

МФТИ

**Отчет по практическим заданиям  
1b и 2b**

**Филак Александр**

Долгопрудный  
2025

# 1 ОТЧЁТ ПО ПРАКТИЧЕСКИМ ЗАДАНИЯМ 2 И 4

## 1.1. Введение

В рамках двух практических заданий были изучены и реализованы методы работы с облаками точек, включая визуализацию с использованием цветовой информации и обработку скалярных полей. Задание 2 было посвящено различным способам назначения цветов точкам и их визуализации в 2D и 3D. Задание 4 охватывало работу со скалярными полями: их создание, преобразование, фильтрацию и визуализацию. Оба задания демонстрируют важные аспекты предобработки и анализа трёхмерных данных.

## 1.2. Теоретическая часть

### 1.2.1. Облако точек

Облако точек — это набор точек в трёхмерном пространстве, каждая из которых имеет координаты  $(X, Y, Z)$ . В дополнение к координатам точки могут содержать дополнительную информацию: цвет (RGB), интенсивность, нормали или произвольные скалярные поля. Облака точек широко используются в компьютерном зрении, робототехнике, картографии и 3D-моделировании.

### 1.2.2. Цветовое представление RGB

Цвет в модели RGB представляется тремя компонентами: красным (R), зелёным (G) и синим (B). Каждый компонент может принимать значения от 0 до 1 (нормализованная форма) или от 0 до 255 (целочисленная форма). Комбинация этих компонентов создаёт итоговый цвет точки.

### 1.2.3. Скалярные поля

Скалярное поле — это функция, сопоставляющая каждой точке пространства некоторое скалярное значение. В контексте облаков точек скалярное поле обычно представлено как дополнительный столбец данных, связанный с каждой точкой. Примеры скалярных полей: высота, температура, интенсивность отражения лидара, расстояние до центра и т.д.

## 1.3. Задание 2: Работа с цветами в облаках точек

### 1.3.1. Постановка задачи

Цель задания — освоить методы визуализации облаков точек с цветовой информацией, научиться назначать цвета точкам на основе их координат и других параметров, сравнить возможности статичной (Matplotlib) и интерактивной (Plotly) визуализации.

### 1.3.2. Ход выполнения

#### 1. 1. Генерация облака точек

Было создано равномерно распределённое облако из 1000 точек в кубе  $[0, 1]^3$ .

#### 2. 2. Назначение цветов по координатам

Цвет каждой точке был назначен на основе её координат:

- Красный компонент = координата X
- Зелёный компонент = координата Y
- Синий компонент = координата Z

