

## **Приветствие + PlaneTracking** *1мин 20с*

### **Слайд\_0**

Здравствуйте, уважаемая комиссия.

Вашему вниманию представлена работа на тему «Технологии VR/AR для 3D-реконструкции помещений: алгоритмы сканирования, обработки изображений, восстановления поверхностей на основе фотографических данных»

### **Слайд\_1**

В данной работе будут рассмотрены и применены некоторые методы для сканирования объектов с фотографий, методы из компьютерного зрения с целью реконструкции помещений на основе фотографических данных для последующего использования, например, в VR-средах.

Для достижения поставленной цели были реализованы программные решения для изученных тем, представленных на слайде, а также соответствующая для этого литература.

### **Слайд\_2**

PlaneTracking - это технология AR, которая позволяет отслеживать и визуализировать виртуальные объекты на плоских поверхностях. Для распознавания и отслеживания в режиме реально времени она использует камеру устройства

В рамках поставленной задачи требуется научиться определять тип поверхности: горизонтальные и вертикальные. Для этого разработана система, в которой при нажатии на горизонтальную плоскость появляется куб, а при нажатии на вертикальную плоскость – сфера.

Основная идея заключается в следующем:

- Из точки касания на экране испускается луч Raycast
- Определяется пересечённая плоскость и вычисляется её нормаль.
- Тип поверхности определяется в зависимости от ориентации нормали

### **Слайд\_3**

На слайде представлены результаты работы программы, и она работает корректно, правильно распознает тип плоскости, используя технологии дополненной реальности.

## Метод плотного потока (сканирование) *1мин 35с*

### Слайд\_4

Метод плотного потока — это алгоритм, который вычисляет поле векторов движения для каждого пикселя в последовательности изображений и предоставляет информацию о движении. Поэтому, имея две фотографии с некоторым шагом, мы можем использовать этот метод для реконструкции комнаты, сегментировав движущиеся объекты от статического фона.

Оптический поток между двумя последовательными кадрами рассчитывается с использованием алгоритма Фаренбака и на выходе мы имеем векторный поток, который представлен в виде горизонтальной и вертикальной компоненты скорости. Далее применяется метод для выделения границ объектов на изображении. И для пикселей на границе вычисляются их координаты в пространстве.

### Слайд\_5

Координаты в пространстве вычисляются методом триангуляции на основе положения камеры на первом кадре, камеры на 2ом кадре и самой точки, для которой вычисляется ее положение. Формулы приведены на слайде.

### Слайд\_6

Для решения задачи предложен метод на основе стереоректификации и алгоритме StereoBM:

1. Стереоректификация применяется для выравнивания пары изображений, чтобы облегчить процесс сопоставления пикселей на обоих изображениях
2. Далее к обработанным стереоизображениям применяется метод StereoBM.

StereoBM сравнивает яркость и цвет пикселей, разбивая изображения на небольшие блоки одинакового размера и находя соответствия между ними.

После нахождения соответствующего блока, алгоритм вычисляет сдвиг между соответствующими пикселями.

3. На выходе алгоритм StereoBM возвращает карту смещения или карту глубины для пары стереоизображений.

\*\*\* Карта глубины — это изображение, в котором каждому пикселю соответствует глубина или расстояние до объекта. Более темные пиксели соответствуют объектам, находящимся ближе.

### Слайд\_7

На слайде представлены результаты применения метода плотного потока к двум последовательным кадрам угла комнаты. В дальнейшем и будем применять метод “плотного потока”, т.к. метод, основанный на стереоректификации и StereoBM, показал искаженные изображения в неправильной перспективе, что ухудшило результаты.

### Слайд\_8

На данном слайде представлен результат применения метода плотного потока к двум последовательным кадрам уже реального изображения.

## DBSCAN + Классификация Змин 40с

### Слайд\_9

Для дальнейшей реконструкции помещения с имеющихся фотографий – необходимо определить распознанные объекты, а именно провести кластеризацию и классификацию.

Для кластеризации облака точек будем использовать алгоритм DBSCAN, а классификацию проведем на анализе пространственных характеристик.

Для начала разберем алгоритм кластеризации. На вход алгоритм принимает облако точек, полученные после применения метода плотного потока. А также параметр  $\epsilon$  – радиус  $\epsilon$  окрестности, число  $m$  – минимальное количество точек в кластере и симметричную функцию расстояния.

Алгоритм находит плотные скопления точек (кластеры), где плотность точек внутри значительно выше, и все они относятся к какому-то классу объектов.

Более того, в областях с шумом плотность точек ниже плотности любого из кластеров. Таким образом, алгоритм также выделяет кластеры с точками, которые являются шумом, полученным в результате сканирования

### Слайд\_10

На слайде представлен результат применения алгоритма DBSCAN в 2D. И на следующем слайде представлен результат применения алгоритма DBSCAN для трехмерного пространства.

