

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт энергетики и электротехники  
(наименование института полностью)

Кафедра «Промышленная электроника»  
(наименование кафедры)

11.04.04 Электроника и нанoeлектроника  
(код и наименование направления подготовки, специальности)

Электронные приборы и устройства  
(направленность (профиль)/специализация)

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

на тему «Разработка метода обнаружения оставленных предметов в системах  
безопасности»

Студент Н.Н. Егорочкин  
(И.О. Фамилия) (личная подпись)

Научный руководитель к.т.н., доцент Е.С. Глибин  
(ученая степень, звание, И.О. Фамилия) (личная подпись)

Руководитель программы д.т.н., профессор В.В. Ивашин  
(ученая степень, звание, И.О. Фамилия) (личная подпись)

« \_\_\_\_ » \_\_\_\_\_ 2018г.

**Допустить к защите**

Заведующий кафедрой к.т.н., доцент А.А. Шевцов  
(ученая степень, звание, И.О. Фамилия) (личная подпись)

« \_\_\_\_ » \_\_\_\_\_ 2018г.

Тольятти 2018

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	5
1 История развития технического зрения.....	9
2 Обработка изображений .....	12
2.1 Разновидности изображений.....	12
2.2 Терминология компьютерного зрения.....	14
3 Детектор оставленных предметов .....	18
3.1 Метод поиска объектов переднего плана .....	19
3.2 Алгоритмы на основе трекинга .....	20
3.3 Сравнение алгоритмов.....	21
3.3.1 Работа в толпе.....	21
3.3.2 Ресурсоемкость.....	21
3.3.3 Пропуски оставленных предметов .....	22
3.3.4 Ложные срабатывания .....	22
3.4 Детектирование человека, который оставил предмет .....	23
3.5 Модификации поиска объектов переднего плана.....	23
3.6 Улучшение методики обновления фона. Обнаружение движения.....	24
3.7 Проверка на статичность .....	24
3.8 Критерии выбора .....	25
4 Выбор программного обеспечения.....	25
4.1 Преимущества MATLAB для компьютерного зрения .....	26
4.2 Недостатки MATLAB для компьютерного зрения:.....	28
4.3 Достоинства OpenCV (C++) для компьютерного зрения.....	28

4.4 Недостатки OpenCV (C++) для компьютерного зрения.....	29
4.5 Достоинства OpenCV (Python) для компьютерного зрения .....	30
4.6 Недостатки OpenCV (Python) для компьютерного зрения .....	30
5 Создание рабочей модели в MathLab.....	31
6 Выбор платы для управления детектором оставленных предметов.....	36
6.1 Что такое Raspberry Pi?.....	37
6.1.1 Преимущества Raspberry Pi.....	38
6.1.2 Недостатки Raspberry Pi .....	39
6.1.3 Когда использовать и не использовать Raspberry Pi .....	39
6.2 Что такое Arduino? .....	39
6.2.1 Преимущества Arduino .....	40
6.2.2 Недостатки Arduino.....	41
6.2.3 Когда использовать и не использовать Arduino.....	41
6.3 Сравнение Ардуино и Raspberry Pi .....	42
7 Работа с Raspberry Pi.....	43
7.1 Аппаратная часть.....	43
7.1.1 Блок питания для Raspberry Pi.....	47
7.1.2 Установка радиаторов на микрокомпьютер.....	48
7.1.3 Сборка корпуса для Raspberry Pi 3 .....	52
7.2 Установка и настройка Raspbian.....	53
7.2.1 Установка Raspbian .....	53
7.2.1.1 Шаг 1. Форматирование microSD-карты .....	54
7.2.1.2 Шаг 2. Запись дистрибутива Raspbian .....	54
7.2.1.3 Шаг 3. Первый запуск Raspbian на Raspberry Pi.....	55
7.2.2 Установка обновлений.....	56

7.2.3 Настройки локализации.....	57
7.2.4 Раскладка клавиатуры.....	58
7.3 Впечатления от использования Raspberry Pi 3.....	60
8 Подключение Raspberry Pi к MathLab Simulink.....	61
9 ЗАКЛЮЧЕНИЕ .....	69
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ .....	71

## **ВВЕДЕНИЕ**

Термин «компьютерное зрение» имеет множество связанных областей: машинное зрение, распознавание зрительных образов, анализ изображений и т.д.<sup>[1]</sup> Однако смысл, скрывающийся за всеми этими определениями, один -- это попытка научить компьютер видеть мир глазами человека, воспринимать его как человек и выполнять в связи с этим различные действия так же, как делал бы это человек, тем самым подменяя или полностью исключая последнего. Необходимость в этом возникает в ситуациях, связанных с риском для жизни, и бывает обусловлена особенностями человека как живого организма, которому свойственно быстро утомляться, пропускать через себя ограниченный объем информации, а также обрабатывать данные с относительно низкой скоростью. Процесс компьютерного зрения представляет собой сложную технологическую цепочку, включающую получение цифрового изображения, обработку изображения с целью выделения значимой информации на изображении и анализ этого предобработанного изображения для решения определенной задачи. Конечно, идеальным представляется создание универсальной самообучающейся системы, которая бы «росла» и «зрела» так же, как это с рождения происходит с любым человеком. Руководствуясь столь высокими целями, разработчики в области компьютерного зрения сегодня решают непростые задачи. Можно сказать, что область компьютерного зрения имеет недолгую по меркам фундаментальных наук, но очень бурную историю зарождения и развития.

Машинное (компьютерное) зрение — множество алгоритмов, технологий и методов, которые разработаны с целью классификации и детектирования объектов и отслеживания перемещения этих объектов в пространстве.

Компьютерное зрение относится к теории и технологии разработки программных систем с точки зрения научной дисциплины, получающие

информацию из различного вида изображений. Видеоданные-последовательно идущие друг за другом кадры (изображения), которые могут использоваться для обработки алгоритмами машинного зрения с определенными целями.

Машинное зрение стремится применять созданные человеком методы, технологии и алгоритмы с целью по созданию высокотехнологичных и высокоэффективных систем. Можно выделить следующие системы:

- Моделирования объектов или окружающей среды (топография и медицинские визуальные данные)
  - Управления процессами (автопилотируемый транспорт, промышленные установки);<sup>[3]</sup>
  - Организации информации (создания и сохранения баз данных с фото- и видеоизображениями);
  - Цифровой обработки фото- и видеоизображений
- Видеонаблюдения;
- Взаимодействия (управление машинами, опираясь на движения человека);
  - Дополнительной реальности;

В данной научно-исследовательской работе рассматривается решение задачи по обеспечению безопасности в местах массового скопления людей, применяя знания по программированию и разработке электронных устройств на базе технологий компьютерного зрения.

К сожалению, развитие систем безопасности не обеспечивает отсутствие террористических актов в том числе и на территории Российской Федерации. В 2017 году произошли две официально зарегистрированных трагедии:

3 апреля — теракт в Петербургском метрополитене

27 декабря — взрыв в магазине «Перекрёсток» в Санкт-Петербурге,

в результате которых погибли десятки людей, а сотни людей пострадали.<sup>[2]</sup>

Применение компьютерного зрения в данной области теоретически может снизить вероятность совершения преступлений с использованием взрывчатых веществ в местах массового скопления людей.

Реализация алгоритма поиска оставленных предметов и отслеживания людей, оставивших их,— задача, с которой может справиться лишь камера с соответствующей программной технологией.<sup>[4]</sup>

В рамках данной научно-исследовательской работы предполагается разработка аппаратного и программного обеспечения для детектирования оставленных предметов. Данная работа может использоваться в качестве примера решения следующих задач компьютерного зрения:

1. Детектирование объектов;
2. Распознавание объектов (оставленный предмет, человек и др.);
3. Исследование влияния различных отрицательных факторов на качество распознавания.

Объектом исследования является схема детектора оставленных предметов, выполненная в программном пакете MathLab Simulink.

Целью научно-исследовательской работы на тему «Разработка метода обнаружения оставленных предметов в системах безопасности» является создание алгоритма обнаружения оставленных предметов.

Задачами исследования являются:

- Изучение известных алгоритмов обнаружения оставленных предметов (ОП)
- Выявление основных проблем обнаружения ОП

**Программная реализация** алгоритма обнаружения оставленных предметов разработана в программном пакете «MathLab Simulink» с применением технологий компьютерного зрения блока ComputerVision ToolBox.. В результате реализован алгоритм распознавания оставленных предметов.

Результаты работы опубликованы в виде научных статей в сборнике «IV Международной научно-практической конференции (школа-семинар) молодых ученых» Тольятти: ТГУ, 2018

а также в сборнике Международной научно-практической конференция "Фундаментальные исследования основных направлений технических и физико-математических наук" от 1 июня 18 года в г. Челябинск.

Магистерская диссертация состоит из 8 глав, в которых решены упомянутые задачи. Общий объем работы составляет 74 страниц, включая 36 рисунков и 1 таблицу. Список используемой литературы включает 31 наименование, включая 12 источников на английском языке.

## **1 История развития технического зрения**

В пятидесятых годах двадцатого века научный мир был охвачен спорами о возможности создания разумной машины, которая могла бы воспринимать окружающий мир.<sup>[5]</sup> Первым шагом на пути к этому было опубликование в 1955 году статьи «Распознавание образов и современные компьютеры» профессором Массачусетского технологического института (MIT) Оливером Селфриджем (Oliver Selfridge). В статье была выдвинута идея, подразумевавшая снабдить компьютеры аудио и видеосистемами. Это, собственно, и привело к появлению термина «компьютерное (или техническое) зрение».

Специфика данного направления заключалась в том, что получаемое изображение это проекция трехмерного мира на плоскость. Это означает, что одного кадра недостаточно для получения всей необходимой информации об интересующем объекте и что нужны еще и дополнительные данные.

Для получения этих данных применялись следующие методы:

- Различные алгоритмы обработки полученных кадров
- Движение камеры

Большинство разработок на тот момент были теоретическими, так как на математический аппарат и техника еще не были развиты так, как это требовалось для экспериментов. Но, современные технологии компьютерного зрения, такие как зрение Asimo, основываются именно на них<sup>[6]</sup>.

В шестидесятые годы двадцатого века появились системы обработки изображений. Многие из этих систем были созданы для удаления помех с фотоснимков. Также появились исследования в области распознавания

печатных символов. Развитие данной технологии привело к негативному результату – появлению так называемых капчей.

Рост всего направления компьютерного зрения начался в тот момент, когда была разрешена проблематика нехватки систем обработки информации и достаточно мощных оптических систем. В те времена развитие систем технического зрения (СТЗ) проходило в основном рамках исследований космоса. Техническое зрение и в настоящее время используется в космосе. Примером является телескоп Хаббл.

Следующим шагом стало предположение аспиранта МИТ Лавренса Робертса, озвученное в семидесятые годы прошлого века, о том, что машина в состоянии построить трёхмерные образы объектов на основе из двухмерных изображений<sup>[7]</sup>. Схема, нарисованная Оливером Тебеулом, изображена на рисунке 1.

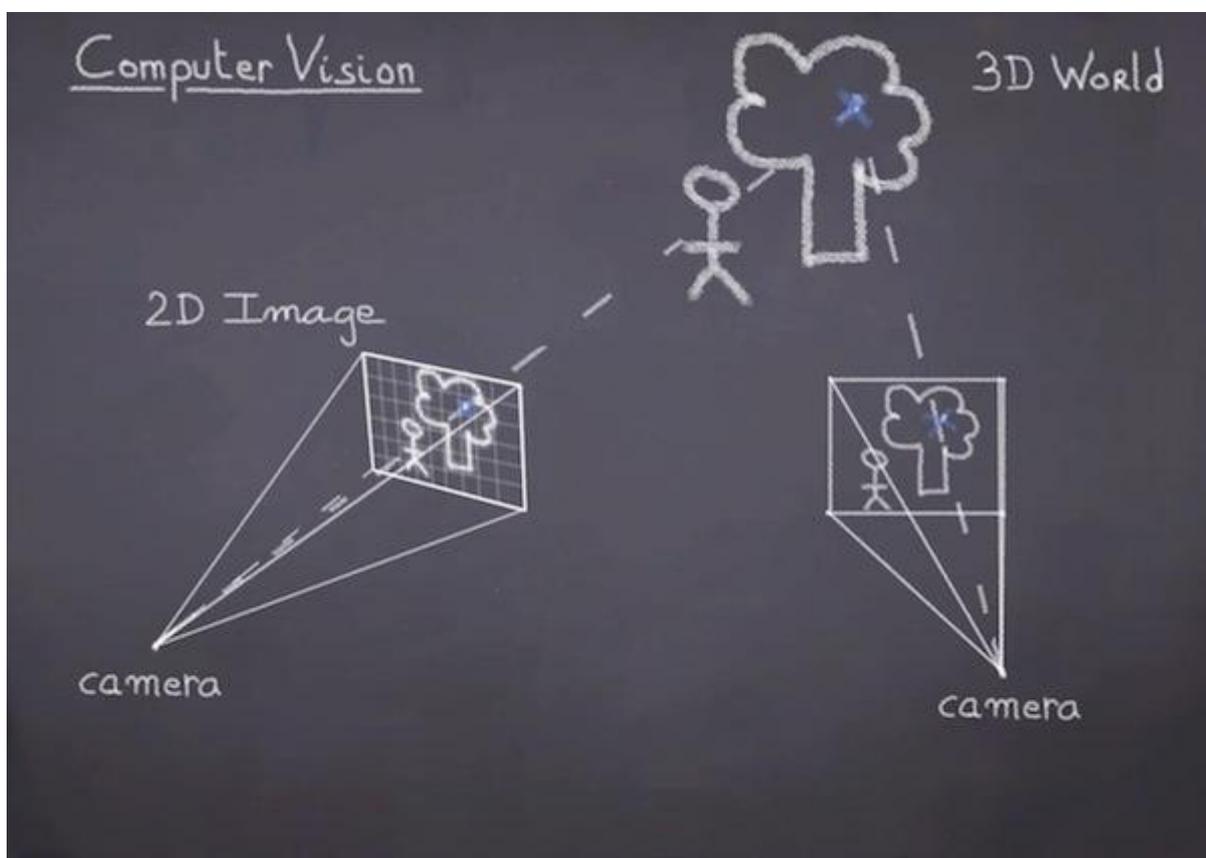


Рисунок 1-Схема, нарисованная Оливером Тебоул (Oliver Teboul)

Данное предположение привело к развитию систем анализа изображений. В дальнейшем появилась технология распознавания лиц в iPhone, фотоаппаратах и др.

В 1979 году труд профессора Гамбургского университета Ганса Гельмута Нагеля стал причиной появления теории анализа динамических сцен. Она позволяла распознавать динамичные объекты. В конце восьмидесятых появились первые роботы с системами технического зрения. Современных примером таких роботов являются робоняни.

В 1980-1990ых появилось новое поколение датчиков. Это способствовало значительному уменьшению цен на оборудование.