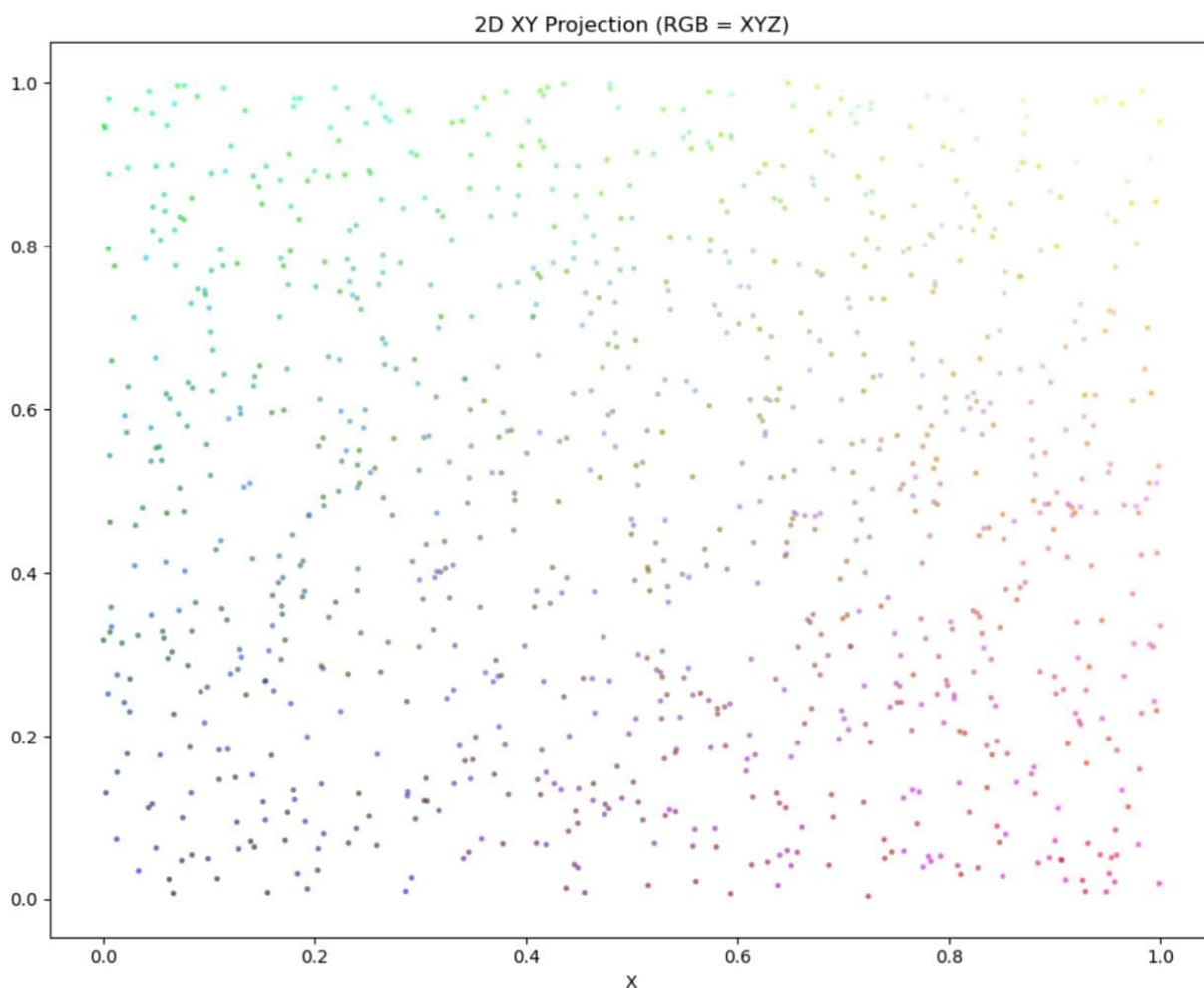


Рис. 1: 2D-визуализация в Matplotlib (RGB = XYZ)

#### 3. 3. 3D-визуализация в Plotly

Создана интерактивная 3D-визуализация, позволяющая вращать сцену, приближать и удалять объект.

