

DOI: 10.32517/0234-0453-2022-37-4-34-45

## ТЕХНИЧЕСКОЕ ЗРЕНИЕ: ПРАКТИКУМ ПО НАСТРОЙКЕ И ПРОГРАММИРОВАНИЮ

Г. Л. Абдулгалимов<sup>1</sup> ✉, О. А. Косино<sup>1</sup>, К. В. Гоголданова<sup>1</sup><sup>1</sup> *Московский педагогический государственный университет, г. Москва, Россия*✉ [agraml@mail.ru](mailto:agraml@mail.ru)

### Аннотация

В статье предложено учебно-методическое обеспечение для будущих учителей информатики и робототехники по настройке и программированию модуля технического зрения TrackingCam. Рассмотрены примеры проектов с модулем технического зрения и Arduino-совместимым микроконтроллером. Описанный в статье дидактический материал по разработке учебно-исследовательских проектов с модулем технического зрения включает две лабораторные работы: первая посвящена настройке модуля технического зрения на выполнение специальных задач по распознаванию конкретных объектов, а вторая — использованию настроек модуля технического зрения для программирования поведения робототехнического самоходного устройства. Предложенный дидактический материал предназначен для организации лабораторных работ по информатике и робототехнике с обучающимися разных возрастных категорий и на разных уровнях образования. Рассмотренные в статье простые примеры могут служить основой для разработки более сложных проектов, включая курсовые и дипломные работы. Предложенные работы состоят из следующих частей: тема; цель; теоретические знания, используемые для решения заданий в текущей работе; порядок выполнения тренировочных заданий; контрольные задания; вопросы для контроля. Таким образом, материал статьи в целом может быть использован для организации практикума по техническому зрению.

**Ключевые слова:** профессиональная подготовка учителя информатики и робототехники, модуль технического зрения, обучение конструированию и программированию, среда разработки Arduino IDE.

### Для цитирования:

Абдулгалимов Г. Л., Косино О. А., Гоголданова К. В. Техническое зрение: практикум по настройке и программированию. *Информатика и образование*. 2022;37(4):34–45. DOI: 10.32517/0234-0453-2022-37-4-34-45

## COMPUTER VISION: A PRACTICUM IN SETTING UP AND PROGRAMMING

G. L. Abdulgalimov<sup>1</sup> ✉, O. A. Kosino<sup>1</sup>, K. V. Gogoldanova<sup>1</sup><sup>1</sup> *Moscow Pedagogical State University, Moscow, Russia*✉ [agraml@mail.ru](mailto:agraml@mail.ru)

### Abstract

The article provides future teachers of informatics and robotics with educational and methodological support for setting up and programming the TrackingCam vision module. The research looked at examples of projects with a computer vision module and an Arduino-compatible microcontroller. The didactic material on developing educational research projects with a vision module described in the article includes two laboratory projects. The first focuses on setting up the vision module to perform special tasks of identifying particular objects. The second project involves using the vision module settings to program the behavior of a self-propelled robotic device. The proposed didactic material is intended for organizing laboratory projects in informatics and robotics for students of different ages and at different levels of education. The simple examples considered in the article can serve as the basis for designing more complex projects, including term articles as well as bachelor and master theses. The proposed projects consist of the following parts: topic; purpose; theoretical knowledge used to complete the tasks in the project in hand; the procedure for performing training tasks; tests; test questions. Thus, the material of the whole article can be used to organize a practicum in computer vision.

**Keywords:** informatics and robotics teacher's professional training, vision module, design and programming training, Arduino integrated development environment (IDE).

### For citation:

Abdulgalimov G. L., Kosino O. A., Gogoldanova K. V. Computer vision: A practicum in setting up and programming. *Informatics and Education*. 2022;37(4):34–45. (In Russian.) DOI: 10.32517/0234-0453-2022-37-4-34-45

### 1. Введение

В последнее десятилетие в России образовательная робототехника как учебная дисциплина для школьников и студентов развивается при активной

поддержке государства. С 2008 года функционирует программа «Робототехника: инженерно-технические кадры инновационной России», которая посвящена популяризации образовательной робототехники и обеспечению равных возможностей

доступа к оборудованию для школьников из разных регионов. В последние два-три года Правительством и Президентом РФ были утверждены «Стратегическое направление в области цифровой трансформации науки и высшего образования» и «Национальная стратегия развития искусственного интеллекта на период до 2030 года», в которых особое внимание уделяется необходимости подготовки детей и молодежи к разработке и внедрению технологических решений на базе искусственного интеллекта (машинное обучение, техническое зрение, интернет вещей и др.) [1–5].

В условиях быстрого развития материально-технического обеспечения курсов робототехники актуальной становится проблема методического обеспечения учебного процесса в соответствии с содержанием курсов, составом робототехнических наборов и возрастной категорией обучающихся. Основное содержание курсов робототехники сводится к конструированию и программированию различных автоматизированных, умных (в том числе самоходных и летательных), электронных и электронно-механических устройств, которые решают прикладные задачи, такие как: включение и выключение конкретного устройства по сигналу от датчика; измерение расстояния до препятствия и своевременная остановка робота; движение робомобиля по линии или по заданному маршруту; сортировка деталей по цвету или по форме; следование за объектом и т. д.

На курсах робототехники используются робототехнические наборы, которые содержат разные составы комплектующих: микроконтроллерный модуль (LEGO EV3, Open CM, Arduino (UNO, MEGA, NANO и др.), Arduino-подобные микроконтроллерные модули («Технолаб», Applied robotics, CM-530, VEX, Roborobo и др.); силовые модули управления отдельными устройствами и двигателями; модули расширения возможностей микроконтроллерной платы; модули беспроводной связи; датчики (движения, расстояния, температуры, влажности, света, газа и др.); светодиодные индикаторы, жидкокристаллические индикаторы; двигатели и приводы (серво-, шаговые и др.); блоки питания; радиодетали и соединительные провода; конструкторские детали и крепежные элементы и др. [6–8].

## 2. Актуальность разработки практикума по настройке и программированию модуля технического зрения

В последние годы робототехнические наборы комплектуются модулями технического зрения (МТЗ), которые используются в ученических и научно-исследовательских проектах. Модуль технического (а также компьютерного или машинного) зрения — аппаратно-программный комплекс, включающий: цифровую камеру, плату управления с процессором и памятью, интерфейсы для связи и передачи дан-

ных, алгоритмы для распознавания объектов и др. МТЗ предназначен для решения таких прикладных задач, как: определение координат и площади объекта в кадре, сортировка объектов по цвету и по форме, движение робомобиля по неразмеченной траектории, поиск и следование за конкретным объектом, игра роботов в футбол и сумо, выполнение логистических операций на складе, автоматизация конвейера сборочного цеха и т. д.

Вопросам разработки средств и методов технического зрения, в том числе более эффективных алгоритмов обработки графических изображений и идентификации объектов, уделяется значительное внимание во многих отечественных (А. В. Бондаренко, Ю. В. Визильтер, С. Ю. Желтов, М. В. Ососков, А. В. Чернявский и др.) и зарубежных (R. Achanta, F. Estrada, D. Forsyth, S. Hemamiz, D. Kriegman, J. Ponce, S. Susstrunk, M. H. Yang и др.) исследованиях и публикациях. Многими учеными (А. Н. Афанасьев, Л. Д. Жимбуева, О. А. Лучанский, М. И. Николаев, О. В. Руденко, М. М. Фролов и др.) исследовались проблемы разработки аппаратного обеспечения прикладных задач технического зрения.

