда нужна бóльшая скорость обучения.

### 3. Обучение MLP в алгоритме NeRF

Исходный набор данных для обучения содержит изображения сцены, соответствующие им позиции камеры в пространстве и ее фокусное расстояние, а также границы сцены [12, с. 413–414].

На каждом шаге обучения берется набор лучей, проходящий через сцену, и выполняется процедура рендеринга, предполагающая вычисление коэффициента пропускания  $\hat{C}$  по формуле (1):

$$\hat{C}(r) = \sum_{i=1}^N T_i (1 - \exp(-\sigma_i \delta_i)) c_i, \text{ где } T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j \delta_j). \quad (1)$$

Здесь  $r$  — пущенный через сцену луч, для которого рассчитывается цвет;  $N$  — количество точек, которые берутся на луче;  $\sigma_i$  — значение плотности для точки  $i$ ;  $\delta_i$  — расстояние между соседними точками на луче  $i + 1$  и  $i$ ;  $c_i$  — RGB-цвет точки в пространстве;  $T_i$  — накопленный коэффициент пропускания вдоль луча.

На выходе для получения значения итогового цвета выполняется суммирование накопленных коэффициентов пропускания, умноженных на предсказанный цвет в точке на пущенном луче. После выполнения процедуры для каждого пикселя получается предсказанное изображение для данной позиции камеры, которое можно сравнить с исходным для получения значения ошибки.

Процедура рендеринга (1) после окончания обучения MLP может использоваться для «предсказания» изображений сцены с других позиций камеры, которые не были представлены в изначальном наборе данных.

Таким образом, значения плотностей не представлены в исходном наборе данных, поскольку для вычисления ошибки нужно только получаемое после процедуры рендеринга изображение [12, с. 414]. Сама же плотность точки  $\sigma_i$  предсказывается нейронной сетью [12, с. 410].

В примерах далее используется упрощенная архитектура нейронной сети [7], с пропущенным шагом иерархического семплирования, а также без передачи нейронной сети направления камеры  $(\theta, \phi)$ . В таком случае каждая точка в пространстве имеет определенный цвет, который не зависит от положения камеры в пространстве (тем самым теряется свет от внешних источников освещения). Архитектура использованной нейронной сети представлена на рисунке 1.

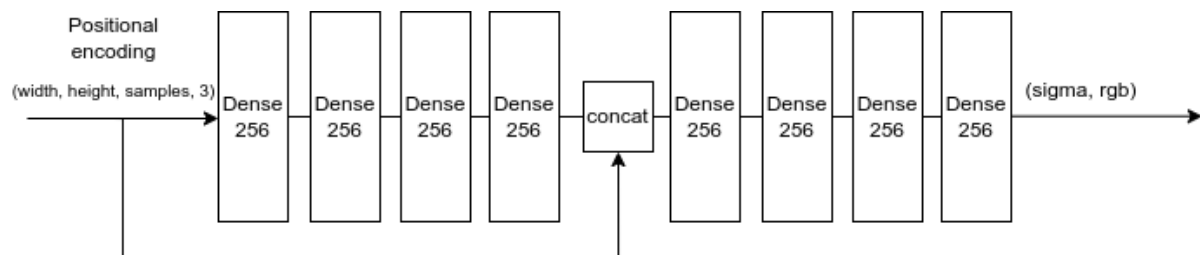


Рис. 1. Архитектура Tiny NeRF

#### 4. Способы реконструкции

Перейдем теперь к способам восстановления модели по абстрактному представлению, полученному с помощью NeRF.

##### 4.1. Использование карты глубин

Для получения карты глубин в формуле (1) необходимо заменить цвет  $c_i$  на значение глубины следующим образом:

$$D(\mathbf{r}) = \sum_{i=1}^N T_i (1 - \exp(-\sigma_i \delta_i)) z_i. \quad (2)$$

Здесь  $z_i$  — это расстояние от текущего положения камеры до точки  $i$ .

Таким образом, можно получить **карту глубин** для некоторого положения камеры в пространстве.

Получив набор карт глубин для нескольких положений камеры, можно реконструировать форму объекта несколькими способами.

##### 4.1.1. Получение модели «склеиванием»

В первом случае можно объединить вместе несколько карт глубин следующим образом. Для каждого пикселя на изображении глубины можно поставить некоторую точку в пространстве (к примеру, расположив соответствующий воксел на сцене). Составив множество вокселей для некоторого набора изображений получаем трехмерную модель.

