

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт Кибернетики

Направление подготовки 09.04.01 Информатика и вычислительная техника

Кафедра Информационных систем и технологий

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема работы
Отслеживание лица человека на основе применения оптического потока

УДК 004.932.75'1

Студент

Группа	ФИО	Подпись	Дата
8ВМ5А	Гельгинберг Александр Владиславович		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор каф. ИСТ	Спицын Владимир Григорьевич	д.т.н., профессор		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент каф. менеджмента	Конотопский Владимир Юрьевич	к.э.н., доцент		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент каф. ЭБЖ	Извеков Владимир Николаевич	к.т.н., доцент		

ДОПУСТИТЬ К ЗАЩИТЕ:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
Зав. кафедрой ИСТ	Мальчуков Андрей Николаевич	к.т.н., доцент		

Томск – 2017 г.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП 09.04.01 Информатика и вычислительная техника

Код результатов	Результат обучения (выпускник должен быть готов)
Общепрофессиональные компетенции	
P1	Воспринимать и самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально-экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте.
P2	Владеть и применять методы и средства получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе в глобальных компьютерных сетях.
P3	Демонстрировать культуру мышления, способность выстраивать логику рассуждений и высказываний, основанных на интерпретации данных, интегрированных из разных областей науки и техники, выносить суждения на основании неполных данных, анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями.
P4	Анализировать и оценивать уровни своих компетенций в сочетании со способностью и готовностью к саморегулированию дальнейшего образования и профессиональной мобильности. Владеть, по крайней мере, одним из иностранных языков на уровне социального и профессионального общения, применять специальную лексику и профессиональную терминологию языка.
Профессиональные компетенции	
P5	Выполнять инновационные инженерные проекты по разработке аппаратных и программных средств автоматизированных систем различного назначения с использованием современных методов проектирования, систем автоматизированного проектирования, передового опыта разработки конкурентно способных изделий.
P6	Планировать и проводить теоретические и экспериментальные исследования в области проектирования аппаратных и программных средств автоматизированных систем с использованием новейших достижений науки и техники, передового отечественного и зарубежного опыта. Критически оценивать полученные данные и делать выводы.
P7	Осуществлять авторское сопровождение процессов проектирования, внедрения и эксплуатации аппаратных и программных средств автоматизированных систем различного назначения.
Общекультурные компетенции	
P8	Использовать на практике умения и навыки в организации исследовательских, проектных работ и профессиональной эксплуатации современного оборудования и приборов, в управлении коллективом.
P9	Осуществлять коммуникации в профессиональной среде и в обществе в целом, активно владеть иностранным языком, разрабатывать документацию, презентовать и защищать результаты инновационной инженерной деятельности, в том числе на иностранном языке.
P10	Совершенствовать и развивать свой интеллектуальный и общекультурный уровень. Проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности.
P11	Демонстрировать способность к самостоятельному обучению новым методам исследования, к изменению научного и научно-производственного профиля своей профессиональной деятельности, способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности, способность к педагогической деятельности.

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт Кибернетики

Направление подготовки 09.04.01 Информатика и вычислительная техника

Кафедра Информационных систем и технологий

УТВЕРЖДАЮ:

Зав. Кафедрой

(Подпись) _____ (Дата) Мальчуков А.Н.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

В форме:

магистерской диссертации

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8ВМ5А	Гельгинбергу Александру Владиславовичу

Тема работы:

Отслеживание лица человека на основе применения оптического потока

Утверждена приказом директора (дата, номер)

№898/с от 20.02.2017 г.

Срок сдачи студентом выполненной работы:

13.06.2017 г.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе

(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).

Разработка метода отслеживания человека по лицу на видеопоследовательности.

<p>Перечень подлежащих исследованию, проектированию и разработке вопросов</p> <p><i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<p>аналитический обзор существующих методов оценки движения и выбор проектных решений;</p> <p>реализация метода отслеживания человека по лицу с применением оптического потока;</p> <p>проведение тестирования предложенного метода.</p>
<p>Перечень графического материала</p> <p><i>(с точным указанием обязательных чертежей)</i></p>	<p>Презентация работы</p>
<p>Консультанты по разделам выпускной квалификационной работы</p> <p><i>(с указанием разделов)</i></p>	
<p>Раздел</p>	<p>Консультант</p>
<p>Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</p>	<p>Конотопский В.Ю.</p>
<p>Социальная ответственность</p>	<p>Извеков В.Н.</p>
<p>Названия разделов, которые должны быть написаны на русском и иностранном языках:</p>	
<p>2. Алгоритм отслеживания лица человека на основе оптического потока</p>	

<p>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</p>	<p>15.02.2017 г.</p>
--------------------------------------------------------------------------------------------------------	----------------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор каф. ИСТ	Спицын В.Г.	д.т.н., профессор		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ5А	Гельгинберг Александр Владиславович		

Министерство образования и науки Российской Федерации
 федеральное государственное автономное образовательное учреждение
 высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
 ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт Кибернетики

Направление подготовки 09.04.01 Информатика и вычислительная техника

Уровень образования магистратура

Кафедра Информационных систем и технологий

Период выполнения весенний семестр 2016/2017 учебного года

Форма представления работы:

магистерская диссертация

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
 выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы:	13.06.2017 г.
------------------------------------------	---------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
15.02.17	Постановка целей и задач, получение исходных данных, составление и утверждение ТЗ, подбор и изучение материалов по тематике.	<i>10 баллов</i>
15.03.17	Обзор литературы и выбор проектных решений.	<i>15 баллов</i>
01.04.17	Реализация метода отслеживание человека по лицу на основе применения оптического потока.	<i>15 баллов</i>
01.05.17	Тестирование алгоритма и анализ результатов. Оформление пояснительной записки.	<i>20 баллов</i>
15.05.17	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	<i>15 баллов</i>
20.05.17	Социальная ответственность	<i>15 баллов</i>
29.05.17	Раздел ВКР, выполненный на иностранном языке	<i>10 баллов</i>

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор каф. ИСТ	Спицын В.Г.	д.т.н., профессор		

СОГЛАСОВАНО:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
Зав. кафедрой ИСТ	Мальчуков А.Н.	к.т.н., доцент		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
8ВМ5А	Гельгинберг Александр Владиславович

Институт	Кибернетики	Кафедра	Информационных систем и технологий
Уровень образования	магистратура	Направление/специальность	09.04.01 «Информатика и вычислительная техника»

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. <i>Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	Должностные оклады, ставки социального налога, тарифы на электроэнергию и т.д.
2. <i>Нормы и нормативы расходования ресурсов</i>	-
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	Ставка НДС

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. <i>Оценка коммерческого и инновационного потенциала НТИ</i>	Коммерческий потенциал
2. <i>Разработка устава научно-технического проекта</i>	-
3. <i>Планирование процесса управления НТИ: структура и график проведения, бюджет, риски и организация закупок</i>	✓
4. <i>Определение <u>ресурсной</u>, финансовой, экономической эффективности</i>	✓

Перечень графического материала (с точным указанием обязательных чертежей):

1. <i>«Портрет» потребителя результатов НТИ</i>	
2. <i>Сегментирование рынка</i>	
3. <i>Оценка конкурентоспособности технических решений</i>	
4. <i>Диаграмма FAST</i>	
5. <i>Матрица SWOT</i>	
6. <i>График проведения и бюджет НТИ</i>	✓
7. <i>Оценка ресурсной, финансовой и экономической эффективности НТИ</i>	✓
8. <i>Потенциальные риски</i>	

Дата выдачи задания для раздела по линейному графику	
-------------------------------------------------------------	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент каф. менеджмента	В. Ю. Конотопский	к.э.н., доцент		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ5А	А. В. Гельгинберг		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»**

Студенту:

Группа	ФИО
8ВМ5А	Гельгинберг Александр Владиславович

Институт	Кибернетики	Кафедра	Информационных систем и технологий
Уровень образования	магистратура	Направление/специальность	09.04.01 «Информатика и вычислительная техника»

Исходные данные к разделу «Социальная ответственность»:

1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Разработка метода отслеживания человека по лицу на основе применения оптического потока
--------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<p>1. Производственная безопасность</p> <p>1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности:</p> <ul style="list-style-type: none"> – физико-химическая природа вредности, её связь с разрабатываемой темой; – действие фактора на организм человека; – приведение допустимых норм с необходимой размерностью (со ссылкой на соответствующий нормативно-технический документ); – предлагаемые средства защиты; – (сначала коллективной защиты, затем – индивидуальные защитные средства). <p>1.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности:</p> <ul style="list-style-type: none"> – механические опасности (источники, средства защиты); – термические опасности (источники, средства защиты); – электробезопасность (в т.ч. статическое электричество, молниезащита – источники, средства защиты); – пожаровзрывобезопасность (причины, профилактические мероприятия, первичные средства пожаротушения). 	<ul style="list-style-type: none"> – Повышенная или пониженная температура; – Повышенная или пониженная влажность воздуха; – Зрительное напряжение; – Отсутствие или недостаток естественного света; – Повышенный уровень шума. – Электромагнитные излучения; – Электрический ток
<p>2. Экологическая безопасность:</p> <ul style="list-style-type: none"> – защита селитебной зоны – анализ воздействия объекта на атмосферу (выбросы); – анализ воздействия объекта на гидросферу (сбросы); – анализ воздействия объекта на литосферу (отходы); – разработать решения по обеспечению экологической безопасности со ссылками на НТД по охране окружающей среды. 	<ul style="list-style-type: none"> – анализ воздействия объекта ВКР и области его использования на ОС; – разработка решений по обеспечению экологической безопасности

<p>3. Безопасность в чрезвычайных ситуациях:</p> <ul style="list-style-type: none"> – перечень возможных ЧС при разработке и эксплуатации проектируемого решения; – выбор наиболее типичной ЧС; – разработка превентивных мер по предупреждению ЧС; – разработка действий в результате возникшей ЧС и мер по ликвидации её последствий. 	<p>Выбор и описание возможных ЧС; типичная ЧС – пожар.</p> <ul style="list-style-type: none"> – разработка превентивных мер по предупреждению ЧС; – разработка действий в результате возникшей ЧС и мер по ликвидации её последствий.
<p>4. Правовые и организационные вопросы обеспечения безопасности:</p> <ul style="list-style-type: none"> – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны. 	<ul style="list-style-type: none"> – специальные правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны программиста.

Дата выдачи задания для раздела по линейному графику	
-------------------------------------------------------------	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент	Извеков В.Н.	к.т.н., доцент		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ВМ5А	Гельгинберг А.В.		

РЕФЕРАТ

Выпускная квалификационная работа 116 с., 42 рисунка, 20 таблиц, 42 источника, 1 приложение.

Ключевые слова: трекинг, оптический поток, лицевой детектор, компьютерное зрение.

Объектом исследования являются алгоритмы отслеживания объектов на видеопоследовательности.

Цель работы – разработка метода отслеживания человека по лицу на видеопоследовательности.

В процессе исследования проводились анализ существующих методов отслеживания объектов на видеопоследовательности.

В результате исследования был разработан метод отслеживания человека по лицу на основе оптического потока.

Основные конструктивные, технологические и технико-эксплуатационные характеристики: программное обеспечение написано на C++ в среде разработки Microsoft Visual Studio 2013, с использованием библиотеки Open Computer Vision и фреймворка Qt (версия 5.5).

Область применения: интеллектуальное видеонаблюдение

Экономическая эффективность/значимость работы заключается в увеличении объема получаемой информации при качественной и быстрой обработке.

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ И НОРМАТИВНЫЕ ССЫЛКИ

СФ	-	Статический фон
ОСФ	-	Относительно статичный фон
ДФ	-	Динамический фон
НК	-	Неподвижная камера
ДК	-	Динамическая камера
ВД	-	Виола-Джонс
ОП	-	Оптический поток
ЛК	-	Лукас-Канаде
SHL	-	Single Hypothesis Localisation
MHL	-	Multiple Hypothesis Localisation

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	14
1. АНАЛИТИЧЕСКИЙ ОБЗОР МЕТОДОВ ОТСЛЕЖИВАНИЯ ОБЪЕКТОВ НА ВИДЕОПОСЛЕДОВАТЕЛЬНОСТИ.....	17
1.1. Стандартные подходы при отслеживании объектов.....	18
1.2. Типовая структура трекера.....	20
1.3. Область движения	22
1.4. Вычитание фона.....	23
1.5. Полный перебор допустимых вариантов смещения.....	26
1.6. Байесовский подход к отслеживанию	28
1.7. Фильтр частиц.....	30
1.8. Отслеживание на основе детектора	32
1.9. Оптический поток.....	37
2. АЛГОРИТМ ОТСЛЕЖИВАНИЯ ЛИЦА ЧЕЛОВЕКА НА ОСНОВЕ ОПТИЧЕСКОГО ПОТОКА.....	44
2.1. Схема работы трекинга лиц.....	44
2.2. Выбор средств разработки.....	46
2.3. Поиск лица.....	48
2.4. Выбор характерной точки.....	49
2.5. Наложения фильтра на кадр	50
2.6. Нахождение вектора скорости	52
2.7. Интерфейс.....	54
3. ТЕСТИРОВАНИЕ И АНАЛИЗ РЕЗУЛЬТАТОВ.....	58
3.1. Тестирование на базе Baker et al	58
3.2. Тестирование на базе Boston Head Tracking	61

4. ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ.....	67
4.1. Организация и планирование работ	67
4.1.1. Продолжительность этапов работ	68
4.1.2. Расчет накопления готовности проекта	73
4.1.3. Расчет сметы затрат на выполнение проекта	73
4.1.4. Расчет затрат на материалы.....	74
4.1.5. Расчет заработной платы	74
4.1.6. Расчет затрат на социальный налог	75
4.1.7. Расчет затрат на электроэнергию	76
4.1.8. Расчет амортизационных расходов	76
4.1.9. Расчет расходов, учитываемых непосредственно на основе платежных (расчетных) документов (кроме суточных).....	77
4.1.10. Расчет прочих расходов.....	77
4.1.11. Расчет общей себестоимости разработки	77
4.1.12. Расчет прибыли	78
4.1.13. Расчет НДС	78
4.1.14. Цена разработки НИР	78
4.2. Оценка экономической эффективности проекта.....	78
4.2.1. Оценка научно-технического уровня НИР.....	78
5. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ	82
5.1. Анализ вредных и опасных факторов, которые может создать объект исследования.....	83
5.2. Обоснование мероприятий по защите персонала предприятия от действия опасных и вредных факторов (техника безопасности и производственная санитария)	84

5.2.1.	Требования к помещениям для работы с ПЭВМ	84
5.2.2.	Микроклимат	85
5.2.3.	Освещение.....	86
5.2.4.	Шум	89
5.2.5.	Электромагнитное излучение	90
5.2.6.	Электрический ток	90
5.3.	Экологическая безопасность	91
5.3.1.	Анализ влияния объекта на окружающую среду.....	92
5.3.2.	Анализ влияния процесса эксплуатации объекта на окружающую среду и меры по защите окружающей среды	92
5.4.	Безопасность в чрезвычайных ситуациях	92
5.4.1.	Анализ вероятных ЧС, которые может инициировать объект исследования.....	92
5.4.2.	Анализ причин, которые могут вызвать ЧС на производстве при внедрении объекта исследования	93
5.4.3.	Обоснование мероприятий по предотвращению ЧС и разработка порядка действий в случае возникновения ЧС	93
5.5.	Правовые и организационные вопросы обеспечения безопасности	95
5.5.1.	Специальные правовые нормы трудового законодательства.....	95
5.5.2.	Требования к рабочему месту при работе с ПЭВМ.....	95
ЗАКЛЮЧЕНИЕ		97
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....		98
Приложение А ALGORITHM OF A HUMAN FACE TRACKING BASED ON THE OPTICAL FLOW		102

ВВЕДЕНИЕ

Исследования в области компьютерного зрения берут свое начало еще в 60-е годы XX века и в настоящее время, с появлением доступного использования мощных вычислительных машин и программного обеспечения, продолжают активно совершенствоваться и развиваться [1]. Методы компьютерного зрения призваны решать задачи, связанные со сбором зрительной информации и её последующим анализом (детектированием и распознаванием различных классов объектов, предварительной обработкой, выделением деталей, позиционированием их в трехмерном пространстве и т.д.). Результатом обычно является видоизмененное изображение или список значений различных параметров изображения, таких как размер объекта, цвет, координаты, скорость, ориентация по отношению к камере и т.п. Данная информация находит широкий спектр применений во многих областях: системах обеспечения связи и безопасности, робототехнике, анализе автомобильного потока на автомагистралях, системах дополненной реальности, научных исследованиях, системах «умный дом», интерактивных и игровых приложениях. Источниками данных для анализа являются изображения, однако на практике информации от статических изображений недостаточно, поэтому чаще всего анализируется видеопоследовательность (в том числе и в реальном времени), изображения с различных камер или наборы трехмерных данных, отсканированных изображений. В свою очередь это влечет за собой высокие требования к производительности используемой вычислительной системы и вычислительной эффективности алгоритмов обработки [2].

Задача отслеживания объектов, трекинг (англ. *tracking*), представляет значительный интерес в области компьютерного зрения и позволяет определять местоположение движущегося объекта (или нескольких объектов) во времени на видеопоследовательности [3]. Многие системы используют различные частные подходы к задаче отслеживания объекта на основе математического анализа, математической статистики, теории принятия решений, обработки изображения, методов вычислительного интеллекта. Однако в большинстве

своем у них обнаруживается ряд проблем, связанных с геометрическими факторами (класс объекта, поза, артикуляция, количество и степень перекрытий) и фотометрическими факторами (освещение, яркость, контрастность), которые оказывают существенное влияние на прогнозируемость движения [4]. Это связано с тем, что задача трекинга объектов на видеопоследовательности без необходимых ограничений является слабо формализуемой, поэтому при ее решении возникает необходимость накладывать ряд ограничений, которые позволяют добиться требуемых результатов.

Одной из частных задач трекинга является отслеживание человека на видеопоследовательности [3]. Поскольку в большинстве случаев на одном кадре присутствуют несколько движущихся объектов, для однозначной идентификации объекта, как человека, можно использовать комбинации методов, позволяющих обнаружить человека по лицу, а затем построить его траекторию с помощью методов отслеживания, последовательно анализируя каждый кадр.

Целью данной работы является разработка метода отслеживания человека по лицу на видеопоследовательности.

Для достижения поставленной цели в работе решаются следующие **задачи**:

1. аналитический обзор существующих методов оценки движения и выбор проектных решений;
2. реализация метода отслеживания человека по лицу с применением оптического потока;
3. проведение тестирования предложенного метода.

Работа состоит из введения, 5 глав, заключения, списка использованных источников из 42 наименований. Работа изложена на 116 страницах машинописного текста, иллюстрируется 42 рисунками и содержит 20 таблиц.

В первой главе представлен аналитический обзор методов отслеживания движения на видеопоследовательности. Рассмотрены особенности построения трекеров. Подробно расписан метод оценки движения на основе оптического потока и его возможных улучшений.

Во второй главе приведена схема и описание разрабатываемого алгоритма, а также обоснование выбора средств разработки.

В третьей главе проведено тестирование разработанного алгоритма и анализ полученных результатов.

Четвертая глава посвящена экономической части разрабатываемого проекта.

В пятой главе рассматриваются вопросы социальной ответственности исполнителя за воздействия его решений и деятельности на общество и окружающую среду и этическое поведение.

Результаты работы были представлены на XIV Международной научно-практической конференции студентов, аспирантов и молодых ученых «Молодежь и современные информационные технологии» (МСИТ, 2016 г., г.Томск).

1. АНАЛИТИЧЕСКИЙ ОБЗОР МЕТОДОВ ОТСЛЕЖИВАНИЯ ОБЪЕКТОВ НА ВИДЕОПОСЛЕДОВАТЕЛЬНОСТИ

Отслеживание объектов на видеопоследовательности исследуется достаточно давно, поэтому в данной области имеются сложившиеся понятия и термины [3, 5].

Алгоритмы, решающие задачу трекинга объекта, называются трекерами (англ. *tracker*). Движение объекта в области пространства (сцены), обозреваемого камерой (наблюдателем), происходит в течение некоторого интервала времени. В свою очередь объекты на сцене бывают двух видов – статические (не меняющие своего абсолютного положения в заданном временном интервале) и динамические (меняющие свое абсолютное положение с течением заданного временного интервала). Задним фоном будем называть совокупность статических объектов, которые не перекрывают сопровождаемые объекты и могут рассматриваться как единое целое.

Динамический объект, принадлежащий искомому классу, называется объектом интереса. За подобными объектами в определенный момент времени закрепляется цель (англ. *target*), которая описывается набором характеристик в рамках заданного способа её представления – состоянием объекта, а функцией изменения состояния с течением времени является траектория. Соответственно трекинг цели – это отслеживание цели на каждом кадре.

Одной из главных проблем при трекинге являются перекрытия или окклюзии (англ. *occlusions*) объектов. Она заключается в том, что если представить сцену в координатах наблюдателя (x, y, z) , то объекты имеющие меньшие значения z могут полностью или частично закрывать объекты с большим по значению z .

Поиск объекта интереса на кадре осуществляется по признакам (англ. *feature*) - характерным особенностям, по которым можно различить объекты между собой. При этом таких признаков может быть несколько, главное, чтобы их было достаточно для максимального описания объекта интереса.

1.1. Стандартные подходы при отслеживании объектов

Степень изменчивости сцены и подвижность наблюдателя (камеры) во многом обуславливают выбор методов и алгоритмов в задачах, связанных с движением объектов. Выделяется три основных принципиально разных случаев движения [3, 6]:

1. неподвижная камера, статичный фон (НК, СФ);
2. движущаяся камера, относительно статичный фон (ДК, ОСФ);
3. движущаяся камера, динамический фон (ДК, ДФ).

Очевидно, что наличие условий для первого пункта – самый простой вариант, поскольку можно определить движение по пикселям, интенсивность которых изменяется относительно интенсивности фоновых пикселей. Зачастую указанные условия создаются при разработке систем видеонаблюдения, где достаточно обозревать некоторую фиксированную территорию.

Движущаяся камера в отличие от неподвижной сама является источником изменения сцены и фона вследствие собственного смещения. Но при этом появляется возможность извлечь больше информации из полученных данных, например, определить трехмерную форму объекта. Примером прикладной задачи, в которой исследователь сталкивается с подобными условиями, может служить широко известная задача склеивания последовательности изображений в панораму, когда камера снимает набор перекрывающихся кадров и необходимо совместить их в единую сцену.