В начале 90-х годов двадцатого века появилась модульная парадигма. Нейробиолог Дэвид Кортни Марр занимался исследованием человеческого зрения. В результате он выделил четыре этапа обработки данных глазом:

1-Преобразование изображений;

2-Сегментация;

3-Выделение геометрической структуры;

4-Определение относительной структуры и семантики.

Еще он выделил три уровня обработки данных:

- Низший уровень, на котором происходит шумовая фильтрация
- Средний уровень, на котором происходит сегментация.

- Высший уровень, на котором происходит соединение объектов с общей картиной мира.

На основе данной теории стали разрабатываться алгоритмы обработки изображений. Это привело к появлению камер, которые способны отслеживать непосредственно перемещение объекта. В 2003 появились коммерческие системы по распознаванию лиц.

К настоящему времени СТЗ – это сформировавшийся раздел кибернетики. Разработке и изучению систем технического зрения посвящаются огромное количество конференций и симпозиумов ежегодно, не принимая во внимание сотни книг и статей, написанных на данную тему.

## **2 Обработка изображений**

### **2.1 Разновидности изображений**

Изображение, которое поступает на вход системы компьютерного зрения, может выступать в качестве любого пространственно-организованного массива измерений некоторых физических величин.<sup>[8]</sup> Если изображения формируются как результаты измерения интенсивности электромагнитного излучения в различных направлениях, то их можно разделить на следующие классы:

- инфракрасные изображения;
- оптические изображения;
- ультрафиолетовые изображения;
- изображения в радиодиапазоне.
- рентгеновские изображения;

Помимо этого, в точках изображений могут быть также записаны измерения и других величин (плотности, дальности и т.п.). Кроме этого изображения различаются объемом информации, который хранится в каждом из пикселей. Здесь выделяют:

- гипер- и мультиспектральные изображения;
- цветные изображения;
- полутоновые изображения;
- бинарные изображения;

Сами изображения также могут быть не только двумерные, но и объемные (характерно для томограмм). Также дополнительно выделяют видеоряды, являющиеся трехмерными массивами измерений. По способу освещения СТЗ разделяют на:

-пассивные, которые работают с изображениями, получаемыми в естественных условиях освещения

- активные, которые используют какой-либо генератор излучения (например, системы со структурированной подсветкой, лазерные дальномеры, радиолокационные системы и т.п.). Активность другого типа СТЗ может заключаться в автоматическом изменении фокусного расстояния, ракурса съемки или других параметров объектива. То есть, любые пространственно-организованные массивы данных можно трактовать как изображения, а это означает, что изображения в качестве источника информации встречается довольно часто. Наиболее часто встречаемые классы изображений, которые принадлежат различным областям, можно выделить:

- микроскопические изображения.
- биомедицинские изображения;

- аэрокосмические изображения;
- (наземные) изображения, полученные вне помещений;
- изображения, полученные внутри помещений;

## **2.2 Терминология компьютерного зрения**

Задачи анализа и компьютерной обработки изображений открывают широкие перспективы автоматизации многих сфер человеческой жизни<sup>[9]</sup>. Наиболее очевидным приложением является придание роботам осязания. Действительно, трудно представить робота, который способен адаптироваться к изменениям ситуации без органов чувств. Адаптивные роботы применяются при исследовании опасных мест на Земле и других планет, в быту, в производстве при сборке изделий. В качестве других приложений можно выделить навигацию беспилотных летательных аппаратов, контроль качества выпускаемой продукции, системы безопасности, биометрии, системы поиска изображений в БД, распознавание и сопровождение целей и т. п.

Разрабатываемые в настоящее время СТЗ предназначены для решения каких-либо конкретных задач и ориентированы на один из перечисленных выше типов изображений и осуществляют работу в ограниченных предметных областях.

Построение СТЗ общего назначения на данный момент является неразрешимой задачей и рассматривается лишь как некая конечная цель исследований в данной области. Если искусственный интеллект (ИИ) трактовать как область у которой в качестве цели является снабжение компьютеров способностями по обработке информации, которые имеют живые организмы, то в качестве одной из подцелей является наделение машин способностью по обработке сенсорной информации. В данном случае компьютерное зрение рассматривается как раздел искусственного

интеллекта. Вполне вероятно, что построение СТЗ общего назначения невозможно без разрешения проблемы ИИ.

Существует ряд терминов, так или иначе относящихся к выполнению операций над изображениями с помощью компьютеров:

- компьютерное зрение (а также техническое зрение, машинное зрение, зрение роботов);
- анализ изображений;
- обработка изображений;
- понимание изображений;
- интерпретация изображения;
- иконика;
- компьютерная графика;
- распознавание образов;
- когнитивная графика.

Компьютерное, техническое и машинное зрение можно считать почти эквивалентными терминами, но все остальные термины обозначают различные понятия. Действительно, все они связаны с преобразованием изобразительной информации, но отличаются тем, что подается на выход, а что – на вход. В случае обработки изображений как выходной, так и входной информацией является изображение. Задачами обработки изображений является улучшение контраста, подчеркивание границ, подавления шумов, реставрация изображений и т.п.

Под интерпретацией изображений понимается преобразование изображения в символьную форму с извлечением какой-либо значимой

информации. К интерпретации изображений относятся задачи построения структурных описаний изображений, текстурного анализа, построения структурных описаний изображений. Анализ изображений включает такие задачи, как выявление изменений по серии снимков, нахождение взаимного пространственного преобразования двух изображений одной и той же сцены, снятой с разных ракурсов, и т.п. Под пониманием изображений подразумевается их преобразование к символьной форме, однако тут на выходе формируются семантические описания изображений.<sup>[10]</sup>

Иконика это общая наука об изображениях. К ней относятся как, вопросы улучшения изображений, так и вопросы анализа изображений.

Когнитивная и компьютерная графика связаны с обратным преобразованием: от символьных представлений к изображениям.

Распознавание образов- отдельная научная дисциплина, которая лишь косвенно связана с изображениями. В ней под образом понимается произвольное описание объекта.

Компьютерное зрение связано с пониманием и анализом изображений, но имеет определенную специфику. Особенностью задач компьютерного зрения является то, что в них на основе изображений необходимо восстановить информацию об объектах, которая имеет физический смысл, в связи с чем требуется использование детальной информации о процессах распространения излучения и формирования изображений оптическими системами.

В методах анализа изображений описания строятся как не связанные с физическими величинами, а сами методы могут применяются для изображений разной природы. Типичными задачами компьютерного зрения являются

- определение взаимных перемещений приемника излучения и объектов сцены;
- построения карт отражательной способности по видимой яркости поверхностей;
- восстановления трехмерной информации об объектах методами
  - структурированной подсветки,
  - стереозрения,
  - фотометрического стерео,
  - формы из движения,
  - формы из затенения,
  - глубины из дефокусировки;
  - распознавания объектов.

Как ранее отмечалось, изображения могут обладать большой изменчивостью. Смена условий освещения, ракурса съемки, взаимные перемещения объектов с частичным загороживанием приводят к достаточно сложным преобразованиям в яркостях пикселей при неизменчивом содержании изображения, что, собственно, и называется изменчивостью изображений. В итоге, отдельные пиксели изображения и даже большие области могут оказаться недостаточными для разрешения задач зрения. В качестве примера, цвет определенного объекта существенно может меняться в зависимости от освещения. Темно-серый лист при достаточном ярком освещении будет более ярким, чем белый лист при тусклом. Однако, человек без труда определяет цвета в независимости от освещения, что является константностью восприятия. Освещение зачастую является сложным: может быть не один источник света, часть объектов могут находиться на свету, а

часть – в тени. Все становится куда сложнее, когда встает вопрос о восстановлении трехмерной информации, не содержащаяся в явном виде в изображении. Семантическое описание сцен и распознавание объектов по изображениям является самой сложной задачей в связи со сложной внутренней структурой объектов и их большим разнообразием и пространственными отношениями.

Изображения не только имеют свойства изменчивости, но обладают неоднозначностью их интерпретации в условиях высокой неопределенности, которая означает, что возможное содержание изображения неизвестно заранее, и на нем могут присутствовать различные объекты. Поскольку начальные эксперименты в области компьютерного зрения по распознаванию объектов путем сравнения их изображений с эталонными, не смогли позволить получить универсальных решений, было решено обратиться к психофизическим и нейрофизиологическим исследованиям зрительной системы животных и человека, умеющие указанные проблемы решать.

### **3 Детектор оставленных предметов**

Программные комплексы современности предоставляют объемный спектр инструментов для необходимого контроля различных объектов<sup>[11]</sup>. С целью обеспечения безопасности от терроризма во многих видеосистемах, расположенных в общественных местах (безопасные города, банки, торговые рынки, центры, аэропорты, вокзалы, метро) является популярным детектор оставленных предметов. Он дает возможность регистрировать объекты, находившиеся в движении, а затем остановившиеся на определенный, заданный ранее, промежуток времени

Выделяют, в основном, два класса алгоритмов, которые позволяют решить задачу по обнаружению оставленных предметов:

-алгоритмы, которые основаны на поиске объектов переднего плана;

-алгоритмы, которые основаны на трекинге, то есть на слежении за подвижными объектами.

### **3.1 Метод поиска объектов переднего плана**

В этом методе используются две модели фона: долгосрочную (медленно меняющуюся, долгую) и краткосрочную (меняющуюся быстро, быструю), и каждая модель имеет свои время обновления и скорость.<sup>[12]</sup>

Время обновления быстро меняющейся модели фона мало, в ней все изменения текущего на данный момент кадра по сравнению с предшествующим быстро фиксируются в изменяющемся краткосрочном фоне. То есть, если в кадре появился новый движущийся объект, то он не станет фоном, в случае если объект стал неподвижным (например искомый оставленный предмет), то он совместится с неподвижным фоном кадра. Другими словами, краткосрочный фон дает возможность определить, что в кадре находился движущийся объект (присутствовало движение), а затем это движение прекратилось (объект стал статичным). Использование исключительно краткосрочной модели не представляется возможным, так как статичный объект (оставленный предмет) достаточно быстро "пропадает", сливается с фоном, и вычислить время его покоя не представляется возможным. В связи с этим одновременно с краткосрочной моделью алгоритм поиска переднего плана подразумевает и долгосрочную модель фона. Данная модель меняется значительно медленнее, новые объекты, попавшие в нее, еще какое-то время, заданное программно, остаются передним планом и лишь затем становятся фоном. Долгосрочную модель используют для того, чтобы обнаружить оставленный объект, произвести оценку, что он попал в сцену какое-то время назад, а не являлся до этого исходным фоном.

Оператор системы слежения имеет возможность задать время, спустя которое предмет, появившийся в сцене, будет считаться оставленным. После

получения этих данных, система вычисляет коэффициент, который, собственно, и скорректирует время обновления кадров долгосрочного фона.

Логика работы данного метода изображена на рис. 2.



Рисунок 2- Логика работы детектора оставленных предметов, основанная на методе поиска объектов переднего плана

Каждый пиксель текущего кадра сравнивается с соответствующими пикселями краткосрочного и долгосрочного фонов. Затем, в зависимости от результата произведенного сравнения, строится гипотеза о появлении в кадре оставленного предмета.

### 3.2 Алгоритмы на основе трекинга

При использовании алгоритмов на основе трекинга для детектирования оставленных предметов система обнаруживает как оставленный предмет, так и человека, оставившего его, а точнее, система обнаруживает разделение одного исходного объекта, находившегося в движении, на два: динамичный и статичный, далее следит за объектом, прекратившим движение.<sup>[13]</sup>

В подходе с использованием трекинга первым шагом также является выделение фона. Затем система выстраивает траектории всех динамичных

объектов в кадре, далее происходит фиксация "разделения" одного подвижного объекта на два. Программа производит анализ скорости и направления движения объектов, их цветовые характеристики, далее эти данные позволяют сопоставить предмет и человека при их "разделении" на два разных объекта.

Существующие на сегодняшний день детекторы оставленных предметов преимущественно используют в своей основе оба метода: алгоритмы, которые осуществляют поиск объектов переднего плана, и метод, в основе которого заключена работа алгоритмов трекинга (построения траекторий перемещения динамичных объектов).

### **3.3 Сравнение алгоритмов**

Каждый из алгоритмов (методов) имеет как определенные преимущества, так и недостатки.

#### **3.3.1 Работа в толпе**

Алгоритмы, которые основаны на поиске объектов переднего плана, показывают хорошие результаты по корректной установке истинного фона. Это возможно в тот момент, когда фон находится в кадре полностью на определенное время, то есть фон не на все время съемки закрывается толпой. Метода, который использует алгоритмы трекинга, испытывает определенное сложности при работе в толпе с выделением отдельного движущегося объекта.

#### **3.3.2 Ресурсоемкость**

Трудность процессов обработки информации и, соответственно, ресурсоемкость метода на основе трекинга выше. Система определяет фон, обнаруживает динамичные объекты, выделяет цветовые характеристики объекта движения, его направление движения и скорость, рассчитывает вероятность перехода движущегося объекта из одной точки

предшествующего кадра в другую точку текущего кадра для правильного присвоения траектории перемещения при движении в кадре одновременно нескольких объектов. Для проведения перечисленных подсчетов необходимы большие вычислительные мощности.



Рисунок 3- Обнаружение оставленного предмета и его владельца

### **3.3.3 Пропуски оставленных предметов**

Для метода поиска объектов переднего плана, трудности могут возникнуть при невозможности верного определения фона (при работе в условиях изменения освещенности и при работе в толпе). Метод, который основывается на трекинге, имеет трудности при работе со сценами с высокой интенсивностью движения, с перекрытиями объектов движения другими движущимися объектами, с потерей объекта на некоторое время из поля зрения камеры.

### **3.3.4 Ложные срабатывания**

Для обоих рассматриваемых методов ложные срабатывания возможны в случае задержки человека в кадре на время, превышающее или равное времени, после которого человек будет считаться оставленным предметом. Это вызвано тем, что система не имеет возможности отличить человека от потенциально интересного объекта. С данной трудностью можно бороться с помощью объединения детектора оставленных предметов с

классификатором, который способен распознавать различные категории объектов и может отличать человека в независимости от его положения от предметов. Ниже, на рисунке 4, приведена сводная таблица сравнительной характеристики методов.

