

Улучшение результатов фотограмметрии, используя нейронные сети

Артюшков Павел Николаевич

Научный руководитель: Воронцов Илья Евгеньевич

Сейчас все популярней становится технология **фотограмметрии** - научно-техническая дисциплина, занимающаяся определением формы, размеров, положения и иных характеристик объектов по их фотографиям.

Еще лет 10 назад, если нужно было отсканировать какой-нибудь объект и получить его виртуальную 3D модель, то это можно было сделать только используя очень дорогой 3D-сканер, и не каждый обладал нужными навыками для этого. Теперь это можно сделать при помощи фотографий с камеры обыкновенного телефона.

Но как бы это хорошо не звучало, у этой технологии все еще есть множество недостатков. Многие детали объекта сильно искажаются или вообще теряются. Например, любая ровная поверхность окажется шероховатой, а острый угол будет сглаженным. Для идеального результата нужно, во-первых, сделать наилучшие условия для фотографий: убедиться, что в объекте нету металлических или отражающих деталей, настроить мягкий свет без теней и следить за четкостью изображений. А во-вторых, вручную обработать результат и избавиться от каких-либо неровностей и огрех. Конечно, не все и не всегда могут это сделать. Я много раз пытался воспользоваться этой технологией, но все результаты не удовлетворяли меня. Отсюда мы и придумали этот проект.

Цель проекта — написать и обучить нейронную сеть для исправления ошибок в отсканированных моделях и улучшения их качества. Наш алгоритм будет брать на вход уже полученную модель после сканирования и исправлять в ней маленькие неровности и ошибки.

Сейчас существует нейронная сеть AI Image Upscaler, которая используется для решения похожей задачи, но на обычных фотографиях. Для её обучения фотографии намеренно размывают и делают менее качественными, чтобы обучить модель предсказывать как будет выглядеть оригинальная фотография, получая на вход только фотографию в плохом качестве. Этот подход перенесен нами на отсканированные 3D модели, а также модифицирован для улучшения качества работы с учётом специфики задачи.

Актуальность проекта

Сейчас существуют такие популярные решения для фотограмметрии как [Agisoft Metashape](#), [Reality Capture](#) от Epic Games, и самое современное и продвинутое

решение - [Polycam](#). Последний продукт позволяет сканировать модели с помощью телефона и делиться сканами с другими людьми как в социальной сети. Это значит, что у этих продуктов есть большая база пользователей. Мое решение пригодно для встраивания в сторонние продукты и позволяет улучшать результат фотограмметрии простым в использовании способом.

Процесса сбора датасета

Возьмем произвольную 3D модель. Для примера это будет небольшой домик. Используя Blender мы отрендерили 300 фотографий вокруг него и применили к ним программное обеспечение для фотограмметрии. На рис. 1 отображено распознанное программой расположение всех камер вокруг объекта.

