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__ _____ __**Зелинский Д. М.**__
(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель _____ _____**Канев А. И.**___
(Подпись, дата) (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)

_____ _____
(И.О.Фамилия)

« _____ » _____ 20 ____ г.

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

по теме _____Кластеризация данных LiDAR_____

_____

_____

Студент группы ____ИУ5-61Б____

_____Зелинский Даниил Михайлович_____
(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

_____

_____

Оформление научно-исследовательской работы:

Расчетно-пояснительная записка на _30_ листах формата А4.
Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

_____

_____

Дата выдачи задания «07» февраля 2023 г.

Руководитель НИР _____ **Канев А.И.**
(Подпись, дата) (И.О.Фамилия)

Студент _____ **Зелинский Д.М.**
(Подпись, дата) (И.О.Фамилия)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

LiDAR – оптическая технология определения дальности путём нацеливания на объект или поверхность с помощью лазера, измерения времени, затраченного на получение отражённого от поверхности сигнала. Сегодня лидар применяется в широком спектре задач, начиная с подводной съёмки, надземного сканирования, навигации, строительства, горного дела, и заканчивая исследованиями природного ландшафта, управлением лесным хозяйством.

Результатом проведения лидарной съёмки является облако точек, отображающее форму поверхностей и объектов, отсканированных во время съёмки. 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, автопилота, системы распознавания лиц, систем картографии.

Целью данной работы является исследование применения методов машинного обучения для сегментации деревьев из облака точек. В данной работе решаются задачи сегментации деревьев из облака точек с использованием методов машинного обучения, оценки качества сегментации деревьев из облака точек.

1 Постановка задачи

В результате лидарной съёмки деревьев было получено множество точек, из которого необходимо сегментировать эти деревья, то есть выделить точки каждого дерева в отдельные кластеры, им соответствующие.

В современном мире, когда важность экологии неоспоримо возрастает, решение поставленной задачи приобретает всё большую востребованность.

Система лидарной съёмки деревьев предназначена для того, чтобы собирать, обрабатывать, учитывать, анализировать и прогнозировать состояние природного лесного массива. Анализируя полученные при съёмке лидаром результаты, можно выделить ценные сведения о лесных кадастрах – экономических, экологических и других качественных и количественных характеристиках леса [1] [2].

Применение лидара 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 в областях ведения лесохозяйства и проведения исследований в экологической сфере [2] [3].

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 лидара 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 [4].

Применение лидара для сканирования лесного массива позволяет эффективно исследовать лесные ресурсы, их качество, количество и доступность для перемещения. Также 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 [5].

Сканирование деревьев осуществляется с помощью воздушного лидара, к которому относятся съёмка с самолёта, с беспилотного летательного аппарата, а также с применением ручного или наземного лидара.

2 Описание данных, используемых методов и метрик

Используемые данные представляют собой двумерный массив координат 10000 точек (Рисунок 2.1), полученных в результате съемки лидаром четырёх близких друг к другу деревьев (вариант 6). Точки, принадлежащие стволу, распределены с более высокой плотностью, чем точки, принадлежащие листве деревьев, что может сказаться на точности методов машинного обучения.

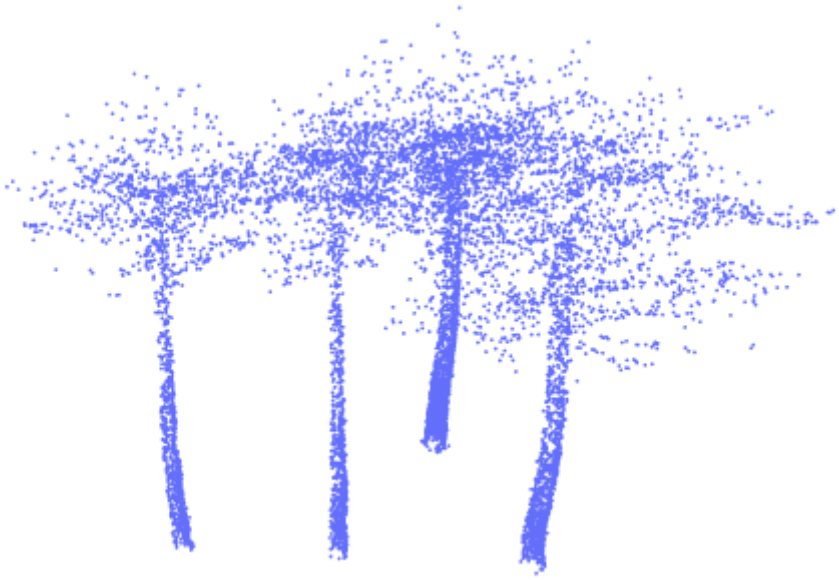

Рисунок 2.1 – Облако точек

Для исследования данных был выбран метод DBSCAN. Он представляет собой плотностный алгоритм пространственной кластеризации с присутствием шума, для работы которого требуется 2 параметра: eps – радиус ε-окрестности точки и количество соседей min_pts. Алгоритм выбирает корневую точку и производит обход по её соседям. Точка – корневая, если в её ε-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 те некорневые точки, которые в своей ε-окрестности не имеют корневых соседей. Для оценки качества результатов кластеризации выбрана метрика Silhouette Score.

3 Выполнение примера по варианту

Для проведения сегментации облака точек были применены алгоритм кластеризации DBSCAN и функции предоставление данных в формате pcd, их цветного графического отображения. При использованных параметрах eps = 0,5 и min_pts = 250 было распознано 5 объектов, что не удовлетворяет поставленной задаче. (Рисунок 3.1)

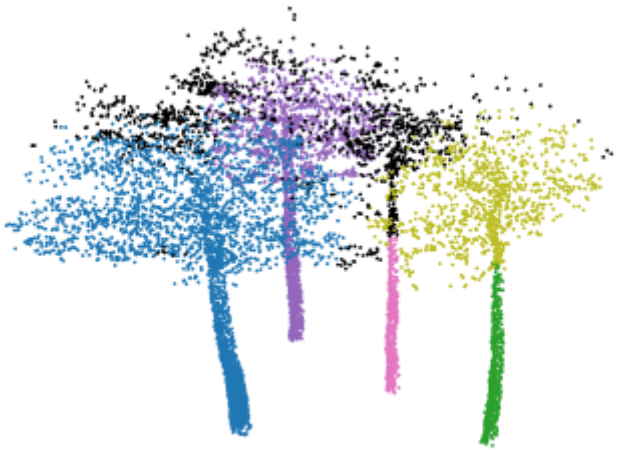

Рисунок 3.1 – Результат сегментации при eps = 0,5, min_pts = 250

Для выявления оптимальных параметров eps и min_pts была произведена процедура выявления числа кластеров для диапазона значений eps от 0,01 до 1 с шагом 0,01 и для диапазона значений min_pts от 2 до 300 с шагом 2, в результате которой был создан набор данных (Рисунок 3.2, Рисунок 3.3).

```
In [15]: %%time
eps_values_vars = np.arange(0.01,1,0.01)
min_pts_vars = np.arange(2,300,2)

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
res_df.to_csv('output.csv')

CPU times: total: 1h 40min 51s
Wall time: 1h 43min 5s
```

Рисунок 3.2 – код создания набора данных подсчёта кластеров

```
In [16]: pd.set_option("display.max_rows", None, "display.max_columns", None)
         res_df
```

Out[16]:

	number_of_clusters	eps_values	min_pts
0	643	0.01	2
1	47	0.01	4
2	3	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 3.3 – набор данных подсчета кластеров

Из всех записей были отобраны только те, clusters_cnt которых оказался равен 5 (Рисунок 3.4).

```
In [24]: res_df_4 = res_df[res_df['number_of_clusters']==5]
         res_df_4
```

Out[24]:

	number_of_clusters	eps_values	min_pts
308	5	0.03	22
463	5	0.04	34
617	5	0.05	44
769	5	0.06	50
770	5	0.06	52
774	5	0.06	60
925	5	0.07	64
927	5	0.07	68
928	5	0.07	70
933	5	0.07	80
1051	5	0.08	18

Рисунок 3.4 – Отфильтрованный набор данных

При наименьших значениях параметров min_pts = 22 и eps = 0,03 большинство точек относят к шумам, т.к. кластеры состоят из небольшого количества тесно сгруппированных точек (Рисунок 3.5).

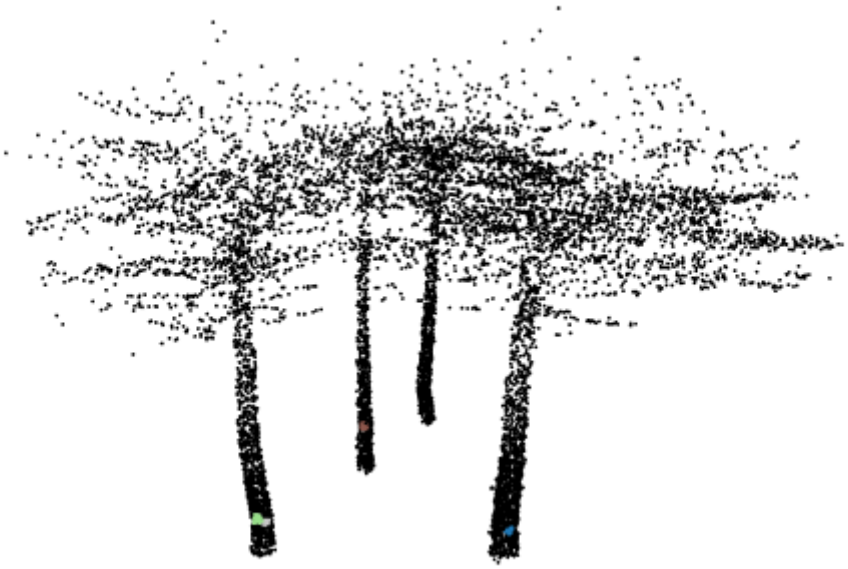

Рисунок 3.5 – Результат сегментации при минимальных eps и min_pts

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 3.6, Рисунок 3.7).

