#### 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, магистр)*	
Руководитель ВКР Шаветов С.В., к.т.н.	_
(Фамилия, И., О., ученое звание, степень)	(Подпись)
К защите допустить	
уководитель ОП <u>Бобцов А.А., профессор, д.т.</u> (Фамилия, И.О., ученое звание, степень	
٠٠ )٢	20

Студент	<u>Терентьев Р.А.</u>	Группа	R3442	Факультет	<u> СУиР</u>
	(Фамилия, И. О.)				
Направлен	нность (профиль), спе	ециализация			
Консульта					
a)			я, И., О., ученое зва	ние, степень)	(Подпись)
ნ)					
·)		(Фамили:	я, И., О., ученое зва	ние, степень)	(Подпись)
ВКР прин	ята ""		20r.		
	ьность ВКР				
ВКР выпо	лнена с оценкой				
Дата защи	ты ""		_20r.		
Секретарь	, ГЭК				(подпись)
					(подписы)
Листов хр	анения				
Лемонстра	ационных материалов	/Чертежей хран	ения		
7	adiromism marepraires	P		· · · · · · · · · · · · · · · · · · ·	

#### Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

# "САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ"

#### **УТВЕРЖДАЮ**

Рук

#### ЗАДАНИЕ на выпускную квалификационную работу

Студенту Терентьеву Роману Андрееви	<u>ичу</u> Гр	уппа _	R3442	_ Факультет	СУиР
РуководительВКР Шавето	в С.В., 1	к.т.н., до	оцент факу	льтета СУиР	
1 Наименование темы: Исследование на видеоизображении в робототехнически	е алгори	итмов об	, I	ты, должность) Я И ЛОКАЛИЗАЦ	ии объектов
Направление подготовки (специальнос	ть)		15.03.0	06	
Направленность (профиль)Инте	еллектуа	альные т	гехнологи	и в робототехн	нике
Квалификация бакалавр					
2 Срок сдачи студентом законченной р 3 Техническое задание и исходные даны Поставлена задача произвести сравнение задачах слежения за объектом. Необходи	ные к р различі	аботе			
1. Провести обзор существующих алго	оритмог	в обнару	жения и л	окализации об	ъектов;
2. Реализовать выбранные алгоритмы	в виде і	комплек	са програм	MM;	
3. Проверить работу полученных алго	_	в реаль	ных услов	иях на базе мо	бильной
робототехнической платформы Robotino;					
4. Сравнить результаты испытаний по		ным хар	<u>актеристи</u>	кам: быстродо	еиствие,
робастность, зависимость от освещения и	I 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
1. Tipesentanna i owei i ome
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.:
<u>Физматкнига, 2010. – 672 с</u>
2. OpenCV Documentation [Электронный ресурс]: https://docs.opencv.org/3.3.1/
<b>7</b> Дата выдачи задания « <u>01</u> » « <u>ноября</u> » 20 <u>18</u> г.
Руководитель ВКР
Задание принял к исполнению «» «» 20г.

#### 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			Роман Андрееві	<u>ич</u>	
Наиманования	TOMLI RKP.	(ФИО) Исследование ал	IFORHTMOR OFHER	уууния и пока	иниесип
		в робототехнический		ужсния и лока	лизации
OOBCRIOD Ha Brig	цеоизооражении	вроототехнич	сских задачах		
Наименование	е организации,	где выполнена 1	ВКРУ	иниверситет И	ITMO
XAPAK	ТЕРИСТИКА І	выпускной	КВАЛИФИКАІ	ционной р	АБОТЫ
1 Цель исследо	вания <u>Сравне</u>	ение алгоримов о	бнаружения и л	окализации об	у <u>ьектов</u>
с помощью т	ехнического зре	ния			
2 Задачи, решас	емые в ВКР	Обзор существу	ющих алгоритм	ов; реализация	я выбранных
алгоритмов;	проверка их раб	оты в реальных у	условиях; сравне	ние алгоритм	OB_
3 Число источн	иков, использов	анных при соста	влении обзора_	Ç	9
4 Полное число	источников, ис	пользованных в	работе	<u>15</u>	
5 В том числе и	сточников по го	одам			
	Отечественных			Иностранных	
Последние 5	От	Более	Последние	От	Более
лет	5 до 10 лет	10 лет	5 лет	5 до 10 лет	10 лет
6	1	-	3	2	3
6 Использовани	ие информацион	I ных ресурсов Int	ernet	<u>Да, 9</u>	
	1 1	1 11		сло ссылок в списке.	литературы)
7 Использовани именно, и в каком разд	-	пакетов компью	герных програм	м и технологи	й (Указать, какие
	Пакеты компьн	отерных программ	и технологий		Параграф работь
OpenCV					2
ROS					3
8 Краткая хара иной мере спра	ктеристика полу вились с постав	ученных результ ленной задачей с	татов <u>Реализо</u> слежения за объе	ованные алгор ектом. В ходе	итмы в той ил сравнения был
		статки проверен			
рекомендации	по их использова	анию в зависимо	сти от конкретн	ых условий и	гребований