#### 4. 4. Альтернативная схема раскраски

3D: Color by Distance to Center

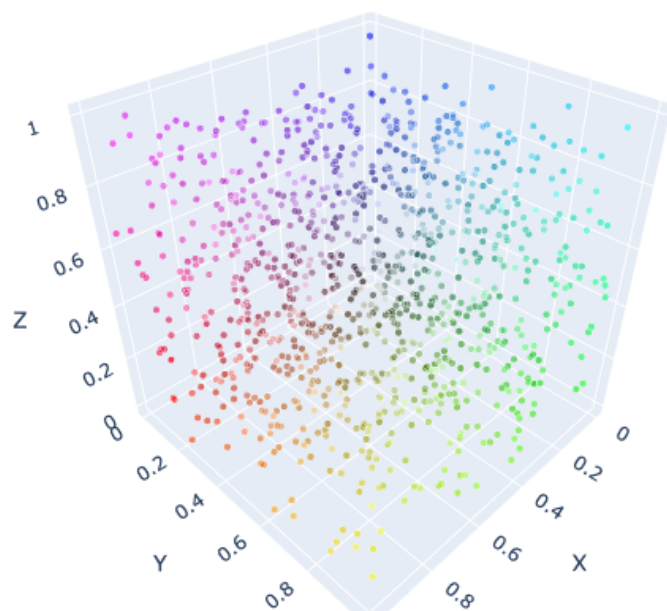


Рис. 2: 3D-визуализация в Plotly

Реализована схема раскраски на основе расстояния от точки до центра куба  $(0.5, 0.5, 0.5)$ . Точки, близкие к центру, окрашены в холодные тона (синий), а удалённые — в тёплые (красный/жёлтый).