Поиск объектов переднего плана vs. алгоритм, использующий трекинг

	Алгоритмы, основанные на поиске объектов переднего плана	Алгоритмы, основанные на трекинге объектов
Работа в толпе	Высокий уровень детектирования достигается при корректной установке истинного фона	Возникают трудности с выделением отдельного движущегося объекта
Сложность процессов обработки	Средняя	Высокая
Ресурсоемкость	Средняя	Высокая
Сложные ситуации, в которых высока погрешность работы метода	При условиях невозможности выделить истинный фон	При условиях высокой интенсивности движения объектов в кадре, временной потере видимости объекта при его пересечении с другими объектами
Ложные срабатывания	При задержке человека в кадре; решается объединением с классификатором	При задержке человека в кадре; решается объединением с классификатором
Обнаружение человека, который принес объект	Невозможно	Возможно

Рисунок 4- Сравнительная таблица двух алгоритмов детектирования оставленных предметов

### 3.4 Детектирование человека, который оставил предмет

Метод, основанный на трекинге позволяет получить данные о всех перемещениях человека, оставившего предмет, в поле зрения камеры. Таким образом, оператор системы имеет возможность нахождения человека, принесшего оставленный предмет, и затем использовать другие интеллектуальные функции ("перехват" или межкамерный трекинг) для дальнейшего его отслеживания.

### 3.5 Модификации поиска объектов переднего плана

Сравнительный анализ двух методов показывает, что менее ресурсоемким и наиболее устойчивым является алгоритм, который использует поиск объектов переднего плана, но в то же время он нестабилен при изменении освещенности и в условиях высокой интенсивности

движения. Для минимизирования влияния изменений условий съемки, разработчиками программных продуктов были реализованы описанные ниже преобразования.<sup>[13]</sup>

### **3.6 Улучшение методики обновления фона. Обнаружение движения**

Использование детектора движения оказывает положительное влияние на качество работы алгоритма в нестабильных условиях съемки (изменение освещенности). По итогам работы детектора движения алгоритм выделяет рамку динамического объекта, после создает маски из рамок всех подвижных объектов кадра. Где в областях кадра нет подвижных объектов. Там происходит обновление фона, где присутствуют маски подвижных объектов- обновление фона не происходит.

Для обеспечения антитеррористической безопасности в подавляющем большинстве видеосистем, которые размещены в общественных местах, детектор оставленных предметов пользуется большой популярностью.

### **3.7 Проверка на статичность**

С целью уменьшения ложных срабатываний и увеличения точности работы детектора используется дополнительная проверка объекта на статичность. Эта модификация исключает временно статичные предметы, которые не представляют интерес. Для проверки статичности заданная программа выделяет рамку возможного оставленного предмета и запоминает его положение. Затем при поступлении нового кадра снова выделяется рамка статичного предмета, координаты рамки сравниваются со значениями предыдущих запомненных координат. В случае если после сравнения получен положительный результат, (координаты рамок совпали), то затем проверка повторяется еще некоторое количество раз (число проверок зависит от заданного времени обнаружения оставленного предмета). Если же после

сравнения результат отрицателен, то счетчик обнуляется и тем самым сохраненная ранее рамка удаляется.

### **3.8 Критерии выбора**

Существующие на сегодняшний день детекторы оставленных предметов используют преимущественно одновременно два метода: в основе одного лежит использование алгоритмов трекинга (построения траекторий перемещения динамических объектов), в основе другого лежат алгоритмы, которые осуществляют поиск объектов переднего плана. Каждый из методов имеет свои условия применения и особенности.

Основным преимуществом детектора, использующего трекинг, является возможность обнаружения непосредственно человека, оставившего предмет, но этот метод требует высоких значений производительности системы, дает сбои при наблюдении за сценами с высокой интенсивностью движения. В свою очередь, метод поиска объектов переднего плана достигает высокой точности обнаружения оставленных предметов при условии правильной установки исходного начального фона, это означает, что, если система зафиксирует чистый фон сцены на некоторое время (в зависимости от производительности программного обеспечения) детектор будет позволять получать высокие результаты обнаружения оставленных предметов. Кроме того, детектор оставленных предметов, основанный на этом методе, обладает достаточно низкой ресурсоемкостью, а модификации, разработанные для этого метода увеличивают точность детектирования и уменьшают число ложных срабатываний.

## **4 Выбор программного обеспечения**

В компьютерном зрении мы сталкиваемся с выбором языка программирования. Буквально десять лет назад не было хороших библиотек для компьютерного зрения.<sup>[14]</sup> Если вы хотели учиться, вы просто брали книгу и начинали кодировать собственную мини-библиотеку алгоритмов

компьютерного зрения. К счастью, сейчас все намного лучше. На сегодняшний день у нас есть несколько вариантов выбора. Имеется выбор из трех инструментариев по компьютерному зрению - OpenCV Python, OpenCV C ++, или MATLAB?

#### **4.1 Преимущества MATLAB для компьютерного зрения**

Относительно недавно Computer Vision была исследовательской областью в зачаточном состоянии. Люди, которые работали в Computer Vision, были в основном в академических или исследовательских лабораториях. Их инструментом выбора был MATLAB, и в течение долгого времени OpenCV не мог предложить то, что предоставлял MATLAB. Вот некоторые из преимуществ использования MATLAB.

Причины использования MATLAB для компьютерного зрения:

Матричная библиотека

Первой причиной является мощная матричная библиотека. Мы имеем возможность рассматривать изображение не только как многомерную матрицу в компьютерном зрении, а еще и используем большую порцию линейной алгебры в различных алгоритмах. Процедуры линейной алгебры MATLAB очень мощные и невероятно быстрые (при правильном использовании). Как только у нас появляется необходимость решить большую разреженную линейную систему как часть алгоритма, то в MATLAB это является кодом одной строки в MATLAB - что-то вроде  $X = A \setminus b$ . Таким образом, в MATLAB один оператор, такой как обратная косая черта ( $\setminus$ ), представляет собой набор мощных алгоритмов, которые MATLAB соответствующим образом выбирает для вас.

Toolboxes:

Что бы вы ни делали, для этого есть набор инструментов. В Matlab есть инструментарий для обработки изображений, панель инструментов для

компьютерного зрения и статистическое и машинное обучение, обеспечивающее реализацию самых разнообразных очень полезных алгоритмов. Функции обычно обеспечивают чистый и понятный интерфейс. Многие проблемы с компьютерным видением часто устанавливаются как проблемы оптимизации. Вы пытаетесь максимизировать или минимизировать некоторые целевые функции при определенных ограничениях. Инструментарий оптимизации в MATLAB обеспечивает отличные реализации многих алгоритмов.

Инструменты визуализации и отладки:

Одно из преимуществ использования MATLAB заключается в том, что написание кода, визуализация результатов и отладка происходит в одной интегрированной среде.

Работа с OpenCV:

Вы можете взаимодействовать с OpenCV с использованием интерфейса OpenCV MATLAB.

Отличная документация:

MATLAB поставляется с отличной документацией и примерами, которые легко доступны в среде IDE. Большая документация позволяет сделать кодирование в MATLAB в два раза быстрее, чем использование OpenCV. Функции MATLAB также лучше разработаны по сравнению с OpenCV.

Большое исследовательское сообщество:

MATLAB чрезвычайно популярен в академических кругах. Последние исследовательские демонстрации часто используют код MATLAB.

## **4.2 Недостатки MATLAB для компьютерного зрения:**

Стоимость лицензионной версии: <sup>[15]</sup>

лицензия MATLAB очень дорога.

Кривая обучения:

MATLAB - это матричный движок. Существует способ MATLAB для написания кода, который отличается от языков программирования общего назначения, таких как C ++ или Python.

Более медленное время выполнения:

Типичная программа MATLAB выполняется много раз медленнее, чем программа на C ++. Встроенные процедуры MATLAB могут быть очень быстрыми, но код, который вы пишете в MATLAB, обычно выполняется намного медленнее. Часто люди в конечном итоге кодируют вычислительно интенсивные части в C и интегрируют их с кодом MATLAB, используя mex

## **4.3 Достоинства OpenCV (C ++)** для компьютерного зрения

Стоимость программы:

Большая часть OpenCV бесплатна. Вы можете свободно использовать OpenCV в своем коммерческом приложении, и при необходимости можете просмотреть исходные коды и исправить проблемы. У вас нет необходимости открывать исходный код вашего проекта, если вы используете OpenCV.

Огромная оптимизированная библиотека:

Коллекция алгоритмов, доступных в OpenCV, затмевает все. Библиотека также оптимизирована для производительности. С помощью OpenCV 3 вы можете использовать Transparent API для простого

использования совместимых с OpenCL устройств (например, GPU) на вашем компьютере. Многие алгоритмы имеют реализацию CUDA.

Платформы и устройства:

Вы можете использовать OpenCV в своем настольном приложении или в качестве бэкэнд вашего веб-приложения. Из-за его сосредоточенности на производительности OpenCV (C / C++) является видной библиотекой по выбору во многих приложениях со встроенным зрением и мобильных приложениях.

Большое сообщество:

Существует большое сообщество разработчиков, которые используют и поддерживают OpenCV. Он был загружен более 9 миллионов раз. В отличие от сообщества MATLAB, состоящего из исследователей, сообщество OpenCV представляет собой смесь людей из многих областей и отраслей. Разработка OpenCV финансируется такими компаниями, как Intel, AMD и Google.

#### **4.4 Недостатки OpenCV (C++) для компьютерного зрения**

Трудность для начинающих:<sup>[16]</sup>

Если у вас нет опыта программирования на C++, использование OpenCV (C++) будет сложным.

Слабая документация:

Документация у OpenCV плохая. Зачастую вам необходимо хорошо понимать алгоритм и фактически читать статью, потому что документация не всегда объясняет, что означают параметры и как они влияют на результат. Документация не всегда поставляется с примером кода, и это затрудняет понимание.

Маленькая компьютерная библиотека обучения:

Инженерам по компьютерному видению часто требуются многие процедуры машинного обучения. OpenCV имеет небольшой набор алгоритмов машинного обучения.

Визуализация и отладка:

Отладка и визуализация сложны в любой среде C ++. Это особенно верно, если вы придумываете новый алгоритм с нуля.

#### **4.5 Достоинства OpenCV (Python) для компьютерного зрения**

Простота использования:

Если вы являетесь программистом на языке python, использование OpenCV (Python) было бы очень простым. Python - простой язык для изучения (особенно по сравнению с C ++). Это также отличный первый язык для изучения.

Python с OpenCV, numpy, scipy, scikit-learn и matplotlib Python предоставляет мощную среду для обучения и экспериментов с Computer Vision и Machine Learning.

Визуализация и отладка:

При использовании OpenCV (Python) у вас есть доступ к огромному количеству библиотек, написанных для Python. Визуализация с использованием matplotlib примерно такая же, как MATLAB. Отладка кода в Python проще, чем на C ++, но уступает «супер-легкости» MATLAB.

#### **4.6 Недостатки OpenCV (Python) для компьютерного зрения**

Слабая документация.

Как было написано выше документация OpenCV (C++) не очень хороша. Но документация на python гораздо хуже. Новичку приходится угадывать, как использовать определенные функции.

Отсутствие поддержки.

Компании, которые поддерживают OpenCV (Intel, AMD, NVidia и т. д.), не обращают внимание в борьбе, когда дело доходит до версии OpenCV на C++. Им необходимо, чтобы вы использовали OpenCV и покупали их оборудование (процессоры / графические процессоры и т. д.) для запуска этих алгоритмов.

Медленное время работы:

По сравнению с C++ ваши программы на Python, как правило, работают медленнее.

В результате сравнения достоинств и недостатков рассмотренных выше программных обеспечений мною был выбран программный пакет Matlab.

## **5 Создание рабочей модели в MathLab**

В ряде случаев, для предотвращения террористических актов на станциях метро, автовокзалах недостаточно лишь наличия персонала по обеспечению безопасности, так как человеческий фактор в данном случае играет решающую роль.<sup>[17]</sup>

Создание электронного детектора оставленных предметов позволит сократить вероятность происхождения катастрофы с человеческими жертвами, поскольку будет происходить круглосуточный мониторинг интересующих областей.

Модель детектора оставленных предметов, выполненная в программном пакете MathLab Simulink, приведена ниже на рисунке 1.

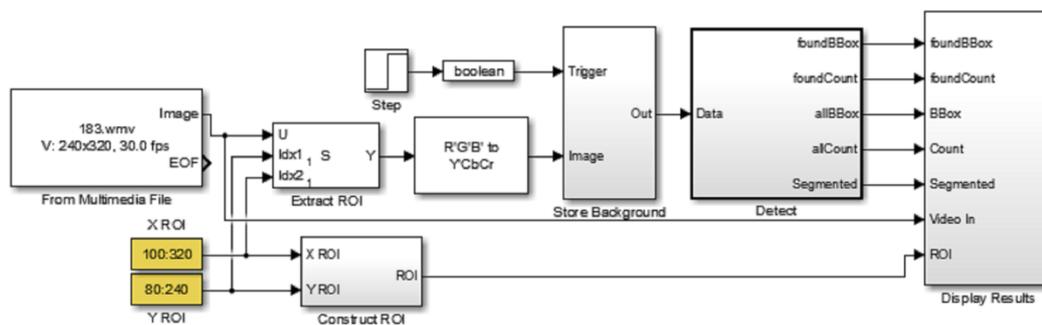


Рисунок 5- Модель детектора оставленных предметов.

В данной модели в качестве исходного материала было использовано видео, полученное при помощи встроенный в ноутбук веб-камеры HP Webcam-101. Подробнее опишем элементы схемы.

Блок From Multimedia File является входным блоком детектора, куда заносится ранее захваченный видеофрагмент. Блоки X ROI и Y ROI задают интересующую область видеофрагмента, данные из этой области поступают на блок Construct ROI, с которого информация передается на блок Display Results, который выводит видеофрагмент с выделенной областью в окно изображений.

Информация с блока From Multimedia File и X ROI и Y ROI поступает на блок Extract ROI, откуда извлекается необходимая информация из видеофрагмента, которая затем приходит на RGB-фильтр. С помощью блоков Step и Boolean и блока Store Background выделяются кадры из заданного видеофрагмента с определенной частотой. С блока Store Background данные поступают на блок Detect (подробное описание блока приведено ниже), с которого вся необходимая информация приходит на дисплей.

Внутренняя структура блока Store Background приведена на рисунке 6.