Полученные гранты, при выполнении работы						
	онференциях по теме выпускной работы <u>Нет</u> (Да, нет)					
а) 1(Библиографическо	ое описание публикаций)					
2	эс оппециие пуоликации)					
3						
6) 1						
· ————————————————————————————————————	ие выступлений на конференциях)					
2	· · · · · · · · · · · · · · · · · ·					
3						
Студент Терентьев Р.А.						
(ФИО)	(подпись)					
Руководитель ВКР <u>Шаветов С.В.</u> (ФИО)	(подпись)					
(MIO)	(подпись)					
"						

Копировал

A4

Формат

#### Введение

Подп. и дата

Инв. № дибл.

CHD.

Взам.

дата

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 компьютеры обладают большей ЛЮДЬМИ 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 изображений, сложности, корректировка таких как 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.

Изм	Лист	№ докцм.	Подп.	Дата

ФСУИР. 107.R3442.001 ПЗ

## PCAND:10183442:001 113

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.

u d									
Подп.									
Ĺ									
Инб. № подл.									
No.						I			
JHD.		-	L	117.7		+		4	
_	ИЗМ.	Лист	L	Nº DO	кум.	$\perp$	Подп.	_	Дата

ИНВ. № дубл.

Взам. инв. №

ФСУИР. 107. R 34 4 2.001 П 3

Лист

Копировал

A4

# 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.

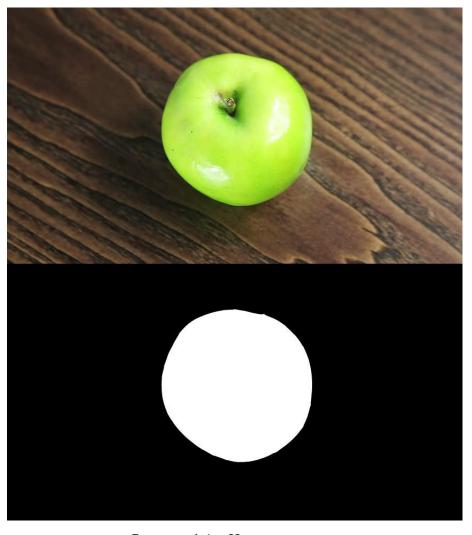


Рисунок 1.1 – Цветовая маска (сверху – исходное изображение, снизу – его бинарная маска)

Изм	Лист	№ докцм.	Подп.	Дата

Взам.

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

Подп. и дата

ИНВ. № дибл.

CHD.

Взам.

дата

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).

_				
Изм.	Лист	№ докцм.	Подп.	Дата

ФСУИР. 107. R 34 4 2.001 П 3

Рисунок 1.2 – Цветовая модель RGB

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

MHB. Nº

CHD.

Взам.

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.

Изм.	Лист	№ докцм.	Подп.	Дата

ФСУИР. 107.R3442.001 ПЗ

HSV удобно представлять в виде цилиндра, где тон определяется полярным углом, насыщенность – радиус-вектором, а яркость – высотой (см. рисунок 1.3).

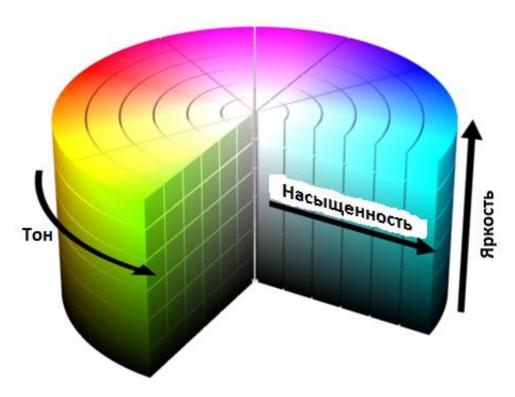


Рисунок 1.3 – Цветовая модель HSV

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

$$RGB \rightarrow HSV$$
:  $\{MAX = \max(R, G, B), MIN = \min(R, G, B)\}$  (1.1)

$$H = egin{cases} rac{30(G-B)}{MAX-MIN}, & ext{при } MAX = R \ 60 + rac{30(B-R)}{MAX-MIN}, & ext{при } MAX = G \ 120 + rac{30(R-G)}{MAX-MIN}, & ext{при } MAX = B \end{cases}$$

если H < 0, то H = H + 180

Изм	Лист	№ докцм.	Подп.	Дата

и дата

WHB. No

CHD.