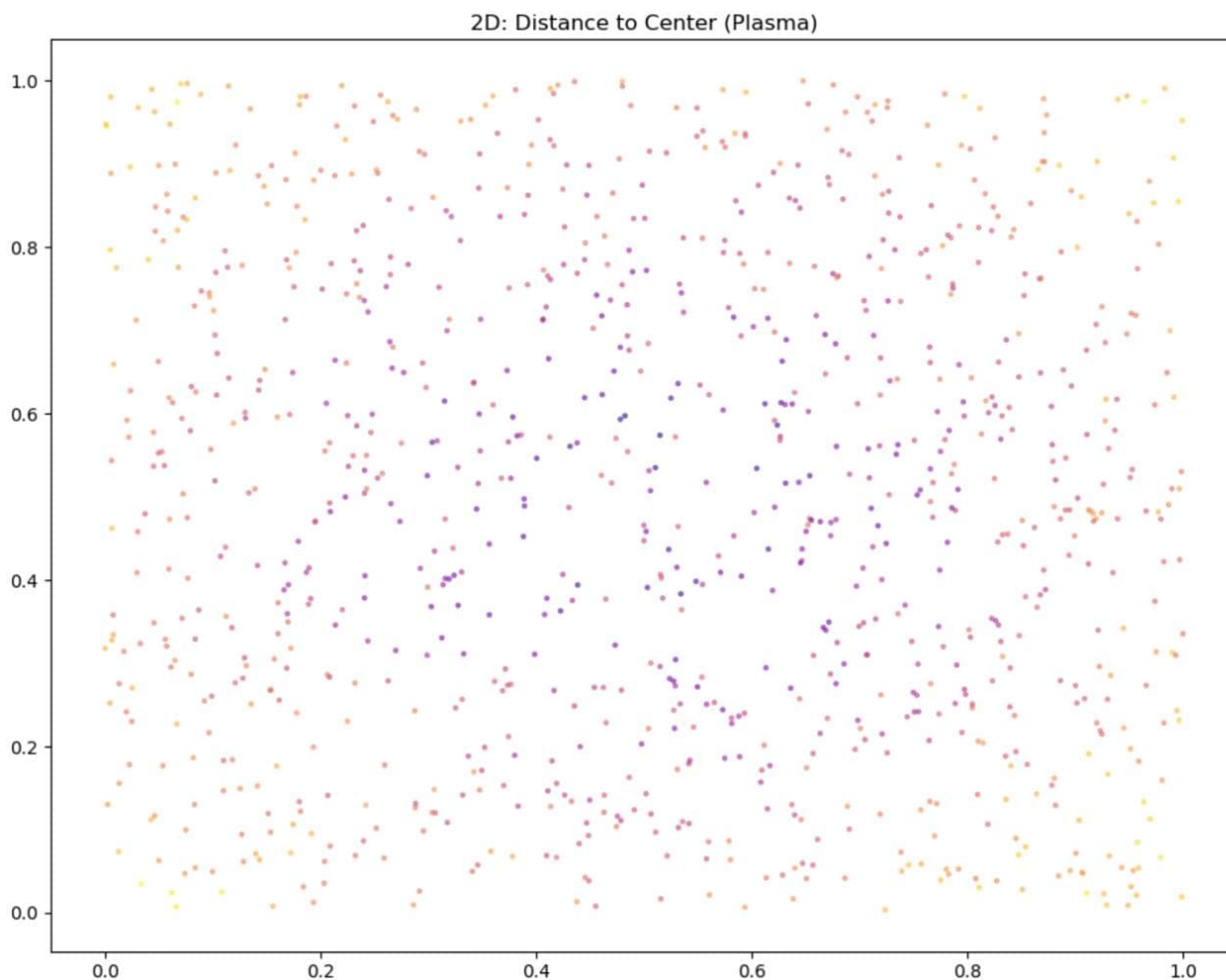


Рис. 3: Альтернативная схема раскраски (расстояние до центра)

### 1.3.3. Анализ результатов

1. **Связь цвета и координат:** Метод  $RGB=XYZ$  интуитивно понятен и позволяет визуально оценить распределение точек в пространстве.
2. **Статичная vs интерактивная визуализация:** Matplotlib подходит для быстрых 2D-просмотров, Plotly — для детального исследования 3D-структур благодаря возможности вращения и масштабирования.
3. **Альтернативные схемы:** Раскраска по расстоянию до центра эффективно выделяет пространственную структуру облака, особенно в 3D-представлении.

## 1.4. Задание 4: Работа со скалярными полями

### 1.4.1. Постановка задачи

Цель задания — освоить методы создания, преобразования и анализа скалярных полей, связанных с облаками точек, включая операции сглаживания, вычисле-

3D: Distance to Center (Plotly)

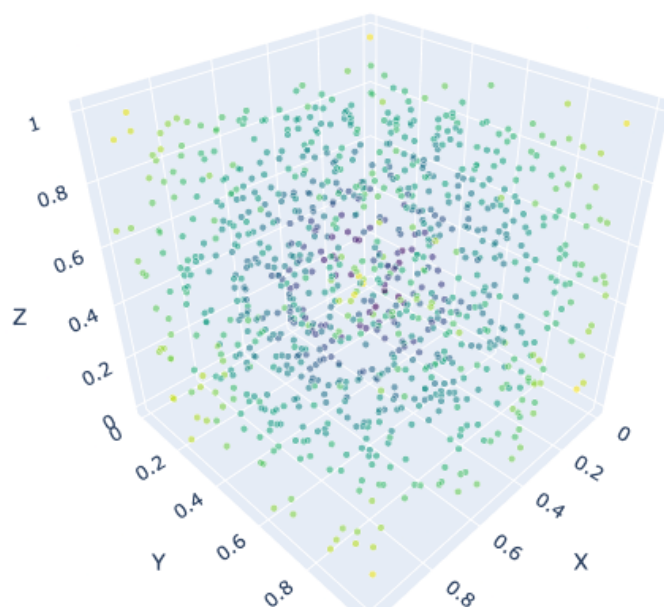


Рис. 4: 3D-визуализация альтернативной схемы в Plotly

ния градиента, фильтрации и визуализации.

#### 1.4.2. Ход выполнения

Работа проводилась с облаком точек "Рождественский медведь" (40 000 точек после субдискретизации).