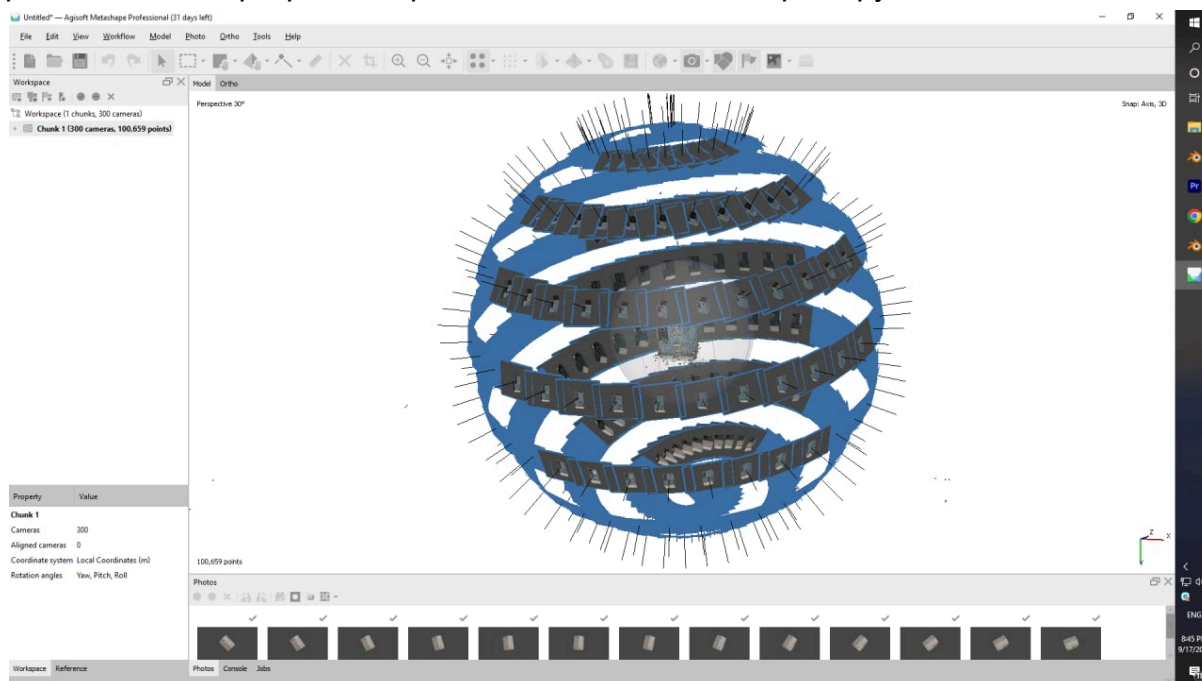


Рис. 1. Фотографии виртуального объекта расположены в точках съёмки и отражают взятые при рендеринге ракурсы. Расположение объектов выявлено и отображено программой.

По этим фотографиям строится облако точек (Dense point cloud) для модели.

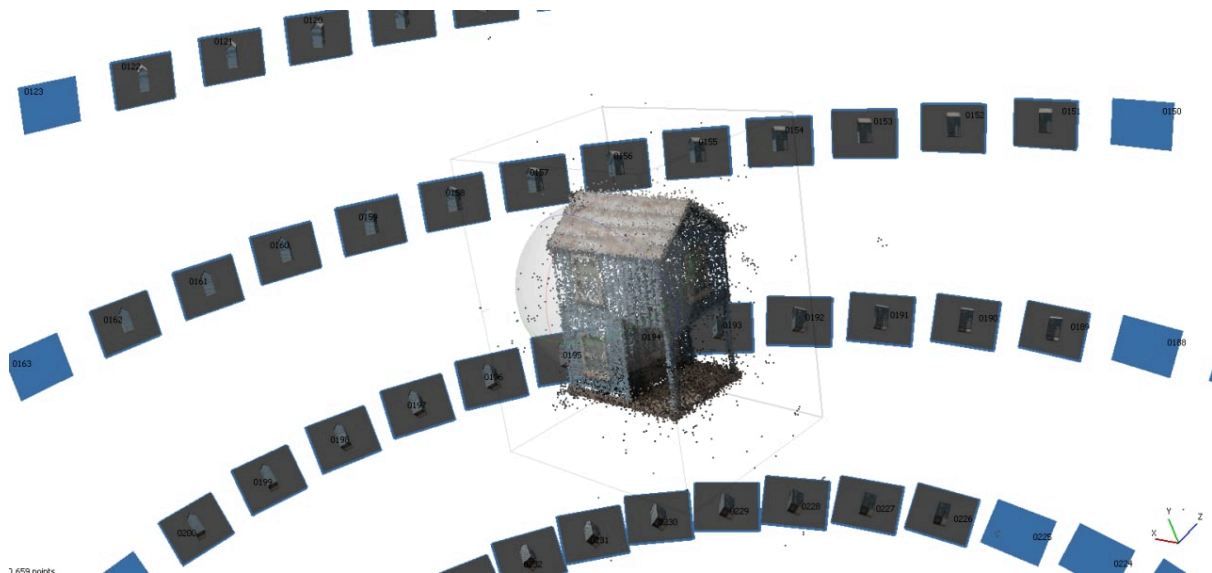
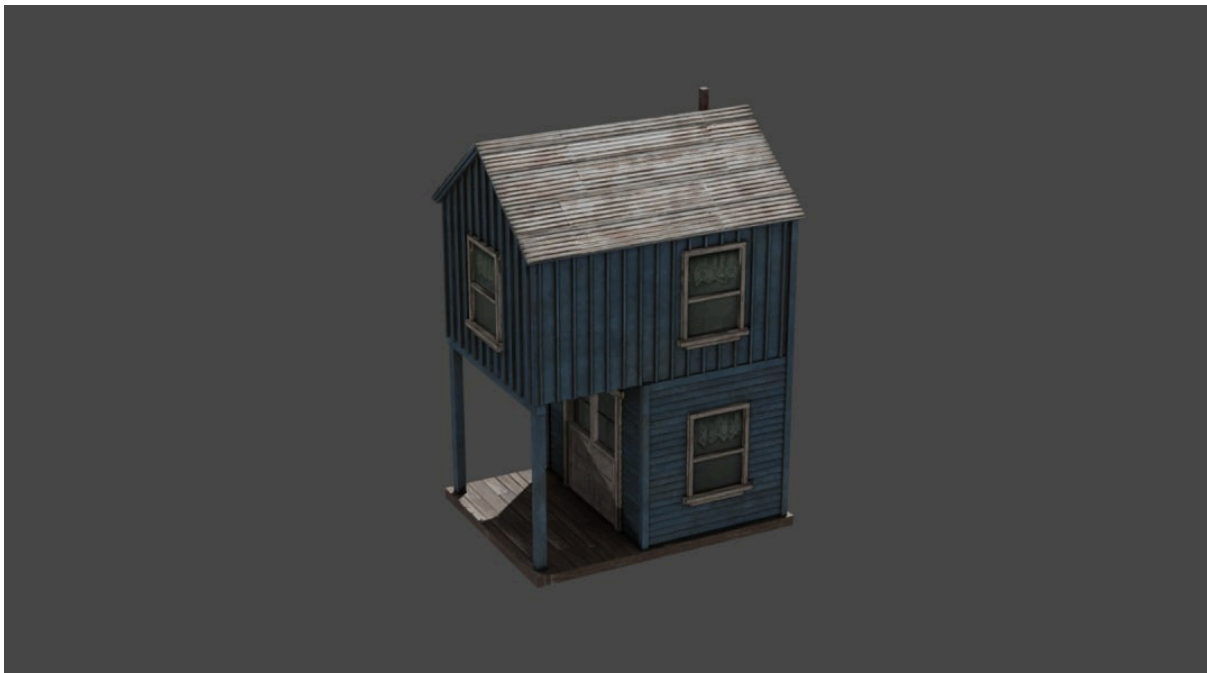