Взам

дата

ΦCYNP.107.R3442.001 Π3

/lucm 10

$$S = \left\{ egin{array}{ll} 0, & ext{при } MAX = 0 \ \hline 255(MAX - MIN) & ext{при } MAX 
otag \end{array} 
ight.$$

$$V = MAX$$

В результате преобразования получим: Н 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);
- а 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.

Изм. Лист Nº дакум. Подл. Дата

Инв. № дибл.

CHD.

Взам.

ФСУИР. 107.R3442.001 ПЗ

/lucm 11

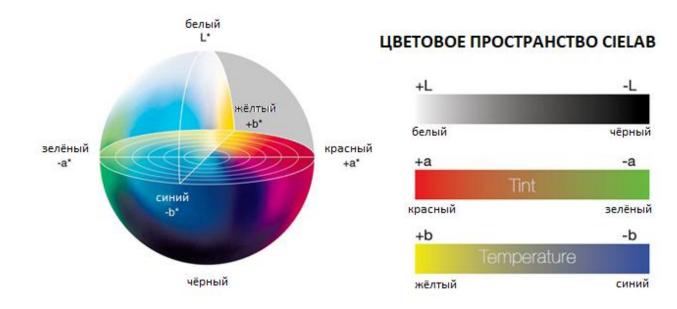


Рисунок 1.4 – Цветовая модель Lab

Для перехода к модели Lab, необходимо сначала осуществить преобразование из RGB в эталонную модель XYZ по следующим формулам:

$$RGB o XYZ$$
:  $\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix}$  (1.2)  $XYZ o Lab$ :  $\left\{ X = \frac{X}{0.950456}, \quad Z = \frac{Z}{1.088754} \right\}$   $L = \left\{ \frac{2.55(116 * Y^{1/3} - 16)}{2.55 * 903.3 * Y}, \quad \text{при } Y > 0.008856 \\ a = 500(f(X) - f(Y)) + 128 \\ b = 200(f(Y) - f(Z)) + 128 \\ rge f(t) = \left\{ \frac{t^{1/3}}{7.787t}, \quad \text{при } t > 0.008856 \\ 7.787t + \frac{16}{116}, \quad \text{при } t \leq 0.008856 \right\}$ 

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

Изм	Лист	№ докцм.	Подп.	Дата

Подп. и дата

συδπ.

WHB. No

CHD.

Взам

и дата

подл.

ФСУИР.107.R3442.001 ПЗ

#### 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 поразному:

- 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 х и по у (угловая точка) [5].

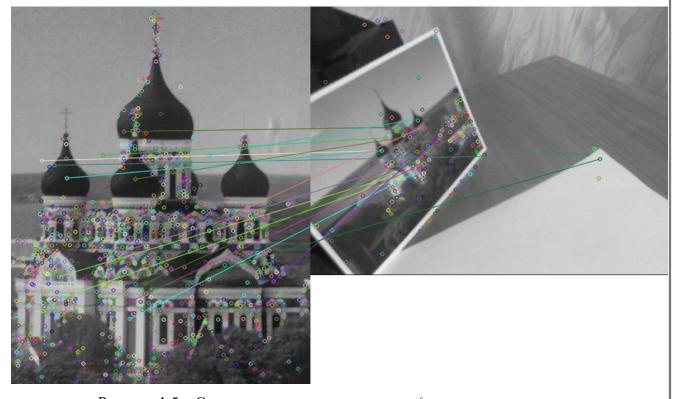


Рисунок 1.5 — Сопоставление ключевых точек (слева — эталонное изображение объекта, справа — изображение, где нужно обнаружить объект)

Изм	Лист	№ докцм.	Подп.	Дата

MHD.

Взам.

ФСУИР. 107. R 34 4 2.001 П 3

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 какуюлибо точку из множества остальных через описание окружающих областей;
- Матчер метод построения соответствий между наборами найденных и описанных точек на разных изображениях.

Рассмотрим характеристики основные некоторых алгоритмов нахождения и описания ключевых точек.

#### 1.2.1 Алгоритм SIFT

Подп. и дата

Инб. № дубл.

M

Взам. инв.

Подп. и дата

«Масштабно-инвариантное преобразование Исходя из названия особенностей» (Scale-Invariant Feature Transform), данный 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 о (о действует как параметр масштабирования). LoG действует как детектор блобов (каплевидных окрестностей), который обнаруживает блобы разных размеров

Изм	Лист	№ докцм.	Подп.	Дата

ФСУИР.107.R3442.001 ПЗ

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

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).