Основными недостатками такого подхода, во-первых, является зашумленность получающегося изображения, так как при объединении шумы постепенно накапливаются в итоговой модели и поэтому необходим дополнительный шаг очистки от него, а во-вторых, влияние перспективы на итоговую реконструкцию. Лучшее качество достигается значительным увеличением фокусного расстояния во время рендеринга, что приближает получающиеся проекции к параллельному переносу. В ином случае ошибки объединения будут заметны на стыках модели.

Результаты реконструирования модели игрушечного трактора таким способом представлены на рисунке 2.

Итоговое построение полигональной модели по облаку точек здесь и далее производилось при помощи функции `create_from_point_cloud_alpha_shape` из библиотеки Open3-D.

##### 4.1.2. Получение модели «вырезанием»

Другим подходом является «вырезание» модели из воксельного куба, с использованием карт глубин.

Для этого можно воспользоваться методом `carve_depth_map` из библиотеки Open3-D, передав для каждого изображения соответствующие ему параметры камеры. Важно отметить, что для корректной работы карта глубины должна быть обработана определенным образом:

```

# обработка карты глубин перед непосредственной реконструкцией
# depth_map - изначальная карта глубин
# size - разрешение сцены (в вокселах)
# translate - смещение камеры относительно центра сцены
% depth_map = size - depth_map * size / translate
    
```

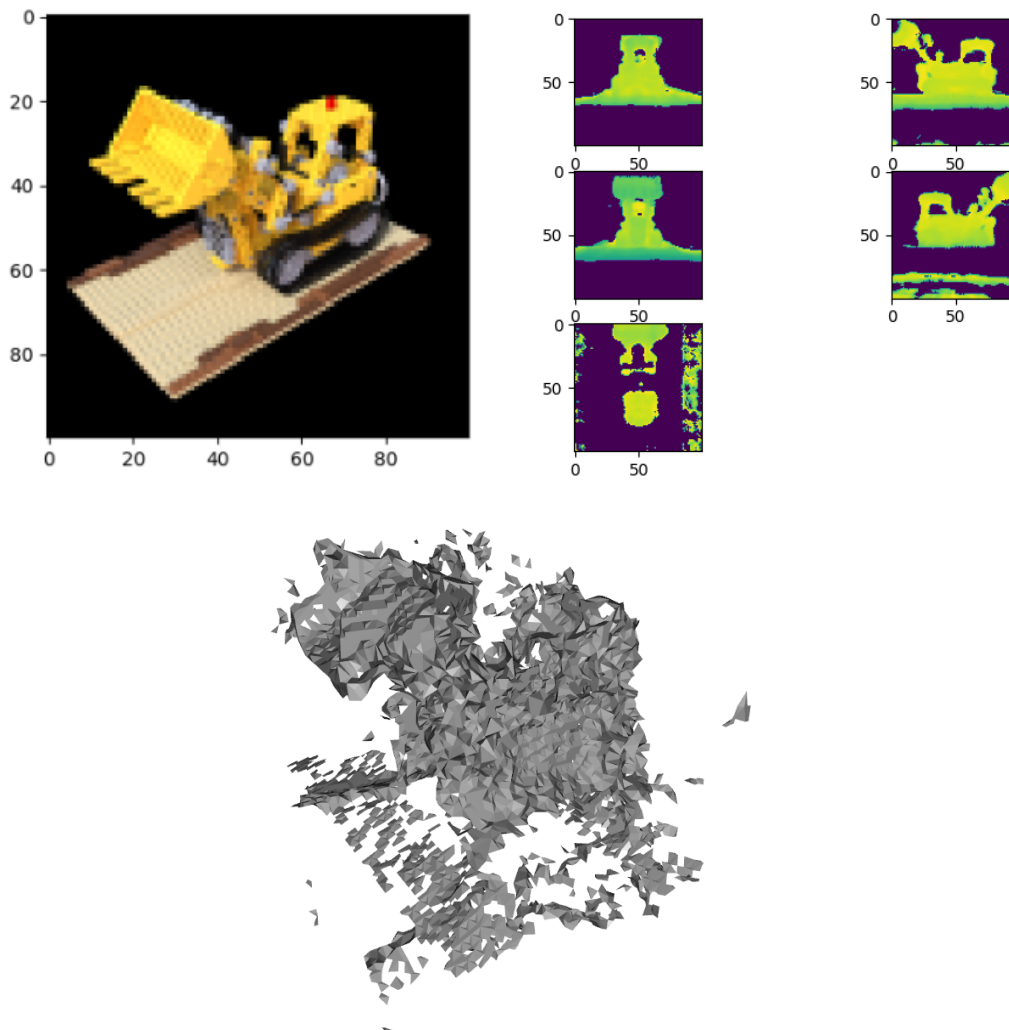


Рис. 2. Реконструкция «склейкой». Слева — пример изображения сцены.

