

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ”

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

«ИССЛЕДОВАНИЕ АЛГОРИТМОВ ОБНАРУЖЕНИЯ И
ЛОКАЛИЗАЦИИ ОБЪЕКТОВ НА ВИДЕОИЗОБРАЖЕНИИ
В РОБОТОТЕХНИЧЕСКИХ ЗАДАЧАХ»

Автор Терентьев Роман Андреевич _____
(Фамилия, Имя, Отчество) (Подпись)

Направление подготовки (специальность) 15.03.06 _____
(код, наименование)

Квалификация бакалавр _____
(бакалавр, магистр)*

Руководитель ВКР Шаветов С.В., к.т.н. _____
(Фамилия, И., О., ученое звание, степень) (Подпись)

К защите допустить

Руководитель ОП Бобцов А.А., профессор, д.т.н. _____
(Фамилия, И.О., ученое звание, степень) (Подпись)

“ _____ ” _____ 20 ____ г.

Санкт-Петербург, 20 19 г.

Студент Терентьев Р.А. _____ Группа R3442 _____ Факультет СУиР _____
(Фамилия, И. О.)

Направленность (профиль), специализация _____

Консультант (ы):

а) _____
(Фамилия, И., О., ученое звание, степень) (Подпись)

б) _____
(Фамилия, И., О., ученое звание, степень) (Подпись)

ВКР принята “ _____ ” _____ 20 ____ г.

Оригинальность ВКР _____ %

ВКР выполнена с оценкой _____

Дата защиты “ _____ ” _____ 20 ____ г.

Секретарь ГЭК _____
(ФИО) (подпись)

Листов хранения _____

Демонстрационных материалов/Чертежей хранения _____

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ”

УТВЕРЖДАЮ

Руководитель ОП

(Фамилия, И.О.)

(подпись)

« ____ » « ____ » 20 ____ г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

Студенту Терентьеву Роману Андреевичу Группа R3442 Факультет СУиР

Руководитель ВКР Шаветов С.В., к.т.н., доцент факультета СУиР

(ФИО, ученое звание, степень, место работы, должность)

1 Наименование темы: Исследование алгоритмов обнаружения и локализации объектов на видеоизображении в робототехнических задачах

Направление подготовки (специальность) 15.03.06

Направленность (профиль) Интеллектуальные технологии в робототехнике

Квалификация бакалавр

2 Срок сдачи студентом законченной работы « 1 » « мая » 20 19 г.

3 Техническое задание и исходные данные к работе

Поставлена задача произвести сравнение различных алгоритмов технического зрения в задачах слежения за объектом. Необходимо:

1. Провести обзор существующих алгоритмов обнаружения и локализации объектов;
2. Реализовать выбранные алгоритмы в виде комплекса программ;
3. Проверить работу полученных алгоритмов в реальных условиях на базе мобильной робототехнической платформы Robotino;
4. Сравнить результаты испытаний по различным характеристикам: быстродействие, робастность, зависимость от освещения и т.д.

4 Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов)

1. Введение

2. Обзор существующих алгоритмов обнаружения и локализации объектов

3. Программная реализация выбранных алгоритмов

4. Проверка работы алгоритмов в реальных условиях

5. Заключение

5 Перечень графического материала (с указанием обязательного материала)

1. Презентация Power Point

6 Исходные материалы и пособия

1. Обработка и анализ изображений в задачах машинного зрения: Учебное издание://

Ю. В. Визильтер, С. Ю. Желтов, А. В. Бондаренко и др. / Под ред. А. К. Розанова – М.:

Физматкнига, 2010. – 672 с

2. OpenCV Documentation [Электронный ресурс]: <https://docs.opencv.org/3.3.1/>

7 Дата выдачи задания « 01 » « _____ ноября _____ » 20 18 г.

Руководитель ВКР _____
(подпись)

Задание принял к исполнению _____ « ____ » « _____ » 20 ____ г.
(подпись)

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ”

АННОТАЦИЯ

ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Студент _____ Терентьев Роман Андреевич _____
(ФИО)

Наименование темы ВКР: _____ Исследование алгоритмов обнаружения и локализации _____
объектов на видеоизображении в робототехнических задачах _____

Наименование организации, где выполнена ВКР _____ Университет ИТМО _____

ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

1 Цель исследования _____ Сравнение алгоритмов обнаружения и локализации объектов _____
с помощью технического зрения _____

2 Задачи, решаемые в ВКР _____ Обзор существующих алгоритмов; реализация выбранных _____
алгоритмов; проверка их работы в реальных условиях; сравнение алгоритмов _____

3 Число источников, использованных при составлении обзора _____ 9 _____

4 Полное число источников, использованных в работе _____ 15 _____

5 В том числе источников по годам

Отечественных			Иностраных		
Последние 5 лет	От 5 до 10 лет	Более 10 лет	Последние 5 лет	От 5 до 10 лет	Более 10 лет
6	1	-	3	2	3

6 Использование информационных ресурсов Internet _____ Да, 9 _____
(Да, нет, число ссылок в списке литературы)

7 Использование современных пакетов компьютерных программ и технологий _____ (Указать, какие
именно, и в каком разделе работы)

Пакеты компьютерных программ и технологий	Параграф работы
OpenCV	2
ROS	3

8 Краткая характеристика полученных результатов _____ Реализованные алгоритмы в той или
иной мере справились с поставленной задачей слежения за объектом. В ходе сравнения были
выявлены достоинства и недостатки проверенных алгоритмов, а также были предложены
рекомендации по их использованию в зависимости от конкретных условий и требований. _____

9 Полученные гранты, при выполнении работы _____
(Название гранта)

10 Наличие публикаций и выступлений на конференциях по теме выпускной работы Нет
(Да, нет)

а) 1 _____
(Библиографическое описание публикаций)
2 _____
3 _____

б) 1 _____
(Библиографическое описание выступлений на конференциях)
2 _____
3 _____

Студент Терентьев Р.А. _____
(ФИО) (подпись)

Руководитель ВКР Шаветов С.В. _____
(ФИО) (подпись)

“ _____ ” _____ 20__ г.

Введение

В настоящее время очувствление роботов (robotic sensing) – одно из самых перспективных направлений развития робототехники. Очувствление предполагает снабжение роботов различными датчиками, имитирующими органы чувств человека: зрение, слух, осязание и др., что делает роботов более похожими на людей, а также повышает их адаптивность к изменяющимся условиям окружающей среды. За ориентацию робота в пространстве отвечает такая научная дисциплина как техническое зрение (computer vision).

Наиболее близким к человеческому зрению является бинокулярная система – две камеры, расположенные на некотором расстоянии друг от друга. Такая система может с помощью математических вычислений оценивать глубину изображения и определять расстояние до объекта. Однако согласование двух камер является довольно сложной задачей и не всегда оправдано, поэтому в большинстве случаев можно обойтись монокулярной системой – одной камерой.

По сравнению с людьми компьютеры обладают большей вычислительной мощностью и оперативной памятью. Но такая простая для человека задача распознавания объектов на изображении для компьютера является совсем не тривиальной, потому как человеческий мозг с рождения обучается различать объекты друг от друга [1]. Невозможность автоматического анализа сцен даже на уровне ребёнка вынуждает разработчиков двигаться постепенно, разбивая задачи технического зрения на ряд подзадач меньшей сложности, таких как корректировка изображений, их стандартизация (нормализация), выделение характерных элементов и т.п. [2]. В связи с этим существует множество методов, позволяющих, в зависимости от задачи, программно реализовать какую-либо способность человеческого зрения.