Рисунок 3.6 – Результаты сегментации при eps = 0,1, min_pts = 116

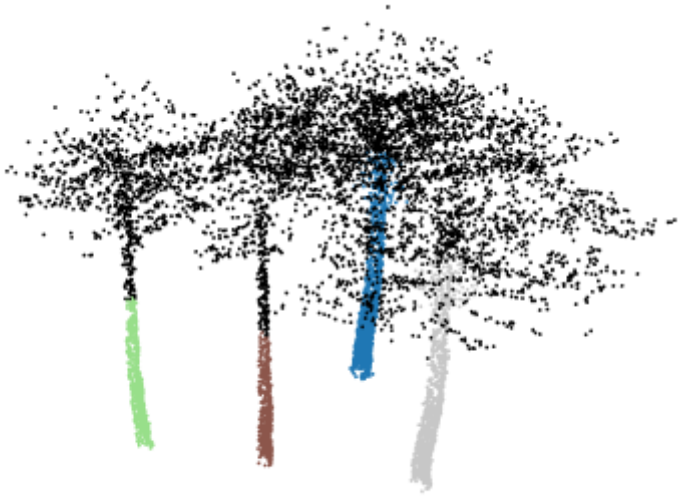

Рисунок 3.7 – Результаты сегментации при eps = 0,2, min_pts = 90

Более точно осуществляется процесс кластеризации при значениях гиперпараметров eps = 0,38 и min_pts = 100. При таких параметрах точки каждого дерева выделяются в отдельные кластеры, тем не менее остаётся некоторое значение точек в листве, которые рассматриваются как выбросы (Рисунок 3.8).

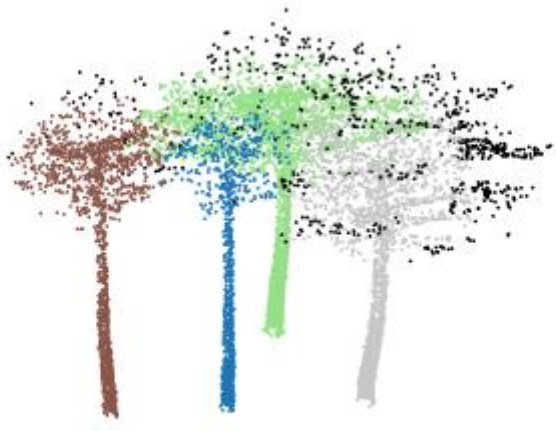

Рисунок 3.8 – Результаты сегментации при eps = 0,3, min_pts = 80

При наиболее высоких значениях гиперпараметров eps = 0,58 и min_pts = 286 почти не наблюдаются шумы, но кластеризация осуществляется неверно (Рисунок 3.9).

Рисунок 3.8 – Результаты сегментации при eps = 0,58, min_pts = 286

Для выявления наиболее оптимальных значений гиперпараметров была использована метрика Silhouette Score для пар eps, min_pts, ограниченных небольшим интервалом значений eps от 0,37 до 0,39 с шагом 0,001 и интервалом значений min_pts от 98 до 120 с шагом 2 (Рисунок 3.9, Рисунок 3.10).

Проанализировав полученный набор данных, можно сделать вывод, что наиболее оптимальные параметры eps = 0,383 и min_pts = 106 определяются максимальным значением метрики Silhouette Score, равным 0,263741. Однако визуально результаты сегментации не являются превосходными (Рисунок 3.11).

In [96]:
```
%%time
eps_values_vars = np.arange(0.37,0.39,0.001)
min_pts_vars = np.arange(98,120,2)

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_df1 = pd.DataFrame(res_data, columns=['number_of_clusters', 'eps_values', 'min_pts', 'silhouette'])
res_df1
res_df1.to_csv('output_silhouette.csv')
```

```
CPU times: total: 14min 45s
Wall time: 13min 33s
```

Рисунок 3.9 - код создания набора данных подсчета значения метрики

In [98]:
```
silhouette_df = res_df1[res_df1['number_of_clusters']==5]
pd.set_option("display.max_rows", None, "display.max_columns", None)
silhouette_df
```

140	5	0.382	114	0.261336
141	5	0.382	116	0.261559
142	5	0.382	118	0.258322
146	5	0.383	104	0.262345
147	5	0.383	106	0.263741
148	5	0.383	108	0.262443
149	5	0.383	110	0.261795
150	5	0.383	112	0.261308
151	5	0.383	114	0.261527
152	5	0.383	116	0.261028
153	5	0.383	118	0.258959

Рисунок 3.10 – отфильтрованный набор данных подсчета значения метрики

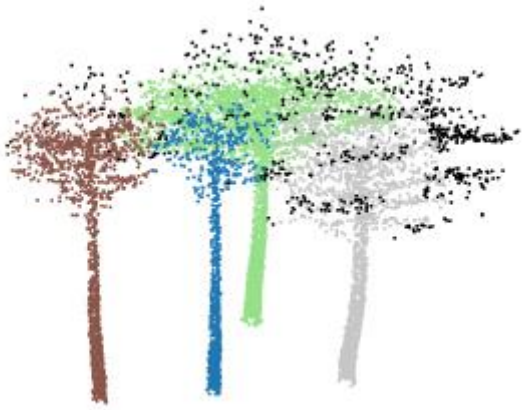

Рисунок 3.11 – Результаты сегментации при eps = 0,383, min_pts = 106

4 Выполнение индивидуального варианта

Был получен индивидуальный вариант облака точек для кластеризации участка леса.

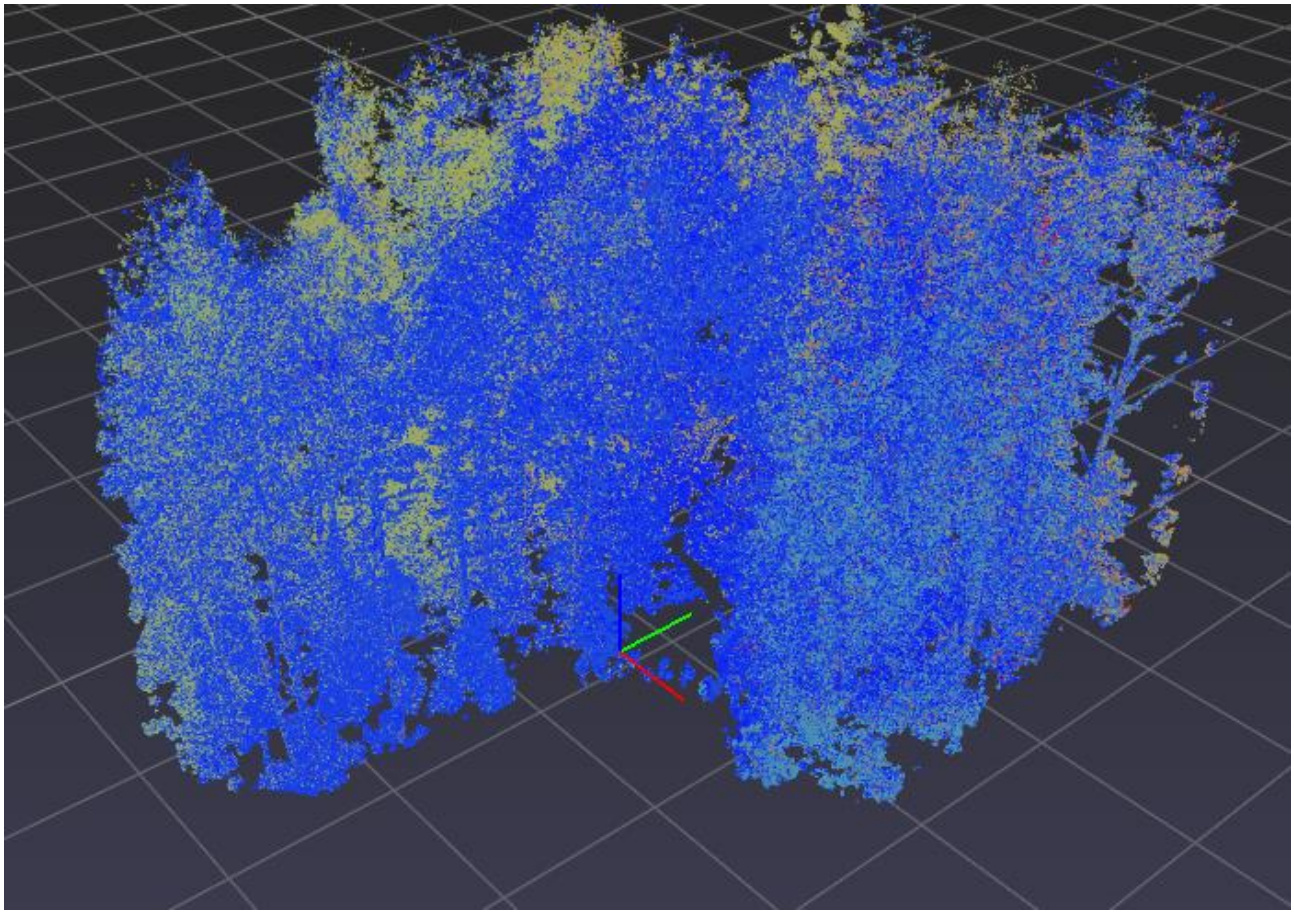

Рисунок 4.1 – Индивидуальное облако точек

В целях экономии вычислительных ресурсов устройства был применён метод градиентной очистки набора данных. Данный алгоритм сначала уменьшает число точек облака до введённого значения num_points, после чего удаляет из полученного набора точки с вероятностью, обратно пропорциональной высоте точек, чтобы уменьшить количество шумов у основания деревьев.

```python
import random

def down_pcd1(data, num_points):
    if data.shape[0] > num_points:
        factor = data.shape[0] // num_points
    else:
        factor = 1
    down_data = data[::factor]

    z_list=[d[2] for d in down_data]
    z_max=max(z_list)



    res_data=[]
    for point in down_data:
        if z_max-point[2] < random.random() * z_max:
            res_data.append(point)
    print(f'amount of points: {len(res_data)}')



    return np.asarray(res_data, dtype=np.float32)
```

