

УДК 378:004

DOI: 10.25686/978-5-8158-2474-4-2025-619-624

Разработка модульной системы технического зрения для соревновательного колесного робота

Д. В. Морохин, А. В. Иванов

Поволжский государственный технологический университет, Йошкар-Ола, Россия

Аннотация. В данной статье представлена разработка модульной системы технического зрения для колесного соревновательного робота, ориентированного на образовательные задачи. Основная цель исследования — создание доступной, но функциональной платформы, сочетающей невысокую стоимость с возможностями автономной навигации и обработки визуальной информации. Проведены комплексные испытания разработанной системы и сравнение ее с коммерческими аналогами. Результаты подтвердили конкурентоспособность решения при значительно меньшей стоимости. Особое внимание уделено модульности архитектуры, позволяющей легко модернизировать систему, включая интеграцию нейросетевых алгоритмов. Практическая значимость работы подтверждена успешным использованием платформы в образовательных соревнованиях.

Ключевые слова: техническое зрение, ROS 2, OpenCV, автономная навигация, машинное обучение, компьютерное зрение, одноплатный компьютер.

Modular vision system for a competitive wheeled robot development

D. V. Morokhin, A. V. Ivanov

Volga State University of Technology, Yoshkar-Ola, Russia

Abstract. This paper presents the development of a modular autonomous control and computer vision system for wheeled competition robots. The research aims to create an affordable yet functional platform combining autonomous navigation capabilities with real-time visual data processing for educational applications. The system architecture integrates an Orange Pi Zero 2 single-board computer and ESP-32 microcontroller, providing sufficient computational power for processing data from a Gxivision-SVGA camera and VL53L0X laser distance sensors. The software implementation utilizes Python and C++ with OpenCV libraries and ROS 2 framework for module interoperability. Comprehensive testing demonstrated line recognition accuracy up to 96 % under various lighting conditions. Comparative analysis with commercial solutions (DJI RoboMaster, VEX Robotics) confirmed the system's competitiveness at significantly lower cost (<\$300). The modular design facilitates future upgrades, including neural network algorithm integration. Practical significance was validated through successful deployment in educational robotics competitions. The solution offers an optimal balance of performance and affordability for STEM education applications.

Keywords: wheeled robot, computer vision, ROS 2, OpenCV, autonavigation, machine learning, microcomputer.

Введение

Актуальность данной работы обоснована растущим интересом к робототехнике как в научном, так и в образовательном контексте. Разработка алгоритмов управления для соревновательных роботов способствует развитию навыков программирования, проектирования и инженерного мышления у студентов и молодых специалистов [1]. Это, в свою очередь, стимулирует интерес к STEM-дисциплинам (наука, технологии, инженерия и математика) и формирует основу для будущих исследований и разработок в области робототехники.

Современная образовательная робототехника требует доступных платформ с функционалом автономной навигации и технического зрения. Анализ коммерческих решений выявил их ключевые ограничения: высокую стоимость (400-1000 тыс. руб.), сложность калибровки и недостаточную гибкость для массового образовательного применения [5, 9-11]. В работах [2, 4] предложены бюджетные альтернативы, однако их производительность недостаточна для обработки видеопотока в реальном времени.

Исследование, проведенное в данной работе, направлено на необходимость создания недорогой, модульной системы с открытой архитектурой, совместимой с Robot Operating System 2 и современными библиотеками компьютерного зрения [8].

Постановка задачи

Целью работы является разработка модульной системы для колесного робота, включающей:

- 1) модуль автономной навигации на базе стека Orange Pi Zero и ESP-32 с датчиками расстояния и освещения;
- 2) систему технического зрения на основе микрокомпьютера Orange Pi Zero 2;
- 3) ROS2 – совместимый программный стек.

Задачи:

- Разработка структуры аппаратного и программного обеспечения колесной платформы.
- Разработка механизма взаимодействия между головным вычислителем и микроконтроллером нижнего уровня через micro-ROS.
- Оптимизация алгоритмов обработки изображений для слабых вычислительных ресурсов.
- Сравнение эффективности с коммерческими аналогами.