Посередине — карты глубин сцены с пороговым отсечением.

Справа — результат трехмерной реконструкции получающейся «склейки» карт глубин вместе с помощью методов библиотеки Open3-D

Данный метод удалит все воксели из куба, у которых каждая граничная точка проецируется на большее значение глубины на карте глубин.

При использовании такого подхода учитывается перспектива камеры, и шум при этом не накапливается, так как «зашумленные» воксели вырезаются, если они отсутствуют хотя бы на одном изображении глубины. Однако итоговое качество реконструкции в первую очередь зависит от числа вычисленных карт глубин.

Результаты реконструкции для 5 ракурсов камеры представлены на рисунке 3.

#### 4.2. Значения плотности для точек в пространстве

Можно применять другой подход. Вместо получения карты глубин, можно использовать непосредственные значения  $\sigma_i$  для получения трехмерной модели.

При пускании луча через нейронное представление сцены, на выходе нейронной сети мы получаем значения плотности для каждой взятой точки на луче. Таким образом, мы обладаем знанием о положении точки в пространстве и ее плотности. Выполнив отсечение по пороговому значению (равному 2 в примере), получаем облако точек сцены.

Далее можно применить метод `VoxelGrid.create_from_point_cloud` из библиотеки `Open3D` для преобразования облака точек в воксельное представление.

