



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ \_\_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_

КАФЕДРА \_\_\_\_\_СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ (ИУ5)\_\_\_\_\_

# РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

**НА ТЕМУ:**

***Кластеризация данных LiDAR***

Студент \_\_\_\_\_ИУ5-61Б\_\_\_\_\_  
(Группа)

\_\_\_\_\_  
(Подпись, дата) \_\_\_\_\_**Абуховский И. А.**\_\_\_\_\_  
(И.О.Фамилия)

Руководитель

\_\_\_\_\_  
(Подпись, дата) \_\_\_\_\_**Канев А. И.**\_\_\_\_\_  
(И.О.Фамилия)

Москва, 2023 г.



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет имени  
Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой ИУ5  
(Индекс)

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ г.  
(И.О.Фамилия)

**З А Д А Н И Е**  
**на выполнение научно-исследовательской работы**

по теме Кластеризация данных LiDAR

Студент группы ИУ5-61Б

Абуховский Иван Александрович  
(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)

Исследовательская

Источник тематики (кафедра, предприятие, НИР) НИР

График выполнения НИР: 25% к \_\_\_\_ нед., 50% к \_\_\_\_ нед., 75% к \_\_\_\_ нед., 100% к \_\_\_\_ нед.

**Техническое задание Исследовать использование методов машинного обучения для решения задачи сегментации деревьев из облака точек и провести визуализацию полученных данных в движке Unity**

**Оформление научно-исследовательской работы:**

Расчетно-пояснительная записка на 24 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

Дата выдачи задания «07» февраля 2023 г.

Руководитель НИР

Студент

Канев А.И.  
(Подпись, дата) (И.О.Фамилия)  
Абуховский И.А.  
(Подпись, дата) (И.О.Фамилия)

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1 Постановка задачи .....	5
2 Описание данных, используемых методов и метрик .....	7
3 Выполнение примера по варианту .....	9
4 Экспорт полученных данных и создание 3D модели.....	16
ЗАКЛЮЧЕНИЕ.....	22
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	23

## **ВВЕДЕНИЕ**

LiDAR (Light Detection and Ranging «обнаружение и определение дальности с помощью света») — технология измерения расстояний путем излучения света и замера времени возвращения этого отражённого света на приёмник. Сегодня LiDAR применяется в широком спектре задач, таких как исследования атмосферы, строительство и горное дело, морские технологии, беспилотный транспорт, промышленная и сервисная робототехника.

Результатом проведения съёмки LiDAR является облако точек, отображающее форму отсканированных поверхностей и объектов. Задача выделения отдельных объектов из облака точек называется сегментацией. Решение этой задачи производится путём объединения точек, описывающих определённый объект, в отдельную группу, являющуюся отображением этого объекта. Данная задача решается в разработке технологий машинного обучения.

Целью данной работы является исследование применения методов машинного обучения для сегментации деревьев из облака точек и последующая визуализация полученных результатов в движке Unity. В данной работе решаются задачи сегментации деревьев из облака точек с использованием методов машинного обучения, оценки качества сегментации деревьев из облака точек, 3D-моделирования облаков точек и их визуализация в движке Unity.

## 1 Постановка задачи

В результате лидарной съёмки деревьев было получено множество точек. Необходимо распределить точки каждого объекта по отдельным кластерам для того, чтобы произвести сегментацию деревьев.

Применение LiDAR для сегментации деревьев в лесных массивах является особенно актуальным в современном тренде развития экологии и природоохранных технологий.

Съёмка деревьев с помощью LiDAR предназначена для того, чтобы собирать, обрабатывать, учитывать, анализировать и прогнозировать состояние природного лесного массива. Исследователи в области экологии и лесного хозяйства отслеживают размер и рост отдельных деревьев в лесу, чтобы сделать вывод о состоянии деревьев. Обычно количественные показатели извлекаются с использованием ручных методов, таких как измерительные ленты, что является устаревшим и малоэффективным методом [1]. Применение LiDAR значительно упрощает данную задачу, позволяя оперативно отслеживать состояние большого количество деревьев.

В случае городских районов классификация пород деревьев с помощью LiDAR привлекает все большее внимание для исследований безопасности, моделирования шума и экологического анализа, поскольку деревья играют важнейшую роль в городских экосистемах для поддержания качества окружающей среды, эстетической красоты городского ландшафта и социального обслуживания жителей. По мере быстрого роста городов городские леса все чаще вытесняются инфраструктурой. Поэтому муниципальные власти стремятся контролировать развитие вблизи зоны зеленого пояса с использованием карт растительного покрова [2].

Также LiDAR позволяет произвести быстрый, точный и полный анализ лесного массива, подробно описать его структуру, оценить особенности ландшафта, в том числе густоту расположения деревьев и другие сведения, существенные для областей ведения лесохозяйства и проведения исследований в экологической сфере [3].

Применение LiDAR для сканирования лесного массива позволяет эффективно исследовать лесные ресурсы, их качество, количество и доступность для перемещения. Данная технология способствует разработке транспортных сетей на основе изучения особенностей ландшафта, что значительно повышает коэффициент полезного действия доставки природных ресурсов и улучшает ведение лесного хозяйства путём внедрения в его процессы современных навигационных методов, повышает рентабельность лесохозяйственных предприятий [4].

Решение задачи выделения из облака точек деревьев путём сегментации позволяет разработать устройства наблюдения за экосистемой лесного массива. Изменение состояния леса зависит от множества факторов, к которым относятся пожароустойчивость и иммунитет флоры, её здоровье и биоразнообразие. Сбор и анализ информации воздействующих на лес экологических факторов позволяют прийти к необходимости решения проблемы поиска средств по предотвращению и исправлению пагубного человеческого влияния на окружающую среду, ослаблению его последствий. Таким образом, технология LiDAR обеспечивает современных учёных возможностью на основе большого количества точных данных делать соответствующие выводы по текущему состоянию экологической среды, быстро принимать эффективные решения по её улучшению [5].

В эпоху развития метавселенных и цифровых двойников крайне важной и актуальной задачей станет визуализация данных, полученных с LiDAR. Выполняя такую задачу, человечество станет на шаг ближе к виртуализации реальных пространств, в нашем случае лесных массивов [7].

