

Министерство образования и науки Российской Федерации
Московский физико-технический институт (государственный университет)
Физтех-школа радиотехники и компьютерных технологий
Кафедра интеллектуальных информационных систем и технологий

Работа с цветами в облаках точек с использованием
Python, NumPy, Matplotlib и Plotly

Автор: студент группы М01-504а
Гавриков Павел Андреевич
Введение в компьютерное зрение

Преподаватель:

Дата выполнения: 07.12.2025

Долгопрудный, 2025

Цели и задачи работы

Цели работы

Изучить основы визуализации облаков точек и работы с цветовыми схемами, используя Python, NumPy, Matplotlib и Plotly, а также освоить методы назначения цветов на основе геометрических свойств точек.

Задачи работы

1. Сгенерировать облако точек размером $N = 1000$, равномерно распределённое в пределах куба $[0, 1]^3$.
2. Назначить каждому элементу облака цвет, используя координаты точки как компоненты цвета: $R = X$, $G = Y$, $B = Z$, обеспечив нормализацию значений в диапазоне $[0, 1]$.
3. Выполнить 2D-визуализацию облака точек в проекции на плоскость XY средствами Matplotlib, применяя назначенные цвета.
4. Реализовать 3D-интерактивную визуализацию точечного облака с помощью Plotly, отображая для каждой точки её положение и цвет.
5. (Опционально) Создать альтернативную схему раскраски, основанную на расстоянии от центра куба:
вычислить расстояние каждой точки до центра $(0.5, 0.5, 0.5)$, выполнить нормализацию расстояний, применить colormap (например, `viridis`) для получения RGB-цветов.

Теоретическая часть

1. Что такое облако точек?
Облако точек — это набор точек в 3D-пространстве, каждая из которых имеет координаты (X, Y, Z) . Его используют для представления формы объектов, полученной, например, с 3D-сканеров или при генерации данных.
2. Как представляется цвет в RGB?
RGB описывает цвет как комбинацию трёх компонент: красной (R), зелёной (G) и синей (B). Каждая компонента обычно задаётся в диапазоне от 0 до 1 (или 0–255), и их смешение образует итоговый цвет.
3. Отличие визуализации в Matplotlib и Plotly
Matplotlib создаёт статичные изображения и хорошо подходит для научных публикаций. Plotly обеспечивает интерактивность — можно вращать, увеличивать и исследовать 3D-облака точек прямо в окне браузера или ноутбука.

Ход работы

Шаг 1. Генерация облака точек

Создаём набор из $N = 1000$ точек, равномерно распределённых внутри куба $[0, 1]^3$. Каждая точка имеет координаты (X, Y, Z), представленные случайными числами в диапазоне от 0 до 1.

Код:

```
import numpy as np  
  
N = 1000  
points = np.random.rand(N, 3) # X, Y, Z координаты в пределах [0, 1]
```

Шаг 2. Назначение цветов на основе координат точек

Каждая координата используется как соответствующий компонент цвета:

- красная компонента $R = X$
- зелёная компонента $G = Y$
- синяя компонента $B = Z$

Так как координаты уже находятся в диапазоне $[0, 1]$, дополнительная нормализация не требуется.

Код:

```
colors = points.copy() # Цвета задаются по координатам точек (R=X,  
G=Y, B=Z)
```

Шаг 3. Визуализация облака точек в 2D (Matplotlib)

Строим проекцию облака на плоскость XY. Цвет каждой точки определяется из массива colors.

Код:

```
import matplotlib.pyplot as plt  
  
plt.figure(figsize=(6, 6))  
plt.scatter(points[:, 0], points[:, 1], c=colors, s=10)  
plt.xlabel("X")  
plt.ylabel("Y")  
plt.title("2D визуализация облака точек (Matplotlib)")  
plt.grid(True)  
plt.show()
```

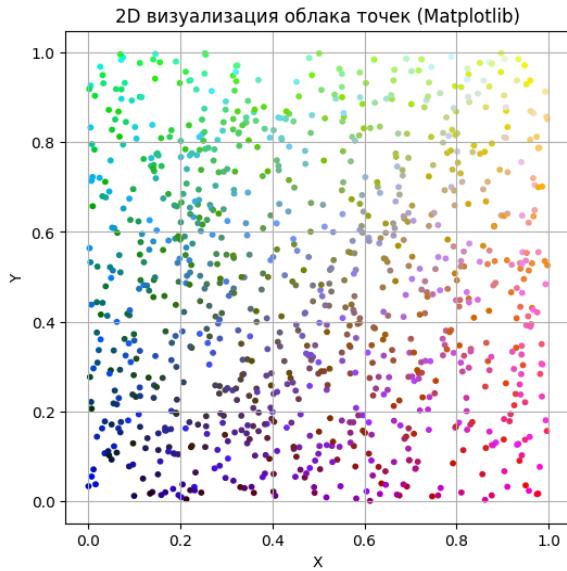


Рис. 1. Визуализация облака точек в 2D

Шаг 4. 3D-визуализация облака в Plotly

Используем интерактивную трёхмерную визуализацию. Цвет каждой точки преобразуется в формат "rgb(r, g, b)" с диапазоном значений от 0 до 255.

