

Задание 1b: Subsampling облака точек с использованием Python и NumPy

Цель:

Научиться выполнять понижение плотности облака точек (subsampling) с помощью Python и библиотеки NumPy, используя разные методы выборки.

Исходные данные:

Для работы используйте облако точек в текстовом формате `.txt` или `.xyz` — три колонки: `x y z`. Пример строки:

```
12.345 67.890 3.210
```

Можно использовать собственный файл или сгенерировать облако с помощью NumPy.

Задание:

1. Загрузка облака точек

Если нет готового файла, сгенерируйте облако точек:

```
import numpy as np

# Случайное облако 100 000 точек в кубе 100x100x100
points = np.random.uniform(0, 100, size=(100000, 3))
np.savetxt('point_cloud.xyz', points)
```

Загрузите облако:

```
points = np.loadtxt('point_cloud.xyz')
```

2. Реализуйте 3 метода subsampling:

A. Random subsampling

Выберите случайные `N` точек (например, 10 000):

```
def random_subsampling(points, n_samples):
    indices = np.random.choice(points.shape[0], n_samples, replace=False)
    return points[indices]
```

B. Uniform grid subsampling (voxel grid)

Разделите пространство на воксели (например, $1 \times 1 \times 1$ м) и оставь по одной точке на воксель. Это базовая реализация без внешних библиотек.

Подсказка: округлите координаты, чтобы сгруппировать точки по "ячейкам":

```
def voxel_grid_subsampling(points, voxel_size):
    coords = (points / voxel_size).astype(int)
    _, unique_indices = np.unique(coords, axis=0, return_index=True)
    return points[unique_indices]
```

C. Farthest Point Sampling (опционально, усложнённый уровень)

Выбери первые k точек так, чтобы каждая следующая была максимально далеко от уже выбранных.

(Можно реализовать позже, если интересен более продвинутый уровень.)

3. Сохраните результаты

Сохраните каждое из субсэмплированных облаков в отдельный файл `.xyz`:

```
np.savetxt('subsampled_random.xyz', subsampled_random)
```

4. Визуализация (по желанию)

Можно использовать `matplotlib` для быстрой 3D-визуализации:

Вопросы (в отчёте):

1. Как влияет размер выборки на визуальное качество облака?
2. Как влияет размер вокселя в `voxel_grid_subsampling` на детализацию?
3. Какой метод эффективнее по времени? (Добавьте `time.time()` замеры)
4. Какой метод лучше сохраняет геометрию объекта?

Что сдавать:

- Python-скрипт (`.py` или Jupyter `.ipynb`)
- Три файла `.xyz` с результатами
- Скриншоты или графики (если делалась визуализация)
- Краткий текстовый отчёт с ответами на вопросы