## 2 Описание данных, используемых методов и метрик

Исходные данные представляют собой двумерный массив координат из 10000 точек (Рисунок 1), полученных в результате съемки лидаром трёх близких друг к другу деревьев (вариант 5). Точки, принадлежащие стволу, распределены с более высокой плотностью, чем точки, принадлежащие листве деревьев, что может сказаться на точности методов машинного обучения.

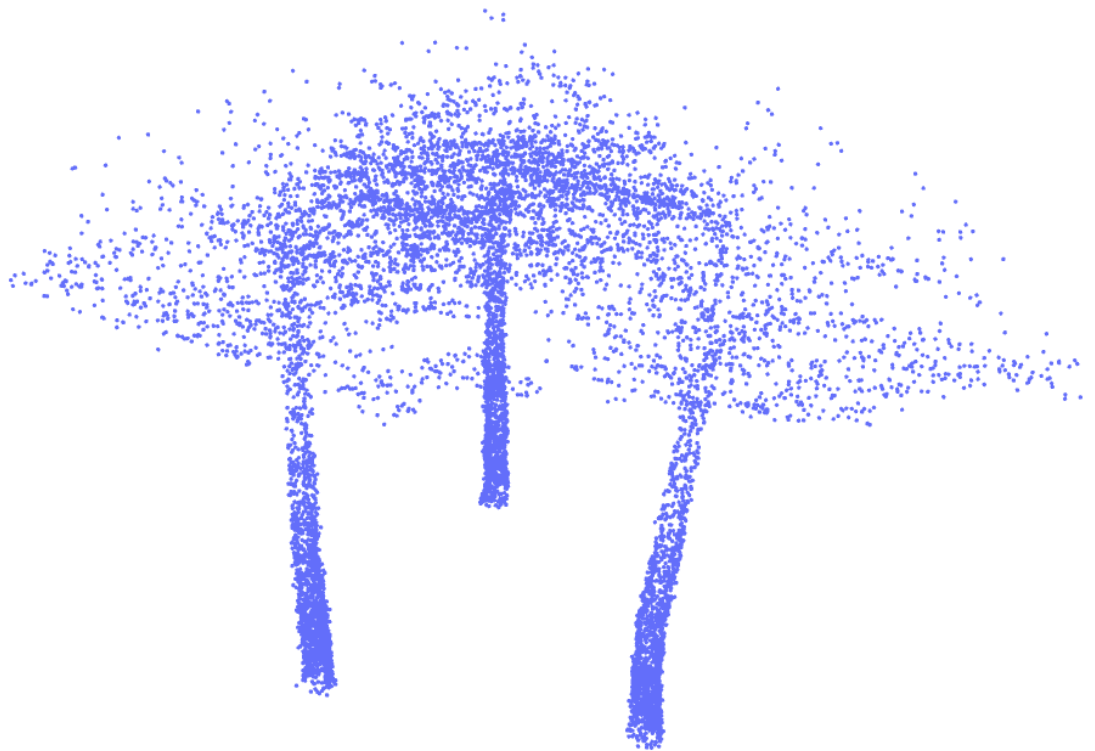


Рисунок 1 – Облако точек

Для исследования данных был выбран метод DBSCAN. Он представляет собой плотностный алгоритм пространственной кластеризации с присутствием шума, для работы которого необходимы 2 параметра:  $\epsilon$  – радиус  $\epsilon$ -окрестности точки и  $\min\_pts$  – количество соседей. Алгоритм выбирает корневую точку и производит обход по её соседям. Точка – корневая, если в её  $\epsilon$ -окрестности находится минимум  $\min\_pts$  точек. Также, если соседняя точка удовлетворяет этим условиям, её соседей добавляют в обход, что обеспечивает кластеризацию точек. Метод обозначает выбросами (или шумами) те некорневые точки, которые

в своей  $\varepsilon$ -окрестности не имеют корневых соседей. Для оценки качества результатов кластеризации выбрана метрика Silhouette Score.

Для генерации облака точек из массива координат и последующей её визуализации используется Open3D – библиотека Python с открытым исходным кодом для обработки трехмерных данных. Она предоставляет множество функций для обработки и визуализации трехмерных данных, включая работу с облаками точек. Библиотека Open3D предоставляет простой интерфейс API, который позволяет работать с 3D-данными и сравнивать их в различных форматах файлов.

Для создания 3D моделей из облаков точек используется ПО AdTree [6]. Данная программа используется для точного и автоматического воссоздания детальных трехмерных моделей деревьев по данным лазерного сканирования, в том числе с помощью LiDAR. Принцип работы следующий: сначала извлекается начальный скелет дерева из входного облака точек путем создания минимального связующего дерева с помощью алгоритма кратчайшего пути Дейкстры, затем начальный скелет дерева обрезается путем итеративного удаления лишних компонентов, после чего с помощью оптимизационного подхода подбирается последовательность цилиндров для аппроксимации геометрии ветвей дерева. Одним из ключевых преимуществ AdTree является его способность работать с крупными базами данных в режиме реального времени.

Для работы с данными используются модули Python: Pandas - библиотека, применяемая для анализа и манипулирования данными с открытым исходным кодом, и NumPy - открытая бесплатная библиотека для работы с многомерными массивами.



### 3 Выполнение примера по варианту

Для проведения сегментации облака точек были применены алгоритм кластеризации DBSCAN и функции предоставления данных в формате `pcd`, их цветного графического отображения. При использованных параметрах  $\text{eps} = 0,3$  и  $\text{min\_pts} = 200$  было распознано 3 объекта, однако значительная часть точек была ошибочно отнесена к шумам, что не удовлетворяет поставленной задаче. (Рисунок 2)