Рис. 2 Пример облака точек, построенного по фотографиям.

И вот изначальная и отсканированная модели в сравнении.



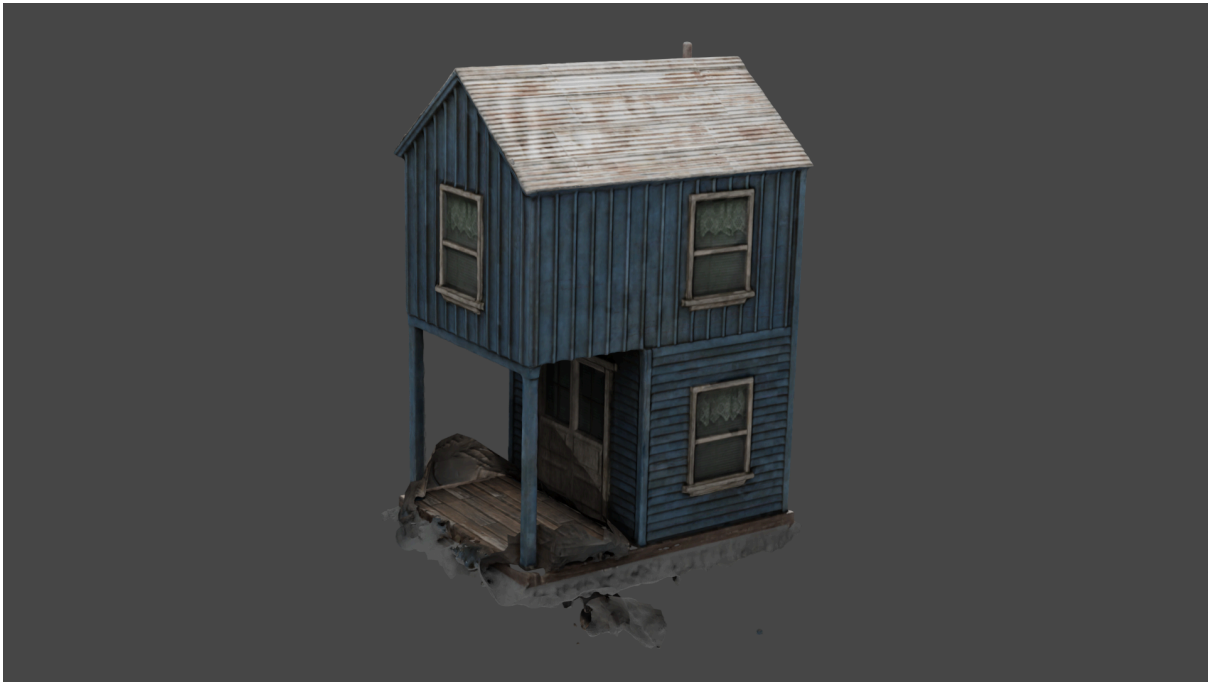
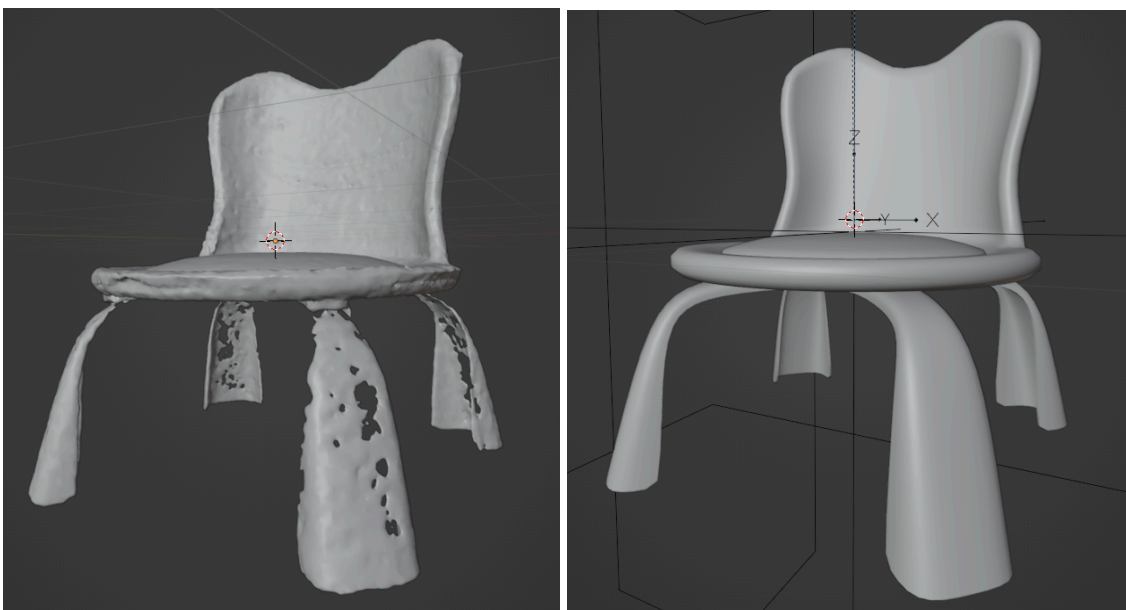


