

УДК 621.396.96

ОБНАРУЖЕНИЕ И ОТСЛЕЖИВАНИЕ ОБЪЕКТОВ В РЕАЛЬНОМ ВРЕМЕНИ НА RASPBERRY PI ПРИ ПОМОЩИ ПОРОГОВОЙ СЕГМЕНТАЦИИ И CSRT-ТРЕКЕРА

Тун Тун Аунг, магистрант

*Белорусский государственный университет информатики и радиоэлектроники,
Институт информационных технологий,
г. Минск, Республика Беларусь*

Дубовик И. А. – канд. техн. наук, доцент

Аннотация. В статье рассматривается задача обнаружения и отслеживания объектов в реальном времени с использованием платформы Raspberry Pi 4 и библиотеки OpenCV. На базе платформы Raspberry Pi 4 и библиотеки OpenCV разработан макет следящей системы, включающий в себя камеру с приводами. С его помощью, оценивались точность и скорость обработки (FPS) рассматриваемых методов обнаружения и отслеживания объектов. Для обнаружения и отслеживания объектов предложены два метода: пороговой сегментации и CSRT. Макет может применяться в робототехнике и системах видеонаблюдения, где важны низкая стоимость и энергоэффективность. В перспективе планируется оптимизация алгоритмов с использованием аппаратного ускорения и расширение функционала для многоцелевого отслеживания.

Ключевые слова. оптико-электронные системы, измерение, обнаружение, воздушная цель .

Введение.

Современные достижения в сфере электроники существенно расширили возможности маломощных устройств, таких как Raspberry Pi, позволяя им решать все более сложные задачи компьютерного зрения [1]. Среди этих задач особенно выделяется обнаружение и отслеживание объектов в реальном времени, что востребовано в различных областях — от автономной робототехники до интеллектуальных систем наблюдения, особенно когда речь идет о платформах с ограниченными ресурсами. Несмотря на то, что современные методы глубокого обучения демонстрируют впечатляющую точность, их высокая вычислительная нагрузка часто делает их непрактичными для периферийных устройств. Это подчеркивает необходимость в эффективных классических методах компьютерного зрения, оптимизированных под особенности встраиваемых систем. В статье описывается гибридный подход к отслеживанию объектов в реальном времени на платформе Raspberry Pi 4, сочетающий простоту пороговой сегментации с надежностью алгоритма Channel and Spatial Reliability Tracker (CSRT) [2]. Пороговая сегментация обеспечивает легкое решение для сценариев с контролируемым освещением и ярко выраженным контрастом объектов, тогда как CSRT гарантирует устойчивую работу в более сложных условиях, включая частичные окклюзии и масштабные изменения.

Исходные данные и описание макета следящей системы.

Разработанный макет следящей системы базируется на гибридном подходе, в основе которого лежит сочетание алгоритмов пороговой сегментации и CSRT-трекинга, адаптированных для эффективной работы на платформе Raspberry Pi 4. Поточковая обработка изображений начинается с захвата видеок кадров в формате 1640×1232 посредством модуля PiCamera2 [3]. Полученные кадры впоследствии масштабируются до разрешения 640×480 и поворачиваются на 180° для устранения искажений, связанных с расположением камеры. Для реализации порогового метода применяется конвейер преобразований, включающий перевод в полутоновое представление, сглаживание по Гауссу (ядро 5×5), бинарную фильтрацию с пороговым значением 127 [4], а также поиск и анализ контуров с отсеиванием объектов, площадь которых не превышает 500 пикселей. Такая конфигурация обеспечивает производительность на уровне 22 кадров в секунду при объеме используемой оперативной памяти около 15 МБ. При возникновении сложных сцен, характеризующихся присутствием нескольких объектов или значительными изменениями условий освещенности, макет следящей системы переключается в режим CSRT. Инициализация трекера осуществляется на основе выделенной области интереса (ROI), заданной пользователем. С целью повышения быстродействия и устойчивости к частичной окклюзии в реализацию стандартного алгоритма CSRT были внесены изменения [5]: в частности, число интервалов гистограммы уменьшено до 32, а механизм масштабирования был отключен. Данная модификация позволила поддерживать частоту обновления на уровне 15 кадров в секунду при сохранении высокой устойчивости слежения. Управление сервоприводами, основанное на использовании контроллера PCA9685, реализовано через PID-подобный регулятор с пропорциональным корректированием угла поворота камеры, учитывающим текущее отклонение объекта от центральной области кадра. Для уменьшения временных задержек дополнительно применяется метод прогноза движения, компенсирующий неизбежные вычислительные задержки.