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № дубл.	Подп. и дата
Изм.	Лист	№ докум.	Подп.	Дата

ФСЧУИР.107.R3442.001 ПЗ

Лист 5

Копировал Формат А4

В данной работе поставлена задача произвести сравнение различных алгоритмов технического зрения в задачах слежения за объектом. Необходимо:

- Провести обзор существующих алгоритмов обнаружения и локализации объектов;
- Реализовать выбранные алгоритмы в виде комплекса программ;
- Проверить работу полученных алгоритмов в реальных условиях на базе мобильной робототехнической платформы Robotino;
- Сравнить результаты испытаний по различным характеристикам: быстродействие, робастность, зависимость от освещения и т.д.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	ФСЧУИР.107.R3442.001 ПЗ					Лист
										6
Изм.	Лист	№ докум.	Подп.	Дата						

1 Обзор существующих алгоритмов обнаружения и локализации

1.1 Цветовой фильтр

Если объект имеет отличный от фона цвет, то целесообразно упростить изображение, сделав его бинарным. Для этого заменим цвет пикселей объекта на белый, а цвет пикселей фона – на чёрный (см. рисунок 1.1). Чтобы обозначить, какие пиксели принадлежат объекту, используют цветовой фильтр. Он представляет собой набор числовых промежутков, применяемых к цветовой модели изображения.

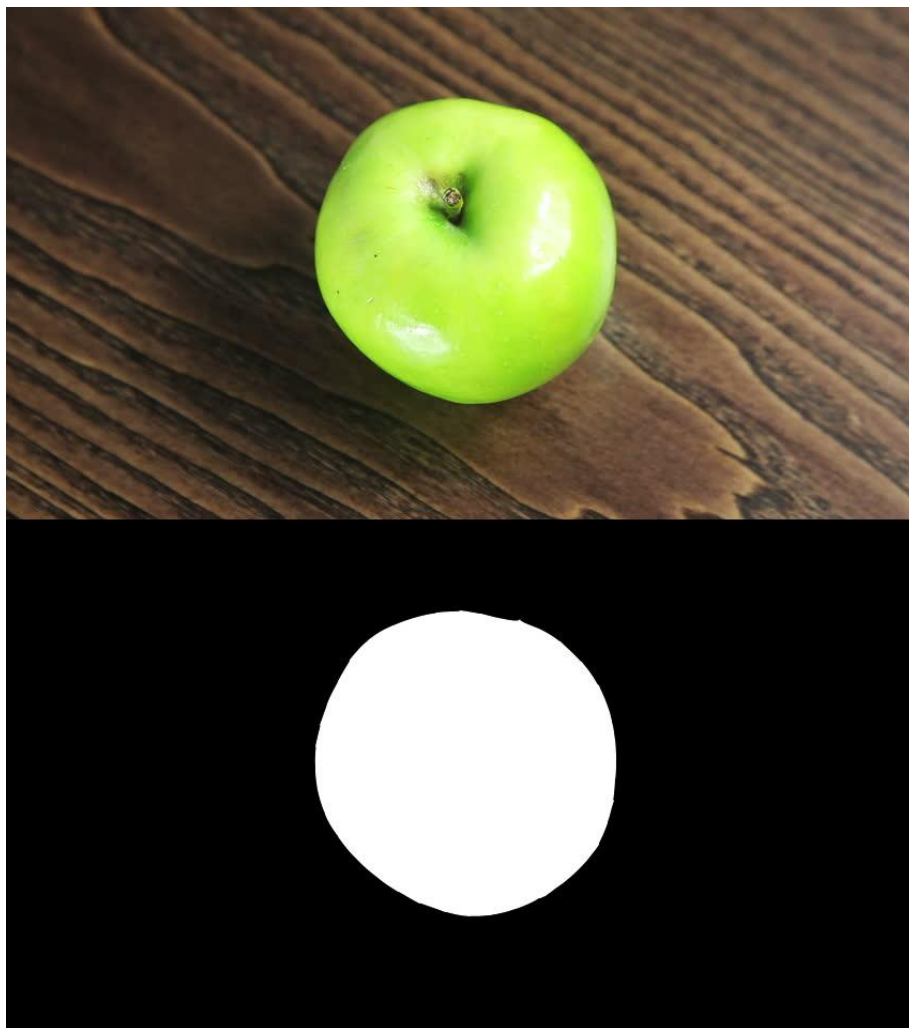


Рисунок 1.1 – Цветовая маска
(сверху – исходное изображение, снизу – его бинарная маска)

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № дубл.	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата

Лист
7

ФСЧУИР.107.R3442.001 ПЗ

Копировал
Формат А4

Цветовая модель – это математический концепт, позволяющий описать изображение в виде трёхмерной матрицы, в которой каждому пикселю соответствует несколько (как правило, три) значений цветовых компонент. Эти компоненты различны для каждой модели. Цветовые модели можно разделить на три группы:

- аппаратно-зависимые – модели данной группы описывают цвет применительно к конкретному воспроизводящему устройству, например, монитору (RGB, CMYK);
- аппаратно-независимые – эта группа моделей используется для того, чтобы дать однозначную информацию о цвете (XYZ, Lab);
- перцепционные – эти модели базируются на особенностях восприятия человека (HSB, HSV, HSL) [3].

Рассмотрим основные характеристики наиболее известных цветовых моделей.

1.1.1 Цветовая модель RGB

Это аддитивная модель, которая описывается как комбинация трёх цветов: красного, зелёного и синего (Red, Green, Blue). Данные цвета принято называть основными, так как человеческий глаз лучше всего различает именно их. RGB является базовой моделью для большинства технических устройств. Для компьютера значения цветовых компонент лежат в диапазоне [0; 255].

RGB можно представить в виде куба, где чёрный цвет – это начало координат, а движение по осям означает добавление одного из основных цветов к чёрному. При одновременном добавлении двух основных цветов получаем дополнительные: фиолетовый, жёлтый, голубой. Смешение всех цветов даёт белый цвет (см. рисунок 1.2).

Инв. № подл.	Подп. и дата		Взам. инв. №		Инв. № докл.		Подп. и дата		
<h3>1.1.1 Цветовая модель RGB</h3>									
<p>Это аддитивная модель, которая описывается как комбинация трёх цветов: красного, зелёного и синего (Red, Green, Blue). Данные цвета принято называть основными, так как человеческий глаз лучше всего различает именно их. RGB является базовой моделью для большинства технических устройств. Для компьютера значения цветовых компонент лежат в диапазоне [0; 255].</p>									
<p>RGB можно представить в виде куба, где чёрный цвет – это начало координат, а движение по осям означает добавление одного из основных цветов к чёрному. При одновременном добавлении двух основных цветов получаем дополнительные: фиолетовый, жёлтый, голубой. Смешение всех цветов даёт белый цвет (см. рисунок 1.2).</p>									
					ФСЧУИР.107.R3442.001 ПЗ				Лист
									8
Изм.	Лист	№ докум.	Подп.	Дата					



К достоинствам данной цветовой модели можно отнести линейность и простоту реализации. Недостатком является зависимость от устройства вывода, из-за этого одно изображение на разных мониторах может выглядеть по-разному.

1.1.2 Цветовая модель HSV

HSV является нелинейным преобразованием модели RGB. Данная модель описывается следующими параметрами:

- Оттенок (Hue) или цветовой тон – обозначает, собственно, цвет;
- Насыщенность (Saturation) – определяет чистоту цвета, т.е. степень визуального отличия от серого цвета;
- Значение цвета (Value) или яркость – показывает освещённость цвета, т.е. степень визуального отличия от чёрного цвета.

Так как модель HSV не является базовой, для её использования необходимо осуществить преобразование цветовых компонент из RGB по следующим формулам [4]:

$$H = \begin{cases} \frac{30(G - B)}{MAX - MIN}, & \text{при } MAX = R \\ 60 + \frac{30(B - R)}{MAX - MIN}, & \text{при } MAX = G \\ 120 + \frac{30(R - G)}{MAX - MIN}, & \text{при } MAX = B \end{cases}$$

ФСЧП.107.R3442.001 ПЗ

$$S = \begin{cases} 0, & \text{при } MAX = 0 \\ \frac{255(MAX - MIN)}{MAX}, & \text{при } MAX \neq 0 \end{cases}$$

$$V = MAX$$

В результате преобразования получим: H в диапазоне [0; 179] (360 значений не описать одним байтом), S, V – [0; 255].

HSV по сравнению с RGB ближе к человеческому восприятию цветов; можно провести аналогию: тон – длина волны света, насыщенность – интенсивность волны, а яркость – количество света. Однако имеется тот же недостаток, связанный с зависимостью от устройства вывода.