Проектирование системы

1. Аппаратная архитектура

Система реализована по модульному принципу (рис. 1). Каждый отдельный блок связан с блоком управления, реализованным на микрокомпьютере Orange Pi Zero 2 [7]. Он выполняет функцию управления остальными блоками.



Рис. 1. Структурная схема робота

Данный микрокомпьютер сочетает в себе хорошую производительность и энергоэффективность, имея 4-ядерный процессор Cortex-A53 1.5 ГГц и графический процессор Mali G31 MP2 с поддержкой OpenGL. Наличие Ethernet, Wi-Fi, Bluetooth, USB-интерфейсов позволяет реализовать блок передачи данных без дополнительных внешних модулей. Поддерживает ОС Linux с большим количеством библиотек для компьютерного зрения. Установленный графический процессор и 1ГБ DDR3 оперативной памяти ограничивает возможности по сравнению с LPDDR4 в других современных решениях. Однако по соотношению цена-качество для задач обработки изображения и выполнения задач с моделями нейросети для распознавания объектов данная модель является оптимальным выбором.

Orange Pi Zero 2 поддерживает подключение USB-камер через стандартные UVC-драйверы, что упрощает интеграцию по сравнению с CSI-интерфейсами. Наличие 26-контактного GPIO обеспечивает прямую интеграцию с датчиками робота.

В числе рассматриваемых аналогов были такие модели как Raspberry Pi, OpenMV-H7, NVIDIA Jetson Nano, Repka Pi, ESP32-CAM. В сравнении с Orange Pi Zero 2 они имеют либо высокую производительность, но высокую стоимость, либо достаточно слабые характеристики для возможности использования обученных моделей нейросетей и распознавания. Одним из главных преимуществ Orange Pi Zero 2 является его гибкость и настраиваемость. Его можно легко адаптировать для работы с различными компонентами системы технического зрения, такими как камеры, сенсоры, устройства вывода и другие.

Блок распознавания является ключевым компонентом системы, отвечая за полный цикл обработки каждого изображения видеопотока. Модуль камеры Gxivision передает изображение в программное обеспечение блока управления для распознавания. Главным преимуществом данной камеры является поддержка стандарта UVC (USB Video Class), которая обеспечивает совместимость по принципу «подключи и работай» с Orange Pi без необходимости установки дополнительных драйверов.

Блок питания обеспечивает стабильное напряжение 5В всем элементам системы. В его роли выступает аккумулятор с понижающим напряжение модулем.

Для организации беспроводной коммуникации используется блок передачи данных, оснащенный Wi-Fi модулем. Этот компонент выполняет двусторонний обмен информацией между управляющими устройствами, обеспечивая эффективную передачу данных. Благодаря встроенным Wi-Fi модулям у Esp-32 и Orange Pi достаточно просто организовать их взаимодействие через фреймворк ROS 2. Команды с одноплатного компьютера будут передаваться на контроллер управления роботом, а данные с датчиков – в обратном направлении.

Для сбора данных с датчиков (лидар VL53L0X, лазерные дальномеры VL53L0X-V2, ИК-бампер FLASH-I2C) и управления моторами (ZK-5AD) используется микроконтроллер ESP-32.

2. Алгоритмы компьютерного зрения

Для тестирования системы была реализована задача следования колесного робота по контрастной линии. При её решении были использованы и исследованы следующие алгоритмы:

2.1. Пороговая бинаризация. Пример кода:

```
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
_, binary = cv2.threshold(gray, 100, 255, cv2.THRESH_BINARY_INV)
```

2.2. Контурный анализ с вычислением моментов. Пример кода:

```
M = cv2.moments(largest_contour)
cx = int(M["m10"])/M["m00"]
```

a. ПИД-регулятор для движения по линии. Пример кода:

$$u(t) = 0.8 \cdot e(t) + 0.05 \int e(t) dt + 0.2 \cdot de(t)/dt$$

3. Программная архитектура

Состав и взаимодействие между программными модулями показаны на рисунке 2.