Наиболее сложные условия формируются при использовании движущейся камеры, когда практически невозможно выделить постоянный фон. Системы автоматического управления автономными роботами являются типичными примерами систем обработки данных, полученных в указанных условиях

Единой классификации методов для отслеживания объектов не существует, но по принципу работы трекеры можно разделить следующим образом [3]:

1. ручной;

2. автоматический;
3. комбинированный (или интерактивный).

В первом случае отслеживание объекта интереса ведется в ручном режиме, во втором случае захват объекта интереса и последующая корректировка происходят полностью автоматически. При комбинированном методе соблюдается компромисс между первыми двумя методами, например, оператор проводит первичное выделение целевого объекта, а отслеживание происходит в автоматическом режиме. На рисунке 1 представлен пример трекинга движущихся объектов.



Рисунок 1. Пример работы трекинга (НК, СФ), использующего в качестве целевого объекта человека

В наиболее простом подходе трекинга рассматривается два кадра, а затем строится траекторию по ним. Сперва отмечаются все перемещения на текущем и предыдущем кадрах, а затем анализируя скорость, направления движения объектов, их размеры, подсчитывается вероятность перемещения объекта по точкам траектории предыдущего и текущего кадров. Наиболее вероятные перемещения присваиваются каждому объекту и складываются в траекторию.

Перемещение объектов в кадре может быть достаточно нетривиально: их траектории могут пересекаться, они могут пропадать и возникать вновь,

несколько объектов могут объединяться или резко менять направление движения. В таких случаях прогнозируемость корректной траектории значительно усложняется и метод построения траектории по двум кадрам для таких сложных перемещений будет давать высокую погрешность. Поэтому чтобы понять, какому объекту какое перемещение соответствует, анализируются цветовые характеристики, скорость и направление движения, положения объектов на кадрах. Результатом является набор наиболее вероятных перемещений объекта, которые образуют траекторию.

Различия двух подходов состоит в том, что на этапе обработки видеопоследовательности учитывается не только положение объекта на текущем кадре, но и вся история его переходов. Для повышения точности трекинга часть производителей использует технологию анализа последовательности кадров и непрерывной постобработки полученных результатов.

1.2. Типовая структура трекера

При отслеживании объектов можно выделить следующие основные подзадачи [3]:

- нахождение признаков объектов (англ. *feature extraction*);
- выбор представления цели (англ. *target representation*);
- сопровождение (англ. *localistion*);
- управление целями (англ. *target management*);
- пост-обработка.

Подобная декомпозиция позволяет сформировать блочную структуру трекера с возможностью модификации отдельных его компонент (рисунке 2).

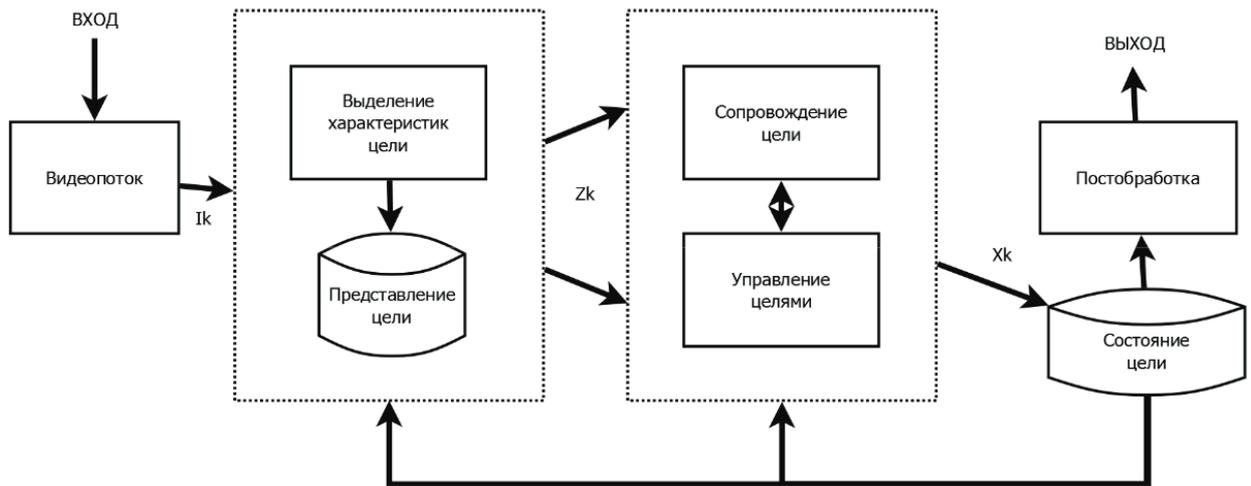


Рисунок 2. Типовая структура трекера [3]

В свою очередь к каждой подзадаче имеется несколько методов решения, выбор которых обусловлен применением трекера для конкретных объектов. Чаще всего категорию трекера определяют именно по используемому алгоритму выявления признаков или по способу представления цели.

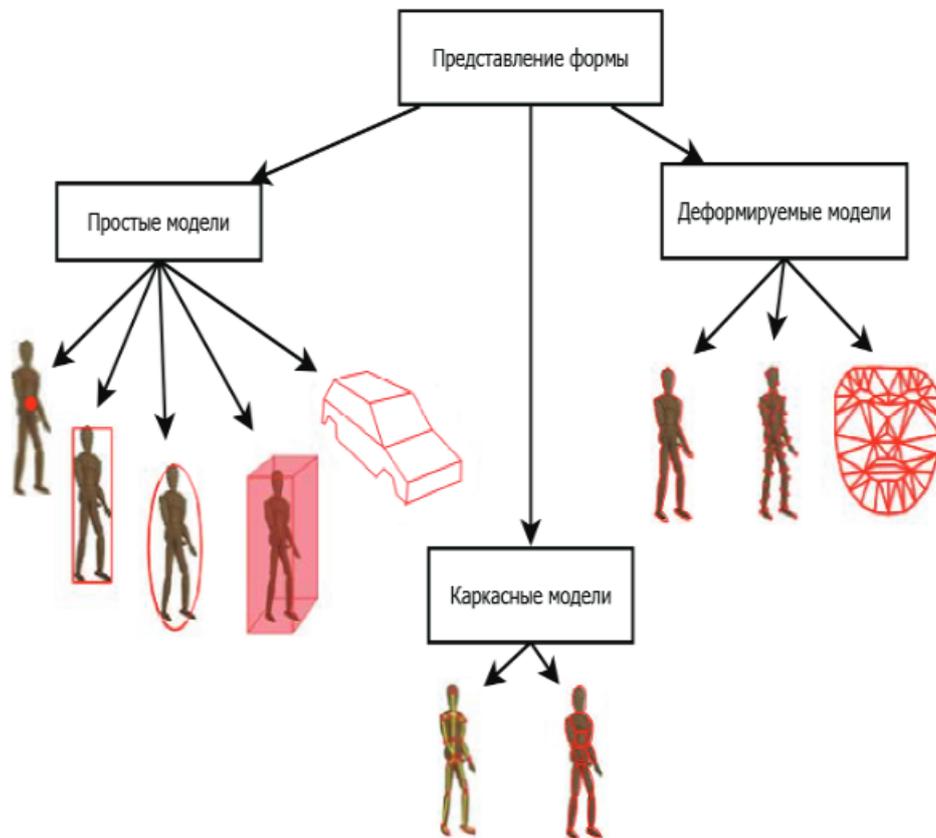


Рисунок 3. Модели представления цели

При выделении признаков могут рассматриваться цвет в различных цветовых пространствах, градиент [3], границы [7], векторное поле или оптический поток [8], особые точки [7], единообразные области [9, 10] и т.д.

Модели представления цели, основные из которых приведены на рисунке 3, обусловлены выбором формы искомого объекта. Цель может быть описана простой формой [11], каркасной моделью, деформируемой моделью [12], точечной моделью [13], гистограммами цвета, ориентации [14] и т.д.

Определение положения цели основано на методах с одной (SHL) или несколькими (MHL) гипотезами. SHL методы основываются на вычислении градиента, а MHL используют регулярную сетку, частотные фильтры или гибридные комбинации [15].

Различные сочетания указанных алгоритмов порождают большое число вариантов отслеживания объектов для разнообразных прикладных задач и ограничений.

1.3. Область движения

Совокупность каждого кадра видео, в которых происходит движение одного или нескольких объектов является областью движения. Для выделения областей движения анализируется последовательность кадров $I_1, I_2, I_3, \dots, I_N$ (рисунок 4).



Рисунок 4. Последовательность кадров

Тогда для любого кадра I_k справедливо равенство (1) [3]:

$$I_k = \{I_k(x, y), 0 \leq x < width, 0 \leq y < height\}, k = 1..N, \quad (1)$$

где $width$ – ширина кадра, $height$ – высота кадра, $I_k(x,y)$ - вектор фиксированной размерности.

Совокупность каждого кадра видео, в которых происходит движение одного или нескольких объектов является областью движения. В результате обработки видео формируется набор бинарных изображений (2):

$$M_k(x,y) = \begin{cases} 1, & (x,y) - \text{пиксели объекта} \\ 0, & (x,y) - \text{пиксели фона} \end{cases}, k = 1..N, \quad (2)$$

где 1 соответствуют пикселям движущихся объектов, а 0 – пиксели фона.

1.4. Вычитание фона

Наиболее простым методом отслеживания перемещения объектов является выделение переднего плана за счет механизма вычитания фона из кадра видео (англ. *background subtraction*) [16-17]. Процедура вычитания предполагает наличие построенной модели фона (3):

$$F = \{F(x,y), 0 \leq x < width, 0 \leq y < height\}, \quad (3)$$

где $width$ – ширина кадра, $height$ – высота кадра.

Помимо запоминания одного типа фона, также существуют способы обновления модели фона по прошествии некоторого времени. Для одноканального изображения (в оттенках серого, когда $I_k(x,y)$, $F(x,y) \in \{0, \dots, 255\}, k=1..N$) процедура вычитания фона разбивается на два этапа.

1. Из значений интенсивности текущего кадра видео попиксельно вычитается значение интенсивности фонового изображения (4):

$$D_k(x,y) = |I_k(x,y) - F_k(x,y)|, k = 1..N, \quad (4)$$

где $I_k(x,y)$ - значение интенсивности k -ого пикселя текущего кадра, $F_k(x,y)$ – значение интенсивности k -ого пикселя фона.

2. Построение бинарного изображения (маски) - из полученной разности (4) выбираем пиксели, принадлежащие фону и объекту. Пиксель принадлежит объекту и окрашивается в белый цвет маски, если разность

интенсивностей фона и текущего кадра для данного пикселя больше некоторого порогового значения, иначе считается, что пиксель принадлежит фону и окрашивается в черный цвет (5).

$$M_k(x, y) = \begin{cases} 255, & D_k(x, y) \geq \tau \\ 0, & D_k(x, y) < \tau \end{cases}, k = 1..N, \quad (5)$$

где $D_k(x, y)$ – значение разности интенсивностей фона и текущего кадра k -ого пикселя, τ - пороговое значение интенсивности.

Качество такого подхода во многом зависит от заранее построенной модели фона, его изменения с течением времени, освещения, теней, зашумленности камеры (рисунок 5).



Рисунок 5. Пример трекинга на основе вычитания фона

В зависимости от механизма построения, различные техники вычитания фона подразделяются на две группы нерекурсивные и рекурсивные [3].

Нерекурсивные методы позволяют обновлять модель фона текущего кадра на основе информации об интенсивностях пикселей набора предшествующих моделей фона и текущего кадра.

При усреднении предшествующих кадров для построения модели фона некоторое количество кадров обозначается, как s . Тогда для кадра $I_k(x,y)$ модель фона $F_k(x,y)$ вычисляется (6).

$$F_k(x, y) = \frac{1}{s} \sum_{j=0}^{s-1} I_{k-j}(x, y) \quad (6)$$

Так же для некоторого фиксированного количества предшествующих кадров можно вычислить медиану, согласно которой будет происходить обновление фона (7):

$$F_k(x, y) = \text{med}_{j=0..s-1}\{I_{k-j}(x, y)\} \quad (7)$$

Преимуществом нерекурсивных методов является простота реализации и достаточно высокая скорость обновления моделей фона при воспроизведении видео. Однако, качество работы этих методов сильно зависит от скорости движения объектов. Как правило, медленно передвигающиеся объекты обнаруживаются плохо, это связано с главной особенностью методов - нерекурсивным обновлением фона по предыдущим кадрам. Более того, не достигается качественный результат при изменении света либо наличии динамического фона (листья деревьев, струящейся воды и т.п.). Для устранения влияния перечисленных эффектов, для кадра I_k представим обновленную модель фона в виде выпуклой оболочки модели фона F_{k-1} с текущим изображением. Такая процедура называется α -смешиванием (α -blend) и выражается (8) [18]:

$$F_k(x, y) = \alpha I_{k-1}(x, y) + (1 + \alpha)F_{k-1}(x, y), \quad (8)$$

где α – весовой коэффициент слияния.

Для обновления фона с помощью рекурсивных методов используется информация об интенсивности пикселей только текущего кадра. Одним из таких методов является гистограммный метод [14].

Все цветовое пространство разбивается на отдельные бины (пространство полутонового изображения будет представлять собой отрезок изменения интенсивности, а пространство цветного изображения – куб с тремя измерениями). Для каждого кадра видео выполняется построение гистограммы. Проходим по каждому пикселю изображения, и в зависимости от интенсивности/цвета пикселя увеличиваем на единицу соответствующий бин гистограммы. Считается, что пиксели принадлежат фону, если величина его бина меньше фиксированного порогового значения яркости/цвета, иначе пиксель принадлежит объекту (рисунок 6).

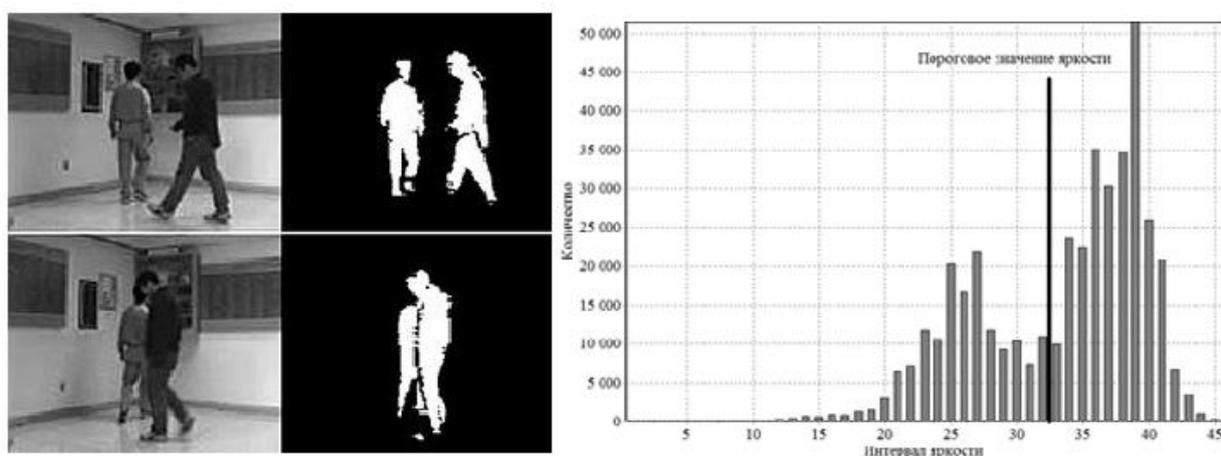


Рисунок 6. Вычитание фона гистограммным методом

Основной недостаток гистограммного метода состоит в том, что необходимо использовать дополнительную память и последующее количество операций обращение к ней возрастает.

1.5. Полный перебор допустимых вариантов смещения

С помощью перебора всех возможных вариантов смещения изображения или его отдельных фрагментов (англ. *translational alignment*) так же возможно оценить движение на видеопоследовательности [19]. Для этого сперва выбирается метрика для оценки степени сходства фрагментов, и затем, задача определения движения сводится к минимизации функции ошибки по всем возможным направлениям смещения (рисунок 7).

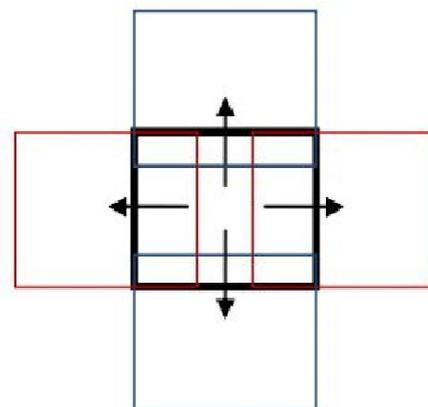


Рисунок 7. Пример перебора вариантов для четырех принципиальных направлений

Наиболее распространенными используемыми функциями ошибки являются:

1. Квадратичная функция ошибки.

Общий вид данной функции определяется уравнением (9):

$$E(u, v) = \sum_i (I_k(x_i + u, y_i + v) - I_{k-1}(x_i, y_i))^2, \quad (9)$$

где (u, v) – вектор смещения изображения или фрагмента, I_k и I_{k-1} значения интенсивности k -ого пикселя текущего и предыдущего кадров.

2. Грубые оценочные функции ошибки.

Общий вид данной функции определяется уравнением (10):

$$E(u, v) = \sum_i \rho(I_k(x_i + u, y_i + v) - I_{k-1}(x_i, y_i)), \quad (10)$$

где $\rho(e)$ – функция, имеющая меньшую степень роста в сравнении с квадратичной функцией.

3. Взвешенная квадратичная функция ошибки.

Общий вид данной функции определяется уравнением (11):

$$E(u, v) = \sum_i w_{k-1}(x_i, y_i) w_k(x_i + u, y_i + v) (I_k(x_i + u, y_i + v) - I_{k-1}(x_i, y_i))^2, \quad (11)$$

где w – весовые коэффициенты (дискретные функции, принимающие нулевые значения за пределами изображения).

4. Кросс-корреляция.

Общий вид формулы определяется уравнением (12):

$$E(u, v) = \sum_i I_k(x_i + u, y_i + v) I_{k-1}(x_i, y_i), \quad (12)$$

Кросс-корреляция пары кадров подразумевает выполнение не минимизации, а максимизации по всем возможным направлениям смещения.

Если метод работает с цветной видеопоследовательностью, то для вычисления функций ошибок всегда можно выполнить конвертирование кадров в полутона, либо ввести дополнительную сумму по числу каналов. На практике метод полного перебора работает достаточно медленно, поэтому имеет смысл использовать его только в совокупности с другими методами.

1.6. Байесовский подход к отслеживанию

Байесовский подход относится к вероятностным методам визуального отслеживания и заключается в построении функции плотности распределения вероятности вектора состояния отслеживаемого объекта [20]. Функция описывает текущую степень знания о состоянии объекта и строится на основе всей полученной в процессе работы метода информации.

Для данного способа характерно следующее математическое представление пространства состояний объекта (13, 14):

$$x_k = f_k(x_{k-1}, v_{k-1}), \quad (13)$$

где x_k , x_{k-1} – состояние объекта, v_{k-1} – стохастическая ошибка, представляющая собой погрешность, получаемую при обновлении состояния.

Формула (13) называется уравнением динамики и описывает изменение состояния объекта при переходе от предыдущего кадра $k-1$ к текущему k . Состояние объекта на текущем кадре зависит от состояния объекта на предыдущем кадре и значения стохастической ошибки. Поскольку v_{k-1} является

случайной величиной с известной статистикой, уравнение динамики неявно задает функцию плотности распределения вероятности $p(x_k / x_{k-1})$.

$$z_k = h_k(x_k, n_k), \quad (14)$$

где z_k – величина наблюдения, n_k – значение случайной ошибки, характеризующей погрешность, получаемую в процессе измерения состояния объекта (рисунок 8).

Уравнением (14) описывает процесс измерения, результатом которого будет некоторая величина z_k , называемая наблюдением, зависящим от текущего состояния объекта x_k и случайной ошибки n_k , являющейся стохастической переменной и аналогично уравнению 13 задает функцию правдоподобия $p(z_k / x_k)$.

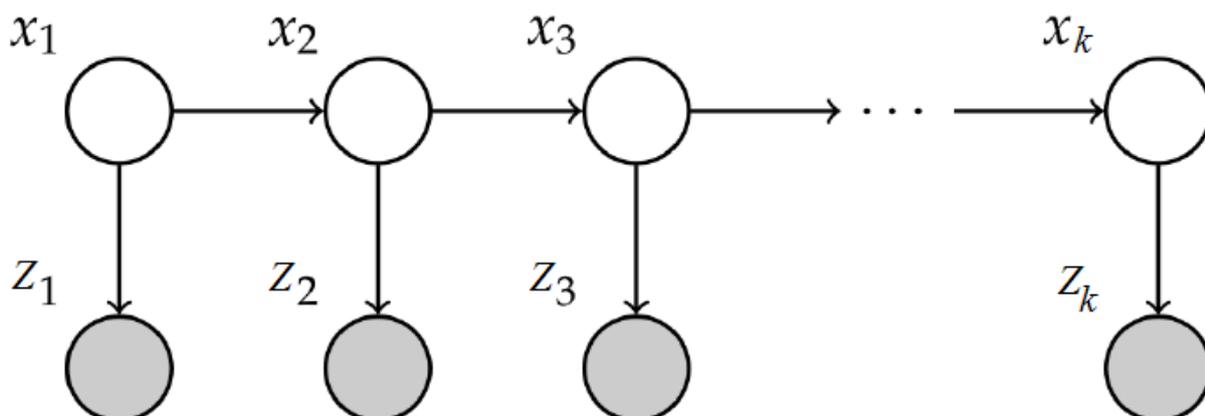


Рисунок 8. Модель состояния объекта

При байесовском подходе проблема отслеживания заключается в рекурсивном вычислении некоторой степени доверия к состоянию объекта x_k на кадре k с учетом всех имеющихся на текущий момент наблюдений. Таким образом вычисление функции плотности распределения вероятности $p(x_k / z_k)$ будет состоять из двух этапов: предсказания и обновления (коррекции).

На этапе предсказания при известном значении искомой функции на предыдущем кадре $p(x_{k-1} / z_{1:k-1})$, с помощью уравнения Чемпена-Колмогорова (15) можно вычислить априорную функцию плотности распределения вероятности состояния объекта на кадре k .

$$p(x_k | z_{1:k-1}) = \int p(x_k | x_{k-1})p(x_{k-1} | z_{1:k-1})dx_{k-1}, \quad (15)$$

где $p(x_k / x_{k-1})$ содержит априорную плотность распределения вероятности смены состояния.

Теперь на кадре k становится известным наблюдение z_k , которое может быть использовано для обновления априорной функции по формуле Байеса (16).

$$p(x_k | x_{1:k}) = \frac{p(z_k | x_k)p(x_k | z_{1:k-1})}{p(z_k | z_{1:k-1})}, \quad (16)$$

где $p(z_k | z_{1:k-1})$ - нормирующий множитель, зависящий от функции правдоподобия.