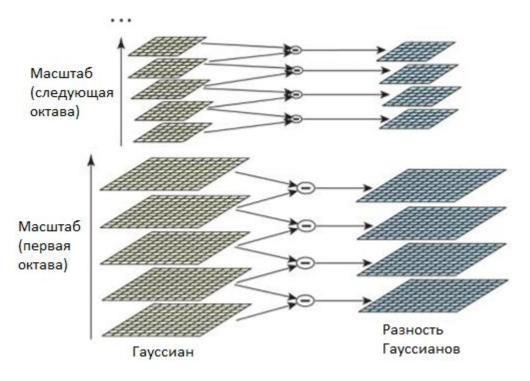


Рисунок 1.6 – Разность Гауссианов

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.

Изм	Лист	№ докцм.	Подп.	Дата

и дата

NHB.

CHD.

Взам.

дата

ФСУИР. 107.R3442.001 ПЗ

/lucm 15

#### 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

и дата

WHB. No

CHD.

Взам.

дата

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 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 16х16 вокруг неё. Она разделяется на 16 подблоков размером 4х4, 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 изменениям освещенности, вращению и т.д. [6]

Изм.	Лист	№ докцм.	Подп.	Дата

ФСУИР. 107.R3442.001 ПЗ

#### 1.2.2 Алгоритм SURF

Алгоритм «Ускоренные робастные особенности» (Speeded-Up Robust Features) является ускоренной версией SIFT с добавлением множества функций. Анализ показывает, что он в 3 раза быстрее, чем SIFT, и сравним с ним по производительности.

SURF вместо DoG использует арифметический усредняющий фильтр (см. рисунок 1.7). Одним из больших преимуществ этого приближения является то, что свертка с усредняющим фильтром может быть легко рассчитана с помощью интегральных изображений, и это можно сделать параллельно для разных масштабов.

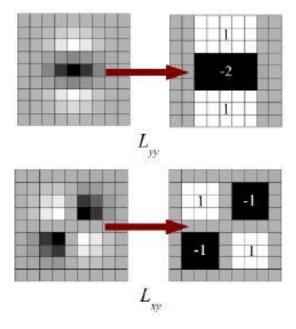


Рисунок 1.7 – Арифметический усредняющий фильтр

Для назначения ориентации SURF использует вейвлет-отклики в горизонтальном и вертикальном направлении для окрестности ключевой точки. Затем они наносятся на график, и доминирующая ориентация оценивается путем вычисления суммы всех откликов в скользящем окне ориентации под углом 60 градусов (см рисунок 1.8).

Изм	Лист	№ докцм.	Подп.	Дата

MHB. Nº

CHD.

Взам.

ФСУИР. 107. R 34 4 2.001 П 3

/lucm 17

Рисунок 1.8 – Назначение ориентации

Для описания ключевых точек SURF использует вейвлет-отклики в горизонтальном и вертикальном направлении. Вокруг ключевой точки берется окрестность размером 20х20. Она разделяется на регионы 4х4, и для каждого региона берутся горизонтальные и вертикальные отклики, формирующие вектор [7].

#### 1.2.3 Алгоритм ORB

Подп. и дата

ИНВ. № дибл.

Взам. инв.

и дата

№ подл.

«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.

Изм. Лист Nº дакум. Подп. Дата

ФСУИР.107.R3442.001 ПЗ

Для описания ключевых точек используется BRIEF. Но BRIEF плохо работает с вращением, поэтому ORB направляет его в соответствии с ориентацией точек. Для любого набора ключевых точек из п бинарных тестов в местоположении ( $x_i$ ,  $y_i$ ) определяется матрица  $S_{2xn}$ , которая содержит координаты этих пикселей. Затем, используя ориентацию пятна  $\theta$ , находится его матрица вращения и S поворачивается, чтобы получить управляемую версию  $S_{\theta}$  (см. рисунок 1.9). ORB дискретизирует угол с шагом 12 градусов и создает таблицу соответствия предварительно рассчитанных шаблонов BRIEF. Пока ориентация ключевой точки  $\theta$  одинакова для всех видов, для вычисления ее дескриптора будет использоваться правильный набор точек  $S_{\theta}$ .

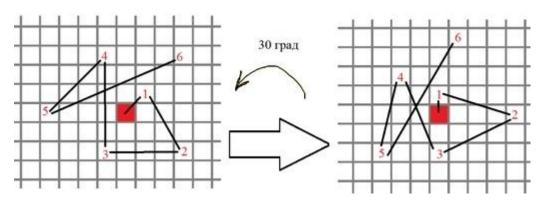


Рисунок 1.9 – Поворот матрицы S

ВRIEF имеет важное свойство: каждый битовый признак имеет большую дисперсию и среднее значение около 0,5. Высокая дисперсия делает функцию более разборчивой, поскольку она по-разному реагирует на входные данные. Но как только он ориентируется в направлении ключевой точки, он теряет это свойство и становится более распределенным. Другим желательным свойством является отсутствие корреляции тестов, поскольку каждый тест будет способствовать получению результата. Чтобы разрешить все эти проблемы, ОRB выполняет жадный поиск среди всех возможных бинарных тестов, чтобы найти те, которые имеют как высокую дисперсию и средние значения, близкие к 0,5, так и являются некоррелированными [8].