Однако при всем обилии и разнообразии исследований в области технического зрения слабо проработаны методические вопросы обучения таким новым знаниям и отсутствуют дидактические материалы по освоению ряда распространенных сегодня МТЗ. Для обучения школьников и студентов решению типовых и проектно-исследовательских задач с использованием МТЗ требуется апробированное учебно-методическое обеспечение под конкретное аппаратное обеспечение: микроконтроллеры, комплектующие, робототехнические наборы и т. д.

Рассмотрим практикум по настройке и программированию МТЗ TrackingCam от известного российского производителя ООО «Прикладная робототехника». МТЗ TrackingCam в последние годы очень популярен в образовательных организациях и в ресурсных центрах (технопарки, кванториумы и др.), так как входит в комплект многих робототехнических наборов ООО «Прикладная робототехника». Предлагаемый нами практикум апробирован на базе курсов повышения квалификации в педагогическом «Кванториуме» в Академии Минпросвещения России. Практикум включает лабораторные работы для: знакомства с характеристиками МТЗ TrackingCam и установки необходимого программного обеспечения; его настройки на распознавание конкретного объекта (образа); использования МТЗ в робототехнических проектах на базе микроконтроллерной платформы Arduino [9–12].

Практикум состоит из двух лабораторных работ. Представленный дидактический материал также может быть интегрирован в любой другой практикум по робототехнике и информатике. Каждая работа состоит из следующих разделов: «Тема», «Цель», «Оборудование», «Теоретические сведения», «Тренировочные задания», «Контрольные задания», «Вопросы для контроля».

### 3. Настройка модуля технического зрения TrackingCam (лабораторная работа)

**Цель:** ознакомиться с основными элементами и характеристиками MTЗ TrackingCam, установить драйвер и изучить возможности программы распознавания объектов.

**Оборудование:** компьютер с доступом в интернет, модуль технического зрения TrackingCam, кабель USB-miniUSB, предметы для опытов по распознаванию: кубики и шарики разных цветов и размеров — от 5 до 25 см в диагонали.

**Теоретические сведения.** MTЗ TrackingCam состоит из аппаратного и программного обеспечения. Аппаратная часть — это цифровая камера с платой управления (рис. 1, а).

На плате управления (рис. 1, б) расположены: 1 — микропроцессор; 2 — кнопка для перезагрузки модуля; 3 — разъем USB для подключения к компьютеру; 4 — разъемы для подключения Dynamixel — совместимых устройств от Robotis; 5 — разъем для подключения к внешним устройствам по интерфейсам: UART, I2C и SPI. На лицевой стороне рядом с камерой расположен светодиод для индикации состояния модуля.

Модуль имеет следующие технические характеристики: 1) разрешение матрицы камеры —  $640 \times 480$  точек; 2) частота кадров — 30 кадр/мин; 3) память — 168 Кбайт (ОЗУ) и 512 Кбайт (Флеш); 4) угол обзора камеры —  $75^\circ$ ; 5) глубина цвета — 8 бит; 6) количество одновременно распознаваемых одинаковых объектов — 255; 7) количество запоминаемых различных образов — до десяти одноцветных и до пяти составных разноцветных; 8) количество цветных областей в составном объекте — от одного до трех; 9) настраиваемые признаки распознаваемых объектов: «Яркость», «Цвет», «Площадь», «Округлость», «Выпуклость», «Вытянутость»,

«Координаты», «Ширина» и «Высота»; 10) габариты модуля —  $3,8 \times 3,8 \times 3,2$  см.

Программное обеспечение для работы с MTЗ TrackingCam состоит из драйвера и программы для настройки распознавания конкретных объектов. Программное обеспечение распространяется свободно на сайте ООО «Прикладная робототехника» [appliedrobotics.ru](http://appliedrobotics.ru), раздел «Учебные материалы», подраздел — «Программное обеспечение».

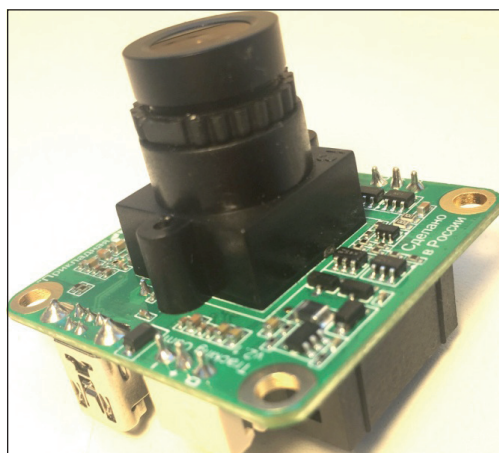
**Тренировочное задание 1.** Ознакомиться с элементами управления и интерфейсами MTЗ TrackingCam, установить драйвер для корректного соединения модуля с компьютером, запустить и ознакомиться со всеми окнами и опциями программного обеспечения настройки распознавания объектов, включить и настроить камеру, а также настроить и сохранить образ цветного шарика в памяти модуля.

Для решения этого задания необходимо выполнить следующие действия:

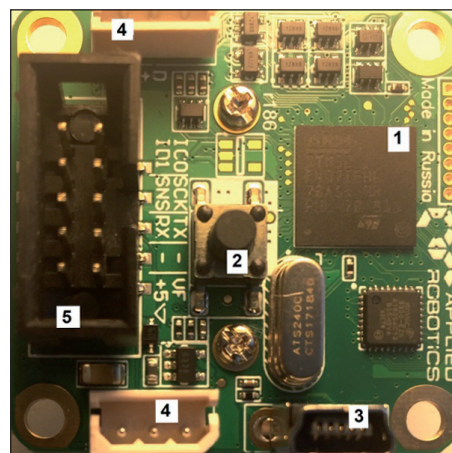
1. Скачать и распаковать архив TrackingCam\_v2.9 с сайта [appliedrobotics.ru](http://appliedrobotics.ru). Для установки драйвера запустить приложение **zadig**, в открывшемся окне выбрать из списка драйвер и щелкнуть кнопку **Install**. Дождаться завершения процесса установки.

2. Подключить MTЗ TrackingCam к компьютеру с помощью USB-кабеля и для проверки корректного подключения запустить «Диспетчер устройств». Об установке драйвера и о подключении модуля к компьютеру свидетельствует запись **Mass Storage in HS Mode** в разделе «Устройства USB» в окне «Диспетчера устройств».

