**Аннотация.**

Работа описывает способ определения положения объектов в пространстве по нескольким точкам и его реализацию.

**Оглавление.**

[Введение. Цели и задачи. 1](#_Toc158146114)

[Что такое ArUco маркеры. 3](#_Toc158146115)

[Способ определения расстояния от камеры до маркера. 4](#_Toc158146116)

[Принцип работы. 5](#_Toc158146117)

Алгоритм работы программы……………………………………………………………………………………………………………………….6

[Math. 7](#_Toc158146118)

[Написание кода. 8](#_Toc158146119)

[Итоги. 10](#_Toc158146120)

# 

# Введение. Цели и задачи.

В настоящее время наблюдается бурный рост технологий в сфере компьютерного зрения. Одно из направлений это: определение положения(координат) объекта в пространстве. Я решил создать свою систему, которая бы помогала машине находить своё положение.

***Целью моего проекта:*** Создать простую, доступную и надёжную систему для определения положения камеры на плоскости с использованием ArUco маркеров.

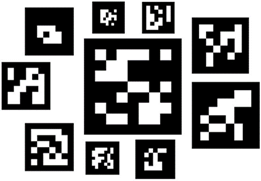
В ходе работы были поставлены следующие ***задачи***:

1. Разработка математического решение данной цели.
2. Реализация алгоритмической части.
3. Написание кода.
4. Тестирование.

# 

# Что такое ArUco маркеры.

ArUco маркеры это чаще всего квадрат с зашифрованным в нём числом, но так же с их помощью можно узнать расстояние до камеры.



Эти маркеры бывают разных форм размеров, но чаще всего они представляют собой чёрно-белые квадраты с помощью них, как я и написал выше можно не только узнать расстояние от камеры до маркера, но и определить угол его наклона по отношению к камере(впрочем как выяснится нам это не понадобитьься.

# Способ определения расстояния от камеры до маркера.

Мы видим этот маркер просто как чёрно-белый квадрат, но вот машина видит его как код несущий в себе много информации, например: если машина знает его реальные размеры и свойства матрицы своей камеры, то она легко может определить расстояние и угол наклона.

1.Способ определения расстояния

Мы знаем: размер маркера, угол обзора камеры.

Чтобы найти расстояние мы можем применить терему синусов

Теорема синусов гласит, что отношения сторон треугольника к синусам противолежащих углов равны и постоянны для всех треугольников. В данном случае, нам нужно знать одну сторону (размер объекта), один угол (угол обзора камеры) и противолежащий угол (угол между объективом и объектом).

Если мы знаем размер объектива (он обычно меньше, чем размер объекта), мы можем использовать теорему синусов, чтобы найти угол между объективом и объектом. Затем мы можем использовать тангенс этого угла, чтобы найти расстояние до объекта.

2. Способ определения угла поворота относительно камеры.

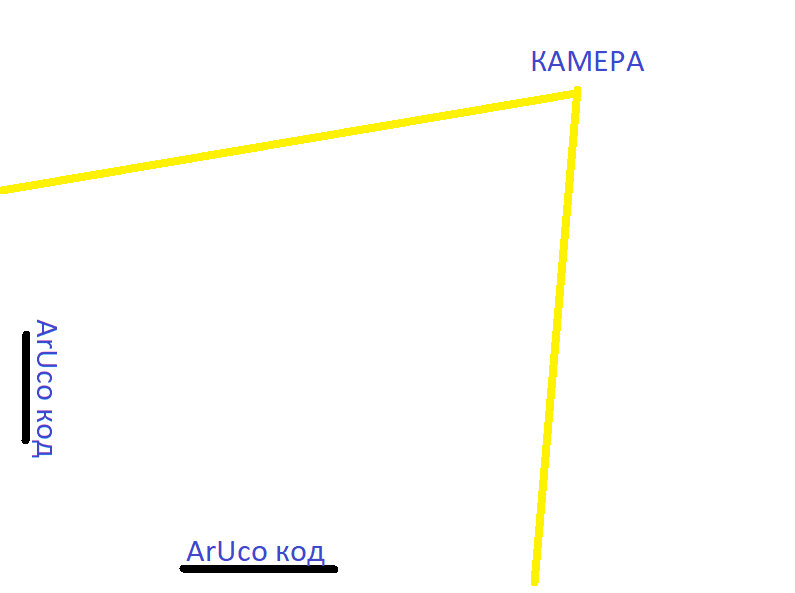
Мы знаем: размеры маркера(длины его сторон), расстояние от камеры до маркера.

Чтобы найти угол поворота мы можем применить пропорцию в которую будут входить нормальные размеры маркера, его нормальный угол поворота к камере(180), его мнимые размеры если смотреть через камеры, X(угол поворота, который нам нужно узнать.

# 

# Принцип работы.

Идея заключается в вычислении положения объекта с помощью закреплённой на нём камеры по двум маркерам.



Вид сверху (желтым обозначен угол обзора камеры)

ТК мы знаем точные позиции маркеров в пространстве мы строим 2 окружности с центрами в этих точках с радиусом равным расстоянию до камеры. Именно на точке пересечения этих двух окружностей и будет располагаться наш объект.

**Алгоритм работы программы.**

Задачи которые должна выполнять программа:

1.Определять маркеры.

2.Вычислять расстояние до них.

3.Вычислять точку пересечения построенных ранее окружностей.

