



Направление подготовки: «Прикладная математика и информатика»  
Профиль подготовки: «Вычислительные методы и суперкомпьютерные технологии»

**«Технологии VR/AR для 3D-реконструкции помещений:  
алгоритмы сканирования, обработки изображений и  
восстановления поверхностей на основе фотографических данных»**

Доцент кафедры математического обеспечения  
и суперкомпьютерных технологий  
Борисов Н.А.

*oooooooooooooooooooooooooooo*



**Для достижения поставленной цели было выполнено:**

## ■ Реализация ПО для реконструкции комнаты с использованием фотографических данных

## ■ Реконструкция комнаты в VR

✓ **Криминалистика:**

**Воссоздание трехмерной среды, сильно приближенной к реальной для исследования ее с разных углов и перспектив**

## Возможность более детально изучить место преступления без физического присутствия на месте происшествия

## ✓ Обучение

## ✓ Архитектура

## ■ Применение в робототехнике

## Понимание окружения, необходимое для навигации, планирования маршрута и взаимодействия с объектами

- Проведен обзор научной литературы:

- ✓ Технологии AR для реконструкции помещений: PlaneTracking
- ✓ Методы распознавания объектов: плотный поток Фарнебака и стереорегификация с использованием stereoBM
- ✓ Алгоритмы кластеризации и классификации объектов из облака точек

■ Выполнена программная реализация изученных методов:

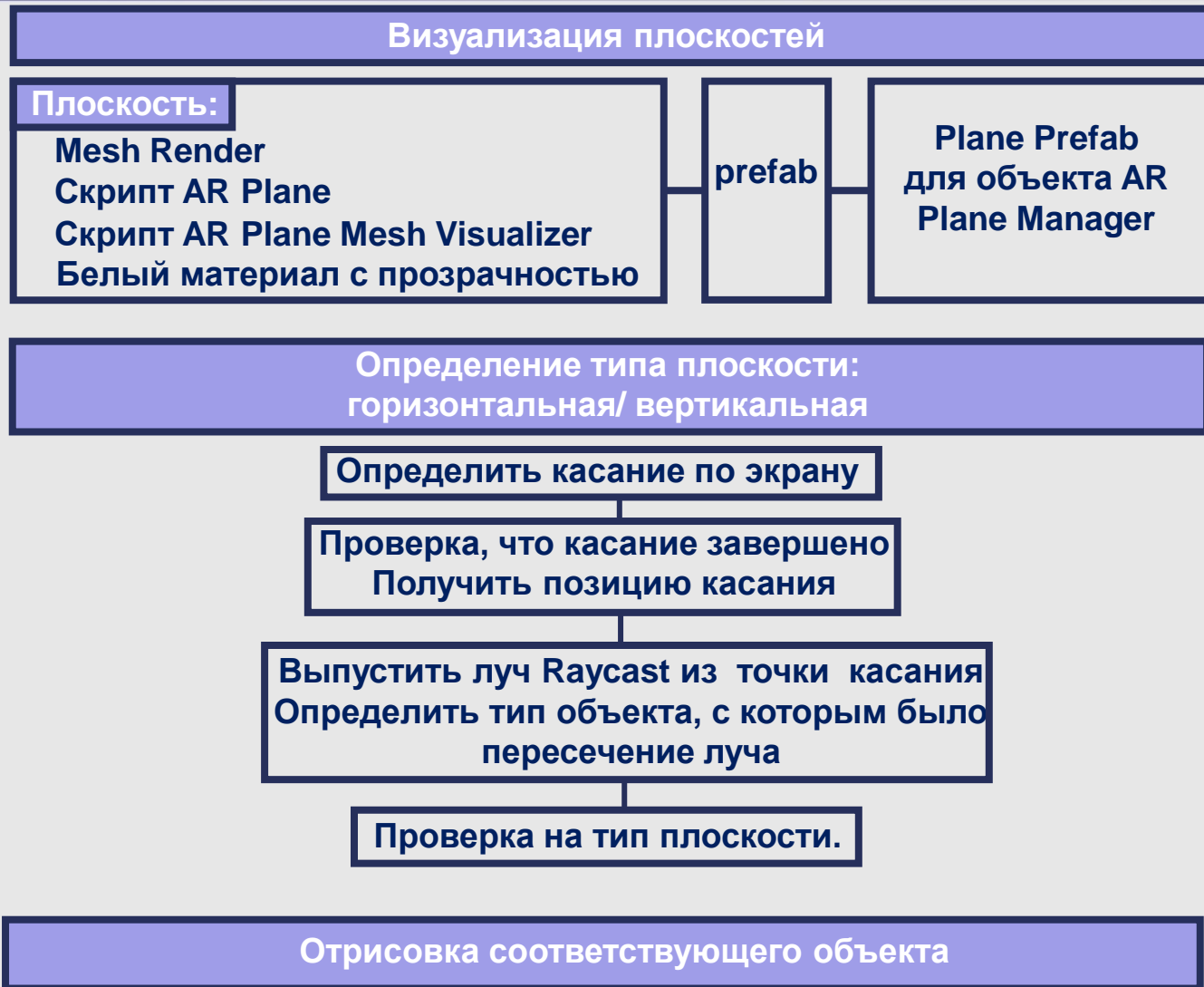
- ✓ **Реконструкция помещения и обработка данных с использованием серии фотографии из центра комнаты в 3DUnity: стены, пол и потолок**
- ✓ **PlaneTracking**
- ✓ **Плотный поток Фарнебака и стереоректификация с использованием stereoBM**
- ✓ **Метод кластеризации DBSCAN**
- ✓ **Метод классификации, основанный на анализе геометрических характеристик и применении операции свертки**
- ✓ **Алгоритм классификации с использованием глубокой сверточной нейронной сети**



# PlaneTracking

## Постановка задачи:

- Выполнить программную реализацию для поиска вертикальных и горизонтальных поверхностей в AR
- При нажатии на горизонтальную плоскость – добавлять на сцену сферу; при нажатии на вертикальную плоскость – добавлять куб

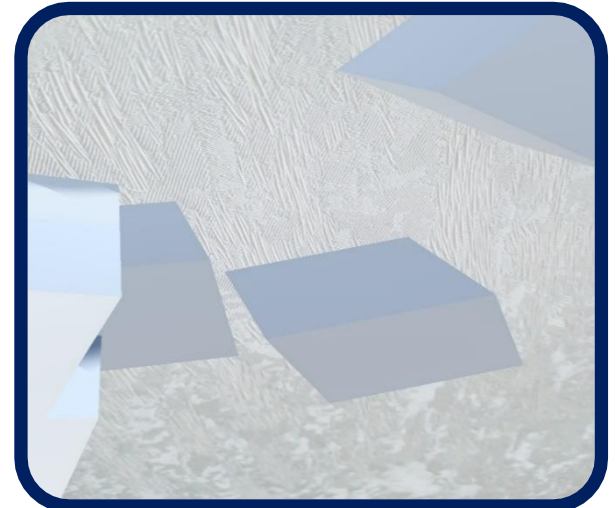


## Постановка задачи:

- Выполнить программную реализацию для поиска вертикальных и горизонтальных поверхностей в AR
- При нажатии на горизонтальную плоскость – добавлять на сцену сферу; при нажатии на вертикальную плоскость – добавлять куб

Программная реализация работает корректно

**Правильно распознает вертикальные и горизонтальные плоскости, используя технологии дополненной реальности AR.**



## Постановка задачи:

- Выполнить программную реализацию для сканирования объектов с фотографии
- Применить метод оптического потока

## Вход:

- ### ■ Две фотографии с некоторым шагом

### Оптический поток методом Фарнебека: calcOpticalFlowFarneback

## Выход:

- оптический поток между двумя изображениями:  $v_x, v_y$   
горизонтальная, вертикальная компонента скорости

- ✓  $(u, v)$  описывает смещение пикселя из `img1` в `img2`

### Преобразование векторов $v_x, v_y$ в полярные координаты

**Выход:**

- Модуль вектора (длина):  $\sqrt{u^2 + v^2}$
- Угол вектора в полярных координатах:  $\arctan \frac{v}{u} \in [0; 2\pi)$

## Метод Canny для обнаружения границ не изображении

## Метод триангуляции: поиск координат пикселей в пространстве

### ***B – шаг между двумя стереоизображениями***

$w = w_1 = w_2$  – ширина изображения

$h = h_1 = h_2$  – высота изображения

$f$  – фокусное расстояние

**X, Y – координаты объекта**

**Z – расстояние до объекта (глубина)**

$$Z = \frac{B \cdot f}{\Delta x}$$

$$\begin{aligned} X &= \frac{Z(x - \frac{w}{2})}{f} \\ Y &= \frac{Z(y - \frac{h}{2})}{f} \end{aligned}$$