Основными программными функциями одноплатного компьютера являются:

- распознавание объектов на изображении;
- взаимодействие с основной системой управления колесным роботом;
- возможность транслировать обработанное изображение с камеры.

Микроконтроллер выполняет свои задачи:

- получение данных от датчиков по протоколу I2C;
- управление коллекторными двигателями.

Также важной общей задачей является обмен данными и командами.

Программное обеспечение системы управления реализовано на Python для одноплатного компьютера. Основная задача программы — обнаружение и отслеживание тёмных объектов на светлом фоне (линия движения) в режиме реального времени с последующей передачей команд микроконтроллеру для управления колесным роботом. Тестирование направлено на проверку корректности работы алгоритма обработки видеопотока, выделения объектов и интеграции модуля с роботом.

Для обеспечения модульности не только в аппаратном, но и программном аспекте было принято решение об использовании Robot Operating System 2 – фреймворка для создания робототехнических систем [3]. Его основные особенности: работа в реальном времени, поддержка IoT-устройств, средства безопасности и интеграция с промышленными стандартами. Это не операционная система в классическом понимании, а набор инструментов, библиотек и соглашений, упрощающих разработку сложных робототехнических приложений.

ROS 2 помогает разделить функционал на независимые узлы (nodes), которые обмениваются данными через топики (topics) и сервисы (services). Полноценная версия ROS в проекте выполняется на микрокомпьютере OrangePi. Но для управления роботом требуется включить в обмен микроконтроллер ESP-32. На нём для функционирования полноценной ROS недостаточно ресурсов. Однако существует реализация Micro-ROS – адаптация ROS 2 для микроконтроллеров, позволяющая им участвовать в ROS-сети как полноценные узлы. Одноплатный компьютер выполняет обработку изображения с камеры (в данном случае определяет линии следования) и публикует команды управления в топики ROS 2. Micro-ROS Agent — промежуточное ПО, запущенное на Orange Pi, которое

преобразует сообщения ROS 2 в формат, понятный micro-ROS на ESP-32, и наоборот. ESP-32 подписывается на топик, получает команды и преобразует их в PWM-сигналы для управления моторами.

На рисунке 2 также показаны некоторые из «топиков» разработанной системы. Ниже опишем ещё один из пакетов системы, который реализует алгоритм движения по линии (ROS 2 пакета Line follower).

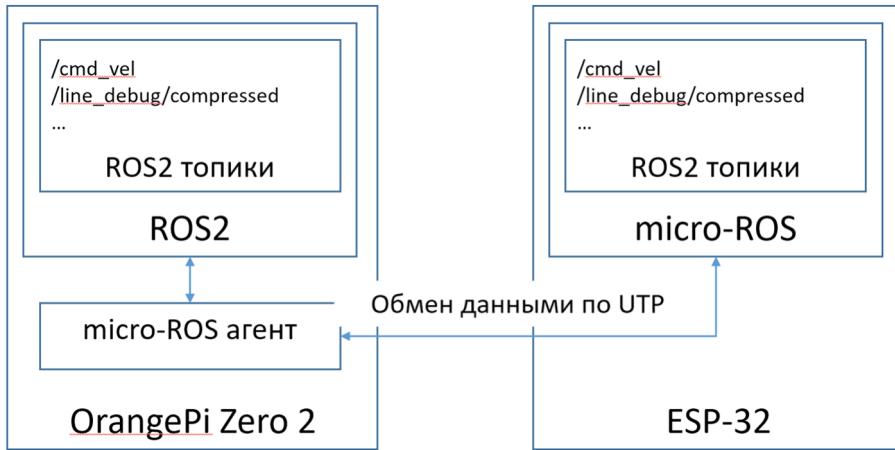


Рис. 2. Структура программного обеспечения системы

ROS 2 пакет состоит из двух python файлов cameraPublisher.py и subscriberImage.py.

В cameraPublisher.py происходит получение и обработка изображения с веб-камеры, а затем её отправка на отладочный топик linedebug/compressed, к которому подписывается subscriberImage.py.