Point Cloud

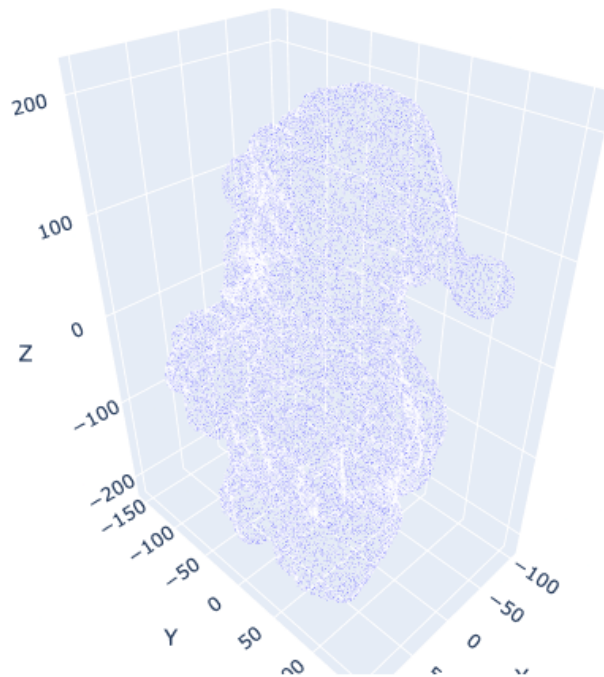


Рис. 5: Исходное облако точек "Рождественский медведь"

#### 1. 1-3. Базовые операции со скалярными полями

Созданы два скалярных поля:

- **const<sub>field</sub>**: постоянное поле со значением 10.0
- **height**: поле высоты (координата Z)

Продемонстрированы операции:

- Умножение поля на число ( $\text{const}_{\text{field}} \times 3 = 30.0$ )
- Сложение с числом ( $30.0 - 5 = 25.0$ )

Результат: постоянное поле осталось константным после операций.

Without smoothing

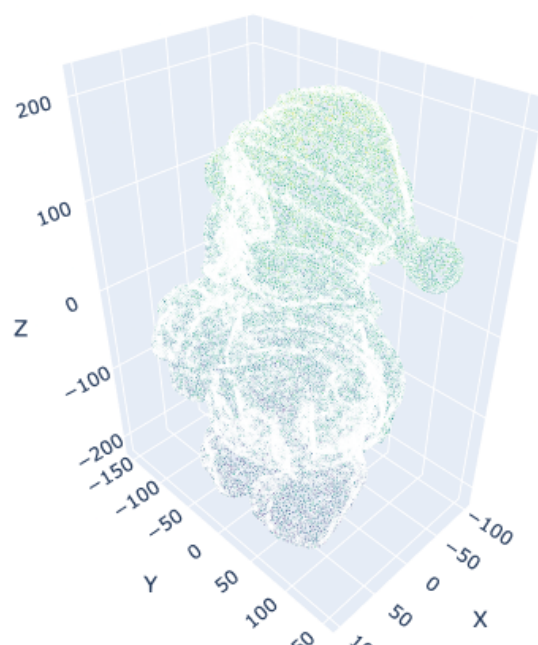


Рис. 6: Поле высоты без сглаживания

With smoothing

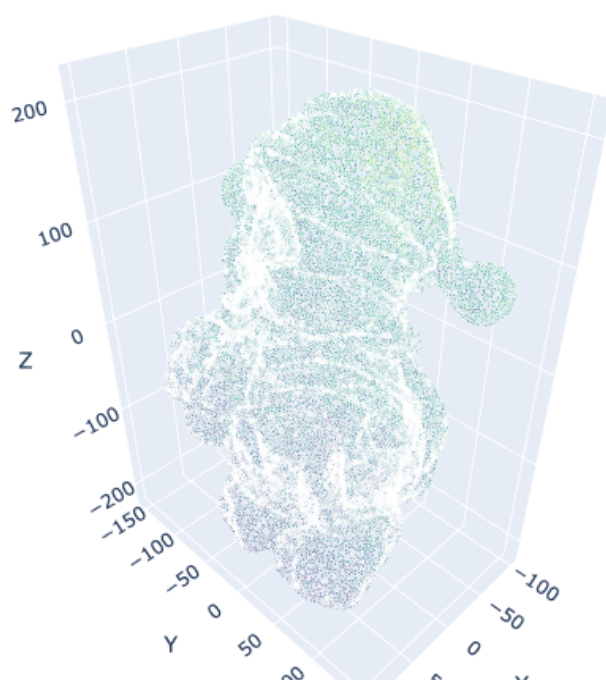


Рис. 7: Поле высоты после гауссова сглаживания

## 2. 4. Гауссово сглаживание

Применено гауссово сглаживание к полю зашумлённой высоты (**noised<sub>height</sub>**). Для постоянного поля сглаживание не даёт эффекта.