1.1.3 Цветовая модель Lab

Эта модель однозначно определяет цвет и не зависит от параметров устройства, поэтому может применяться в качестве посредника при конвертировании между другими цветовыми моделями. Lab описывается следующими параметрами:

- L – светлота (от чёрного к белому);
- a – цветовой компонент (от зелёного к красному);
- b – цветовой компонент (от синего к жёлтому).

Все параметры должны быть перпендикулярны друг другу. Одно из возможных представлений модели Lab изображено на рисунке 1.4.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	ФСЧУИР.107.R3442.001 ПЗ					Лист
										11
					Изм.	Лист	№ докум.	Подп.	Дата	


$$RGB \rightarrow XYZ: \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0,412453 & 0,357580 & 0,180423 \\ 0,212671 & 0,715160 & 0,072169 \\ 0,019334 & 0,119193 & 0,950227 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1.2)$$

$$XYZ \rightarrow Lab: \quad \left\{ X = \frac{X}{0.950456}, \quad Z = \frac{Z}{1.088754} \right\}$$

$$L = \begin{cases} 2,55(116 * Y^{1/3} - 16), & \text{при } Y > 0,008856 \\ 2,55 * 903,3 * Y, & \text{при } Y \leq 0,008856 \end{cases}$$

$$a = 500(f(X) - f(Y)) + 128$$

$$b = 200(f(Y) - f(Z)) + 128$$

$$\text{где } f(t) = \begin{cases} t^{1/3}, & \text{при } t > 0,008856 \\ 7,787t + \frac{16}{116}, & \text{при } t \leq 0,008856 \end{cases}$$

В результате преобразования получим: L, a, b в диапазоне $[0; 255]$.

1.2 Метод ключевых точек

Если имеется эталонное изображение объекта, то можно осуществлять его поиск на других изображениях с помощью сопоставления ключевых (или особых) точек (см. рисунок 1.5). Термин «ключевая точка» трактуется по-разному:

- Каждая такая точка имеет характерную окрестность, т.е. отличается от всех других точек в некоторой окрестности;
- Такая точка объекта, которая с большой долей вероятности будет найдена на другом изображении этого же объекта;
- Место резкого перепада градиента на изображении по x и по y (угловая точка) [5].

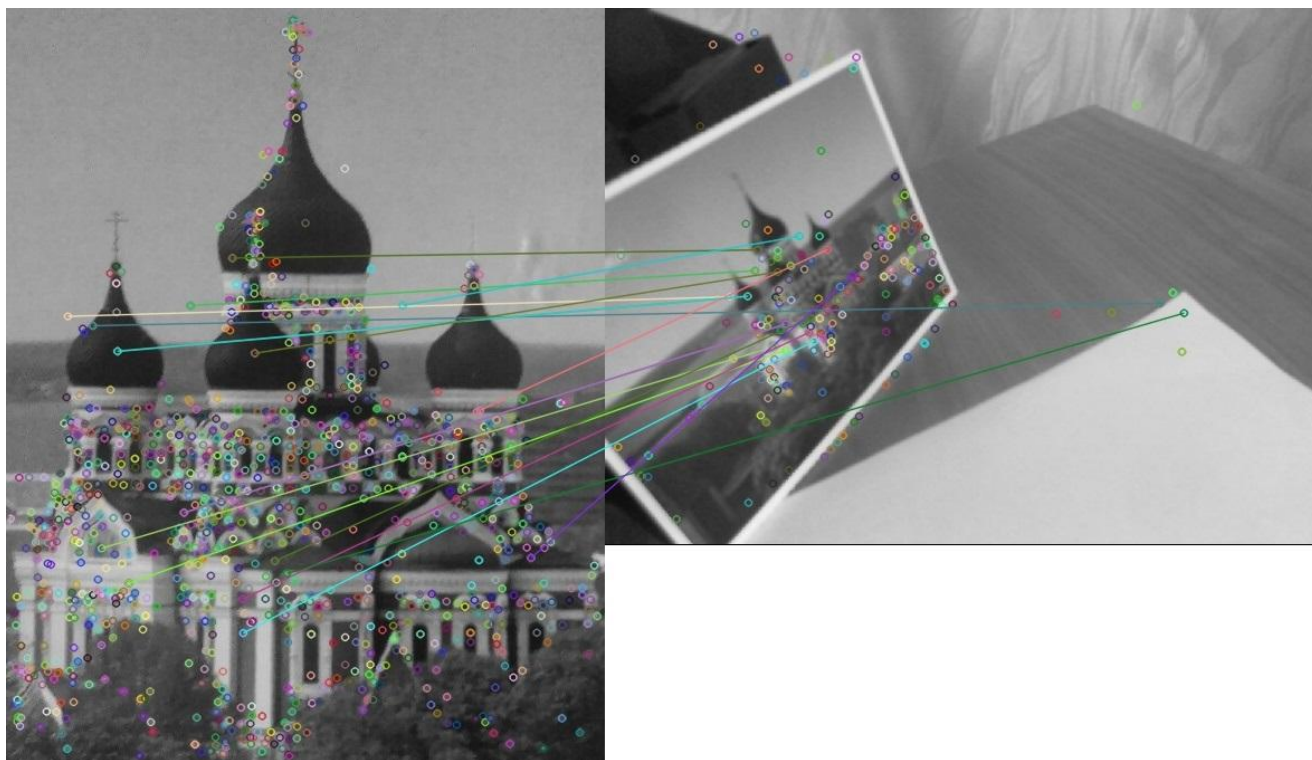


Рисунок 1.5 – Сопоставление ключевых точек (слева – эталонное изображение объекта, справа – изображение, где нужно обнаружить объект)

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата
Изм.	Лист	№ докум.	Подп.	Дата

ФСЧУИР.107.R3442.001 ПЗ

Копировал _____ Формат А4

Лист	13
------	----

Основная идея метода заключается в том, что изображение заменяется некоторой моделью, состоящей из его ключевых точек. Так как нет строгого определения ключевой особенности изображения, существует множество алгоритмов, и каждый определяет особенности по-своему, но везде используются три составляющие:

- Детектор – метод нахождения точек на изображении, позволяющий определить их координаты;
- Дескриптор – метод описания найденных точек, выделяющий какую-либо точку из множества остальных через описание окружающих областей;
- Матчер – метод построения соответствий между наборами найденных и описанных точек на разных изображениях.

Рассмотрим основные характеристики некоторых алгоритмов нахождения и описания ключевых точек.

1.2.1 Алгоритм SIFT

Исходя из названия «Масштабно-инвариантное преобразование особенностей» (Scale-Invariant Feature Transform), данный алгоритм направлен на решение проблемы детектирования ключевых точек при масштабировании изображения. SIFT содержит следующие шаги:

1) Обнаружение экстремумов в масштабном пространстве

При изменении размера изображения угол может превратиться в кривую, и в данном окне его уже не обнаружить. Для этого используется масштабная фильтрация. При этом должен быть найден Лапласиан Гауссиана (LoG) для изображения с различными значениями σ (σ действует как параметр масштабирования). LoG действует как детектор блобов (каплевидных окрестностей), который обнаруживает блобы разных размеров из-за

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	1.2.1 Алгоритм SIFT
<p>Исходя из названия «Масштабно-инвариантное преобразование особенностей» (Scale-Invariant Feature Transform), данный алгоритм направлен на решение проблемы детектирования ключевых точек при масштабировании изображения. SIFT содержит следующие шаги:</p>					
<p>1) Обнаружение экстремумов в масштабном пространстве</p>					
<p>При изменении размера изображения угол может превратиться в кривую, и в данном окне его уже не обнаружить. Для этого используется масштабная фильтрация. При этом должен быть найден Лапласиан Гауссиана (LoG) для изображения с различными значениями σ (σ действует как параметр масштабирования). LoG действует как детектор блоков (каплевидных окрестностей), который обнаруживает blobs разных размеров из-за</p>					
Изм.	Лист	№ докум.	Подп.	Дата	ФСЧУИР.107.R3442.001 ПЗ
					Лист
					14

изменения σ . Таким образом, мы можем найти локальные максимумы по масштабу и пространству, которые дают нам список значений (x, y, σ) , что означает, что существует потенциальная ключевая точка в точке (x, y) в масштабе σ .