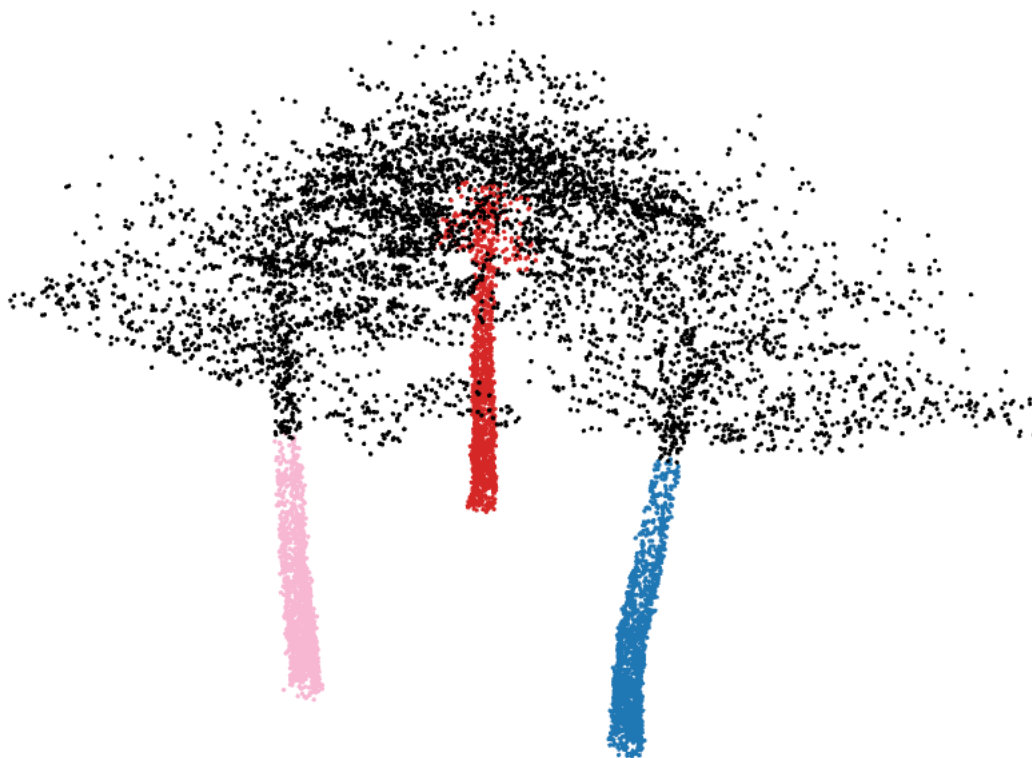


Рисунок 2 – Результат сегментации при  $\text{eps} = 0,3$ ,  $\text{min\_pts} = 200$

Для выявления оптимальных параметров  $\text{eps}$  и  $\text{min\_pts}$  была произведена процедура выявления числа кластеров для диапазона значений  $\text{eps}$  от 0,01 до 1 с шагом 0,01 и для диапазона значений  $\text{min\_pts}$  от 2 до 600 с шагом 2, в результате которой был создан набор данных (Рисунок 3, Рисунок 4).

### Прогон первый

```
Ввод [90]: %%time
eps_values_vars = np.arange(0.01,1,0.01)
min_pts_vars = np.arange(2,600,2)

dbscan_params = list(product(eps_values_vars, min_pts_vars))
number_of_clusters = []
eps_values = []
min_pts = []
for p in dbscan_params:
    dbscan_cluster = DBSCAN(eps=p[0], min_samples=p[1]).fit(X)
    eps_values.append(p[0])
    min_pts.append(p[1])
    number_of_clusters.append(len(np.unique(dbscan_cluster.labels_)))

res_data = list(zip(number_of_clusters,eps_values, min_pts))
res_df = pd.DataFrame(res_data, columns=['number_of_clusters', 'eps_values', 'min_pts'])
res_df
```

CPU times: user 1h 9min 49s, sys: 13.3 s, total: 1h 10min 2s  
Wall time: 1h 9min 57s

Рисунок 3 – код создания набора данных подсчёта кластеров

```
Ввод [21]: pd.set_option("display.max_rows", None, "display.max_columns", None)
res_df
```

Out[21]:

	number_of_clusters	eps_values	min_pts
0	729	0.01	2
1	58	0.01	4
2	2	0.01	6
3	1	0.01	8
4	1	0.01	10
5	1	0.01	12
6	1	0.01	14
7	1	0.01	16
8	1	0.01	18
9	1	0.01	20
10	1	0.01	22

Рисунок 4 – набор данных подсчета кластеров

Из всех записей были отобраны только те, количество кластеров в которых оказалось равным 4 (Рисунок 5).

```
Ввод [22]: res_df_4 = res_df[res_df['number_of_clusters']==4]
res_df_4
```

Out[22]:

	number_of_clusters	eps_values	min_pts
464	4	0.04	36
617	4	0.05	44
620	4	0.05	50
623	4	0.05	56
767	4	0.06	46
768	4	0.06	48
773	4	0.06	58
774	4	0.06	60
776	4	0.06	64
777	4	0.06	66
778	4	0.06	68

Рисунок 5 – Отфильтрованный набор данных

При наименьших значениях параметров  $\text{min\_pts} = 36$  и  $\text{eps} = 0,04$  большинство точек относят к шумам, т.к. кластеры состоят из небольшого количества тесно сгруппированных точек. Также кластеризация производится некорректно (Рисунок 6).



Рисунок 6 – Результат сегментации при минимальных  $\text{eps}$  и  $\text{min\_pts}$

Увеличивая значения гиперпараметров, можно наблюдать картину покрытия кластерами всё более широких областей стволов деревьев, т.к. в них точки расположены наиболее близко друг к другу, в отличие от области листвы, точки которой рассматриваются как шумы. (Рисунок 7, Рисунок 8).



Рисунок 7 – Результаты сегментации при  $\text{eps} = 0,19$ ,  $\text{min\_pts} = 298$



Рисунок 8 – Результаты сегментации при  $\text{eps} = 0,27$ ,  $\text{min\_pts} = 160$

Наиболее точно осуществляется процесс кластеризации при значениях гиперпараметров  $\text{eps} = 0,521$  и  $\text{min\_pts} = 298$ . При таких параметрах точки каждого дерева выделяются в отдельные кластеры, тем не менее остаётся некоторое значение точек в листве, которые рассматриваются как шумы (Рисунок 9).