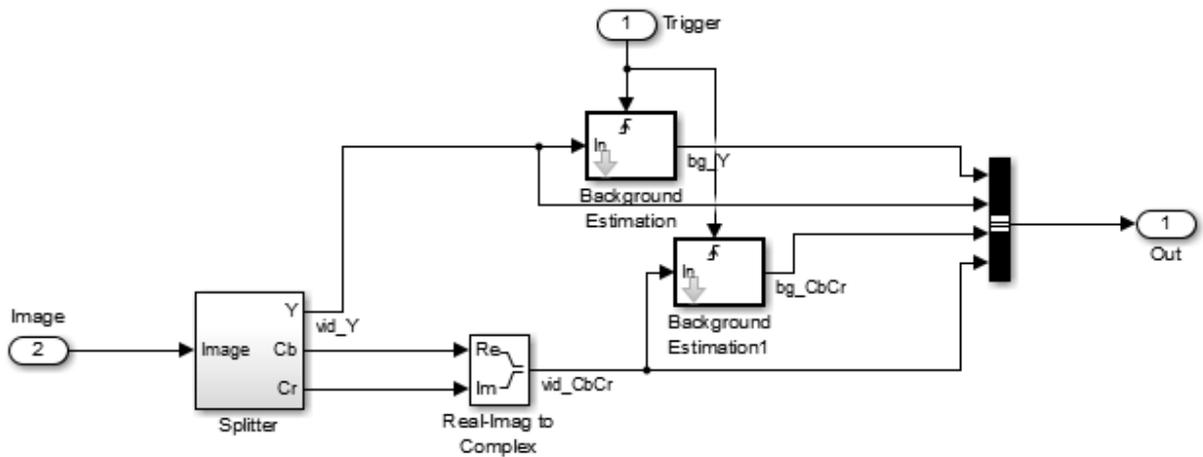


Рисунок 6- Внутренняя структура блока Store Background

Блок Splitter разделяет видефрагмент на составляющие, мнимая и действительная часть которых объединяется в блоке Real-Imag to Complex для дальнейшего анализа. С помощью Background Estimation и Background Estimation1 происходит сравнение текущего кадра с предыдущим, затем необходимые впоследствии данные через мультиплексор поступают на выход out. Затем информация поступает на блок Detect, внутренняя структура которого представлена на рисунке 7.

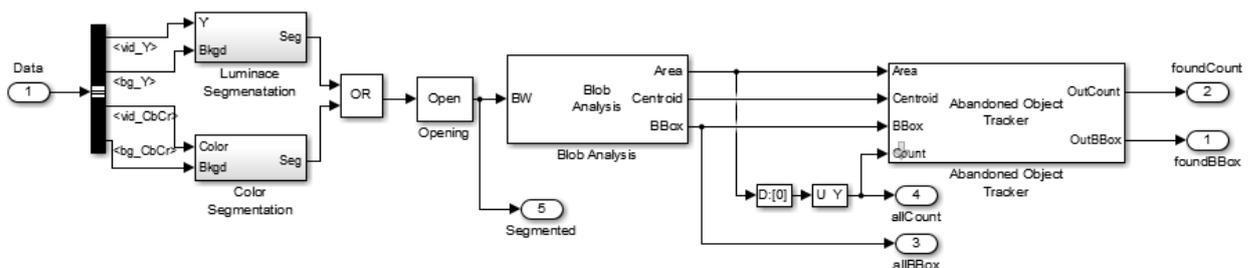


Рисунок 7- Внутренняя структура блока Detect

Далее опишем принцип работы модели, представленной на рисунке 5. Для захвата видефрагмента было использовано встроенное в программный пакет MathLab приложение Image Acquisition Toolbox. Скрин окна данного приложения приведен на рисунке 8.

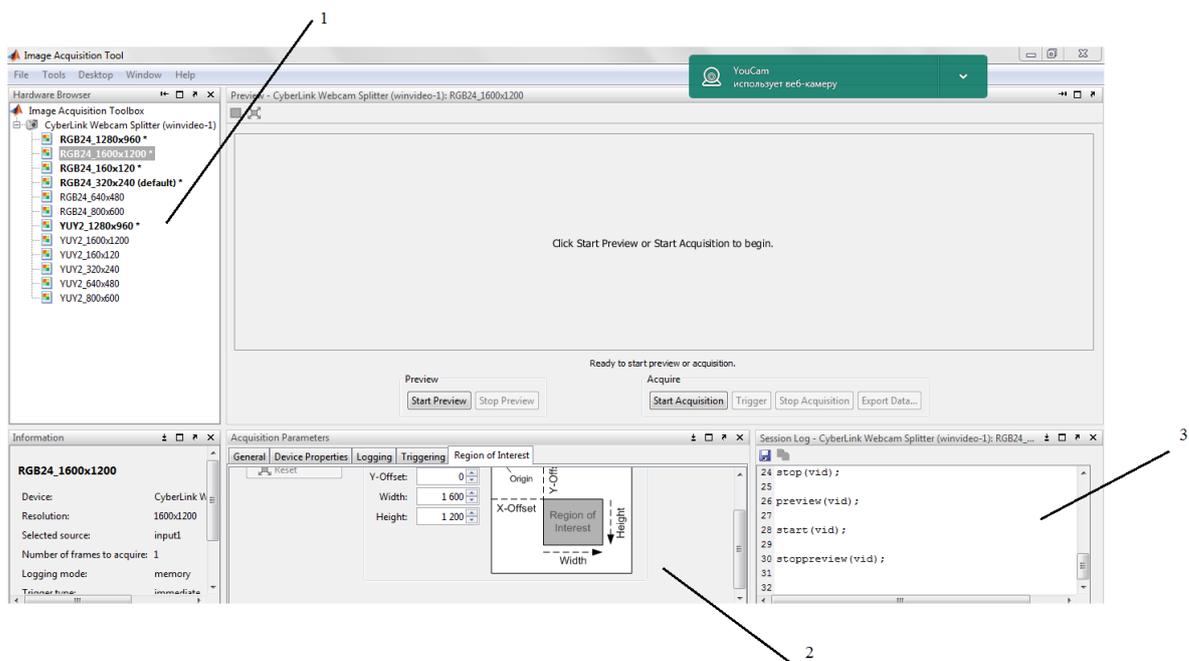


Рисунок 8- Скрин окна приложения Image Acquisition Toolbox

Данное приложение позволяет выбрать необходимое разрешение, получаемого видеофрагмента в окне Hardware Browser (область 1), выделить интересующую разработчика область в блоке Region of Interest (область 2), а также получить программный код на языке C++, для возможного дальнейшего использования в других программных пакетах (область 3).

В результате захвата видео и запуска модели, изображенной на рисунке 5 были получены следующие результаты, приведенные ниже на рисунке 9.

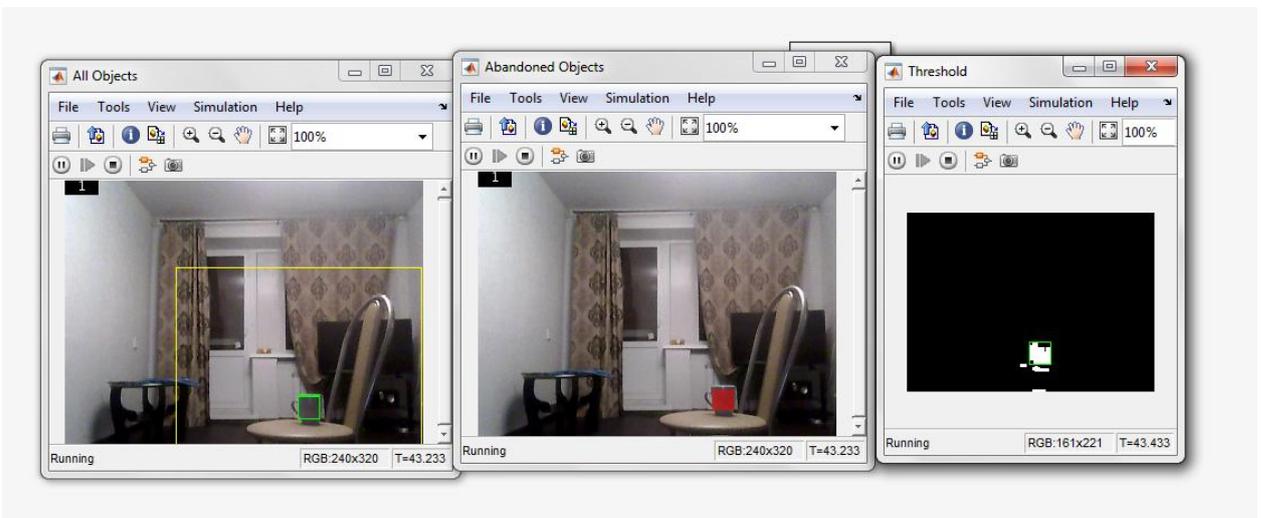
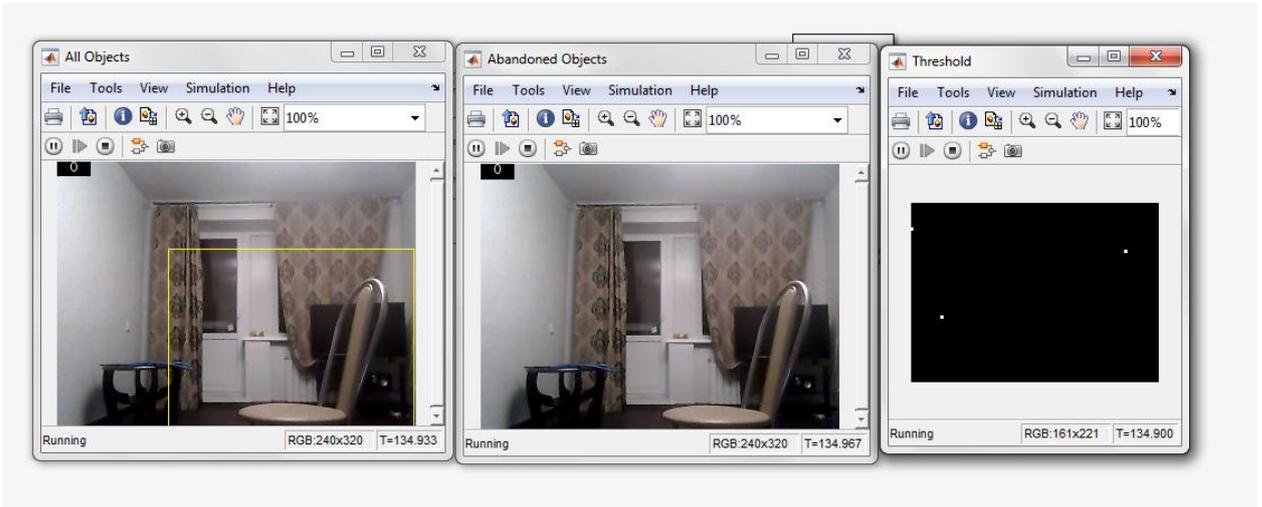


Рисунок 9- Результат работы модели детектора оставленных предметов.

На верхней части рисунка 9 изображено начальное состояние исследуемой области, на нижней части- результат обнаружения появившегося в интересующей области объекта. Расположение интересующей области выбирается самостоятельно, и может захватывать как маленькую область (с наиболее вероятным местом оставления предметов), так и весь кадр, если вся область захвата изображения представляет возможную опасность. На рисунке 4 представлено непосредственно появление объекта в зоне обнаружения. Как видно из рисунка в захватываемой области выделяются все появившиеся объекты, затем посредством вычитания фона выделенными остаются только стационарные, то есть прекратившие движение после появления объекты (рисунок 10).

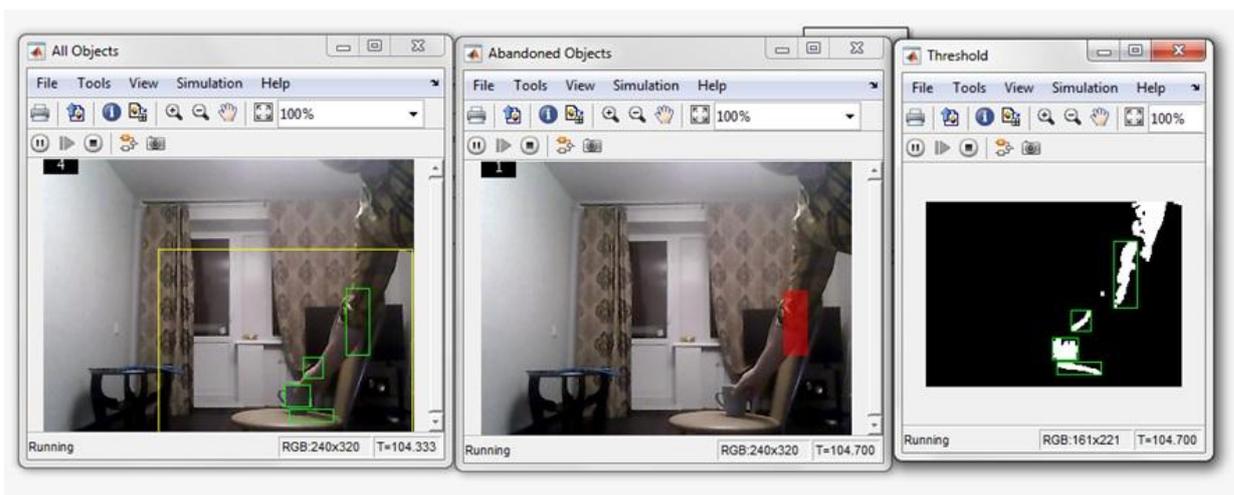


Рисунок 10- Появление нового объекта в исследуемой зоне

## **6 Выбор платы для управления детектором оставленных предметов**

Arduino и Raspberry Pi - самые популярные устройства среди студентов-инженеров, любителей и профессионалов.<sup>[18]</sup> Опытные и профессионалы знают различия и полезность между Arduino и Raspberry Pi. Но как новички, так и студенты просто путаются между ними, например, какую панель легко учиться или какую доску использовать для своего проекта или почему они должны использовать Arduino или Raspberry Pi и наоборот. Рассмотрим все функции, которые могут облегчить выбор решения по Arduino и Raspberry Pi.

### **Различия между Arduino и Raspberry Pi**

Различия между Arduino и Raspberry Pi в основном включают то, что такое Raspberry Pi, Arduino, преимущества, недостатки и различия между ними. Две платформы изображены на рисунке.



Рисунок 11- Arduino и Raspberry Pi

### **6.1 Что такое Raspberry Pi?**

Платформа Raspberry Pi - полностью функциональный компьютер или полноценная кредитная карта. Она имеет все атрибуты компьютера с выделенной памятью, процессором и графической картой для вывода через HDMI. Она даже запускает специально разработанную версию операционной системы Linux, и ее легко установить в большинстве программ Linux, и с небольшим усилием использовал Raspberry Pi как функционирующий эмулятор видеоигр или медиа-стример. Отдельно Raspberry Pi изображена на рисунке.



Рисунок 12- Raspberry Pi

Несмотря на то, что Raspberry Pi не предлагает встроенное хранилище, мы можем использовать SD-карты в качестве флэш-памяти в общей системе, позволяя вам быстро менять различные версии операционной системы или обновления программного обеспечения для отладки. Поскольку это устройство является независимым сетевым подключением, вы также можете настроить его для доступа через SSH или передать ему файлы с помощью FTP. Он имеет 512 МБ ОЗУ, 700 МГц микропроцессорную и аппаратную поддержку SPI, I2C и Serial.

### **6.1.1 Преимущества Raspberry Pi**

Ниже приведены некоторые из основных преимуществ Raspberry Pi.

- Очень легко подключиться к Интернету
- Доступен весь пакет программного обеспечения Linux
- Может быть запрограммирован с использованием различных языков программирования

### **6.1.2 Недостатки Raspberry Pi**

Ниже приведены некоторые из основных ограничений малины Pi.