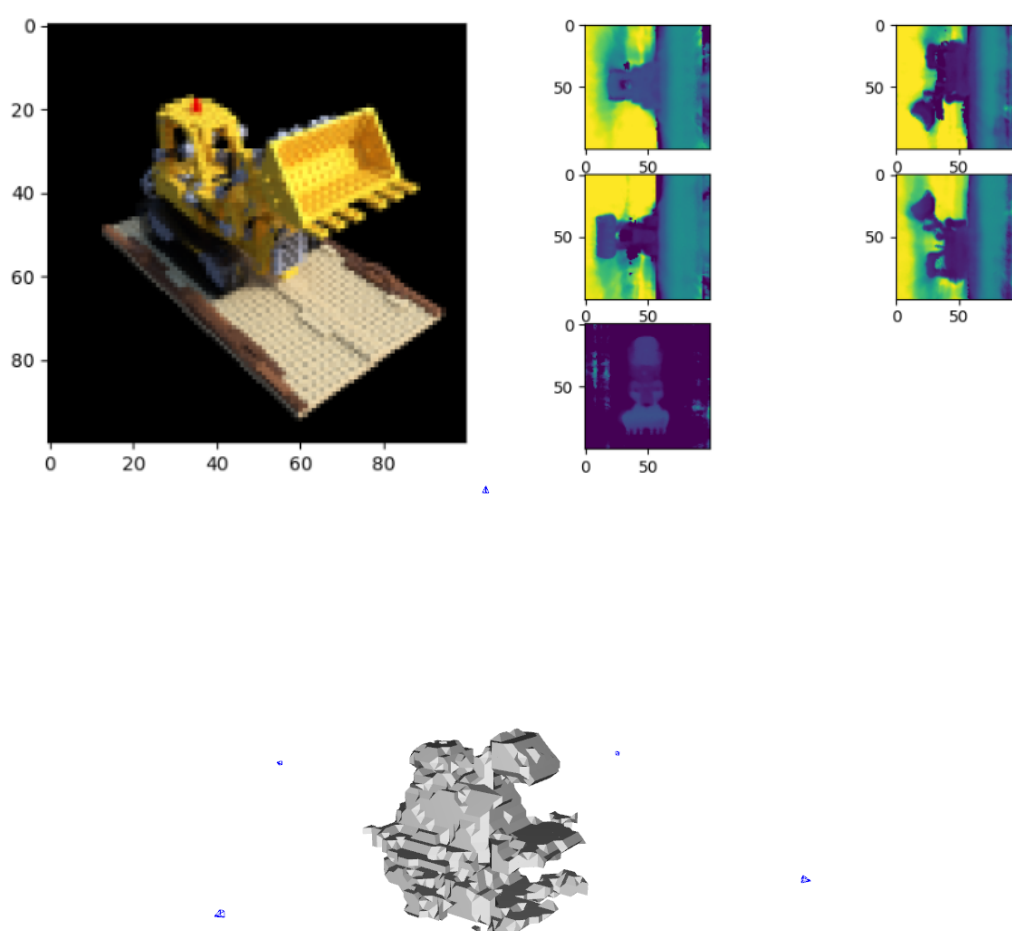


Рис. 3. Реконструкция «вырезанием». Слева — пример изображения сцены.

Посередине — карты глубин для «вырезания».

Справа — результат трехмерной реконструкции точек, полученных «вырезанием» модели из цельного куба

В примере для получения целевого облака набор лучей был пущен для одного ракурса камеры. Таким образом, точность получающейся реконструкции также можно увеличивать, получая облака точек для нескольких ракурсов и объединяя их вместе.

На рисунке 4 изображен пример такой реконструкции.

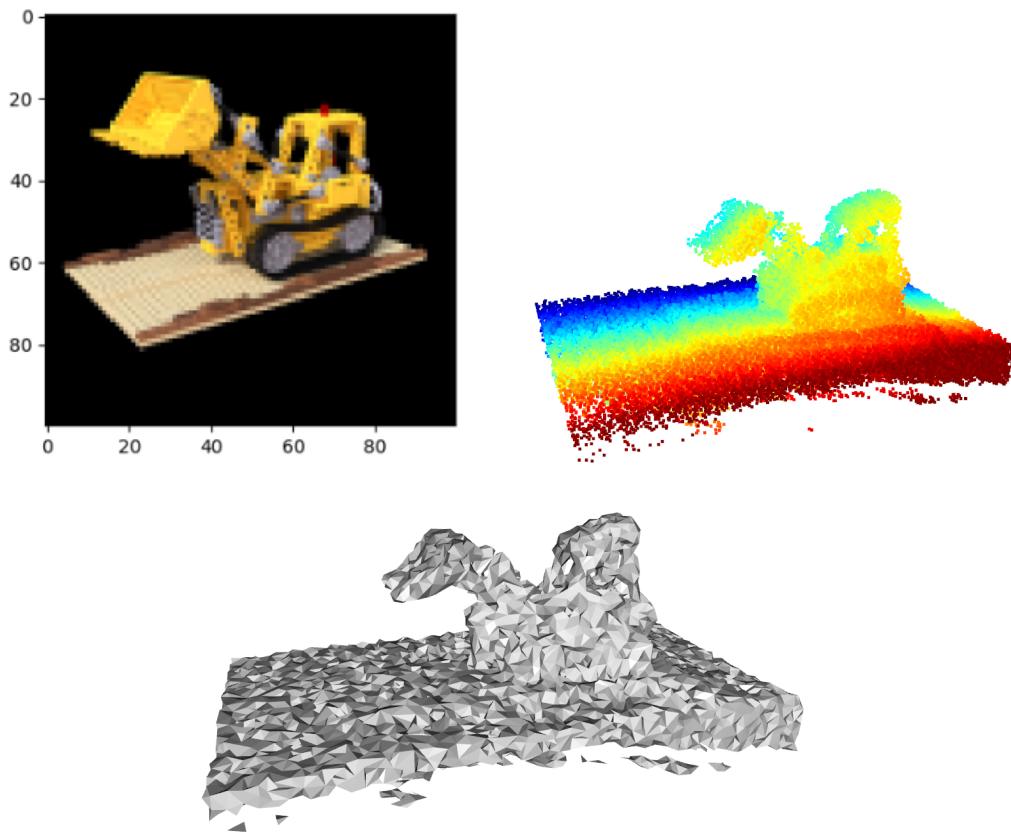


Рис. 4. Реконструкция с использованием значений плотности.  
*Слева* — пример изображения сцены.  
*Посередине* — облако точек после реконструкции.  
*Справа* — трехмерная реконструкция облака точек

## 5. Сравнение времени

В таблице представлено сравнение времени для перечисленных выше способов. Реконструкция непосредственно по значениям плотности позволяет получить приемлемую реконструкцию за меньшее число необходимых выполнений процедуры рендеринга.