Изм	Лист	№ докцм.	Подп.	Дата

и дата

WHB. No

CHD.

ВЗДМ.

дата

ФСУИР. 107. R 34 4 2.001 П 3

#### 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).

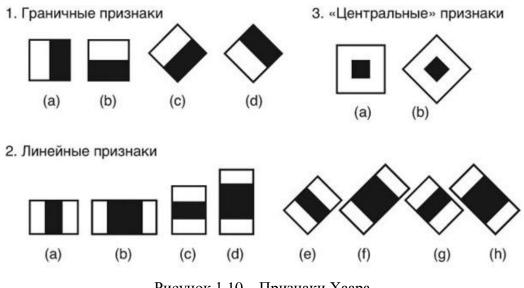


Рисунок 1.10 – Признаки Хаара

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.

Изм	Лист	№ докцм.	Подп.	Дата

WHB. NO

CHD.

Взам.

дата

ФСУИР.107.R3442.001 ПЗ

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.

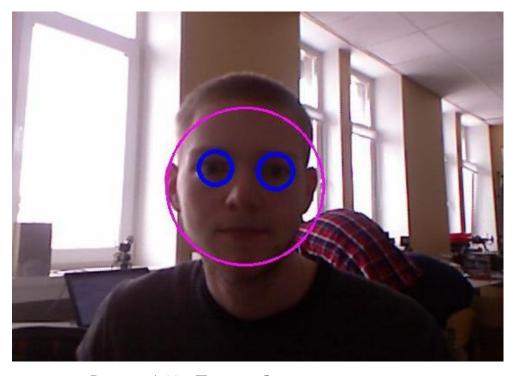


Рисунок 1.11 – Пример обнаружения лица и глаз

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].

Изм	Лист	№ докцм.	Подп.	Дата

MHB. Nº

CHD.

ВЗДМ.

ФСУИР. 107. R 344 2.001 П 3

### 2 Программная реализация выбранных алгоритмов

Для реализации были выбраны алгоритмы на основе цветового фильтра и ключевых точек, так как они не требуют длительного обучения и позволяют достаточно быстро изменять объект слежения. К тому же данные алгоритмы помимо детектирования объекта ещё и определяют его размер. Следует отметить, что речь идёт не о фактическом размере, а о количестве пикселей, относящихся к объекту. По этой величине можно судить о том, насколько далеко находится объект от точки обзора.

Программы Python высокоуровневом написаны на языке библиотек: работы NumPy применением дополнительных ДЛЯ OpenCV функций многомерными массивами для использования компьютерного зрения.

#### 2.1 Цветовой фильтр

Подп. и дата

Инв. № дубл.

CHD.

Взам

дата

На вход программы подаётся изображение или видеопоток, на котором объект слежения должен быть чётко различим по цвету от фона. Далее программа работает по следующему алгоритму:

#### 1) Преобразование цветовой модели

RGB является базовой цветовой моделью, поэтому, чтобы перейти к использованию других моделей, необходимо воспользоваться функцией cvtColor, которая совершает преобразование по формулам (1.1) и (1.2) для HSV и Lab соответственно.

#### 2) Низкочастотная фильтрация

Для снижения уровня шума и удаления высокочастотных компонент на изображении применяют функцию GaussianBlur. Уравнение Гаусса для двух измерений:

Изм	Лист	№ докцм.	Подп.	Дата

ФСУИР. 107.R3442.001 ПЗ

где х - расстояние от начала координат по горизонтальной оси, у расстояние по вертикальной оси, а  $\sigma$  – 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 для построения матрицы свертки, которая применяется изображению. Для исходному каждого нового значения арифметическое определяется среднее взвешенное 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.

_				
Изм.	Лист	№ докум.	Подп.	Дата

MHB. Nº BUDA

Взам. инв.

ФСУИР. 107. R 34 4 2.001 П 3

/lucm 23

Рисунок 2.1 – Процесс создания цветовой маски

#### 4) Морфологические преобразования

Полученная цветовая маска содержит дефекты формы, которые можно убрать с помощью морфологических преобразований. Например, операция открытия (размыкания) позволяет удалить мелкие внешние шумы, а операция закрытия (замыкания) — заполнить внутренние пустоты (см. рисунок 2.2) [10].

