

Разработка инструмента сжатия облаков точек

Выполнил:

Трушников Андрей, гр. 7381

Руководитель:

Заславский Марк Маркович, к.т.н., доцент

Цель и задачи

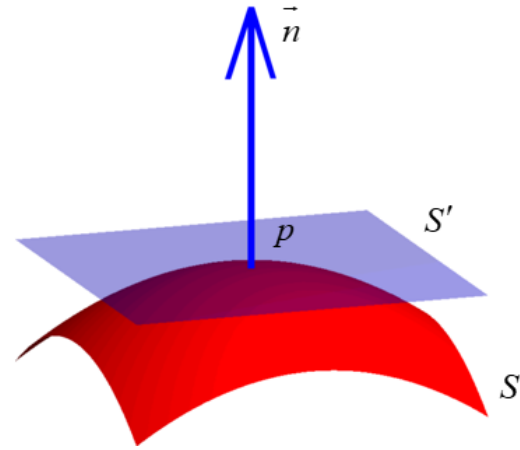
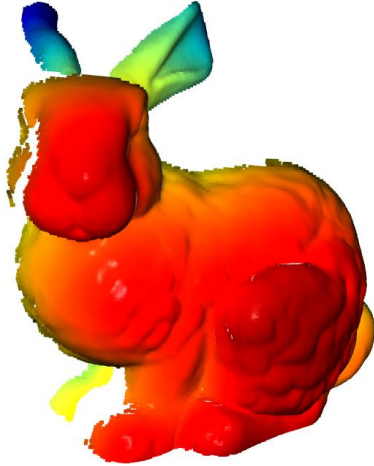
Актуальность: Облако точек может содержать тысячи или даже миллионы точек данных, из-за чего его обработка является дорогостоящей операцией как с точки зрения вычислений, так и с точки зрения хранения.

Цель: разработать прототип инструмента сжатия облаков точек с графическим и консольным интерфейсом.

Задачи:

1. Изучить предметную область и существующие методы сжатия облаков точек
2. Сделать обзор методов сжатия и выбрать два из них
3. Реализовать прототип с выбранными методами сжатия облаков
4. Провести эксперимент по сжатию облаков точек, оценить статистики по итогам экспериментов

Облако точек и его свойства.



Свойства:

- Плотность

$$Density = \frac{3 \cdot N}{4 \cdot \pi \cdot r^3} \quad [1]$$

- Вектор нормали
- Метрика RMSE (среднеквадратичная ошибка)

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N \|\hat{x}_i - x_i\|_2^2} \quad [2]$$

Обзор методов сжатия облаков точек.

| Альтернатива/Критерий | A | B | C | D |
|-----------------------|---|---|---|----|
| 1 | + | + | + | - |
| 2 | - | + | + | - |
| 3 | - | + | + | +- |
| 4 | - | + | - | - |

1. Point cloud simplification with preserved edge based on normal vector
2. A Linear Programming Approach for 3D Point Cloud Simplification
3. Feature Preserving and Uniformity-controllable Point Cloud Simplification on Graph
4. 3D Point Cloud Simplification Based on k-Nearest Neighbor and Clustering

Реализация первого метода сжатия облака точек

- Поиск и сохранение граничных точек в облаке.

$$\|C_i - p_i\| > \lambda \cdot Z_i(V_i), \quad [3]$$

где C_i - центроид множества ближайших соседей точки p_i , λ - фиксированный параметр, $Z_i(V_i)$ - евклидово расстояние до ближайшей точки p_i .

- Деление облака точек на кубы.
- Удаление наименее важных неграничных точек.

$$i(p) = \frac{\sum_{i=1}^k |(p - q_i) n_i|}{k}, \quad [4]$$

где p - неграничная точка, q_i - точка, принадлежащая множеству ближайших соседей точки p , n_i - вектор нормали в точке q_i .

Реализация третьего метода сжатия облака точек

- Деление облака точек на кубы.
- Потери в геометрических особенностях.