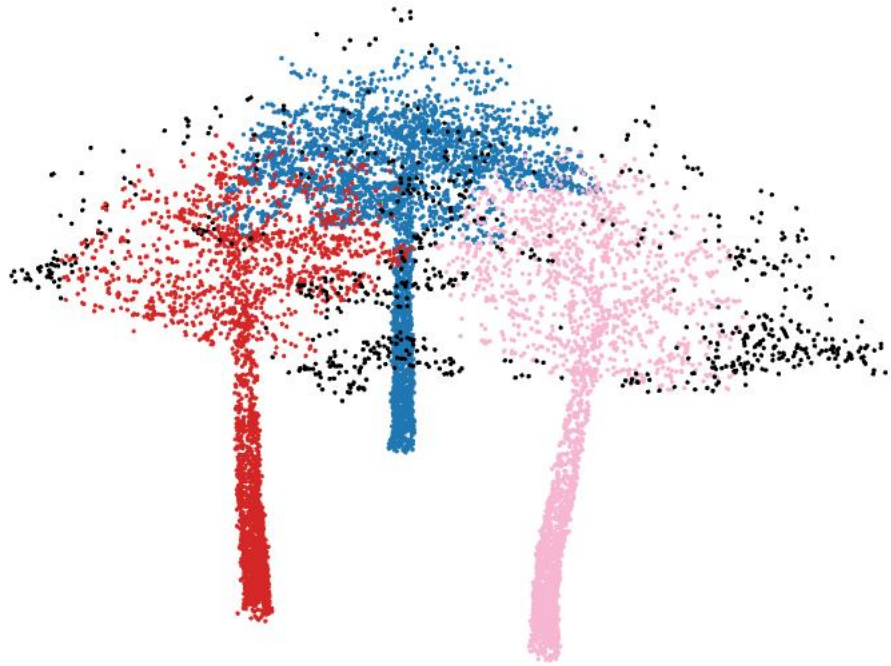


Рисунок 9 – Результаты сегментации при  $\text{eps} = 0,521$ ,  $\text{min\_pts} = 298$

При более высоких значениях гиперпараметров  $\text{eps} = 0,53$  и  $\text{min\_pts} = 454$  шумов становится значительно больше (Рисунок 10).

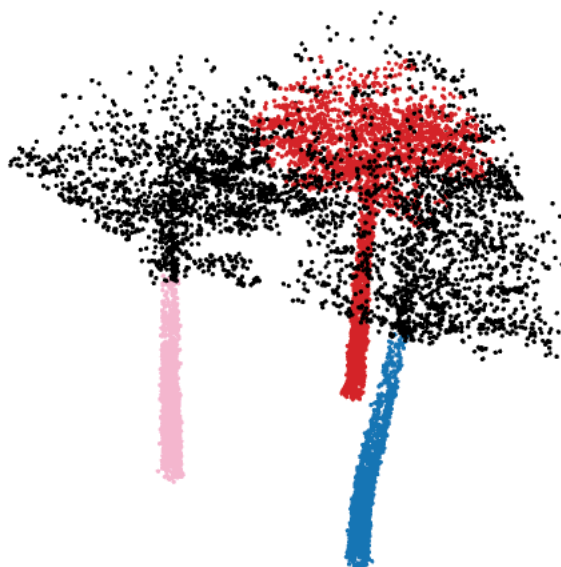


Рисунок 10 – Результаты сегментации при  $\text{eps} = 0,53$ ,  $\text{min\_pts} = 454$

Для выявления наиболее оптимальных значений гиперпараметров была использована метрика Silhouette Score для пар  $\text{eps}$ ,  $\text{min\_pts}$ , ограниченных небольшим интервалом значений  $\text{eps}$  от 0,51 до 0,54 с шагом 0,001 и интервалом значений  $\text{min\_pts}$  от 280 до 320 с шагом 2 (Рисунок 11, Рисунок 12).

Проанализировав полученный набор данных, можно сделать вывод, что наиболее оптимальные параметры  $\text{eps} = 0,521$  и  $\text{min\_pts} = 298$  определяются максимальным значением метрики Silhouette Score, равным 0,361839. Однако визуально результаты сегментации не являются превосходными (Рисунок 13).

```
Ввод [143]: from sklearn import metrics

Ввод [144]: %%time
eps_values_vars = np.arange(0.51,0.54,0.001)
min_pts_vars = np.arange(280,320,2)

dbscan_params = list(product(eps_values_vars, min_pts_vars))
number_of_clusters = []
eps_values = []
min_pts = []
silhouette = []
for p in dbscan_params:
    dbscan_cluster = DBSCAN(eps=p[0], min_samples=p[1]).fit(X)
    eps_values.append(p[0])
    min_pts.append(p[1])
    number_of_clusters.append(len(np.unique(dbscan_cluster.labels_)))
    silhouette.append(metrics.silhouette_score(X, dbscan_cluster.labels_))
res_data = list(zip(number_of_clusters,eps_values, min_pts, silhouette))
res_dfl = pd.DataFrame(res_data, columns=['number_of_clusters', 'eps_values', 'min_pts', 'silhouette'])
res_dfl.to_csv('output_silhouette.csv')

CPU times: user 20min 12s, sys: 6min 51s, total: 27min 3s
Wall time: 11min 39s
```

Рисунок 11 - код создания набора данных подсчета значения метрики

```

Ввод [145]: silhouette_df = res_df[res_df['number_of_clusters']==4]
pd.set_option("display.max_rows", None, "display.max_columns", None)
silhouette_df.sort_values("silhouette")

```

45	4	0.512	290	0.361524
107	4	0.515	294	0.361528
168	4	0.518	296	0.361533
66	4	0.513	292	0.361544
149	4	0.517	298	0.361554
397	4	0.529	314	0.361570
418	4	0.530	316	0.361573
230	4	0.521	300	0.361584
209	4	0.520	298	0.361588
398	4	0.529	316	0.361588
250	4	0.522	300	0.361776
229	4	0.521	298	0.361839

Рисунок 12 – отфильтрованный набор данных подсчета значения метрики



Рисунок 13 – Результаты сегментации при  $\text{eps} = 0,521$ ,  $\text{min\_pts} = 298$

## 4 Экспорт полученных данных и создание 3D модели

После сегментации исходных данных мы получаем три группы точек, формирующие наши деревья. Для дальнейшей визуализации исходных данных сформируем 3D модели каждого дерева (Рисунок 14).

```
Ввод [19]: for i in range(max_label):  
            dataset = pd.DataFrame({'x': obj_points[i][:, 0], 'y': obj_points[i][:, 1], 'z': obj_points[i][:, 2]})  
            np = dataset.to_numpy()  
            pcd1 = o3d.geometry.PointCloud()  
            pcd1.points = o3d.utility.Vector3dVector(np)  
            o3d.io.write_point_cloud(f"./tree_{i}.ply", pcd1)
```

Рисунок 14 – Формирование и визуализация получившихся множеств точек

В первую очередь координаты точек каждого из деревьев (x, y, z) конвертируются в формат массива NumPy для дальнейшего построения облака точек необходимого расширения (Рисунок 14, 2 и 3 строки). Затем мы создаём переменную, которая будет хранить облако точек, после чего заполняем её координатами из созданного выше массива NumPy и сохраняем в качестве облака точек в файлы с названиями “tree\_N.ply”, где N – порядковый номер дерева (Рисунок 14, строки 4-6). Методы из данных строк относятся к библиотеке Open3D.