Рисунок 4.2 – Алгоритм градиентной очистки

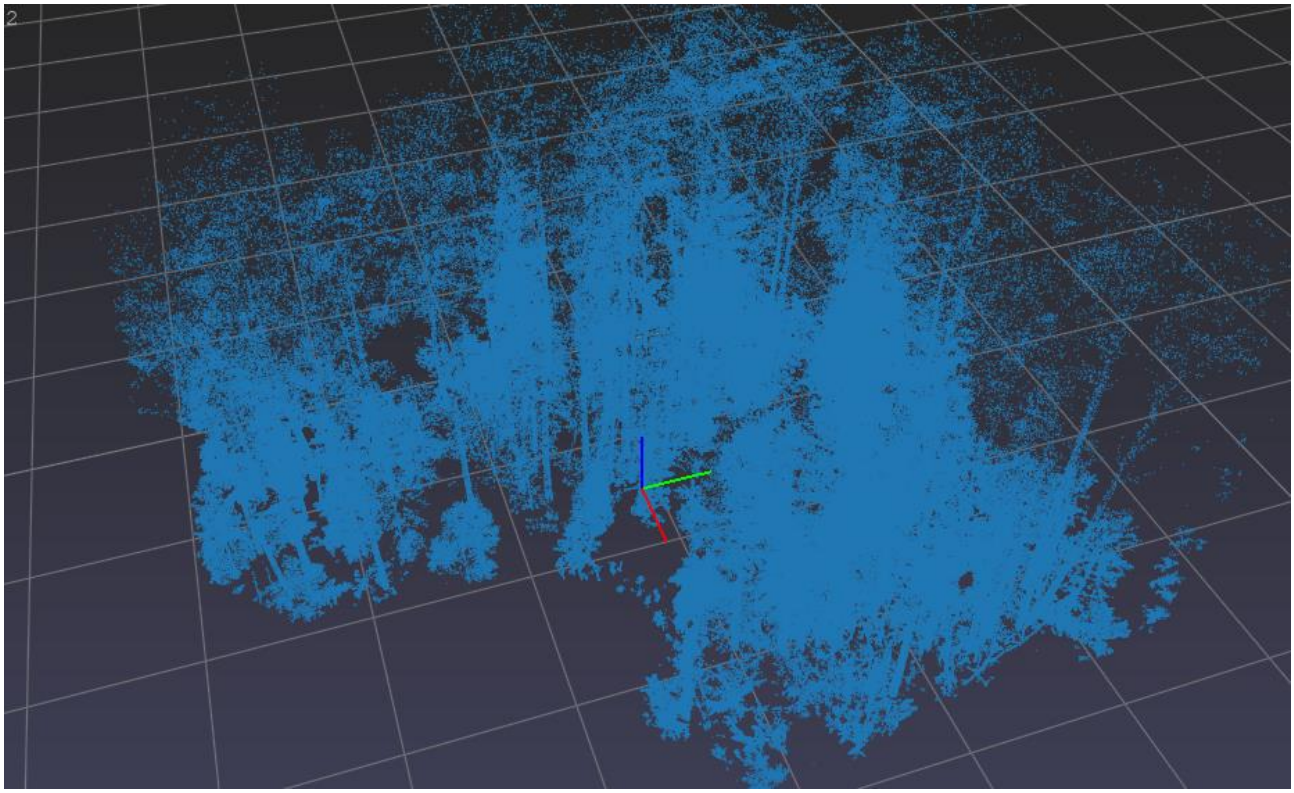

Рисунок 4.3 – Результат прореженияя точек до количества в 900 тысяч

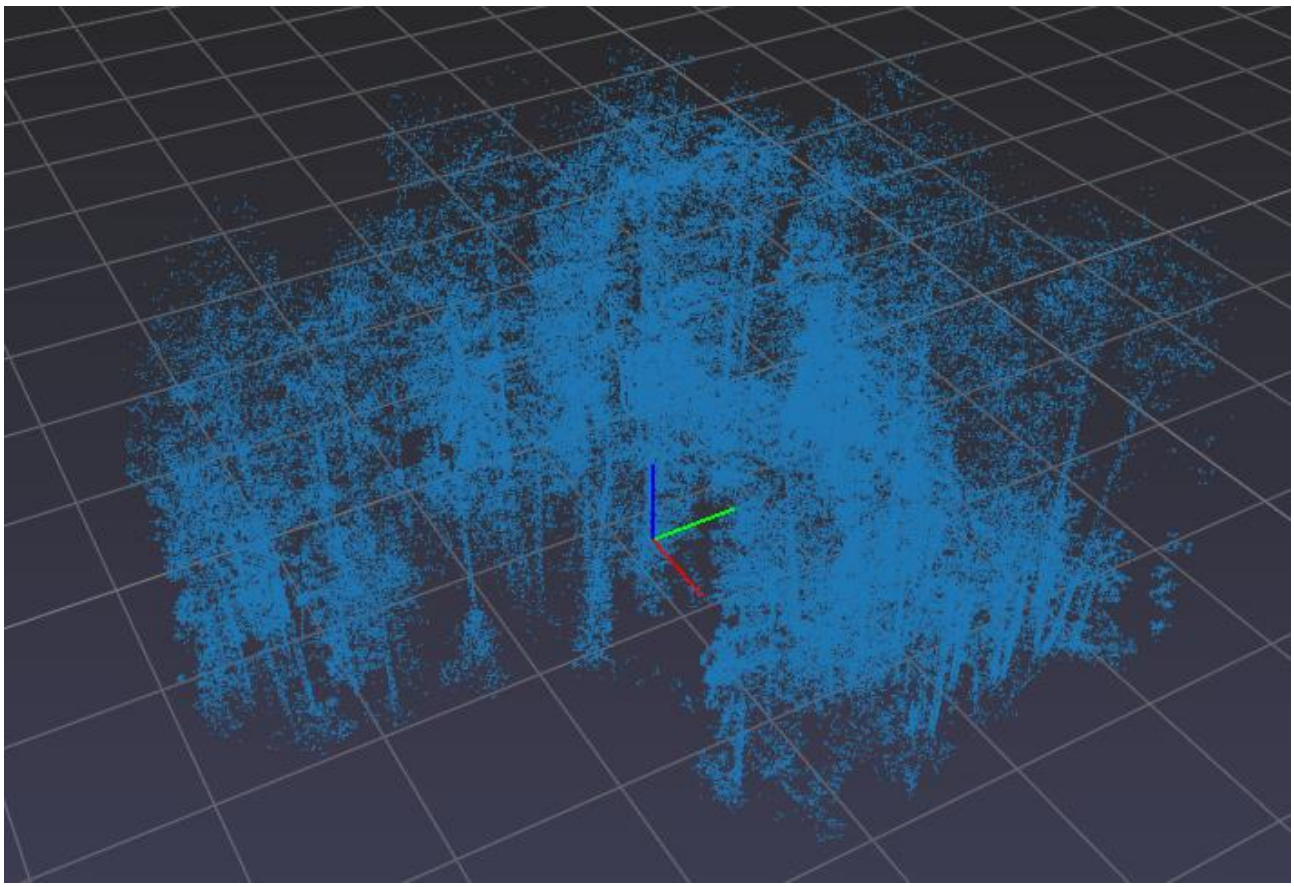

Рисунок 4.4 – Результат прорежения облака с последующим градиентным удалением точек

Таким образом, для сегментации облака размерностью в 170 тысяч точек используем метод DBSCAN с различными параметрами min_pts и eps и постараемся выделить лучшую комбинацию этих гиперпараметров.