3. После успешного подключения MTЗ к компьютеру запустить приложение TrackingCam. Программа откроется в шести отдельных, свободно перемещаемых по экрану окнах (рис. 2, а и 2, б). Рассмотрим назначение каждого окна: 1) окно настройки модуля TrackingCam и настройки различных параметров распознаваемого объекта. В этом окне нужно щел-



а



б

Рис. 1. Модуль технического зрения TrackingCam:

а — общий вид модуля; б — элементы платы управления

Fig. 1. TrackingCam vision module:

а — general view of the module; б — elements of the control board

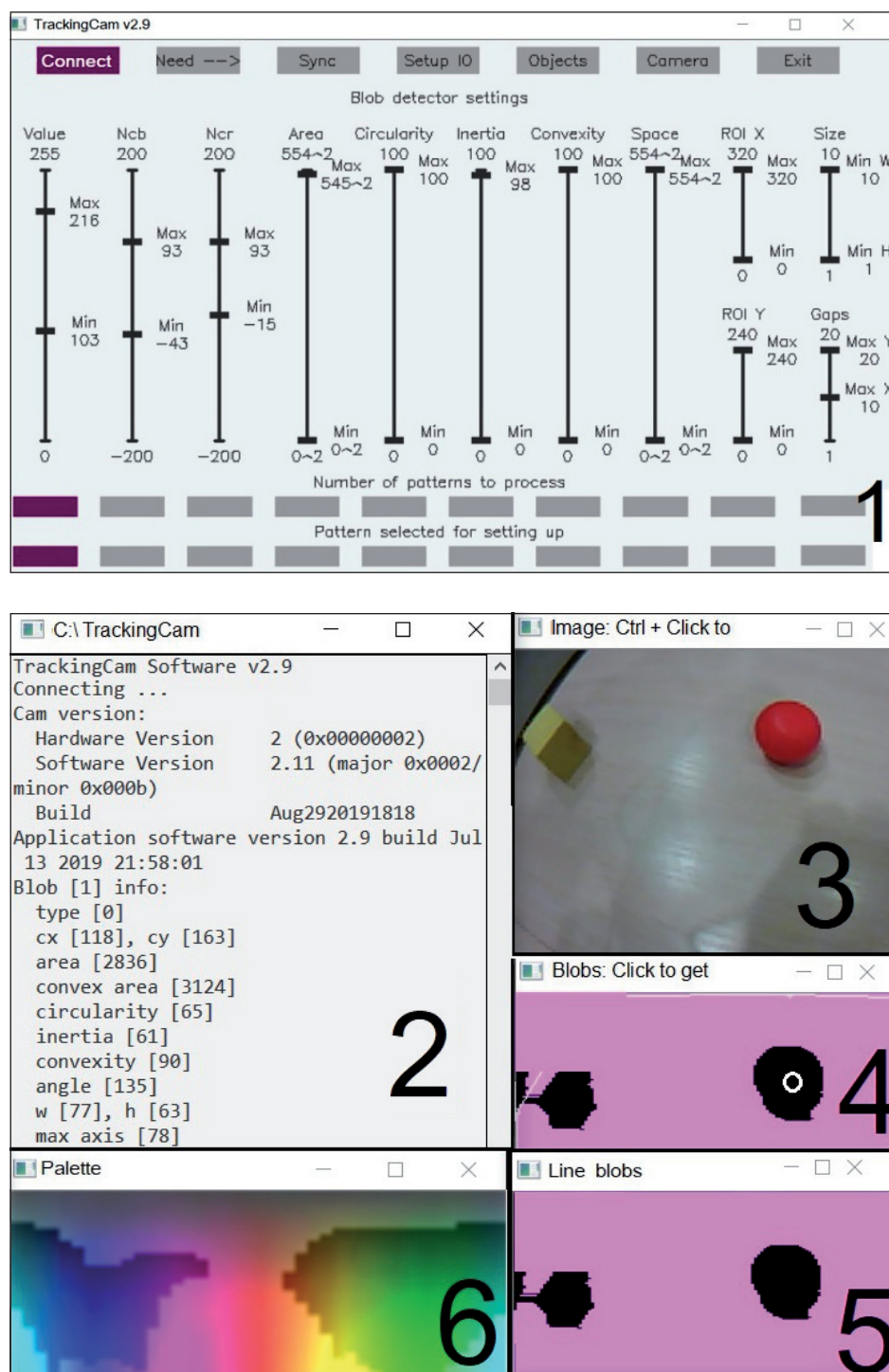


Рис. 2. Окна приложения TrackingCam

Fig. 2. TrackingCam application windows

кнута кнопкой мыши на **Connect** для включения (активизации) камеры, после чего в окне № 3 (рис. 2, б) должно появиться изображение (видеопоток) с камеры. Четкость изображения настраивается вручную, прокручивая объектив в одну или в другую сторону. После настройки объектива его положение фиксируется специальным кольцом и винтом; 2) консольное окно для вывода строки об успешном подключении модуля и параметров распознанных объектов; 3) окно

вывода изображения с камеры имеет заголовок **Image: Ctrl + Click to select...**, в котором читается подсказка: как выбирать тот или иной объект распознавания в кадре, т. е. необходимо, зажав на клавиатуре Ctrl, кликнуть мышью по конкретному объекту в кадре; 4) окно отображения очертаний образов выбранных объектов имеет заголовок **Blobs: Click to get info**, в котором читается подсказка: кликните на объект для вывода информации о нем. Информация

выводится в окне № 2; 5) окно **Line blobs** отображает образ объекта, который после настроек параметров в окне № 1 должен быть похож на реальный объект из окна № 3 по цвету и по форме; 6) окно палитры цветов для наблюдения соответствия цвета объекта, выбранного в окне № 5.

4. Настроить MT3 TrackingCam для распознавания объекта, например, шарика оранжевого цвета. Для этого нужно выполнить следующую последовательность действий: а) расположить в кадре шарик на расстоянии 15–25 см от камеры так, чтобы не было сливающегося фона и других похожих предметов; б) удерживая клавишу Ctrl, кликнуть в окне № 3 по изображению шарика; в) с помощью регуляторов в окне № 1 (ниже описано назначение этих регуляторов) добиться более реалистичного отображения шарика в окне № 5 по цвету и по форме; г) для отображения параметров шарика в окне № 2 перейти в окно № 4, навести курсор в виде кружочка на шарик и кликнуть мышью; д) для сохранения образа шарика в память модуля нажать кнопку **Sync** в окне № 1, а в появившемся окне выбрать **Push current settings to camera** и нажать кнопку **Close**, далее — кнопку **Save**, после этого надпись на этой кнопке изменится на **Saved**; е) после успешного выполнения всех этих действий модуль будет настроен на распознавание оранжевого шарика. В этом можно убедиться так: отключить от компьютера USB-кабель модуля и подключить его к блоку питания 5 вольт, например, от смартфона; светодиод модуля на лицевой стороне рядом с камерой помирает 3–5 с, указывая на загрузку внутренних настроек, и далее загорится красным цветом. Если мы наведем камеру на оранжевый шарик, то цвет светодиода изменится на светло-голубой. Это означает, что искомый объект по настройкам модуля распознан и находится в кадре.

5. Ознакомиться с элементами окон приложения TrackingCam. Рассмотрим элементы главного окна настроек (окно № 1 на рис. 2, а).

*Верхний ряд кнопок:* а) **Connect** — активация модуля в программе, после чего начинается видеоизображение с камеры, а название кнопки меняется на **Connected**; б) **Need→** — под этой кнопкой скрывается кнопка для сохранения настроек **Save**, которая после успешного сохранения настроек в память модуля переименовывается в **Saved**, однако для перехода к скрытой кнопке **Save** необходимо сначала нажать соседнюю кнопку **Sync**; в) **Sync** — при нажатии этой кнопки открывается окно, где видны четыре варианта выбора: **Load settings from camera** («Загрузить настройки (ранее сохраненные) с камеры»); **Push current settings to camera** («Перенести (сохранить) текущие настройки в камеру»); **Load settings from file** («Загрузить настройки из файла»); **Save settings to file** («Сохранить настройки в файл»); г) **Setup IO** — эта кнопка открывает окно для настройки интерфейсов (UART, I2C и SPI), соединения MT3 и микроконтроллерной платы, где задаются параметры: стандарт интерфейса, адрес, скорость обмена данными, напряжение питания;

д) **Objects** — эта кнопка открывает окно со сложной структурой и большим количеством опций. Оно предназначено для объединения двух или трех цветных образов в один комбинированный образ или для распознавания разноцветных объектов. Это окно подробнее рассмотрено ниже; е) **Camera** — окно настройки цветových составляющих видеопотока камеры, здесь по умолчанию установлен автоматический режим; ж) **Exit** — выход из приложения.