После выполнении данных команд с помощью функции `o3d.visualization.draw_geometries` выведем окна с визуализацией трёх отдельных множеств точек – наших сегментированных деревьев (Рисунок 15, Рисунок 16, Рисунок 17).



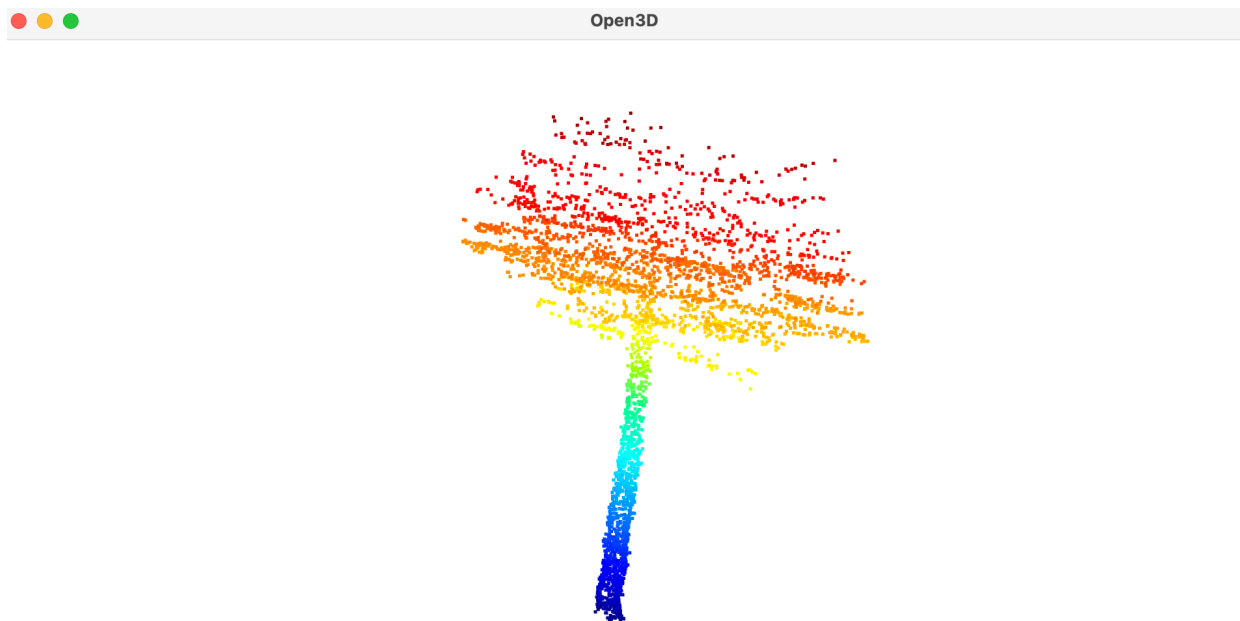


Рисунок 15 – Дерево 1 в виде облака точек

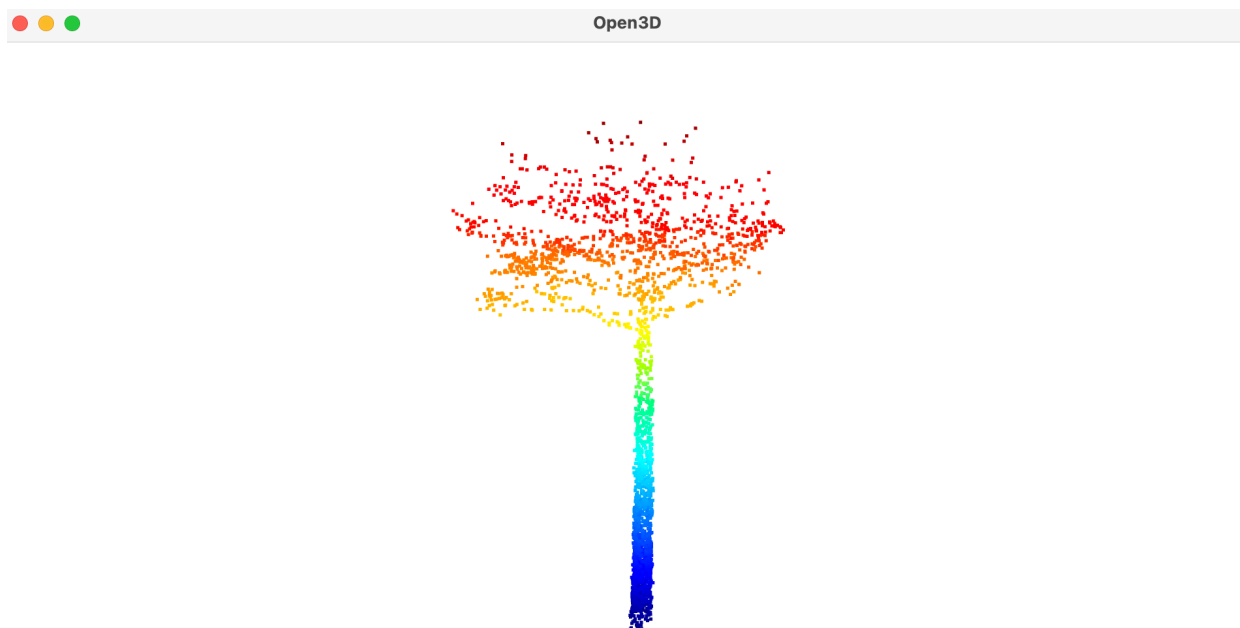


Рисунок 16 – Дерево 2 в виде облака точек

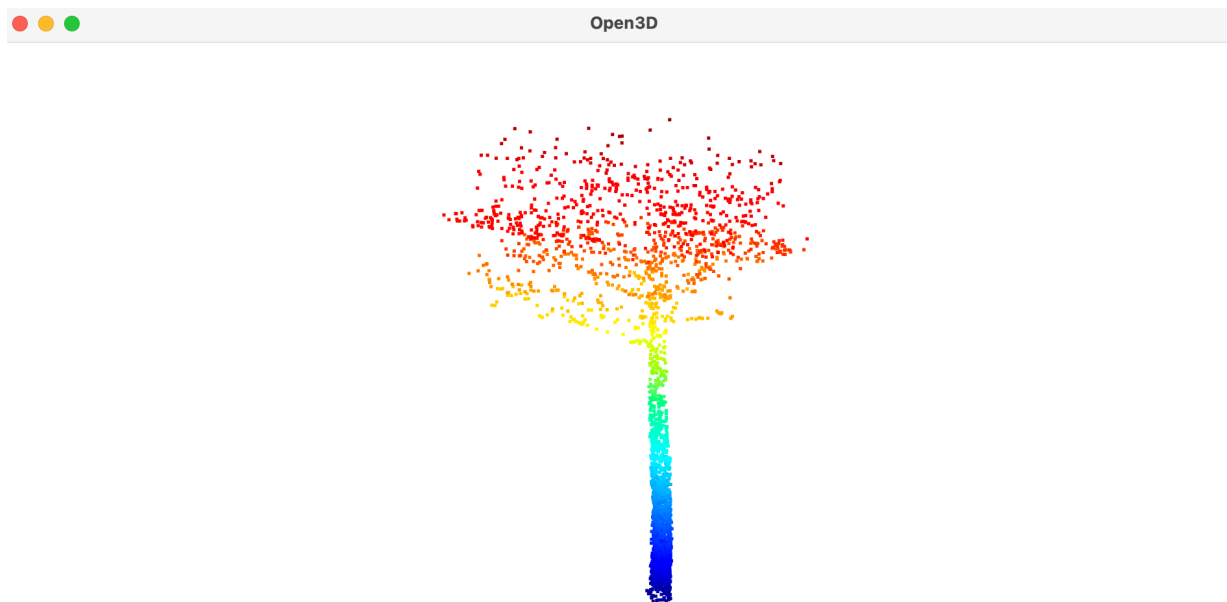


Рисунок 17 – Дерево 3 в виде облака точек

После визуализации в нашей файловой системе появятся три файла с названиями “tree\_N.ply”, где N – порядковый номер дерева. С помощью онлайн конвертера форматов <https://imagnetostl.com/convert/file/ply/to/xyz> переведём выходные файлы в формат XYZ, который принимается ПО для создания 3D модели из облака точек [6].

В ПО AdTree воспользуемся функцией Open и откроем файлы с расширением .ply, которые создали выше (Рисунок 18).

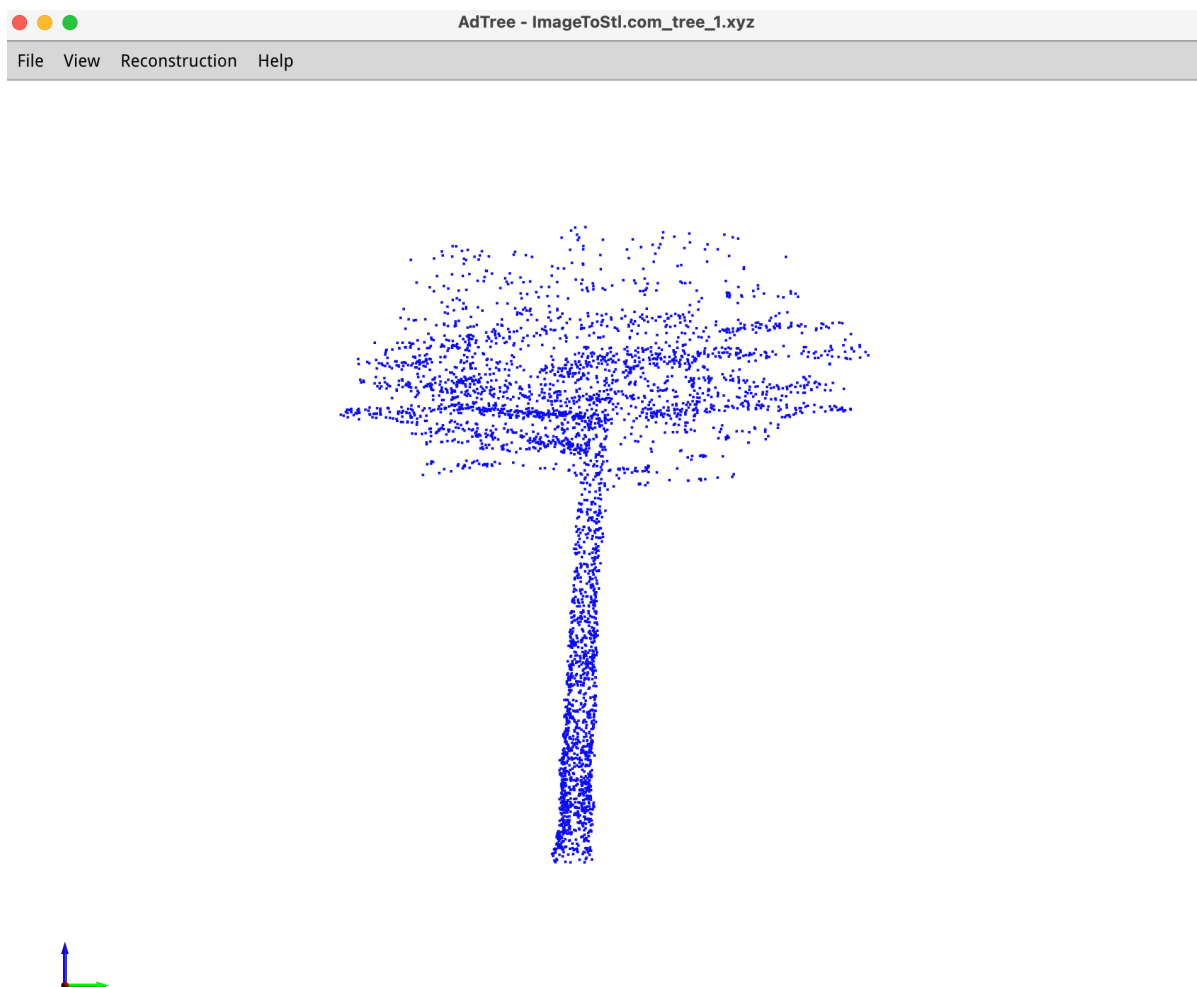


Рисунок 18 – Дерево 1 в виде облака точек в ПО для создания модели

С помощью функции Reconstruct Skeleton получим 3D модель и экспортируем её в формате .obj, применяя функцию Save Branches (Рисунок 19).

