Министерство науки и высшего образования

Российской федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Рязанский государственный радиотехнический университет имени В. Ф. Уткина»

Кафедра АИТУ

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
 К КУРСОВОЙ РАБОТЕ**

**ПО ДИСЦИПЛИНЕ «Методы локализации, позиционирования и навигации»**

**(бакалавриат)**

на тему

**«Разработка системы навигации внутри помещения с использованием визуальных меток»**

Направление подготовки: *15.03.06 Мехатроника и робототехника*

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Лукша С. С.)

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Филин В.Г.)

Рязань 2025

**ВВЕДЕНИЕ**

Современные технологии робототехники стремительно развиваются, что приводит к увеличению спроса на автономные системы. Мобильные роботы широко используются в логистике, сельском хозяйстве, медицине, исследованиях и других отраслях. Одной из ключевых задач, стоящих перед такими роботами, является точное определение своей ориентации в пространстве для выполнения заданий.

Ориентация в пространстве — это базовая задача для любого мобильного робота, определяющая его способность перемещаться и выполнять задачи. Существующие решения, такие как инерциальные измерительные системы (IMU), GPS, магнитометры или камеры, имеют свои ограничения, включая зависимость от условий окружающей среды, электромагнитных помех, погрешностей при длительных измерениях. Один из таких способов является использование визуальных меток для определения местоположения робота.

Особенности использование визуальных меток:

1. Очень точно определяет местоположение в ограниченной области, где GPS акселерометры будут менее точны;
2. Визуальные метки не требуют больших затрат;
3. На визуальные метки не будут влиять условия окружающей среды если они будут располагаться внутри помещений
4. Простата в использовании

**Визуальные метки**

Визуальные метки - это навигационные элементы, которые помогают роботу определять своё местоположение. Под этими метками понимают QRкод, ArUco-маркеры. Такие метки позволяют роботу понимать свое местоположение в пространстве с помощью, а робот распознает эти метки с помощью камеры и машинного зрения.

Виды:

ArUco-маркеры — это специализированные двоичные квадратные метки, предназначенные для обнаружения и идентификации объектов в пространстве с помощью методов компьютерного зрения.

Каждый маркер состоит из широкой чёрной рамки и внутренней двоичной матрицы, которая кодирует его идентификатор (ID). Чёрная рамка облегчает быстрое обнаружение, а двоичная матрица обеспечивает уникальную идентификацию и исправление ошибок.

|  |
| --- |
|  |
| Рисунок 1.1 – Пример ArUco-маркера |

Aztec Code (Ацтек-код) — двумерный матричный штрих код, используемый для кодирования чисел, текста или двоичных данных. Aztec Code часто используется в приложениях, требующих кодирования большого количества данных, таких как транспорт и логистика, а также в маркетинге и продаже билетов. В ряде стран он используется и в правительственной документации.

|  |
| --- |
|  |
| Рисунок 1.2 – Пример Aztec-маркера |

MaxiCode (Макси-код) — это двумерный штрих-код, изобретённый в 1992 году для систем, принимающих грузы. MaxiCode широко используется в логистике и судоходстве. Он будет считываться, даже при повреждении этикетки, и при этом вероятность неправильного считывания будет крайне мала.

|  |
| --- |
|  |
| Рисунок 1.3 – Пример MaxiCode |

QR-код (от англ. Quick Response — «быстрый отклик») — это двумерный штрихкод, который состоит из чёрно-белых квадратных модулей, расположенных на белом фоне. Он легко считывается цифровым устройством и хранит данные в виде серии пикселей в квадратной сетке. QR-код может содержать текстовую и контактную информацию, URL-адреса, географические координаты и другие типы данных. Кроме того, такой код устойчив к повреждениям: он остаётся читаемым даже при потере до 30% информации.

|  |
| --- |
|  |
| Рисунок 1 – Пример QR-кода |

1. Поисковой узор
2. Сведения о формате
3. Полосы синхронизации
4. Информация о версии
5. Тихая зона
6. Выравнивающий узор

Каждый тег с QR-кодом представляет одну позицию на карте. Каждый тег с QR-кодом размещен в нужных местах. Ниже представлен пример положение каждого тега и схема демонстрационного моделирования.

|  |
| --- |
|  |
| Рисунок 2 – Схема расположения визуальных меток |

1. r1 - начальная позиция (номер 1).
2. qr2 находится напротив позиции qr1 (номер 2).
3. qr3 остается на позиции qr1 (номер 3).
4. qr4 находится справа от позиции qr1 (номер 4).