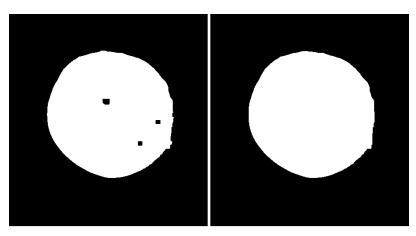


Рисунок 2.2 – Морфологические преобразования (слева – открытие, справа – закрытие)

#### 5) Нахождение контура

MHB. Nº

CHD.

ВЗДМ.

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.

Изм	Лист	№ докцм.	Подп.	Дата

ФСУИР.107.R3442.001 ПЗ

#### 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]:

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

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

$$x_c = M_{10}/M_{00}, \quad y_c = M_{01}/M_{00}$$
(2.3)

Рисунок 2.3 иллюстрирует результат работы цветового фильтра.

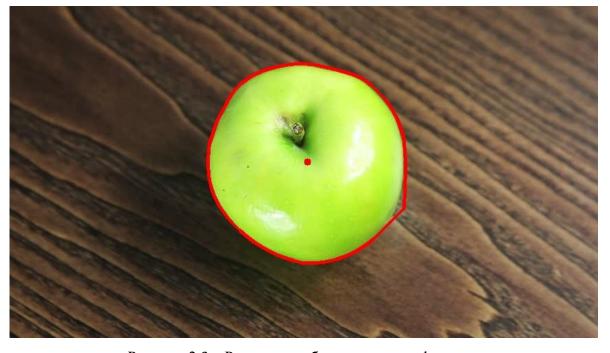


Рисунок 2.3 – Результат работы цветового фильтра

Изм	Лист	№ докцм.	Подп.	Дата

WHO. No

CHD.

Взам.

дата

ФСУИР.107.R3442.001 ПЗ

#### 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:

$$RGB \to Gray$$
:  $Y = 0.299 * R + 0.587 * G + 0.114 * B$  (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

Подп. и дата

NHB. No

CHD.

Взам.

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).

Изм	Лист	№ докцм.	Подп.	Дата

ФСУИР. 107. R 344 2.001 П 3

#### 4) Нахождение «хороших точек»

Найденные совпадения необходимо отфильтровать. Исследования показывают, что существует зависимость вероятности верного/неверного сопоставления от отношения ближайшего совпадения ко второму ближайшему (см. рисунок 2.4). Таким образом, можно определить критерий «хорошей точки»: это точка с отношением ближайшего совпадения ко второму ближайшему менее 0,8. Отклонив «плохие точки», мы исключим 90% ложных совпадений и отбросим менее 5% правильных совпадений [6].



Рисунок 2.4 — Зависимость вероятности верного/неверного сопоставления от отношения ближайшего совпадения ко второму ближайшему

#### 5) Проективное преобразование

Теперь, имея набор «хороших» точек на эталонном изображении и сопоставленный ему набор точек на другом изображении, мы можем найти между ними соответствие через проективное преобразование (гомографию). Данное преобразование осуществляется с помощью матрицы  $H_{3x3}$ , которое отображает точки на одном изображении в соответствующие точки на другом:

Изм.	Лист	№ докцм.	Подп.	Дата

и дата

MHD.

CHD.

Взам.

дата

ФСУИР. 107.R3442.001 ПЗ

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

#### 6) 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.

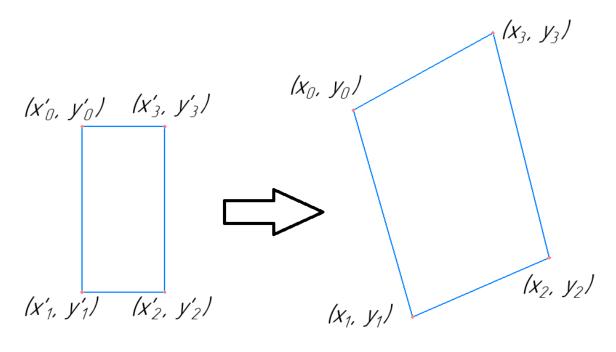


Рисунок 2.5 – Перспективное преобразование

Изм. Лист № дакум. Подп. Дата

CHD.

ВЗДМ.

подл.

ФСУИР.107.R3442.001 ПЗ

#### 7) Вычисление площади

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

$$S = a * b$$
, где 
$$a = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}$$
 
$$b = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

#### 8) Определение координат центра

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

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

Рисунок 2.6 иллюстрирует результат работы метода ключевых точек.



Рисунок 2.6 – Результат работы метода ключевых точек

Изм	Лист	№ докцм.	Подп.	Дата

NHB

CHD.

Взам.