Но LoG достаточно требователен, поэтому SIFT использует разность Гауссианов (DoG), которая является приближением LoG. DoG представляет из себя разность гауссова размытия изображения с двумя разными σ . Этот процесс выполняется для разных октав изображения в гауссовой пирамиде (см рисунок 1.6).

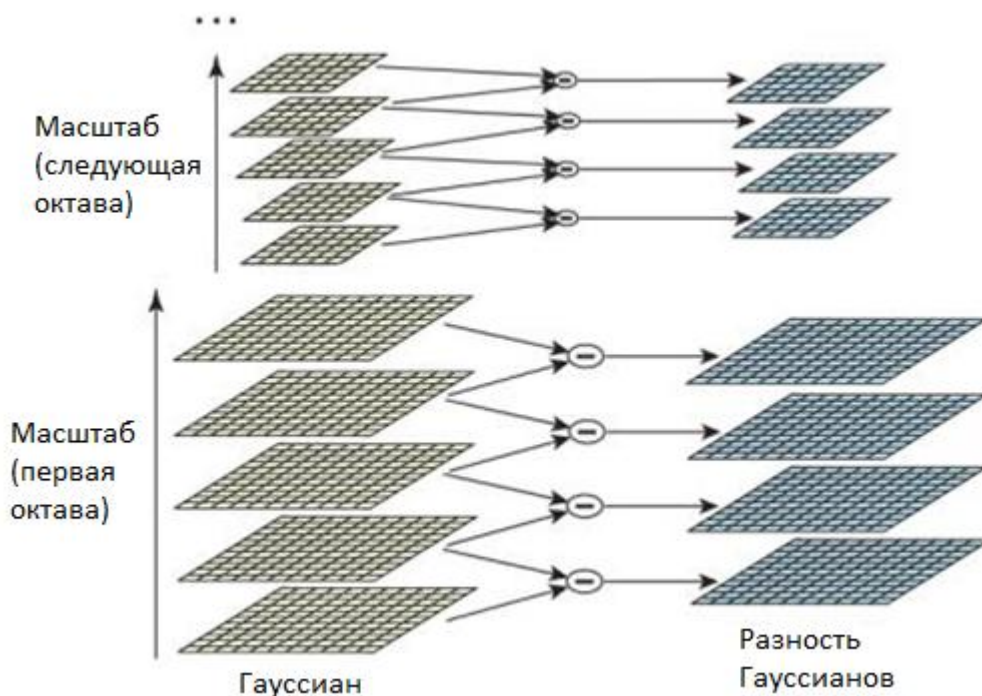


Рисунок 1.6 – Разность Гауссианов

Как только DoG найдена, изображения исследуются на локальные экстремумы в масштабе и пространстве. Например, один пиксель в изображении сравнивается с его 8 соседями, а также с 9 пикселями в следующем масштабе и 9 пикселями в предыдущих масштабах. Если это локальный экстремум, то это потенциальная ключевая точка.

Инд. № подл.	Подп. и дата
Взам. инд. №	Инд. № дубл.
Подп. и дата	
Инд. № подл.	

Изм.	Лист	№ докум.	Подп.	Дата

2) Локализация ключевых точек

Как только координаты потенциальных ключевых точек найдены, их необходимо уточнить. Для этого используется разложение ряда Тейлора, и если интенсивность этих экстремумов меньше порогового значения, они отклоняются. У DoG высокий отклик на грани, поэтому они также должны быть удалены. Из углового детектора Харриса мы знаем, что для граней одно собственное значение больше другого, и если это отношение больше порогового значения, эта ключевая точка отбрасывается. Таким образом, на данном этапе устраняются малоконтрастные ключевые точки и граничные точки.

3) Назначение ориентации

Каждой ключевой точке назначается ориентация для достижения инвариантности к вращению изображения. Вокруг ключевой точки берется окрестность в зависимости от масштаба, и в этой области вычисляются величина и направление градиента. Создается гистограмма ориентации с 36 ячейками, охватывающими 360 градусов. При расчёте ориентации учитывается самый высокий пик в гистограмме, а также любой пик выше 80%. Так создаются ключевые точки с одинаковым расположением и масштабом, но в разных направлениях. Это способствует стабильности сопоставления.

4) Описание ключевых точек

Теперь создадим дескриптор ключевой точки. Для этого берётся окрестность 16x16 вокруг неё. Она разделяется на 16 подблоков размером 4x4, и для каждого подблока создается гистограмма ориентации. Она представлена в виде вектора для формирования дескриптора. В дополнение к этому, предпринимаются некоторые меры для достижения устойчивости к изменениям освещенности, вращению и т.д. [6]

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № дубл.	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата

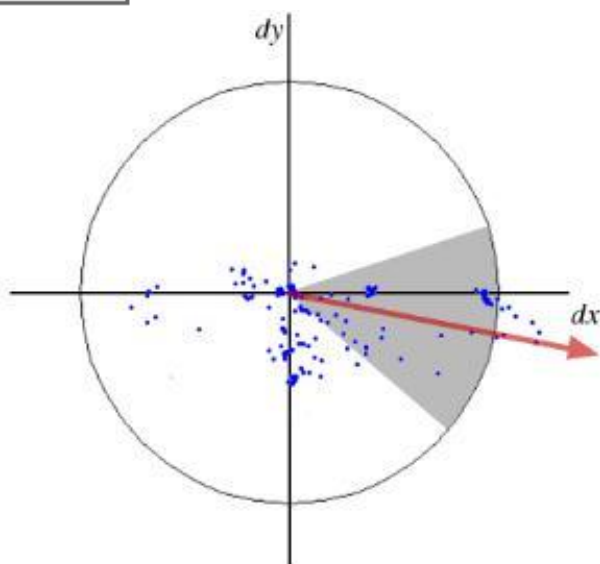


Рисунок 1.8 – Назначение ориентации

Для описания ключевых точек SURF использует вейвлет-отклики в горизонтальном и вертикальном направлении. Вокруг ключевой точки берется окрестность размером 20x20. Она разделяется на регионы 4x4, и для каждого региона берутся горизонтальные и вертикальные отклики, формирующие вектор [7].

1.2.3 Алгоритм ORB

«Oriented FAST and Rotated BRIEF» базируется на двух других алгоритмах: детекторе «Особенности ускоренного испытания сегмента» (Features from Accelerated Segment Test) и дескрипторе «Бинарные робастные независимые элементарные особенности» (Binary Robust Independent Elementary Features); а также содержит множество модификаций для повышения производительности.

Для нахождения ключевых точек сначала используется FAST, а затем угловая мера Харриса, чтобы найти верхние N точек среди них. Но FAST сам по себе не учитывает ориентацию, поэтому ORB вычисляет взвешенный по интенсивности центроид пятна с углом в центре. Направление вектора от этой угловой точки к центроиду дает ориентацию.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

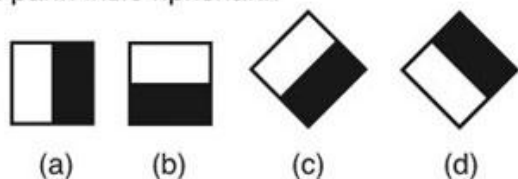
Изм.	Лист	№ докум.	Подп.	Дата

1.3 Метод Виолы-Джонса

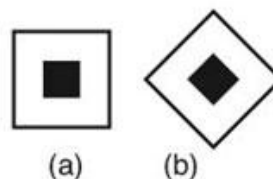
Данный метод создавался и чаще всего используется для обнаружения лиц, но может детектировать и другие объекты. Он основан на машинном обучении, где каскадная функция обучается из множества положительных и отрицательных изображений, а затем она используется для обнаружения объектов на других изображениях.