- Доступ к аппаратным средствам не в режиме реального времени. Если CPU занят, тогда сопряжение с оборудованием может быть отложено
- Не хватает мощности для индуктивных нагрузок
- Нет встроенного аналого-цифрового преобразователя.
- Конструкция аппаратного обеспечения не является открытым исходным кодом. Несмотря на то, что это не очень важно, для некоторых пользователей это может нарушить функциональность.

### **6.1.3 Когда использовать и не использовать Raspberry Pi**

Если у вашего проекта очень ограниченное аппаратное взаимодействие, но оно немного сложно со стороны программного обеспечения или необходимо подключиться к Интернету, тогда вы должны просто пойти с Raspberry Pi. <sup>[19]</sup> Кроме того, если вы хотите программировать с использованием различных языков программирования, вам следует выбрать «Raspberry Pi».

С другой стороны, если вашему проекту требуется много аппаратных интерфейсов и нужно читать данные с большого количества датчиков или контролировать количество устройств, то Raspberry Pi может быть не лучшим выбором.

## **6.2 Что такое Arduino?**

Arduino - это коллекция из трех компонентов. Есть платформа прототипа оборудования, язык Arduino и библиотеки IDE и библиотеки. Платы Arduino - это микроконтроллеры, а не полноценный компьютер. Они не запускают полную операционную систему, а просто записывают код и

выполняются, когда интерпретирует их прошивка. Платформа Arduino изображена на рисунке.



Рисунок 13 -Arduino

Основной целью платы Arduino является взаимодействие с устройствами и датчиками, поэтому она отлично подходит для аппаратных проектов, в которых вы просто хотите, чтобы все отвечало на различные показания датчиков и ручной ввод. Возможно, это не так много, но на самом деле это очень сложная система, которая позволяет вам лучше управлять своими устройствами. Он отлично подходит для взаимодействия с другими устройствами и приводами, где полная операционная система будет перегружена для обработки простых действий чтения и ответа. Он имеет 8-битный микроконтроллер AVR и аппаратную поддержку SPI, I2C и Serial.

### 6.2.1 Преимущества Arduino

Ниже приведены некоторые из основных преимуществ Arduino.

Очень легко начать.

Может использоваться для приложений реального времени для аппаратного и программного обеспечения, а IDE - с открытым исходным кодом.

Нет необходимости в большой базе знаний о программировании, необходимых для создания базовых материалов.

Он очень прост в распространении и содержит множество пользовательских экранов и библиотек. Щиты доступны, чтобы делать что-то привлекательное.

### **6.2.2 Недостатки Arduino**

Ниже приведены некоторые из основных ограничений Arduino.

Arduino не очень мощная плата по сравнению с Raspberry Pi

Вам необходимо программировать, используя лишь Arduino или C / C++

Подключение к Интернету затруднено, хоть и не является невозможным. Возможно проведение анализа Arduino с помощью YQL и JSON.

### **6.2.3 Когда использовать и не использовать Arduino**

Если ваш проект требует от вас взаимодействия с множеством внешних требований к оборудованию, вам следует подумать об использовании Arduino.<sup>[20]</sup>

С другой стороны, если ваш проект требует многого, так что вам нужно написать сложный программный стек или протоколы программного обеспечения, тогда Arduino может оказаться не лучшим выбором.

### 6.3 Сравнение Ардуино и Raspberry Pi

Выбор какой платы вы хотите, зависит от типа проекта, который вы хотите сделать, и от вашего опыта программирования. Если у вас нет опыта программирования или электроники, вы найдете в Arduino более приемлемую кривую обучения, чем Raspberry Pi, так как вам придется изучать их одновременно. Arduino уже давно существует, и есть много полезных руководств, для начала работы с ним. Если у вас есть опыт программирования, у вас не будет проблемы с тем, чтобы сразу выбрать Arduino.

Arduino основан на оборудовании, что означает, что вы получите далеко не все компоненты: ЖК-дисплеи, светодиоды, резисторы, двигатели и т. Д. В зависимости от того, какой проект вы хотите сделать. Вам не нужно никакого опыта или компонентов, чтобы заставить Raspberry Pi сделать что-то. Просто подключи и работай. Сравнительные характеристики приведены в таблице 1.

Таблица 1- Сравнительные характеристики Raspberry Pi и Arduino

<b>Особенности</b>	<b>Raspberry Pi</b>	<b>Arduino</b>
Языки программирования	Без ограничений	Arduini, C/C++
Скорость процессора	700 MHz	16MHz
Подключение к интернету	Легко	Относительно не просто
Конструкция оборудования	Закрытый корпус	Источник с открытым исходным кодом
Работа в реальном времени	Затруднена	Относительно проста
Переход от аналогового к цифровому	Нет	Да

Если вы хотите создать аппаратный проект, то Arduino - лучший выбор. Входной сигнал задается аналоговым, а выход - PWM и весь спектр совместимости, который Raspberry Pi не может делать изначально.<sup>[21]</sup> Кроме того, большие контакты ввода / вывода позволяют подключать несколько датчиков и компонентов обратной связи. Arduino, однако, не такой мощный. Arduino может отправлять данные на ваш ПК или Raspberry Pi, через последовательный порт, и затем вы можете создать программу для чтения этих данных и сделать то, что вам необходимо.

Если вы хотите создать программный проект, то Raspberry Pi - это выход. Возможности обработки видео, аудио и данных из интернета делают его победителем в данном аспекте. Нет необходимости прикреплять внешние компоненты, поэтому нет необходимости изучать электронику

Таким образом, это все о различиях между Arduino и Raspberry Pi.

Учитывая достоинства и недостатки рассмотренных выше Arduino и Raspberry Pi выбран был последний.

## **7 Работа с Raspberry Pi**

В данной главе будет описан процесс знакомства с микрокомпьютером — от распаковки до первого запуска операционной системы и выполнения первоначальных настроек.<sup>[22]</sup>

### **7.1 Аппаратная часть**

Для начала работы необходимо:

- Raspberry Pi 3 Model B
- Акриловый корпус для Raspberry Pi 3B / 2B
- Набор медных радиаторов для Raspberry Pi
- Карта памяти microSD объемом от 8Гб
- Блок питания 5V 2.5A

Это те элементы, из которых состоит микрокомпьютер в сборе. Также необходимо иметь следующую периферию:

- Проводная мышь
- Проводная клавиатура
- Картридер для microSD-карт или устройство, которое способно выполнять его функцию
- Монитор или телевизор с HDMI-входом
- Картридер для записи образа операционной системы
- Устройства ввода
- Экран для вывода информации

Для удобства будет не лишним иметь:

- Bluetooth-мышь
- Bluetooth-клавиатура

Стоит отметить, управлять «Rasberry» беспроводными клавиатурой и мышью и удобнее, чем проводными.

Представлю небольшой обзор Raspberry Pi 3 Model B, упаковка которого представлена на рисунке.



Рисунок 14- Raspberry Pi в упаковке

Микрокомпьютер Raspberry Pi 3 Model B поставляется в картонной коробке бело-красной расцветки с логотипом в виде малины, являющимся фирменным.



Рисунок 15 - Брошюра-памятка

Комплект поставки — сама плата Raspberry Pi 3 в антистатическом пакете и небольшая брошюра-памятка по технике безопасности (рисунок ), в котором указана следующая информация

- не накрывать «Raspberry» во включенном виде
- избегать перегрева
- не разгонять
- избегать механического повреждения печатной платы
- не использовать в сырых и влажных местах и т.д.

Платы Raspberry Pi являются коммерческой разработкой, и выпускаются только двумя компаниями: Element14 в Китае и RS Components в Англии.

Платы производства RS Components/Allied Electronics и Element14 идентичны и единственным различием между ними является дизайн упаковки — на ней будут логотипы одного из производителей. [23]

На печатной плате Raspberry Pi 3 английского производства имеется надпись «made in UK». На плате Element14 надпись выглядит как «made in PRC» (People's Republic of China).

Что представляет из себя Raspberry Pi 3 Model) B (рисунок в техническом плане?



Рисунок 16- Raspberry Pi

- Процессор: ARM Cortex-A53 (4 ядра)
- SoC: Broadcom BCM2837
- Встроенные адаптеры: 10/100Мбит Ethernet, Bluetooth 4.1 LE, Wi-Fi 802.11n
- Графический процессор: Broadcom VideoCore IV
- Оперативная память: 1Гб LDDR2
- Разъем под microSD-карту
- Порты: 4xUSB 2.0, HDMI, 3.5мм аудиовыход, 40-pin GPIO (как на Arduino), Camera Serial Interface (CSI), Display Serial Interface (DSI)

Главные отличия Raspberry Pi 3 от предшествующего поколения— встроенные адаптеры Wi-Fi и Bluetooth и более мощный процессор.

Ранее при использовании Raspberry Pi имелась необходимость 2 USB-порта из четырех имеющихся затыкать адаптерами Wi-Fi и Bluetooth, а теперь они непосредственно распаяны на печатной плате и оба этих интерфейса работают без ранее необходимых адаптеров. <sup>[24]</sup> Кроме того «Raspberry» стала выглядеть эстетичнее и еще удешевило конечную стоимость эксплуатации платы, так как даже относительно недорогие USB-адаптеры Wi-Fi и Bluetooth стоят не менее 5 долларов за каждый.

### **7.1.1 Блок питания для Raspberry Pi**

Питается Raspberry Pi через microUSB-разъем, который используется и на большинстве современных мобильных гаджетах.

Raspberry Pi довольно требователен к электропитанию — для хорошей работы необходимо использовать источник питания, который способен выдавать не менее 2А тока, а лучше 2,5А.

При недостаточном питании Raspberry Pi может запускаться, но будет очень медленно работать.

Raspberry Pi может работать медленно по 2 причинам: либо недостаточно питания, либо используется microSD-карта с классом скорости ниже 10 класса скорости. Если существует нехватка питания, то на подключенном к Raspberry Pi экране в правом верхнем углу начнет отображаться значок молнии желтого цвета.

Питания может не доставать по 2 причинам: плохой USB-кабель или слабый источник питания.

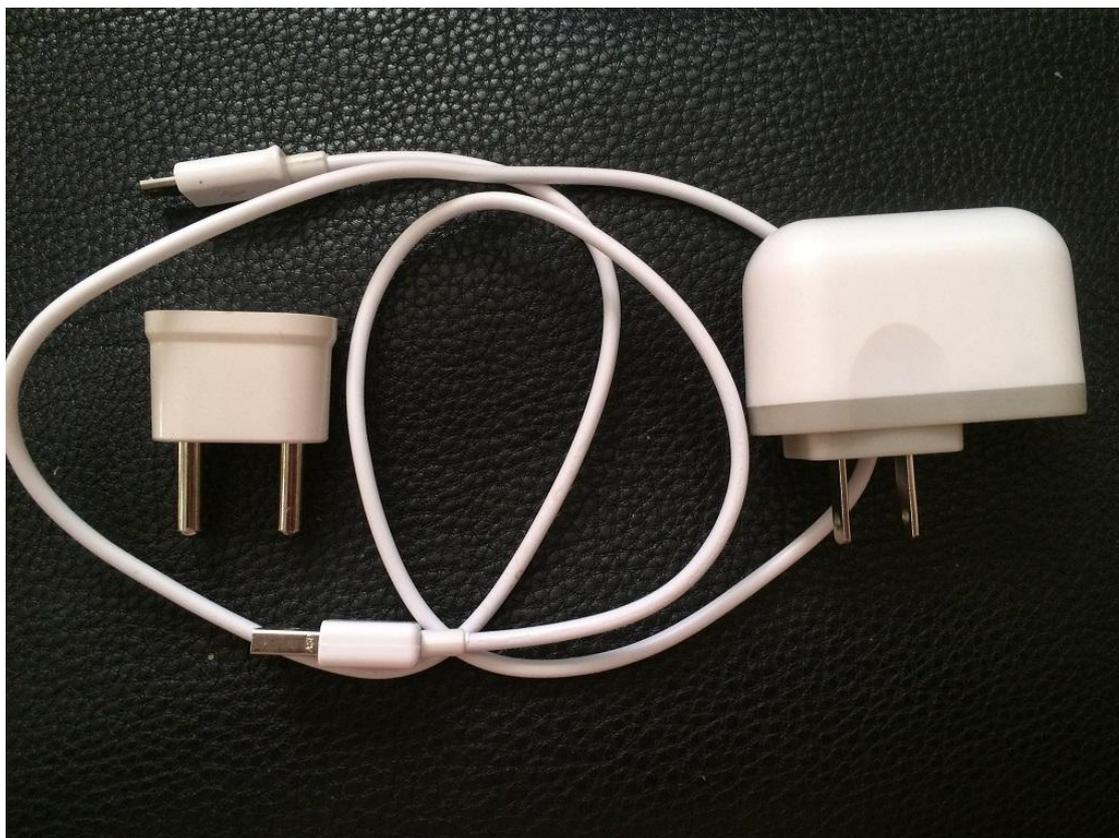


Рисунок 17- Блок питания и USB-кабель

Так что к выбору блока питания необходимо подходить осознанно.

### 7.1.2 Установка радиаторов на микрокомпьютер



Рисунок 18- Радиаторы на Raspberry Pi

При подготовке микрокомпьютера к запуску первым делом необходимо позаботиться о его охлаждении, так как перед нами компьютер с чипами и процессором, имеющими свойство нагреваться во время работы. [25]

КНР производит и продает наборы различных радиаторов для микрокомпьютеров. Они бывают керамическими, медными и алюминиевыми. Лучше выбирать набор из меди, так как теплоемкость таких радиаторов больше чем у алюминиевых.



Рисунок 19- Внешний вид радиаторов

Плоская пластина должна быть закреплена на чипе оперативной памяти с нижней стороны, а два крупных радиатора крепятся непосредственно на верхней части платы.

Радиаторы крепятся к чипам обычным двусторонним скотчем.



Рисунок 20- Установка радиаторов

Перед наклейкой радиаторов чипы следует обезжирить спиртом. Чип слева (рисунок) - однокристальная система Broadcom BCM2837 с интегрированным графическим адаптером и процессором. В синтетических тестах процессор греется до порядка  $100^{\circ}\text{C}$ , таким образом оставлять его без радиаторов охлаждения нельзя. [26]

Возле USB-портов располагается контроллер USB. Он тоже греется, хоть и не настолько сильно, как однокристальная система. Второй радиатор будет закреплен именно на нем. На рисунках ниже изображена установка радиаторов.



Рисунок 21- Установка радиаторов

2 верхних радиатора надежно закреплены на своих местах.