## 3. 5. Вычисление градиента

Вычислен градиент поля высоты. Для постоянного поля градиент равен нулю.

### Task 5: Hight field

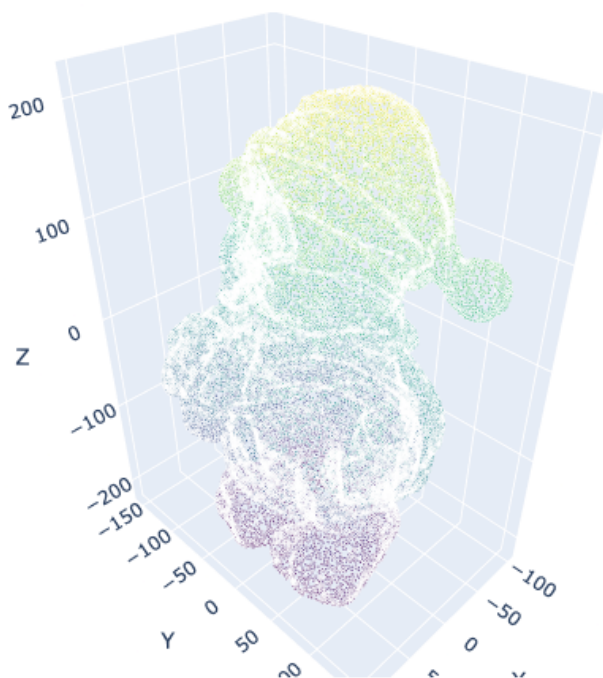


Рис. 8: Поле высоты

## 4. 6. Сглаживание скользящим средним

Применён фильтр скользящего среднего к зашумлённому полю высоты.

## 5. 7. Преобразование скалярного поля в RGB

Скалярные поля преобразованы в цвета с использованием цветовой карты **viridis**.

## 6. 8. Статистический анализ

Вычислены статистические параметры полей:

Task 4: Gradient of hight field

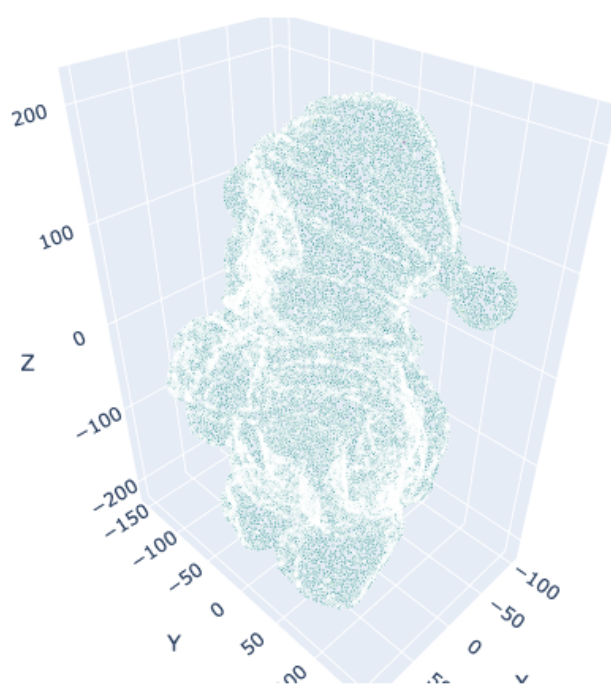


Рис. 9: Градиент поля высоты (почти постоянный)