Навигация робота с использованием QR-кодов предполагает следующие подходы:

* Использование QR-кодов в качестве ориентиров. Информация о местоположении хранится в QR-кодах, которые стратегически размещены в рабочей среде. Робот, оснащённый промышленной камерой, указывающей на потолок, чтобы читать QR-коды с высокой скоростью. Позиция робота оценивается на основе позиционной связи между QR-кодами и роботом. 3
* Использование QR-кодов для навигации в помещении. Робот начинает движение и инициализирует сканирование с помощью камеры, направленной на потолок. С помощью шаблонов-искателей робот идентифицирует QR-код в кадре. После этого QR-код декодируется, и робот определяется текущее местоположение. Поскольку один из углов QR-кода не содержит шаблон-искатель, этот угол определяет ориентацию робота.

**Средства разработки системы навигации**

**Robot Operating System (ROS)** — это фреймворк для программирования роботов. ROS является «программным клеем», который дает возможность разработчикам сосредоточиться на своей конкретной задаче. Хотя ROS не является операционной системой, он предоставляет сервисы, такие как аппаратная абстракция, низкоуровневое управление устройствами, реализация часто используемых функций, передача сообщений между процессами и управление пакетами (плагинами).

Основные особенности ROS 2:

* Поддержка различных языков программирования. ROS 2 поддерживает несколько языков программирования, таких как C++, Python и другие, что упрощает разработку приложений для роботов. 1
* Улучшенная поддержка реального времени. ROS 2 предлагает улучшенные возможности реального времени, что позволяет разрабатывать робототехнические системы с высокой отзывчивостью.
* Расширенная масштабируемость. ROS 2 обладает расширенным механизмом коммуникации и поддерживает работу в распределённых системах, что делает его подходящим для крупных и сложных проектов.

**Matlab** — язык программирования, который входит в одноимённый набор программ, предназначенных для научных вычислений и машинного обучения. Язык Matlab относится к высокоуровневым, интерпретируемым языкам со слабой динамической типизацией. Он рассчитан на математические вычисления: поддерживает сотни видов математических функций, матричные структуры данных, векторы, многочлены, интерполяцию и дифференциальные уравнения.

Среда разработки Matlab позволяет сгенерировать код на языке C++, поддерживает языки программирования такие как C++ и Python, поддреживает библиотеку OpenCV которую поддерживает ROS2. Также Matlab позволяет работать совместно с ROS2 при помощи встроенного набора инструментов ROS Toolbox.

ROS Toolbox предоставляет интерфейс, соединяющий MATLAB и Simulink с операционной системой робота (ROS и ROS 2). С помощью набора инструментов можно спроектировать сеть узлов ROS и объединить узлы ROS, сгенерированные MATLAB или Simulink, с существующей сетью ROS.

|  |
| --- |
|  |
| Рисунок 3 – Схема взаимодействия MATLAB и ROS2 |

**Gazebo** — это динамический 3D симулятор с открытым исходным кодом, который развивается Open Source Robotic Foundation и довольно тесно взаимодействует с ROS. Gazebo позволяет точно и эффективно моделировать роботов как в сложных условиях помещений, так и снаружи.

Симулятор состоит из сервера gzserver, который занимается просчетом физики, столкновений и симуляцией сенсоров. К серверу могут подсоединяться клиенты, например, gzclient (для десктопа) и gzweb (для браузера). Именно они занимаются рендерингом моделей.

Все это дает возможность тестировать сложные робототехнические системы в виртуальном пространстве гораздо быстрее и без риска нанести ущерб дорогостоящим настоящим роботам.

Gazebo включен в полный установочный пакет ROS, поэтому дополнительно ничего устанавливать не нужно. Для headless конфигурации требуется gzweb.

**OpenCV**

Библиотека компьютерного зрения с открытым исходным кодом OpenCV (Open Source Computer Vision Library) – библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения. Реализована на языке C/C++, также разрабатывается для Python, Java, Ruby, Matlab, Lua и других языков.

Функции OpenCV:

* интерпретация изображений;
* калибровка камеры по эталону;
* устранение оптических искажений;
* определение сходства;
* анализ перемещения объекта;
* определение формы объекта и слежение за объектом;
* 3D-реконструкция;
* сегментация объекта;
* распознавание жестов.