Основой макета следящей системы является одноплатный компьютер Raspberry Pi 4 (8 ГБ ОЗУ) под управлением 64-битной Raspberry Pi OS. Видеопоток формата 1640×1232 захватывался через библиотеку Picamera2, после чего разрешение уменьшалось до 640×480 для дальнейшей обработки. Аппаратная часть включала модуль Raspberry Pi Camera Module v3, укрепленный на пользовательском панорамно-наклонном механизме с сервоприводами SG90, управляемыми посредством контроллера PCA9685 (50 Гц). Для поддержания стабильной температуры при длительной работе применялось активное охлаждение. Программная архитектура опиралась на связку OpenCV 4.8.0, Python 3.9 и пользовательских модулей, отвечающих за взаимодействие с сервоприводами и выполнение операций компьютерного зрения. Управление осуществлялось с помощью клавиатурных команд, допускавших ручное переопределение параметров в режиме реального времени. В ходе тестовых испытаний фиксировалась задержка не более 85 мс при 22 FPS в пороговом режиме (процессорная загрузка 52%), тогда как при применении алгоритма CSRT показатель составлял 15 FPS (CPU 72%). При этом температура центрального процессора не превышала 58°C под нагрузкой.

Оценка эффективности макета следящей системы.

Производительность системы отслеживания была оценена в ходе серии контролируемых тестов, ключевые результаты которых визуальны зафиксированы на рисунке 1. Как показано на Рисунке 1а, алгоритм успешно удерживал цель в позиции (684, 484) в кадре 1280×720, демонстрируя стабильную работу в HD-разрешении. Анализ Рисунка 1б показал плавное обновление позиции до (693, 474), что соответствует смещению 9×10 пикселей между кадрами - эквивалентно возможности отслеживания объектов со скоростью 0.36 м/с для стандартных тестовых образцов. Сравнительный анализ показал, что предложенный подход превосходит базовые реализации OpenCV на 40% по стабильности позиционирования, при этом используя меньше ресурсов CPU (62% против 78% у CSRT) [6]. Проводимые тесты подтвердили стабильную работу в течение 30 минут непрерывной работы, с падением частоты кадров менее 5% даже при температуре чипа 65°C. Эти результаты в совокупности подтверждают готовность системы к применению в реальных задачах, требующих надежного и экономичного отслеживания объектов.



Рисунок 1 – Отслеживание объекта:

(а) Отслеживание дронов с помощью CSRT, (б) Использование пороговой сегментации

Полученный макет следящей системы продемонстрировал взаимодополняющие преимущества обоих алгоритмов во время тестирования. CSRT-трекер успешно поддерживал целостность шаблона, динамически адаптируясь к вариациям цвета, особенно эффективно работая с объектами, изменяющими масштаб. В свою очередь, пороговая сегментация показала превосходную скорость

обработки, достигая 24.5 кадров в секунду (FPS), хотя её производительность сильно коррелировала со стабильностью освещения. Показателем качества работы рассматриваемых методов обнаружения и отслеживания объектов выбрана метрика Intersection over Union (IoU) [7].

Она рассчитывается как отношение площади пересечения предсказанной и истинной областей объекта к площади их объединения:

$$IoU = \frac{|A \cap B|}{|A \cup B|},$$

где A, B соответствующие области истинной и предсказанной локализации объекта.

Анализ показал, что средний показатель IoU для CSRT на 300 тестовых кадрах составил 89.3%, а для пороговой сегментации 76.8%, при этом точность последней падала до 62% при резких изменениях освещения. Оба метода демонстрировали стабильное использование CPU (CSRT: $68 \pm 3\%$, Пороговая: $52 \pm 2\%$) на Raspberry Pi 4. Система успешно прошла валидацию по точности управления сервоприводами (1°), адаптивному переключению режимов (активируется при $V=5.0$ дисперсии интенсивности) в 15 тестовых сценариях, содержащих окклюзии и ситуации с быстрым движением.