- ✓  $x, y$  - пиксель
- ✓  $\Delta x$  – изменение положения пикселя на фотографии после шага В
- ✓ Если  $Z$  подставить в формулу для  $X$  и  $Y$ , то они не зависят от  $f$



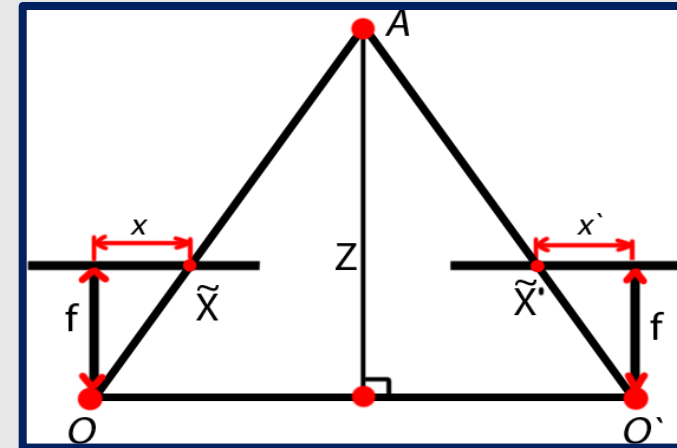
## Постановка задачи:

- Выполнить программную реализацию для сканирования объектов с фотографии
- Применить метод оптического потока

## Вход:

- Две фотографии с некоторым шагом

- $O$  – положение камеры 1 (начало системы координат для 1-ой камеры)
- $O'$  – положение камеры 2 (начало системы координат для 2-ой камеры)
- $A$  – точка 3D пространства на пересечении лучей из камеры  $O$  и  $O'$



✓ Координаты т.А ( $X, Y, Z$ ) в системе координат левой камеры

✓ Координаты т.А ( $X', Y', Z'$ ) в системе координат правой камеры.

- $\tilde{X}$  и  $\tilde{X}'$  проекция т. А на изображение камеры  $O$  и камеры  $O'$  соответственно
- $\Delta x$  диспаратность - разница между координатами проекции точки  $\tilde{X}$  и  $\tilde{X}'$  на изображение левой и правой камеры

$$\Delta x = x - x'$$

$\Delta x = \sqrt{x^2 + y^2}$  смещение пикселя между двумя кадрами

$$Z = \frac{B \cdot f}{\Delta x}, X = \frac{Z \left( x - \frac{w}{2} \right)}{f}, Y = \frac{Z \left( y - \frac{h}{2} \right)}{f}$$



## Постановка задачи:

- Выполнить программную реализацию для сканирования объектов с фотографии
- Применить метод , основанный на стереоректификации и stereoBM

## Вход:

- Две фотографии с некоторым шагом

### Чтение изображений в оттенках серого

### Стереоректификация

- ✓ Поиск ключевых точек на изображениях: Oriented FAST and Rotated BRIEF
- ✓ Сопоставление ключевых точек: BFMatcher
- ✓ Выбираем координаты соответствующих ключевых точек
- ✓ Поиск фундаментальной матрицы  $x_1^T F x_2 = 0$ : findFundamentalMat

1.  $\_, <\text{матрица преобразований}>_{1,2} = \text{cv2.stereoRectifyUncalibrated}(\text{<ключ. точки img1>, <ключ. точки img2, <фундаментальная матрица>, img\_size})$

2.  $\text{stereoRectify\_img1} = \text{cv2.warpPerspective}(\text{img1, <матрица преобразований>}_1, \text{img\_size})$

### Построение карты глубины

#### Метод StereoBM

Выход:  
карта диспаратности

- ✓ Карта диспаратности содержит информацию о различиях между точками на двух изображениях
- ✓ Можем определить меру сдвига между пикселями
- ✓ Имеем представление о глубине объектов в сцене
- ✓ Чем больше значение диспаратности для пикселя, тем дальше находится соответствующий объект от камеры



✓ Метод “плотного потока” отработал корректно и правильно выделил смещенные объекты на всех сериях стереоизображений

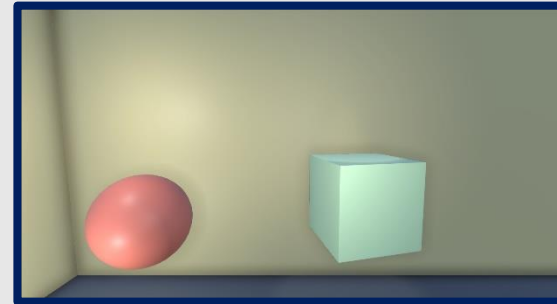
✓ Метод, основанный на стереоректификации и StereoBM, показал искаженные изображения в неправильной перспективе, что и ухудшило результаты.

Требуется предварительная правильная обработка изображения перед поиском карты глубины, т.к. в методе не учитываются характеристики камеры при стереоректификации

✓ В дальнейшем будем применять метод “плотного потока”

## Результаты работы метода плотного потока

Левый кадр



Правый кадр

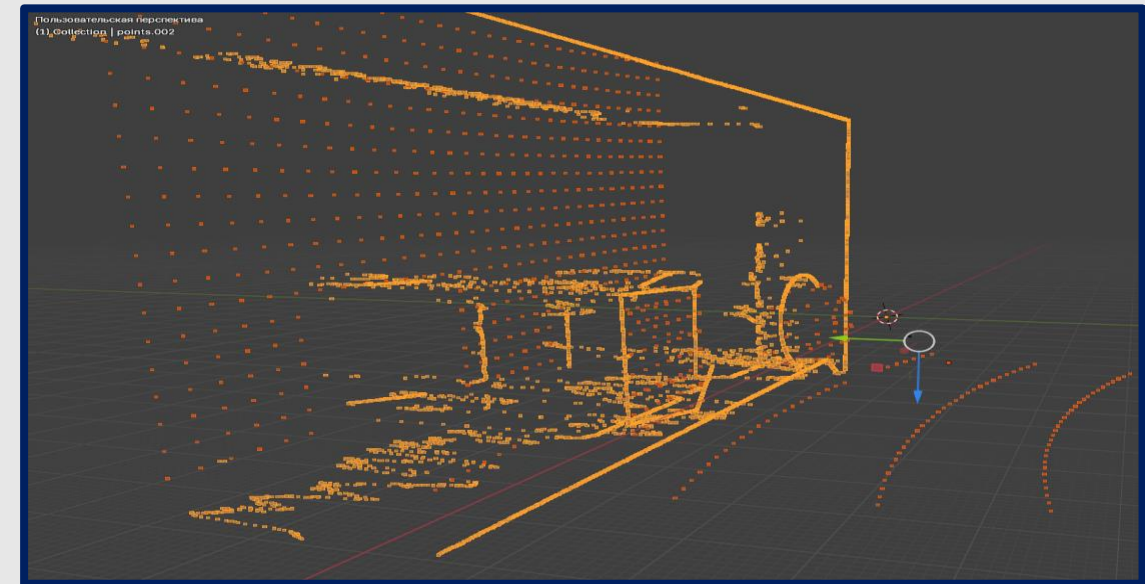
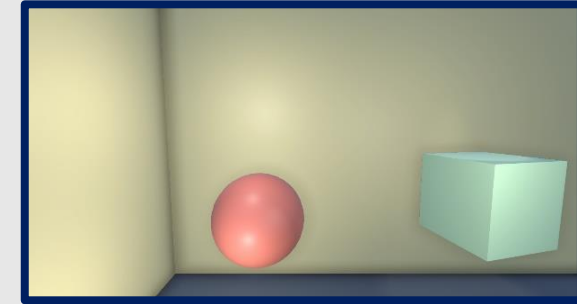