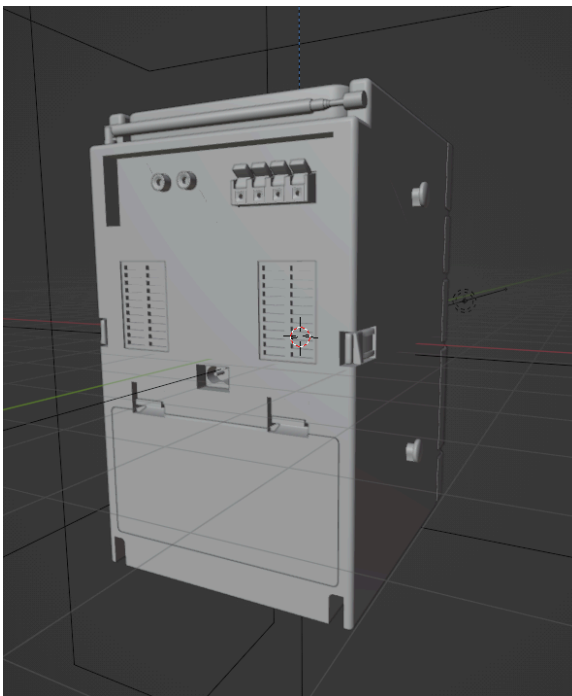
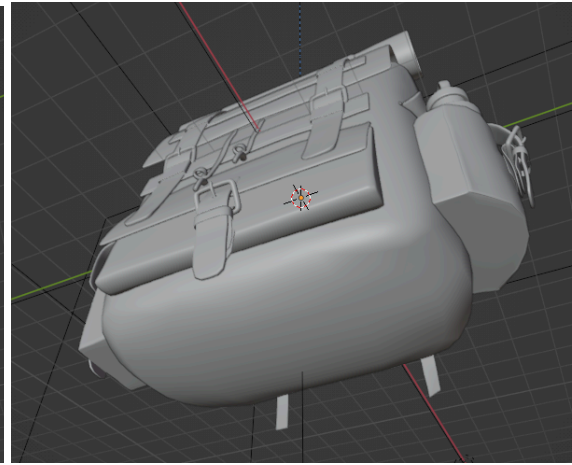
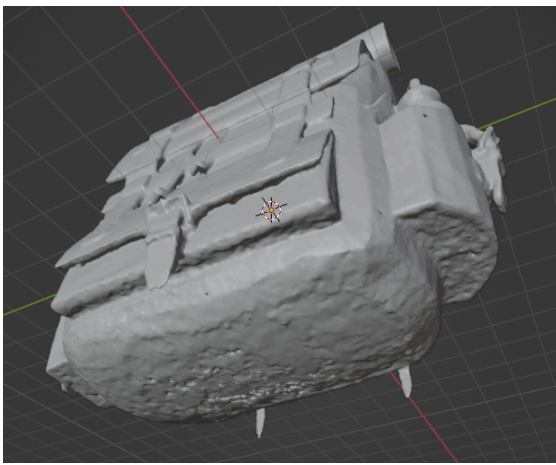
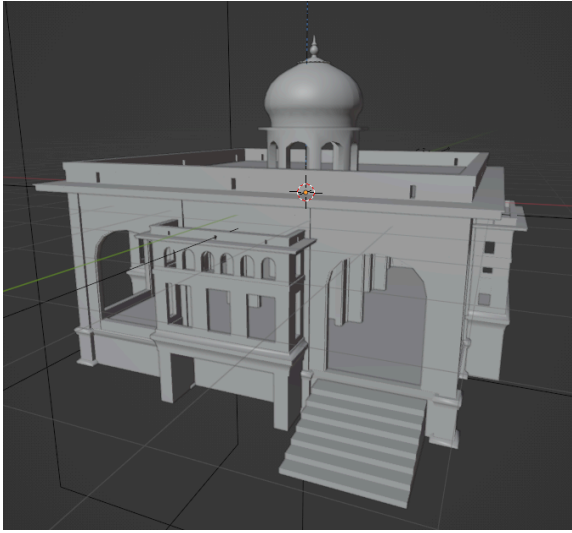
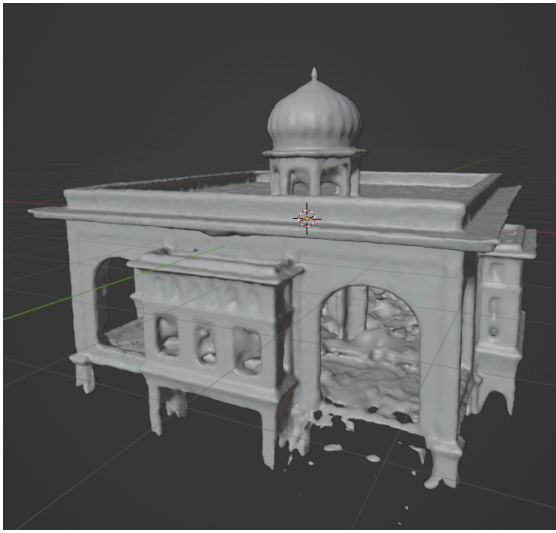
Рис. 3 Пример исходной и отсканированной 3D модели