Замеры были сделаны для изображений в разрешении  $100 \times 100$  пикселей для 32 точек на пускаемых через сцену лучах.

Время обучения MLP занимает 5 минут на видеокарте GeForce RTX 2080 (8 ГБ) и до 24 минут на GeForce RTX 2060 (6 ГБ). Время было замерено для случая, когда не используется иерархическое семплирование.

### Замеры времени каждого способа реконструкции модели

Название метода	Количество кадров	Время реконструкции, сек.
«Склеивание» карт глубин	5	6
«Вырезание» карт глубин	5	4,75
Значения плотности для точек	5	4,17
Значения плотности для точек	1	2,97



Таким образом, время непосредственной реконструкции по нейронному представлению является малым относительно всего текущего конвейера.

### Заключение

Итоговый конвейер позволяет получать модели объектов в пределах 5–10 минут при наличии их изображений с различных позиций камеры. Для реконструкции объектов из естественной среды понадобится дополнительный шаг для оценки положений камеры относительно объекта.

На время влияет как желаемое разрешение реконструкции, так и частота семплирования трехмерной сцены. Повышение этих двух параметров ведет к более качественной реконструкции, однако увеличивает общее время конвейера.

Представленные способы реконструкции далее будут использоваться для задачи распознавания трехмерных объектов в естественных сценах и отслеживания динамики их изменения. Подобная система позволит реже выполнять процесс обучения MLP, вместо этого полагаясь на перенос изменений из естественной среды на уже построенную модель.

Кроме того, само обучение должно выполняться на стороне сервера, обладающего достаточными мощностями для получения приемлемой скорости. Клиентское оборудование (будь это компьютер пользователя или мобильная плата) в таком случае может являться источником изображений, а также положений камеры, с которых они были сняты.

Таким образом, полученный способ реконструкции трехмерной модели по изображениям позволит создать систему обнаружения и отслеживания трехмерных объектов при помощи камеры в реальном времени.

### СПИСОК ЛИТЕРАТУРЫ

1. Гадасин, Д. В. Трехмерная реконструкция объекта по одному изображению с использованием глубоких сверточных нейронных сетей / Д. В. Гадасин, А. В. Шведов, И. А. Кузин // Т-Comm: телекоммуникации и транспорт. — 2022. — Т. 16, № 7. — С. 29–34. — DOI: <http://dx.doi.org/10.36724/2072-8735-2022-16-7-29-35>

2. Камеры глубины — тихая революция (когда роботы будут видеть). Ч. 1. — Электрон. текстовые дан. — Режим доступа: <https://habr.com/ru/articles/457524>. — Загл. с экрана.

3. Клячин, А. А. Алгоритм восстановления поверхности объекта по его изображению / А. А. Клячин, В. А. Клячин // Математическая физика и компьютерное моделирование. — 2021. — Т. 24, № 1. — С. 16–24. — DOI: <https://doi.org/10.15688/mpcm.jvolsu.2021.1.2>

4. Клячин, А. А. Теоремы существования и единственности решения обратных задач проективной геометрии для 3D реконструкции по фотоснимкам / А. А. Клячин, В. А. Клячин // Чебышевский сборник. — 2020. — Т. 21, № 4. — С. 117–128. — DOI: <https://doi.org/10.22405/2226-8383-2020-21-4-117-128>

5. Клячин, В. А. Алгоритм 3D реконструкции поверхности вращения по ее проекции / В. А. Клячин, Е. Г. Григорьева // Сибирский журнал индустриальной математики. — 2020. — Т. 23, № 1. — С. 84–92. — DOI: <https://doi.org/10.33048/SIBJIM.2020.23.108>

6. Клячин, В. А. Теоремы единственности восстановления прообраза при вырожденном преобразовании / В. А. Клячин, Е. Г. Григорьева // Математическая физика и компьютерное моделирование. — 2022. — Т. 25, № 2. — С. 17–22. — DOI: <https://doi.org/10.15688/mpcm.jvolsu.2022.2.2>

