*Ледерер П.А., Кочубей Д.С****Научный руководитель: Волошин А.В.,***Южный федеральный университетИКТИБ ЮФУ, г. Таганрог

**Разработка алгоритма управления автономным мобильным роботом на основе цветовой сегментации**

*В статье представлены результаты разработки мобильной автономной роботизированной платформы, созданной на базе микрокомпьютера RaspberryPI, разобран метод ориентирования робота с помощью цветовой сегментации. Управляющая программа написана на языке Python 3.8, для решения задач распознавания цветов используется библиотека компьютерного зрения OpenCV, ядром системы является фреймворк ROS.*

МОБИЛЬНЫЙ РОБОТ, КОМПЬЮТЕРНОЕ ЗРЕНИЕ, ROS, RASPBERRY PI, АВТОНОМНАЯ РОБОТОТЕХНИКА, ЦВЕТОВАЯ СЕГМЕНТАЦИЯ.

Создание мобильных роботов одно из наиболее востребованных и перспективных направлений в технике. Подобные робототехнические системы находят применение во многих сферах жизни человека. С каждым годом всё больше внимания уделяется автономным комплексам, способным функционировать самостоятельно, без вмешательства оператора. Для решения подобных задач робототехническая система должна быть оборудована массой датчиков и систем мониторинга, таких как камеры, ультразвуковые и лазерные дальномеры, стереокамеры и т. д. Помимо этого, необходимо так же снабдить робота средствами синхронизации и анализа приходящих с датчиков данных для последующей выдачи управляющих сигналов.

Проект реализован на базе микрокомпьютера RaspberryPI 4, имеющего достаточно ресурсов для разработки при низкой потребляемой мощности и удовлетворительном быстродействии для поставленной задачи. Микрокомпьютер занимается опросом датчиков, получением с них необходимой информации и обеспечивает связь между ними используя фреймворк ROS.

Основной системой робота, обеспечивающей его автономную работу, является система компьютерного зрения. На рисунке 1 представлена блок схема, описывающая алгоритм работы системы.

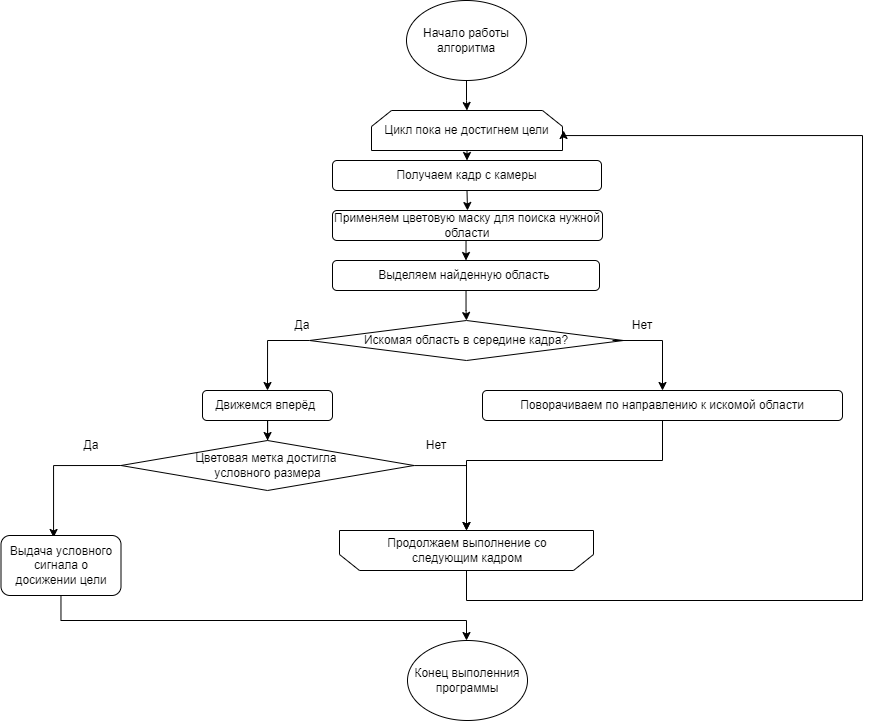


Рисунок 1 – Кадр с выделенной цветовой меткой

Более подробная последовательность действий алгоритма:

1. Система получает условный сигнал с указанием на какую цветовую метку реагировать.
2. Средствами библиотеки OpenCV робот получает кадр с камеры.
3. Кадр переводится из формата RGB (Red, Greem, Blue) в более удобный для дальнейшей обработки формат HSV (Hue, Saturation, Value).
4. С помощью заранее подготовленных цветовых масок на кадре возможно выделить область с меткой.
5. Зная разрешение полученного с камеры кадра возможно посчитать насколько от центра кадра сдвинута искомая область
6. В случае, если метка находится правее или левее центра изображения система выдаёт команду на двигатели, после чего робот начинает поворачивать в соответствующую сторону.
7. Поворот происходит до тех пор, пока искомая цветовая метка не окажется прямо перед роботом.
8. Происходит вычисление площади цветовой метки, если она оказывается больше, чем условная (полученная экспериментальным путём), алгоритм завершает работу.