На рис. 3 видно, что в зонах где было меньше света и которые просто были сильнее скрыты от наблюдателя, появляются проблемы, особенно на краях.

Эти примеры были получены нами, работая с программой фотограмметрии вручную и в хороших условиях. На рис. 4 отображено еще несколько реальных, непростых для сканирования примеров из датасета. Они не имеют текстур, что упрощает обучение.

На этих примерах видно как испортилась детализация модели, и иногда даже пропадают какие-то детали





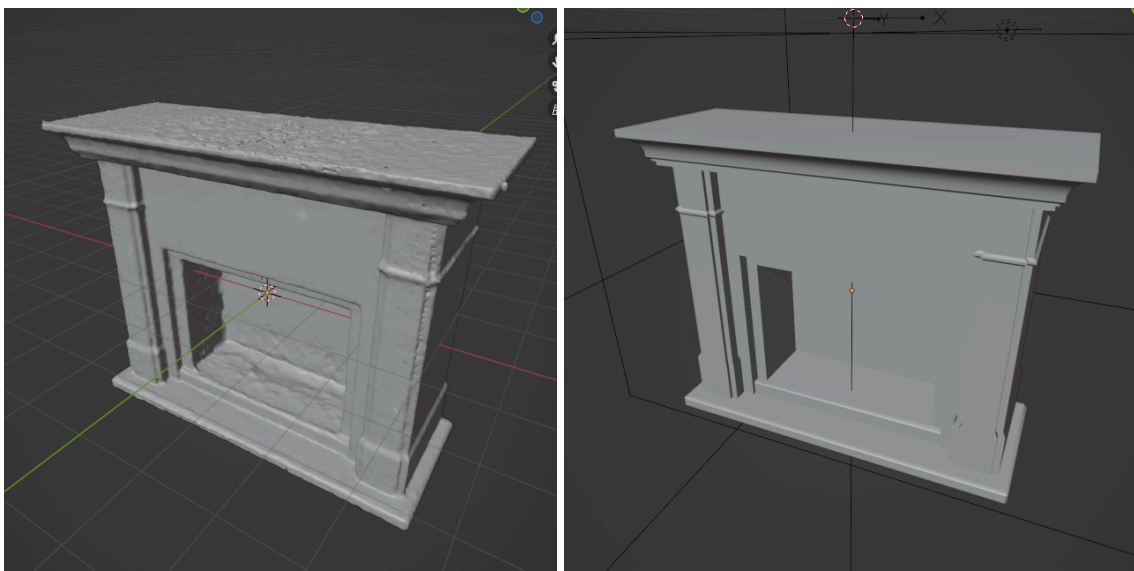


Рис. 4 Другие примеры записей из датасета. Первое изображение - отсканированная модель, второе - оригинальная.

Техническая часть

Для сбора датасета выбирались произвольные качественные модели в открытом доступе из интернета. Изначально была идея самим генерировать произвольные многогранники, но мы решили, что тогда нейросеть плохо научится работать в ситуациях из реального мира, ведь там нету рандомно-сгенерированных многогранников. Дальше мы рендерим фотографии модели со всех возможных ракурсов с помощью Blender и передаем их в сервис фотограмметрии Agisoft Metashape, чтобы потом получить финальную отсканированную модель. Таким образом получается датасет из оригинальных моделей и отсканированных. Для обеспечения одинакового расположения обеих моделей в пространстве, мы сами устанавливаем координаты всех камер, вместо того чтобы предоставить эту задачу программе. И дополнительно, это ускоряет время обработки. Наша задача — обучить нейронную сеть предсказывать, как выглядела оригинальная модель только по её отсканированной версии.