4. Вычислять координаты камеры

Алгоритм работы:

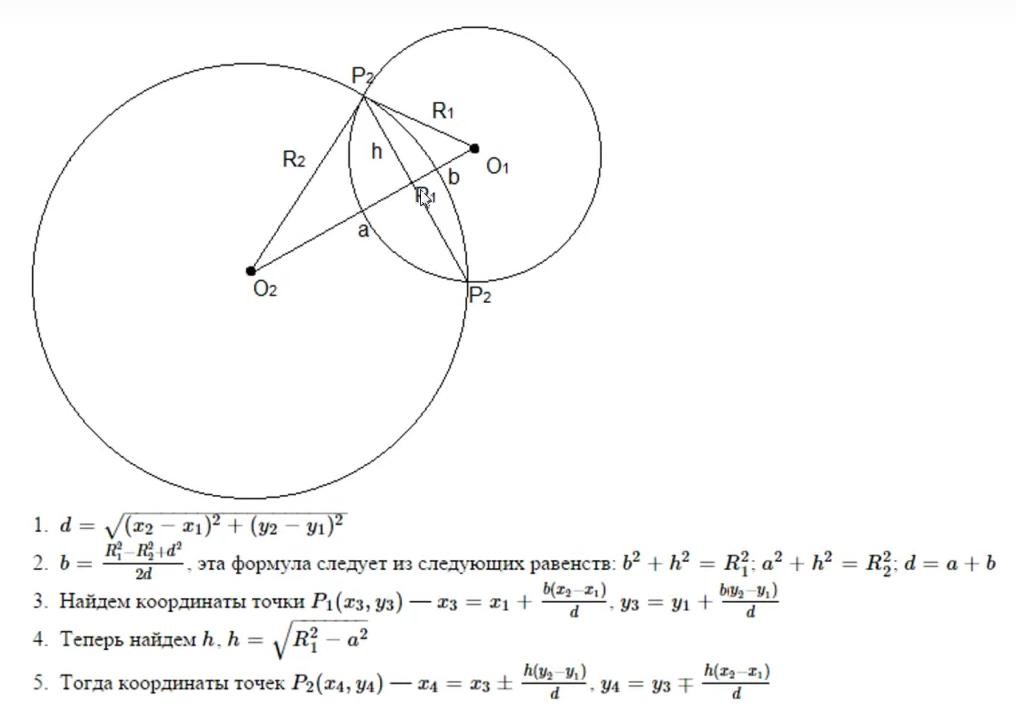
1. Задаём известные нам величины, а именно: размеры маркеров, коэффициент преломления матрицы(коэффициент рыбьего глаза), угол обзора камеры, положения маркеров в пространстве.
2. Определение маркеров в пространстве относительно камеры.
3. Нахождение расстояния до маркеров.
4. Визуально для пользователя строим две окружности в окне с помощью PyGame.
5. Находим координаты точек пересечения окружностей относительно точки (0;0) координатной плоскости.
6. Отбрасываем одну не нужную точку( её координаты будут меньшие)
7. Визуализируем эту точку в окне с помощью PyGame.

# Математическая модель.

В этом проекте львиную долю решает математика, а именно геометрия.

Первая математическая задача, которая стоит перед нами это определение расстояния до маркеров, мы её уже разобрали, поэтому пока пропускаем её.

Вторая задача: определить координат объекта

В принципе это все вычисления, но я думаю их следует немного пояснить.

В результате вычислений, точки пересечения окружностей P2, центрами которых являются ArUco маркеры, являются кандидатами для определения положения камеры. Т.к. в итоге вычислений мы получаем по две координаты x и y, возникает вопрос: «какие же координаты выбрать для этой точки?» Ответ: «с наибольшими значениями х и у». Т.к. если выбрать с наименьшими значениями координаты, то получится, что угол обзора камеры равен или больше, чем угол О2 Р2 О1, этого не может быть.

# Програмная реализация.

Я долго размышлял над выбором оптимального языка программирования на котором будет удобнее и эффективнее писать код.

Было несколько претендентов на эту роль:

1.Assembler.

2.Bash.

3.Python.

4.C++.

В итоге я выбрал Python, тк это лёгкий язык на котором присутствует огромное количество нужных мне модулей для этого проекта.

C++ я не выбрал, тк он не оптимизирован для работ с камерой в отличие от Python.

Assembler имеет хорошую скорость работы, лёгкость в освоении, но к сожалению на нём ужасно считать десятичные числа.

Bash был откинут, тк я планирую улучшать этот проект и выпускать его в массы, а этот язык нормально работает только на ядре linux.

В Python как я уже написал присутствует много нужных мне для этого проекта библиотек, а именно:

1.OpenCV – с помощью неё можно получать и обрабатывать видеопоток с камеры.

2.Math – встроенный модуль, который позволяет легко работать с числами(возводить в степень, брать коосинус, синус, логарифм).

3.time – с помощью неё можно оптимизировать работу программы, уменьшив количество кадров в секунду.

4.PyGame – с помощью неё мы сможем выводить картинку в отдельное окно, визуализируя происходящее.

**Тестирование.**

После тестирование были выявлены такие проблемы как:

1. Медленная скорость работы
2. Неточность данных
3. Периодические выводы неверных(битых) значения.

Как я их исправлял:

1. Уменьшил разрешение видеопотока, за счёт чего уменьшилось время на их обработку.
2. Разделил код на 3 основных файла: файл со всеми переменными, файл со всей математикой и файл в котором был алгоритм получения и обработки видеопотока.
3. Применил медианное усреднение к нужным нам файлам.

Так же для сбора данных я использовал библиотеку pandas с помощью которой начал автоматически помещать все данные в таблицу для их анализа.

# Итоги.

**Вывод:** Я создал,доступную и надёжную систему для определения положения камеры на плоскости с использованием ArUco маркеров.

Литература.

1. Бриггс Д. Python для детей. Самоучитель по программированию: МИФ, 2023

2. Борзунов С., Кургалин С. Алгебра и геометрия с примерами на Python. : Лань, 2021

3. Шакирьянов Э. Компьютерное зрение на Python: Лаборатория знаний, 2021