*Регуляторы настроек Blob detector settings:* **Value** — яркость; **Ncb** — синяя цветоразность; **Ncr** — красная цветоразность; **Area** — площадь (в пикселях) цветного образа (пятна); **Circularity** — округлость или угловатость (меняется от 0 до 100, где 0 — многоугольник, а 100 — круг); **Inertia** — вытянутость (0 — вытянутая узкая фигура, 100 — круг); **Convexity** — выпуклость или присутствие дырок и впадин (0 — много впадин и дыр, 100 — полностью выпуклая фигура (многоугольник)); **Space** — площадь описанного многоугольника; **ROI X** и **ROI Y** — диапазон координат по горизонтали и по вертикали; **Size** — ширина и высота; **Gaps** — разрывы между пикселями по горизонтали и по вертикали.

*Нижние ряды кнопок:* **Number of patterns to process** — количество различных образов, которые модуль будет распознавать; **Pattern selected for setting up** — выбор конкретного образа для настройки.

6. Ознакомиться с информацией в окне № 2 (рис. 2, б): а) строка **Blob [3] info** указывает на количество распознанных образов одинакового или разных типов, например, если модуль был обучен на распознавание оранжевых шариков и зеленых кубиков, то, возможно, в кадре находятся три шарика или два шарика и один кубик и т. д., а сколько и чего именно обнаружено камерой, определяется по параметру **type**; б) **type** — указывает на номер образа объекта (шарик, кубик и др.), сохраненного в памяти модуля. Нумерация начинается с нуля и продолжается в той последовательности, в которой образы настраивались и сохранились в памяти (всего 10, от 0 до 9); в) **cx** — координата центра образа по горизонтали от 0 до 320; г) **cy** — координата центра образа по вертикали от 0 до 240; д) **area** — площадь образа в пикселях, меняется при приближении и отдалении объекта от камеры. Например, шарик для настольного тенниса имеет площадь 20 пикселей на расстоянии 50 см от объектива, а на расстоянии 5 см имеет площадь 2000 пикселей. Используя этот параметр, можно вычислить реальный размер объекта и расстояние до него; е) **convex area** — площадь описанного около образа многоугольника; ж) **circularity** — значение округлости; з) **inertia** — вытянутость; и) **convexity** — выпуклость; к) **angle** — угол поворота; л) **w** — ширина образа; м) **h** — высота образа; н) **max axis** — длина большей диагонали описанного многоугольника.

*Тренировочное задание 2.* Объединить два объекта разных цветов в один комбинированный объект и настроить модуль на распознавание этого двухцветного объекта, например, кубик и шарик разных

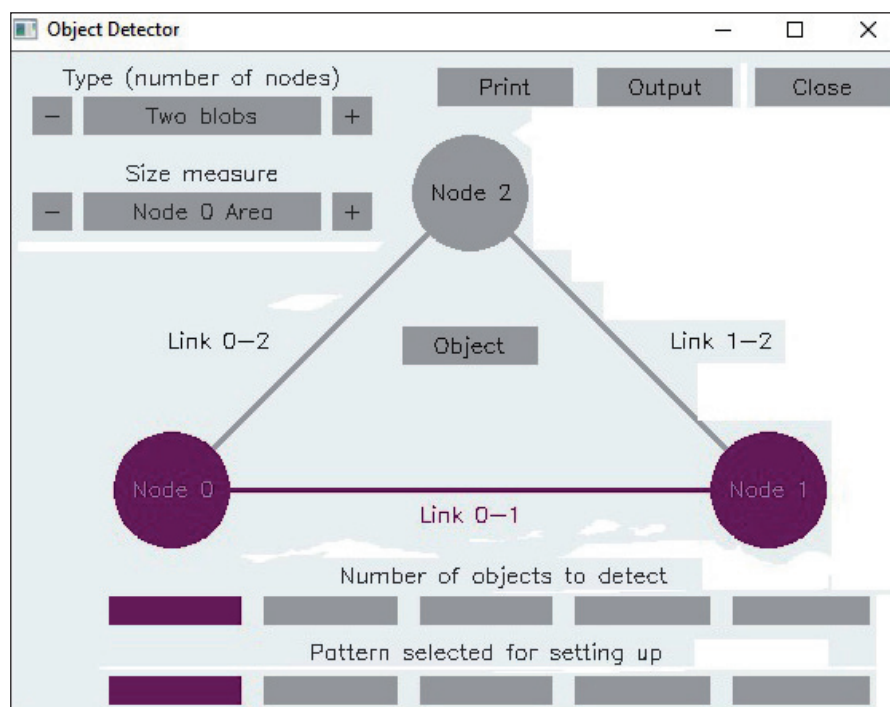


Рис. 3. Окно для объединения разноцветных объектов

Fig. 3. Window for combining multicolored objects

цветов можно поставить рядом и распознать как один комбинированный разноцветный объект.

Рассмотрим порядок действий по распознаванию разноцветных объектов:

1) запустить приложение TrackingCam и активировать камеру нажатием кнопки **Connect** в главном окне;

2) выставить перед камерой на расстоянии 20–30 см кубик и шарик разных цветов;

3) выбрать два распознаваемых объекта в разделе **Number of patterns to process** нажатием на два прямоугольника из десяти; далее поочередно настраивать распознавание образов кубика и шарика, выделяя в разделе **Pattern selected for setting up** сначала первый прямоугольник, а затем — второй;

4) после настройки на распознавание обоих объектов в отдельности нужно нажать кнопку **Objects**

в главном окне (рис. 2, а) и открыть окно для работы с объединением (комбинированием) разноцветных объектов (рис. 3);

5) в разделе **Type (number of nodes)** нажатием кнопки «+» или «-» задать количество цветных областей, которые нужно объединить: **Single blob** — одна цветная область, **Two blobs** — две области, **Three blobs** — три области. Для выполнения нашего задания нужно выбрать **Two blobs**;

6) в разделах **Number of objects to detect** и **Pattern selected for setting up** выделить по одному делению из пяти возможных (это означает, что мы работаем только с одним, комбинированным из двух частей, разноцветным объектом), после чего объекты будут объединены, т. е. на каждом объекте по центру появятся кружочки, соединенные между собой линией, как показано на рисунке 4;

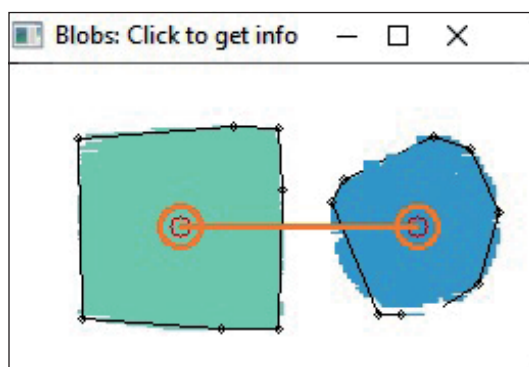


Рис. 4. Образование комбинированного разноцветного объекта

Fig. 4. Creating a combined multicolored object

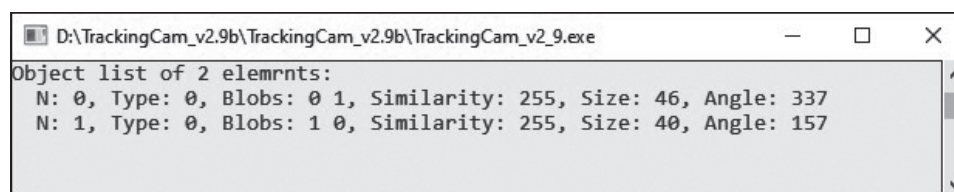


Рис. 5. Информация о комбинированном разноцветном объекте

Fig. 5. Information about the combined multicolored object

7) выводить информацию о комбинированном разноцветном объекте в консольное окно для отображения параметров (рис. 5) с помощью кнопки **Print** (рис. 3);