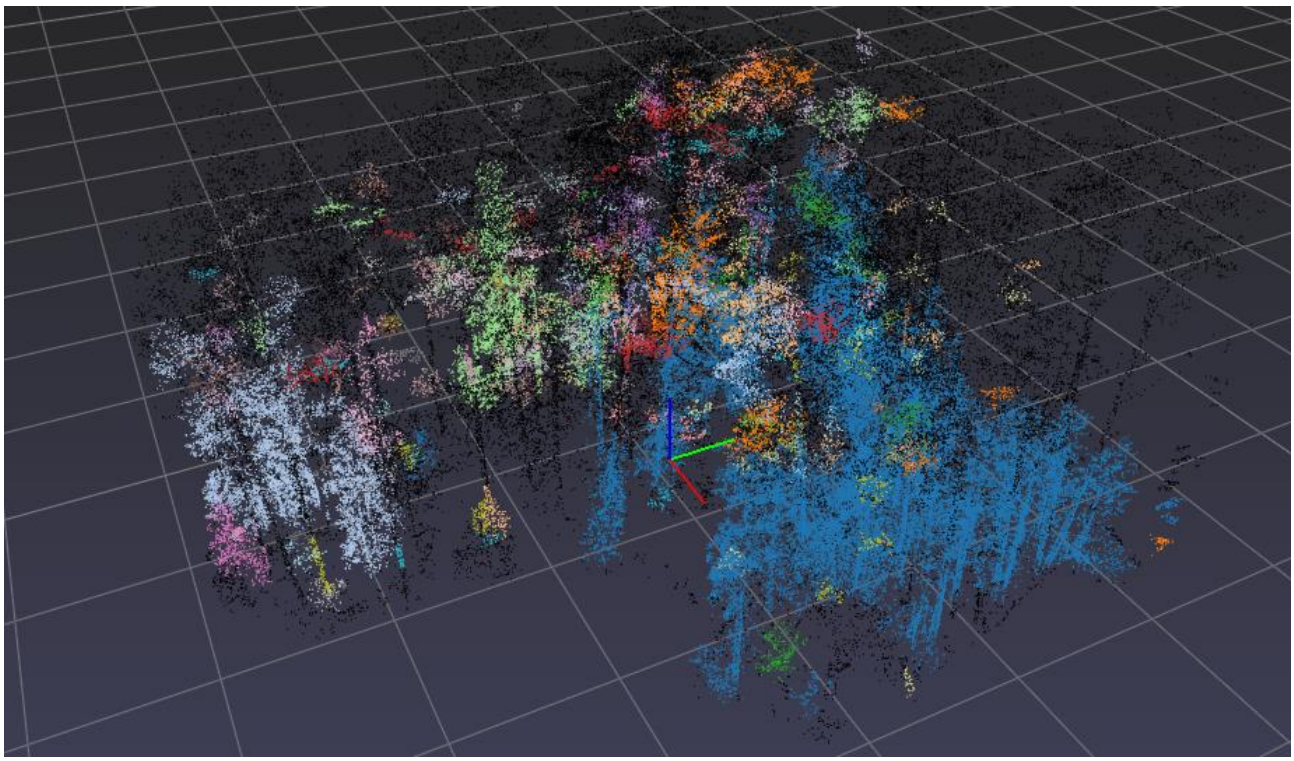

Рисунок 4.5 – Результаты сегментации при eps = 0,8, min_pts = 40

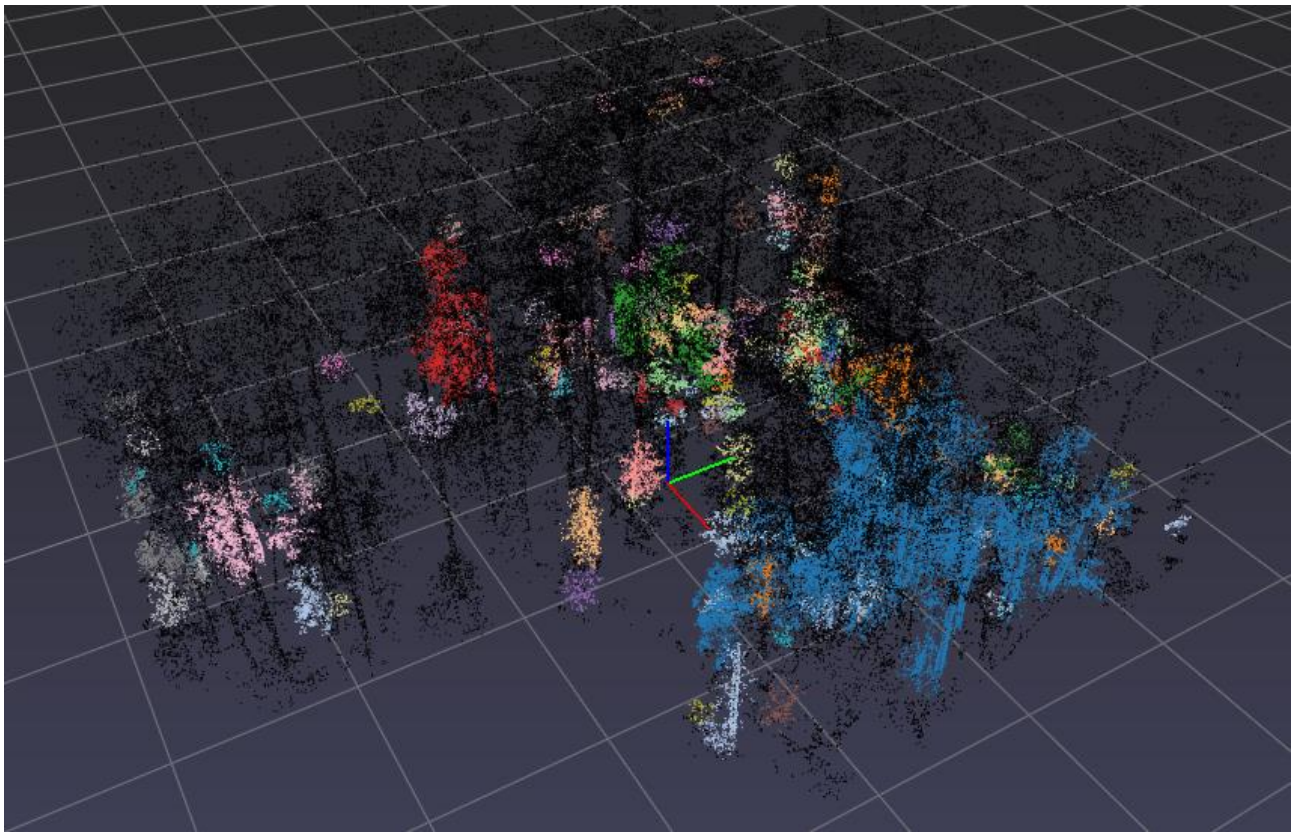

Рисунок 4.6 – Результаты сегментации при eps = 0,8, min_pts = 70

Увеличим число точек в облаке до 400 тыс.

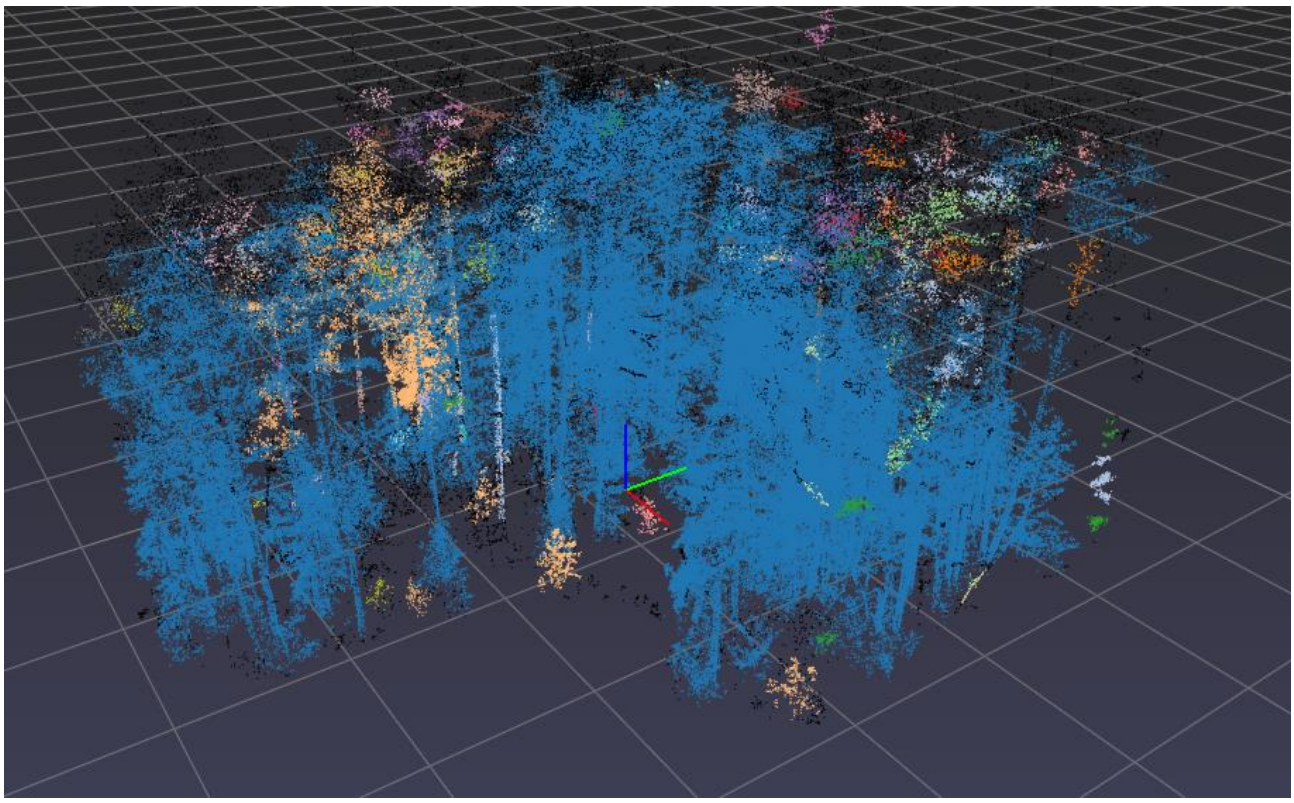

Рисунок 4.7 – Результаты сегментации при eps = 0,8, min_pts = 40, число точек ≈ 400 тыс.