Рекурсивные соотношения (15) и (16) формируют базис для нахождения оптимального решения, однако, в общем случае это решение не может быть получено аналитическим путем, но его можно найти с помощью аппроксимации.

1.7. Фильтр частиц

Функция плотности распределения вектора состояний может быть аппроксимирована набором частиц, каждая из которых характеризуется собственным весом (под частицей подразумевается некоторое возможное состояние объекта) [7, 21]. Множество S_k (17) задает приближенную функцию плотности распределения вероятности $p(x_k / z_{1:k})$ для состояния x_k при заданном наборе наблюдений $z_{1:k}$.

$$S_k = \left\{ (s_k^i, w_k^i), i = \overline{1, N}, \sum_{i=1}^N w_k^i = 1 \right\}, \quad (17)$$

где w_k^i – вес частицы, задаваемый с помощью функции правдоподобия. Вес частицы показывает вероятность, с которой объект примет состояние, описываемое данной частицей.

Базовым алгоритмом фильтра частиц является алгоритм воспроизведения условной плотности (англ. *conditional density propagation*), на его основе строятся различные модификации под необходимое отслеживание. Основной идеей метода является построение множества частиц S_k на основе множества с предыдущего шага S_{k-1} . Алгоритм включает в себя 3 этапа.

1. Предвыборка.

Выбирается некоторая частица из множества S_{k-1} для помещения в множество S_k . Вероятность попадания частицы в новое множество равна её весу w_{k-1}^i . В результате предвыборки множество S_k будет заполнено частицами из предыдущего множества с наибольшим весом. Выборка проводится на основе распределения вероятности смены состояний $p(x_k / x_{k-1})$, который является одним из настраиваемых параметров фильтра частиц, оказывающим существенное влияние на получаемый результат.

2. Предсказание.

Каждая частица сформированного множества S_k обновляется согласно уравнению динамики (18).

$$s_k^i = f_k(s_{k-1}^i, w_{k-1}^i), i = 1..N. \quad (18)$$

Этот этап алгоритма соответствует вычислению априорной функции плотности распределения вероятности (15).

3. Коррекция.

Происходит обновление весов частиц. Новые веса задаются с помощью выбранной функции правдоподобия, после обновления происходит нормализация весов, таким образом, чтобы их общая сумма была равна единице (19).

$$\hat{w}_k^i = \frac{w_k^i}{\sum_{j=1}^N w_k^j}, i = 1..N. \quad (19)$$

Результатом всех вышеописанных операций на каждом шаге итерации k получаем искомое множество частиц $S_k = \{s_k^i, \hat{w}_k^i\}_{i=1}^N$. Чтобы оценить

состояние объекта на текущем кадре берется «средняя» частица полученного множества (20).

$$x_k = E[S_k] = \sum_{i=1}^N s_k^i w_k^i \quad (20)$$

Недостатком использования фильтров частиц является необходимость первоначально уметь эффективно генерировать выборку из распределения $p(x_k / x_{k-1})$, поэтому рекомендуется брать простые распределения, допускающие эффективную генерацию выборки.

1.8. Отслеживание на основе детектора

Детекторы, определяющие положение цели на конкретном кадре, так же могут быть использованы для анализа движения объекта. За счет использование заданной модели для определения цели, трекерами могут быть:

1. статические детекторы (модель цели не изменяется с течением времени [22]);
2. адаптивные детекторы (модель цели адаптируется под изменения изображения цели на видеопоследовательности [19]);
3. комбинированные детекторы [23];
4. детекторы использующие неизменные части шаблонов [24];
5. порождающие детекторы (генерирующие множество вариантов моделей цели [25]).

Задача детектирования является совершенно самостоятельной задачей и имеет ряд успешных решений, однако наиболее популярным и хорошо зарекомендованным подходом является метод Виолы-Джонса [26]. Разработанный в 2001 году метод до сих пор является основополагающим для поиска различных объектов (лиц, глаз, автомобильных номеров и т.д.). Метод Виолы-Джонса основан на следующих основных принципах:

1. использование принципа сканирующего окна;
2. поиск нужных объектов происходит с помощью признаков Хаара;

3. использование бустинга для выбора наиболее подходящего признака для искомого объекта на данной части изображения;
4. использование бинарного классификатора, дающего результат «верно» или «ложь» при поступлении на его вход признака.

Кадр, на котором имеются искомые объекты, представлен в виде двумерной матрицы пикселей размерности $weight*height$. Значение пикселей принимает значение от 0 до 255 в случае полутонового изображения, а от 0 до 255^3 в случае цветного. Алгоритм должен определить искомые объекты и их черты и отметить их, поиск осуществляется в активной области изображения прямоугольными признаками, которые описывают найденный объект (21):

$$rectangle_i = \{x, y, weight, height, a\}, \quad (21)$$

где x, y – координаты центра i -ого прямоугольника, $weight, height$ – ширина и высота, a – угол наклона прямоугольника к вертикальной оси.

В методе Виолы-Джонса используется интегральное представление изображений. Изображение представляет собой матрицу, которая совпадает по размерам с исходным изображением. В каждом её элементе хранится сумма интенсивностей всех пикселей, которые находятся слева и выше данного элемента. Расчет элементов матрицы задается уравнением (22):

$$L(x, y) = \sum_{i=0, j=0}^{i \leq x, j \leq y} I(i, j), \quad (22)$$

где $I(i, j)$ – интенсивность пикселя исходного изображения.

Каждый элемент матрицы $L(x, y)$ – это сумма пикселей в прямоугольнике от $(0,0)$ до (x,y) . Значение каждого пикселя (x,y) равно сумме значений всех пикселей, находящихся слева и выше данного пикселя (x,y) . По времени расчет матрицы пропорционален числу пикселей в изображении и рассчитывается по формуле (23):

$$L(x, y) = I(x, y) - L(x - 1, y - 1) + L(x, y - 1) + L(x - 1, y) \quad (23)$$

Пример расчета проиллюстрирован на рисунке 9. Пусть в прямоугольнике ABCD есть интересующий объект D:

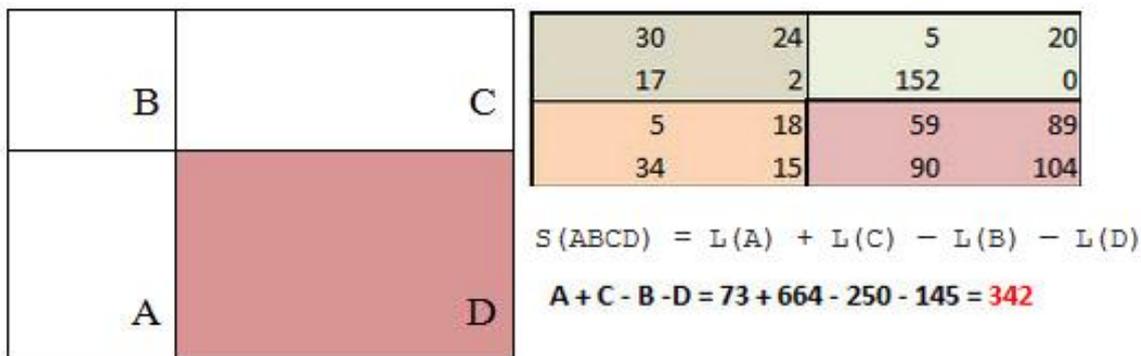


Рисунок 9. Пример расчета интегрального значения изображения

Значение функции Хаара вычисляется, как разность сумм интенсивностей пикселей в черных и белых областях примитива. В методе Виолы-Джонса были использованы четыре прямоугольных примитива, а в последующем их количество был значительно расширено (рисунок 10).

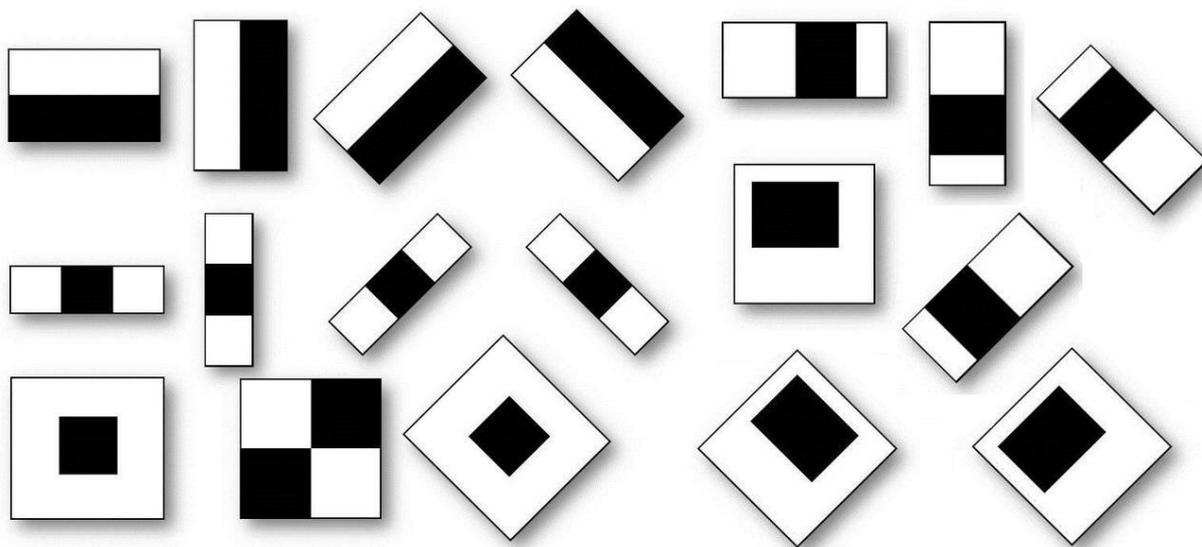


Рисунок 10. Примитивы Хаара

Алгоритм *AdaBoost* (*adaptive boosting*) применяется для объединения произвольного числа классификаторов в композицию и производить обучение на одном наборе примеров, поочередно применяя их на различных шагах. Алгоритм был предложен Йовом Фройндом и Робертом Шапиром в 1999 году [27]. В результате работы алгоритма бустинга формируется простой классификатор вида (24):

$$h_j(z) = \begin{cases} 1, & \text{если } p_j f_j(z) < p_j \theta_j \\ 0 & \text{иначе} \end{cases}, \quad (24)$$

где p_j – направление знака неравенства, $f_j(z)$ – вычисленное значение признака, θ_j – значение порога.

Алгоритм производит вызов слабых классификаторов в цикле, обновляя после каждого вызова распределения весов, указывающих на важность каждого объекта обучающего множества. На каждой последующей итерации значение веса неверно классифицированного объекта возрастает, что вызывает повышенное внимание нового комитета классификаторов на этом объекте (рисунок 11).

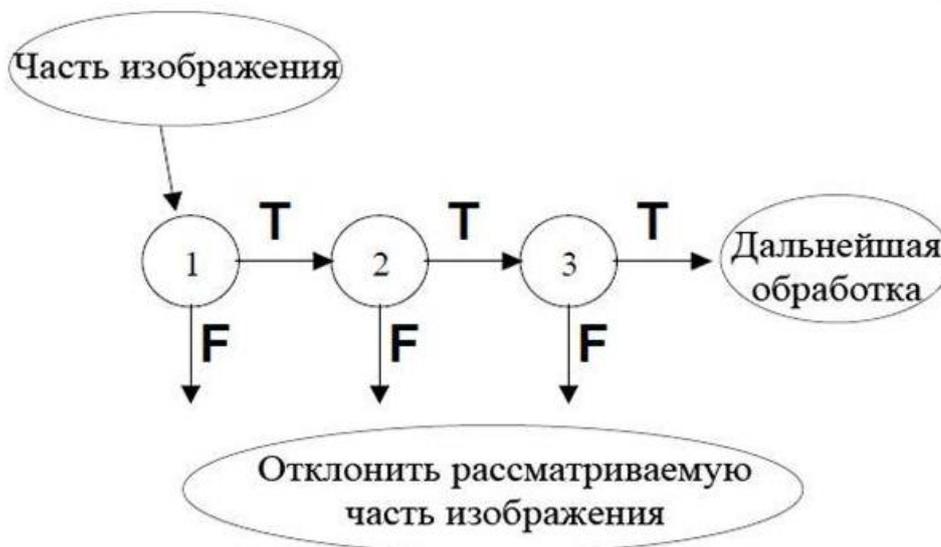


Рисунок 11. Структура каскада

Трекинг на основе детектирования заключается в запоминании пространственной информации об объекте. Объект, обнаруженный детектором в кадре, описывается уравнением (25):

$$object_{kn} = \{image_{kn}, rectangle_{kn}\}, \quad (25)$$

где $image_{k,n}$ – изображение объекта, $rectangle_{k,n}$ – положение прямоугольника, k – номер кадра, n – номер объекта. По формуле 21 прямоугольную область можно представить в виде набора из двух точек $p_1(x_1, y_1)$, $p_2(x_2, y_2)$, описывающих положение верхнего левого угла и нижнего правого угла $rectangle_{k,n}$.

В данном подходе допускается предположение, что объекты из кадра в кадр меняют свою позицию незначительно. Для того, чтобы определить, насколько объект с $rectangle_{k,n}$ на текущем кадре совпадает с $rectangle_{k-1,n}$, нужно учитывать насколько близок центр этих двух прямоугольников на обоих кадрах. Поэтому вводят некоторую меру качества, определяющую долю пересечения двух областей и последующего их объединения в трек. Например, на рисунке 12 показано, что процент совпадения прямоугольных свыше 33% соответствует треку одного объекта.

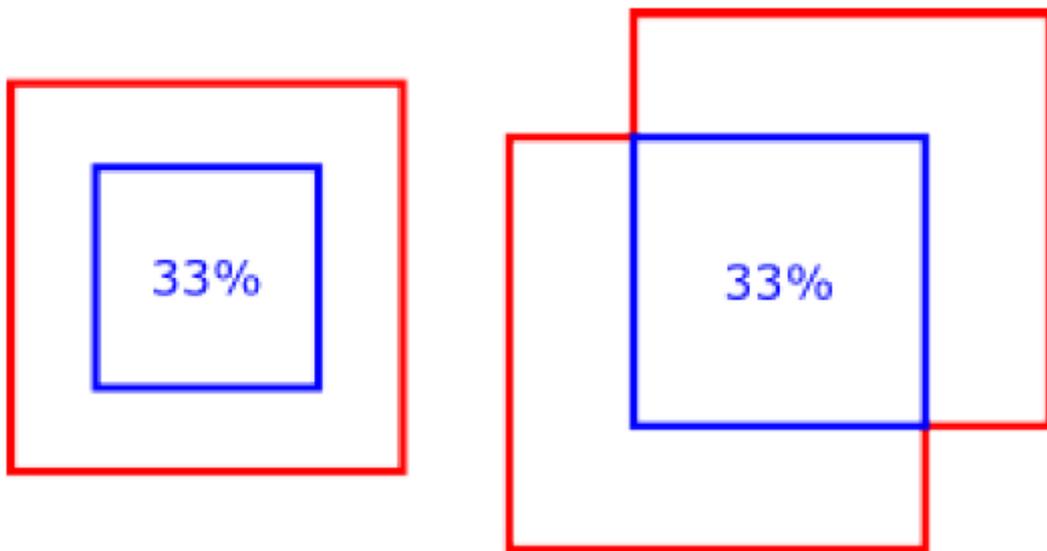


Рисунок 12. Мера пересечения областей

На рисунке 13 показан пример работы трекинга на основе детектора Виолы-Джонса, который в качестве целевых объектов составляет треки из лиц.



Рисунок 13. Пример работы трекинга на основе детектора

Недостатком такого подхода является то, что качество работы трекинга напрямую зависит от качества работы детектора, поэтому при потере объекта прерывается и трек.

1.9. Оптический поток

Оптический поток (англ. *optical flow*) – это векторное поле скоростей пикселей, которое переводит одно изображение в другое и представляет собой сдвиг каждой точки между этими изображениями, определяет видимое движение объектов, поверхностей или краев сцены, получаемое в результате перемещения наблюдателя относительно сцены (рисунок 14).



Рисунок 14. Видимый оптический поток

В основе понятия оптического потока лежат предположения о том, что при изменении положения конкретного пикселя от кадра к кадру его яркость и интенсивность не меняются, а ближайшие точки, принадлежащие одному

объекту, в плоскости изображения двигаются с одинаковой скоростью (рисунок 15).



Рисунок 15. Соответствие пикселей на двух изображениях

Обозначим яркость пикселя $I(x,y,t)$ с координатами (x,y) в момент времени t , тогда уравнение потока в момент времени $t+\Delta t$ записывается следующим образом (26):

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t), \quad (26)$$

где Δx , Δy смещение конкретного пикселя на новом кадре, Δt – изменение времени.

Первую часть уравнения можно выразить с помощью разложения в ряд Тейлора (27):

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \Delta x \frac{\partial I}{\partial x} + \Delta y \frac{\partial I}{\partial y} + \Delta t \frac{\partial I}{\partial t}. \quad (27)$$

Поскольку за небольшой промежуток времени пиксель смещается незначительно, записывается следующее равенство (28):

$$\Delta x \frac{\partial I}{\partial x} + \Delta y \frac{\partial I}{\partial y} + \Delta t \frac{\partial I}{\partial t} = 0. \quad (28)$$

После преобразования, уравнение 28 приобретает вид (29):

$$\frac{\partial I \Delta x}{\partial x \Delta t} + \frac{\partial I \Delta y}{\partial y \Delta t} + \frac{\partial I}{\partial t} = 0. \quad (29)$$

С каждым пикселем изображения можно связать некоторый вектор скорости, который определяет, какое расстояние "прошел" пиксель в течение

временного промежутка между предыдущим и текущим кадрами. С помощью этого уравнения можно найти и построить вектор скорости пикселя (30):

$$I_x v_x + I_y v_y + I_t = 0, \quad (30)$$

где v_x, v_y — определяют вектор скорости в направлениях x, y (рисунок 16).

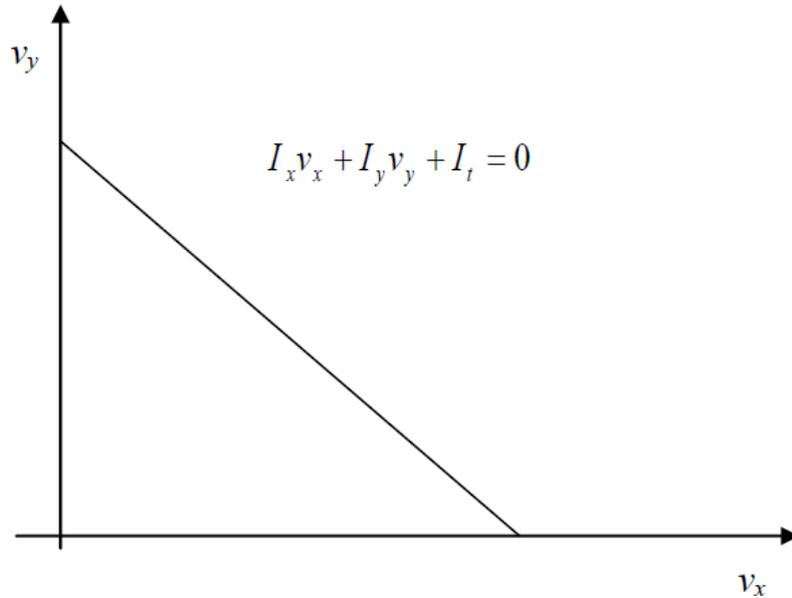


Рисунок 16. Вектор скорости оптического потока для одного пикселя

Уравнение оптического потока (30) содержит две неизвестные и не может быть однозначно решено. Алгоритм Лукаса — Канаде обходит эту неоднозначность за счет использования информации о соседних пикселях в каждой точке. Метод основан на предположении, что в локальной окрестности каждого пикселя значение оптического потока одинаково (рисунок 17), таким образом можно записать основное уравнение оптического потока для всех пикселей окрестности (31).

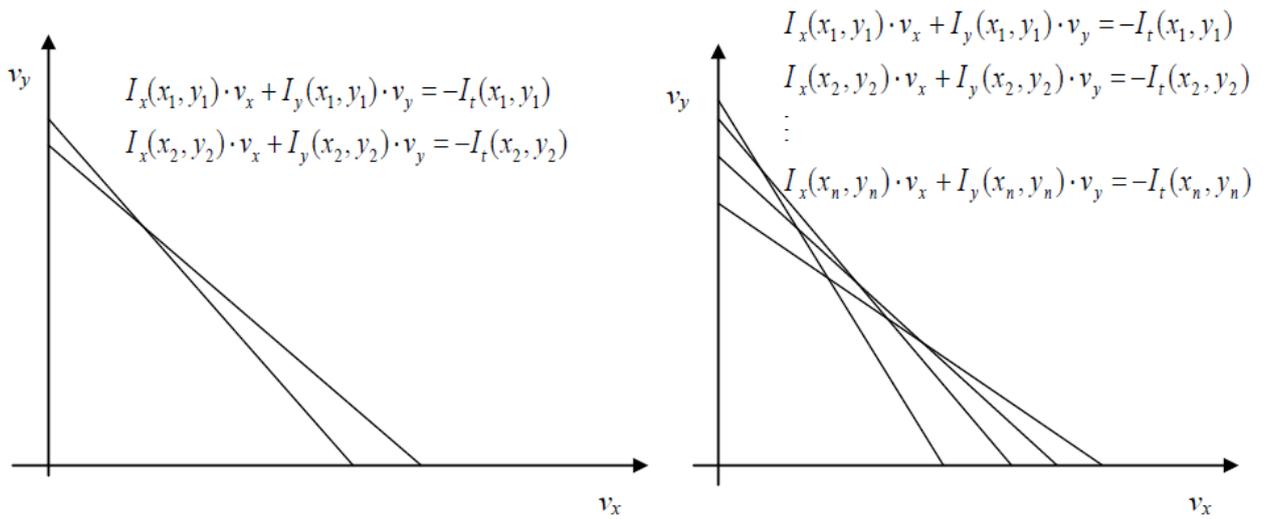


Рисунок 17. Вектора скорости оптического потока для n пикселей

$$\begin{aligned}
 I_x(x_1, y_1)v_x + I_y(x_1, y_1)v_y &= -I_t(x_1, y_1) \\
 I_x(x_2, y_2)v_x + I_y(x_2, y_2)v_y &= -I_t(x_2, y_2) \\
 &\dots \\
 I_x(x_n, y_n)v_x + I_y(x_n, y_n)v_y &= -I_t(x_n, y_n)
 \end{aligned} \tag{31}$$

Полученная система из n уравнений (формула 31) может быть записана в матричном виде (32):

$$\begin{pmatrix} I_x(x_1, y_1) & I_y(x_1, y_1) \\ I_x(x_2, y_2) & I_y(x_2, y_2) \\ \dots & \dots \\ I_x(x_n, y_n) & I_y(x_n, y_n) \end{pmatrix} \cdot \begin{pmatrix} v_x \\ v_y \end{pmatrix} = - \begin{pmatrix} I_t(x_1, y_1) \\ I_t(x_2, y_2) \\ \dots \\ I_t(x_n, y_n) \end{pmatrix} \tag{32}$$

Система (32) теперь имеет больше уравнений, чем неизвестных и поэтому она обычно переопределена. Поэтому метод Лукаса-Канада получает компромиссное решение методом наименьших квадратов, решая систему 2×2 (33):

$$(A^T A)v = A^T b \rightarrow v = (A^T A)^{-1} A^T b, \tag{33}$$

где A – матрица градиента всех пикселей, v – вектор смещения, b – вектор изменения цвета для всех точек окрестности.