## Рисунок 22- Установка медной пластины

Пластина закреплена на модуле 1Гб LDDR2 памяти с нижней стороны. Данный модуль сильно не нагревается, поэтому установка полноразмерного радиатора на нем не является необходимым. Но медная пластина вряд ли помешает, так как ее наличие может защитить чип от возможных механических повреждений. [27]

### 7.1.3 Сборка корпуса для Raspberry Pi 3

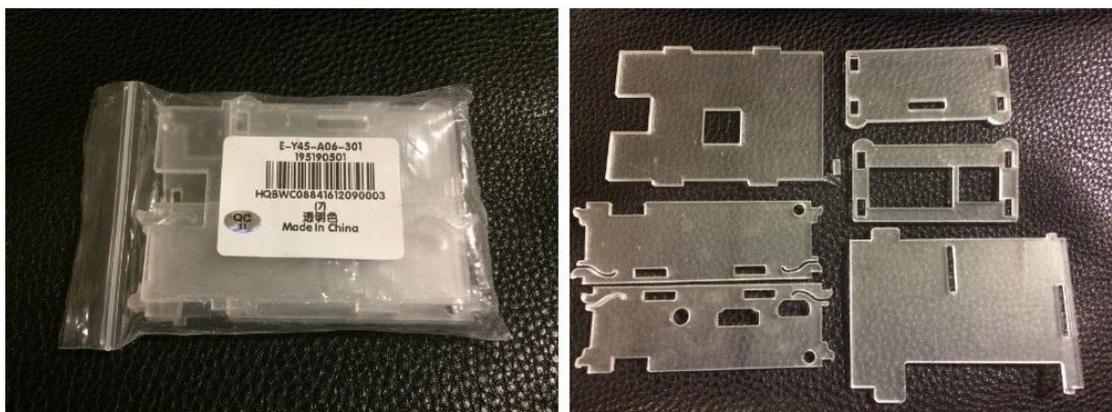


Рисунок 23- корпус Raspberry Pi

После установки радиаторов далее необходимо установить микрокомпьютер в корпус. На этом приготовления к запуску закончены.

Корпус, как собственно и радиаторы, продается отдельно. Можно обойтись и без корпуса — но плата микрокомпьютера может быть случайно повреждена каким-то образом или погибнуть от разряда статического электричества.

Кроме того, установка в корпусе придает конструкции эстетичный и законченный вид.

Корпус, изображенный на рисунке, сделан из оргстекла и поставляется набором для самостоятельной сборки.

Набор состоит из 6 акриловых пластин: подъемная крышка, днище и 4 боковушки. Микрокомпьютер в сборе с набором из пластин корпуса изображен на рисунке 24.



Рисунок 24- Raspberry Pi в корпусе

Единственный недостаток корпуса в том, что он быстро собирает на себя жировые отпечатки пальцев, которые портят вид.

## **7.2 Установка и настройка Raspbian**

Raspberry Pi установлен в корпус, на чипы наклеены радиаторы, найден подходящий источник питания.

Следующий шаг- сделать базовые настройки и установить на микрокомпьютер операционную систему.

### **7.2.1 Установка Raspbian**

На настоящий момент для Raspberry Pi существует 6 официальных операционных систем и порядка 30 неофициальных, включая всем известный Android.

В подавляющем большинстве руководств по Raspberry Pi рекомендуют использовать установщик под названием NOOBS (New Out Of

The Box Software). Это официальный установщик от разработчиков Raspberry Pi, который содержит в себе дистрибутив Raspbian и возможность загрузки из остальных пяти официальных дистрибутивов и установить любой из них на микрокомпьютер.

NOOBS имеет смысл использовать для установки одновременно двух или более разных ОС на одну Raspberry Pi, с возможностью выбора между установленными ОС при включении.

Если же планировать использовать на Raspberry Pi только одну операционную систему то можно поступить другим более простым способом: записать ОС на microSD-карту.

Потребуется непосредственно microSD-карта и картридер или устройство, которое способно его заменить (смартфон, mp3-плеер, планшет, видеорегиистратор,— любое устройство, которое отображает установленную карту памяти в качестве накопителя при подключении к компьютеру).

Процесс установки Raspbian сводится к трем шагам. <sup>[28]</sup>

#### **7.2.1.1 Шаг 1. Форматирование microSD-карты**

MicroSD-карту необходимо отформатировать в формате FAT32.

Для этого можно использовать утилиту SD Card Formatter.

Средство для форматирования Windows не подходит, так как при попытке запуска Raspbian карты памяти отформатированной через него выводится ошибка «error resizing existing FAT partition».

#### **7.2.1.2 Шаг 2. Запись дистрибутива Raspbian**

После подготовки microSD-карты нужно с официального сайта скачать дистрибутив Raspbian.

Образ дистрибутива заархивирован, так что после скачивания его необходимо распаковать в любую папку.

Распакованный образ Raspbian необходимо записать на отформатированную microSD-карту, для чего подойдет утилита Rufus.

### 7.2.1.3 Шаг 3. Первый запуск Raspbian на Raspberry Pi

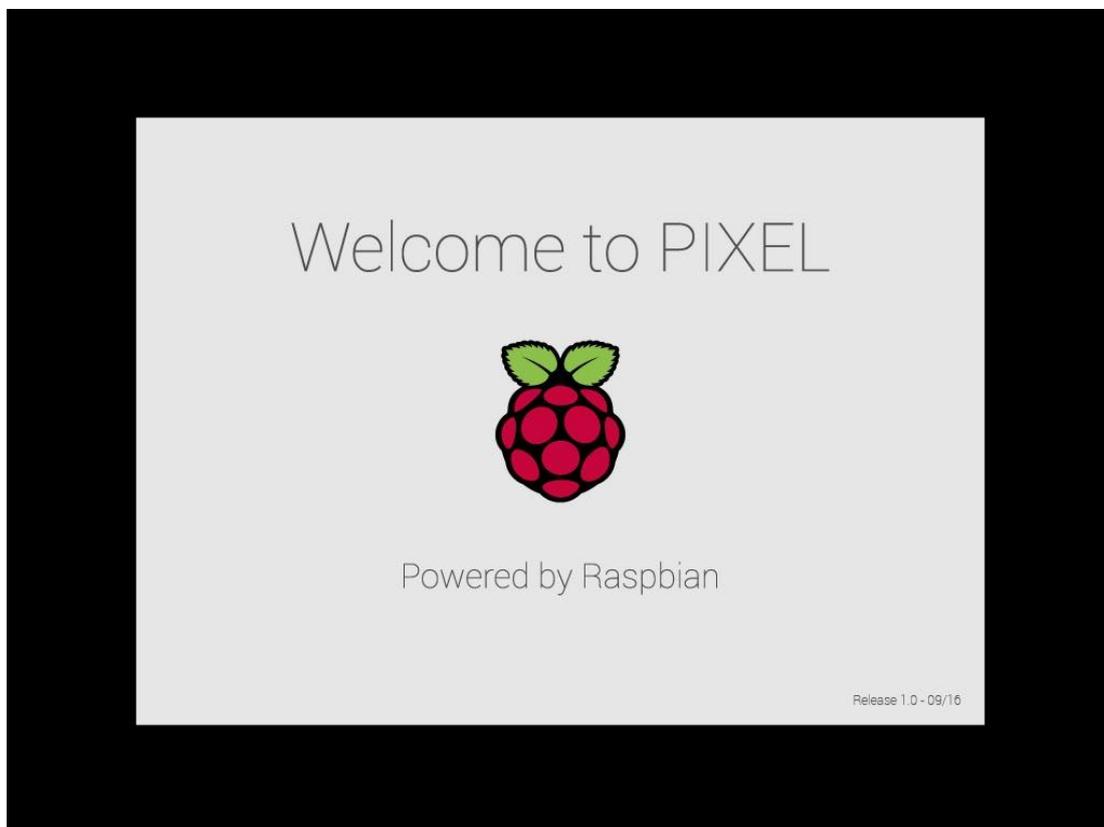


Рисунок 25- Скрин запуска Raspbian

Процедуру установки на этом можно считать завершенной.

Карту памяти с записанным Raspbian необходимо установить в Raspberry Pi и подключить к нему питание и монитор, после чего произойдет первый запуск операционной системы.

Теперь можно подключить Raspberry Pi к Wi-Fi сети, а при наличии Bluetooth клавиатуры и мыши, можно для большего удобства использования провести их сопряжение с микрокомпьютером.

### 7.2.2 Установка обновлений

После установки Raspbian и подключения к интернету посредством Wi-Fi и Ethernet необходимо обновить все пакеты до актуальной версии.

Это производится через терминал с помощью двух команд:

```
sudo apt-get upgrade  
sudo apt-get update
```

Вторая команда загружает сведения о наличии обновлений пакетов, первая непосредственно загружает сами обновления.

Также можно обновить еще прошивку Raspberry Pi следующей командой:

```
sudo rpi-update
```

После завершения установки обновлений Raspberry Pi необходимо перезагрузить:

```
reboot
```

Если все сделано верно, то после перезагрузки система сразу уведомит нас об обновлении до актуальной версии.

Большинство команд в \*nix-системах начинаются с `sudo`, что означает, что исполнение команды запускается с правами суперпользователя, что аналогично «Запуску от имени администратора» в ОС Windows.

Затем убираем с краев экрана черные поля. (В части случаев после загрузки Raspbian пользователь может обнаружить черные поля по краям экрана).

Это означает то, что развертка выставлена неверно.

Исправляется данная проблема путем правки конфигурации:

```
sudo leafpad
```

Затем открываем файл /boot/config.txt и далее необходимо найти в нем строку:

```
#disable_overscan=1
```

Потом убираем символ #, превращая тем самым команду из комментария в исполняемую:

```
disable_overscan=1
```

Интерфейс Raspbian после перезагрузки будет отображаться на экране без черных полей.

### 7.2.3 Настройки локализации

Raspbian настроен под британского пользователя по умолчанию.

Изменить настройки локализации можно в настройках, как через консоль:

```
sudo raspi-config
```

так и через графический интерфейс (Menu -> Parameters -> Raspberry Pi Configuration)

Запускаем «Raspberry Pi Configuration».

На вкладке System выбираем «Change Password...», задаем root-пароль. Он отсутствует по умолчанию, что может помешать в дальнейшем выполнению части команд.

Переключаемся на вкладку Localisation.

«Set Locale...» — находим Country, выбираем RU (Russia), Character Set: UTF-8. Язык системы можно оставить как есть а можно поменять на «ru (Russian. Русификация Raspbian выполнена не на все 100%, и потому при выборе русского языка придется довольствоваться смешанными русским и английским текстами в интерфейсе.

«Set Timezone...» — здесь можно выбрать часовой пояс. Так как территория Российской Федерации захватывает и Азию и Европу, то в зависимости от положения во вкладке «Area» выбирается «Europe» либо «Asia», а в «Location» — город, часовой пояс которого совпадает с часовым поясом вашей местности.

В качестве примера, на Урале часовой пояс UTC/GMT+5, что называется «Екатеринбургским часовым поясом». В «Location» в Raspbian нет такого населенного пункта, но есть имеется Ashkhabat, с таким же часовым поясом.

«Set WiFi Country» — здесь выставляем «RU Russia», хотя необходимости в этом нет, так как встроенный адаптер работает по умолчанию на локальных настройках. Эта настройка скорее всего важна для некоторых стран так называемого третьего мира, где Wi-Fi стандарты ограничиваются в милитаристских целях.

#### **7.2.4 Раскладка клавиатуры**

Необходимые раскладки клавиатуры и переключение между ними добавляются вводом команды:

```
setxkbmap us,ru -option grp:alt_shift_toggle
```

В этом случае добавляются английская и русская раскладка. Английская является основной, а переключение происходит по сочетанию клавиш Shift +Alt.

Можно изменить сочетание на Shift+Ctrl, внося соответствующую правку в эту команду.

Небольшая проблема в том, что Raspbian не может запомнить эту команду, и после перезагрузки Raspberry Pi ее приходится вводить заново, что не совсем удобно.

Ввиду этого открываем редактор текста:

```
sudo leafpad
```

Правим системные настройки клавиатуры, которые хранятся в файле /etc/default/keyboard.

Заменяем строки:

```
XKBOPTIONS=""  
XKBLayout="gb"
```

На:

```
XKBOPTIONS="grp:alt_shift_toggle,grp_led:scroll"  
XKBLayout="us,ru"
```

Добавляем в систему графический индикатор клавиатурной раскладки:

```
sudo apt-get install gxkb
```

Поместим его в автозагрузку. Это также делается через редактирование текстовых файлов:

```
sudo leafpad
```

Затем открываем файл `/home/pi/.config/lxsession/LXDE-pi/autostart` и добавляем в его конец строку:

```
@gxkb
```

Теперь осталось перезагрузить Raspberry Pi для окончательного применения изменений:

```
reboot
```

Базовую настройку можно считать завершённой.

### 7.3 Впечатления от использования Raspberry Pi 3

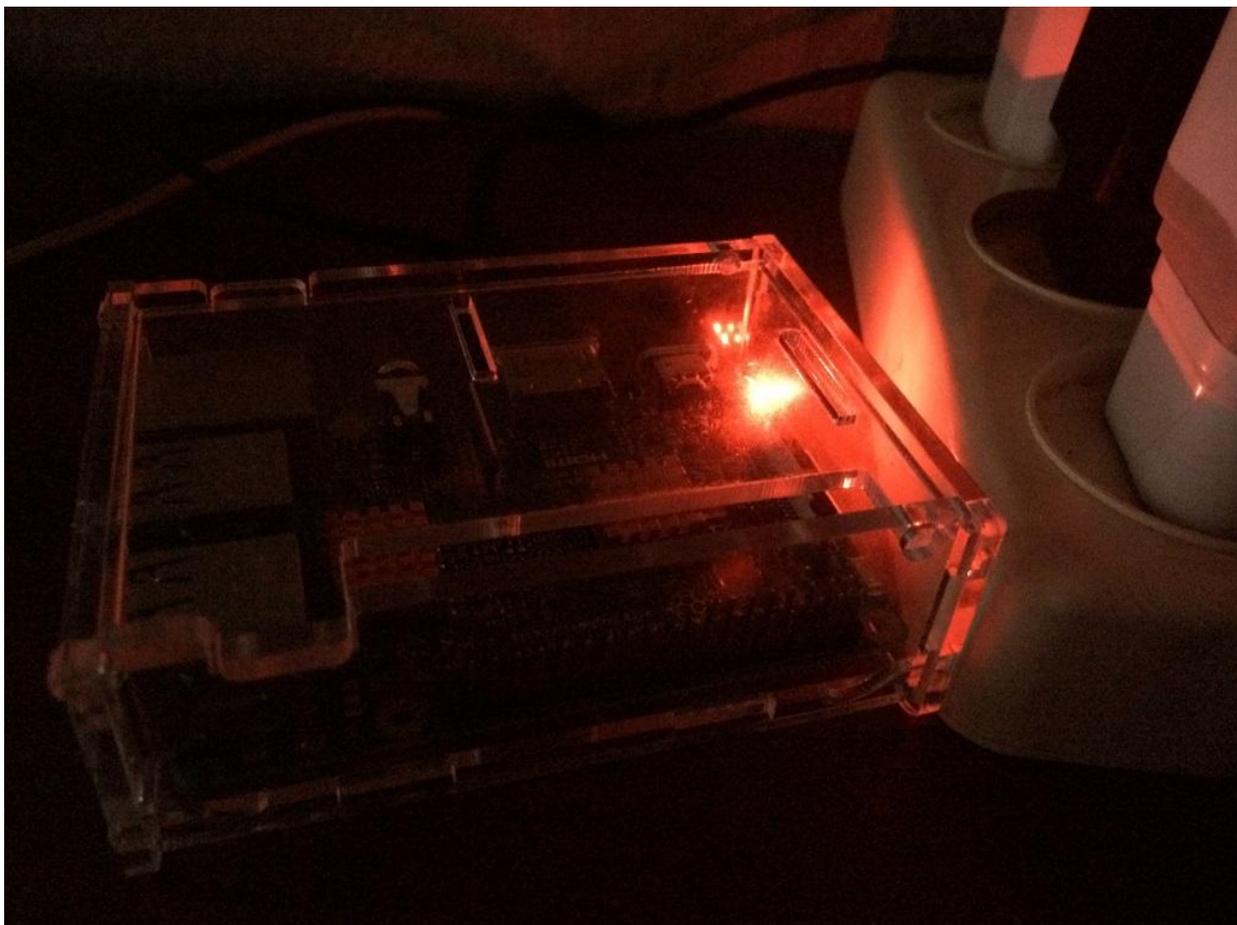


Рисунок 26- Raspberry Pi во время работы

При использовании Raspberry Pi первое на что обращаешь внимание —бесшумность его работы.