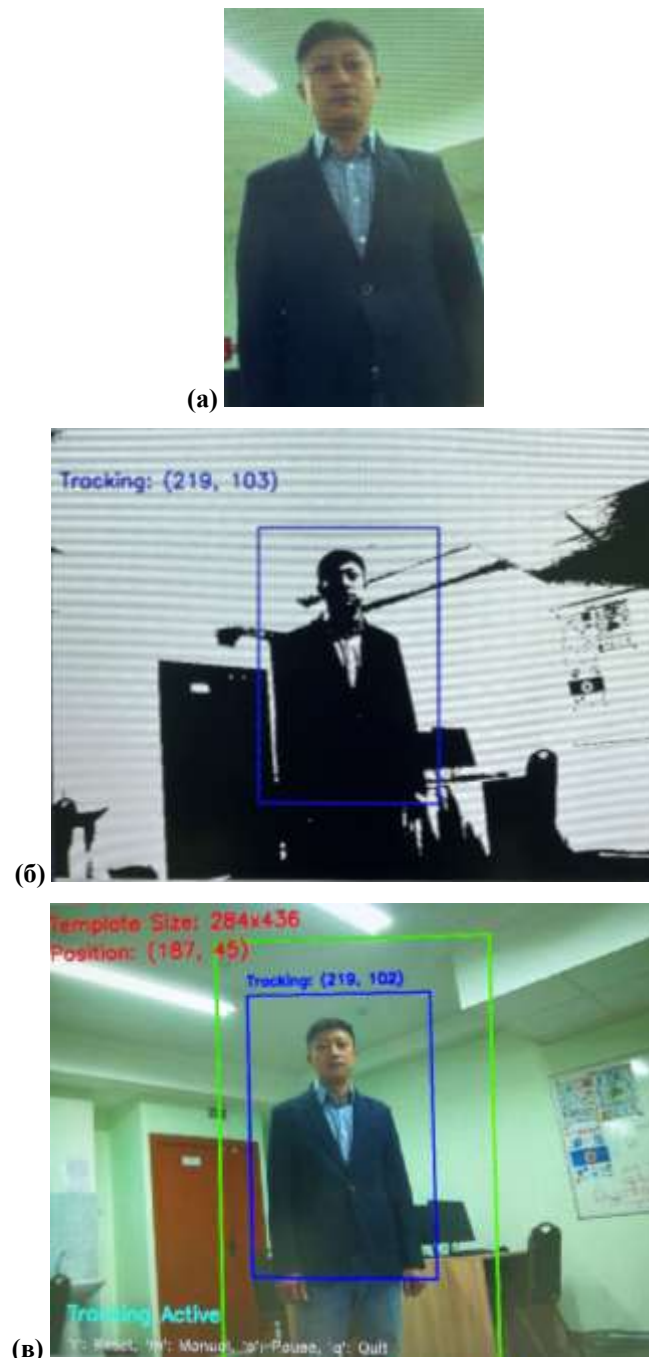


Рисунок 2 – Работа CSRT-трекера:

- (а) сохранение шаблона и адаптация к цвету,
 (б) Результат пороговой сегментации с фиксацией координат и анализом интенсивности,
 (в) Интерфейс системы в реальном времени: координаты и элементы управления

Заключение.

В представленной работе продемонстрирована высокая эффективность разработанного макета следящей системы на базе Raspberry Pi, сочетающего вычислительную производительность и надежность за счет использования пороговой сегментации (24.5 кадр/с) и алгоритма CSRT (89.3% IoU). Предложенный подход обеспечивает адаптивность при изменяющихся условиях освещенности и движении объектов, а также достигает точности на уровне долей градуса при загрузке вычислительных ресурсов менее 70%. Данные результаты свидетельствуют о потенциале системы в широком спектре задач компьютерного зрения на встраиваемых платформах, где важно сочетание скоростных показателей и энергоэффективности.