Плюсы этого метода прежде всего в его локальности, можно искать поток совершенно в любой точке не придавая значению, что творится за ее пределами. Так же метод Лукаса-Канаде нечувствителен к шуму это связано с тем что, когда мы усредняем по соседям, чувствительность к шуму резко падает.

Метод показывает себя неэффективным в том случае, когда не совпадают предположение о постоянстве цвета или же происходит движение в окрестности выбранного пикселя. Для решения этой проблемы можно использовать различные улучшения метода Лукаса-Канада, в частности иерархическую схему.

Конструируется пара пирамид для последовательно идущих изображений посредством масштабирования исходных изображений (рисунок 18).

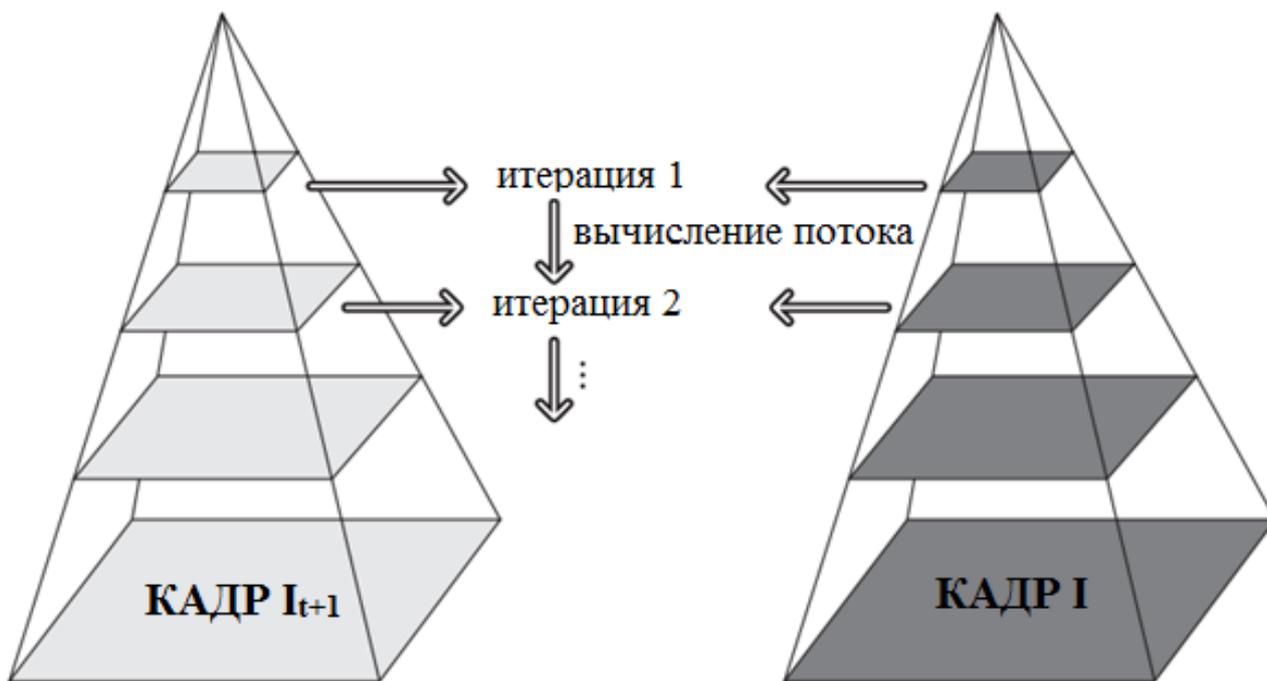


Рисунок 18. Пирамиды с несколькими уровнями разрешений

Последующий поиск выполняется от мелких изображений к более крупным, в результате чего постепенно отсекаются направления смещения, в которых заведомо не происходит движение. Максимальное движение на самом низком разрешении происходит в пределах одного пикселя. Если на некотором уровне обнаружился вектор смещения для определенного

фрагмента, то на следующем – этот вектор используется для предсказания расположения соответствующего фрагмента.

Например, для изображения I размером 640×480 изображения I_1, I_2, I_3 и I_4 при масштабирующем коэффициенте $0,5$ будут иметь размеры $320 \times 240, 160 \times 120, 80 \times 60$ и 40×30 . Наибольшей индекс пирамидальных изображений равен количеству уровней пирамиды. Поэтому выбор числа уровней пирамиды зависит от размера захваченного изображения и размера объекта, который должен быть отслежен.

Интересующая точка $p(p_x, p_y)$ и соответствующая ей точка $p_l(p_{xl}, p_{yl})$ на изображениях пирамиды вычисляются с использованием формулы 34, где l – соответствующий уровень пирамиды (34).

$$p_l = \frac{p}{2^l}, \quad (34)$$

Иерархическая схема не всегда позволяет получить качественный результат, т.к. часто невозможно увеличить/уменьшить изображение до определенных размеров и при этом не размыть необходимые признаки для поиска областей движения.

Фильтрация изображения важна во многих приложениях, включая сглаживание, удаление шума или обнаружения края. При реализации оптического потока этап предварительной фильтрации играет существенную роль не только для точного вычисления значения потока, но и для улучшения производительности. Как правило, при вычислении оптического потока фильтрация используется на начальном уровне пирамиды исходных входных изображений, и только после этого пирамида масштабируется. Классическим методом предварительная фильтрации является использования Гаусс фильтра. Определение ядра фильтра вычисляется по формуле (35) [28].

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (35)$$

где σ – стандартное отклонение распределения Гаусса.

В случае двух измерений низкочастотная и высокочастотная проекция представляет собой концентрическую окружность от центральной точки (рисунок 19).

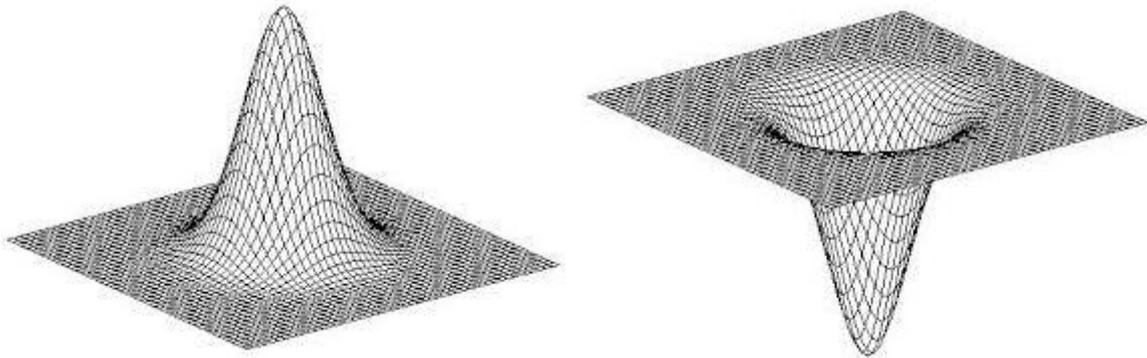


Рисунок 19. Низкочастотный и высокочастотный фильтры Гаусса

На изображении низкие частоты описывают геометрию изображения, а высокие частоты характеризуют шум. При фильтрации применяется фильтр низких частот для того, чтобы подчеркнуть геометрию изображения несущую информацию и устранить ненужные помехи (рисунок 20).

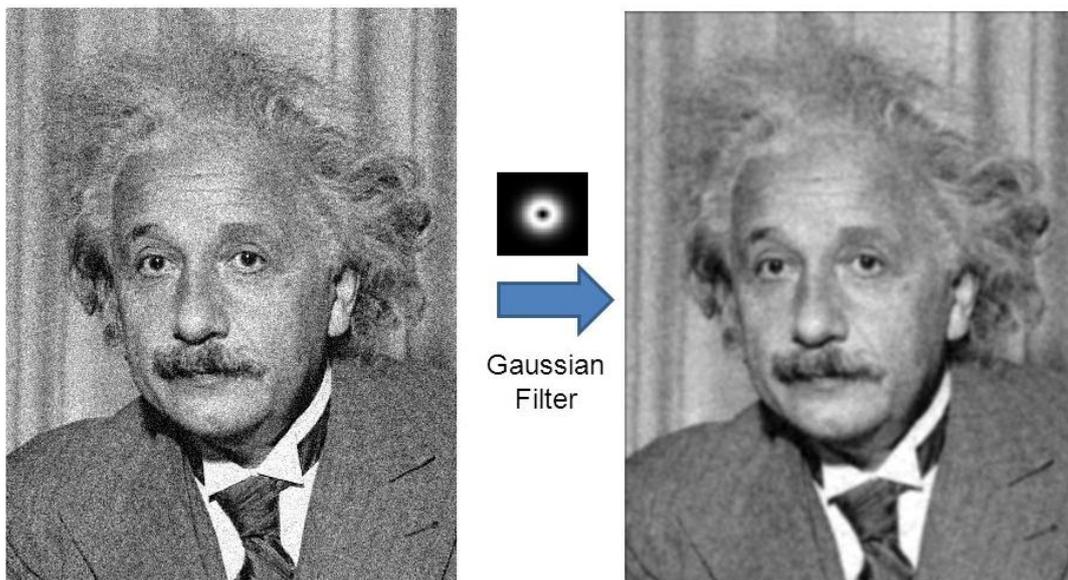


Рисунок 20. Применение фильтра Гаусса

Применение фильтра делает изображение немного размытым, но зато эффективно сглаживает мелкий шум, который может ухудшить показатели оптического потока.

2. АЛГОРИТМ ОТСЛЕЖИВАНИЯ ЛИЦА ЧЕЛОВЕКА НА ОСНОВЕ ОПТИЧЕСКОГО ПОТОКА

Трекинг лица человека описывает процесс слежения за лицом через каждый кадр видеопоследовательности, поэтому на первом шаге необходимо, чтобы лицо было обнаружено. Для этой задачи подходит алгоритм детектирования с открытым исходным кодом Виолы-Джонса, который является надежным не только для распознавания лиц, но и для класса других объектов. Обнаружение выполняется путем применения классификатора на нескольких окнах внутри изображения. Однако, если лицо искажается, каскадный классификатор может не сработать и обнаружение внезапно прервется.

Для устранения этого недостатка в данной работе лицевой трекер использует преимущества оптического потока, который описывает принадлежность пикселей локальной области найденного на кадре лица и вычисляет значения вектора скоростей между двумя последовательно идущими кадрами. В качестве алгоритма вычисления оптического потока выбран метод Лукаса-Канаде за его надежность и гибкость к различным модификациям. В данной работе были применены следующие модификации для ускорения вычисления потока: активация детектора каждые N кадров, вычисления потока для набора заданных точек, пирамидальная структура и применение фильтрации методом Гаусса на первом уровне пирамиды изображений.

Программное обеспечение написано на $C++$ в среде разработки *Microsoft Visual Studio 2013*, с использованием библиотеки *Open Computer Vision* и фреймворка *Qt* (версия 5.5).

2.1. Схема работы трекинга лиц

На рисунке 21 приведена упрощенная схема работы лицевого трекинга, разделенная на несколько этапов:



Рисунок 21. Упрощенная схема работы лицевого трекера

Входной видеопоток. Как видно из рисунка 21 алгоритм работает с потоковым видео, которое описывается как последовательность входных кадров $I_1, I_2 \dots I_n$.

Поиск лица. Детектор Виолы-Джонса на данном этапе фиксирует найденное лицо и его область.

Расстановка характерных точек. В пределах обнаруженной детектором области фиксируются характерные точки. В данном алгоритме в качестве

характерной точки выбрана середина описанного квадратом детектора лица, запоминаются его координаты и интенсивность.

Наложение фильтра на кадр. Для устранения шума применяется фильтр Гаусса со значением стандартного отклонения $\sigma = 1$.

Построение пирамидальной структуры для двух кадров. Два последовательно идущих кадра масштабируются, формируя 4 уровня разрешений.

Расчет вектора скорости. Методом Лукаса-Канаде вычисляются вектор скорости на уровнях обоих кадров - от меньшего к большему.

Визуализация смещения. В результате характерная точка на следующем кадре плавно смещается в окрестностях закрепленной области.

Обновление информации каждые N кадров. Под обновлением информации подразумевается повторная активация детектора по прошествии заданного количества кадров. Это необходимо для того, чтобы обновить положение характерных точек на лицах уже присутствующих в кадре людей, а также тех, кто только появился в поле зрения наблюдателя. Кроме того, активация детектора на каждом N-ом кадре позволяет сэкономить вычислительные ресурсы.

2.2. Выбор средств разработки

Для программной реализации предъявлены следующие основные требования:

1. Алгоритм должен быть реализован в виде программы для операционных систем семейства Windows;
2. Наличие удобного *GUI (graphical user interface)* с выводом наглядного результата и возможностью задавать свои параметры.

Поскольку вопрос производительности является немаловажным, в качестве языка программирования было решено использовать C++. Выбор обусловлен тем, что большинство имеющихся библиотек компьютерного зрения написано с помощью этого высокоуровневого языка. Подобная связка

позволит обеспечить максимально удобное взаимодействия подключения и настройки необходимых инструментов, а также хорошей скорости выполнения.

Среди библиотек компьютерного зрения с открытым исходным кодом самыми популярными являются *AForge.Net*, *ROS (Robot Operation System)* и *OpenCv*. Первые две библиотеки не подходят для данной реализации, поскольку *AForge.Net* - это *C#* фреймворк, а *ROS* применяется для создания ПО роботов. В данной работе была выбрана библиотека *OpenCV*. В настоящее время она является самой известной и наиболее развитой библиотекой компьютерного зрения. Она распространяется под лицензией *BSD* с открытым исходным кодом, в результате чего её разрешено использовать как для академических целей, так и для коммерческих. В её состав входят модули: обработки изображений, анализа движений, детектирования объектов, методов машинного обучения и т.д. Все это способствует тому, что в основе немало числа проектов и прочих библиотек компьютерного зрения лежат реализации алгоритмов из *OpenCV*.

Опираясь на вышеперечисленные выборы средой разработки была выбрана *Microsoft Visual Studio 2013. IDE* обладает огромным набором средств и возможностей, позволяя разрабатывать как консольные приложения, так и приложениям с графическим интерфейсом. Также среда позволяет добавлять сторонние плагины и библиотеки, что позволяет серьёзно расширить её функциональность.

Проектирование *GUI* выполнено с помощью библиотеки *Qt*, которая является кросс-платформенным тулkitом (*toolkit*) для быстрой разработки интерфейсов приложений на языке *C++*. Так же *Qt* предоставляет разработчику целостный фреймворк со своим набором библиотек классов, которыми можно практически полностью заменить системно-зависимый код, что существенно снижает частоту таких трудно отлавливаемых ошибок, как утечки памяти, необработанные исключения и т.д. Использование *Qt* позволяет писать значительно меньше кода, чем при использовании других библиотек. Сам же код выглядит проще, логичнее и понятнее, его легче поддерживать и развивать.

Qt предоставляет три вида лицензии: *Qt Commercial*, *GNU GPL* и *GNU LGPL*. Все они позволяют создавать ПО с собственной лицензией, при этом исходный код библиотеки одинаков для каждого типа распространения.

2.3. Поиск лица

Описанный в предыдущем разделе метод Виолы-Джонса имеет открытую реализацию в библиотеке компьютерного зрения *OpenCV*. В библиотеке реализован класс *CascadeClassifiers*, а также есть уже подобранный для лиц набор признаков Хаара *haarcascade_frontalface.xml* и *haarcascade_frontalface_alt.xml*.

Для осуществления поиска лица на вход метода классификатора *detectMultiScale()* необходимо подать кадр, вектор для областей, в которых были обнаружены лица, коэффициент увеличения скользящего окна при заходе на новый проход (по умолчанию это коэффициент имеет значение 1.1), количество соседних окон, в которых тоже должно было определиться лицо, минимальный и максимальный размер сканирующего окна (рисунок 22).

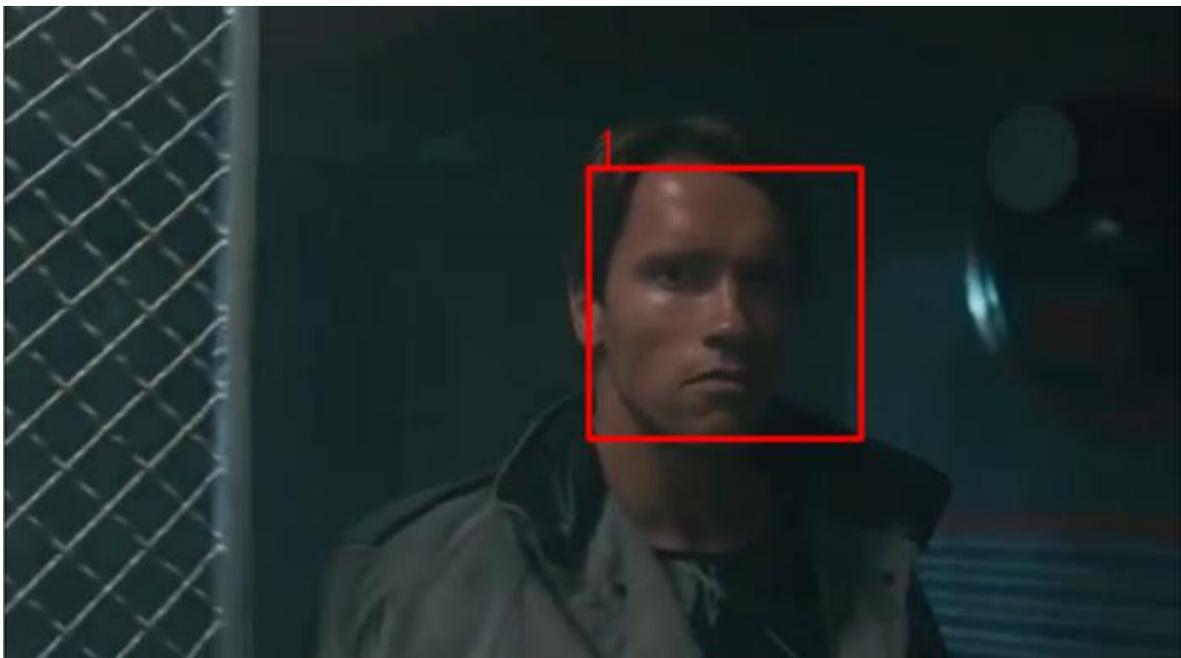


Рисунок 22. Область найденного лица

На основе описанных выше методов в программе был создан класс *FaceDetector.h* с основными методами *int FaceDetector::detect()*, где реализуется

работа классификатора, а так же метод для отображения прямоугольной области *bool FaceDetector::getFace()*.

Отсчёт N-ого кадра реализован методом в классе, отвечающем за обработку кадров в текущий момент, *FrameProcessingRunnable.h*. Цикл отсчитывает заданное число N, после чего включает методы детектора для обновления/получения информации на 1 кадр и снова передает управление оптическому потоку.

2.4. Выбор характерной точки

В качестве характерной точки было решено использовать центр обнаруженного детектором лица. Поскольку оптический поток рассматривает вектор смещения в окрестности некоторой области, то использовать информацию об одной точке p_c в предположительном центре лица вполне достаточно (рисунок Рисунок 23), чтобы максимально качественно зафиксировать его на протяжении N кадров.

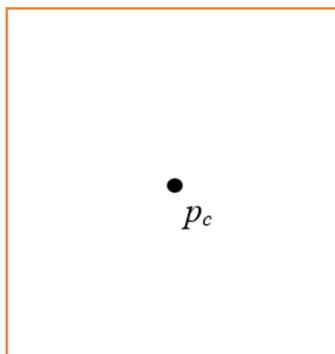


Рисунок 23. Центр обнаруженной области

Координаты центра области находятся по формуле (36):

$$p_c = \left(\frac{x_1+x_2}{2}; \frac{y_1+y_2}{2} \right), \quad (36)$$

где x_1, y_1 и x_2, y_2 – координаты верхнего левого и нижнего правого угла прямоугольной области.

На рисунке 24 приведен кадр с установленной характерной точкой.

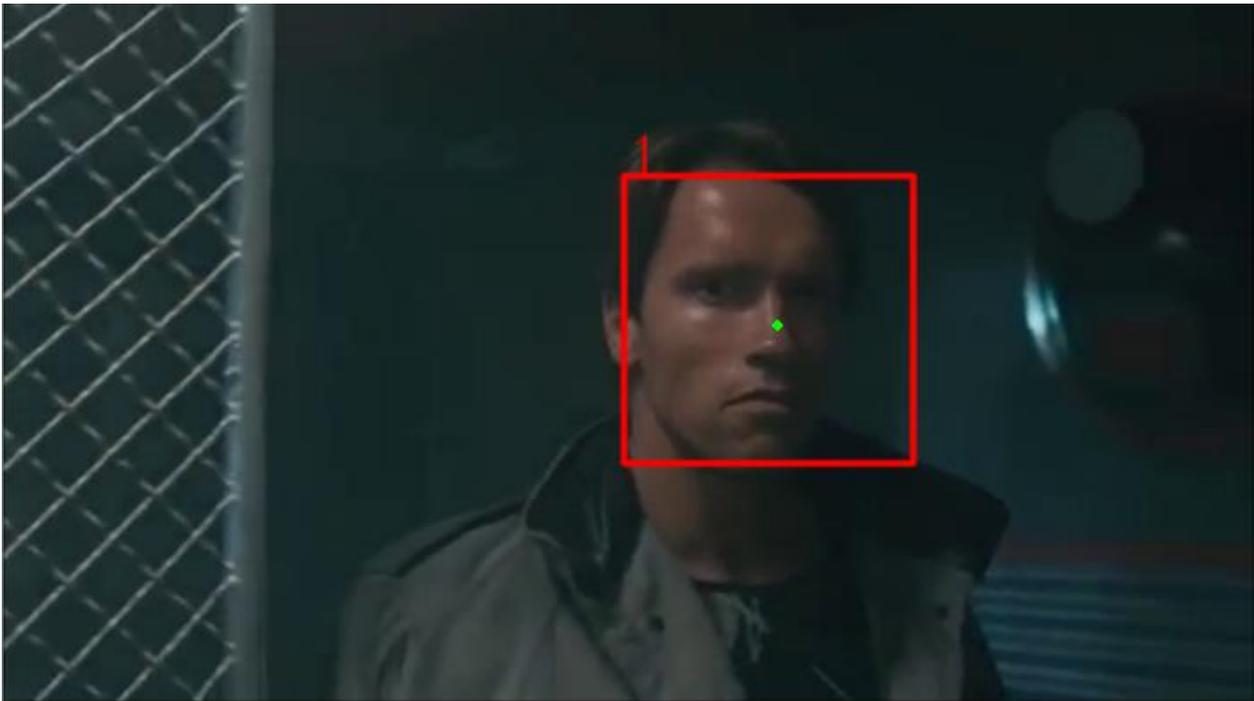


Рисунок 24. Характерная точка

За расстановку характерной точки отвечает метод *int OpticalFlow::putSpecialPoint()* класса *OpticalFlow*.