Построение матрицы смежности:

$$W_{i,j} = \begin{cases} \exp\left(-\frac{\|x_i - x_j\|_2^2}{\sigma^2}\right) & , j \in N_i, \\ 0, \text{ в противном случае} \end{cases} \quad [5]$$

где σ - это параметр, N_i - множество ближайших соседей вершины i

Нормализация матрицы смежности:

$$\tilde{W} = I - D^{-1}W, \quad [6]$$

где I - это единичная матрица, D - матрица степеней

$$D = \sum_j W_{i,j} \quad [7]$$

Реализация третьего метода сжатия облака точек

Матрица Кирхгофа LX :

$$(LX)(i) = \tilde{X}_i = x_i - \sum \tilde{W}_{i,j} x_j$$

Потери в геометрических особенностях:

$$l_f(\psi) = \|\psi LX - LX\|_2^2, \quad [8]$$

- Потери равномерности плотности облака точек.

$$l_e(\psi) = \|A\psi l - \alpha k l\|_2^2,$$

где l - это вектор столбец с единичными элементами,
 k - количество ближайших соседей, α - скорость упрощения облака

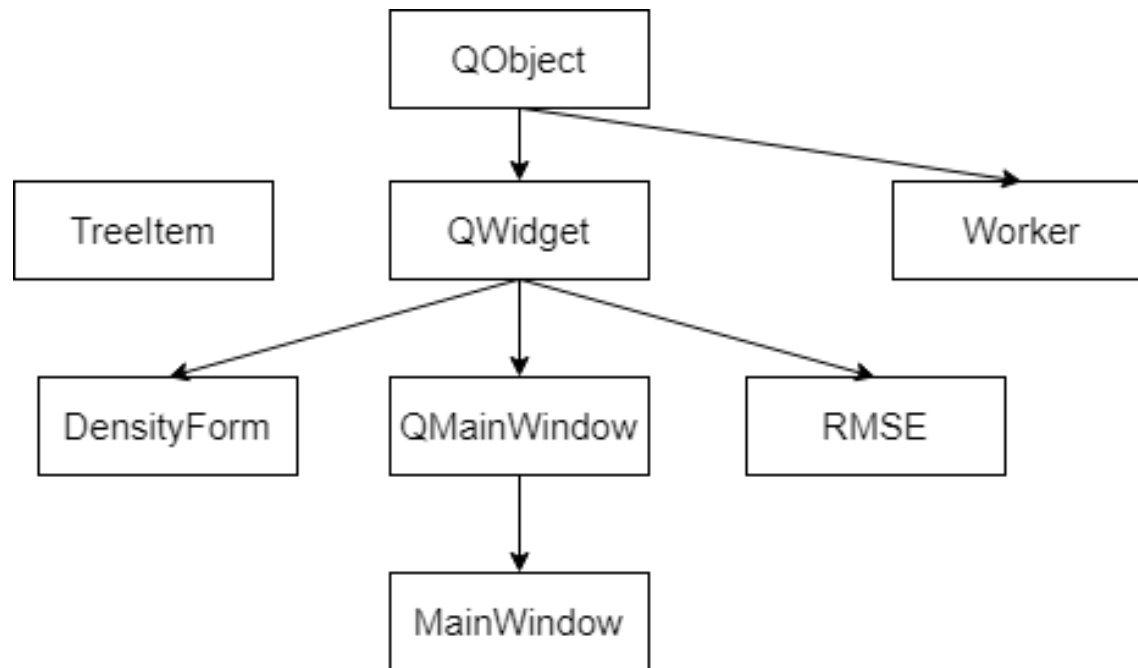
$$\min_{\psi} \|\psi LX - LX\|_2^2 + \lambda \|A\psi l - \alpha k l\|_2^2,$$