8) нажать кнопку **Object** в центре окна (рис. 3), а также на круги с надписями **Node**, чтобы открывать окна для корректировки параметров цветных областей в комбинированном объекте;

9) после выполненных настроек сохранить образ комбинированного разноцветного объекта в памяти модуля через кнопку **Sync** в главном окне;

10) чтобы убедиться в сохранении настроек распознавания комбинированного разноцветного объекта, нужно: отключить модуль технического зрения от компьютера и подключить его к блоку питания на 5 вольт (можно подключить к зарядному устройству от смартфона); выставить перед камерой кубик и шарик, так же как при настройке; обратить внимание на цвета светодиода состояния на модуле рядом с камерой: красный цвет означает, что распознаваемый объект вне кадра, т. е. не распознан; зеленый — комбинированный разноцветный объект распознан и находится в кадре; светло-голубой — распознан одинарный одноцветный объект (или только часть комбинированного разноцветного объекта).

#### Контрольные задания

1. Настроить MT3 TrackingCam на распознавание кубика любого однотонного цвета. Вывести параметры распознавания. Сохранить настройки и проверить распознавание кубика по светодиоду состояния.

2. Настроить MT3 TrackingCam на распознавание трех-пяти однотонных объектов, различных по цвету и по форме. Сохранить настройки и проверить распознавание по светодиоду состояния.

3. Настроить MT3 TrackingCam на распознавание объекта, комбинированного из: а) двух шариков разных цветов; б) трех составляющих: двух кубиков разных цветов и шарика, не совпадающего с кубиками цвета. Сохранить настройки и проверить распознавание по светодиоду состояния.

#### Контрольные вопросы

1. Что такое техническое зрение?
2. Каково разрешение матрицы на цифровой камере модуля MT3 TrackingCam?
3. Какие интерфейсы для связи устройств реализованы на MT3 TrackingCam?
4. Сколько различных одноцветных образов можно сохранить в MT3 TrackingCam?

5. Какой архив нужно скачать и с какого сайта для первоначальной настройки и дальнейшей работы с MT3 TrackingCam?

6. Как нужно сохранять уже настроенный образ объекта в память MT3 TrackingCam?

7. Какие параметры объекта используются для настройки и распознавания его образа в MT3 TrackingCam?

8. На что указывают светло-голубой и зеленый цвета светодиода состояния на MT3 TrackingCam?

## 4. Использование модуля технического зрения в робототехнических проектах на базе микроконтроллерной платформы Arduino (лабораторная работа)

**Цель:** подключить MT3 TrackingCam к микроконтроллерной платформе Arduino и использовать параметры распознавания из MT3 в проекте слежения робомобиля за конкретным объектом.

**Оборудование:** компьютер с доступом в интернет, модуль технического зрения TrackingCam, предметы для опытов по распознаванию: кубики и шары разных цветов, микроконтроллерная плата Arduino MEGA или UNO и двухколесный робомобиль Arduino с моторами постоянного тока и платой управления, соединительные провода.

**Теоретические сведения.** На плате MT3 TrackingCam расположен разъем DC3-10P (рис. 6), в котором реализованы три стандартных интерфейса UART, I2C и SPI для подключения модуля к микроконтроллерным платам, например, Arduino MEGA или Arduino UNO. На плате MT3 рядом с разъемом интерфейсов указано назначение всех контактов. Для определения нумерации контактов нужно расположить разъем, вертикально повернув прорез в корпусе разъема влево, тогда нумерация контактов читается

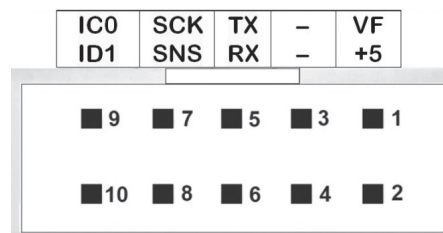


Рис. 6. Назначение контактов интерфейсного разъема MT3 TrackingCam

Fig. 6. Pin assignment of OC TrackingCam interface connector

слева направо и сверху вниз (рис. 6). Контакты 2 и 4 предназначены для питания модуля, причем контакт 2 — это +5В (Vcc), а контакт 4 — «Земля» (Gnd). Контакты 1 и 3 используются для вывода напряжения 5 вольт из модуля для питания другого устройства, обычно эти контакты не применяются, чтобы не перегружать модуль.

Рассмотрим порядок соединения модуля технического зрения и платы Arduino по интерфейсам UART, I2C и SPI. В таблице отражены соответствия контактов на MT3 и на плате Arduino.

Таблица/ Table

#### Соответствия интерфейсных контактов на плате Arduino и на MT3

#### Correspondence of interface connector contacts on the Arduino board and with overcurrent protection

| UART    |     | I2C     |     | SPI     |     |
|---------|-----|---------|-----|---------|-----|
| Arduino | MT3 | Arduino | MT3 | Arduino | MT3 |
| TX1     | RX  | SCL     | IC0 | MISO    | TX  |
| RX1     | TX  | SDA     | ID1 | MOSI    | RX  |
|         |     |         |     | SCK     | SCK |
|         |     |         |     | SS      | SNS |

Далее кратко поясним особенности использования этих интерфейсов в проектах.

UART (Universal Asynchronous Receiver-Transmitter) — универсальный асинхронный приемопередатчик для организации связи между цифровыми устройствами со скоростью передачи данных до 2 Мбит/с. UART использует две двунаправленные линии связи RX и TX. Причем RX от передатчика соединяется с TX приемника, а TX передатчика — с RX приемника. Так, один интерфейс UART может быть использован для связи только одной пары устройств между собой. В плату Arduino UNO встроены только один интерфейс UART (на цифровых портах под номерами 0 и 1), а на плате Arduino MEGA реализованы четыре интерфейса UART (на контактах: 0–1, 19–18, 17–16 и 15–14). Следует заметить, что «Монитор порта» в Arduino IDE работает по интерфейсу UART. Поэтому если в Arduino-проекте используется «Монитор порта», то единственный интерфейс окажется занят. И если в проекте кроме «Монитора порта» требуется использование UART для подключения различных датчиков и модулей (типа MT3), то нужно выбрать плату Arduino MEGA, в котором реализовано до четырех интерфейсов UART.

I<sup>2</sup>C (ИЦ — ай ты си) (Inter-Integrated Circuit) — последовательная асимметричная шина для связи между цифровыми электронными устройствами. Все операции по шине I<sup>2</sup>C осуществляются при помощи двух линий: последовательных данных (SDA) и синхронизации (SCL). Шина применяется для соединения низкоскоростных (до 100 Кбит/с) периферийных устройств с микропроцессорами и микроконтроллерами. По одному интерфейсу I<sup>2</sup>C можно подключить до 127 устройств, которым присваиваются адреса

в диапазоне 1–127. На всех платах Arduino реализован интерфейс I<sup>2</sup>C, который легко определить по названиям контактов: SDA и SCL, например, в конце ряда цифровых портов. I<sup>2</sup>C дублируется в двух-трех местах на платах Arduino (распиновка плат Arduino представлена на [wiki.amperka.ru](http://wiki.amperka.ru)).