рис.1: Пример1: результат сканирования комнаты





## Результаты работы метода плотного потока

Левый кадр



Правый кадр



рис.2: Пример2: две фотографии с некоторым шагом

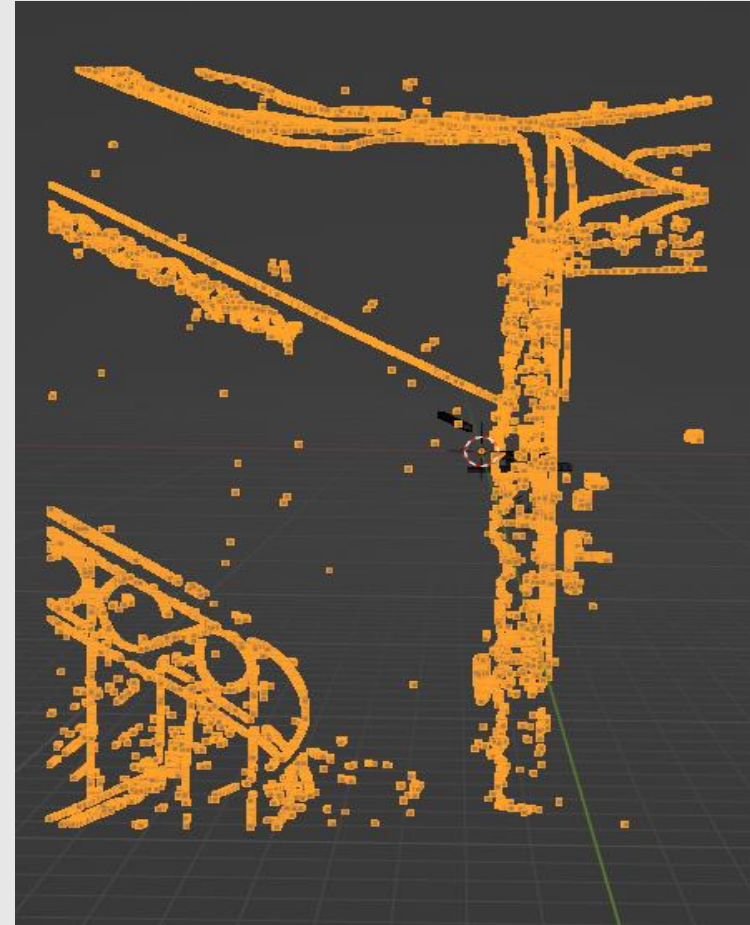


рис.3: Пример2: результат сканирования



✓ Для дальнейшей реконструкции помещения необходимо определить отсканированные объекты

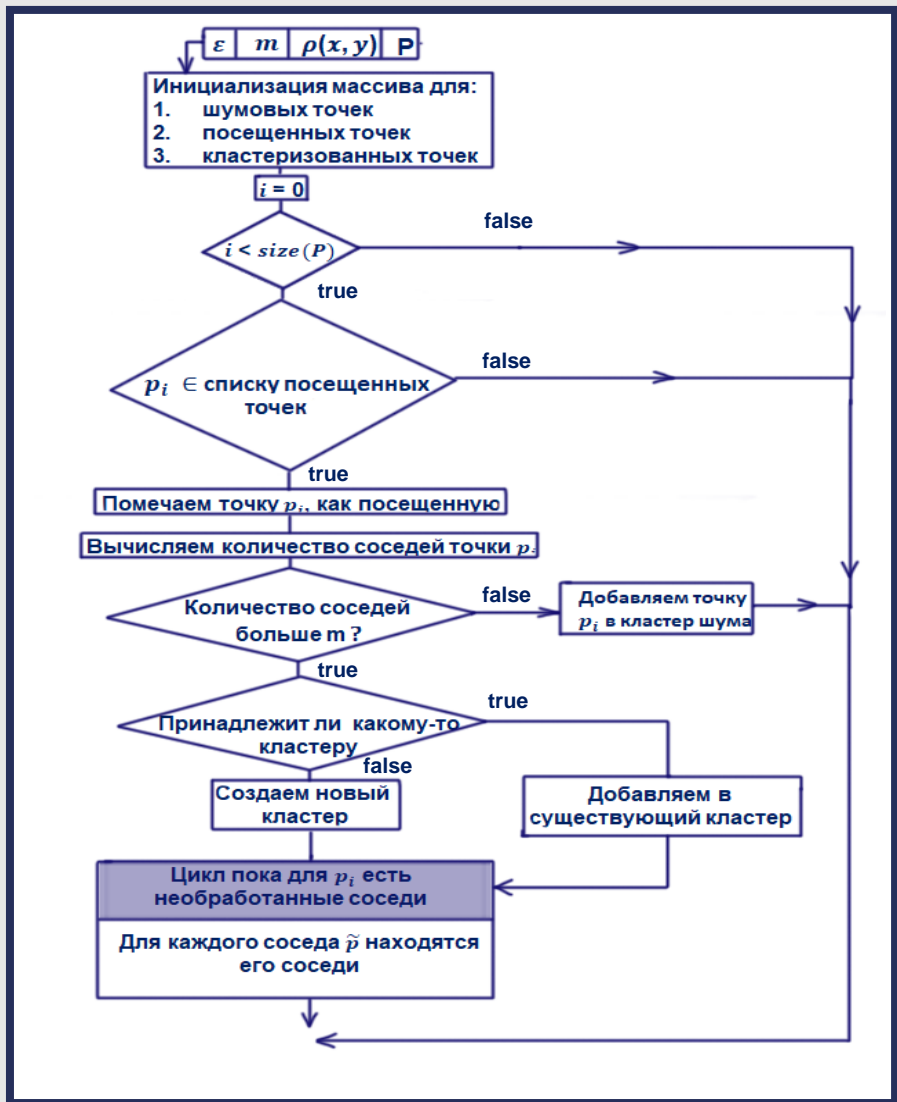
## Постановка задачи:

- Выполнить программную реализацию для кластеризации облака точек

## Вход:

- $\varepsilon$  : радиус  $\varepsilon$ -окрестности
- $m$ : минимальное количество точек кластера
- $\rho(x, y)$ : симметричная функция расстояния
- $P$ : облако точек, полученные после применения метода плотного потока для сканирования объектов

✓ Выполнив кластеризацию, разбиение облака точек на однородные подобласти, мы упростим процесс классификации распознанных объектов



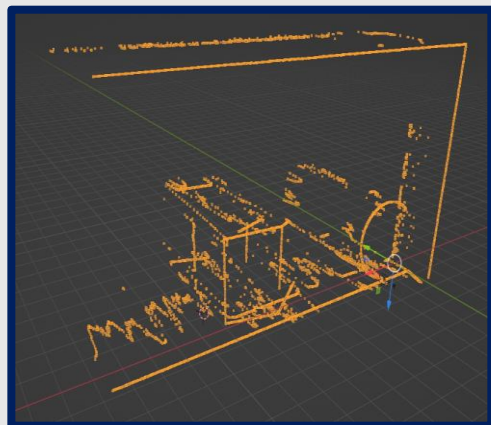
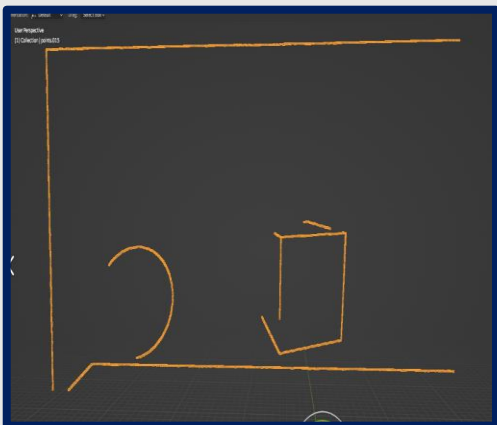
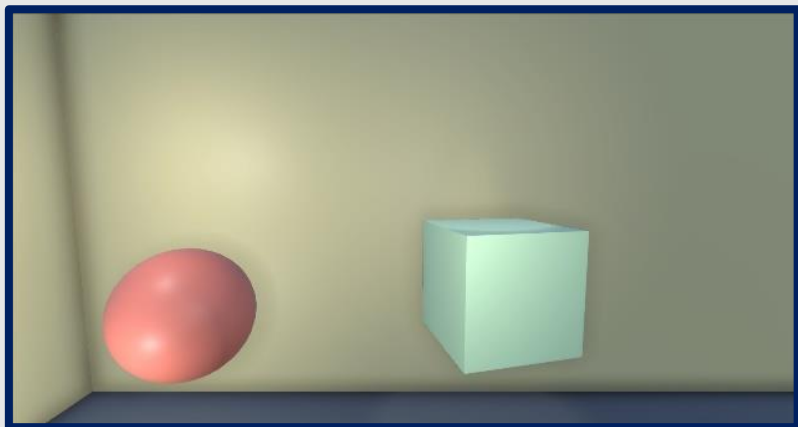