Учитывая, что если используется только пассивное охлаждение и нет воздушных кулеров и обычного HDD на основе подвижных частей, то шум издавать просто нечему.

Факт существования подобного компьютера — именно, полноценного компьютера, полностью бесшумного — потрясает воображение.

Теперь насчет производительности.

Мощности Raspberry Pi 3 Model B вполне хватает на выполнение офисных и мультимедийных задач: просмотр видео, комфортную работу в офисном пакете и серфинг не тяжелых сайтов в сети.

Просмотр онлайн-видео дается «Raspberry» уже с трудом, а при открытии 4-5 вкладок ютуба одновременно Raspbian может начать подвисать.

С воспроизведением онлайн-видео все обстоит более чем хорошо в специально заточенных под мультимедийную составляющую операционных системах — например, OpenELEC.

Дистрибутив Raspbian предназначен для управления периферией, подключенной по GPIO и для обучения программированию.

Для тех, кто заинтересовался покупкой Raspberry Pi 3 хочу сказать, что цена на этот микрокомпьютер колеблется в пределах \$35-\$40.

## **8 Подключение Raspberry Pi к MathLab Simulink**

В среде MathLab есть приложение, которое обеспечивает связь и совместную работу MathLab и Simulink с Raspberry Pi.<sup>[29]</sup> Этот инструмент дает возможность использовать Raspberry Pi (RPi) в качестве интерфейса связи MathLab с внешней средой при решении задач управления, моделирования, синтеза систем, идентификации, тестирования и наблюдения и других. Специализированные средства MathLab могут задействоваться и для разработки алгоритмов автономной работы Raspberry Pi.

## Общие сведения

Установка пакета MathLab для работы с Raspberry Pi и установка ОС Raspbian Wheezy на SD карту rpi для взаимодействия с пакетом MathLab

Установка пакета MathLab для работы с Raspberry Pi выполняется через Интернет. Взаимодействие MathLab с RPi осуществляется через операционную среду (ОС) Linux: Raspbian Wheezy можно установить на SD карту Raspberry Pi в процессе установки пакета MathLab. Установка может быть выполнена несколькими методами:

1. Через МатЛАБ Toolstrip > Add-Ons > Get Hardware Support Packages

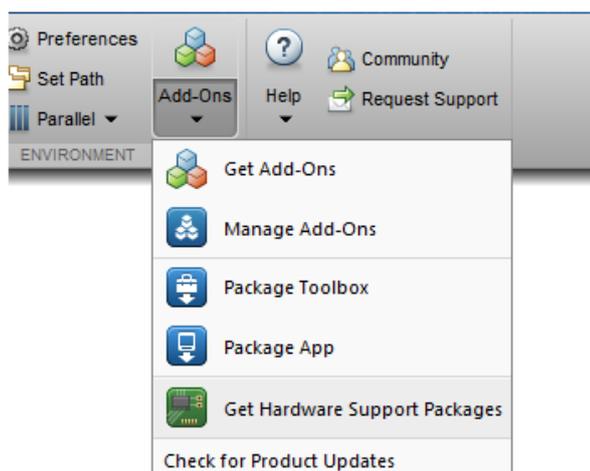


Рисунок 27– Get Hardware Support Packages

Выбираем Install from Internet

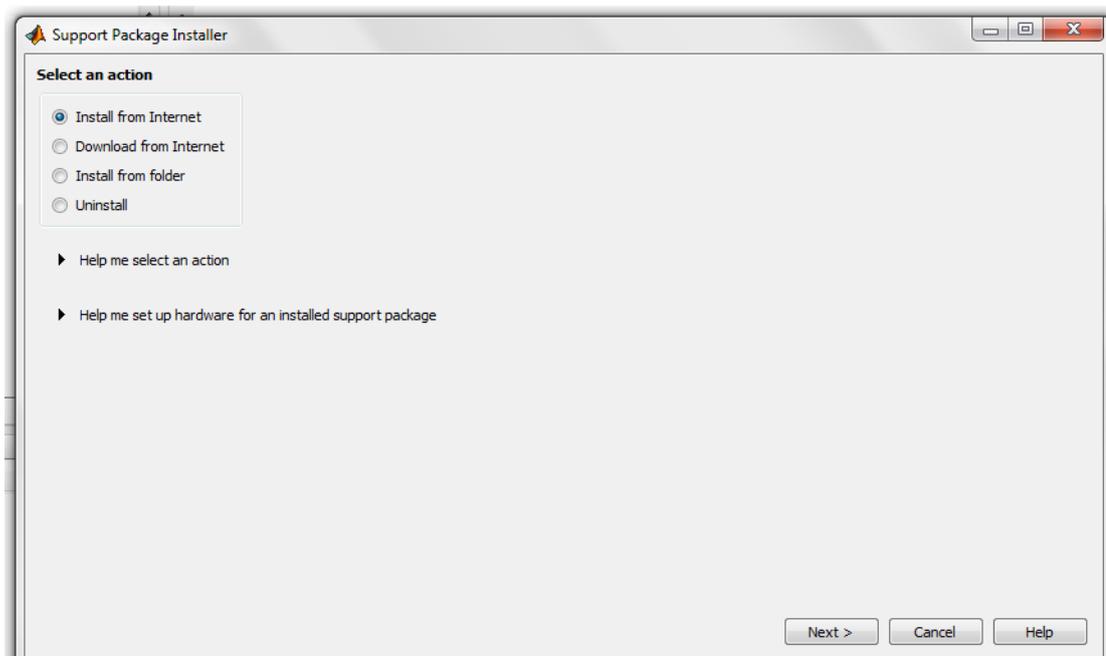


Рисунок 28- Скрин выбора метода установки

Находим необходимое приложение и нажимаем Next, после чего приложения для MathLab и MathLab/Simulink установятся на ваш компьютер.

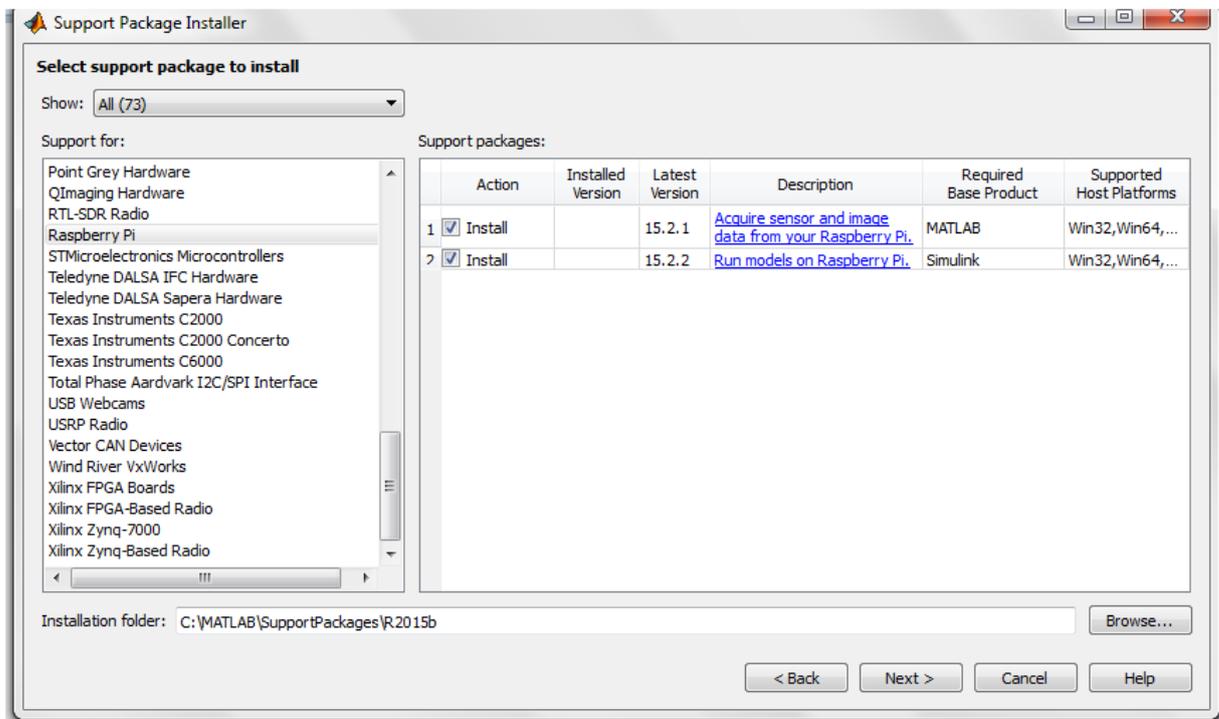


Рисунок 29- Установка приложений

## 2. С официального сайта

1. С официального сайта МатЛАБ <http://www.mathworks.com/hardware-support/raspberry-pisimulink.html> загрузите установочный файл пакета MathLab для работы с RPi. <sup>[30]</sup>

2. Запустите MathLab и настройте его на каталог с загруженным файлом.

3. В среде MathLab запустите установочный файл.

4. Дойдите до следующего указателя и нажмите Log In.

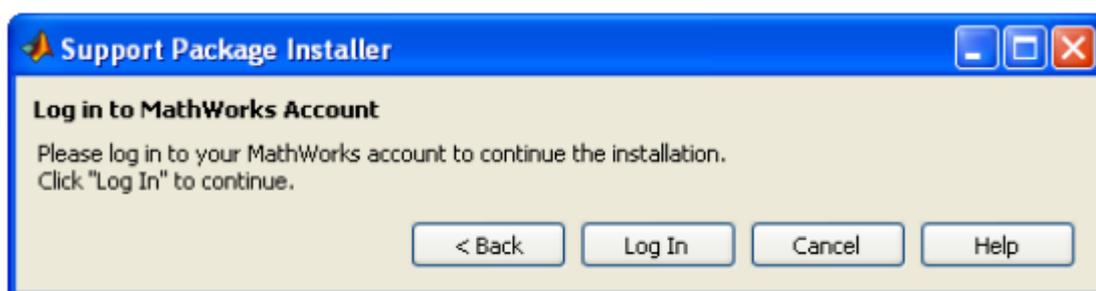


Рисунок 30- Начало установки

5. Зарегистрируйтесь на сайте MathLab и/или введите параметры вашей ранее зарегистрированной учетной записи.



Рисунок 31- Регистрация на сайте MathLab

6. Выполняйте последующие шаги по установке пакета и дождитесь окончания загрузки Операционной Среды Linux : Raspbian Wheezy.

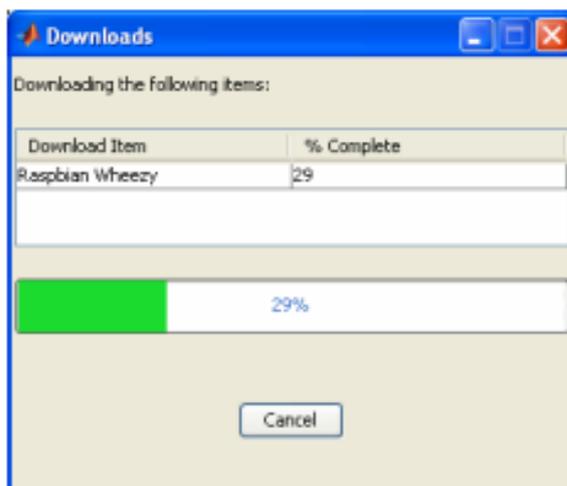
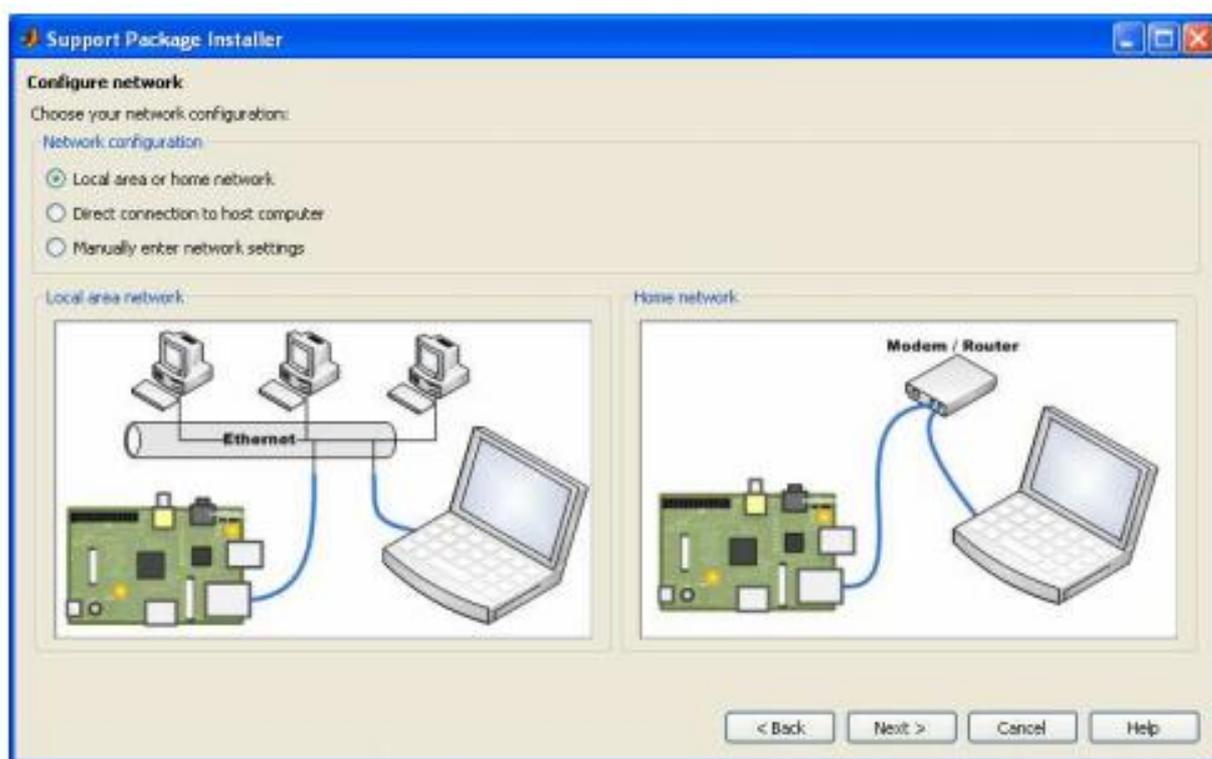


Рисунок 32- Скрин загрузки

7. Выберите вариант подключения Raspberry Pi к хост компьютеру с Linux. Это необходимо для начальной настройки Raspbian Wheezy на SD карту RPi.