7. Реализация архитектуры Tiny NeRF. — Электрон. текстовые дан. — Режим доступа: [https://github.com/bmild/nerf/blob/master/tiny\\_nerf.ipynb](https://github.com/bmild/nerf/blob/master/tiny_nerf.ipynb). — Загл. с экрана.
8. Few-Shot Single-View 3D Reconstruction with Memory Prior Contrastive Network / Z. Xing, Y. J. Chen, Z. X. Ling, X. D. Zhou, Y. Xiang // ECCV 2022: Computer Vision — ECCV 2022. — 2022. — № 1. — P. 55–70. — DOI: [https://doi.org/10.1007/978-3-031-19769-7\\_4](https://doi.org/10.1007/978-3-031-19769-7_4)
9. Jackson, A. S. Learning Deep Architectures for AI / A. S. Jackson // Foundations. — 2009. — № 2. — P. 1–55. — DOI: <http://dx.doi.org/10.1561/22000000006>
10. Large Pose 3D Face Reconstruction from a Single Image Via Direct Volumetric CNN Regression / A. S. Jackson, A. Bulat, V. Argyriou, G. Tzimiropoulos // 2017 IEEE International Conference on Computer Vision (ICCV). — 2017. — P. 1031–1039. — DOI: <http://dx.doi.org/10.1109/ICCV.2017.117>
11. Multilayer Perceptron Explained with a Real-Life Example and Python Code: Sentiment Analysis. — Electronic text data. — Mode of access: <https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141>. — Title from screen.
12. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis / B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, R. Ng // Computer Vision — ECCV 2020. — 2020. — № 12346. — P. 405–421. — DOI: [https://doi.org/10.1007/978-3-030-58452-8\\_24](https://doi.org/10.1007/978-3-030-58452-8_24)
13. Neural 3D Reconstruction in the Wild / J. M. Sun, X. Chen, Q. Q. Wang, Z. Q. Li, H. Averbuch-Elor, X. W. Zhou, N. Snavely // ACM SIGGRAPH 2022 Conference Proceedings. — NY : Association for Computing Machinery, 2022. — P. 1–9. — DOI: <https://doi.org/10.1145/3528233.3530718>
14. RenderDiffusion: Image Diffusion for 3D Reconstruction, Inpainting and Generation / T. Anciukevicius, Z. X. Xu, M. Fisher, P. Henderson, H. Bilen, N. J. Mitra, P. Guerrero // CVPR 2023: IEEE Conference on Computer Vision and Pattern Recognition. — 2023. — P. 12608–12618. — DOI: <https://doi.org/10.1109/CVPR52729.2023.01213>

## **REFERENCES**

1. Gadasin D.V., Shvedov A.V., Kuzin I.A. Trekhmernaya rekonstruktsiya obyektov po odnomu izobrazheniyu s ispolzovaniem glubokikh svyortochnykh neyronnykh setey [Three-Dimensional Reconstruction of an Object From a Single Image Using Deep Convolutional Neural Networks]. *T-Comm: telekommunikatsii i transport*, 2022, vol. 16, no. 7, pp. 29–34. DOI: <http://dx.doi.org/10.36724/2072-8735-2022-16-7-29-35>
2. *Kamery glubiny — tikhaya revolyutsiya (kogda roboty budut videt). Ch. 1* [Depth Cameras — a Silent Revolution (When Robots Could See). Part 1]. URL: <https://habr.com/ru/articles/457524>
3. Klyachin A.A., Klyachin V.A. Algoritm vosstanovleniya poverkhnosti obyektov po ego izobrazheniyu [Algorithm for Restoring the Surface of an Object from its Image]. *Matematicheskaya fizika i kompyuternoe modelirovanie* [Mathematical Physics and Computer Simulation], 2021, vol. 24, no. 1, pp. 16–24. DOI: <https://doi.org/10.15688/mpcm.jvolsu.2021.1.2>
4. Klyachin A.A., Klyachin V.A. Teoremy sushchestvovaniya i edinstvennosti resheniya obratnykh zadach proektivnoy geometrii dlya 3D rekonstruktsii po fotosnimkam [Existence and Uniqueness Theorems for Solutions of Inverse Problems of Projective Geometry for 3D Reconstruction from Photographs]. *Chebyshevskiy sbornik*, 2020, vol. 21, no. 4, pp. 117–128. DOI: <https://doi.org/10.22405/2226-8383-2020-21-4-117-128>
5. Klyachin V.A., Grigoryeva E.G. Algoritm 3D rekonstruktsii poverkhnosti vrashcheniya po eyo proektsii [A 3D Reconstruction Algorithm of a Surface of Revolution from its Projection]. *Sibirskiy zhurnal industrialnoy matematiki* [Journal of Applied and Industrial Mathematics], 2020, vol. 23, no. 1, pp. 84–92. DOI: <https://doi.org/10.33048/SIBJIM.2020.23.108>