На рисунках 2 и 3 представлены примеры кадров, полученных с бортовой камеры робота, до и после применения фильтров и выделения искомой зоны. Красным на фото подписаны координаты вершин прямоугольной метки, требующиеся для определение её площади.

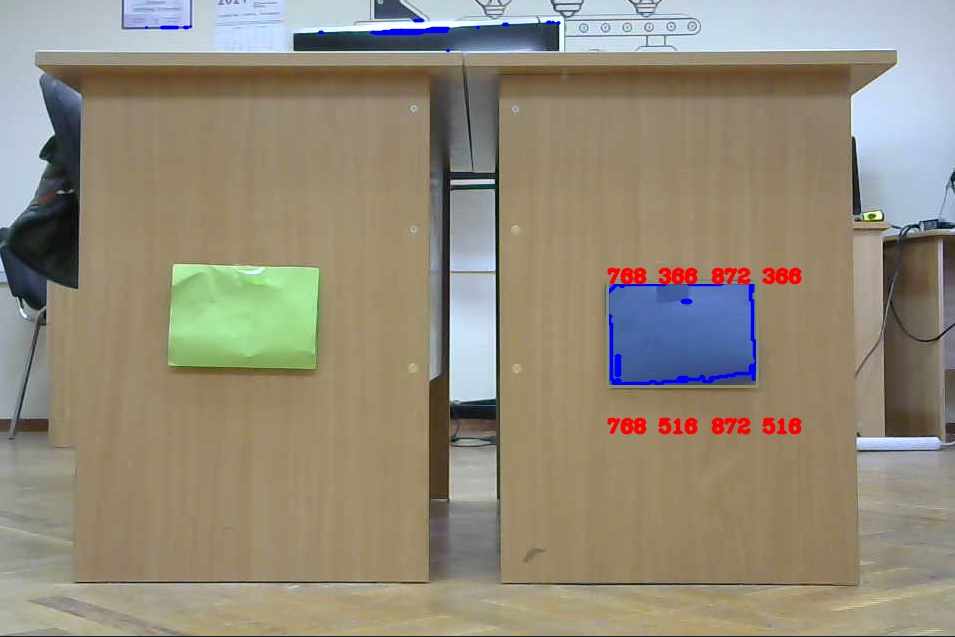


Рисунок 2 – Кадр с выделенной цветовой меткой

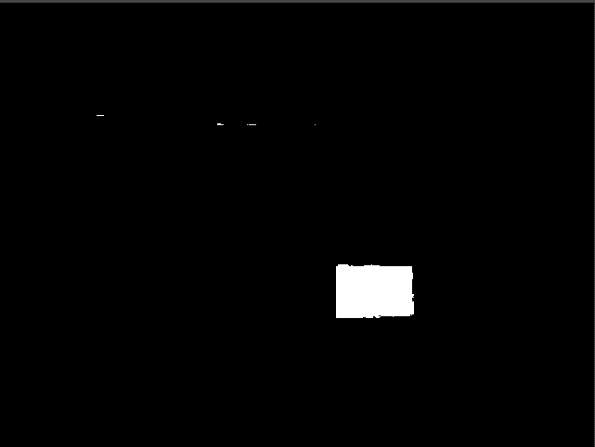


Рисунок 3 – Результат наложения цветовой маски на кадр

Стоит отметить, что в случае, если перед роботом резко возникнет препятствие такого же цвета, как и искомая метка, то происходит сверка данных с ультразвукового датчика расстояния расположенного в нижней части шасси робота и данных с предыдущего кадра. В случае если изменения произошли менее чем за 1 секунду (~5 кадров), они признаются “внезапными”, и робот не рассматривает их как конечный пункт назначения.  
  
 У данного подхода большое количество проблем, которые могут быть частично решены комбинированием алгоритмов компьютерного зрения со свёрточными нейронными сетями. Подобное решение позволит проводить более комплексный анализ окружения и увеличит адаптивность робота. Однако для обработки такого потока данных уже недостаточно мощностей Raspberry Pi 4, требуется использовать специализированные микрокомпьютеры, такие как Nvidia Jetson Nano или Nvidia Jetson Xavier. Так же в качестве дальнейшего развития возможно сменить обыкновенные камеры на стереокамеры для реализации SLAM (simultaneous localization and mapping) — метода одновременной локализации и построение карты использующегося во множестве передовых робототехнических систем.

Над списком литературы надо ещё подумать, надо всё-таки сослаться на что-то более менее по теме и побольше, я не знаю как это лучше сделать, надо подумать будет, думаю будем дальше читать и перечитывать статьи и что-то найдём.

1. Context - dependent color segmentation for Aibo robots / A. Palma-Amestoy Rodrigo [и др.] // IEEE 3rd Latin American Robotics Symposium. – 2006. – С. 128-136

2. Vision Spectra [Электронный ресурс]. – Режим доступа: https://www.photonics.com/a58196/The\_Role\_of\_Machine\_Vision\_in\_the\_Autom. – Дата доступа: 2021.

3. Популярная механика [Электронный ресурс]. – Режим доступа: https://www.popmech.ru/technologies/238704-glazami-robota-chto-takoe-mashinnoe-zrenie/. – Дата доступа: 2021.