### Рисунок 33- Скрин установки Raspbian Wheezy

8. Для установки Linux операционной среды Raspbian Wheezy на SD карту (объем должен быть не менее 8ГБ) вставьте карту в считывающее устройство компьютера.

Обратите внимание, что Если на SD карте имеются посторонние данные, то они будут впоследствии удалены при установке Raspbian Wheezy.



Рисунок 34- Установка SD карты

9. Запустите установку Raspbian Wheezy (приблизительный объем данных около 2.8 ГБ) на SD карту RPi.

10. После установки ОС вставьте SD карту в приемник Raspberry Pi, подключите компьютер к сети и подайте питание на RPi.

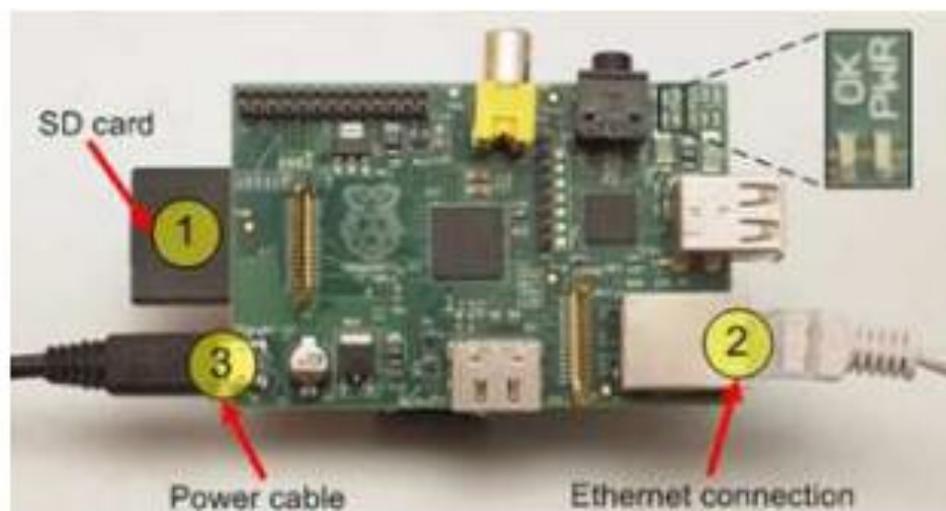


Рисунок 35- Подсоединение кабелей

11. Перейдите к следующему пункту для сканирования сетевых устройств. При достижении успеха при сканировании установщик определит IP адреса RPi. Сохраните Host name, IP адрес, password и user name.

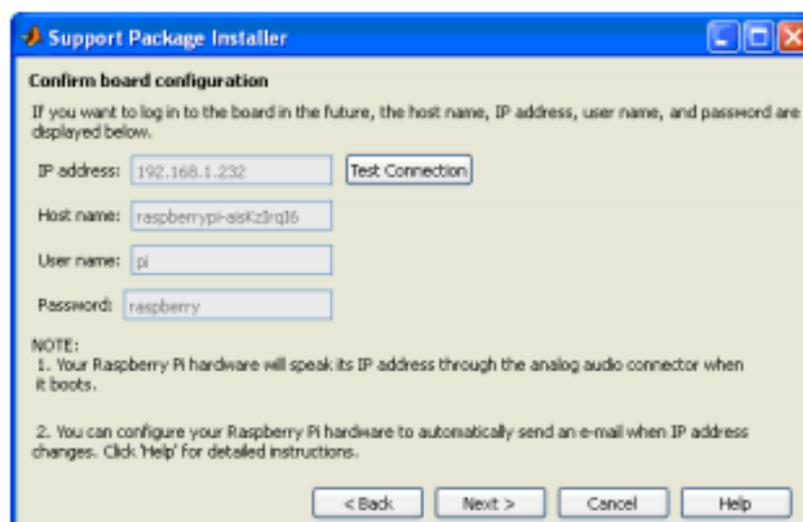


Рисунок 36- Сохранение параметров

12. Для завершения установки необходимо нажать Finish.

Примечание: Установку пакета Raspbian Wheezy из среды MathLab можно выполнить командой MathLab Targetupdater

ПОЛУЧЕНИЕ СЕТЕВЫХ ПАРАМЕТРОВ ДЛЯ СТАТИЧЕСКОГО IP СОЕДИНЕНИЯ

Для статического соединения:

- IP адрес каждого из сетевых устройств должен быть уникальным
- Subnet mask маска сети должна быть единой для всех сетевых устройств
- обычно используется единый шлюз (Gateway) для всех сетевых устройств

Для просмотра сетевых параметров необходимо набрать в командной строке основного компьютера

```
start
```

```
> run
```

```
> cmd
```

```
> ipconfig.
```

Если известен IP адрес одного из устройств, например, 192.168.1.XXX, то адреса других сетевых устройств лежат в диапазоне 192.168.1.3 .. 192.168.1.254, а маска сети для них остаётся такой же. <sup>[31]</sup>

## **ЗАКЛЮЧЕНИЕ**

В результате выполнения магистерской диссертационной работы выполнены основные задачи, поставленные индивидуальным планом научно-исследовательской работы. Последовательное выполнение поставленных задач привело к достижению основной цели диссертационной работы: реализации программно-аппаратного комплекса устройства для детектирования оставленных предметов.

Основываясь на существующем опыте обнаружения оставленных предметов, а также учитывая возможные ошибки в детектировании, которые могут возникать у современных комплексов, предложена модель устройства, которое при должной отладке станет бюджетной альтернативной существующим системам.

Существующие на данный момент методы и алгоритмы детектирования и отслеживания перемещения объектов еще далеки от совершенства. Совмещение метода поиска объектов переднего плана и треккинга одно из лучших решений для обнаружения объектов в видеопотоке. Таким образом, выбор данной совокупности алгоритмов для реализации в научно-исследовательской считаю оправданным.

Выбранная программный пакет MathLab Simulink позволяет реализовать поставленные задачи в программной части разрабатываемого устройства.

С помощью построения структурной схемы создаваемого комплекса удалось произвести подбор необходимых электронных составляющих, которые отличаются небольшой стоимостью и компактностью, что позволило разработать схемотехническую часть системы.

В результате проведения исследования и изучения доступной информации был обоснован выбор программного обеспечения для выполнения работы, а также выбор микрокомпьютера Raspberry Pi в качестве устройства управления.

Работа разработанного устройства сильно зависит от условий освещенности и интенсивности движения в зоне детектирования, что требует большой производительности. Путем настройки параметров схемы можно учитывать данные параметры непосредственно на ПК, на который установлено необходимое ПО.

Полученные результаты работы программ и исследований говорят о том, что подобные комплексы, детектирующие оставленные предметы можно создавать с наименее дорогими компонентами, что не только не скажется на качестве систем безопасности, но и удешевит установку в местах массового скопления людей, а значит и увеличит их доступность и использование на большем количестве опасных с точки зрения терроризма мест.

## СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Компьютерное зрение [Электронный ресурс] URL: [https://ru.wikipedia.org/wiki/Компьютерное\\_зрение](https://ru.wikipedia.org/wiki/Компьютерное_зрение) (дата обращения 15.05.18)
2. Терракты, совершенные в России [Электронный ресурс] URL: [https://ru.wikipedia.org/wiki/Терракты,\\_совершённые\\_в\\_России](https://ru.wikipedia.org/wiki/Терракты,_совершённые_в_России) (дата обращения 15.05.18)
3. Bechtel W. The Cardinal Mercier Lectures at the Catholic University of Louvain: An Exemplar Neural Mechanism: The Brain's Visual Processing System. 2003.
4. Datal N., Triggs B. Histograms of oriented gradients for human detection // IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2005. V. 1.
5. Ciresan D. C., Meier U., Masci J., Schmidhuber J. Multi-Column Deep Neural Network for Traffic Sign Classification // Neural Networks, 2012.
6. Viola P., Jones M. Robust Real-time Object Detection // Workshop on Statistical and Computation Theories Vision. 2001
7. Bay H., Tuytelaars T., Van Gool L. SURF: Speeded Up Robust Features // Proc. 9th European Conf. on Computer Vision. Graz, Austria. 2006. V. 3951.
8. Крик детектора [электронный ресурс] URL: [http://www.itv.ru/experience\\_the\\_next\\_magazine/articles/2242/](http://www.itv.ru/experience_the_next_magazine/articles/2242/) (дата обращения 6.03.18)
9. MathLab Examples [электронный ресурс] URL: [https://www.mathworks.com/help/matlab/examples.html?s\\_cid=doc\\_ftr](https://www.mathworks.com/help/matlab/examples.html?s_cid=doc_ftr) (дата обращения 6.03.18)
10. Системы компьютерного зрения [Электронный ресурс] URL: <http://books.ifmo.ru/file/pdf/2110.pdf> (дата обращения 1.05.18)

11. Компьютерное зрение [Электронный ресурс] URL: [http://info-farm.ru/alphabet\\_index/k/kompyuternoe-zrenie.html](http://info-farm.ru/alphabet_index/k/kompyuternoe-zrenie.html)  
(дата обращения 15.05.18)
12. Желтов С.Ю. и др. Обработка и анализ изображений в задачах машинного зрения. — М.: Физматкнига, 2010.
13. А.А. Лукьяница, А.Г. Шишкин. Цифровая обработка видеоизображений. — М.: «Ай-Эс-Эс Пресс», 2009
14. Computer Vision System Toolbox [Электронный ресурс]. // MathWorks // URL: [https://matlab.ru/products/computer-vision-system-toolbox/computer-vision-system-toolbox\\_rus.pdf](https://matlab.ru/products/computer-vision-system-toolbox/computer-vision-system-toolbox_rus.pdf) (Дата обращения: 12.10.2017).
15. Дэвид Форсайт, Жан Понс. Компьютерное зрение. Современный подход. — М.: «Вильямс», 2004.
16. О. П. Соловей, Н.Н. Иванов, Алгоритм трекинга объектов в реальном времени с обработкой ошибок, доклады Белорусского государственного университета информатики и радиоэлектроники, 2013. — 5 с.
17. Журавлев Ю.И. Об алгебраическом подходе к решению задач распознавания и классификации // Проблемы кибернетики. — М.: Наука, 1978, вып. 33. — С. 5-68.
18. Computer Vision: Algorithms and applications [Электронный ресурс] URL: <http://szeliski.org/Book/>  
(дата обращения 15.05.18)
19. Прэтт У. Цифровая обработка изображений: Пер. с англ. — М.: Мир, 1982. — Кн. 2 — 480 с., ил.
20. Методы компьютерной обработки изображений / Под ред. В.А. Сойфера. — 2-е изд., испр. — М.: ФИЗМАТЛИТ, 2003. — 784 с. — ISBN 5-9221-0270-2
21. Р. Гонсалес, Р. Вудс. Цифровая обработка изображений, Москва: Техносфера, 2005. — 1072 с. ISBN 5-94836-028-8.

22. Быстрые алгоритмы в цифровой обработке изображений / Т. С. Хуанг, Дж.-О. Эклунд, Г. Дж. Нус-сбаумер и др.; Под ред. Т. С. Хуанга: Пер. с англ. — М.: Радио и связь, 1984. — 224 с, ил.
23. А. С. Потапов, Р. О. Малашин/ СИСТЕМЫ КОМПЬЮТЕРНОГО ЗРЕНИЯ. — М.: Мир, 2004.
24. Шапиро Л., Стокман Д./ Компьютерное зрение — М.: Бинوم. ЛЗ
25. Aekta Patel, Hand Gesture and Neural Network Based Human Computer Interface [Электронный ресурс] URL: [http://ijera.com/papers/Vol4\\_issue6/Version%201/S04601119125.pdf](http://ijera.com/papers/Vol4_issue6/Version%201/S04601119125.pdf) (дата обращения 9.02.17)
26. N. R. Raajan, R. Krishna Kumar, S. Raghuraman, N. Ganapathy Sundaram, T. Vignesh Eye-hand Hybrid Gesture Recognition System for Human Machine Interface [Электронный ресурс] URL: <http://www.enggjournals.com/ijet/docs/IJET13-05-02-295.pdf> (дата обращения 15.02.17)
27. Zhenwen Gui, Yongtian Wang, Yue Liu, and Jing Chen, Mobile Visual Recognition on Smartphones [Электронный ресурс] URL: <https://www.hindawi.com/journals/js/2013/843727/> (дата обращения 19.02.17)
28. Snehlata Barde, A S Zadgaonkar, G R Sinha, PCA based Multimodal Biometrics using Ear and Face Modalities [Электронный ресурс] URL: <http://www.mecs-press.org/ijitcs/ijitcs-v6-n5/IJITCS-V6-N5-6.pdf> (дата обращения 27.02.17)
29. Jingting Zeng, Haibin Ling, Longin Jan Latecki, Shanon Fitzhugh, and Guodong Guo, Analysis of Facial Images across Age Progression by Humans [Электронный ресурс] URL: <https://www.hindawi.com/journals/isrn/2012/505974/> (дата обращения 1.03.17)
30. Xiaoyan Wang, Ming Xia, Huiwen Cai, Yong Gao, and Carlo Cattani, Hidden-Markov-Models-Based Dynamic Hand Gesture Recognition [Электронный ресурс] URL: <https://www.hindawi.com/journals/mpc/2012/986134/> (дата обращения 9.03.17)

31. Surendra Pal Singh , Kamal Jain , V. Ravibabu Mandla, 3D Scene Reconstruction from Video Camera for Virtual 3D City Modeling [Электронный ресурс] URL: <http://www.ajer.org/papers/v3%281%29/T031140148.pdf> (дата обращения 16.03.17)