SPI (Serial Peripheral Interface) — последовательный периферийный интерфейс синхронной передачи данных между микроконтроллерами и периферийными устройствами с частотой синхронизации до десятков МГц. В SPI используются четыре цифровых сигнала: MOSI (Master Out Slave In) — выход ведущего, вход ведомого; MISO (Master In Slave Out) — вход ведущего, выход ведомого; SCLK или SCK (Serial Clock) — тактовый сигнал; CS или SS (Chip Select, Slave Select) — выбор конкретного ведомого устройства. Интерфейс SPI позволяет подключать к одному ведущему устройству одно или несколько ведомых устройств. На всех типах плат Arduino реализован интерфейс SPI (распиновка плат Arduino представлена на [wiki.amperka.ru](http://wiki.amperka.ru)).

Для настраивания интерфейса соединения MT3 и платы Arduino после настройки MT3 на распознавание какого-то объекта нужно перейти в окно настройки интерфейсов нажатием кнопки **Setup IO** в главном окне (рис. 1, а). В открывшемся окне вводятся следующие настройки интерфейсов: а) UART (UART full duplex, Baud rate 115200, 51); б) I2C (I2C, 51); в) SPI (Spi); напряжение во всех интерфейсах — 5 вольт.

Для программирования Arduino-проектов с модулем технического зрения используются библиотеки интерфейсов «TrackingCamDxlUart», «TrackingCamI2C» и «TrackingCamSPI», в которых реализованы функции: а) объявление произвольной переменной типа TrackingCam, например, `trCam1: TrackingCamDxlUart trCam1, TrackingCamI2C trCam1, TrackingCamSPI trCam1`; б) инициализация и установка адреса и скорости: `trCam1.init(51, 1, 115200, 30), trCam1.init(51, 400000), trCam1.init(24, 1000000)`; в) количество распознаваемых объектов и образов: `trCam1.readObjects(1), trCam1.readBlobs(4)`; г) возвращение параметров распознанного образа: `trCam1.blob[1].type, trCam1.blob[1].dummy, trCam1.blob[1].cx, trCam1.blob[1].cy, trCam1.blob[1].area, trCam1.blob[1].left, trCam1.blob[1].right, trCam1.blob[1].top, trCam1.blob[1].bottom`.

**Тренировочное задание 1.** Изучить схемы подключения MT3 TrackingCam к интерфейсам микроконтроллерной платы Arduino UNO или Arduino MEGA; скачать и подключить к Arduino IDE библиотеки интерфейсов UART, I2C и SPI для MT3; настроить MT3 на распознавание шарика оранжевого цвета; запустить пример, встроенный в Arduino IDE, для отображения параметров распознанного образа в «Мониторе порта».

Для выполнения этого задания нужно произвести следующие действия:

1. Скачать с сайта ООО «Прикладная робототехника» ([appliedrobotics.ru](http://appliedrobotics.ru), раздел «Учебные

материалы», подраздел «Программное обеспечение») библиотеки для программирования MT3 в среде Arduino IDE: «TrackingCamDxlUart» — для интерфейса UART, «TrackingCamI2C» — для I2C и «TrackingCamSPI» — для SPI.

2. Подключить все три архива библиотек интерфейсов к среде разработки Arduino IDE следующим образом: запустить среду Arduino IDE, открыть меню **Скетч**, перейти в пункт «Подключить библиотеку» и выбрать **Добавить .ZIP библиотеку...** Далее в открывшемся окне выбрать один из архивов библиотек: «TrackingCamDxlUart», «TrackingCamI2C» или «TrackingCamSPI» и нажать кнопку **Open**. Повторить эти действия для подключения каждой библиотеки. Подключенные библиотеки отображаются в списке **Скетч** → **Подключить библиотеку**.

3. Подключить MT3 TrackingCam к компьютеру и настроить его на распознавание оранжевого шарика. Далее нажать кнопку **Setup IO** и в открывшемся окне выбрать интерфейс **I2C**. После выбора интерфейса нажать кнопку **Apply**, а затем — **Close**.

4. Сохранить все выбранные настройки в MT3: **Sync** → **Push current settings to camera**, и отключить USB-кабель модуля от компьютера.

5. Подключить с помощью соединительных проводов MT3 TrackingCam к плате Arduino UNO или MEGA (табл.) по интерфейсу I2C (IC0-SCL, ID1-SDA), а далее подключить провода питания (+5 подключить к 5 вольт, а «-» подключить к GND).

6. Подключить плату Arduino (с уже подсоединенным к нему MT3) к компьютеру и запустить среду разработки Arduino IDE. Далее перейти в меню **Инструменты** и настроить соответственно **Порт** и **Плата**.

7. Открыть встроенный пример для отображения параметров распознанного объекта в «Мониторе порта»: **Файл** → **Примеры** → **TrackingCamI2C** → **readBlobs**. Загрузить скетч в плату Arduino: **Скетч** → **Загрузка**. В нижней части окна Arduino IDE появится сообщение: «Загрузка завершена».

8. Запустить **Монитор порта** (в меню **Инструменты**), выставить скорость 115200 бод, т. е. такую же скорость, какая объявлена в скетче. Проверить параметры распознавания шарика, которые отобра-

жаются в «Мониторе порта», и определить, какие параметры соответствуют разным переменным из скетча, например, чему равны значения параметров **area**, **cx**, **cy** и др.

9. Выставить шарик перед камерой прямо по центру на расстоянии 10 см. Перейти в «Монитор порта», остановить «Автопрокрутку» (убрав галочку в нижней части окна) и скопировать или переписать все числовые значения параметров распознавания с соответствующими переменными и с их названиями.

*Тренировочное задание 2.* Закрепить MT3 TrackingCam на двухколесный роботомобиль на базе платы Arduino; подключить модуль к плате Arduino по интерфейсу I2C; настроить MT3 на распознавание оранжевого шарика; разработать программу (скетч) для следования роботомобиля за распознаваемым объектом, т. е. за оранжевым шариком.

Для решения этого задания нужно выполнить следующие действия:

1. Собрать двухколесный роботомобиль на базе платы Arduino UNO или MEGA с двумя двигателями постоянного тока и платой управления для двигателей на основе драйвера L298N; замкнув переключики на плате управления, задать режим без регулирования скорости вращения двигателей; для управления направлением вращения двигателей использовать следующие цифровые порты на плате Arduino: IN1-4, IN2-5, IN3-10, IN4-11.

2. Установить MT3 TrackingCam на роботомобиль на высоте 10–20 см от уровня пола и под небольшим наклоном (30–40°); соединить MT3 и плату Arduino по интерфейсу I2C, пользуясь таблицей и схемой платы Arduino; соединить провода питания: **5v** на Arduino и **+5** на MT3, а также **GND** на Arduino и «-» на MT3.

3. Настроить MT3 TrackingCam на распознавание оранжевого шарика, выставленного в центре кадра перед камерой на расстоянии 10 см.