рис.4: результат применения метода плотного потока

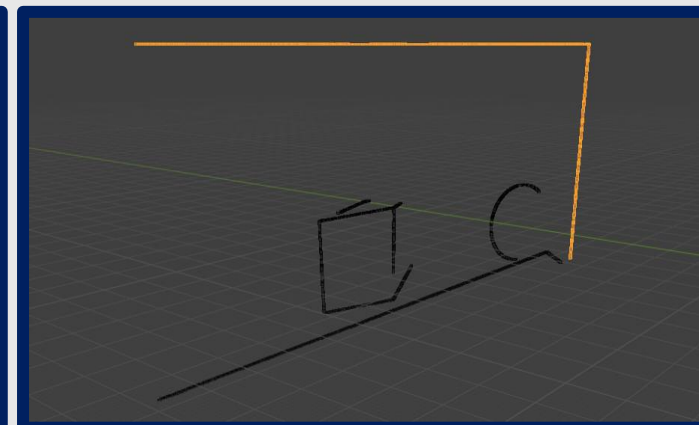
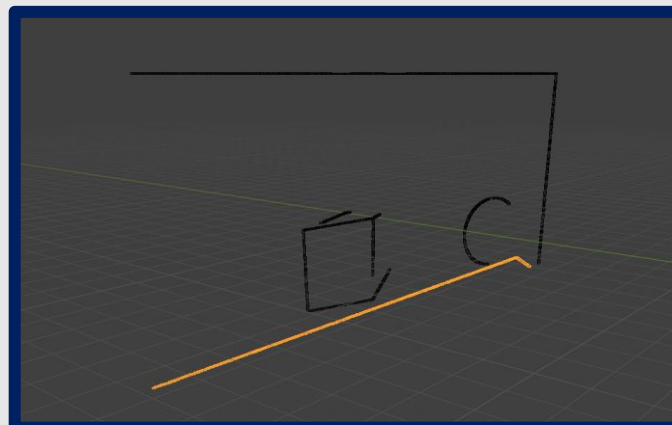
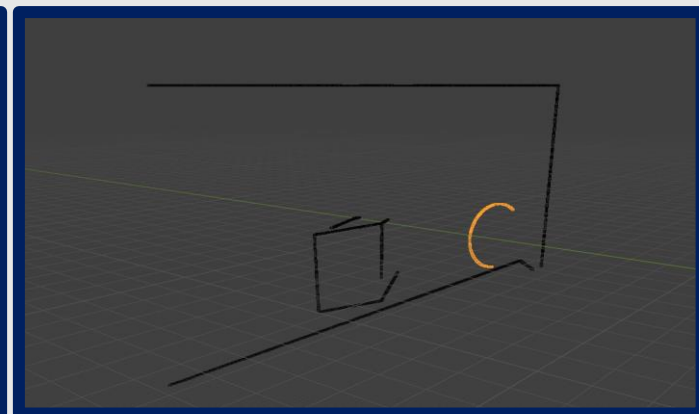
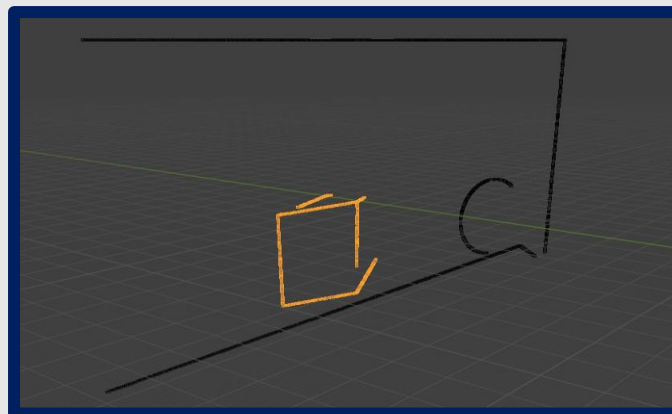


рис.5: результат кластеризации



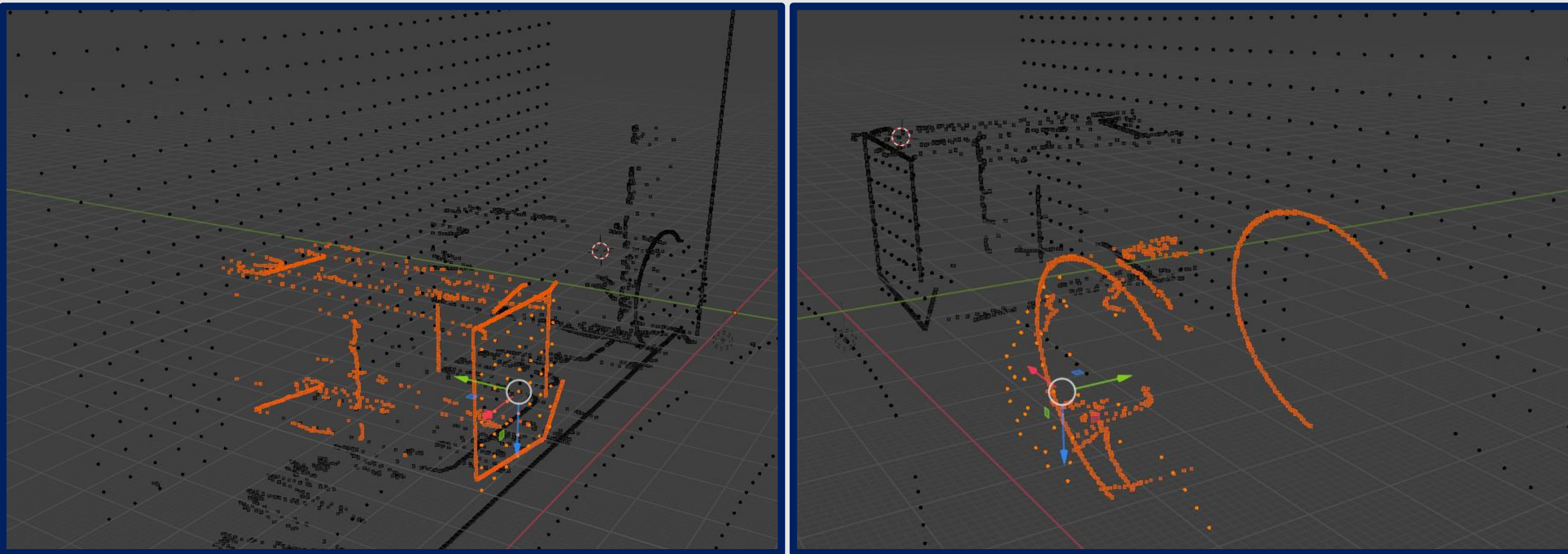


рис.6: результат кластеризации



## Постановка задачи:

- Для классификации необходимо применить метод, основанный на анализе пространственных характеристик
- Применить операции свертки для извлечения характеристик

✓ Анализ расстояния точек до центра объекта позволяет определить его углы.

- Построение графика расстояний позволяет выявить углы, как точки с наибольшим удалением от центра, в отличие от точек на ребрах, имеющих меньшие расстояния.
- Построив график расстояния от номера точки  $i$ , мы можем выделить количество точек, подозрительных на угловые
- Более это заметно на графике расстояния от угла относительно центра: угловым точкам, соответствуют на графике значения локальных экстремумов

### Вычисление центра многоугольника

$$\bar{x} = \frac{x_1 + \dots + x_n}{n}$$

$$\bar{y} = \frac{y_1 + \dots + y_n}{n}$$

### Вычисление вектора расстояний до центра

Координаты точек облака:  $(x_i; y_i) \quad i = 1, \dots, k$

$$d_i = (x_i - \bar{x})^2 + (y_i - \bar{y})^2$$

График  $(i, d_i)$

Нормировка  $d_i$  к диапазону от 0 до 1

### Вычисление вектора полярных углов

$$\Delta x = \bar{x} - x_i$$

$$\Delta y = \bar{y} - y_i$$

$$\alpha_i = \text{np.arctan2}(\Delta x, \Delta y)$$

График  $(\alpha_i, d_i)$

### Применение свертки

$$\tilde{d}_i = \sqrt{\left((1 - \tilde{\alpha}_i^2)^2 + 1\right) - 1}$$

✓  $\tilde{\alpha}_i$  отмасштабированный угол к диапазону от 0 до 1024

✓ Строим график  $(\tilde{\alpha}_i; \tilde{d}_i)$



## Постановка задачи:

- Для классификации необходимо применить метод, основанный на анализе пространственных характеристик
- Применить операции свертки для извлечения характеристик

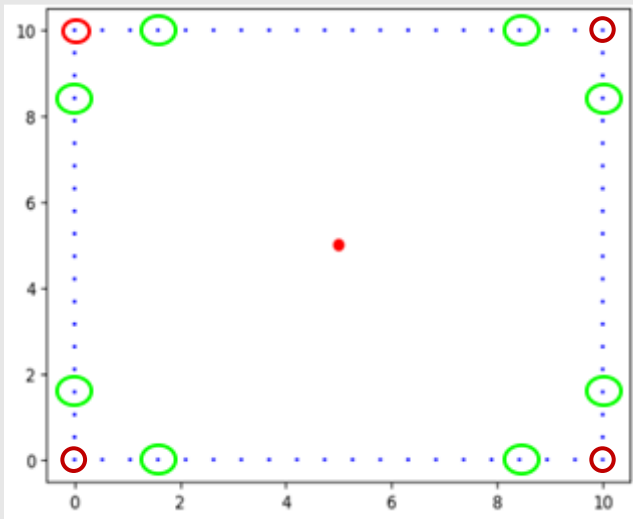


рис.7: Пример «сгенерированный квадрат»