Нейронная сеть для коррекции моделей

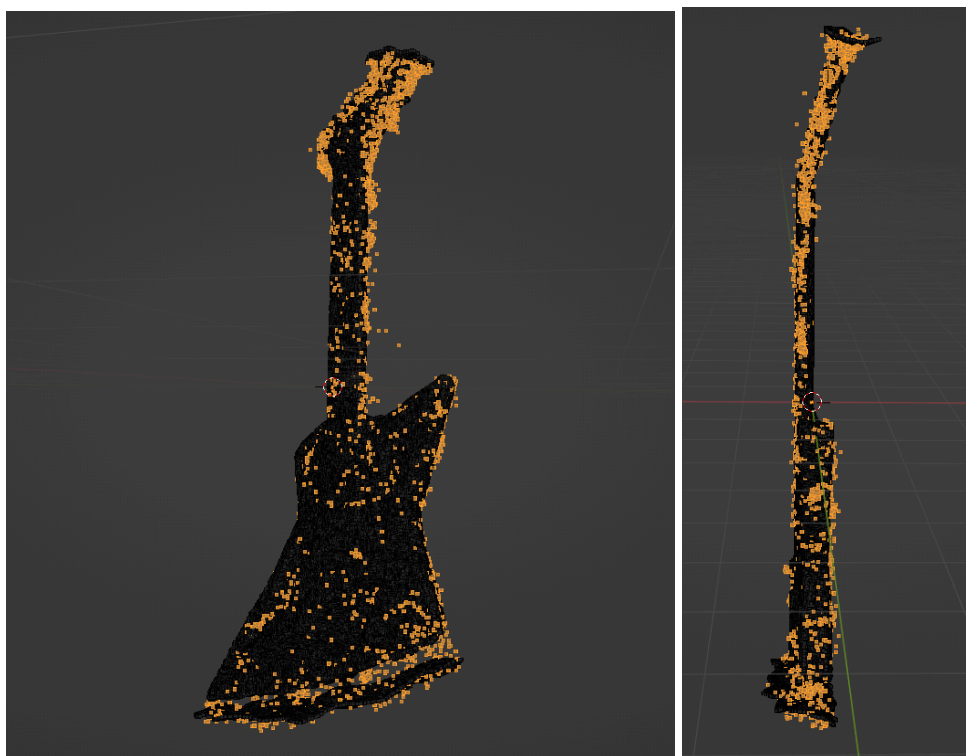
Работать с моделями в нейронных сетях намного сложнее, чем с изображениями, потому что картинки можно представить как постоянный (одной размерности) набор пикселей - точки с определенным цветом. Но для работы с моделями нужно выбирать как ее представлять. Есть несколько способов: разбивать модель на маленькие кубики или воксели (voxels), или же представлять модель как набор вершин и граней. Первый способ сильно уменьшает качество, а второй исключает все грани фигуры. Но современные модели умеют объединять оба этих метода, чтобы наиболее оптимально работать с моделями. Например, так делает PolyGen

[<https://www.deepmind.com/publications/polygen-an-autoregressive-generative-model-of-3d-meshes>] от исследователей компании DeepMind.

По причине низкого предоставляемого разрешения результата и большого объема памяти, мы отказались от работы с воксельным представлением данных. В данный момент наша модель работает с облаком точек для упрощения задачи, но в будущем планируется строить и грани.

Результаты

- Написан распределенный сервис для **построения обучающего набора данных**, запускаемый на сети из различных вычислительных устройств. Он работает на базе Python-библиотеки **bpy** (blender for python) для рендера различных углов модели и [Metashape API](#) для фотограмметрии.
- Вручную **отобраны качественные модели** при помощи своего инструмента для валидации датасетов.
- Собран датасет размером в **279 валидированных моделей**.
- Написан код для обучения **рекуррентной нейронной сети RNN типа seq2seq** с механизмом внимания для преобразования облаков точек в другой набор точек, более приближенный к изначальной, качественной модели. Код оптимизирован для ускорения работы: в нём реализована подгрузка данных с кешированием, нормализация облаков вершин и разбиение вершин на компактные поднаборы.
- Используя библиотеку **pytorch3d** реализован алгоритм для взятия тех вершин модели, которые находятся на ее поверхности: то есть видны камере
- На рис. 5 приведён **пример работы** сети на валидационной модели (нейронная сеть не обучалась на ней)