**Метод отслеживания местоположения**

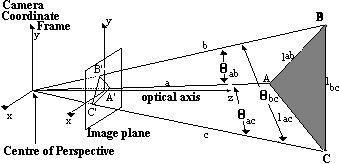
Оптические методы

Оптические методы представляют собой совокупность алгоритмов компьютерного зрения и отслеживающих устройств, в роли которых выступают камеры видимого или инфракрасного диапазона, стереокамеры и камеры глубины. В зависимости от выбора системы отсчёта выделяют два подхода для отслеживания положения:

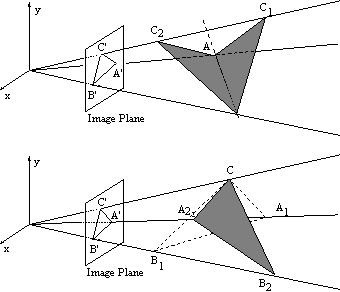
* Outside-in подход подразумевает присутствие неподвижного внешнего наблюдателя (камера), определяющего положение движущегося объекта по характерным точкам. Используется в Oculus Rift (Constrellation), PSVR, OSVR и множестве Motion Capture систем.
* Inside-out подход предполагает наличие на движущемся объекте оптического сенсора, благодаря которому возможно отслеживать движение относительно неподвижных точек в окружающем пространстве. Используется в Microsoft Hololens, Project Tango (SLAM), SteamVR Lighthouse (гибридный вариант, т.к. есть базовые станции).

Задача Perspective-n-Point (PnP)

При оптическом отслеживании для определения положения объекта в пространстве решается так называемая задача PnP (Perspective-n-Point), когда по перспективной проекции объекта на плоскость сенсора камеры необходимо определить положение объекта в 3D-пространстве.



Для заданной 3D-модели объекта и 2D-проекции объекта на плоскость камеры решается система уравнений. В результате чего получается множество возможных решений. Количество решений зависит от числа точек в 3D-модели объекта. Однозначное решение для определения 6-DoF положения объекта можно получить как минимум при 4 точках. Для треугольника получается от 2 до 4 возможных решений, то есть положение не может быть определено однозначно:



Решение предлагается достаточно большим количеством алгоритмов, реализованных в виде библиотек:

* POSIT
* [posest](http://www.ics.forth.gr/%7elourakis/posest/)
* [OpenCV](http://opencv.org/) ([solvePnP](http://docs.opencv.org/3.1.0/d9/d0c/group__calib3d.html" \l "ga549c2075fac14829ff4a58bc931c033d" \t "_blank))

**Список использованных источников**

1. Набор инструментов ROS - <https://www.mathworks.com/help/ros/index.html?s_tid=CRUX_lftnav>
2. OpenAI Gym+ROS+Gazebo: обучение автономного робота в домашних условиях. Часть 1 / Хабр - <https://habr.com/ru/articles/441218/>
3. Использование ROS 2 для взаимодействия с Gazebo — Gazebo ionic documentation - <https://gazebosim.org/docs/latest/ros2_integration/>
4. GitHub - georgealexakis/factory\_robot: Factory Robot - это реализация автономного навигационного робота с обнаружением QR-кода и визуальным обслуживанием. Реализация состоит из различных подходов к навигации. - <https://github.com/georgealexakis/factory_robot?ysclid=m53tfvb9zt535426629>
5. Обмен данными между matlab simulink и ros с применением пакета ros-toolbox - <https://earchive.tpu.ru/bitstream/11683/62234/1/conference_tpu-2020-C04_p225-227.pdf?ysclid=m53suhvo4w590021672>
6. QR-код: что это такое, как он работает и зачем нужен - <https://trends.rbc.ru/trends/industry/6189517c9a79475deb5dbf9a?ysclid=m53timzbns732036731>
7. Библиотеки образов компьютерного зрения (computer vision image library) - <https://shalaginov.com/2020/05/18/cv-library/>
8. Локализация по Aruco маркерам – <https://habr.com/ru/articles/482220/>
9. Обнаружение маркеров ArUco – <https://docs.opencv.org/3.4/d5/dae/tutorial_aruco_detection.html>
10. Обзор методов и технологий отслеживания положения для виртуальной реальности – <https://habr.com/ru/articles/397757/>
11. Обнаружение и декодирование QR-кода на изображении с помощью OpenCV – <https://lindevs.com/detect-and-decode-qr-code-in-image-using-opencv>