2.5. Наложения фильтра на кадр

Слабым местом алгоритма вычисления оптического потока является то, что на некоторых фрагментах изображения интенсивность может неравномерно меняться без явных закономерностей, что обусловлено наличием границ в этих областях или из-за шума в кадре. Контурные и прочие резкие перепады являются составляющей высокочастотной части Фурье спектра [28]. Применение низкочастотного фильтра Гаусса позволяет ослабить высокие частоты и пропустить низкие. Результирующее изображение будет содержать меньше резких деталей в следствие чего будет немного размытым (сглаженным). Такой результат облегчает нахождение потока по заданным характерным точкам, поскольку интенсивность в определённой окрестности нормализуется.

Для построения пространственного фильтра Гаусса (который и является фильтром низких частот) задается матрица свертки. В данной работе построение матрицы размером 3x3 реализовано в классе *GaussianFiltering.h* в приведенном ниже коде:

```

int k =3;
float div = 1.0f/(2.0f*(float)Math.PI*sigma*sigma);
float[,] A = new float[k,k];
for (int i = 0; i < k; i++)
{
    for (int j = 0; j < k; j++)
    {
        A[i, j] = div * (float)Math.Exp(-(Math.Pow(k-i-1,2)+Math.Pow(k-j-1,2))/(2*sigma*sigma));
    }
}

```

На рисунке 25 приведен пример изображения для различных значений стандартного отклонения.

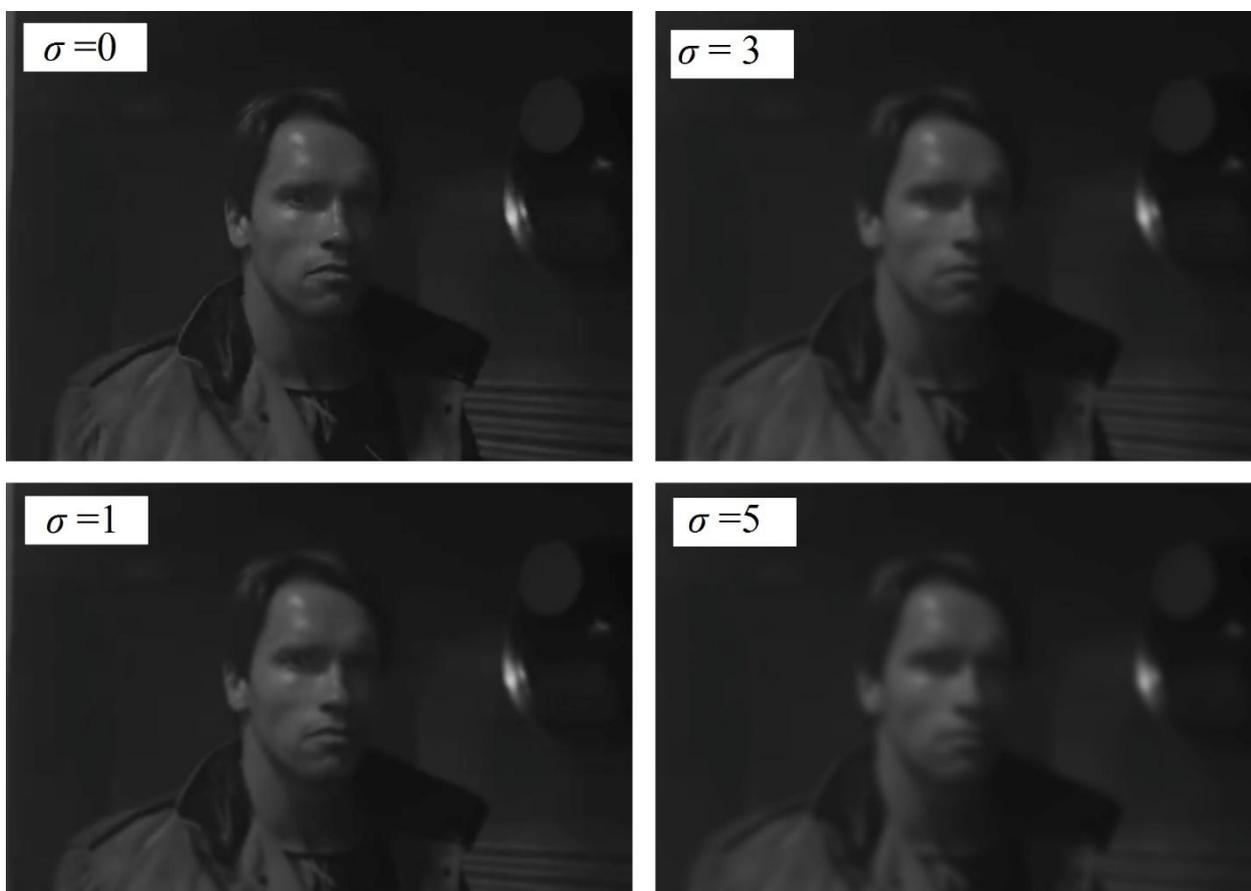


Рисунок 25. Изображение после фильтрации с разными значениями стандартного отклонения

Как видно из рисунка 25 σ при высоких значениях дает неприемлемый результат с сильным размытием. Поэтому в данной работе было решено выбрать $\sigma = 1$.

2.6. Нахождение вектора скорости

На данном этапе применяется *multi-scaling*, построение пирамиды изображений разного масштаба. В данной работе пара кадров масштабируется в 2 раза по каждой оси до получения 4 уровней (рисунок 26).

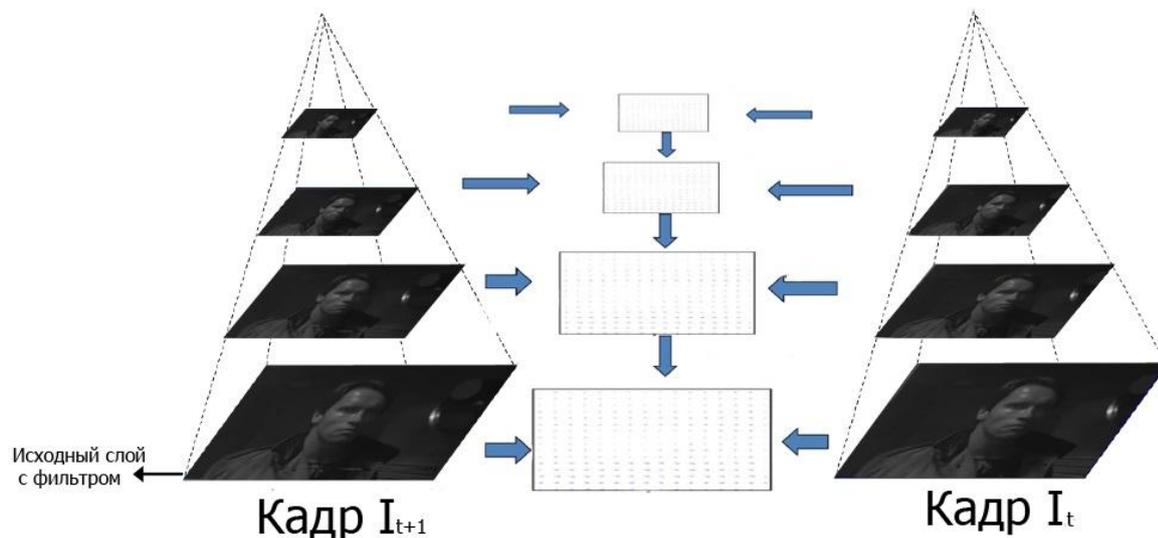


Рисунок 26. Построение пирамиды по двум кадрам

Данный способ позволяет обнаружить движение в больших пространственных масштабах, начиная анализ от меньшего изображения к большему. Сдвиг на маленьком изображении будет соответствовать сдвигу на большом изображении.

Функция `cvCalcOpticalFlowPyrLk()` используется для реализации пирамидального вычисления оптического потока методом Лукаса-Канаде. Эта функция имеет следующий набор входных параметров:

```
void cvCalcOpticalFlowPyrLk(  
    const CvArr*      imgA  
    ,const CvArr*      imgB  
    ,CvArr*            pyrA  
    ,CvArr*            pyrB  
    ,CvPoint2D32f*    featuresA  
    ,CvPoint2D32f*    featuresB  
    ,int               count  
    ,CvSize            winSize  
    ,int               level  
    ,char*             status  
    ,float*            track_error  
    ,CvTermCriteria    criteria  
    ,int               flags
```

```
);
```

Первые два аргумента – это текущее и следующее изображения (8-ми битные, одноканальные) последовательности. Следующие два аргумента – буферы, выделяемые для хранения пирамид изображений. Аргумент *featuresA* содержит координаты характерных точек, которые нужно найти, а *featuresB* – аналогичный массив, в котором размещаются координаты новых вычисленных точек из массива *featuresA*. Параметр *count* – количество точек в массиве *featuresA*. Параметр *winSize* – задает размер локальной области, в которой вычисляется значение перемещения. Параметр *level* – задает глубину пирамиды изображений, если установлен в 0, то пирамида не используется. Параметр *track_error* соответствует разности чисел между координатами конкретной отслеживаемой точки на текущем и следующем кадре. Аргумент *criteria* задает условия для окончания вычислений. Данная структура используется во многих алгоритмах для генерации необходимого условия останова.:

```
cvTermCriteria(  
    int type // CV_TERMCRIT_ITER, CV_TERMCRIT_EPS  
    ,int max_iter  
    ,double epsilon  
);
```

Первый аргумент *CV_TERMCRIT_ITER* или *CV_TERMCRIT_EPS* сообщает алгоритму момент прекращения – либо после некоторого количества итераций либо, когда метрика сходимости достигнет некоторого малого значения, а следующие два параметра (*max_iter*, *epsilon*) задают уже непосредственно сами критерии останова.

Последний аргумент *flags* предназначен для контроля внутренней организации функции.

В результате реализации алгоритма был получен рабочий прототип слежения за лицом человека, работающий по принципу «детектор-оптический поток», проиллюстрированный на рисунке 27.

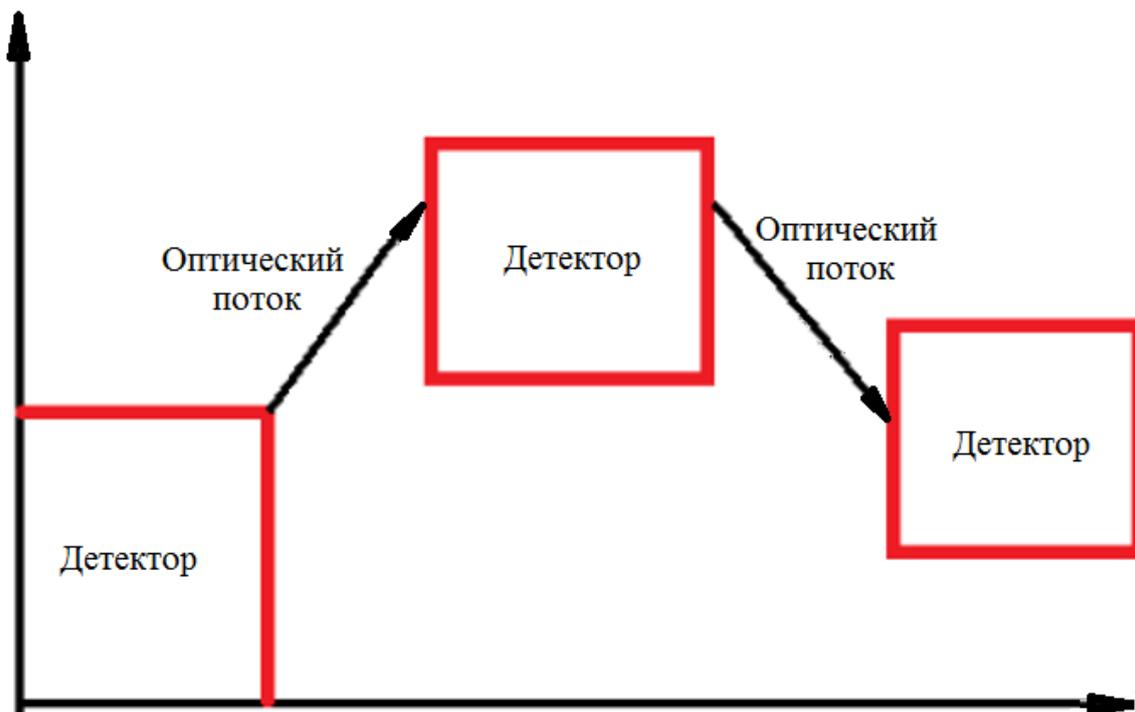


Рисунок 27. Схема последовательности движения лица в кадре

В изображения заносятся кадры видеопоследовательности с координатами найденной ранее характерной точки, которая подлежит слежению *featuresA*. После выполнения функции проверяется состояние массива для определения точек, которые удалось отследить, а затем уточняются найденные новые локации данных точек *featuresB*.

2.7. Интерфейс

Для тестирования алгоритма был разработан графический интерфейс с применением средств библиотеки *Qt*. Главной особенностью при подключении библиотеки к проекту на языке *C++* стало наличие дополнительного предпроцессора – мета-объектного компилятора (МОС). Он необходим для обработки *Qt* классов, прежде чем передать программу компилятору языка *C++*.

Разработанная программа состоит из нескольких модулей, осуществляющих основные функции и алгоритмы, применяемые в методе: модуль настройки, сохранения и загрузки конфигураций (рисунок 28), модуль

запуска и отображения результатов отслеживания (рисунок 29). В соответствии с шаблонами *Qt GUI* для интерфейса сгенерированы следующие файлы:

- `main.cpp` – файл основной программы;
- `qtfrsmainwindow.h` – заголовочный файл класса основного окна;
- `qtfrsmainwindow.cpp` – файл реализации класса окна;
- `qtfrsmainwindow.ui` – файл описания класса образа окна;
- `qtfrsconfigurationmanager.h` – заголовочный файл класса менеджера конфигураций;
- `qtfrsconfigurationmanager.cpp` – файл реализации класса менеджера конфигураций;
- `qtfrsconfigurationmanager.ui` – файл описания класса образа менеджера конфигураций.

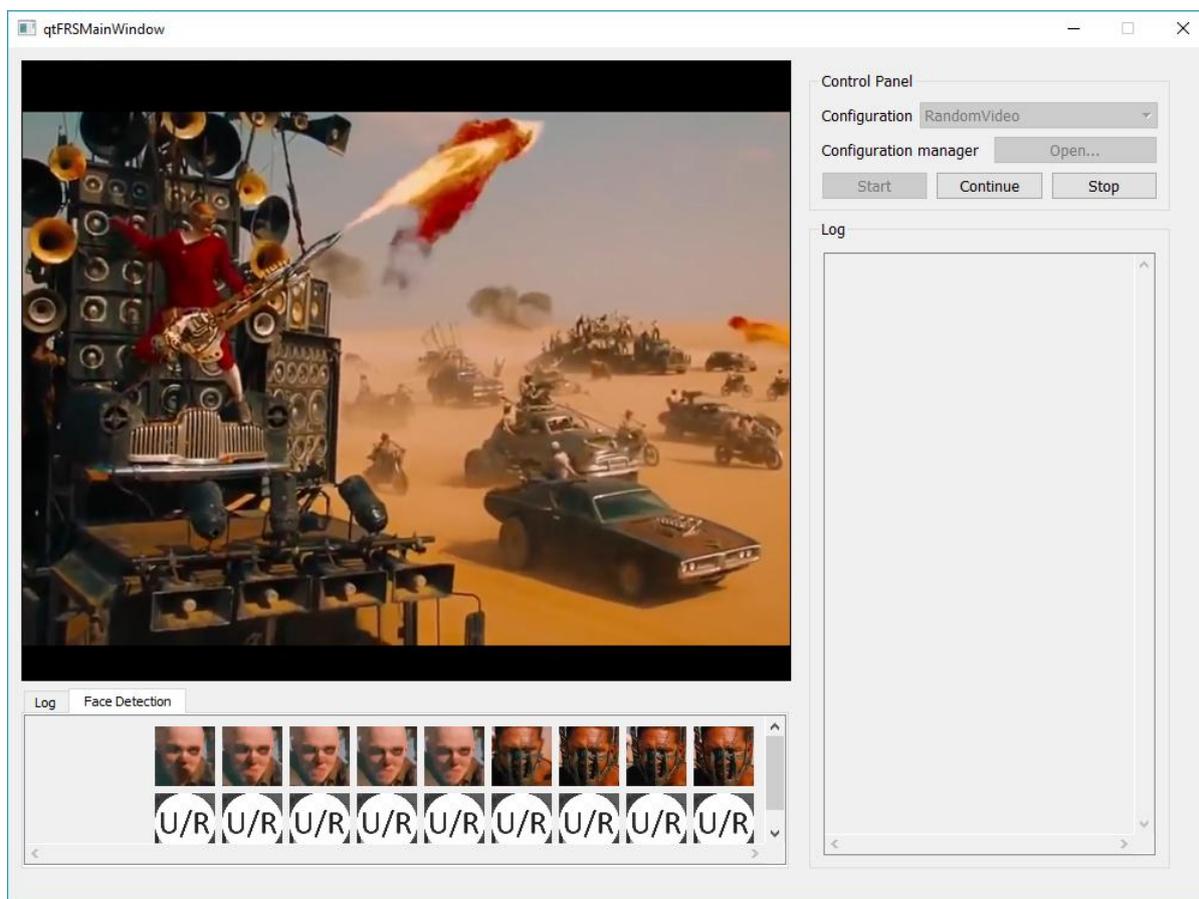


Рисунок 28. Модуль основного окна

Основное окно содержит главный экран для отображения видео (*Dashboard*), вкладку для отображения лиц, найденных детектором (*Detection*), вкладок для отображения вспомогательной информации, которая активируются при необходимости (*Log*) и панель управления (*Control Panel*) с кнопками управления видеопотоком (*Start, Pause/Continue, Stop*), выпадающим списком выбора конфигурации (*Configuration*) и менеджером настройки конфигураций (*Configuration manager*).

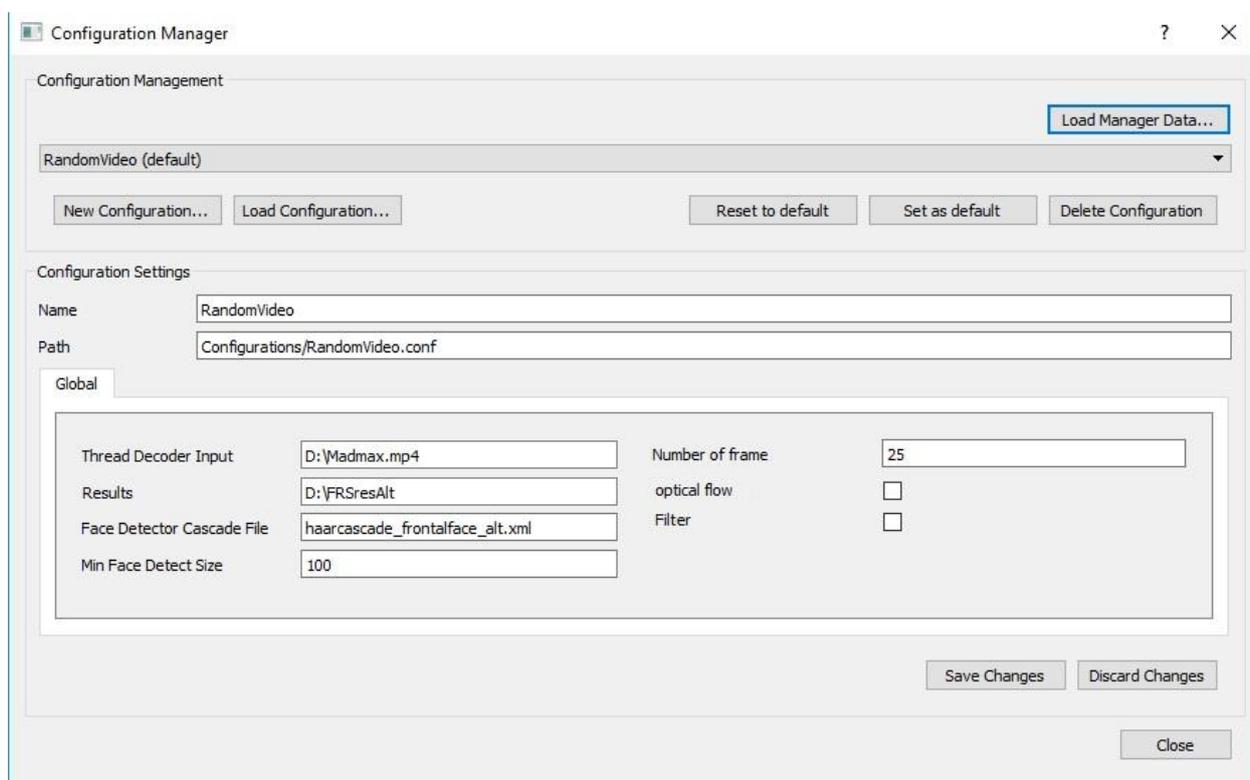


Рисунок 29. Модуль менеджера конфигураций

Для быстрой смены настроек используется меню менеджера конфигураций. Меню имеет два раздела: *Configuration Management* и *Configuration Settings*. В верхней части меню доступны следующие кнопки для переключения:

- *Load Manager Data...* - загрузка параметров по умолчанию из файла *FRSConfigManagerData.conf*;
- Выпадающий список выбора имеющихся конфигураций в папке проекта;
- *New Configuration* – создание новой конфигурации;

- *Load Configuration* – загрузка сторонних конфигураций в папку проекта;
- *Reset to default* – установка текущих параметров конфигурации аналогично конфигурации с меткой «*default*»;
- *Set as default* – установка конфигурации значения «*default*». При повторном запуске проекта по умолчанию будет выбрана именно эта конфигурация;
- *Delete Configuration* – удаление конфигурации.