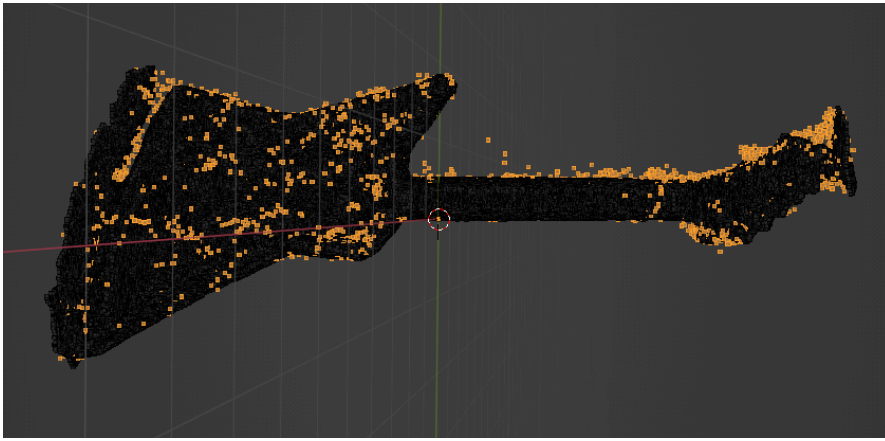


Рис. 5 Пример финального предсказания нейронной сети на полной 3D модели

Можно заметить что нейронная сеть близко повторяет искомую модель, но для существенных улучшений нужно обучить намного большую сеть

Можно ближе присмотреться к отдельным предсказаниям сети на чанках. На рис. 6 чёрные точки взяты из искомой target 3D модели, а оранжевые - это предсказания нашей нейронной сети