4. Подключить плату Arduino на роботомобиле к компьютеру и запустить среду разработки Arduino IDE; набрать следующий скетч для платы Arduino UNO или MEGA, откомпилировать и загрузить в плату Arduino на роботомобиле:

```
#include "TrackingCamI2C.h" //Подключение библиотеки для интерфейса I2C
TrackingCamI2C trackingCam; //Объявление переменной trackingCam

void setup() {
  trackingCam.init(51, 400000); //Установка настроек интерфейса I2C
  pinMode(4,OUTPUT); pinMode(5,OUTPUT); //Объявление портов 1 мотора
  pinMode(10,OUTPUT); pinMode(11,OUTPUT); //Объявление портов 2 мотора
}

void loop() {
  int n = trackingCam.readBlobs(1); //n-переменная хранения количества образов
  int a = trackingCam.blob[0].area; //a-переменная площади образа
  int c = trackingCam.blob[0].cx; //c-переменная координаты по горизонтали

  if(n==0){a=0;c=0;} //если нет распознанных образов, то сброс значений a и c

  digitalWrite(4,LOW); digitalWrite(5,LOW); // стоп моторов и сброс портов
  digitalWrite(10,LOW) digitalWrite(11,LOW);
```

```

delay(200);
  if(a<1500 && c>145 && c<155){ // движение вперёд, если образ в центре
digitalWrite(4,HIGH); digitalWrite(10,HIGH);
delay(15); }
  if(a<1500 && c>155){ //движение направо, так как образ не в центре
digitalWrite(10,HIGH);
delay(15); }
  if(a<1500 && c<145 && c>0){ //движение налево, так как образ не в центре
digitalWrite(4,HIGH);
delay(15); }
}

```

Код для решения этого задания для Arduino-подобной платы Mega 2560 из робототехнического

набора ООО «Прикладная робототехника», в котором комплектуется также MT3 TrackingCam, выглядит так:

```

#include "TrackingCamI2C.h"
TrackingCamI2C trackingCam;

void setup() {
  trackingCam.init(51, 400000);
  pinMode(44,OUTPUT); pinMode(45,OUTPUT); pinMode(46,OUTPUT); pinMode(47,OUTPUT);
}

void loop() {
  int n = trackingCam.readBlobs(1);
  int a = trackingCam.blob[0].area;
  int c = trackingCam.blob[0].cx;
  if(n==0){a=0;c=0;}
  digitalWrite(45,LOW); digitalWrite(47,LOW); analogWrite(44,0); analogWrite(46,0); // стоп
  delay(200);
  if(a<8500 && c>130 && c<170) { //вперёд
    digitalWrite(45,LOW); digitalWrite(47,LOW); analogWrite(44,70); analogWrite(46,70);
    delay(50); }
  if(a<8500 && c>170) { // направо
    digitalWrite(45,LOW); digitalWrite(47,LOW); analogWrite(44,70); analogWrite(46,0);
    delay(30); }
  if(a<8500 && c<130 && c>0) { // налево
    digitalWrite(45,LOW); digitalWrite(47,LOW); analogWrite(44,0); analogWrite(46,70);
    delay(30); }
}

```

5. Разобраться в работе всех блоков скетча, в котором реализован следующий алгоритм: роботомобиль совершает повороты соответственно влево или вправо, пока объект не окажется по горизонтали в центре кадра, а далее роботомобиль приближается к объекту, двигаясь прямо, пока площадь объекта не будет равной площади, если бы камера располагалась на расстоянии 10–15 см от объекта.

6. Включить питание роботомобиля и выставить шарик перед камерой: левее или правее от центра на 30–45° от камеры и на расстоянии 70–90 см; светодиод на камере должен мигать и загореться светло-голубым цветом; далее роботомобиль должен обнаружить шарик, распознать и приблизиться к нему.

#### Контрольные задания

1. Описать работу всех блоков скетча Arduino для приближения роботомобиля к распознанному объекту. Определить, по каким числовым значениям и каким параметрам распознавания происходит движение роботомобиля налево, направо и вперед.

2. Установить на роботомобиль светодиод, подключенный к плате Arduino, настроить MT3 на роботомобиле на распознавание кубика синего или зеленого цвета и модернизировать скетч следующим образом: роботомобиль приближается к распознанному объекту (кубику), зажигается светодиод на 5 с, далее роботомобиль возвращается в заданную точку полигона.

3. Установить на роботомобиль механизм с сервоприводом для захвата распознанных объектов и модернизировать скетч так, чтобы после распознавания и приближения к объекту роботомобиль мог бы захватить распознанный объект и вернуть его в конкретную точку.

#### Контрольные вопросы

1. Какие интерфейсы для подключения к микроконтроллерным платам Arduino встроены в MT3 TrackingCam?

2. Как определяется нумерация контактов в разъеме DC3-10P, где реализованы все интерфейсы на MT3 TrackingCam?

3. Какие контакты в разъеме DC3-10P используются для питания MT3 TrackingCam?

4. Что такое UART (назначение контактов, количество связываемых устройств, скорость передачи информации)?

5. Что такое I2C (название и назначение контактов, количество связываемых устройств, скорость передачи информации)?

6. Что такое SPI (название и назначение контактов, скорость передачи информации)?

7. Какие настройки в окне **Setup IO** программы TrackingCam нужно выбрать для связи MT3 TrackingCam и платы Arduino по интерфейсу UART?

8. Какие настройки в окне **Setup IO** программы TrackingCam нужно выбрать для связи MT3 TrackingCam и платы Arduino по интерфейсу I2C?

9. Какие контакты в разъеме DC3-10P MT3 TrackingCam соответствуют контактам интерфейса I2C?

10. Какой алгоритм реализован в скетче Arduino для приближения робомобиля к распознанному MT3 объекту?

## 5. Выводы

Результаты исследования представляют собой материалы по теоретической и практической подготовке обучающихся, а также контролю полученных ими знаний и умений по такой актуальной проблеме, как: использование модулей технического зрения в составе мобильных роботов для решения различных прикладных задач.

Представленный дидактический материал может быть использован для организации практикума по техническому зрению педагогами по робототехнике на различных уровнях современного образования. Практикум по усмотрению педагога и в зависимости от уровня обучающихся может быть разделен на две лабораторные работы: 1) изучение аппаратного и программного обеспечения модуля технического зрения; 2) использование модуля технического зрения в составе мобильного робота с целью решения задач распознавания, преследования и захвата объектов.

Рассмотренный практикум позволит решать более сложные учебные и проектно-исследовательские задачи с использованием последних достижений в сфере технического зрения, например, таких как: средства и методы стереозрения, технологии распознавания живых объектов: лиц или жестов и др. [18–20].

## Список источников / References

1. Федеральный закон от 29.12.2012 № 273-ФЗ: «Об образовании в Российской Федерации». Режим доступа: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_140174/](http://www.consultant.ru/document/cons_doc_LAW_140174/)

[Federal Law No. 273 FZ dated 29.12.2012: “On Education in the Russian Federation”. (In Russian.) Available at: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_140174/](http://www.consultant.ru/document/cons_doc_LAW_140174/)]

2. Постановление Правительства Российской Федерации от 26.12.2017 № 1642: «Об утверждении государственной программы Российской Федерации “Развитие

образования”». Режим доступа: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_286474/](http://www.consultant.ru/document/cons_doc_LAW_286474/)

[Decree of the Government of the Russian Federation No. 1642 dated 26.12.2017: “On Approval of the State Program of the Russian Federation “Development of Education””. (In Russian.) Available at: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_286474/](http://www.consultant.ru/document/cons_doc_LAW_286474/)]

3. Указ Президента Российской Федерации от 10.10.2019 № 490: «О развитии искусственного интеллекта в Российской Федерации». Режим доступа: <http://publication.pravo.gov.ru/Document/View/0001201910110003>