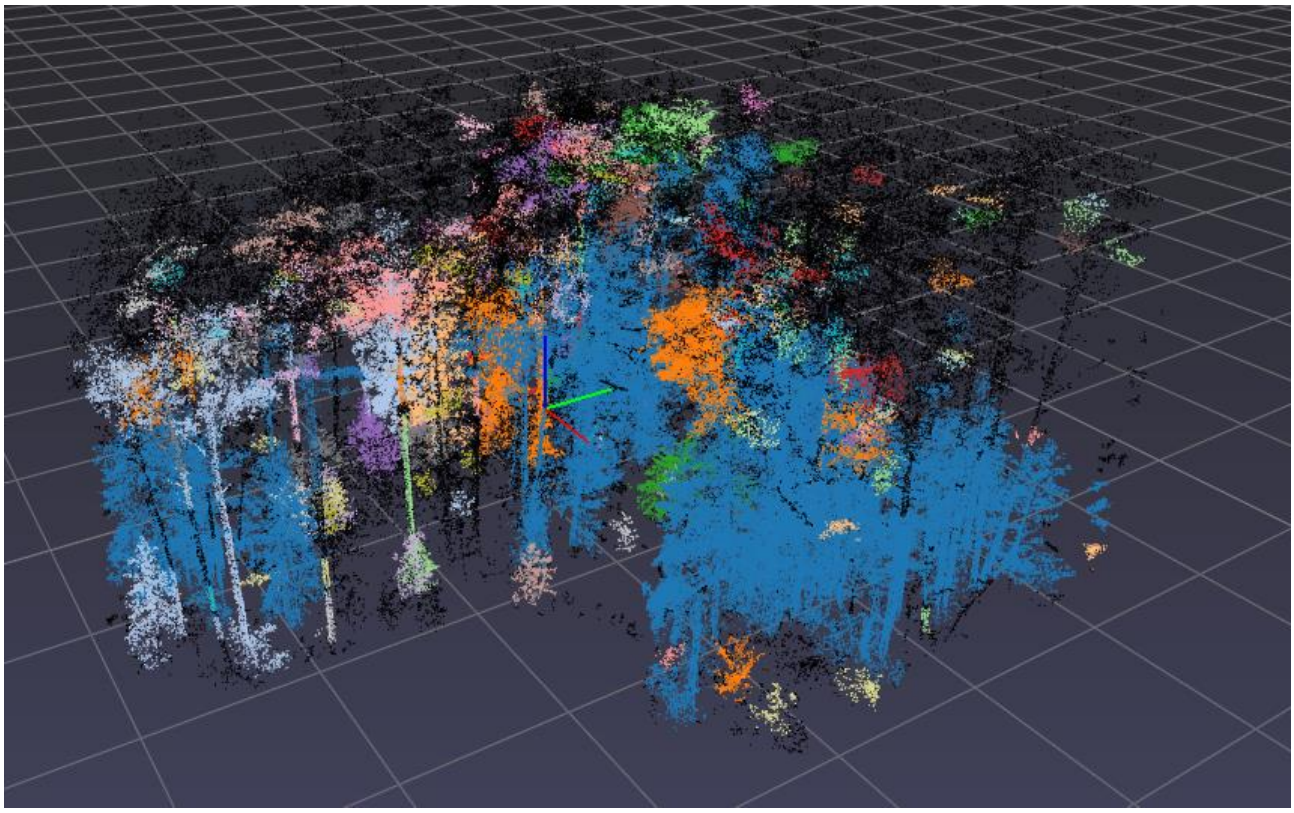

Рисунок 4.8 – Результаты сегментации при eps = 0,8, min_pts = 70, число точек ≈ 400 тыс.

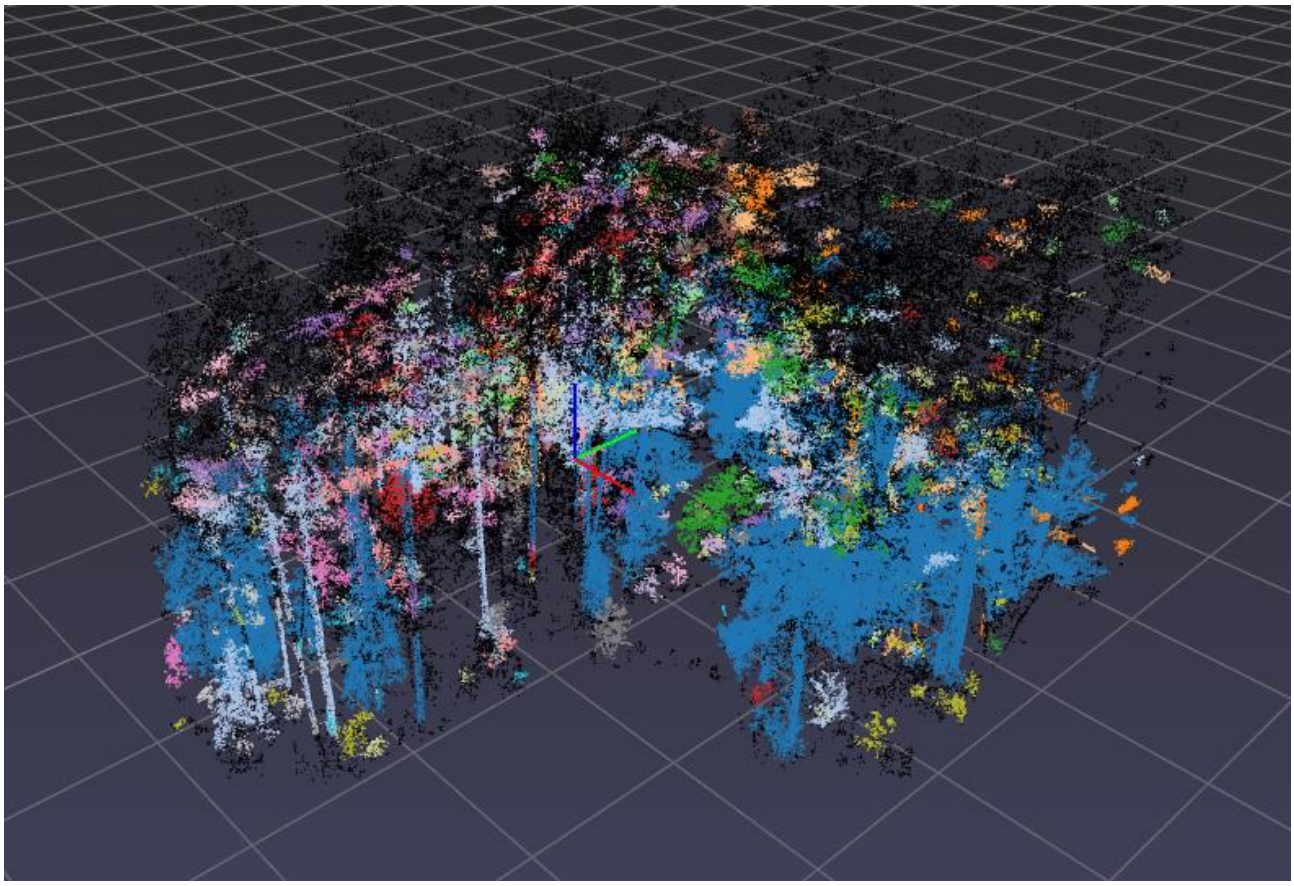

Рисунок 4.9 – Результаты сегментации при eps = 0,5, min_pts = 30, число точек ≈ 400 тыс.

Уменьшим число точек в облаке до 200 тыс.

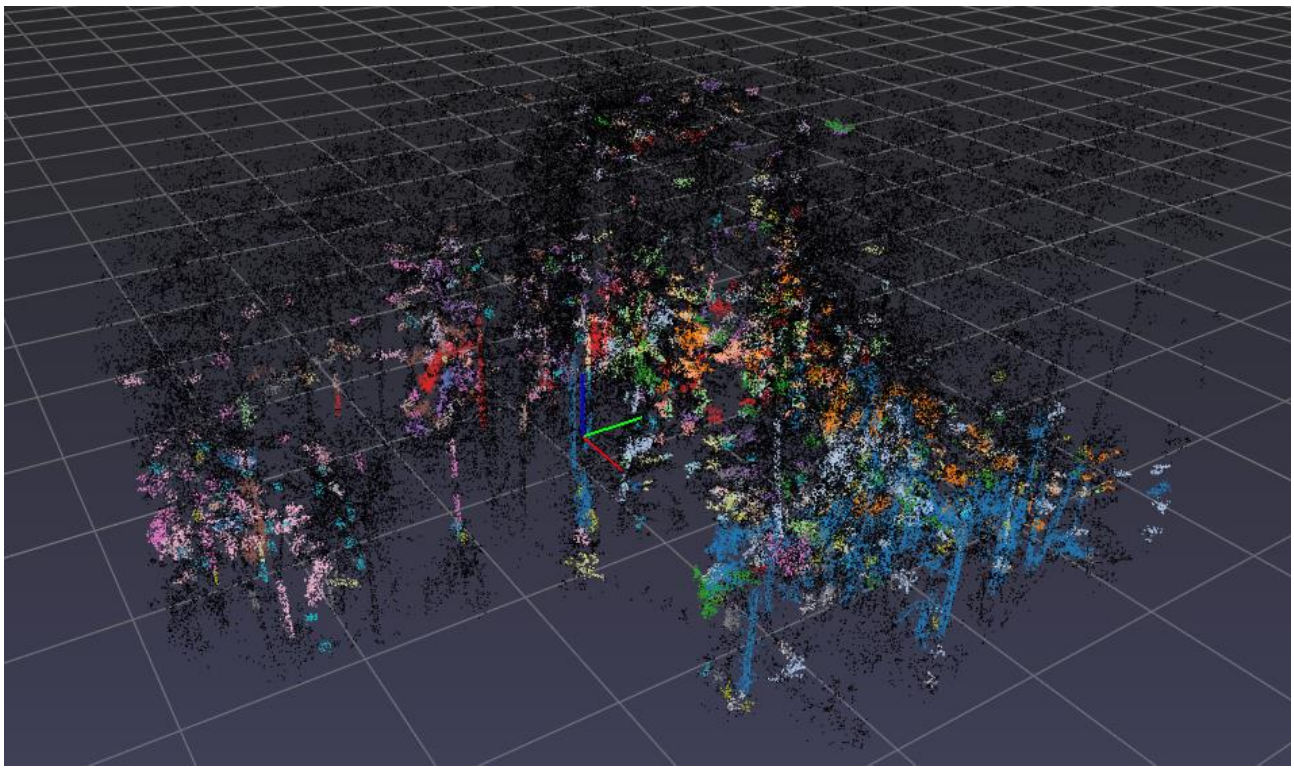

Рисунок 4.10 - Результаты сегментации при eps = 0,4, min_pts = 20, число точек ≈ 200 тыс.