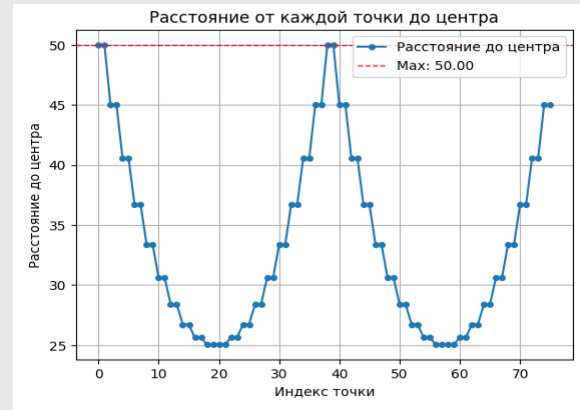


рис.8: График 1 ( $i, d_i$ )



рис.9: График 2 ( $\alpha_i, d_i$ )



рис.10: График 3 ( $\tilde{\alpha}_i, \tilde{d}_i$ )





## Постановка задачи:

- Для классификации необходимо применить метод, основанный на анализе пространственных характеристик
- Применить операции свертки для извлечения характеристик

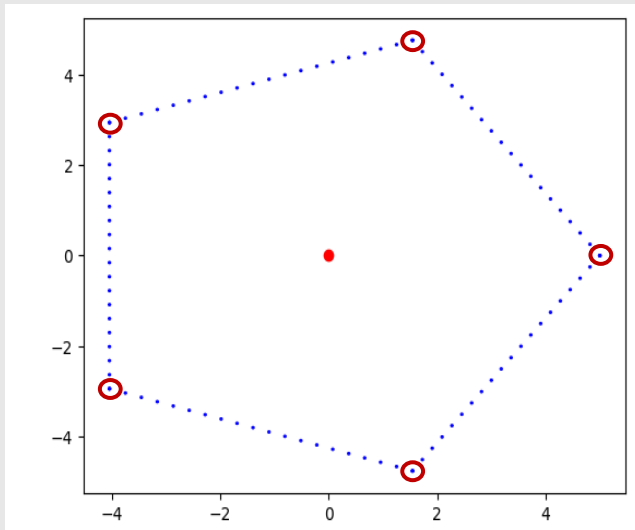


рис.11: Пример «сгенерированный пятиугольник»

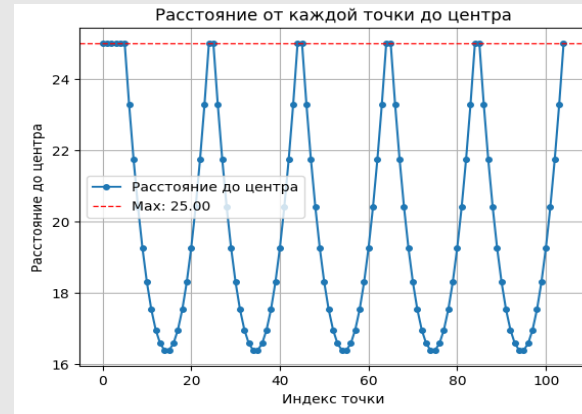


рис.12: График 1 ( $i, d_i$ )

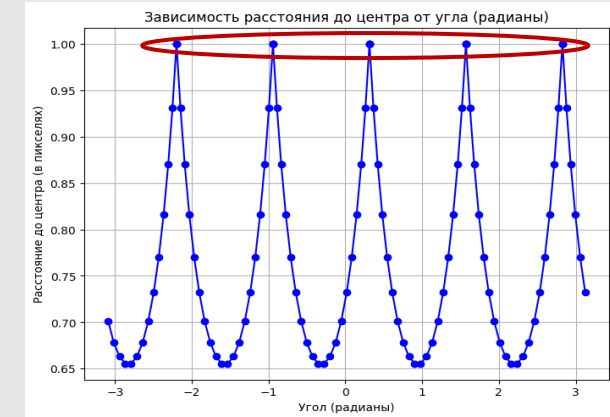


рис.13: График 2 ( $\alpha_i, d_i$ )



рис.14: График 3 ( $\tilde{\alpha}_i, \tilde{d}_i$ )



## Постановка задачи:

- Для классификации необходимо применить метод, основанный на анализе пространственных характеристик
- Применить операции свертки для извлечения характеристик

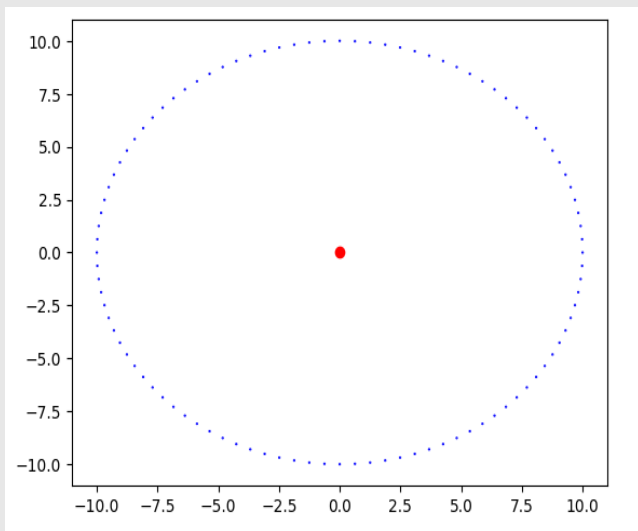


рис.15: Пример «сгенерированная окружность»

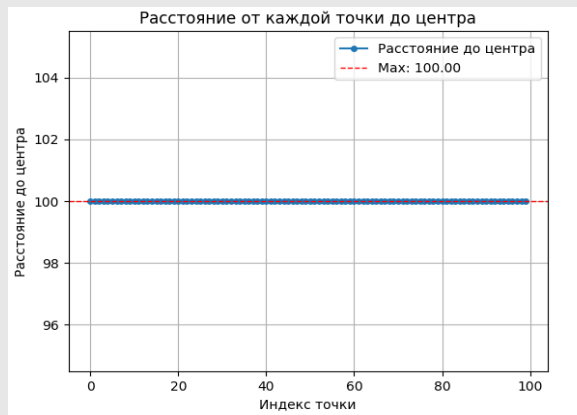


рис.16: График 1 ( $i, d_i$ )

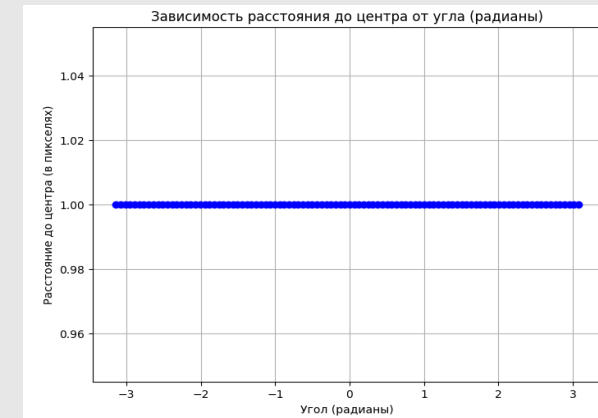


рис.17: График 2 ( $\alpha_i, d_i$ )



рис.18: График 3 ( $\tilde{\alpha}_i, \tilde{d}_i$ )



## Постановка задачи:

- Для классификации необходимо применить метод, основанный на анализе пространственных характеристик
- Применить операции свертки для извлечения характеристик

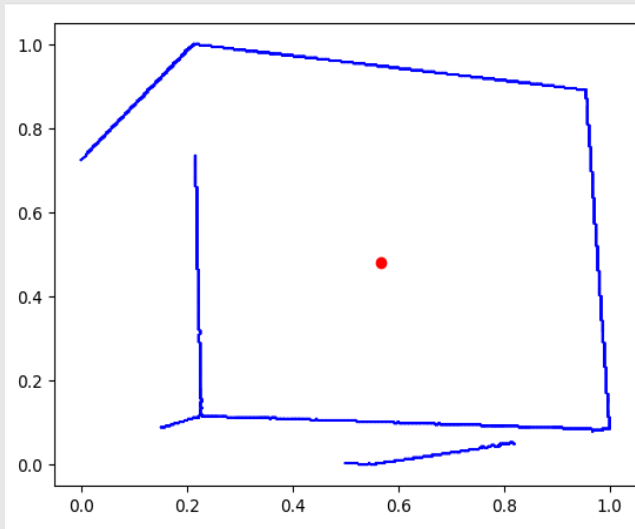


рис.19: Пример «квадрат из облака точек»