В нижней части меню вводятся следующие параметры:

- *Name* – название конфигурации;
- *Path* – расположение конфигурации;
- *Thread Decoder Input* – путь к входной видеопоследовательности;
- *Result* – расположение текстового файла, в который заносится результат в виде числовых показателей;
- *Face Detector Cascade File* – путь к файлу с набором признаков Хаара;
- *Min Face Detect Size* – минимально возможный размер найденного лица, лица меньше этого значения – игнорируются;
- *Number of Frame* – значение, показывающее через какой промежуток кадров активируется детектор. Если пустой – то детектор выключен, если 0 – то работает на каждом кадре;
- *Optical flow* – при включении активирует методы вычисления оптического потока;
- *Filter* – при включении активирует фильтр кадра;
- *Save Changes* – сохранить изменения;
- *Discard Changes* – отменить изменения;
- *Close* – закрыть окно настроек менеджера конфигураций.

3. ТЕСТИРОВАНИЕ И АНАЛИЗ РЕЗУЛЬТАТОВ

Проведение тестирования, подтверждающего эффективность разработанного алгоритма, проводилось в два этапа. На первом этапе проверялась эффективность вычисления оптического потока с применением низкочастотного фильтра Гаусса на базе синтетических данных *Baker et al* [29]. На втором этапе проводилось отслеживание лица человека с использованием видеофайлов эталонной базы *Boston Head Tracking* [30]. Каждый этап эксперимента сопровождается подробным описанием использованной базы и методологией его проведения. Тестирование проводилось на компьютере под управлением операционной системы *Windows 10*, x64 обладающего следующими характеристиками: *Intel Core i-5-2410M CPU @ 2.30 Гц*, 4 Гб ОЗУ, *Intel(R) HD Graphics 3000*. Результаты тестирования методов представлены в виде диаграмм.

3.1. Тестирование на базе *Baker et al*

База данных для тестирования оптического потока *Baker et al* состоит из восьми пар изображений с набором файлов *ground-truth (GT)*, оценивающим параметры вектора скорости. Для хранения файлов *GT* используется разработанный авторами базы формат *.flo*, где хранятся координаты характерных точек изображения, а также значение смещения горизонтальной (u) и вертикальной (v) составляющей вектора скорости оптического потока. Начальные значения точек и *GT* были получены путем съемки сцены с очень маленьким шагом, в результате чего изменения на фотографиях практически не видны человеческому взгляду (рисунок 30). На поверхность каждой сцены был нанесен тонкий слой флуоресцентной краски для съемки при УФ освещении. Шаг сдвига камеры отсчитывался с помощью компьютера таким образом, что никакая точка сцены не перемещалась более, чем на 2 пикселя между двумя кадрами. Характерные точки и величина *ground-truth* были получены путем анализа изображений с высоким разрешением (3504x2336) при УФ освещении в исходной последовательности. В базе разрешение изображений преобразовано до 584x388.



Рисунок 30. Изображения базы Baker et al

Для тестирования оптического потока были выполнены следующие шаги:

- считывание двух фреймов входных изображений из базы;
- считывание *ground-truth*. Для этого в программу был добавлен модуль обработки файлов расширения *.flo*, предоставляемый на сайте [29] вместе с базой изображений;
- запуск метода оптического потока без модификаций, а после с предобработкой кадра фильтром Гаусса. Параметры запуска в менеджере

конфигураций: (*Number of frame = empty; optical flow = on; Filter = off*) и (*Number of frame = empty; optical flow = on; Filter = on*) соответственно;

– вычисление показателя эффективности.

Наиболее часто используемый показатель эффективности оптического потока – ошибка в конечной точке потока (*endpoint error / EE*) определяется по формуле (37):

$$EE = \sqrt{(u - u_{GT})^2 + (v - v_{GT})^2}, \quad (37)$$

где u, v – координаты найденного векторы скорости, u_{GT}, v_{GT} – координаты вектора *ground-truth*.

Диаграмма средних значений EE для каждого набора изображений представлена на рисунке 31.

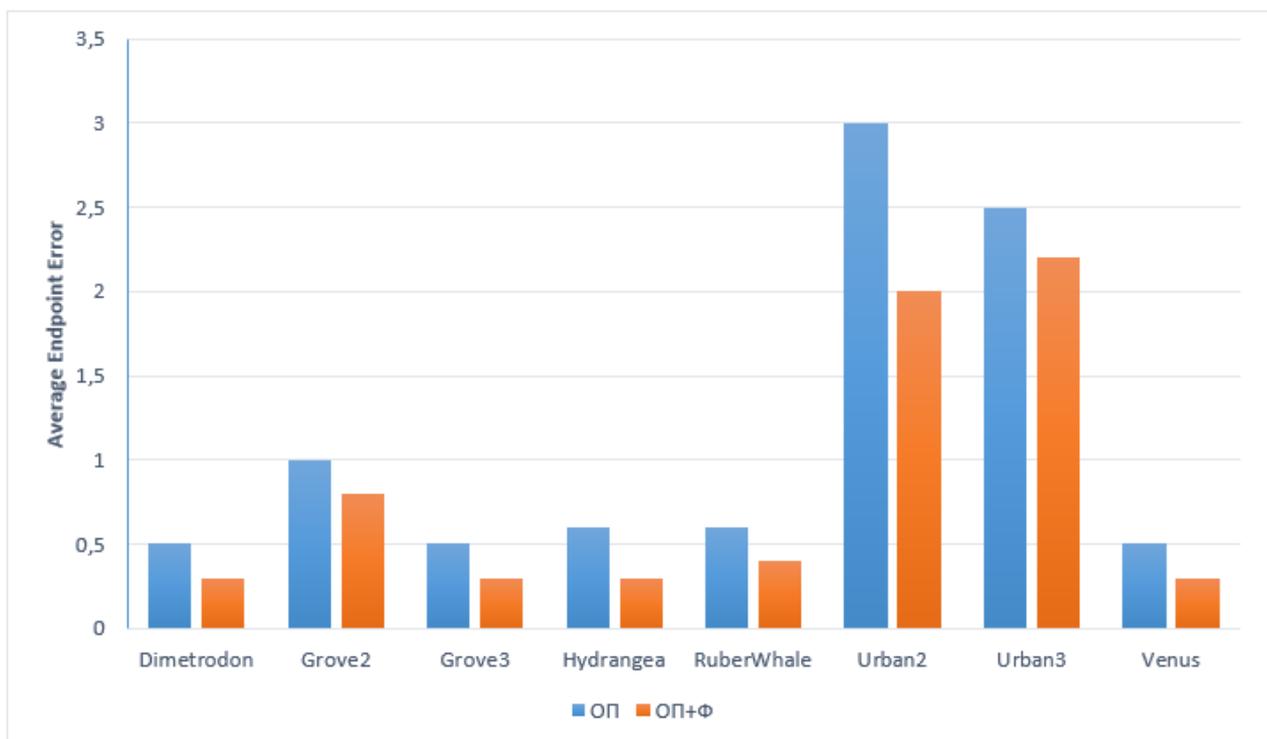


Рисунок 31. Диаграмма средних ошибок в конечных точках

Результаты, отраженные на диаграмме, показывают, что среднее значение EE получились меньше, когда изображение предварительно обрабатывали фильтром (0,8 пикселя против 1,2 пикселя). Это связано с тем, что интенсивность кадра более равномерна, в следствие чего отслеживание объекта будет происходить достовернее.

3.2. Тестирование на базе Boston Head Tracking

Предложенный алгоритм отслеживания лица был протестирован на базе данных *Boston Head Tracking* [30]. База предназначена для тестирования методов отслеживания головы человека при наличии больших поворотов, изменений выражения лица и освещенности (рисунок 32).



Рисунок 32. Кадры из базы Boston Head Tracking

Тестирование было проведено на 50 видеороликах с разрешением 320*240 при частоте 30 кадров в секунду. На каждом видеоролике представлен один человек, который двигает головой в разные стороны на протяжении 6 секунд. Общее количество кадров для каждого видео 199.

Метод, с которым сравнивается предложенный в данной работе алгоритм – это трекер на основе детектора лица Виолы-Джонса. Условия для сравнения методов следующие: трекер на основе детектора работает на каждом кадре, а при подключении оптического потока активируется на каждом десятом кадре.

Для тестирования алгоритма были выполнены следующие шаги:

- загрузка видеоролика через менеджера конфигураций;
- поочередная обработка видеоролика двумя методами: трекером на основе детектора, а затем трекером на основе детектора + ОП. Для каждого способа в менеджере конфигураций задаются следующие настройки: (*Number of frame = 1; optical flow = off; Filter = off*) и (*Number of frame = 10; optical flow = on; Filter = on*) соответственно.
- собранные в текстовый файл данные анализируются для вычисления среднего показателя обнаружений на кадре.

Работа двух сравниваемых методов показана на рисунке 33.

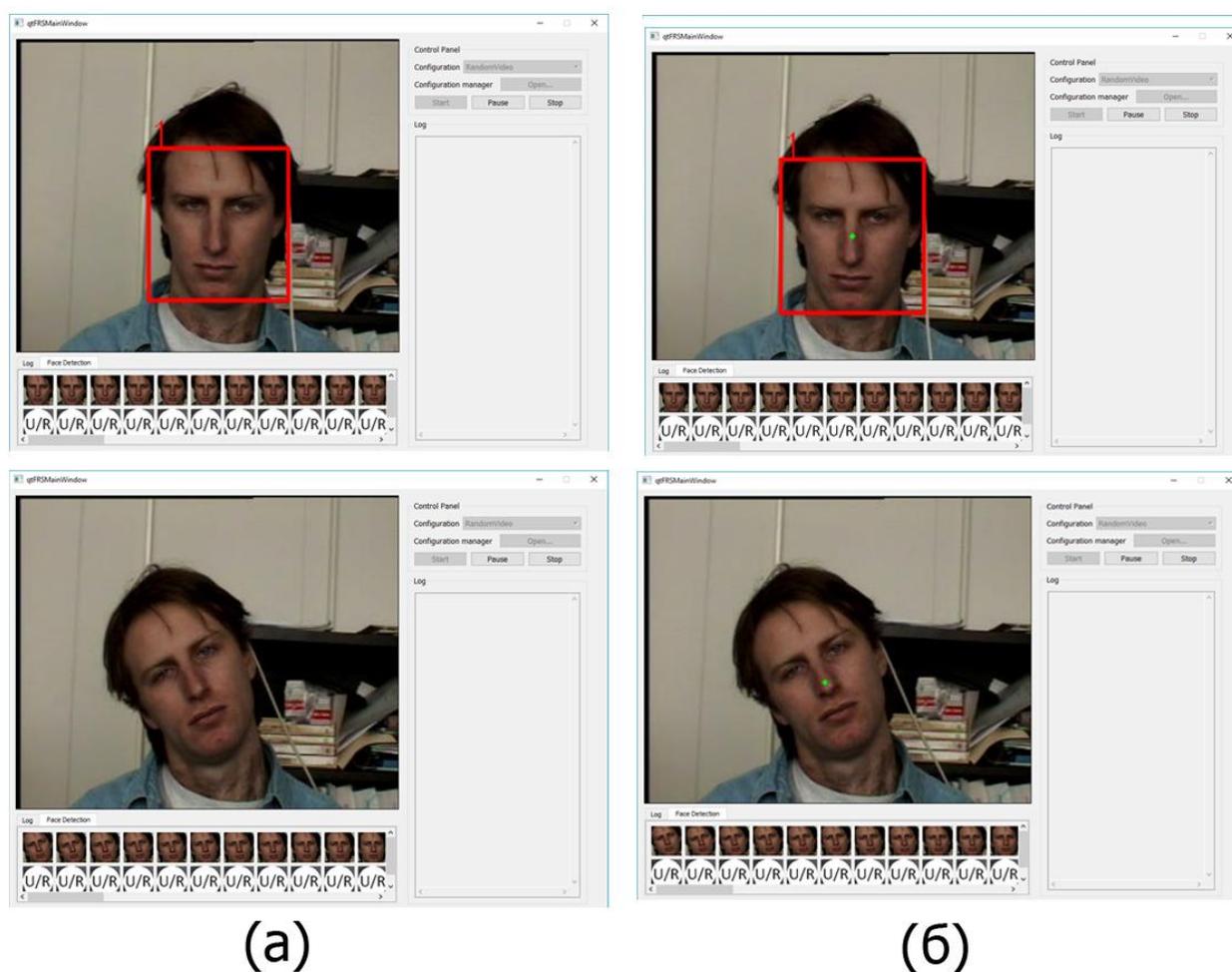


Рисунок 33. Процесс работы методов: (а) детектор; (б) детектор+ОП

На рисунке 33(а) видно, что лицевой трекер, основанный только на детекторе, теряет объект при нетривиальном положении головы, в то время как при использовании связки детектор + оптический поток (рисунок

33(б))фокусировка остается, что позволяет не потерять зафиксированный объект.

Качество отслеживания Vd (38) вычисляется путем суммирования кадровых обнаружений лиц (d_i) в видео и делением полученной суммы на общее количество кадров (I). Найденное значение Vd выражается в процентах (38):

$$Vd = \frac{\sum_{i=1}^n d_i}{I} * 100\%. \quad (38)$$

Ни рисунках 34 - 38 приведены диаграммы с полученным значением Vd для каждого метода. Пять диаграмм соответствуют набору из 10 видео для 5 разных людей. На горизонтальной оси указаны названия видеофайлов.

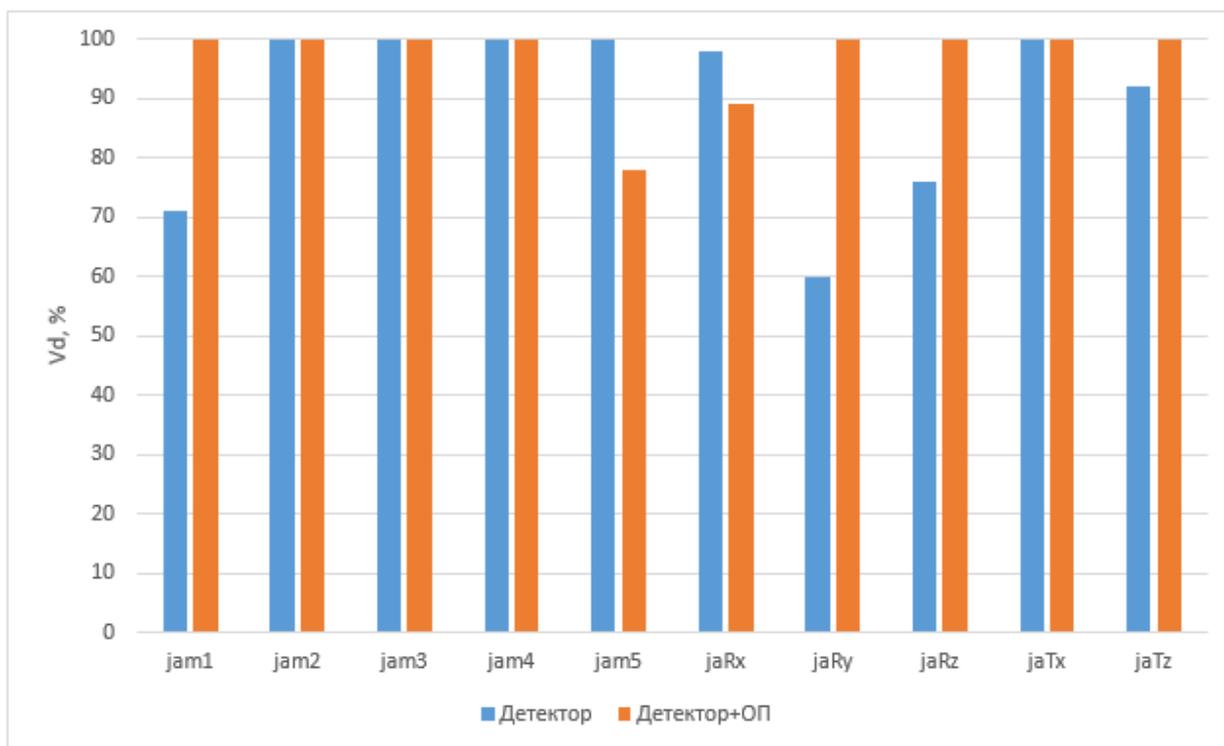


Рисунок 34. Процент обнаруженных кадров с лицами для первого набора видеофайлов

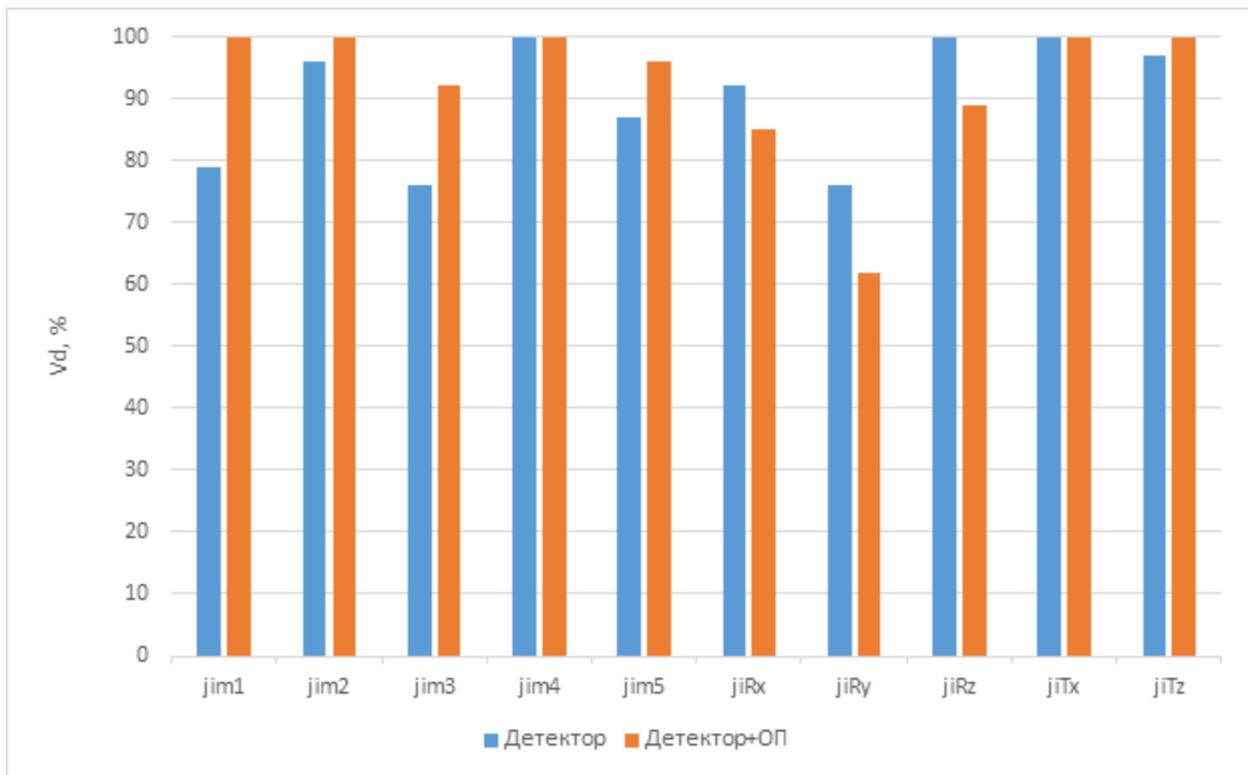


Рисунок 35. Процент обнаруженных кадров с лицами для второго набора видеофайлов

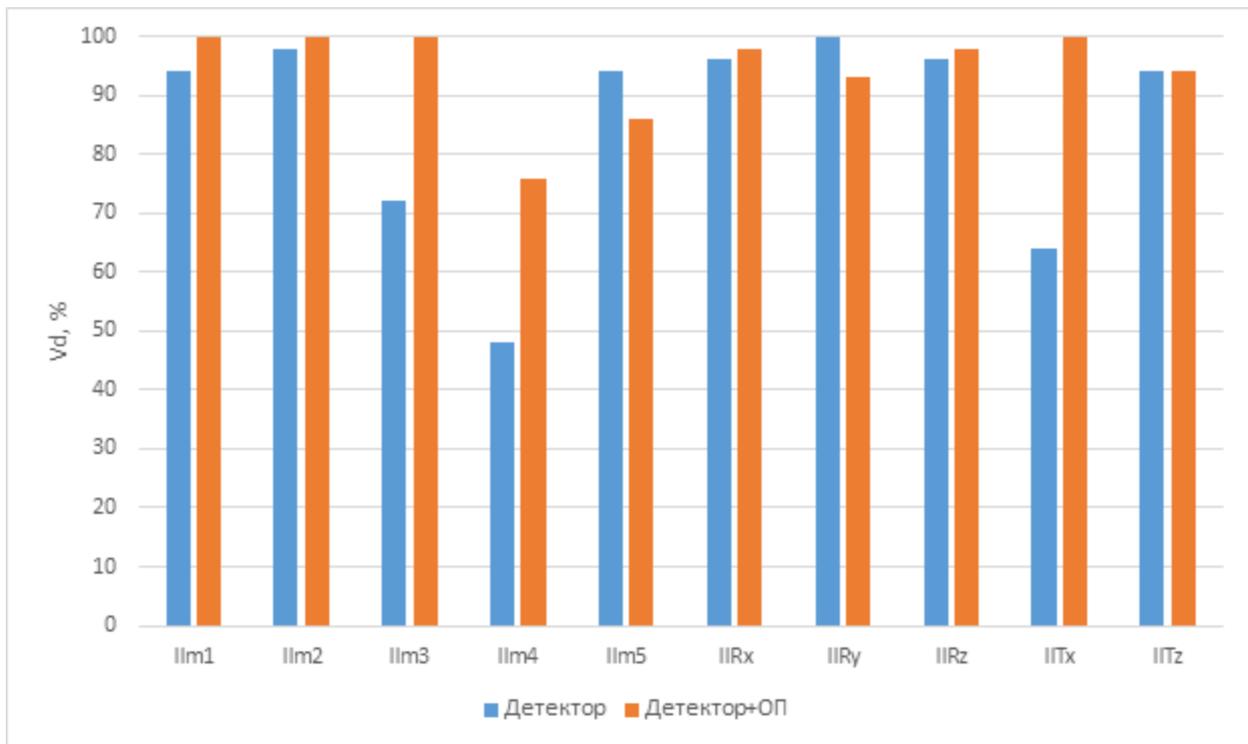


Рисунок 36. Процент обнаруженных кадров с лицами для третьего набора видеофайлов

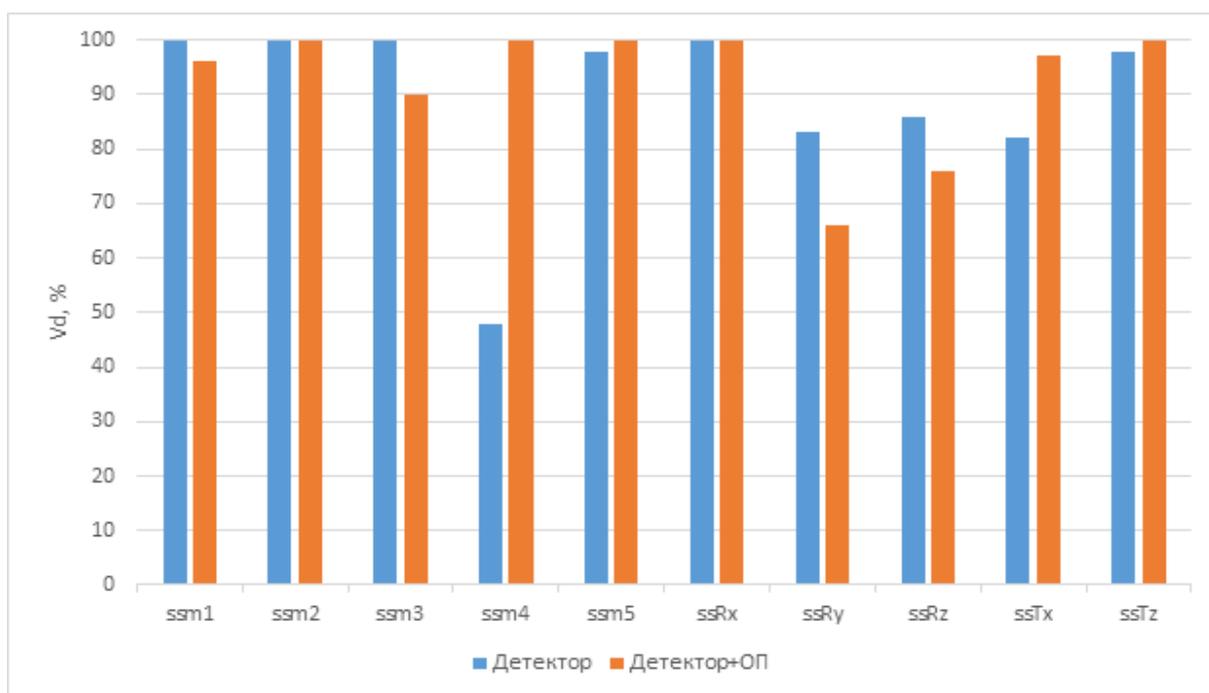


Рисунок 37. Процент обнаруженных кадров с лицами для четвертого набора видеофайлов