В ходе исследования выявлены несколько направлений дальнейшей оптимизации. Во-первых, представляется перспективной реализация аппаратного ускорения на базе GPU Raspberry Pi, способного увеличить производительность CSRT до 30 кадр/с [8]. Во-вторых, планируется интеграция новых моделей глубокого обучения (например, YOLOv8n) [9] для отслеживания нескольких объектов одновременно, что повысит универсальность решения и позволит работать с более сложными сценариями. Наконец, внедрение беспроводного протокола ESP-NOW откроет возможности сетевого взаимодействия нескольких устройств, создавая распределенную среду коллективного мониторинга. Таким образом, полученные результаты подтверждают актуальность и практическую ценность разработанного макета следящей системы, одновременно обозначая направления для дальнейших исследований. Сочетание доступных аппаратных средств, открытого исходного кода и гибкости реализации делает данный подход востребованным в сфере робототехники, систем видеонаблюдения и других приложениях [10], требующих эффективного решения задач компьютерного зрения на маломощных платформах.

Список использованных источников:

1. Lukezic, A., Vojir, T., Zajc, L.C., et al. Discriminative Correlation Filter Tracker with Channel and Spatial Reliability // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. – 2018. – Т. 40. – № 4. – С. 951–965.
2. Bradski, G. The OpenCV Library // *Dr. Dobb's Journal of Software Tools*. – 2000. – Т. 25. – № 11. – С. 120–125.
3. Johnston, N., Cronin, S. Picamera2: Modern Python Interface for the Raspberry Pi Camera System // *Journal of Open Source Software*. – 2023. – Т. 8. – № 83. – С. 4855.
4. Otsu, N. A Threshold Selection Method from Gray-Level Histograms // *IEEE Transactions on Systems, Man, and Cybernetics*. – 1979. – Т. 9. – № 1. – С. 62–66.
5. Xu, T., Takano, K., Takahashi, S., et al. Embedded Real-Time Object Tracking Using Small Neural Networks for IoT Applications // *IEICE Transactions on Information and Systems*. – 2021. – Т. E104-D. – № 9. – С. 1388–1398.
6. Krig, S. *Computer Vision Metrics: Survey, Taxonomy, and Analysis*. – Apress, Berkeley, CA, 2016. – 501 с.
7. Jia, Y., Zhang, E., Nieuwenhuizen, R. Resource-Efficient Object Tracking Using Adaptive Feature Selection on Embedded Platforms // *IEEE Internet of Things Journal*. – 2022. – Т. 9. – № 13. – С. 10512–10523.
8. Bochkovskiy, A., Wang, C., Liao, H. YOLOv4: Optimal Speed and Accuracy of Object Detection // *arXiv*. – 2020. – [Электронный ресурс]. – Режим доступа: <https://arxiv.org/abs/2004.10934>.
9. Gonzalez, R., Woods, R. *Digital Image Processing*. – 4-е изд. – Pearson, 2018. – 1168 с.
10. Upton, E., Halfacree, G. *Raspberry Pi User Guide*. – 4-е изд. – Wiley, 2017. – 312 с.

UDC 621.396.96

REAL-TIME OBJECT DETECTION AND TRACKING ON RASPBERRY PI USING THRESHOLD SEGMENTATION AND CSRT TRACKER

Tun Tun Aung, master's student

*Institute of Information Technologies of the Belarusian State University of Informatics and Radioelectronics,
Minsk, Republic of Belarus*

Dubovik I. A. – Candidate of Technical Sciences, Associate Professor

Abstract. In the article, the problem of real-time object detection and tracking using the Raspberry Pi 4 platform and the OpenCV library is discussed. Two approaches to object detection and tracking are proposed: one based on threshold segmentation, and another using the CSRT tracker. A prototype tracking system was developed on the Raspberry Pi 4 platform with OpenCV, incorporating a camera and servomechanisms. This setup was used to evaluate the accuracy and processing speed (FPS) of the proposed detection and tracking methods. The developed prototype can be applied in robotics and video surveillance systems where low cost and energy efficiency are crucial. In future include algorithm optimization through hardware acceleration and extending functionalities for multi-target tracking.

Keywords. optoelectronic systems, measurement, detection, air target.