6. Klyachin V.A., Grigoryeva E.G. Teoremy edinstvennosti vosstanovleniya proobraza pri vyrozhdennom preobrazovanii [Uniqueness Theorem of Reconstruction of Preimage by its Image under Degenerate Mapping]. *Matematicheskaya fizika i kompyuternoe modelirovanie* [Mathematical Physics and Computer Simulation], 2022, vol. 25, no. 2, pp. 17-22. DOI: <https://doi.org/10.15688/mpcm.jvolsu.2022.2.2>

7. *Realizatsiya arkhitektury Tiny NeRF* [Tiny NeRF Implementation]. URL: [https://github.com/bmild/nerf/blob/master/tiny\\_nerf.ipynb](https://github.com/bmild/nerf/blob/master/tiny_nerf.ipynb)

8. Xing Z., Chen Y.J., Ling Z.X., Zhou X.D., Xiang Y. Few-Shot Single-View 3D Reconstruction with Memory Prior Contrastive Network. *ECCV 2022: Computer Vision – ECCV 2022*, 2022, no. 1, pp. 55-70. DOI: [https://doi.org/10.1007/978-3-031-19769-7\\_4](https://doi.org/10.1007/978-3-031-19769-7_4)

9. Jackson A.S. Learning Deep Architectures for AI. *Foundations*, 2009, no. 2, pp. 1-55. DOI: <http://dx.doi.org/10.1561/22000000006>

10. Jackson A.S., Bulat A., Argyriou V., Tzimiropoulos G. Large Pose 3D Face Reconstruction From a Single Image Via Direct Volumetric CNN Regression. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1031-1039. DOI: <http://dx.doi.org/10.1109/ICCV.2017.117>

11. *Multilayer Perceptron Explained with a Real-Life Example and Python Code: Sentiment Analysis*. URL: <https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141>

12. Mildenhall B., Srinivasan P.P., Tancik M., Barron J.T., Ramamoorthi R., Ng R. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *Computer Vision – ECCV 2020*, 2020, no. 12346, pp. 405-421. DOI: [https://doi.org/10.1007/978-3-030-58452-8\\_24](https://doi.org/10.1007/978-3-030-58452-8_24)

13. Sun J.M., Chen X., Wang Q.Q., Li Z.Q., Averbuch-Elor H., Zhou X.W., Snavely N. Neural 3D Reconstruction in the Wild. *ACM SIGGRAPH 2022 Conference Proceedings*. NY, Association for Computing Machinery, 2022, pp. 1-9. DOI: <https://doi.org/10.1145/3528233.3530718>

14. Anciukevicius T., Xu Z.X., Fisher M., Henderson P., Bilen H., Mitra N.J., Guerrero P. RenderDiffusion: Image Diffusion for 3D Reconstruction, Inpainting and Generation. *CVPR 2023: IEEE Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12608-12618. DOI: <https://doi.org/10.1109/CVPR52729.2023.01213>

## BUILDING A 3D-MODEL OF AN OBJECT FROM A SET OF ITS IMAGES USING A NEURAL NETWORK BASED ON THE NeRF ALGORITHM

**Alexander Yu. Dryaba**

Postgraduate Student, Department of Computer Sciences  
and Experimental Mathematics,  
Volgograd State University  
cash.dryaba@mail.ru  
<https://orcid.org/0009-0002-9587-9179>  
Prosp. Universitetsky, 100, 400062 Volgograd, Russian Federation

**Abstract.** The work was conducted as part of the development of a computer vision system for analyzing the environment, which could be utilized, for instance, by an autonomous mobile robot. This system involves using a camera to gather information about the surrounding environment. The paper presents methods for reconstructing three-dimensional models of objects solely from a set of 2D-images, using the NeRF algorithm to obtain a representation of a three-dimensional scene in the form of weights of a multilayer perceptron.

Each method includes an estimate of the algorithm's time consumption. Based on the data obtained, it was concluded that it is feasible to recognize the shapes of objects from a natural environment within 5–10 minutes, provided that the neural network training step is transferred to the server side.

**Key words:** 3D-reconstruction, NeRF, depth map, MLP, volume rendering.