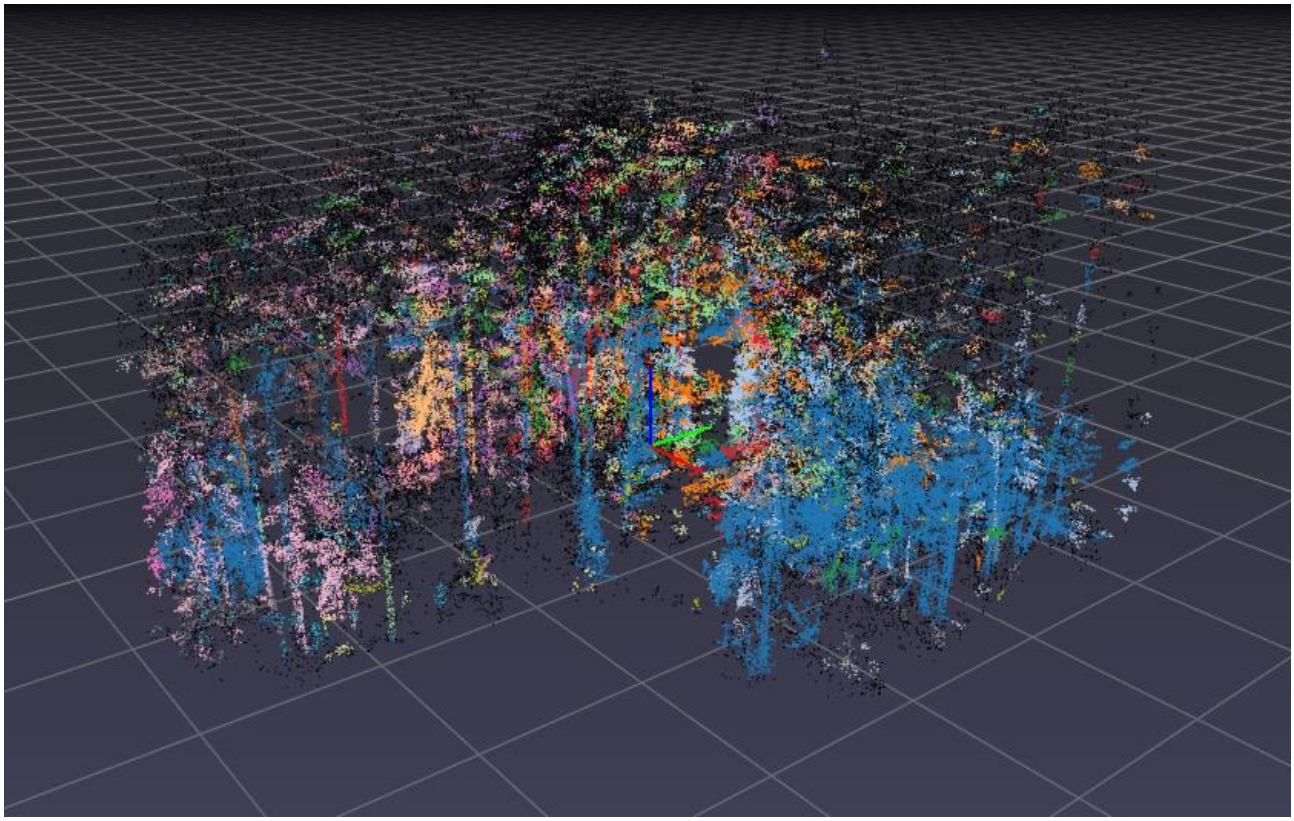

Рисунок 4.11 - Результаты сегментации при eps = 0,4, min_pts = 10, число точек ≈ 200 тыс.

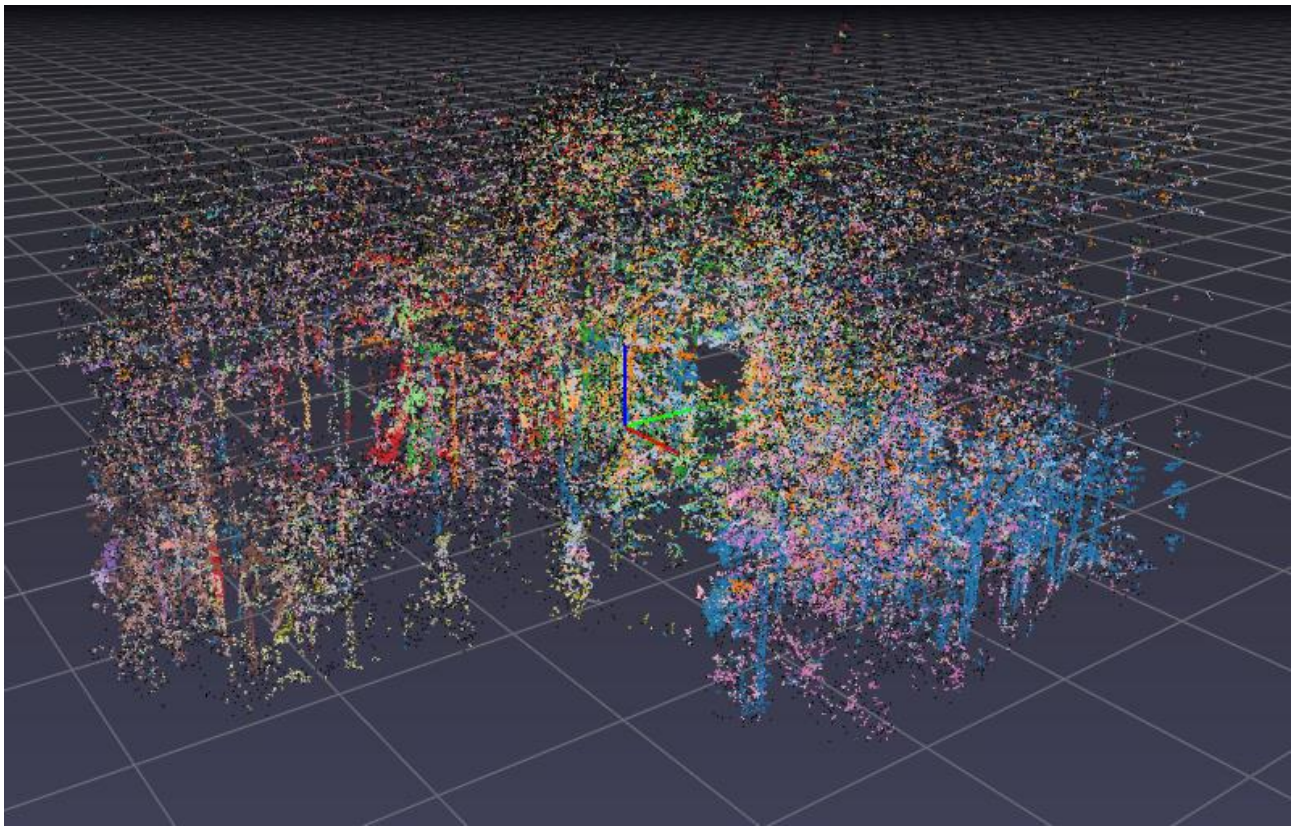

Рисунок 4.12 – Результаты сегментации при eps = 0,2, min_pts = 2, число точек ≈ 200 тыс.

Для улучшения точности подбора гиперпараметров был применён метод локтя, позволяющий определить оптимальное значение eps. Число ближайших соседей n было принято равным 10, что является оптимальным для количества точек в рассматриваемом облаке.

```
n_neighbors=10
nearest_neighbors = NearestNeighbors(n_neighbors=n_neighbors+1)
neighbors = nearest_neighbors.fit(X)
distances, indices = neighbors.kneighbors(X)
distances = np.sort(distances[:,n_neighbors], axis=0)

i = np.arange(len(distances))
knee = KneeLocator(i, distances, S=1, curve='convex', direction='increasing', interp_method='polynomial')
fig = plt.figure(figsize=(20, 10))
knee.plot_knee()
plt.xlabel("Points")
plt.ylabel("Distance")

print(distances[knee.knee])
```

```
0.6853264826933158
```

Рисунок 4.13 – Метод локтя для поиска оптимального значения eps

Таким образом, в методе локтя для каждой точки из облака вычисляется её среднее расстояние до n ближайших соседей. На графике по оси абсцисс откладывается количество рассмотренных точек, а по оси ординат – вычисленное значение среднего расстояния.

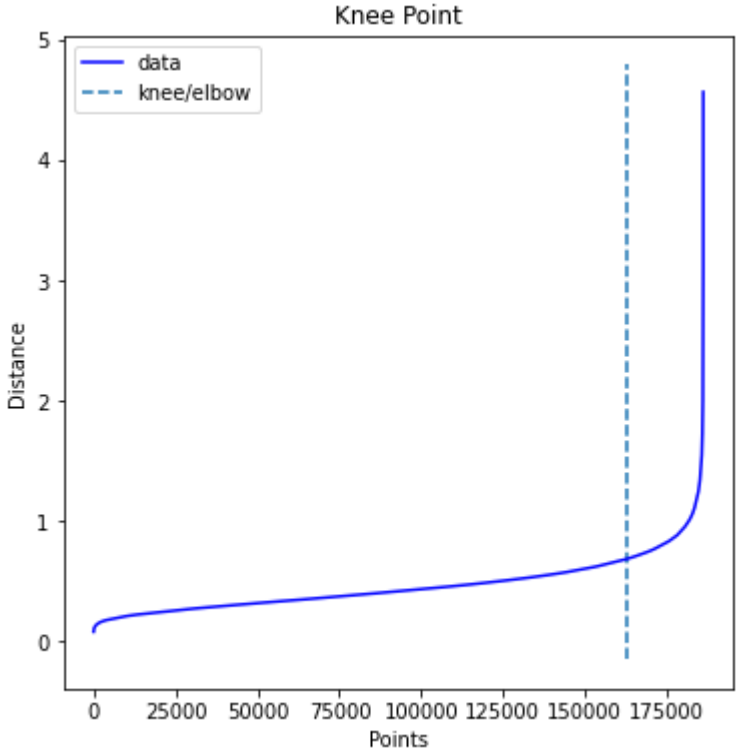

Рисунок 4.14 – График среднего расстояния между точками

Далее необходимо вычислить значение ординаты в точке сгиба графика: это значение и будет величиной, оптимальной для параметра eps. В нашем случае, eps = 0,685.