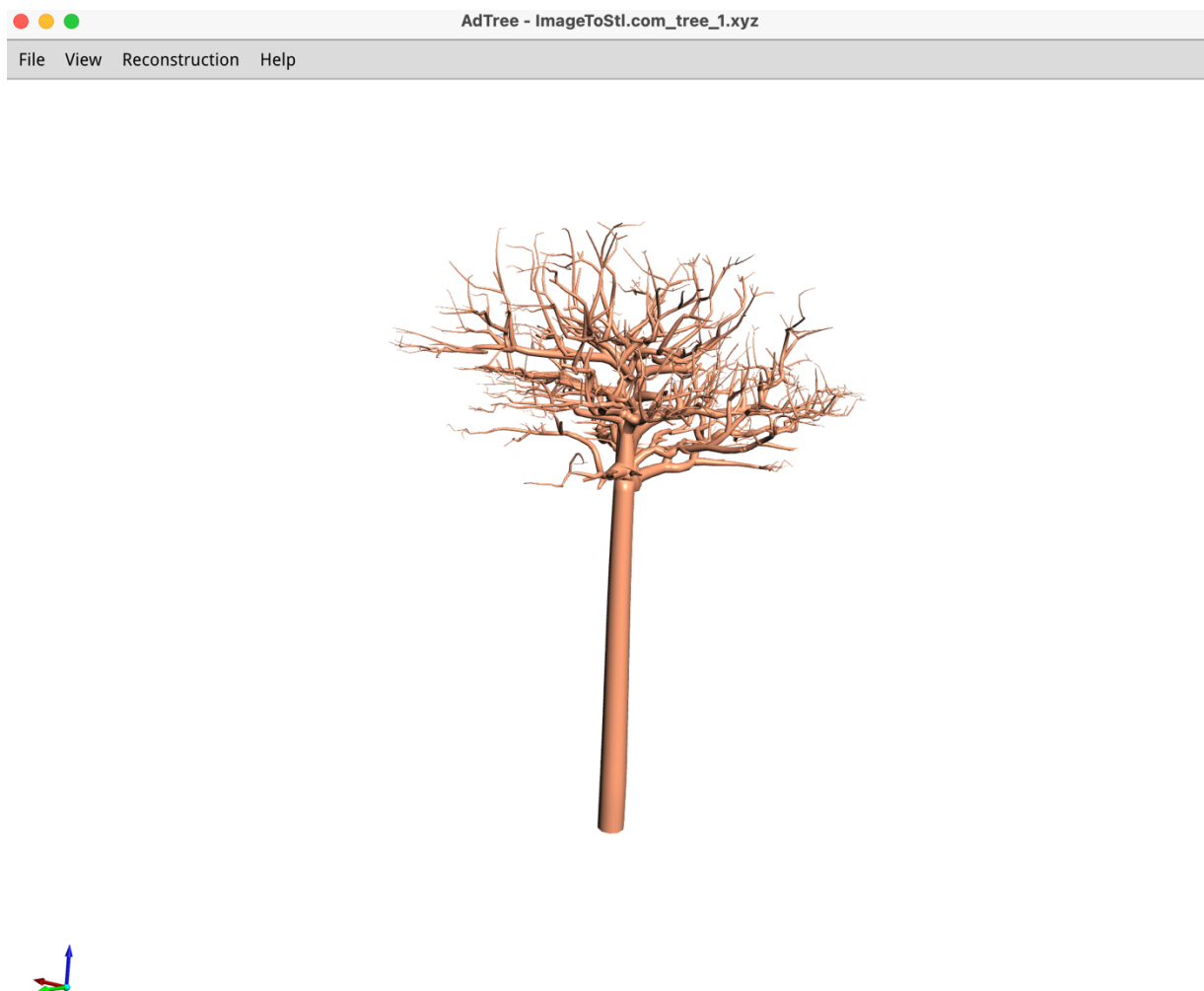


Рисунок 19 – 3D модель дерева 1 в ПО для создания модели

При желании можно добавить листья на дерево с помощью функции Add Leaves (Рисунок 20).

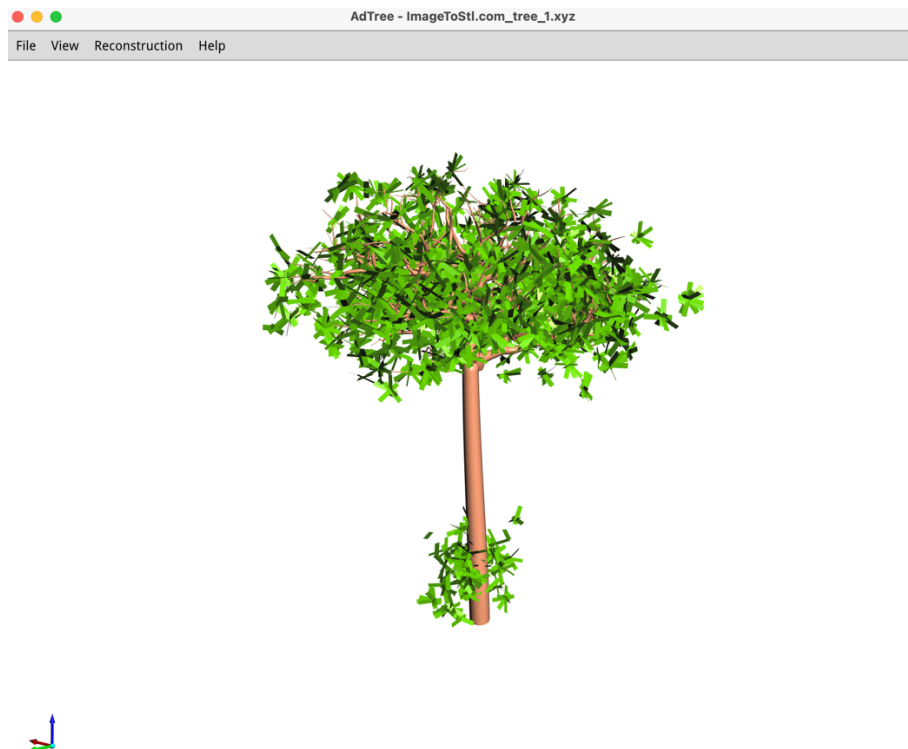


Рисунок 20 – 3D модель дерева 1 в ПО для создания модели с листвой

Легко наблюдать, что из-за неидеальной работы ПО листья добавляются и на ствол дерева, поэтому добавлять листву без особенной необходимости не стоит.