Изначально алгоритму требуется много положительных изображений (с лицами) и отрицательных изображений (без лиц) для обучения классификатора. Затем, чтобы извлечь из него особенности, используются признаки Хаара. Каждый признак представляет собой одно значение, полученное путем вычитания суммы пикселей под белым прямоугольником из суммы пикселей под черным прямоугольником (см. рисунок 1.10).

1. Граничные признаки



3. «Центральные» признаки



2. Линейные признаки

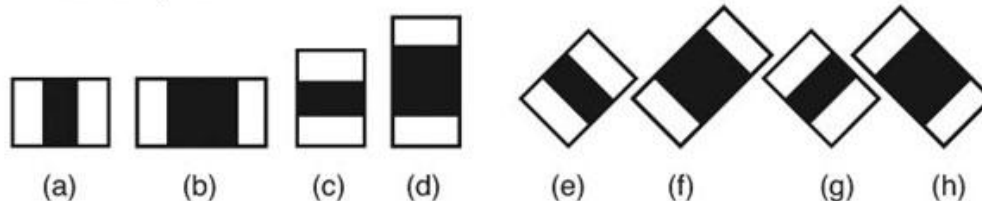


Рисунок 1.10 – Признаки Хаара

Теперь все возможные размеры и расположение каждого признака должны использоваться для расчета множества особенностей. Но на самом деле нет смысла применять каждый признак, т.к. это требует много времени. Для выбора лучших признаков используется метод адаптивного усиления (AdaBoost). Он находит наилучший порог, который классифицирует лица на положительные и отрицательные, и выбирает признаки с минимальным уровнем ошибок.

Инд. № подл.	Подп. и дата
Взам. инд. №	Инд. № докл.
Изм.	Лист
№ докум.	Подп.
Дата	

Также нет смысла применять признаки ко всему изображению, т.к. на нём могут быть регионы без лиц. Для этого было введено понятие каскада классификаторов. Вместо применения всех признаков ко всему изображению, они группируются по различным каскадам и применяются по одному. Если регион не прошёл первый каскад, оставшиеся признаки на нем не рассматриваются. Если прошёл, применяются следующие каскады. Регион, который проходит все каскады, является областью лица. На рисунке 1.11 показан пример обнаружения лица и глаз с помощью метода Виолы-Джонса.

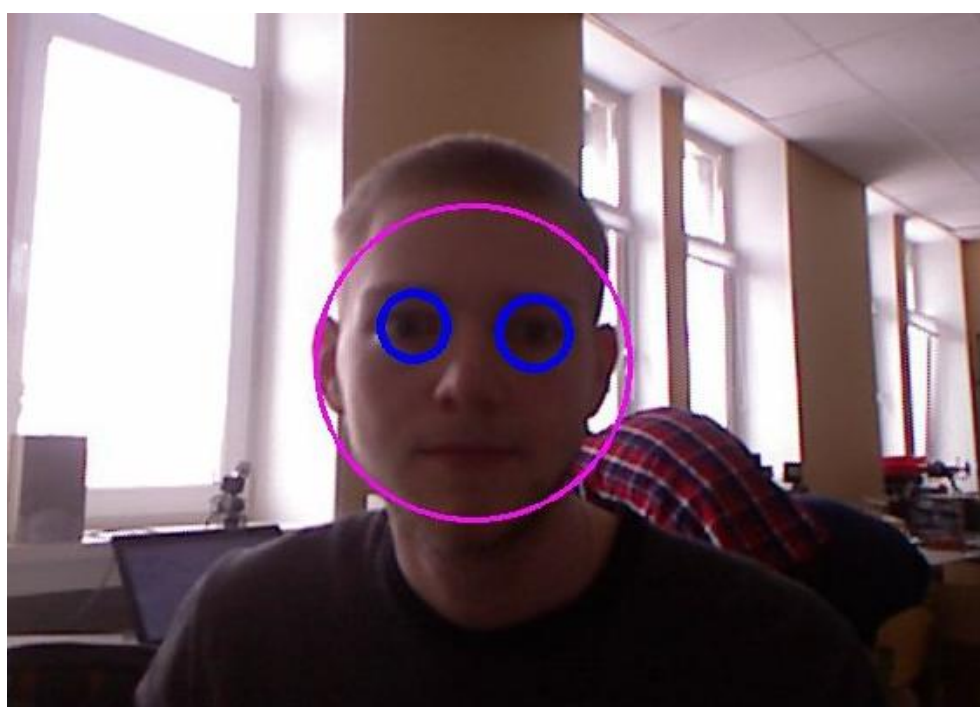


Рисунок 1.11 – Пример обнаружения лица и глаз

Хорошо обученный классификатор обладает наибольшей точностью среди всех известных методов. Но сам процесс обучения является недостатком метода, так как требует большого количества положительных и отрицательных примеров для создания выборки [9].

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № докл.	Подп. и дата
Изм.	Лист	№ докум.	Подп.	Дата

ФСЧУИР.107.R3442.001 ПЗ

Копировал

Лист

21

Формат А4

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.1)$$

где x – расстояние от начала координат по горизонтальной оси, y – расстояние по вертикальной оси, а σ – стандартное отклонение гауссовского распределения. При применении в двух измерениях эта формула дает поверхность, контуры которой являются концентрическими окружностями с распределением Гаусса от центральной точки. Значения этого распределения используются для построения матрицы свертки, которая применяется к исходному изображению. Для каждого нового значения пикселя определяется среднее арифметическое взвешенное в окрестности пикселя. Значение исходного пикселя получает самый тяжелый вес (с самым высоким гауссовым значением), а соседние пиксели получают меньший вес по мере увеличения их расстояния до исходного пикселя. В результате применения фильтра получим слегка размытое изображение.

3) Цветовая фильтрация

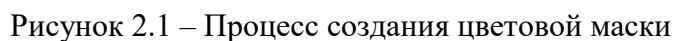
Для вычисления цветовых параметров объекта слежения была реализована функция с обратной связью через компьютерную мышь. Функция работает следующим образом: пользователь кликает мышью на какой-либо пиксель объекта, и значения его цветовых компонент запоминаются в качестве цветового диапазона объекта. Далее выбирается другой пиксель, и, если его параметры выходят за пределы диапазона, то он расширяется. Для ускорения фильтрации учитывается также цвет восьми соседей указанного пикселя. Наконец с помощью функции `inRange` создаётся бинарное изображение, где пиксели в вычисленном цветовом диапазоне окрашиваются в белый цвет, а пиксели вне диапазона – в чёрный. На рисунке 2.1 показан процесс создания цветовой маски.

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № докл.	Подп. и дата
Изм.	Лист	№ докум.	Подп.	Дата

ФСЧУИР.107.R3442.001 ПЗ

Копировал _____ Формат А4

Лист	23
------	----



4) Морфологические преобразования

Рисунок 2.2 – Морфологические преобразования
(слева – открытие, справа – закрытие)

5) Нахождение контура

Контур – это кривая, соединяющая пограничные точки одинакового цвета. Функция `findContours` определяет все контуры на изображении, поэтому необходимо учесть возможность попадания в область индикации других объектов. С помощью функции `contourArea`, которая через теорему Грина находит площадь области, ограниченной контуром, выделяем наибольший контур.

6) Вычисление площади

Также площадь объекта можно найти с помощью моментов. Момент изображения – это суммарная характеристика пятна, представляющая собой совокупность всех пикселей этого пятна. При этом имеется множество подвидов моментов, характеризующие разные свойства изображения. Момент нулевого порядка представляет собой сумму всех пикселей пятна, что эквивалентно его площади [11]:

$$S = M_{00} = \sum_{x=0}^w \sum_{y=0}^h f(x, y), \quad \text{где} \quad (2.2)$$

$$f(x, y) = \begin{cases} 1, & \text{если пиксель белый} \\ 0, & \text{если пиксель чёрный} \end{cases}$$

7) Определение координат центра

Для нахождения центра объекта используем моменты первого порядка:

$$M_{10} = \sum \sum x f(x, y), \quad M_{01} = \sum \sum y f(x, y) \quad (2.3)$$

$$x_c = M_{10}/M_{00}, \quad y_c = M_{01}/M_{00}$$

Рисунок 2.3 иллюстрирует результат работы цветового фильтра.