ФСУИР.107.R3442.001 ПЗ

Формат

### 3 Проверка работы алгоритмов в реальных условиях

Оба реализованных алгоритма содержат следующие выходные данные: площадь объекта S и координаты его центра (x<sub>c</sub>, y<sub>c</sub>). Чтобы робот мог следить за объектом, эти данные необходимо интерпретировать, т.е. добавить в алгоритмы команды управления роботом в зависимости от выходных данных.

#### 3.1 Объект управления

В качестве объекта управления в испытаниях используется мобильная робототехническая платформа Robotino от компании Festo Didactic, изображённая на рисунке 3.1. Robotino содержит бортовой компьютер, который может обрабатывать как показания встроенных датчиков, так и команды пользователя с ноутбука. Но главной его особенностью являются три омни-колеса на расстоянии 120° друг от друга, позволяющие роботу передвигаться по плоскости в любом направлении, а также вращаться на месте [13].



Рисунок 3.1 – Объект управления

№ докцм. Подп.

MHB. Nº

CHD.

Взам.

подл.

ФСУИР. 107.R3442.001 ПЗ

Лист 30

Формат

#### 3.2 Законы правления

Для связи ноутбука пользователя с роботом используется фреймворк Robot Operating System (ROS). Связь осуществляется по принципу Подписчик-Публикатор (Publisher-Subscriber). Публикатор размещает данные в теме (topic) /cmd\_vel, которая управляет колёсами робота. Подписчик получает данные из темы и передаёт команды колёсам. Данные представляют собой сообщения типа geometry\_msgs/Twist, состоящие из трёх линейных и трёх угловых скоростей. Но в данное работе будут использоваться только команды «движение вперёд», «стоп» и «поворот влево/вправо» [14].

В зависимости от параметров объекта на видеоизображении (площадь S и координата  $x_c$ ) роботу передаются следующие команды:

- S < 4000 «поворот влево» (объекта нет в поле зрения камеры, нужно вращаться на месте, пока он не будет обнаружен);
- x<sub>c</sub> < 295 «поворот влево» (объект слева);
- $x_c > 345$  «поворот вправо» (объект справа);
- $295 < x_c < 345 «движение вперёд» (объект прямо по курсу);$
- S > 40000 «стоп» (объект достаточно близко).

Код программы, реализующей алгоритм цветового фильтра, указан в Приложении А, метод ключевых точек — в Приложении Б. Приложение В содержит пакет программ для связи с роботом. В Приложение Г помещены изображения, демонстрирующие работу программ.

Изм. Лист № докум. Подп. Дата

Подп. и дата

Инб. № дубл.

CHD.

Взам.

и дата

подл.

ФСУИР. 107. R 34 4 2.001 П 3

/lucm 31

## 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

Апрориты	Цветовой фильтр			Метод ключевых точек			
Алгоритм	RGB	HSV	Lab	SIFT	SURF	ORB	
FPS		25		8	10	16	
Быстродействие,		40		125	100	62,5	
мс		40		123   100   62		02,5	
Зависимость от	высокая средняя н		низкая	шакап		низкая	
освещения	высокал	средилл	пизкал	шэкил			
Зависимость от	низкая		низкая	низкая	высокая		
масштабирования							
Зависимость от		низкая		THORSE			
вращения		пизкал		низкая			
Зависимость от	средняя	низкая	низкая	низкая	низкая	средняя	
перекрытия	Средилл	пизкал	ППЭКСИ	пизкал	пизкал	среднии	
Робастность	низкая	высокая	высокая	высокая	средняя	средняя	

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

- Среди цветовых 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.

Изм.	Лист	№ докцм.	Подп.	Дата

ИНВ. № дұбл.

CHD.

Взам.

ФСУИР. 107. R 344 2.001 П 3

#### 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].

Изм. Лист Nº докум. Подп. Дата

Инв. № дибл.

CHD.

Взам.

подл.

ФСУИР. 107. R 34 4 2.001 П 3

/lucm 33

#### Список использованных источников

- 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)

Изм	Лист	№ докцм.	Подп.	Дата

и дата

WHB. No

CHD.

Взам.

дата

ФСУИР. 107. R 34 4 2.001 П 3

## EU 100'Z††E&ZOL'ANFJФ

- 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 pecypc]: электрон. статья, 2016. **URL**: https://www.learnopencv.com/homography-examples-using-opencv-pythonс/ (дата обращения 22.04.2019)
- 13. Robotino Wiki [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.

Изм	Лист	№ докум.	Подп.	Дата

Инв. № дубл.

CHD.

Взам.