Импортируем финальные модели в Unity, применяя встроенный в движок модуль импорта FBX моделей (built-in Unity FBX Importer) (Рисунок 21).

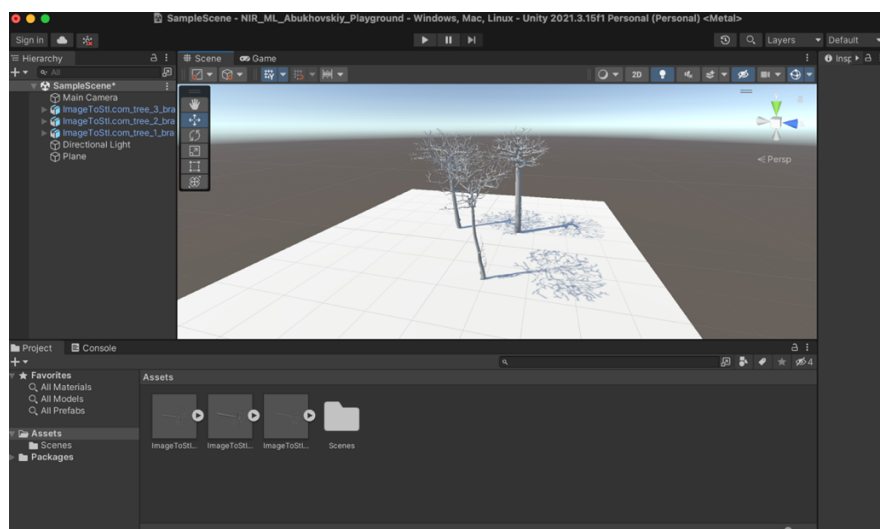


Рисунок 21 – 3D модели деревьев в Unity

## **ЗАКЛЮЧЕНИЕ**

В результате выполнения научно-исследовательской работы были выполнены следующие задачи:

1. Был осуществлён процесс сегментации деревьев из облака точек с использованием метода DBSCAN, в результате которого было получено три кластера, соответствующих каждому дереву.
2. Для оценки качества кластеризации была применена метрика Silhouette Score. Наилучшие результаты были достигнуты со значением метрики, равным 0,361839.
3. Была создана и успешно перенесена в Unity 3D модель каждого из сегментированных деревьев.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Alexander Proudman, Milad Ramezani, Maurice Fallon. Online Estimation of Diameter at Breast Height (DBH) of Forest Trees Using a Handheld LiDAR: [Электронный ресурс]. // arXiv.org. 2023. Дата обновления: 31.01.2023. URL: <https://arxiv.org/pdf/2108.01552.pdf> (Дата обращения: 14.04.2023).
2. Haiyan Guana, Yongtao Yu, Zheng Ji, Jonathan Li, and Qi Zhang. Deep learning-based tree classification using mobile LiDAR data: [Электронный ресурс]. // uwaterloo.ca. 2023. Дата обновления: 27.01.2022. URL: [https://uwaterloo.ca/geospatial-intelligence/sites/default/files/uploads/files/2015\\_guan\\_yu\\_ji\\_li\\_zhang\\_rsl.pdf](https://uwaterloo.ca/geospatial-intelligence/sites/default/files/uploads/files/2015_guan_yu_ji_li_zhang_rsl.pdf) (Дата обращения: 14.04.2023).
3. Georgi Tinchev, Adrian Penate-Sanchez, Maurice Fallon. Real-time LIDAR localization in natural and urban environments: [Электронный ресурс]. // arXiv.org. 2023. Дата обновления: 31.01.2023. URL: <https://arxiv.org/abs/2301.13583> (Дата обращения: 14.04.2023).
4. Jonathan Williams, Carola-Bibiane Schonlieb, Tom Swinfield, Juheon Lee, Xiaohao Cai, Lan Qie, David A. Coomes. Three-dimensional Segmentation of Trees Through a Flexible Multi-Class Graph Cut Algorithm (MCGC): [Электронный ресурс]. // arXiv.org. 2023. Дата обновления: 20.03.2019. URL: <https://arxiv.org/abs/1903.08481> (Дата обращения: 14.04.2023).
5. Juan Castorena, L. Turin Dickman, Adam J. Killebrew, James R Gattiker, Rod Linn, E. Louise Loudermilk. Automated Structural-level Alignment of Multi-view TLS and ALS Point Clouds in Forestry: [Электронный ресурс]. // arXiv.org. 2023. Дата обновления: 25.02.2023. URL: <https://arxiv.org/abs/2302.12989> (Дата обращения: 14.04.2023).
6. Shenglan Du, Roderik Lindenbergh, Hugo Ledoux, Jantien Stoter, Liangliang Nan. AdTree: Accurate, Detailed, and Automatic Modelling of Laser-Scanned Trees: [Электронный ресурс]. // 3d.bk.tudelft.nl. 2023. Дата обновления: 04.09.2019. URL:

[https://3d.bk.tudelft.nl/liangliang/publications/2019/adtree/AdTree\\_RS-2019.pdf](https://3d.bk.tudelft.nl/liangliang/publications/2019/adtree/AdTree_RS-2019.pdf)

(Дата обращения: 14.05.2023).

7. Yanchao Liu, Jianwei Guo, Bedrich Benes, Oliver Deussen, Xiaopeng Zhang, Hui Huang. TreePartNet: neural decomposition of point clouds for 3D tree reconstruction: [Электронный ресурс]. // dl.acm.org. 2023. Дата обновления: 10.12.2021. URL: <https://kops.uni-konstanz.de/server/api/core/bitstreams/1a88047e-f533-4f5f-ba2b-23fad6e2ee87/content> (Дата обращения: 14.05.2023).