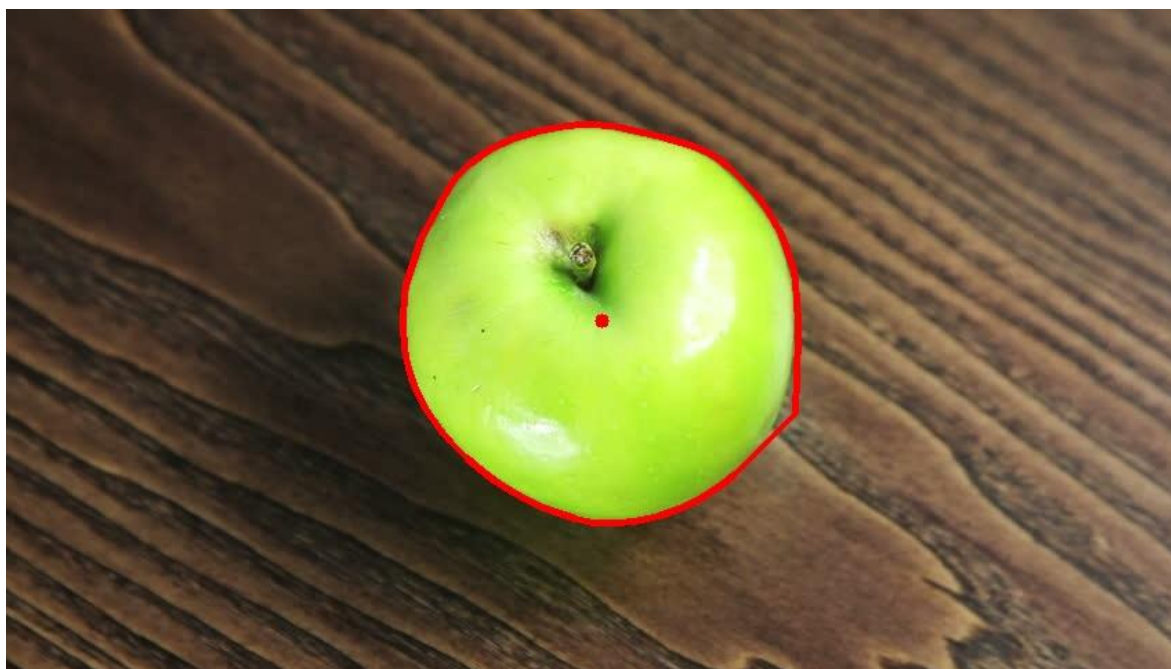


Рисунок 2.3 – Результат работы цветового фильтра

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № докл.	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата

2.2 Метод ключевых точек

На вход программы подаётся 2 изображения: первое, называемое эталонным, должно содержать только объект слежения без искажений; второе – это изображение или видеопоток, на котором программа будет искать объект. Дальнейшие действия таковы:

1) Преобразование цветовой модели

Полутоновое изображение имеет те же особые точки, что и цветное, однако занимает в 3 раза меньше памяти и быстрее обрабатывается. Поэтому целесообразно для обоих изображений совершить преобразование из RGB в оттенки серого:

$$RGB \rightarrow Gray: Y = 0,299 * R + 0,587 * G + 0,114 * B \quad (2.4)$$

2) Выбор алгоритма, нахождение и описание ключевых точек

С помощью функций `xfeatures.2d.SIFT_create` / `xfeatures.2d.SURF_create` / `ORB_create` выбирается один из алгоритмов, описанных в п.п. 1.2.1 – 1.2.3. Для одновременного нахождения и описания ключевых точек используется функция `detectAndCompute`.

3) Поиск совпадений

В качестве метода построения соответствий ключевых точек на обоих изображениях будем использовать метод полного перебора (Brute-Force Matcher). Он сопоставляет дескриптор каждой точки в первом наборе со всеми другими точками во втором наборе с использованием вычисления расстояния. Далее с помощью функции `knnMatch` запоминаются k ближайших совпадений (в нашем случае $k = 2$).

Формат А4

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} * \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \quad (2.5)$$

При использовании функции findHomography ложные совпадения могут сильно повлиять на гомографию. Для решения этой проблемы применяется метод RANSAC (Согласование случайных выборок), который принимает во внимание лишь те совпадения, которые обеспечивают удовлетворительную оценку, а выбросы не учитывает [12].

б) Перспективное преобразование

Наконец, возьмём 4 точки по краям эталонного изображения и с помощью функции perspectiveTransform отобразим их на второе изображение через вычисленную матрицу гомографии (см. рисунок 2.5). Соединив точки на втором изображении по периметру, получим контур найденного объекта.

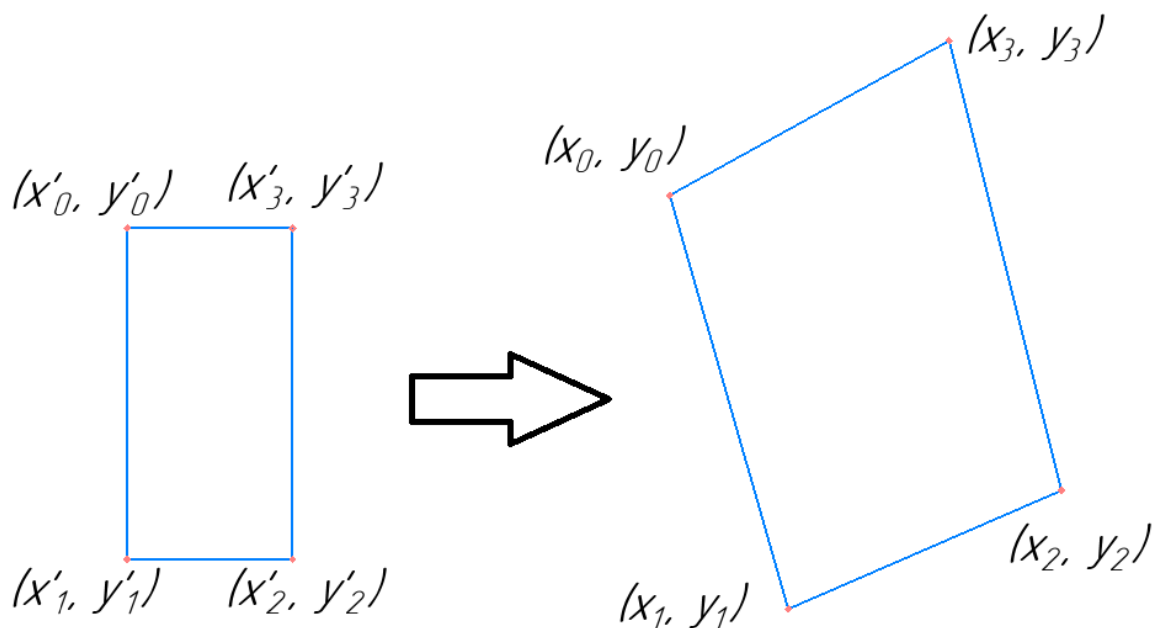


Рисунок 2.5 – Перспективное преобразование

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата
<div><div><div><div><div>(x'_0, y'_0)</div><div>(x'_3, y'_3)</div></div><div><div>(x'_1, y'_1)</div><div>(x'_2, y'_2)</div></div></div><div><div></div><div></div></div><div><div>(x_0, y_0)</div><div>(x_1, y_1)</div><div>(x_2, y_2)</div><div>(x_3, y_3)</div></div></div></div> <p>Рисунок 2.5 – Перспективное преобразование</p>				
ФСЧУИР.107.R3442.001 ПЗ				
Изм.	Лист	№ докум.	Подп.	Дата

Лист
28

7) Вычисление площади

Площадь объекта можно приблизительно найти, перемножив две смежные грани контура, а их длины посчитаем через координаты:

$$S = a * b, \text{ где} \quad (2.6)$$

$$a = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}$$

$$b = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

8) Определение координат центра

Центром объекта приблизительно можно считать центр диагонали между чётными углами контура:

$$(x_c, y_c) = ((x_0, y_0) + (x_2, y_2))/2 \quad (2.7)$$

Рисунок 2.6 иллюстрирует результат работы метода ключевых точек.