Далее происходит расчет управления роботом, а именно определение того, в какой части кадра по горизонтали находится объект в виде линии.

Затем в работу включается программа micro ros на ESP32, она подписывается на twistmsg и получает значения переменных для управления моторами linear.x и angular.z:

- twist.linear.x – линейная скорость вдоль оси X (движется вперёд/назад);
 - twist.angular.z – угловая скорость вокруг оси Z (вращение по часовой стрелке/против часовой стрелки).

Значения параметров обычно указываются в метрах в секунду (м/с) для linear.x и радианах в секунду (рад/с) для angular.z. Для дифференциальных роботов эти два параметра являются основными для управления движением.

Результаты экспериментов

В результате работы реализована колесная платформа с возможностью расширения (рис. 3). С ней был проведен ряд экспериментов.

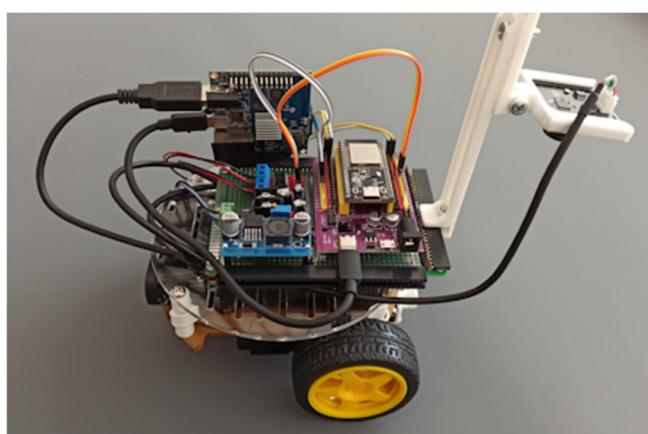


Рис. 3. Прототип колёсной платформы

Основной задачей экспериментов была оценка количественных и качественных характеристик работы системы и сравнение их с аналогами.

Первый эксперимент – оценка точности распознавания и задержки получения данных от алгоритма распознавания в зависимости от освещения (табл. 1). Здесь результаты ожидаемые: камера лучше и быстрее работает при хорошем освещении.

Таблица 1. Тесты технического зрения

Условия	Точность	Задержка
>500 люкс	96 %	110 мс
300-500 люкс	94 %	120 мс
<300 люкс	85 %	150 мс

Также производилось сравнение с аналогами. В их качестве использовались DJI RoboMaster EP от компании DJI, VEX Robotics, система технического зрения Robotiq Wrist Camera для роботов Universal Robots. Разработанная система значительно меньше (в 5-10 раз) стоит, имеет более низкое (в 3-4 раза) энергопотребление, однако показывает более низкую (в 3-4 раза) скорость обработки.

Кроме того, разработанная платформа прошла испытания на трассе. Здесь сравнение с аналогами не проводилось в связи с разными видами приводов, датчиков и контроллеров, что в большинстве случаев говорит о разных областях использования. Кроме того, результаты сильно зависят от качества алгоритмов распознавания и управления, а также их программной реализации.

Разработанная система показала следующие усредненные результаты: время движения по прямой линии (10 м) – 32 с, точность движения по линии сложной траектории – 91 %, среднее время прохождения лабиринта с максимальной длиной маршрута 10 м – 2 мин. 15 с.

Одним из важнейших результатов стало использование разработанной платформы при проведении Всероссийской студенческой олимпиады «Программирование контроллеров» в 2025 году [6]. Было собрано 15 комплектов платформы. Олимпиада продолжалась три дня, во время которых оборудование использовалось в достаточно жестких условиях. При этом не было зарегистрировано ни одного серьёзного отказа оборудования.

Обсуждение результатов

1. В качестве базовых ограничений разработанной системы можно отметить, что черно-белая камера не поддерживает цветовые маркеры, а также то, что достаточно большая задержка при распознавании линии ограничивает максимальную скорость движения.

2. Важнейшим преимуществом разработанной системы является более низкая стоимость по сравнению с аналогами. Заложенная модульность позволяет заменять компоненты и использовать новые программные обработчики основных событий.