### Слайд\_11

В качестве примера – на слайде представлен кластер куба и сферы. Как видно, алгоритм отработал корректно. Всего для данного примера 4 кластера: куб, сфера, пол и стена + потолок.

### Слайд\_12

Выполнив кластеризацию, разбив облако точек на подобласти, мы упростили процесс классификации распознанных объектов.

В рамках задачи предполагается, что известен набор объектов, которые могут находиться в помещении. Это позволяет проводить классификацию распознанных объектов на основе анализа их геометрических характеристик. Поэтому и воспользуемся таким методом:

1. Центр многоугольника вычисляется как среднее арифметическое координат всех точек кластера.
2. Далее мы вычисляем расстояния от каждой точки до центра
3. И строим график индекса точки из облака в зависимости от расстояния до центра
4. Для анализа формы многоугольника мы перешли к полярной системе координат, где для каждой точки кластера вычислили угол  $\alpha_i$  между осью  $OX$  и вектором к точке  $p_i$  относительно центра  $(\bar{x}, \bar{y})$ .
5. На графике зависимости  $d_i$  от  $\alpha_i$  вершины многоугольника проявляются локальные максимумы, что позволяет точно определить их количество и расположение
6. Для автоматического детектирования экстремумов применяется свертка, и теперь локальные экстремумы располагаются строго в верхней полуплоскости графика

## Слайд\_13

Далее представлены результаты применения описанного метода на идеальных примерах, когда облако точек описывает фигуру простых объектов: четырехугольник, пятиугольник и окружность.

Можем наблюдать на графиках соответствие по точкам: на каждом графике по 4 угловых точки и 8 точек, находящихся на сторонах фигуры. А на 2ом и 3ем графике 4 локальных экстремума, что соответствует угловым точкам четырехугольника.

## Слайд\_14

Далее пример пятиугольника, и все аналогично корректно работает. На последнем графике наблюдаем 5 лок экстремумов

## Слайд\_15

Для классификации окружности мы можем использовать 1ый и 2ой график, где видим, что расстояние везде постоянное.

Третий график показывает два локальных экстремума. Согласно методу, если их количество равно 1, 2 или превышает заданное  $k$ , фигура считается окружностью, а не  $k$ -угольником.

## Слайд\_16

Теперь рассмотрим пример для классификации, полученный после сканирования помещения в примере, рассмотренном ранее. Необходимо было определить, что определенный кластер – это кластер куба или сферы.

Для куба получены такие результаты: по графикам сложно точно определить количество углов или другие характеристики, позволяющие однозначно идентифицировать фигуру.

## Слайд\_17

Поэтому были применены методы для очистки облака точек. Для очистки облака точек использовались следующие методы:

1. Удаление соседних точек, находящихся на расстоянии меньше заданного  $\epsilon$ .
2. Фильтрация точек по отклонению от центра: удалены точки, расстояние которых до центра выходит за допустимый предел

И результаты получились значительно лучше. На графике 2-3 выделены точки, которые соответствуют угловым точкам, и они являются локальными экстремумами.

## Слайд\_18

Для сферы также применили очистку и по третьем графику можем определить, что это окружность, т.к. всего 2 экстремума

## Практическое применение 1мин 35с

### Слайд\_19

Следующий шаг выполненной работы — практическое применение: создание 3D-модели, сохранённой в файл, которую можно загрузить в любой 3D-движок. При этом на входе у нас лишь одно изображение.

Формирование 3D-модели выполняется с помощью алгоритма реконструкции поверхности – с помощью алгоритма Пуассона. Основные входные данные метода - облако точек и нормали к полигонам, из которых состоит 3D-модель

1. 3D-облако точек мы получим на основе карты глубины, используя предобученную модель GLPN. Хотя для этого при наличии двух последовательных кадров можно было бы использовать оптический поток
2. Далее по приведенным формулам мы вычисляем координаты точек из облака, основываясь на карте глубины.
3. Нормали к полигонам mesh-сетки вычисляются на основе  $k$  ближайших точек, используя метод главных компонент. Для каждой точки мы получаем на выходе нормаль

### Слайд\_20

Алгоритм Пуассона на вход принимает облако точек и вычисленные нормали.

1. Введем индикаторную функцию для каждой точки, которая возвращает 1, если точка находится внутри модели и 0, иначе.  
Эта функция изначально неизвестна, но с помощью этого метода мы ее можем найти, тем самым реконструировав поверхность
2. Индикаторная функция находится из уравнения Пуассона: оператор Лапласа, примененный к индикаторной функции равен дивергенции поле нормалей.
3. Само уравнение Пуассона следует из приведенной Леммы на слайде, доказательство которой описано в одной из работы из списка литературы.

### Слайд\_21

На данном слайде представлен результат применения метода реконструкции поверхности для спальни: облако точек, облако точек на контуре, и сама 3D модель

### Слайд\_22

На данном слайде результат применения метода для фотографии кухни. Как видно, метод отработал корректно, в результате чего мы получили отсканированную 3D модель помещения.

### Слайд\_23

\*\*\*конец\*\*\*