Рисунок 2.6 – Результат работы метода ключевых точек

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата
Изм.	Лист	№ докум.	Подп.	Дата

Лист 29

Формат А4

4 Сравнение алгоритмов

Результаты сравнения представлены в Таблице 4.1. Частота кадров и быстродействие представлены в количественных единицах, остальное – в качественных. При этом зависимости обозначают недостатки алгоритмов, а робастность – устойчивость к помехам.

Таблица 4.1 – Сравнение алгоритмов

Алгоритм	Цветовой фильтр			Метод ключевых точек		
	RGB	HSV	Lab	SIFT	SURF	ORB
FPS	25			8	10	16
Быстродействие, мс	40			125	100	62,5
Зависимость от освещения	высокая	средняя	низкая	низкая		
Зависимость от масштабирования	низкая			низкая	низкая	высокая
Зависимость от вращения	низкая			низкая		
Зависимость от перекрытия	средняя	низкая	низкая	низкая	низкая	средняя
Робастность	низкая	высокая	высокая	высокая	средняя	средняя

По данным таблицы можно составить некоторые рекомендации для случаев, когда заранее известны характеристики объекта слежения:

- Среди цветowych фильтров по всем параметрам выигрывает Lab;
- Среди методов ключевых точек выигрыш в скорости даёт проигрыш в точности, и наоборот;
- Для быстродвижущихся объектов следует использовать Lab (если объект имеет характерный цвет) или ORB (если объект имеет неоднородную структуру);
- Если скорость не важна, лучше использовать SIFT как самый точный;
- Если объект может сильно отдалиться, не стоит пользоваться ORB.

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № докл.	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата

Заключение

В ходе работы был проведён обзор существующих алгоритмов обнаружения и локализации объектов на видеоизображении. Два из них (цветовой фильтр и метод ключевых точек) были программно реализованы и проверены с помощью робота Robotino.

Реализованные алгоритмы в той или иной мере справились с поставленной задачей слежения за объектом. В ходе сравнения были выявлены достоинства и недостатки проверенных алгоритмов, а также были предложены рекомендации по их использованию в зависимости от конкретных условий и требований.

Дальнейшие работы в данном направлении могут быть связаны с автоматизацией алгоритмов. Например, чтобы алгоритм цветового фильтра строил бинарную маску без участия пользователя. Или, для метода ключевых точек, чтобы в качестве эталонного изображения выступал один из кадров видеопотока [15].

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	ФСЧУИР.107.R3442.001 ПЗ					Лист
										33
Изм.	Лист	№ докум.	Подп.	Дата						

Список использованных источников

1. Компания Яндекс – Технологии – Компьютерное зрение
[Электронный ресурс]: электрон. статья, 2014,
URL: https://yandex.ru/company/technologies/computer_vision
(дата обращения 11.04.2019)
2. Обработка и анализ изображений в задачах машинного зрения:
Учебное издание:// Ю. В. Визильтер, С. Ю. Желтов, А. В. Бондаренко и
др. / Под ред. А. К. Розанова – М.: Физматкнига, 2010. – 672 с
3. Что такое цветовая модель [Электронный ресурс]: электрон. статья,
2015, URL: dopechatnik.ru/cvetovye-modeli/ (дата обращения 14.04.2019)
4. OpenCV Documentation [Электронный ресурс]: электрон. статья, 2017,
URL: <https://docs.opencv.org/3.3.1/> (дата обращения 15.03.2019)
5. Распознавание плоских объектов OpenCV 2.4 / Хабр [Электронный
ресурс]: электрон. статья, 2012, URL: <https://habr.com/ru/post/155651/>
(дата обращения 21.04.2019)
6. Lowe D. Distinctive image features from scale-invariant keypoints IJCV 60,
P. 91–110, 2004
7. Bay H., Ess A., Gool L.V., Tuytelaars T. SURF: speed up robust features
CVIU, Vol.110, No.3, 2008. – pp. 346-359
8. Bradski G, Konolige K., Rabaud V., Rublee E. ORB: an efficient alternative
to SIFT or SURF CVPR'11. 2011. P. 2564-2571
9. Jones M.J, Viola P. Rapid object detection using a boosted cascade of
simple features In Proceedings IEEE Conf. on Computer Vision and Pattern
Recognition. – 2001
10. НОУ ИНТУИТ | Лекция | Базовые операции обработки изображений
[Электронный ресурс]: электрон. статья, 2019, URL:
<https://www.intuit.ru/studies/courses/10621/1105> (дата обращения
22.04.2019)

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата
Изм.	Лист	№ докум.	Подп.	Дата
<p>ФСЧУИР.107.Р3442.001 ПЗ</p>				Лист
				34

11. Image Moments – AI Shack [Электронный ресурс]: электрон. статья, 2010, URL: aishack.in/tutorials/image-moments/ (дата обращения 22.04.2019)
12. Homography Examples using OpenCV (Python/C++) | Learn OpenCV [Электронный ресурс]: электрон. статья, 2016, URL: <https://www.learnopencv.com/homography-examples-using-opencv-python-c/> (дата обращения 22.04.2019)
13. RobotinoWiki [Электронный ресурс]: электрон. статья, 2019, URL: <https://wiki.openrobotino.org> (дата обращения 25.04.2019)
14. ru – ROS Wiki [Электронный ресурс]: электрон. статья, 2015, URL: wiki.ros.org/ru (дата обращения 25.04.2019)
15. Блинников А.А, Бойков В.И., Быстров С.В., Николаев Н.А., Нуйя О.С. Правила оформления пояснительной записки и конструкторской документации. – СПб: Университет ИТМО, 2014 – 55 с.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № докл.	Подп. и дата	<p>ФСЧУИР.107.R3442.001 ПЗ</p>					Лист
										35
Изм.	Лист	№ докум.	Подп.	Дата						

Приложение А

Листинг программы colormask.py

```
#!/usr/bin/python
import cv2
import numpy as np
import rospy
from geometry_msgs.msg import Twist

def nothing(x):
    pass

rospy.init_node('mto')

pub = rospy.Publisher('cmd_vel', Twist, queue_size=10)
msg = Twist()

pt = 0
mmin = np.array([255, 255, 255])
mmax = np.array([0, 0, 0])
fl = 0

# mouse callback function
def masking(event,x,y,flags,param):
    global pt,mmin,mmax
    if event == cv2.EVENT_LBUTTONDOWN:
        for g in range(3):
            for h in range(3):
                pt = (imgtr[y+g-1,x+h-1])
                for i in range(3):
                    if pt[i] < mmin[i]:
                        mmin[i] = pt[i]
                    if pt[i] > mmax[i]:
                        mmax[i] = pt[i]

cap = cv2.VideoCapture(1)
cv2.namedWindow('imagetrans')
cv2.setMouseCallback('imagetrans',masking)
```

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № докл.	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата


```

        msg.angular.z = 0.1
    elif dArea > 40000:
        msg.linear.x = 0
        msg.angular.z = 0
    elif cx<295:
        msg.angular.z = 0.1
        msg.linear.x = 0
    elif cx>345:
        msg.angular.z = -0.1
        msg.linear.x = 0
    elif 295<cx<345:
        msg.angular.z = 0
        msg.linear.x = 0.02

    pub.publish(msg)
except:
    print('OpenCV_Error')
    continue

fps = int(cv2.getTickFrequency() / (cv2.getTickCount() - timer))
cv2.putText(img, "FPS : " + str(fps), (50,50),
cv2.FONT_HERSHEY_SIMPLEX, 0.75, (0,0,0), 2)
cv2.imshow('image',img)
cv2.imshow('imagerans',imgtr)
cv2.imshow('mask',mask)

k = cv2.waitKey(1) & 0xFF
if k == 32:
    fl = 1
elif k == ord('r'):
    fl = 0
    mmin = np.array([255, 255, 255])
    mmax = np.array([0, 0, 0])
elif k == 27:
    break

cap.release()
cv2.destroyAllWindows()