Код:

```
import plotly.graph_objects as go

fig = go.Figure()
data=[go.Scatter3d(
    x=points[:, 0],
    y=points[:, 1],
    z=points[:, 2],
    mode='markers',
    marker=dict(
        size=3,
        color=[
            'rgb({}, {}, {})'.format(int(r * 255), int(g * 255), int(b * 255))
            for r, g, b in colors], ))]
fig.update_layout(
    title='3D визуализация облака точек (Plotly)',
    scene=dict(
        xaxis_title='X',
        yaxis_title='Y',
        zaxis_title='Z' ))
```

```
fig.show()
```

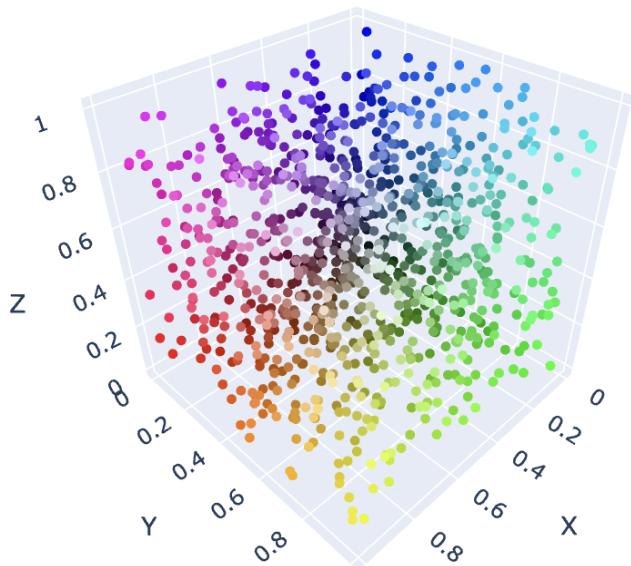


Рис. 2. Визуализация облака точек в 3D

Шаг 5. Дополнительная раскраска (по желанию)

Формируем альтернативную цветовую карту на основе расстояния до центра куба (0.5, 0.5, 0.5). Чем ближе точка к центру, тем более тёмной она будет по палитре viridis.

Код:

```
center = np.array([0.5, 0.5, 0.5])
dist = np.linalg.norm(points - center, axis=1)
norm_dist = (dist - dist.min()) / (dist.max() - dist.min())
colors_alt = plt.cm.viridis(norm_dist)[:, :3] # Берём только RGB
компоненты
```

```
fig_alt.update_layout(title='3D визуализация с альтернативным  
раскрашиванием (Plotly)')  
fig_alt.show()
```

Ответы на вопросы

1. Почему для RGB значения используются числа от 0 до 1 или от 0 до 255?
Это два стандартных диапазона представления интенсивности цвета:
нормированный (0–1) удобен для математических операций в Python, а 0–255 — традиционный формат для экранов и изображений. Они полностью эквивалентны: $1 = 255/255$.
2. Как можно задать цвет точек на основе скалярной величины?
Сначала вычисляют скаляр (например, расстояние или высоту), затем нормализуют его и передают в colormap, который преобразует его в RGB-цвет. Это позволяет визуализировать распределение величины с помощью градиента цветов.
3. Чем отличается статичная визуализация (Matplotlib) от интерактивной (Plotly)?
Matplotlib создаёт статичные изображения и подходит для отчётов и публикаций. Plotly — интерактивный инструмент, позволяющий вращать, масштабировать и исследовать данные прямо в графике.
4. Можно ли использовать другую цветовую карту (colormap) в Matplotlib? Как?
Да, colormap можно выбрать заменой имени карты, например:
`plt.scatter(x, y, c=data, cmap='plasma')`. Matplotlib поддерживает множество схем: '`viridis`', '`plasma`', '`inferno`', '`jet`' и др.
5. Что произойдёт, если не нормализовать значения при генерации цветов?
Цвета могут быть интерпретированы неправильно: при выходе за допустимый диапазон они становятся слишком яркими, слишком тёмными или Matplotlib/Plotly просто обрежут значения. Итоговая визуализация будет некорректной и непредсказуемой.

Выводы

В работе были изучены способы задания цветов точкам на основе их координат и геометрических свойств, а также методы визуализации облаков точек в 2D и 3D. Дополнительно освоена работа с цветовой картой и нормализацией значений для корректного отображения цветов.

Интерактивная визуализация позволяет свободно вращать, масштабировать и исследовать облако точек под разными углами, что облегчает анализ формы и структуры данных. Такой подход особенно полезен при работе с большими трёхмерными моделями, где статичных изображений недостаточно для понимания геометрии.

Визуализация облаков точек применяется в 3D-сканировании, робототехнике, цифровой картографии, системах навигации и анализе данных лидаров. Также такие методы используются в компьютерной графике, архитектуре, геодезии и обработке медицинских изображений.