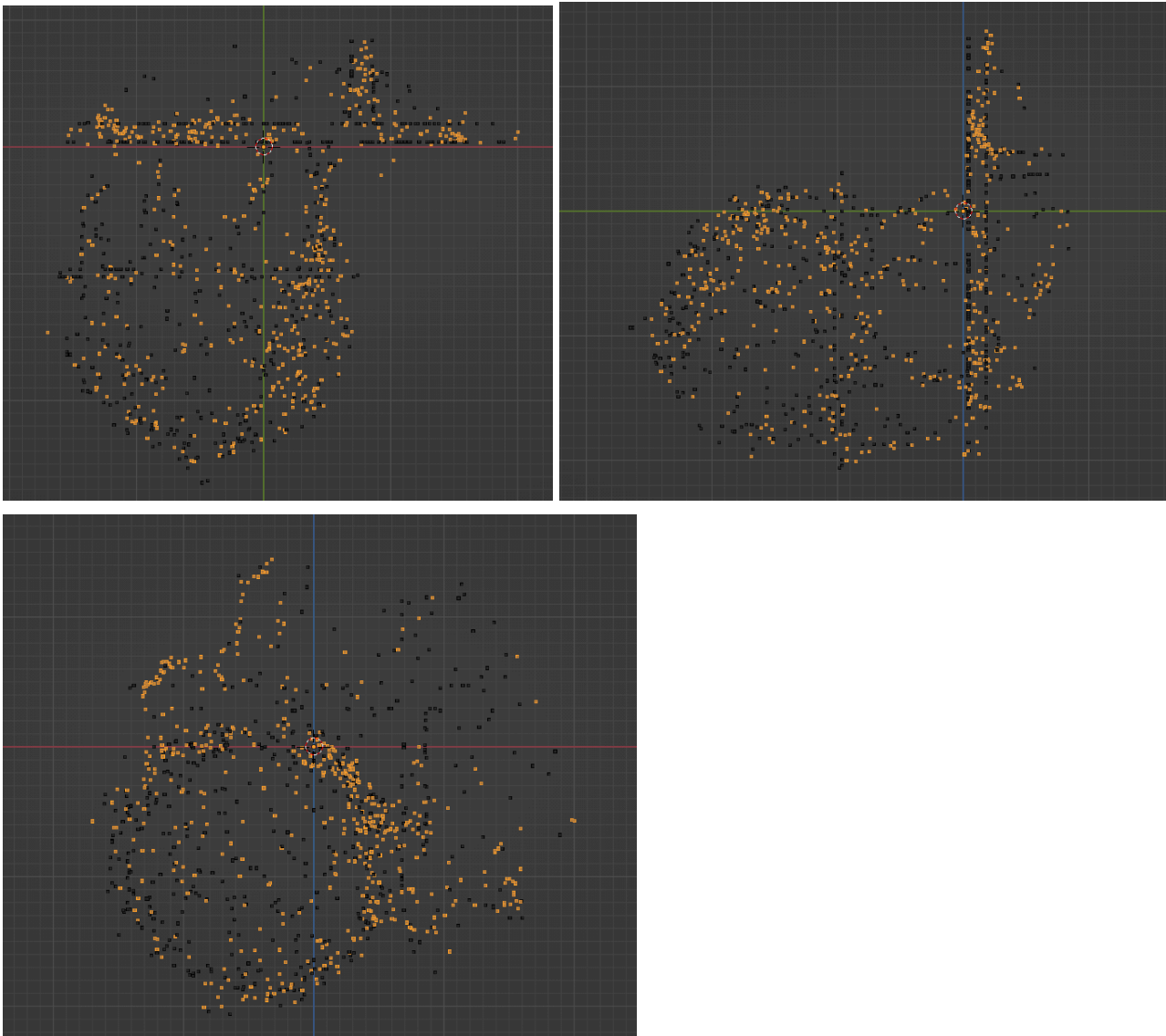


Рис. 6 Визуализация предсказаний модели на компактных поднаборах данных

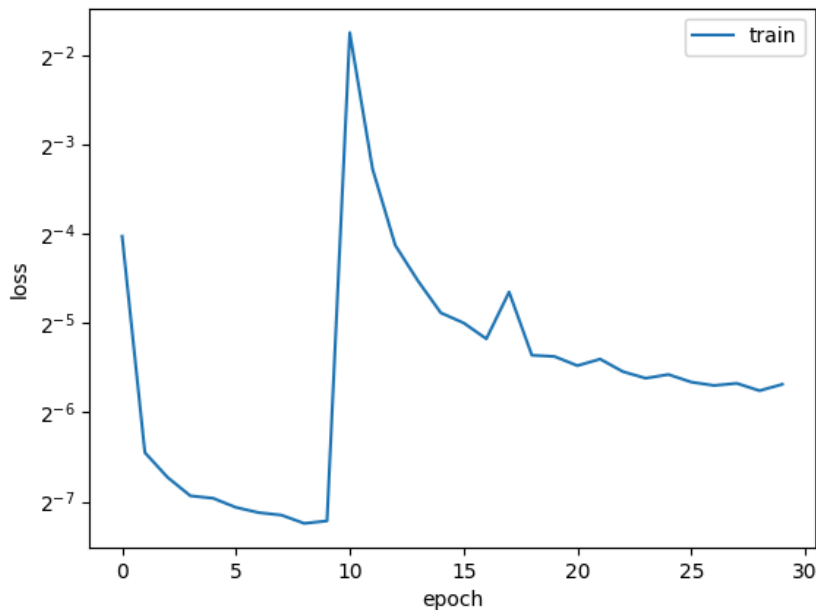
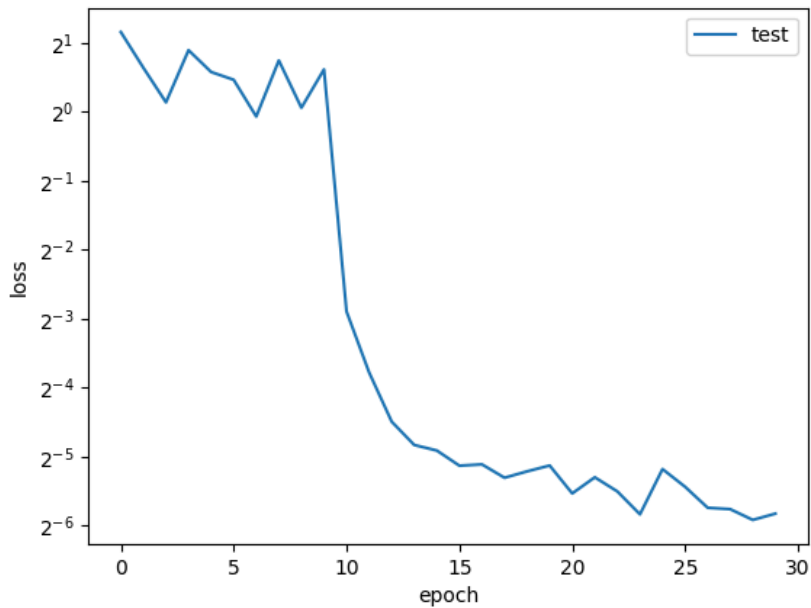


Рис. 7 Графики ошибок на train и test выборке. Здесь виден скачок на 10 эпохе. Он связан с тем, что вплоть до 10-й эпохи использовалась методика teacher forcing, когда рекуррентная сеть на вход получает размеченные данные, что позволяет ускорить обучение.

Эта модель достигла ошибки **0.018** на выборке обучения и **0.016** на тестовой выборке по метрике [chamfer distance](#). Ошибка считалась на 500 вершинах.

Планы на будущее

- Качественно обучить модель на большем датасете и на большем количестве вершин, чтобы действительно улучшать качество 3D модели.
- Вторая нейронная сеть для более глобальной работы с 3D моделью. Текущая реализация хороша в улучшении маленьких локальных фрагментов модели, но не в исправлении сильных погрешностей модели. Это можно видеть на примере скана дома (см. Рис. 3), где добавляются новые облака лишних точек рядом с моделью.
- Работать не только с облаками вершин, но и с гранями.
- Также работать с текстурами моделей.
- Расширить поле действия нейронной сети еще и на [NeRF](#) (Neural Radiance Fields)