Task 6: Noised height without filtering

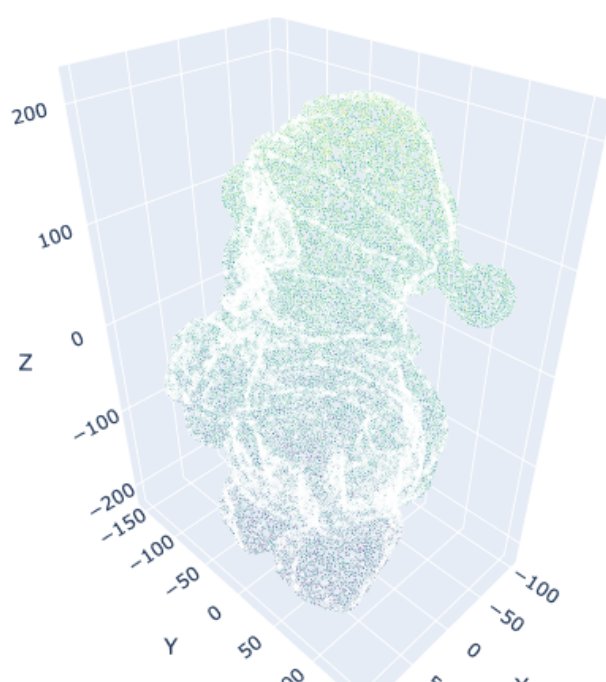


Рис. 10: Зашумлённая высота без фильтра

Noised height field with filtering

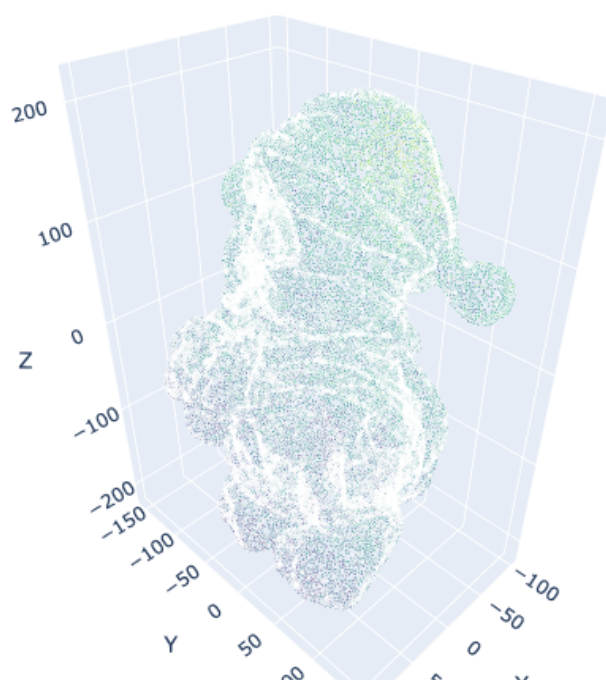


Рис. 11: Зашумлённая высота после фильтра скользящего среднего

Поле	Среднее	Ст. отклонение	Минимум	Максимум	Медиана
const <sub>field</sub>	25.000000	0.00000	25.000000	25.000000	25.0000
height <sub>field</sub>	-6.420886	111.32608	-214.142593	214.158005	-7.8196

#### 7. 9-10. Нормализация и интерполяция

Поля нормализованы в диапазон  $[0, 1]$ . Реализована интерполяция пропущенных значений (NaN) с использованием линейной интерполяции.

#### 8. 11. Фильтрация по значению поля

Выполнена фильтрация точек по значениям скалярных полей:

- const<sub>field</sub>: все точки (значение 25.0 в диапазоне  $[0, 26]$ )
- height: точки с высотой от -100 до 100

После фильтрации получено подмножество точек.

#### 9. 12. Использование скалярного поля как координаты

Продемонстрирована возможность замены координаты Z на значение скалярного поля.

#### 10. 13. Удаление скалярного поля

Продемонстрирована операция удаления скалярного поля из облака точек.