$$\psi_{i,j} \in \{0,1\}, i=1,2,\dots,N$$

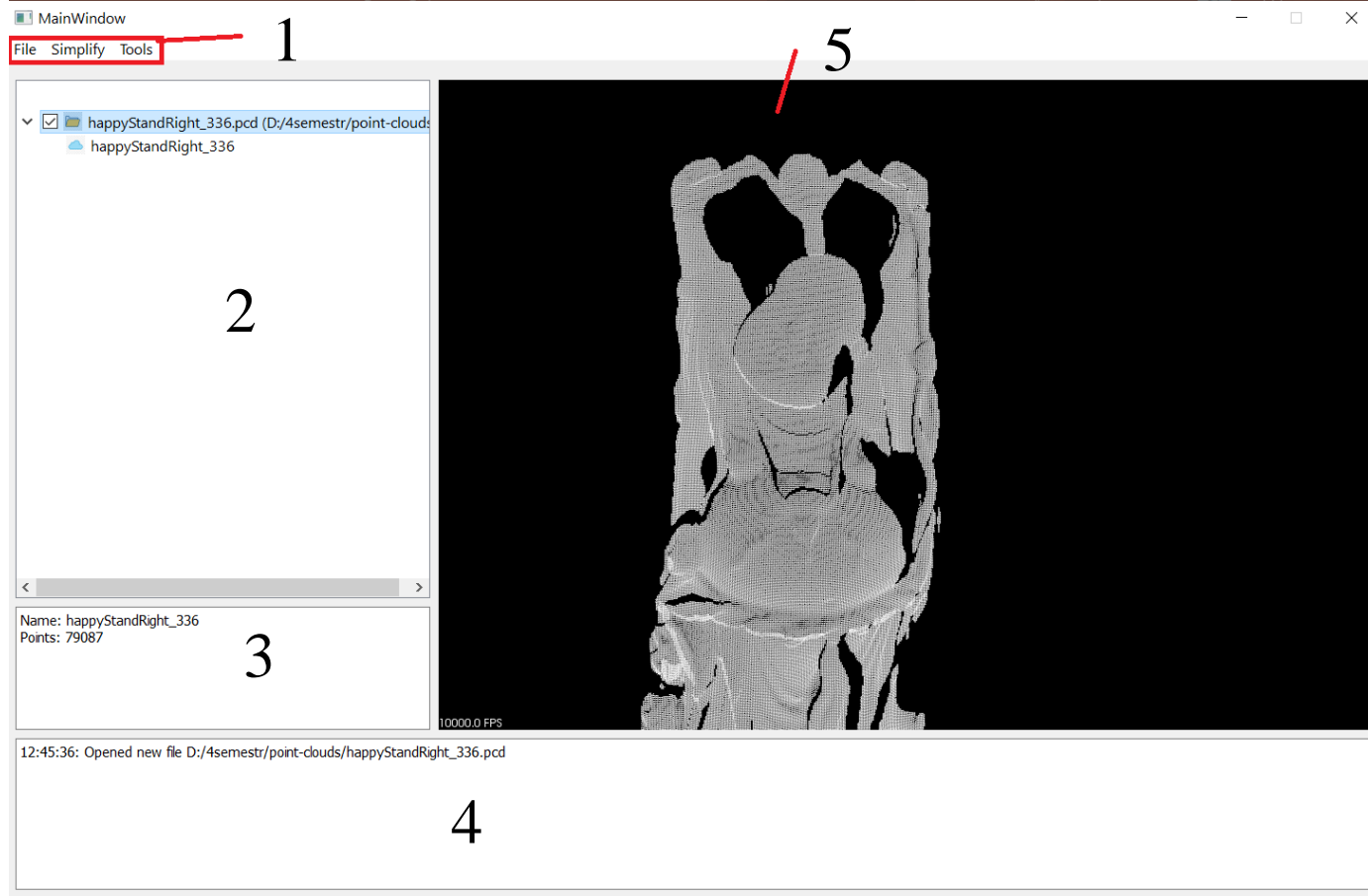
$$\psi_{i,j} = 0, i \neq j$$

$$\text{tr}(\psi) = \alpha N$$

Реализация графического и консольного интерфейса



Демонстрация графического и консольного интерфейса

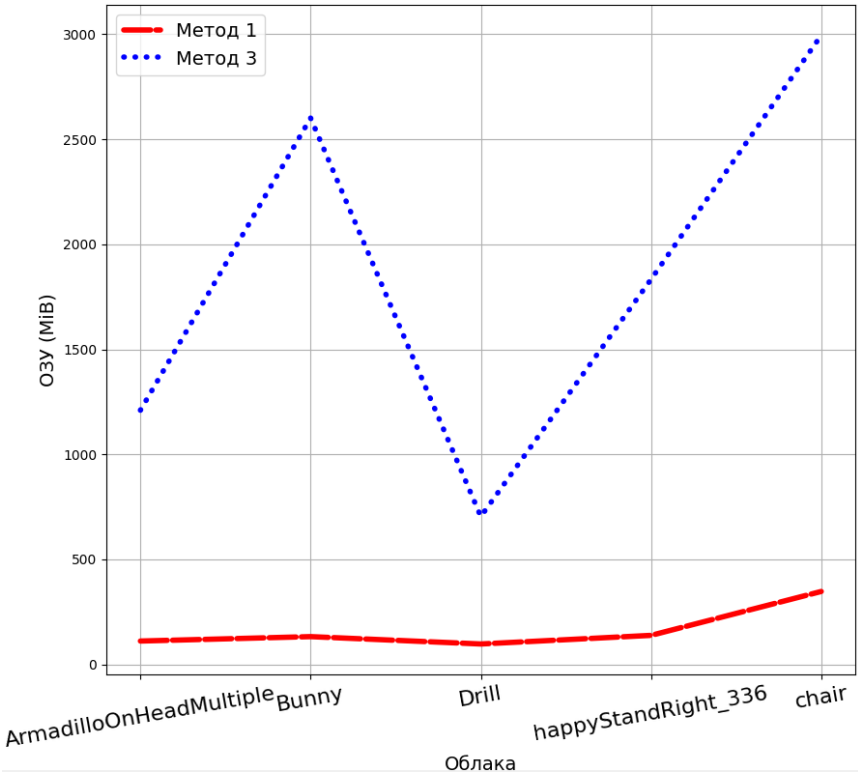
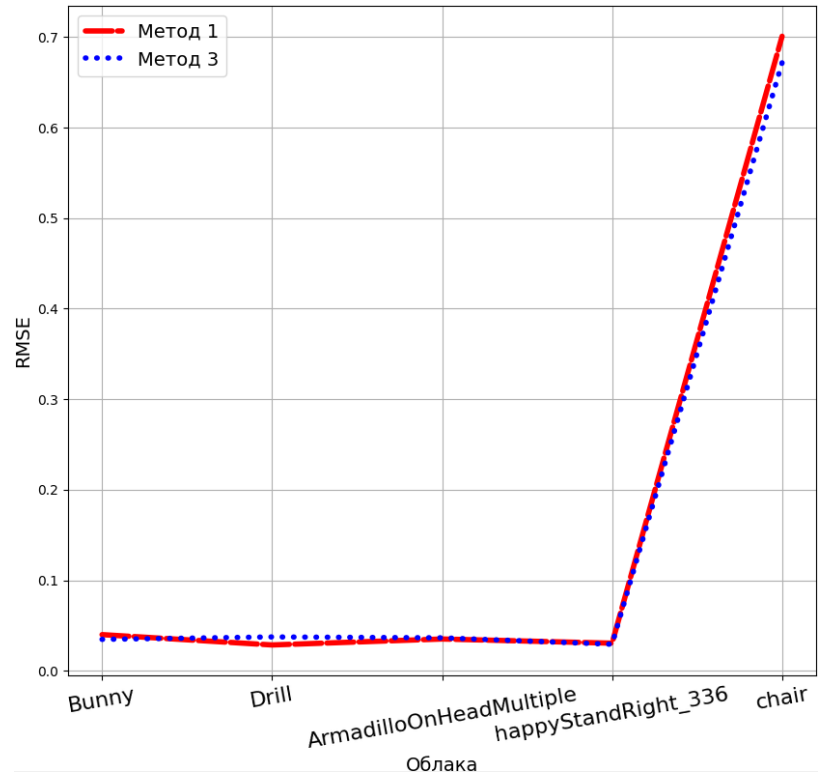


```
PS D:\4semestr\diploma\pythonProject\console> python main.py drill_1.pcd 1

Сообщение: Сжатие первым методом
[KSearching... |#####| 100%
[KSimplifying... |#####| 2/2
Результаты.
Исходное кол-во точек: 4154
После сжатия: 911
RMSE: 0.028595611145592757
Затраченное время: 8.06 секунд
Облако точек сохранено по пути: D:/4semestr/diploma/pythonProject/console
/drill_1_simplified.pcd
```

Результаты экспериментов по сжатию облаков точек

| Облако точек | Размер до сжатия (КБ) | Размер после сжатия (КБ) | Продолжительность (сек) | Метод |
|---------------------------|-----------------------|--------------------------|-------------------------|-------|
| Bunny | 629 | 108 | 141.198 | 1 |
| | | 134 | 244.251 | 3 |
| Drill | 65 | 31,9 | 7.412 | 1 |
| | | 14,6 | 13.8911 | 3 |
| Armadillo On HeadMultiple | 513 | 201 | 122.092 | 1 |
| | | 116 | 167.727 | 3 |
| happyStandRight_336 | 1265,572 | 283 | 335.568 | 1 |
| | | 265 | 536.929 | 3 |
| chair | 3897,854 | 650 | 462.75 | 1 |
| | | 378 | 2943.78 | 3 |



Заключение

- В результате обзора методов сжатия облаков было решено реализовать 1 и 3
- Методы сжатия реализованы на языке программирования Python.
- Графический интерфейс прототипа сделан с помощью фреймворка Qt, консольный интерфейс на языке программирования Python
- В ходе экспериментов по сжатию облаков выяснилось, что метод 1 потребляет меньше ОЗУ, значения ошибки RMSE для обоих методов были примерно одинаковыми

В дальнейшем, чтобы ускорить скорость сжатия методом 1 необходимо реализовать его работу на видеокарте. Также для улучшения качества сжатия обоих методов необходимо использовать алгоритм разбиения на кубы облака точек, которой возвращал бы кубы, содержащие количество точек из заданного диапазона.

Апробация работы

- Репозиторий проекта
https://github.com/moevm/bsc_trushnikov.

DEMO