Последующий поиск оптимального гиперпараметра min_pts производится перебором значений.

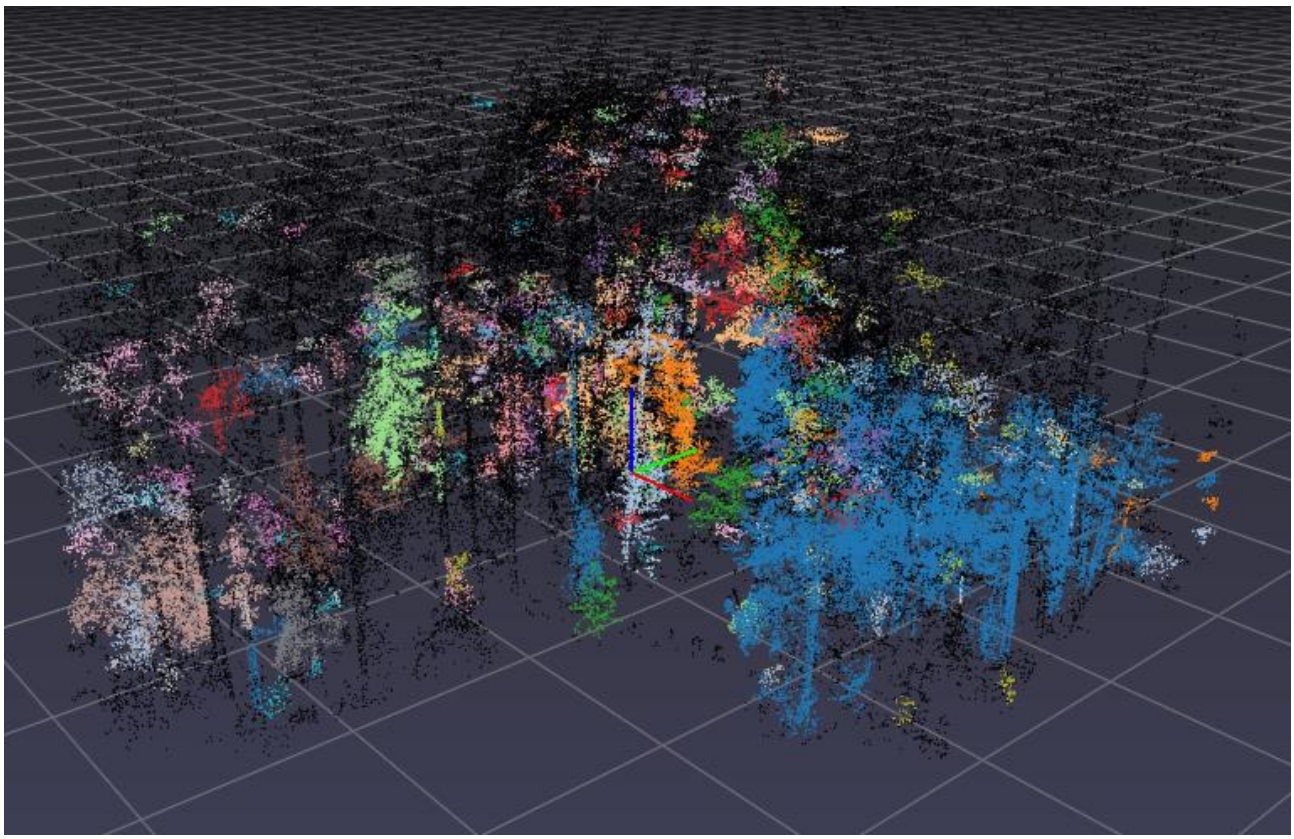

Рисунок 4.15 – Результаты сегментации при eps = 0,685, min_pts = 40, число точек ≈ 200 тыс.

На приведённой выше иллюстрации заметно, что при значении min_pts=40 происходит неудовлетворительная сегментация леса. Основания деревьев сливаются в общий кластер, в то время как вершины разбиты редкими небольшими кластерами. Для улучшения сегментации было принято решение понизить значение min_pts.

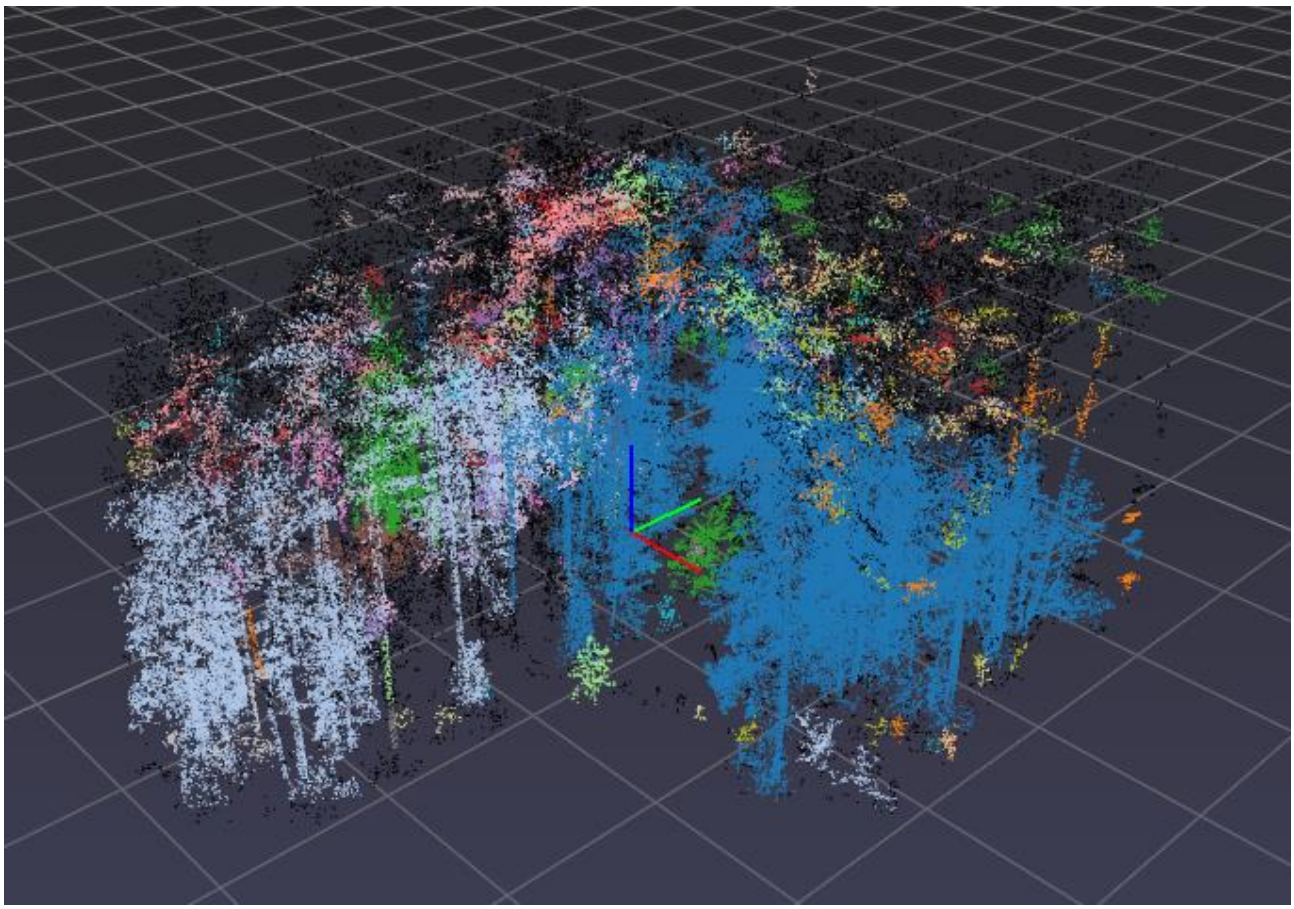

Рисунок 4.15 – Результаты сегментации при eps = 0,685, min_pts = 20, число точек ≈ 200 тыс.

На Рисунке 4.15 видно, что уже несколько деревьев, близко расположенных друг к другу, объединены в кластеры практически полностью, с основания до вершины. Тем не менее, деревья, расположенные на удалении друг от друга, сегментируются на разбитые участки, не соответствующие морфологии деревьев.