[Decree of the President of the Russian Federation No. 490 dated 10.10.2019: “On the development of artificial intelligence in the Russian Federation”. (In Russian.) Available at: <http://publication.pravo.gov.ru/Document/View/0001201910110003>]

4. Распоряжение Правительства Российской Федерации от 21.12.2021 № 3759-р: «Об утверждении стратегического направления в области цифровой трансформации науки и высшего образования». Режим доступа: <https://www.garant.ru/products/ipo/prime/doc/403203308/#1100>

[Decree of the Government of the Russian Federation No. 3759-r dated 21.12.2021: “On assessing the accuracy of determining the area in the field of digital transformation of science and higher education”. (In Russian.) Available at: <https://www.garant.ru/products/ipo/prime/doc/403203308/#1100>]

5. Программа «Робототехника: инженерно-технические кадры инновационной России». Режим доступа: <https://www.russianrobotics.ru/about-the-program/general-information/>

[Robotics: Engineering and technical personnel of innovative Russia. (In Russian.) Available at: <https://www.russianrobotics.ru/about-the-program/general-information/>]

6. Каталог цифрового оборудования для современного образования. Режим доступа: <https://robotbaza.ru/collection/all>

[Catalog of digital equipment for modern education. (In Russian.) Available at: <https://robotbaza.ru/collection/all>]

7. Международный стандарт ISO 8373: 2012 «Роботы и роботизированные устройства». Режим доступа: <https://www.iso.org/standard/55890.html>

[ISO 8373: 2012 Robots and robotic devices. (In Russian.) Available at: <https://www.iso.org/standard/55890.html>]

8. Абдулгалимов Г. Л. Использование эмулятора цифровой электроники в условиях дистанционного обучения. *Материалы Международной научно-практической конференции «Информатизация образования — 2021»*. Липецк: ЛГТУ; 2021:176–179. Режим доступа: <https://portalsga.ru/data/3573.pdf>

[Abdulgalimov G. L. Use of digital electronics emulators in the conditions of distance learning. *Informatization of Education — 2021. Proceeding International Scientific and Practical Conference*. Lipeck, LGTU; 2021:176–179. (In Russian.) Available at: <https://portalsga.ru/data/3573.pdf>]

9. Воронников С. А., Девятериков Е. А., Панфилов А. О. Техническое зрение роботов с использованием TrackingCam. М.: Applied Robotics. Электронная книга; 2017. 71 с.

[Vorotnikov S. A., Devyaterikov E. A., Panfilov A. O. Technical vision of robots using TrackingCam. М.: Applied Robotics. Electronic book; 2017. 71 p. (In Russian.)]

10. Прикладная робототехника. Учебные материалы. Режим доступа: <https://appliedrobotics.ru>

[Applied Robotics. Educational materials. (In Russian.) Available at: <https://appliedrobotics.ru/>]

11. Абдулгалимов Г. Л., Косино О. А., Субочева М. Л. Основы образовательной робототехники (на примере Ардуино). М.: Перо; 2018. 148 с.

[Abdulgalimov G. L., Kosino O. A., Subocheva M. L. Fundamentals of educational robotics (on the example of Arduino). Moscow, Pero; 2018. 148 p. (In Russian.)]

12. База знаний Амперки. Режим доступа: <http://wiki.amperka.ru/>

[Amperka's knowledge base. (In Russian.) Available at: <http://wiki.amperka.ru/>]

13. Липпман С. Б., Лажойе Ж., Му Б. Э. Язык программирования C++. Базовый курс. 5-е изд. М.: Вильямс; 2014. 1120 с.

[Lippman S. B., Lazhoye J., Mu B. E. C++ programming language. Basic course. 5th ed. Moscow, Williams; 2014. 1120 p. (In Russian.)]

14. Монк С. Программируем Arduino. Профессиональная работа со скетчами. СПб.: Питер; 2017. 252 с.

[Monk S. We program Arduino. Professional work with sketches. St. Petersburg, Piter, 2017. 252 p. (In Russian.)]

15. Forsyth D., Ponce J. Computer vision: A modern approach. Prentice Hall, August; 2002. 673 p.

16. Визильтер Ю. В., Желтов С. Ю., Бондаренко А. В., Ососков М. В., Моржин А. В. Обработка и анализ изображений в задачах машинного зрения. М.: Физматкнига; 2010. 672 с.

[Vizil'ter Yu. V., Zheltov S. Yu., Bondarenko A. V., Ososkov M. B., Morzhin A. V. Image processing and analysis in machine vision problems. Moscow, Fizmatkniga; 2010. 672 p. (In Russian.)]

17. Achanta R., Hemami S., Estrada F., Susstrunk S. Frequency-tuned salient region detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; 2009:1597–1604.

18. Kriegman D., Yang M. H., Ahuja N. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2002;24(1):34–58.

19. Szeliski R. Computer vision: Algorithms and applications. Springer; 2010. 925 p.

20. Chellappa R., Wilson C., Sirohey S. Human and machine recognition of faces: A survey. *Proceedings of the IEEE*. 1995;(83(5)):705–741.

#### Информация об авторах

Абдулгалимов Грамудин Латифович, доктор пед. наук, доцент, профессор кафедры технологических и информацион-

ных систем, Институт физики, технологии и информационных систем, Московский педагогический государственный университет, г. Москва, Россия; *ORCID*: <https://orcid.org/0000-0002-9744-0584>; *e-mail*: [agraml@mail.ru](mailto:agraml@mail.ru)

Косино Ольга Алексеевна, канд. пед. наук, доцент кафедры технологических и информационных систем, Институт физики, технологии и информационных систем, Московский педагогический государственный университет, г. Москва, Россия; *ORCID*: <https://orcid.org/0000-0001-7061-0750>; *e-mail*: [kosino-oa@yandex.ru](mailto:kosino-oa@yandex.ru)

Гоголданова Кермен Вячеславовна, старший преподаватель кафедры технологических и информационных систем, Институт физики, технологии и информационных систем, Московский педагогический государственный университет, г. Москва, Россия; *ORCID*: <https://orcid.org/0000-0001-6785-6514>; *e-mail*: [gkermen@mail.ru](mailto:gkermen@mail.ru)

#### Information about the authors

Gramudin L. Abdulgaliyev, Doctor of Sciences (Education), Docent, Professor at the Department of Technological and Information Systems, Institute of Physics, Technology, and Informational Systems, Moscow Pedagogical State University, Moscow, Russia; *ORCID*: <https://orcid.org/0000-0002-9744-0584>; *e-mail*: [agraml@mail.ru](mailto:agraml@mail.ru)

Olga A. Kosino, Candidate of Sciences (Education), Associate Professor at the Department of Technological and Information Systems, Institute of Physics, Technology, and Informational Systems, Moscow Pedagogical State University, Moscow, Russia; *ORCID*: <https://orcid.org/0000-0001-7061-0750>; *e-mail*: [kosino-oa@yandex.ru](mailto:kosino-oa@yandex.ru)

Kermen V. Gogoldanova, Senior Lecturer at the Department of Technological and Information Systems, Institute of Physics, Technology, and Informational Systems, Moscow Pedagogical State University, Russia; *ORCID*: <https://orcid.org/0000-0001-6785-6514>; *e-mail*: [gkermen@mail.ru](mailto:gkermen@mail.ru)

Поступила в редакцию / Received: 12.04.2022.

Поступила после рецензирования / Revised: 11.07.2022.

Принята к печати / Accepted: 02.08.2022.