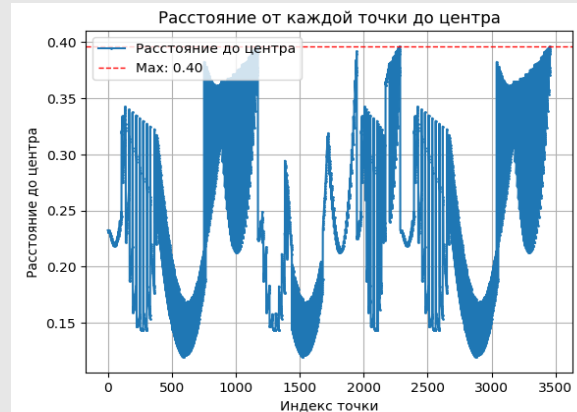


рис.20: График 1 ( $i, d_i$ )

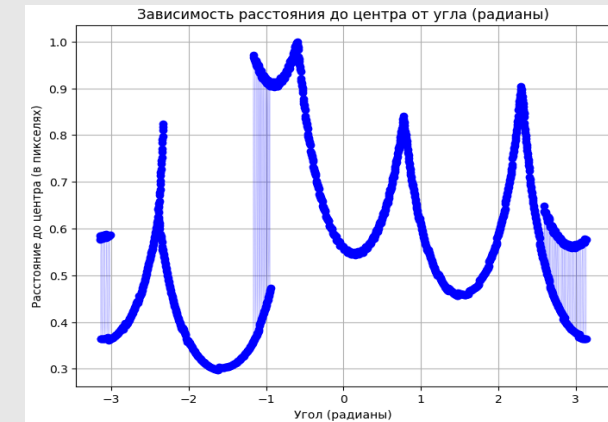


рис.21: График 2 ( $\alpha_i, d_i$ )

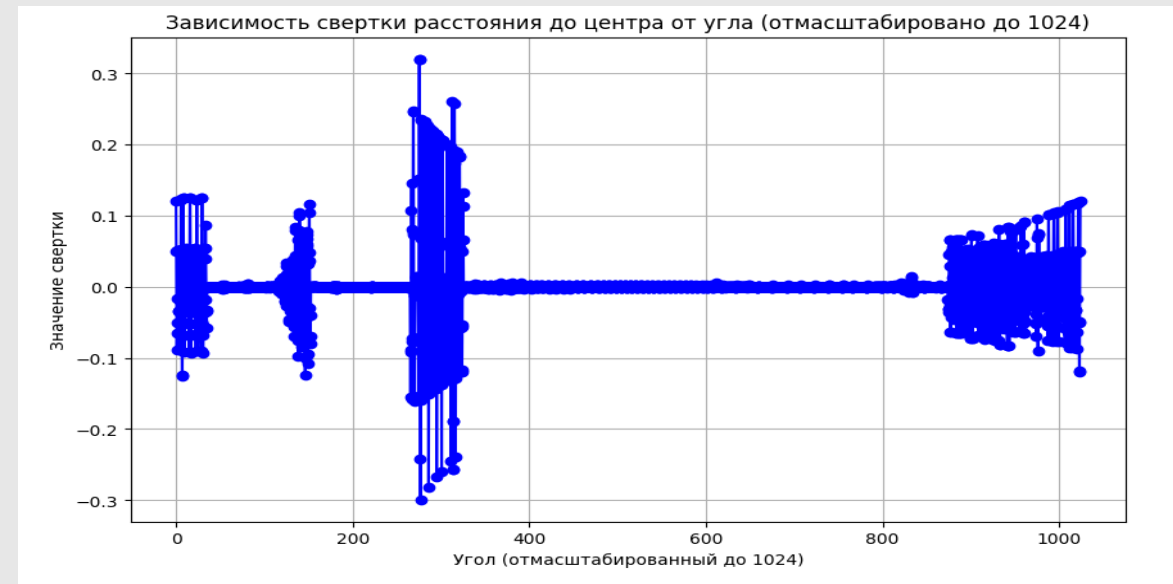


рис.22: График 3 ( $\tilde{\alpha}_i, \tilde{d}_i$ )



## Постановка задачи:

- Для классификации необходимо применить метод, основанный на анализе пространственных характеристик
- Применить операции свертки для извлечения характеристик

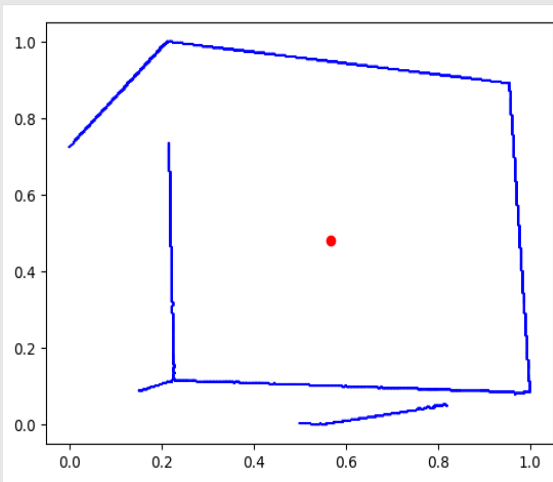


рис.23: Пример «квадрат из облака точек после предобработки»

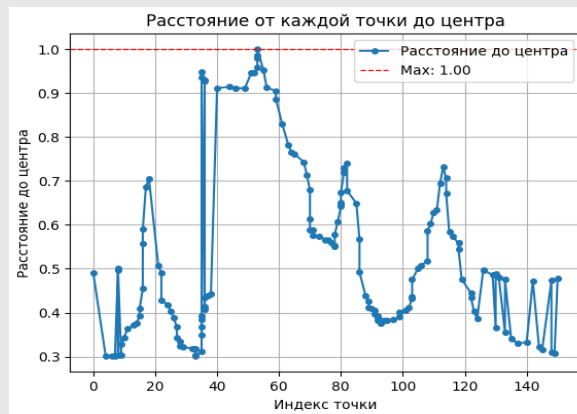
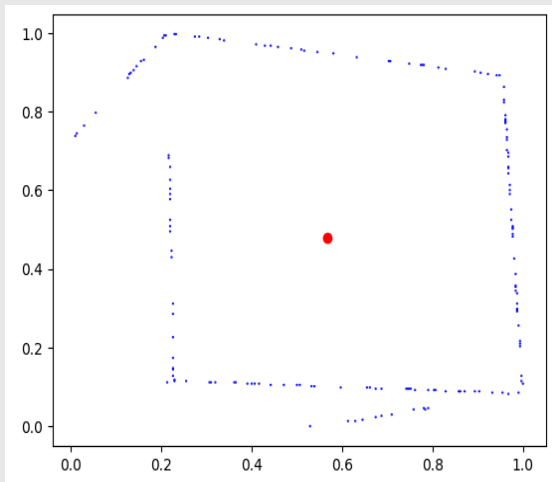


рис.24: График 1 ( $i, d_i$ )

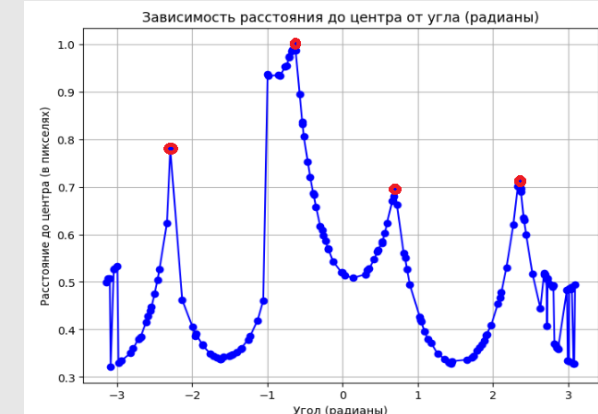


рис.25: График 2 ( $\alpha_i, d_i$ )



рис.26: График 3 ( $\tilde{\alpha}_i, \tilde{d}_i$ )



## Постановка задачи:

- Для классификации необходимо применить метод, основанный на анализе пространственных характеристик
- Применить операции свертки для извлечения характеристик

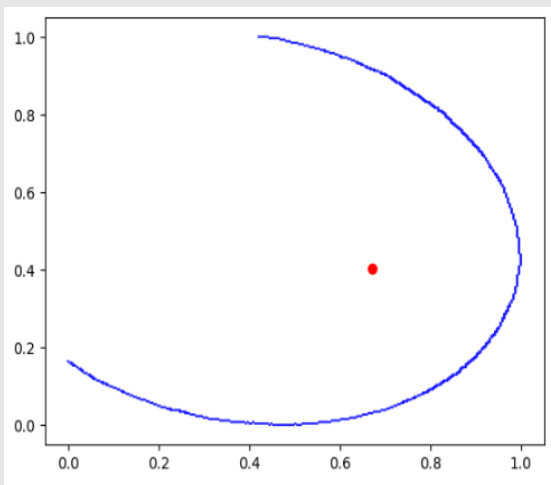


рис.27: Пример «окружность из облака точек после предобработки»