```

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № дубл.	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата

Приложение Б

Листинг программы keypoints.py

```
#!/usr/bin/python
import cv2
import numpy as np
import rospy
from geometry_msgs.msg import Twist

def nothing(x):
    pass

rospy.init_node('mto')

pub = rospy.Publisher('cmd_vel', Twist, queue_size=10)
msg = Twist()

x = np.array([0, 0, 0])
y = np.array([0, 0, 0])
fl = 0

img = cv2.imread('/home/administry/catkin_ws/src/diploma/scrabble.jpeg',
cv2.IMREAD_GRAYSCALE) # queryiamge
cap = cv2.VideoCapture(1)

# Features
Feat = cv2.xfeatures2d.SIFT_create()
#Feat = cv2.xfeatures2d.SURF_create(300)
#Feat = cv2.ORB_create(nfeatures=1500)
kp_image, desc_image = Feat.detectAndCompute(img, None)

# Feature matching
Matcher = cv2.BFMatcher()
#Matcher = cv2.BFMatcher(cv2.NORM_HAMMING)

while True:
    timer = cv2.getTickCount()
    _, frame = cap.read()
```

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № дудл.	Подп. и дата

```

grayframe = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) # trainimage

kp_grayframe, desc_grayframe = Feat.detectAndCompute(grayframe, None)
matches = Matcher.knnMatch(desc_image, desc_grayframe, k=2)

good_points = []
for m, n in matches:
    if m.distance < 0.8*n.distance:
        good_points.append(m)

img3 = cv2.drawMatches(img, kp_image, grayframe, kp_grayframe,
good_points, grayframe)

# Homography
fps = int(cv2.getTickFrequency() / (cv2.getTickCount() - timer))
cv2.putText(grayframe, "FPS : " + str(fps), (50,50),
cv2.FONT_HERSHEY_SIMPLEX, 0.75, (0,0,0), 2)
if len(good_points) > 9:
    query_pts = np.float32([kp_image[m.queryIdx].pt for m in
good_points]).reshape(-1, 1, 2)
    train_pts = np.float32([kp_grayframe[m.trainIdx].pt for m in
good_points]).reshape(-1, 1, 2)

    matrix, mask = cv2.findHomography(query_pts, train_pts, cv2.RANSAC,
5.0)

# Perspective transform
h, w = img.shape
pts = np.float32([[0, 0], [0, h], [w, h], [w, 0]]).reshape(-1, 1, 2)
try:
    dst = np.int32(cv2.perspectiveTransform(pts, matrix))
    for i in range(3):
        x[i] = dst[i][0][0]
        y[i] = dst[i][0][1]
    a = ((x[1]-x[0])**2 + (y[1]-y[0])**2)**0.5
    b = ((x[1]-x[2])**2 + (y[1]-y[2])**2)**0.5
    dArea = a*b
    #print(dArea)
    cx = int((x[0]+x[2])/2)

```

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № дубл.	Подп. и дата

```
cy = int((y[0]+y[2])/2)
cv2.circle(grayframe, (cx, cy), 5, (0,0,0), -1)
```

```
if fl == 1:
    if dArea < 4000:
        msg.linear.x = 0
        msg.angular.z = 0.1
    elif dArea > 40000:
        msg.linear.x = 0
        msg.angular.z = 0
    elif cx<295:
        msg.angular.z = 0.1
        msg.linear.x = 0
    elif cx>345:
        msg.angular.z = -0.1
        msg.linear.x = 0
    elif 295<cx<345:
        msg.angular.z = 0
        msg.linear.x = 0.02
    pub.publish(msg)
except:
    print('TransformError')
    continue
```

```
homography = cv2.polylines(grayframe, [dst], True, (0, 0, 0), 2)
cv2.imshow("Homography", homography)
```

```
else:
    cv2.imshow("Homography", grayframe)
```

```
cv2.imshow("img3", img3)
```

```
key = cv2.waitKey(1)
if key == 32:
    fl = 1
elif key == 27:
    break
```

```
cap.release()
cv2.destroyAllWindows()
```

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № дубл.	Подп. и дата

Приложение В

Листинги программ для управления роботом

CMakeLists.txt

```
cmake_minimum_required(VERSION 2.8.3)
project(diploma)

add_compile_options(-std=c++11)

find_package(catkin REQUIRED
  rospy
  geometry_msgs
)

find_package(OpenCV REQUIRED)
include_directories(${OpenCV_INCLUDE_DIRS})

include_directories( ${catkin_INCLUDE_DIRS} )

catkin_package()

add_executable(
  colormask.py
  keypoints.py
)

target_link_libraries(
  colormask.py
  keypoints.py
  ${catkin_LIBRARIES}
  ${OpenCV_LIBRARIES}
)
```

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № дубл.	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата

```
<launch>

  <!-- Launching of control node -->
  <node pkg = "diploma" type = "colormask.py" name = "robot_control_node">
    <!-- node pkg = "diploma" type = "keypoints.py" name = "robot_control_node"
-->

    <!-- remap from = "cmd_vel" to = "/mobile_base/commands/velocity"/ -->
  </node>

</launch>
```

Приложение Г

Демонстрация работы программ



Рисунок Г.1 – Цветовой фильтр

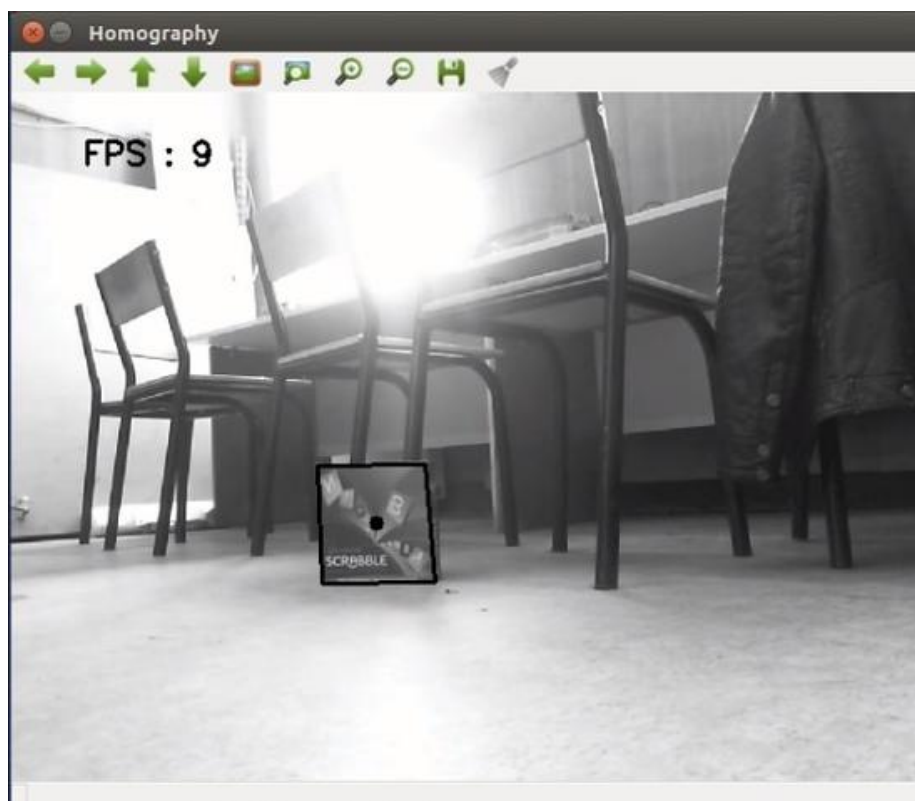


Рисунок Г.2 – Метод ключевых точек

Инд. № подл.	Подп. и дата	Взам. инд. №	Инд. № дубл.	Подп. и дата
Изм.	Лист	№ докум.	Подп.	Дата
<p>ФСУИР.107.R3442.001 ПЗ</p> <p>Копировал</p> <p>Формат А4</p>				
				Лист
				44