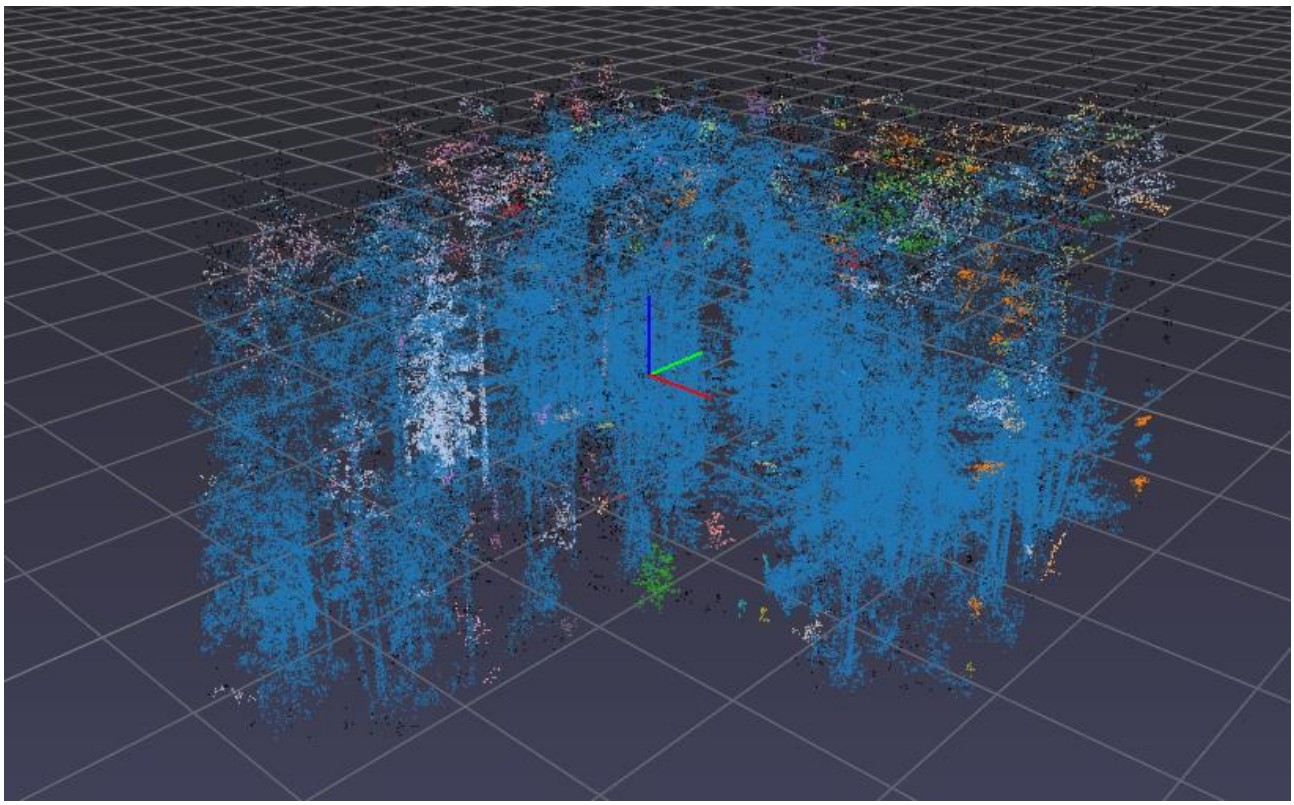

Рисунок 4.16 – Результаты сегментации при eps = 0,685, min_pts = 10, число точек ≈ 200 тыс.

При заданных величинах гиперпараметров практически все деревья сегментированы в единый кластер, а ветви некоторых деревьев разбиты на малочисленные кластеры.

Построим таблицу сегментации облака точек для фиксированного параметра eps = 0,685, определённого выше, и для диапазона min_pts с границами min_pts = 20, min_pts = 40 и шагом 1 (Рисунок 4.17). На таблице предоставлена информация о количестве полученных кластеров, о значениях eps, min_pts, а также для оценки качества каждой проведённой сегментации при различных параметрах применена метрика Silhouette Score.

	no_of_clusters	epsilon_values	minimum_points	silhouette score
0	405	0.686078	20	-0.495715
1	424	0.686078	21	-0.471497
2	443	0.686078	22	-0.443455
3	444	0.686078	23	-0.437352
4	444	0.686078	24	-0.439955
5	436	0.686078	25	-0.385975
6	441	0.686078	26	-0.387286
7	443	0.686078	27	-0.376340
8	432	0.686078	28	-0.386236
9	420	0.686078	29	-0.388329
10	425	0.686078	30	-0.398606
11	409	0.686078	31	-0.402084
12	408	0.686078	32	-0.409731
13	401	0.686078	33	-0.414497
14	402	0.686078	34	-0.411333
15	399	0.686078	35	-0.418939
16	393	0.686078	36	-0.428726

Рисунок 4.17 – Таблица сегментации с различными значениями min_pts и фиксированным eps

Проанализировав таблицу, можно сделать вывод, что метод DBSCAN не подходит для сегментации деревьев в облаке с большим количеством точек, что подтверждается отрицательным значением метрики.

Производились также попытки применить иной метод сегментации облака точек, представленный в библиотеке sklearn, такой как OPTICS, но он оказался более ресурсозатратным и менее эффективным в поиске оптимальных гиперпараметров по сравнению с используемым выше DBSCAN.

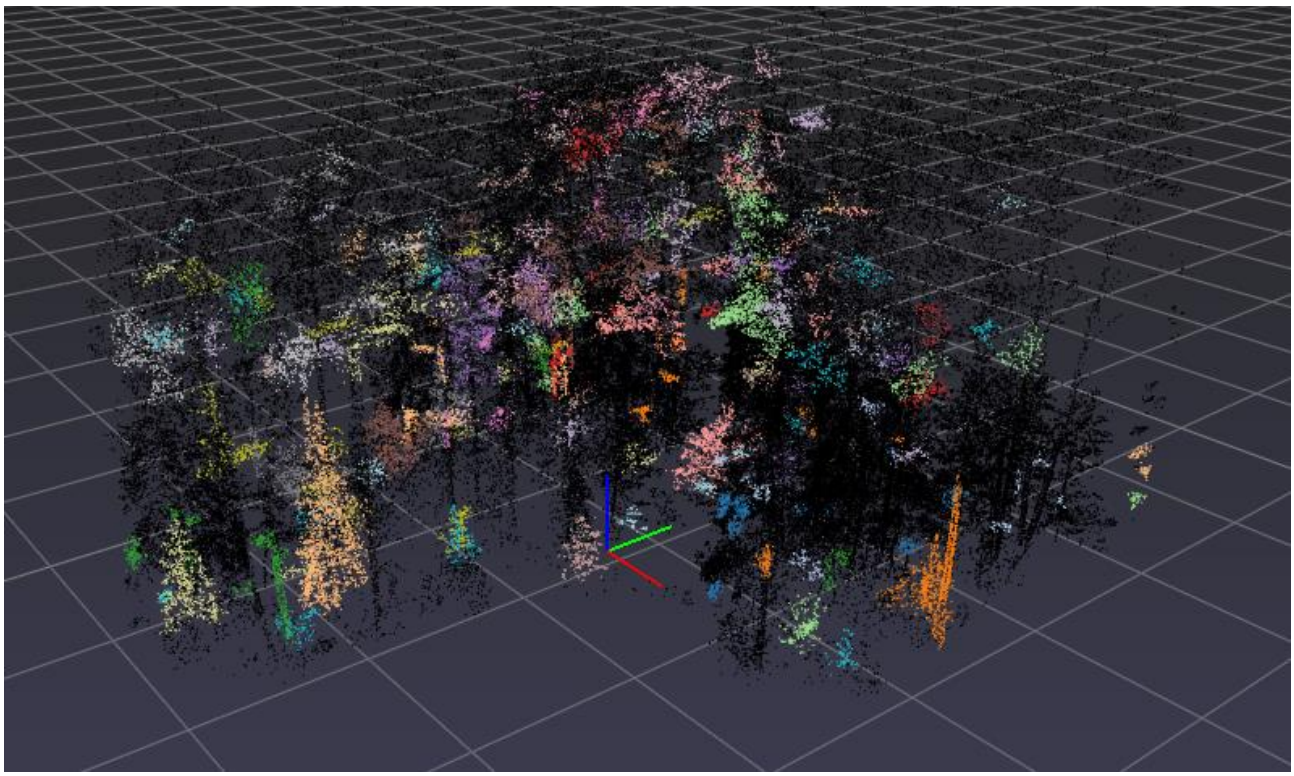

Рисунок 4.18 – Результаты сегментации алгоритмом OPTICS

Таким образом, проведя различные вариации сегментации облака точек, становится очевидна непрактичность применения методов, не специализированных для проводимой работы сегментации деревьев, ввиду большого количества данных и неоднородности расположения деревьев в пространстве.

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 четыре кластера, соответствующих каждому дереву.
2. Для оценки качества кластеризации была применена метрика Silhouette Score. Наилучшие результаты сегментации четырёх кластеров были достигнуты со значением метрики, равным 0,263741.
3. Перед кластеризацией индивидуального варианта облака точек была произведена градиентная очистка точек для снижения количества шумов у основания деревьев.
4. Проведение многочисленных попыток сегментировать полученное облако точек не привело к удовлетворительному результату, делая необходимой разработку специализированного алгоритма сегментации леса и его применение в исследовательской работе.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. 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: 30.03.2023).

2. Jie Shao, Wei Yao, Peng Wan, Lei Luo, Jiaxin Lyu, Wuming Zhang. Efficient divide-and-conquer registration of UAV and ground LiDAR point clouds through canopy shape context: [Электронный ресурс]. // arXiv.org. 2023. Дата обновления: 27.01.2022. URL: https://arxiv.org/abs/2201.11296 (Дата обращения: 30.03.2023).

3. Ekaterina Kalinicheva, Loic Landrieu, Clément Mallet, Nesrine Chehata. Multi-Layer Modeling of Dense Vegetation from Aerial LiDAR Scans: [Электронный ресурс]. // arXiv.org. 2023. Дата обновления: 25.04.2022. URL: https://arxiv.org/abs/2204.11620 (Дата обращения: 30.03.2023).

4. 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: 30.03.2023).

5. 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: 30.03.2023).