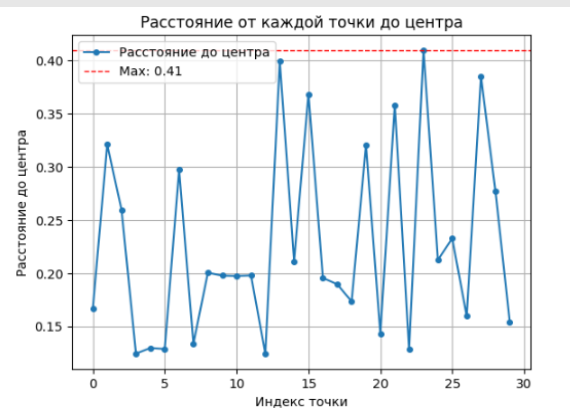
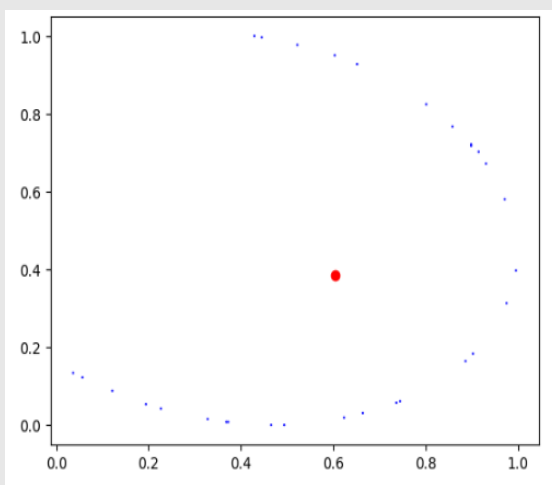


рис.28: График 1 ( $i, d_i$ )

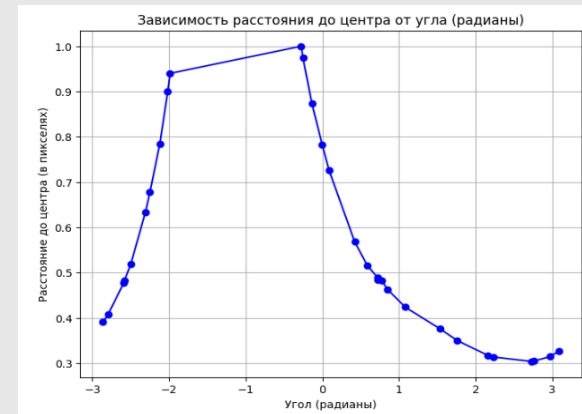


рис.29: График 2 ( $\alpha_i, d_i$ )



рис.30: График 3 ( $\tilde{\alpha}_i, \tilde{d}_i$ )



# Практическое применение (реконструкция помещения)

- ✓ Применение ранее разработанных программных решений позволяет выполнить реконструкцию помещений, а именно получить ее трёхмерную модель повышенной детализации

## Постановка задачи:

- Выполнить программную реализацию реконструкции сцены с фотографии, либо отдельных ее элементов

## Вход:

- Одно изображение помещения

## Выход:

- 3D – облако точек
- 3D – модель, состоящая из полигонов

### Оценка карты глубины D

- Для получения  $D = (D_{x,y})$  используем модель глубокого обучения GLPN (CNN)
- Модель GLPN предобучена на «NYU Depth V2»
  - «NYU Depth V2» включает в себя сцены различных помещений
  - 1449 пар изображений RGB-глубина  $640 \times 480$  пкс.
- Выход:  $D = (D_{x,y}) \in R^{w \times h}$

### 3D – облако точек / 3D – облако граничных точек

- На основе RGBD-изображения  $I_z$ :

$$z_{x^*,y^*} = D_{x^*,y^*}$$

$$x = \frac{(x^* - c_{x^*}) \cdot z_{x^*,y^*}}{f}$$

$$y = \frac{(y^* - c_{y^*}) \cdot z_{x^*,y^*}}{f}$$

- Выход:  $P = \{p_i\} = \{(x_i, y_i, z_i) : i = \overline{0, n-1}\}$ 
  - В мн-ве  $P = \{p_i\}$  можем оставить только точки, принадлежащие границам объектов

### Нормали к полигонам mesh-сетки:

- `estimate_normals()`: для каждой точки  $p^*$ 
  - Находит её  $k$  ближайших соседей
  - Вычисляется точка  $\tilde{p}$
  - Вычисляется матрица ковариации  $C = (C_{ij}) \in R^{3 \times 3}$
  - Вычисление с.ч. и с.в. матрицы  $C$ :  $\lambda_1, \lambda_2, \lambda_3; v_1, v_2, v_3$

Итог:  $n^* = v_1$

$$\tilde{p} = \frac{1}{k} \sum_{i=0}^k p_i$$

$$C = \sum_{i=1}^k (p_i - \tilde{p})(p_i - \tilde{p})^T$$





# Практическое применение (реконструкция помещения)

- ✓ Применение ранее разработанных программных решений позволяет выполнить реконструкцию помещений, а именно получить ее трёхмерную модель повышенной детализации

## Постановка задачи:

- Выполнить программную реализацию реконструкции сцены с фотографии, либо отдельных ее элементов

## Вход:

- Одно изображение помещения

## Выход:

- 3D – облако точек
- 3D – модель, состоящая из полигонов

## Алгоритм Пуассона (Reconstruct surface)

### ■ Вход:

- Предобработанное облако точек:  $\tilde{P} \subseteq P = \{p_i = (x_i, y_i, z_i): i = \overline{0, n-1}\}$
- Множество векторов нормалей к полигону меш сетки в каждой точке  $p_i \in \tilde{P}$ :

$$N = \{n_i: i = \overline{0, n-1}\}$$

### ■ Решение:

- ✓ Векторное поле нормалей  $\vec{V}$
- ✓ Индикаторная функция  $\chi(p)$ :  $\chi(p_i) = \begin{cases} 1, p_i \in \text{int}M \\ 0, p_i \notin R^3 \setminus M \end{cases}$
- ✓ Лемма:

Пусть дано тело  $M$  с границей  $\partial M$ ,  $\chi_M$  – индикаторная функция  $M$ .  $\vec{N}_{\partial M}(p)$  – нормаль к поверхности, направленная внутрь, в точке  $p \in \partial M$ . Сглаживающий фильтр  $\tilde{F}_p(q) = F(q - p)$ , переводящий в точку  $p$ . Тогда градиент сглаженной индикаторной функции равен векторному полю, полученному сглаживанием поля нормалей поверхности:

$$\nabla(\chi F)|_{q_0} = \int_{\partial M} F(q_0 - p) \vec{N}_{\partial M}(p) dp$$