3. В качестве перспективных путей развития проекта можно предложить интеграцию нейросети для детекции объектов и использование скоростной цветной камеры. Рассмотрим подробнее реализацию нейросети.

Orange Pi Zero 2 позволяет реализовать распознавание объектов, используя различные методы машинного обучения, например, сверточные нейронные сети (Convolutional Neural Networks, CNN). Они используются для классификации изображений и детектирования объектов на изображении. Для реализации этого подхода на Orange Pi Zero 2 используется установленный фреймворк TensorFlow Lite, который предоставляет оптимизированные модели глубокого обучения, специально созданные для работы на встраиваемых устройствах.

Для реализации технического зрения с распознаванием на колесном роботе можно сформировать свою модель машинного обучения или применять уже существующие и проверенные модели.

Наиболее популярные модели и их применимость к платформе.

- EfficientDet Lite (например, efficientdet_lite0.tflite):
 - легковесная модель, оптимизированная для работы на устройствах с ограниченными ресурсами;
 - поддерживается TensorFlow Lite и хорошо интегрируется с OpenCV;

- позволяет распознавать несколько классов объектов с приемлемой точностью.
- MobileNet SSD (Single Shot Detector):
 - быстрая и компактная модель для распознавания объектов;
 - часто применяется на одноплатных компьютерах и мобильных устройствах;
 - может работать в режиме реального времени при разрешении 300×300 или 640×480;
 - поддерживается TensorFlow Lite.
- YOLOv5n (Nano) и Tiny YOLO:
 - упрощённые версии YOLO, оптимизированные для маломощных устройств;
 - позволяют достичь порядка 10-20 FPS;
 - требуют оптимизации и квантования моделей для повышения производительности.

На данный момент наибольшие перспективы ожидаются от применения YOLOv5n.

Выводы

Разработана система технического зрения для применения в колесных робототехнических платформах, которая может быть использована в образовательных целях и для проведения соревнований. Точность распознавания линии составила примерно 96 %. Важной характеристикой подобных систем является стоимость. Удалось достичь значения этого показателя около 25 тыс. руб.

На основании проведенных испытаний доказана эффективность связки микрокомпьютера Orange Pi Zero 2 и микроконтроллера ESP-32 для решения образовательных задач. В области программной части системы предложенный стек ROS 2 и micro-ROS обеспечил стабильное взаимодействие модулей.

Список литературы

1. Джозеф Лентин. Изучение робототехники с использованием Python. СПБ.: Просвещение, 2019. 250 с.
2. Малогабаритный колесный автономный робот (МКАР). URL: <https://spbpu.com/proekty/malogabaritnyj-kolesnyj-avtonomnyj-robot-mkar>
3. Основные понятия о ROS и ее компонентах. URL: <https://cyberleninka.ru/article/n/osnovnye-ponyatiya-o-robot-operating-system-i-ee-komponentah/viewer>
4. Робототехническая платформа Omegabot: Education. URL: <https://robotbaza.ru/product/robototekhnicheskaya-platforma-omegabot-education>
5. Техническое зрение FANUC iRVision. URL: <http://www.robotsvarka.ru/tehnicheskoe-zrenie-fanuc-irvision>
6. VII ВСЕРОССИЙСКАЯ СТУДЕНЧЕСКАЯ ОЛИМПИАДА «Программирование контроллеров». URL: <https://robots.volgatech.net/>
7. Orange Pi Zero – подключение и настройка. URL: <https://istarik.ru/blog/orangepi/82.html>.
8. OpenCV. URL: <https://ru.wikipedia.org/wiki/OpenCV>
9. DJI RoboMaster EP. URL: <https://dji-rus.ru/robomaster-ep/robomaster-ep/>
10. VEX Robotics. URL: <https://educube.ru/products/robototekhnicheskiy-komplekt-na-baze-vex-iq-startovyy-s-tehnicheskim-zreniem/>
11. Wrist Camera - Robotiq. URL: <https://robotiq.com/products/wrist-camera>