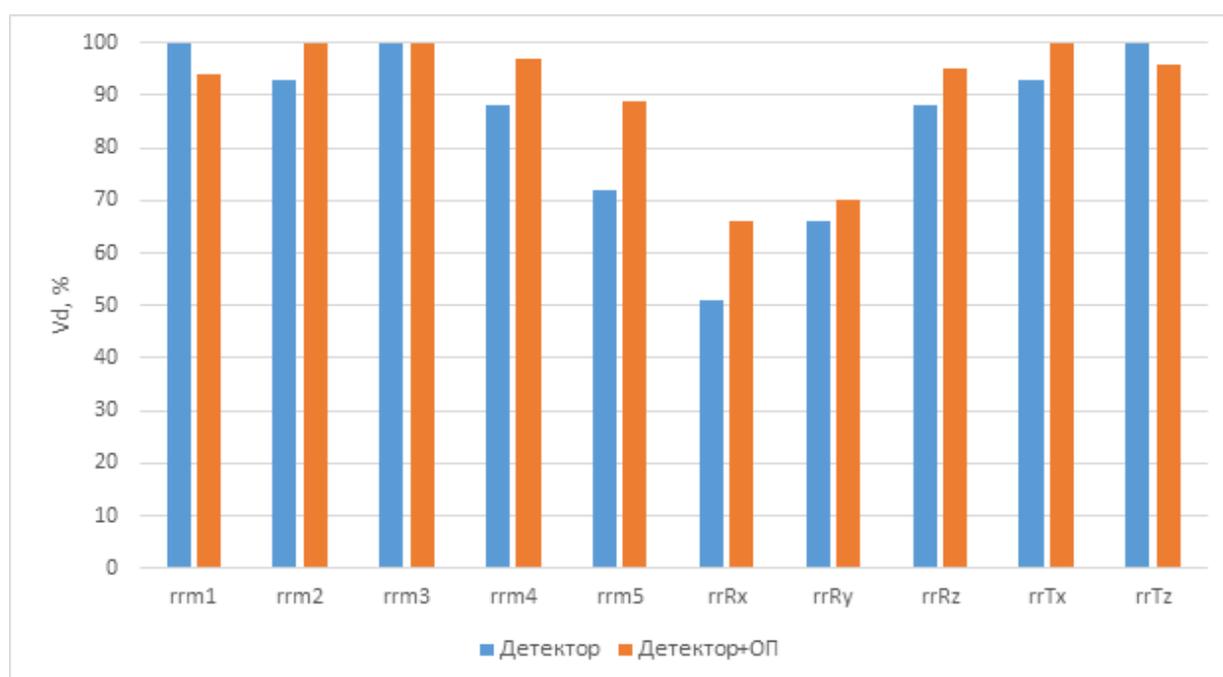


Рисунок 38. Процент обнаруженных кадров с лицами для пятого набора видеофайлов

Средняя длина трека \overline{Trk} (38) вычисляется путем суммирования всех треков trk_i (последовательности кадров от начала отслеживания и до прерывания) для видео и деления на их общее количество n :

$$\overline{Trk} = \frac{\sum_{i=1}^n trk_i}{n}. \quad (39)$$

На рисунке 39 приведена диаграмма средней длины трека \overline{Trk} для каждого набора данных.

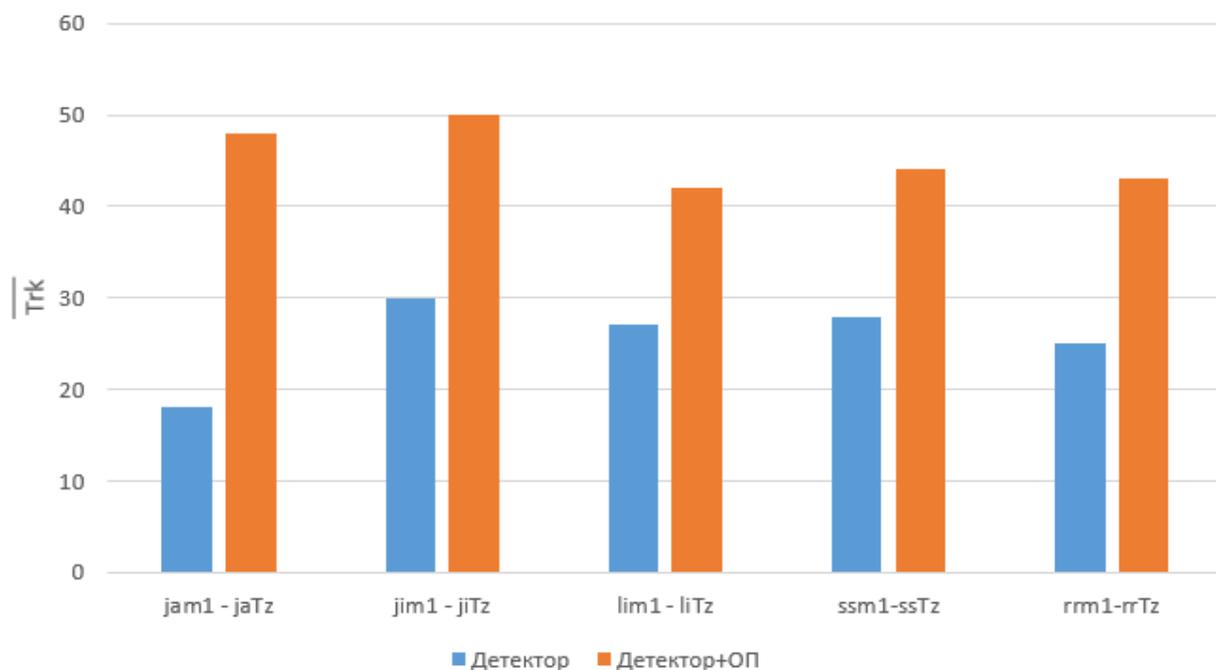


Рисунок 39. Средняя длина трека

В результате средний показатель обнаружения для всех видео составляет 88,04% при средней длине трека 26 кадров для отслеживания на основе детектора ВД и 93,36% для разработанного алгоритма отслеживания лиц на основе ОП при средней длине трека 45 кадров.

4. ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ

Целью данного раздела является определение оценки коммерческого потенциала, перспективности и альтернатив проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения, а также планирование и формирование бюджета научных исследований, определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования.

Научно-исследовательская работа направлена на разработку метода отслеживания лица человека на основе применения оптического потока.

Преимуществом такого подхода является эффективное взаимодействие лицевого детектора с методом оценки движения на основе оптического подхода, при котором движущийся объект определяется с высокой точностью. За счет того, что детектор обновляет информацию каждые n кадров, достигается оптимальное использование ресурсов вычислительной системы.

Алгоритм, разрабатываемый в рамках данной работы, может быть использован в системах наблюдения и обеспечения безопасности на объектах с ограниченным доступом.

4.1. Организация и планирование работ

При организации процесса реализации конкретного проекта необходимо оптимально планировать занятость каждого из его участников и сроки проведения отдельных работ.

На данном этапе составляется полный перечень проводимых работ, определяются их исполнители и оптимальная продолжительность. Наглядным результатом планирования работ является сетевой, либо линейный график реализации проекта. В большинстве случаев предпочтительным является линейный график. Перечень работ приведен в таблице 1.

Таблица 1 – Перечень работ и загрузка исполнителей

Этапы работы	Исполнители	Загрузка исполнителей
Постановка целей и задач, получение исходных данных	НР	НР – 100%
Составление и утверждение ТЗ	НР, И	НР – 90% И – 10%
Подбор и изучение материалов по тематике	НР, И	НР – 30% И – 70%
Разработка календарного плана	НР, И	НР – 90% И – 10%
Обсуждение литературы	НР, И	НР – 50% И – 50%
Выбор проектных решений	НР, И	НР – 30% И – 70%
Реализация алгоритма	НР, И	НР – 10% И – 90%
Тестирование алгоритма, анализ результатов	НР, И	НР – 30% И – 70%
Оформление пояснительной записки	И	И – 100%
Подведение итогов	НР, И	НР – 50% И – 50%

Примечание к таблице 1:

НР – научный руководитель.

И – исполнитель.

4.1.1. Продолжительность этапов работ

Расчёт продолжительности этапов работ может осуществляться двумя методами: технико-экономическим или опытно-статистическим.

Используем опытно-статистический метод, который реализуется двумя способами: аналоговым или вероятностным.

Для определения ожидаемого значения продолжительности работ $t_{ож}$ применяется вероятностный метод (40) – метод двух оценок t_{min} и t_{max} .

$$t_{ож} = \frac{3t_{min} + 2t_{max}}{5}, \quad (40)$$

где t_{min} – минимальная трудоемкость работ, чел/дн.;

t_{max} – максимальная трудоемкость работ, чел/дн.

Для выполнения перечисленных в таблице 2 работ требуются специалисты: исполнитель и научный руководитель.

Рассчитаем длительность этапов в рабочих днях, а затем переведем в календарные дни по формуле (41):

$$T_{РД} = \frac{t_{ож}}{K_{ВН}} \cdot K_{Д}, \quad (41)$$

где $t_{ож}$ – трудоемкость работы, чел/дн.;

$K_{ВН}$ – коэффициент выполнения работ ($K_{ВН} = 1$);

$K_{Д}$ – коэффициент, учитывающий дополнительное время на компенсации и согласование работ ($K_{Д} = 1.2$).

Расчёт продолжительности этапа в календарных днях ведется по формуле (42):

$$T_{КД} = T_{РД} \cdot T_{К}, \quad (42)$$

где $T_{РД}$ – продолжительность выполнения этапа в рабочих днях;

$T_{КД}$ – продолжительность выполнения этапа в календарных днях;

$T_{К}$ – коэффициент календарности.

Коэффициент календарности рассчитывается по формуле (43):

$$T_{К} = \frac{T_{КАЛ}}{T_{КАЛ} - T_{ВД} - T_{ПД}}, \quad (43)$$

где $T_{\text{КАЛ}}$ – календарные дни ($T_{\text{КАЛ}} = 365$);

$T_{\text{ВД}}$ – выходные дни ($T_{\text{ВД}} = 52$);

$T_{\text{ПД}}$ – праздничные дни ($T_{\text{ПД}} = 14$).

$$T_{\text{к}} = \frac{365}{365 - 52 - 14} = 1,2$$

В таблице 2 приведены длительность этапов работ и число исполнителей, занятых на каждом этапе.

Величины трудоемкости этапов по исполнителям $T_{\text{кд}}$ позволяют построить линейный график осуществления проекта (таблица 3).

Таблица 2 – Трудозатраты на выполнение проекта

Этап	Исполнители	Продолжительность работ, дни			Длительность работ, чел/дн.			
		t_{min}	t_{max}	$t_{ож}$	Т _{рд}		Т _{кд}	
					НР	И	НР	И
Постановка целей и задач, получение исходных данных	НР	2	4	2,8	3,36	-	4,088	-
Составление и утверждение ТЗ	НР, И	2	4	2,8	3,024	0,336	3,6792	0,4088
Подбор и изучение материалов по тематике	НР, И	12	15	13,2	4,752	11,088	5,7816	13,4904
Разработка календарного плана	НР, И	2	4	2,8	3,024	0,336	3,6792	0,4088
Обсуждение литературы	НР, И	4	6	4,8	2,88	2,88	3,504	3,504
Выбор проектных решений	НР, И	6	10	7,6	2,736	6,384	3,3288	7,7672
Реализация алгоритма	НР, И	10	18	13,2	1,584	14,256	1,9272	17,3448
Тестирование алгоритма и анализ результатов	НР, И	10	15	12	4,32	10,08	5,256	12,264
Оформление пояснительной записки	И	10	15	12	-	14,4	-	17,52
Подведение итогов	НР, И	4	6	4,8	2,88	2,88	3,504	3,504
Итого:				76	28,56	62,64	34,75	76,21

Таблица 3 – Линейный график работ

Этап	НР	И	Март			Апрель			Май			Июнь	
			10	20	30	40	50	60	70	80	90	100	110
1	4,088	-	■										
2	3,6792	0,4088	■ 										
3	5,7816	13,49		■ ■	■								
4	3,6792	0,4088			■ 								
5	3,504	3,504				■ ■							
6	3,3288	7,7672				■ ■							
7	1,9272	17,344					■ ■						
8	5,256	12,264						■ ■					
9	-	17,52								■ ■			
10	3,504	3,504									■ ■		

НР - ■ И - ■

4.1.2. Расчет накопления готовности проекта

Величина накопления готовности работы показывает, на сколько процентов выполнена работа по окончанию текущего этапа.

Степень готовности определяется формулой (44):

$$CG_i = \frac{TP_{Hi}}{TP_{общ}} \cdot 100\%, \quad (44)$$

где TP_{Hi} – нарастающая трудоемкость с момента начала работы i -го этапа; $TP_{общ}$ – общая трудоемкость проекта.

Заполним таблицу 5 величинами CG_i (%) и удельным весом этапа TP_i (%).

Таблица 5 – Нарастание технической готовности работы и удельный вес каждого этапа

Этап	TP_i , %	CG_i (%)
Постановка целей и задач, получение исходных данных	3,684211	3,684211
Составление и утверждение ТЗ	3,684211	7,368421
Подбор и изучение материалов по тематике	17,36842	24,73684
Разработка календарного плана	3,684211	28,42105
Обсуждение литературы	6,315789	34,73684
Выбор проектных решений	10	44,73684
Реализация алгоритма	17,36842	62,10526
Тестирование алгоритма и анализ результатов	15,78947	77,89474
Оформление пояснительной записки	15,78947	93,68421
Подведение итогов	6,315789	100

4.1.3. Расчет сметы затрат на выполнение проекта

В состав затрат на создание проекта включается величина всех расходов, необходимых для реализации комплекса работ, составляющих

содержание данной разработки. Расчет сметной стоимости ее выполнения производится по следующим статьям затрат:

- материалы и покупные изделия;
- заработная плата;
- социальный налог;
- расходы на электроэнергию (без освещения);
- амортизационные отчисления;
- командировочные расходы;
- оплата услуг связи;
- арендная плата за пользование имуществом;
- прочие услуги (сторонних организаций);
- прочие (накладные расходы) расходы.

4.1.4. Расчет затрат на материалы

Так как все работы выполнялись преимущественно на ПЭВМ, то они не потребовали затрат на материалы. По данной статье можно принять $C_{\text{мат}} = 0$.

4.1.5. Расчет заработной платы

Расчет основной заработной платы выполняется на основе трудоемкости выполнения каждого этапа и величины месячного оклада исполнителя. Среднедневная заработная плата рассчитывается по формуле (45):

$$ЗП_{\text{дн-т}} = \frac{МО}{D_{\text{раб}}} \quad (45)$$

где МО – месячный оклад; $D_{\text{раб}}$ – количество рабочих дней в месяце.

Расчёты затрат на полную заработную плату приведены в таблице 6. При расчёте учитывалось, что в году 299 рабочих дня при шестидневной рабочей неделе, следовательно, в месяце 24,92 рабочих. Затраты времени на выполнение работы по каждому исполнителю брались из таблицы 4.

Для учета в составе зарплаты премий, дополнительной зарплаты и районной надбавки используется следующий ряд коэффициентов: $K_{\text{ПР}} = 1,1$, коэффициент дополнительной заработной платы $K_{\text{ДОП}} = 1,188$ для шестидневной ($K_{\text{ДОП}} = 1,113$ для пятидневной) рабочей недели и районный коэффициент $K_{\text{РК}} = 1,3$.

$$K = K_{\text{ПР}} * K_{\text{РК}} * K_{\text{доп}}$$

Таблица 6 – Затраты на основную заработную плату

Исполнитель	Оклад, руб./мес.	Среднедневная ставка, руб./день	Затраты времени, дни	К	Фонд з/платы , руб.
НР	33 162,87	1 330,77	29	1,699	65 568
И	14 874,45	596,89	63	1,592	59 865
Итого:					125 434

4.1.6. Расчет затрат на социальный налог

Затраты по этой статье составляют отчисления по единому социальному налогу (ЕСН), включающему в себя отчисления в пенсионный фонд, на социальное и медицинское страхование

Отчисления по заработной плате определяются по следующей формуле (46):

$$C_{\text{соц}} = K_{\text{соц}} * C_{\text{осн}}, \quad (46)$$

где $K_{\text{соц}}$ – коэффициент, учитывающий размер отчислений из заработной платы = 27%.

Вычислим социальные отчисления для имеющихся данных:

$$C_{\text{соц}} = 0,27 * 129 458 = 33 867,18 \text{ руб}$$

4.1.7. Расчет затрат на электроэнергию

Данный вид расходов включает в себя затраты на электроэнергию, потраченную в ходе выполнения проекта на работу используемого оборудования, рассчитываемые по формуле (47):

$$C_{\text{эл.об.}} = P_{\text{об}} * t_{\text{об}} * Ц_{\text{э}}, \quad (47)$$

где $P_{\text{об}}$ – мощность, потребляемая оборудованием, кВт; $Ц_{\text{э}}$ – тариф на 1 кВт·час; $t_{\text{об}}$ – время работы оборудования, час.

Для $Ц_{\text{э}} = 5,782$ руб./кВт·час (с НДС) заполним таблицу 7.

Наименование оборудования	Время работы оборудования $t_{\text{об}} = T_{\text{пл}} * K_t$, час	Потребляемая мощность $P_{\text{об}}$, кВт	Затраты $Э_{\text{об}}$, руб.
ПК	504*0,6	0,45	786,82
Итого:			786,82

4.1.8. Расчет амортизационных расходов

Амортизационные отчисления рассчитываются на время использования оборудования за время выполнения проекта по формуле (48):

$$C_{\text{ам}} = \frac{N_A * Ц_{\text{об}} * t_{\text{рф}} * n}{F_d}, \quad (48)$$

где N_A – годовая норма амортизации, $N_A = 33\%$; $Ц_{\text{об}}$ – цена оборудования, $Ц_{\text{об}} = 35\,000$ руб.; F_d – действительный годовой фонд рабочего времени, $F_d = 299 * 8 = 2392$ часов; $t_{\text{вт}}$ – время работы вычислительной техники во время проведения НИР, $t_{\text{вт}} = 62,64 * 8 = 501,12$ часа; n – число задействованных ПЭВМ, $n = 1$.

Вычислим амортизационные отчисления:

$$C_{\text{ам}} = \frac{0,33 * 35000 * 501,12}{2392} = 2419,71 \text{ руб}$$

4.1.9. Расчет расходов, учитываемых непосредственно на основе платежных (расчетных) документов (кроме суточных)

Так как в ходе работы не было совершено командировок, не привлекались сторонние организации, не арендовалось помещение, то данную статью расходов считаем равной нулю.

$$C_{\text{НП}} = 0 \text{ руб.}$$

4.1.10. Расчет прочих расходов

В статье «Прочие расходы» отражены расходы на разработку проекта, которые не учтены в предыдущих статьях. Их следует принять равными 10 % от единовременных затрат на разработку и вычисляют по формуле (49):

$$C_{\text{ПРОЧ}} = (C_{\text{МАТ}} + C_{\text{ЗП}} + C_{\text{СОЦ}} + C_{\text{ЭЛ.ОБ.}} + C_{\text{АМ}} + C_{\text{НП}}) * 0,1 \quad (49)$$

Рассчитаем прочие расходы, поставив числа в формулу (10):

$$C_{\text{ПРОЧ}} = (125\,434 + 33\,867,18 + 786,82 + 2419,71) * 0,1 = 16\,250,77 \text{ руб}$$

4.1.11. Расчет общей себестоимости разработки

Определим общую себестоимость, сведя все сметы затрат в таблицу 8.