### 1.4.3. Анализ результатов

1. **Операции с полями:** Базовые операции (сложение, умножение) работают корректно, что подтверждает гибкость работы со скалярными полями.
2. **Сглаживание:** Гауссов фильтр и скользящее среднее эффективно уменьшают шум в данных, что особенно важно для зашумлённых измерений.
3. **Градиент:** Вычисление градиента позволяет выделять области с быстрым изменением значений поля, что полезно для анализа рельефа или других пространственных характеристик.
4. **Визуализация:** Преобразование скалярных полей в цвета с помощью цветных карт делает данные более наглядными и интерпретируемыми.

## 1.5. Ответы на контрольные вопросы

1. **Почему для RGB значения используются числа от 0 до 1 или от 0 до 255?** Диапазоны 0-1 и 0-255 являются стандартными представлениями интенсивности цвета. Первый удобен для вычислений с плавающей точкой, второй — для хранения и отображения.

Задание 12:  $Z = \text{const}$  field

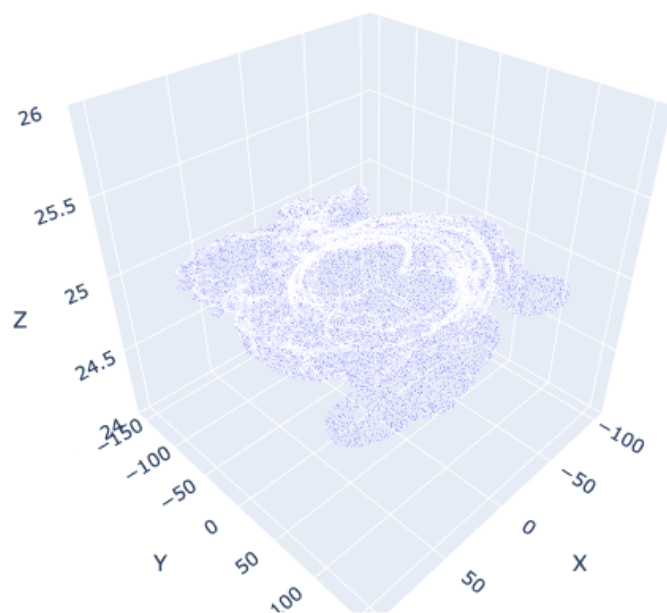


Рис. 12: Использование скалярного поля как координаты  $Z$

2. **Как можно задать цвет точек на основе скалярной величины?** Скалярное значение нормализуется в диапазон  $[0, 1]$ , затем применяется цветовая карта (colormap), которая сопоставляет каждому значению цвет RGB.
3. **Чем отличается статичная визуализация (Matplotlib) от интерактивной (Plotly)?** Matplotlib создаёт статические изображения, Plotly — интерактивные веб-визуализации с возможностью вращения, масштабирования, выделения областей.
4. **Как выбрать подходящую цветовую карту?** Для последовательных данных (высота, температура) подходят плавные карты (viridis, plasma), для категориальных данных — дискретные карты (tab10, Set3).
5. **Что произойдёт, если не нормализовать значения при генерации цветов?** Цвета будут некорректными: значения за пределами  $[0, 1]$  будут обрезаны, что приведёт к потере информации и искажению визуализации.

## 1.6. Выводы

1. **Задание 2:** Освоены методы визуализации облаков точек с цветовой информацией. Интерактивная визуализация в Plotly оказалась значительно удобнее для исследования 3D-структур по сравнению со статичной в Matplotlib.
2. **Задание 4:** Освоены основные операции работы со скалярными полями: создание, преобразование, сглаживание, вычисление градиента, фильтрация. Эти операции являются фундаментальными для анализа пространственных данных.

## 1.7. Заключение

В ходе выполнения заданий 2 и 4 были успешно освоены ключевые аспекты работы с облаками точек: визуализация с использованием цветовой информации и обработка скалярных полей. Реализованные методы и алгоритмы образуют базовый инструментарий для работы с трёхмерными данными, который может быть расширен и адаптирован для решения конкретных прикладных задач в компьютерном зрении, робототехнике, геодезии и других областях.