Подп. и дата

ФСУИР. 107.R3442.001 ПЗ

### 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 LBUTTONDBLCLK:
    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)
```

Изм. Лист N<sup>a</sup> докум. Подп. Дато

и дата

WHO. No

CHD.

Взам.

и дата

подл.

ФСУИР.107.R3442.001 ПЗ

timer = cv2.getTickCount()

while(1):

```
flag, img = cap.read()
            imgtr = img
            #imgtr = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
            #imgtr = cv2.cvtColor(img, cv2.COLOR_BGR2LAB)
            imgtr = cv2.GaussianBlur(imgtr,(5,5),0)
            mask = cv2.inRange(imgtr, mmin, mmax)
            # morphology
            kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (9, 9))
            mask = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel)
            mask = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kernel)
            # contours
            _,contours,hierarchy = cv2.findContours(mask,1,1)
            if fl == 1:
                try:
                  max_area = 0
                  pos = 0
и дата
                  for i in contours:
                      area = cv2.contourArea(i)
                      if area > max_area:
                            max_area = area
                            pos = i
                  hull = cv2.convexHull(pos)
WHB. No
                  cv2.drawContours(img,[hull],-1,(0,0,0),2)
                  moments = cv2.moments(pos, 1)
                  dM01 = moments['m01']
CHD.
                  dM10 = moments['m10']
Взам.
                  dArea = moments['m00']
                  #print(dArea)
                  cx = int(dM10 / dArea)
дата
                  cy = int(dM01 / dArea)
                  cv2.circle(img, (cx, cy), 5, (0,0,0), -1)
                  if dArea < 4000:
подл.
                      msg.linear.x = 0
```

Лист

№ докцм.

Подп.

msg.angular.z = 0.1

elif dArea > 40000:

msg.linear.x = 0msg.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),
и дата
          cv2.FONT_HERSHEY_SIMPLEX, 0.75, (0,0,0), 2)
             cv2.imshow('image',img)
             cv2.imshow('imagetrans',imgtr)
             cv2.imshow('mask',mask)
             k = cv2.waitKey(1) & 0xFF
WHO. No
             if k == 32:
                fl = 1
             elif k == ord('r'):
CHD.
                f1 = 0
Взам.
                mmin = np.array([255, 255, 255])
                mmax = np.array([0, 0, 0])
             elif k == 27:
и дата
                break
          cap.release()
          cv2.destroyAllWindows()
№ подл.
                                             ФСУИР.107.R3442.001 ПЗ
```

Лист

A4

№ докцм.

Подп.

## 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])
f1 = 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()
```

Изм. Лист № докум. Подп. Дат

и дата

NHO

CHD.

Взам.

дата

подл.

ФСУИР.107.R3442.001 ПЗ

и дата

WHB. No

CHD.

Взам.

дата

подл.

№ докцм.

Подп.

```
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)
```

Лист 40

ФСУИР.107.R3442.001 ПЗ

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:
Подп. и дата
                   print('TransformError')
                   continue
               homography = cv2.polylines(grayframe, [dst], True, (0, 0, 0), 2)
               cv2.imshow("Homography", homography)
             else:
WHB. No
               cv2.imshow("Homography", grayframe)
             cv2.imshow("img3", img3)
CHD.
Взам.
             key = cv2.waitKey(1)
             if key == 32:
                 fl = 1
и дата
             elif key == 27:
               break
           cap.release()
№ подл.
          cv2.destroyAllWindows()
                                              ФСУИР.107.R3442.001 ПЗ
```

Лист 41

№ докцм.

Подп.

A4

## Приложение В

## Листинги программ для управления роботом

#### CMakeLists.txt

и дата

MHB. Nº L

CHD.

Взам.

дата

подл

```
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}
```

## EU 100'77788'101'dN6)4

#### package.xml

```
<?xml version="1.0"?>
<package format="2">
  <name>diploma</name>
  <version>0.2.0</version>
  <description>The diploma package</description>
                 email="jason03091997@gmail.com">
                                                                  Terentey
                                                        Roman
</maintainer>
  clicense>TODO</license>
  <buildtool_depend>catkin</buildtool_depend>
  <depend>rospy</depend>
  <depend>geometry_msgs</depend>
  <export>
  </export>
</package>
```

#### main.launch

и дата

MHB.

CHD.

Взам.

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>
```

Изм	Лист	№ докцм.	Подп.	Дата

ФСУИР.107.R3442.001 ПЗ

## Приложение Г Демонстрация работы программ



Рисунок Г.1 – Цветовой фильтр



Рисунок  $\Gamma.2$  – Метод ключевых точек

Изм.	Лист	№ докум.	Подп.	Дата

Подп. и дата

Инв. № дцбл.

Взам

Подп. и дата

Инб. № подл.

ФСУИР. 107.R3442.001 ПЗ

Лист 44

Формат