$$\nabla \chi = \vec{V}$$



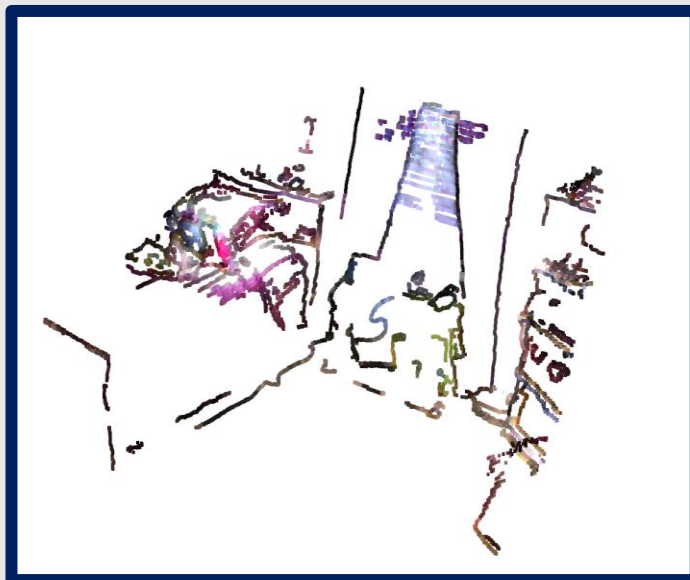
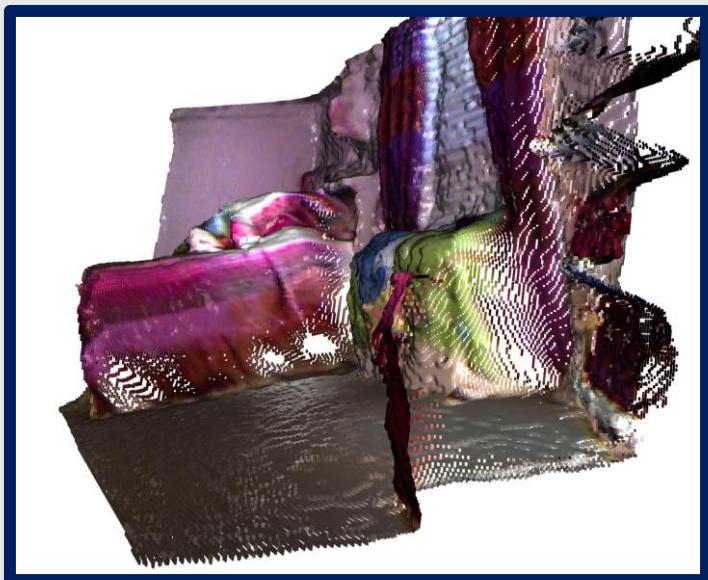
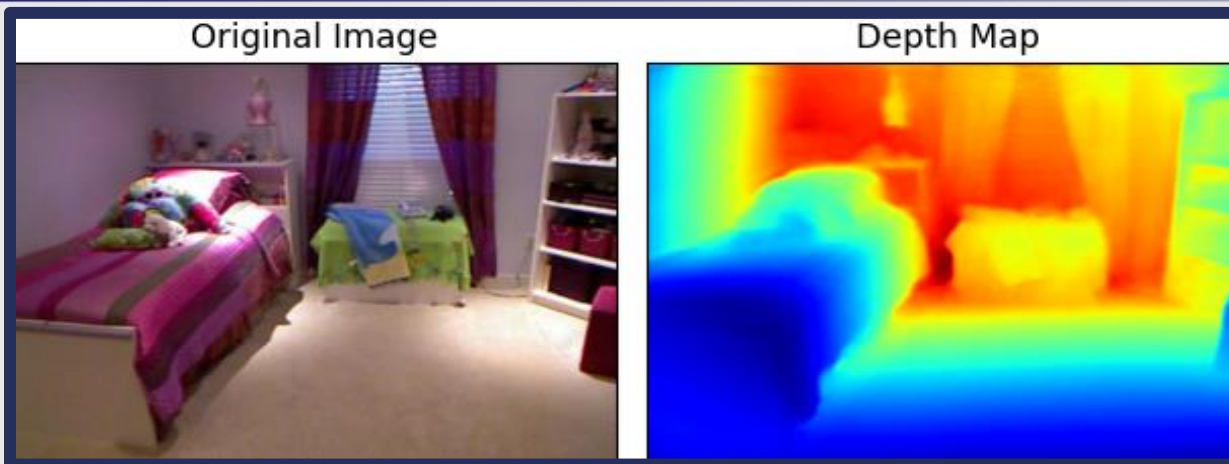
$$\nabla \nabla \chi = \Delta \chi = \nabla \vec{V}$$

- ✓ Уравнение Пуассона с неизвестной функцией  $\chi(p)$ :

$$\Delta \chi = \nabla \vec{V}$$

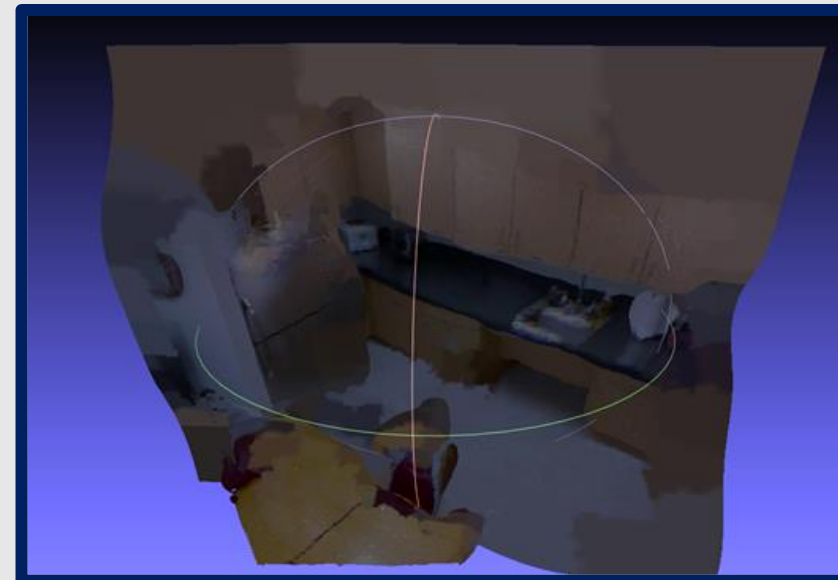
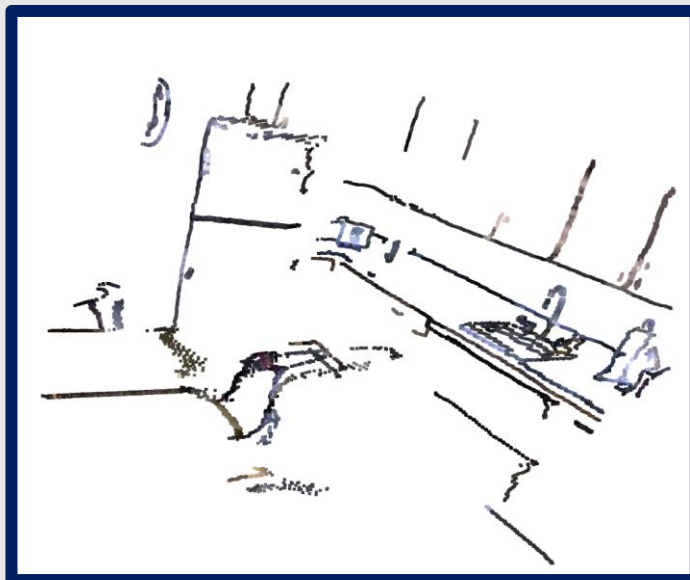
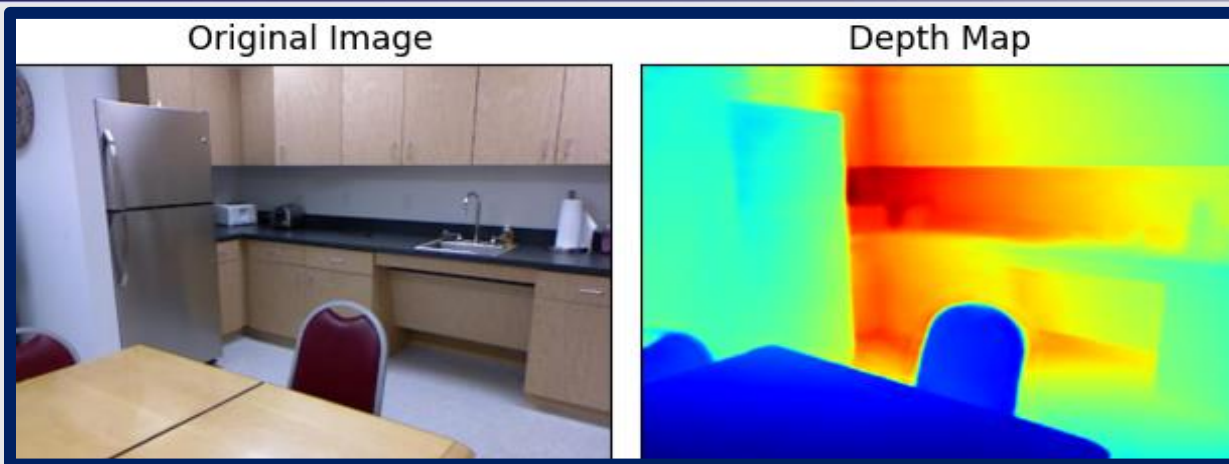


# Реконструкция помещения (результаты)





# Реконструкция помещения (результаты)



# Заключение

**В ходе выполнения данной работы были успешно реализованы и исследованы следующие методы и алгоритмы:**

## ✓ PlaneTracking:

- выполнена программная реализация для технологии отслеживания плоскостей в дополненной реальности (AR)

## ✓ Метод оптического потока.

### Метод стереоректификация + stereoBM:

- используются для отслеживания объектов и реконструкции помещения
- среди рассмотренных подходов метод оптического потока оказался наиболее эффективным

## ✓ DBSCAN.

### Классификация на основе геометрических характеристик:

- метод кластеризации используется для дальнейшего упрощения классификации отсканированных объектов

## ✓ Метод реконструкции поверхности Пуассона:

- применён для восстановления трёхмерных моделей, сохраненных в файл.
- модель реконструированной помещения в дальнейшем можно загрузить в любой 3D-движок





УНИВЕРСИТЕТ  
ЛОБАЧЕВСКОГО

**СПАСИБО ЗА ВНИМАНИЕ!**

3823М1ПМвм Розанов Д.И.