Таблица 8 – Смета затрат на разработку проекта

Статья затрат	Условное обозначение	Сумма, руб
Материалы и покупные изделия	$C_{\text{МАТ}}$	0
Основная заработная плата	$C_{\text{ЗП}}$	125 434
Отчисления в социальные фонды	$C_{\text{СОЦ}}$	33 867,18
Расходы на электроэнергию	$C_{\text{ЭЛ.}}$	786,82
Амортизационные отчисления	$C_{\text{АМ}}$	2419,71
Непосредственно учитываемые расходы	$C_{\text{НР}}$	0
Прочие расходы	$C_{\text{ПРОЧ}}$	16 250,77
Итого:		178 758,48

Таким образом, затраты на разработку составили $C = 178\,758,48$ руб.

4.1.12. Расчет прибыли

Прибыль составляет 20 % от расходов на разработку. Таким образом, прибыль равна $178\,758,48 * 0,2 = 35\,751,8$ руб.

4.1.13. Расчет НДС

НДС составляет 18% от суммы затрат на разработку и прибыли. Таким образом НДС составляет $(178\,758,48 + 35\,751,8) * 0,18 = 42\,902,04$ руб.

4.1.14. Цена разработки НИР

Цена равна сумме полной себестоимости, прибыли и НДС. Таким образом $C_{\text{НИР(КР)}} = 170\,807,52 + 34\,161,5 + 42\,902,04 = 257\,412,36$ руб.

4.2. Оценка экономической эффективности проекта

Алгоритм отслеживания лица человека на основе применения оптического потока, описанный в данной работе, может быть применен в коммерческих системах обеспечения безопасности, увеличив количество объектов, которая система может анализировать одновременно. Увеличив пропускную способность в системах «свой - чужой» повышается вероятность не допустить на объект посторонних лиц. При таком подходе получаем преимущества в безопасности, однако в данном случае невозможно количественно оценить экономический эффект проекта, поскольку объект внедрения еще неизвестен.

4.2.1. Оценка научно-технического уровня НИР

Научно-технический уровень характеризует влияние проекта на уровень и динамику обеспечения научно-технического прогресса в данной области. Для оценки научной ценности, технической значимости и эффективности, планируемых и выполняемых НИР, используется метод балльных оценок.

Сущность метода заключается в том, что на основе оценок признаков работы определяется интегральный показатель (индекс) ее научно-технического уровня по формуле (11):

Используя метод бальных оценок, определяем коэффициент ее научно-технического уровня, по формуле (50):

$$K_{НТУ} = \sum_{i=1}^3 R_i \cdot n_i, \quad (50)$$

где $K_{НТУ}$ – коэффициент научно-технического уровня; R_i – весовой коэффициент i -го признака научно-технического эффекта; n_i – количественная оценка i -го признака научно-технического эффекта, в баллах.

Таблица 9 – Весовые коэффициенты признаков НТУ

Признаки научно-технического эффекта НИР	Характеристика признака НИР	Ri
Уровень новизны	Систематизируются и обобщаются сведения, определяются пути дальнейших исследований	00,4
Теоретический уровень	Разработка способа (алгоритм, программа мероприятий, устройство, вещество и т.п.)	00,1
Возможность реализации	Время реализации в течение первых лет	00,5

Таблица 10 – Баллы для оценки уровня новизны

Уровень новизны	Характеристика уровня новизны	Баллы
Принципиально новая	Новое направление в науке и технике, новые факты и закономерности, новая теория, вещество, способ	8 – 10
Новая	По-новому объясняются те же факты, закономерности, новые понятия дополняют ранее полученные результаты	5 – 7
Относительно новая	Систематизируются, обобщаются имеющиеся сведения, новые связи между известными факторами	2 – 4
Не обладает новизной	Результат, который ранее был известен	0

Таблица 11 – Баллы значимости теоретических уровней

Теоретический уровень полученных результатов	Баллы
Установка закона, разработка новой теории	10
Глубокая разработка проблемы, многоспектральный анализ, взаимодействия между факторами с наличием объяснений	8
Разработка способа (алгоритм, программа и т. д.)	6
Элементарный анализ связей между фактами (наличие гипотезы, объяснения версии, практических рекомендаций)	2
Описание отдельных элементарных факторов, изложение наблюдений, опыта, результатов измерений	0,5

Таблица 12 – Возможность реализации научных, теоретических результатов по времени и масштабам

Время реализации	Баллы
В течение первых лет	10
От 5 до 10 лет	4
Свыше 10 лет	2

Таблица 13 – Оценка уровня научно-технического эффекта

Уровень НТЭ	Показатель НТЭ
Низкий	1-4
Средний	4-7
Высокий	8-10

Для рассматриваемого проекта частные оценки уровня n_i и их краткое обоснование даны в таблице 14:

Таблица 14 – Сводная таблица оценки научно-технического уровня НИР

Фактор НТУ	Значимость	Уровень фактора	Балл	Обоснование выбранного балла
Уровень новизны	0,4	Относительно новая	4	Известные методы применены в сочетании, не описанном в известных открытых источниках
Теоретический уровень	0,1	Разработка способа	6	Разработан алгоритм позволяющий использовать детектор на n-ом кадре
Возможность реализации	0,5	В течение первых лет	10	Интеграция в системы распознавания лиц

Исходя из оценки признаков, показатель научно-технического уровня для данного проекта составил:

$$K_{НТУ} = 0,4 * 4 + 0,1 * 6 + 0,5 * 10 = 1,2 + 0,6 + 5 = 7,2$$

Согласно таблице 13 проект «отслеживание лица человека на основе применения оптического потока» имеет средний уровень научно-технического эффекта.

5. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

Представление понятия «Социальная ответственность» сформулировано в международном стандарте (МС) IS CSR-08260008000: 2011 «Социальная ответственность организации».

В соответствии с МС - Социальная ответственность - ответственность организации за воздействие ее решений и деятельности на общество и окружающую среду через прозрачное и этичное поведение, которое:

- содействует устойчивому развитию, включая здоровье и благосостояние общества;
- учитывает ожидания заинтересованных сторон;
- соответствует применяемому законодательству и согласуется с международными нормами поведения (включая промышленную безопасность и условия труда, экологическую безопасность);
- интегрировано в деятельность всей организации и применяется во всех ее взаимоотношениях (включая промышленную безопасность и условия труда, экологическую безопасность).

Введение

Объект исследования – разработка метода отслеживания лица человека на основе применения оптического потока.

Согласно техническому заданию разработанный метод будет автоматически детектировать и анализировать движение человека по лицу на видеопотоке. Функционирование данной системы осуществляет ПЭВМ и подключенная к нему по сети ip-камера. Алгоритм, разрабатываемый в рамках данной работы может быть использован в системах наблюдения и обеспечения безопасности на объектах с ограниченным доступом.

В разделе будет произведен анализ вредных и опасных производственных факторов, даны рекомендации по их снижению, а также

будет рассмотрены чрезвычайные ситуации характерные для рассматриваемого производственного объекта.

5.1. Анализ вредных и опасных факторов, которые может создать объект исследования

Согласно номенклатуре, опасные и вредные факторы по ГОСТ 12.0.003-74 делятся на следующие группы:

- физические;
- химические;
- психофизиологические;
- биологические;

Перечень опасных и вредных факторов, влияющих на персонал в заданных условиях деятельности, представлен в таблице 15.

Таблица 15 – Перечень опасных и вредных факторов при разработке ПО

Источник фактора, наименование видов работ	Факторы		Нормативные документы
	Вредные	Опасные	
<ul style="list-style-type: none"> • работа с ПЭВМ; 	<ul style="list-style-type: none"> • Повышенная или пониженная температура; • Повышенная или пониженная влажность воздуха; • Зрительное напряжение; • Отсутствие или недостаток естественного света; • Повышенный уровень шума. • Электромагнитны е излучения; 	<ul style="list-style-type: none"> •Электрический ток. 	<ul style="list-style-type: none"> •Гигиенические требования к микроклимату производственных помещений СанПиН 2.2.4-548-96; •Нормы естественного и искусственного освещения предприятий, СНиП 23-05-95; •Допустимые уровни шумов в производственных помещениях. ГОСТ 12.1.003-83. ССБТ; •Гигиенические требования к персональным электронно-вычислительным машинам и организации работы, СанПиН 2.2.2/2.4.1340-03;

5.2. Обоснование мероприятий по защите персонала предприятия от действия опасных и вредных факторов (техника безопасности и производственная санитария)

5.2.1. Требования к помещениям для работы с ПЭВМ

Основным документом, определяющим условия труда на персональных ЭВМ, являются «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы». Санитарные нормы и правила СанПиН 2.2.2/2.4.1340-03, которые были введены 30 июня 2003 года [31].

Научно-исследовательская работа выполнялась в помещении кабинета (офиса) 220а Института кибернетики, оснащённого персональными компьютерами (ПК). Под ПК будем понимать совокупность монитора, системного блока, клавиатуры, мыши и проводов для подключения.

Кабинет представляет собой комнату длиной 6 м, шириной 3 м, и высотой 3м. Естественное освещение осуществляется посредством окна размером 1,7м x 1,5м. Дверь – деревянная, одностворчатая, покрытая лаком светло-коричневого цвета. Высота двери – 2 м., ширина - 1 м. Стены комнаты окрашены вододисперсионной краской. Цвет стен – бледно-голубой. Потолок побелен белым цветом. Пол покрывает линолеум светло-серого цвета. Площадь кабинета составляет 18 м², объем – 54 м³. Также данное помещение оборудовано датчиками дыма.

Офисное помещение относится к классу помещений без повышенной опасности, так как отсутствуют условия, создающие повышенную или особо повышенную опасность.

Помещение оборудовано на два рабочих места. Пункт 3.4 СанПиН 2.2.2/2.4.1340-03 определяет требования к минимальной площади и объему на одно рабочее место. По нему при периметральном расположении площадь

одного рабочего места должна быть не менее 4,5 м². Для данного помещения это требования выполняются [31].

5.2.2. Микроклимат

Микроклимат производственных помещений – это климат внутренней среды этих помещений, который определяется действующими на организм человека сочетаниями температуры, влажности и скорости движения воздуха, а также интенсивности теплового излучения от нагретых поверхностей.

Определяющими факторами при определении микроклимата производственного помещения являются:

- температура;
- влажность воздуха;
- скорость движения воздуха.

Оптимальные значения этих характеристик зависят от сезона (холодный, тёплый), а также от категории физической тяжести работы. Для инженера-программиста она является лёгкой (Ia), так как работа проводится сидя, без систематических физических нагрузок.

Согласно требованиям [32], оптимальные параметры микроклимата в офисах приведены в таблице 16.

Таблица 16 - Оптимальные значения характеристик микроклимата

Период года	Категория работы	Температура воздуха, °С	Относительная влажность, %	Скорость движения воздуха, м/с
Холодный	Ia	22-24	40-60	0,1
Тёплый	Ia	23-25	40-60	0,1

Допустимые микроклиматические условия установлены по критериям допустимого теплового и функционального состояния человека на период 8-часовой рабочей смены. Они устанавливаются в случаях, когда по

технологическим требованиям, техническим и экономически обоснованным причинам не могут быть обеспечены оптимальные величины.

Допустимые величины показателей микроклимата на рабочих местах представлены в таблице 17.

Таблица 17 - Допустимые значения характеристик микроклимата

Период года	Категория работы	Температура воздуха, °С	Относительная влажность, %	Скорость движения воздуха, м/с
Холодный	Ia	20-25	15-75	0,1
Тёплый	Ia	21-28	15-75	0,1-0,2

Для создания благоприятных условий труда и повышения производительности, необходимо поддерживать оптимальные параметры микроклимата производственных помещений. Для этого предусмотрены следующие средства: центральное отопление в холодный период года, вентиляция (искусственная и естественная), искусственное кондиционирование.

5.2.3. Освещение

Освещение – получение, распределение и использование световой энергии для обеспечения благоприятных условий видения предметов и объектов.

В рабочем помещении сочетаются естественное освещение (через окна) и искусственное освещение (использование ламп при недостатке естественного освещения).

Светильники в помещении располагаются равномерно по площади потолка, тем самым обеспечивая равномерное освещение рабочих мест.

Освещённость на рабочем месте должна соответствовать характеру зрительной работы, который определяется наименьшим размером объекта различения, контрастом объекта с фоном и характеристикой фона. Согласно

СниП 23-05-95 разряд зрительных работ программиста относится к разряду III подразряду Г (высокой точности), параметры искусственного освещения указаны в таблице 18 [31, 33].

Таблица 18 - Нормативные значения освещённости

Характер зрительной работы	Разряд зрительной работы	Подразряд зрительной работы	Искусственное освещение		Естественное освещение КЕО e_n , % при боковом
			Освещенность при системе общего освещения, лк	Коэффициент пульсации, K_p , %	
Различение объектов высокой точности	III	Г	400	15	1,2

Рассчитаем фактическую освещенность рассматриваемого помещения. Длина и ширина аудитории равны соответственно 6 и 3 м, высота – 3 м, количество светильников $N = 3$, количество ламп внутри светильника $n=2$. Расположение светильников в кабинете показано на рисунке 40.

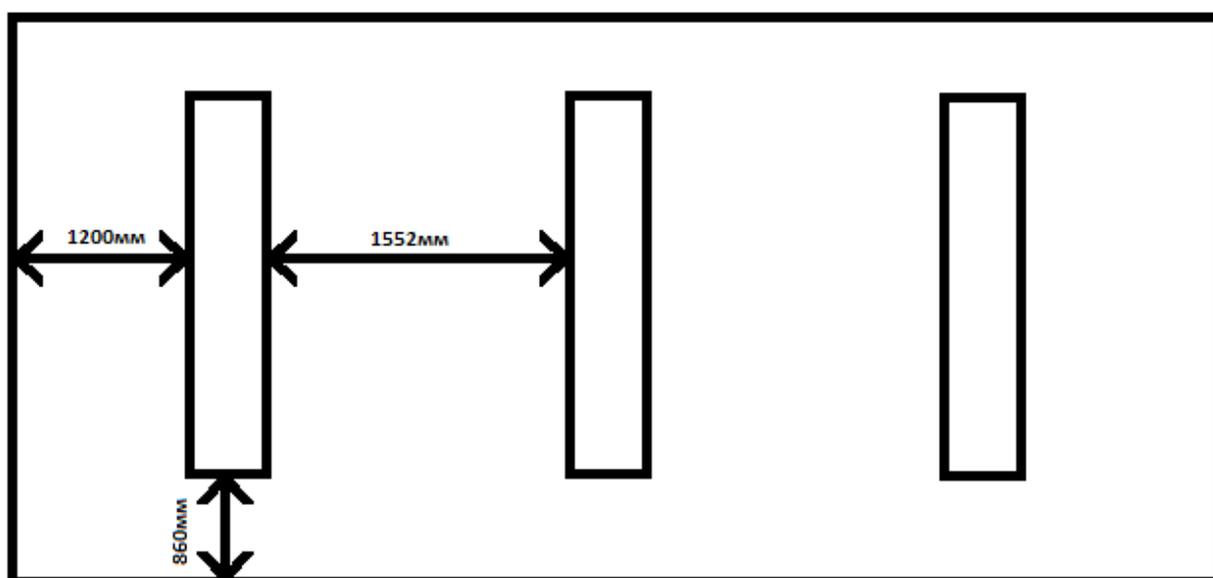


Рисунок 40. Расположение светильников в кабинете

Рассчитаем индекс помещения (51):

$$i = \frac{S}{h \cdot (A + B)}, \quad (51)$$

где i – индекс помещения; S – площадь помещения, м^2 ; h – высота помещения, м; A – длина помещения, м; B – ширина помещения.

Получили индекс помещения $i \approx 0,7$.

Так как поверхность стен окрашена в бледно-голубой цвет, с окнами без штор, то коэффициент отражения поверхности стен $P_{\text{ст}} = 50\%$. Так как поверхность потолка светлая, то коэффициент отражения поверхности потолка $P_{\text{п}} = 30\%$. Исходя из значения индекса помещения и коэффициентов отражения потолка и стен можно определить, что коэффициент использования рассматриваемого светового светильника с люминесцентными лампами люминесцентными лампами ЛБ-80 равен 38%. Рассчитаем освещенность помещения по формуле (52):

$$E_{\text{факт}} = \frac{N \cdot n \cdot \Phi_{\text{ст}} \cdot \eta}{S \cdot K_z \cdot Z}, \quad (52)$$

где $E_{\text{н}}$ – фактическая освещенность; N – число светильников в помещении; n – число ламп в светильнике; $\Phi_{\text{ст}}$ – величина стандартного светового потока, лм; η – коэффициент использования светового потока; S – площадь помещения; K_z – коэффициент запаса; Z – коэффициент неравномерности освещения.

Зная, что $\Phi_{\text{ст}} = 4800$ (лм) для люминесцентных ламп ЛБ-80, K_z для помещений с малым выделением пыли равен 1,5, а Z для люминесцентных ламп равен 1,1, рассчитаем значение фактической освещенности по формуле (52).

$$E_{\text{факт}} = 368.091 \text{ (лм)}$$

Данное значение не удовлетворяет нормативное. Рассчитаем численную оценку разности между фактическим значением освещенности и нормативным (53).

$$\Delta E = \frac{(E_{\text{факт}} - E_{\text{н}})}{E_{\text{н}}} \cdot 100\%, \quad (53)$$

где ΔE – показатель разности между фактической освещенностью и нормативной; $E_{\text{факт}}$ – фактическое значение освещенности; $E_{\text{н}}$ – нормативное значение освещенности.

Подставив значения в формулу, вычислим $\Delta E \approx 7\%$. Допустимый показатель отклонения составляет 20%, таким образом, можно сделать вывод о том, что фактическая освещенность находится в пределах допустимой нормы.

5.2.4. Шум

Шум определяется как звук, оцениваемый негативно и наносящий вред здоровью.

Основными источниками шума в помещении являются:

- система охлаждения центральных процессоров;
- жесткие диски;

При выполнении основной работы на ПЭВМ уровень шума на рабочем месте не должен превышать 50 дБ. Допустимые уровни звукового давления в помещениях для персонала, осуществляющего эксплуатацию ЭВМ при разных значениях частот, приведены в таблице 19 [34].

Таблица 19 - Допустимые уровни звука на рабочем месте

Вид трудовой деятельности, рабочее место	Уровни звукового давления, дБ, в октавных полосах со среднегеометрическими частотами, Гц									Уровни звука и эквивалентного звука (в дБА)
	331,5	663	1125	2250	5500	11000	22000	44000	88000	
Конструкторские бюро, программисты, лаборатории	88	77	66	55	44	44	44	44	33	50

В случае превышения норм необходимы мероприятия по защите от действия шума (защита временем, расстоянием, экранирования источника, рабочей зоны, замены оборудования, использование СИЗ).

Для снижения уровня шума, производимого персональными компьютерами, рекомендуется регулярно проводить их техническое обслуживание: чистка от пыли, замена смазывающих веществ; также применяются звукопоглощающие материалы.

5.2.5. Электромагнитное излучение

Воздействие электромагнитного излучения на человека зависит от напряженностей электрического и магнитного полей, потока энергии, частоты колебаний, размера облучаемого тела.

Работа проводилась на современном компьютере, где значения электромагнитного излучения малы и отвечают требованиям, которые приведены в таблице 20 [35].

Таблица 20 - Допустимые уровни электромагнитных полей

Наименование параметров	Допустимые значения
Напряженность электромагнитного поля на расстоянии 50 см. вокруг ВДТ по электрической составляющей должна быть не более: <ul style="list-style-type: none"> • в диапазоне частот 5 Гц – 2 кГц • в диапазоне частот 2 – 400 кГц 	25 В/м 2.5 В/м
Плотность магнитного потока должна быть не более: <ul style="list-style-type: none"> • в диапазоне частот 5 Гц – 2 кГц • в диапазоне частот 2 – 400 кГц 	250 нТл 25 нТл
Напряженность электростатического поля:	20 кВ/м

Основной способ снижения вредного воздействия – это увеличение расстояния от источника (не менее 50 см от пользователя). При работе за компьютером специальные экраны и другие средств индивидуальной защиты применены не были.

5.2.6. Электрический ток

В связи с наличием электрооборудования для данного производственного объекта характерным является возможность поражения

электрическим током. Для снижения данного риска необходимо соблюдать нормы электробезопасности [6].

Опасность поражения электрическим током, в отличие от прочих опасностей, усугубляется тем, что человек не в состоянии без специальных приборов обнаружить напряжение дистанционно, как, например, движущейся части, раскалённый объект, открытые люки, не ограждённые площадки на больших высотах. Опасность обнаруживается обычно уже тогда, когда человек уже поражён.

Помещение, где расположены персональные вычислительные машины, относится к помещениям без повышенной опасности [10], так как отсутствуют следующие факторы:

- Сырость;
- Токопроводящая пыль;
- Токопроводящие полы.
- Мероприятия защиты при электробезопасности следующие:
- Отключать электрооборудование при его ремонте;
- Периодически снимать электростатическое напряжение, касаясь пальцами рук, заземленных поверхностей;

Для безопасности во время гроз необходимо удостовериться о наличие молниеотвода, и того факта что все розетки в офисном помещении заземлены.

С целью защиты от поражения электрическим током, возникающим между корпусом приборов и инструментом при пробое сетевого напряжения на корпус, корпуса приборов и инструментов должны быть заземлены.

Все работы по устранению неисправностей должен производить квалифицированный персонал.

Необходимо постоянно следить за исправностью электропроводки

5.3. Экологическая безопасность

5.3.1. Анализ влияния объекта на окружающую среду

Помещение с персональным компьютером относится к пятому классу, размер санитарно-защитной зоны которого равен 50 метров, так как для работы на персональном компьютере не является экологически опасной [39].

5.3.2. Анализ влияния процесса эксплуатации объекта на окружающую среду и меры по защите окружающей среды

Основное влияние на окружающую среду заключается в образовании и поступлении твердых отходов в виде отработанных ПК, их компонентов и содержащихся в них вредных веществ. В случае выхода из строя ПК, они списываются и отправляются на специальный склад, который при необходимости принимает меры по утилизации списанной техники и комплектующих.

5.4. Безопасность в чрезвычайных ситуациях

5.4.1. Анализ вероятных ЧС, которые может инициировать объект исследования

Чрезвычайная ситуация- это обстановка на определённой территории или акватории, сложившаяся в результате аварии, опасного природного явления, катастрофы, стихийного или иного бедствия, которые могут повлечь или повлекли за собой человеческие жертвы, ущерб здоровью людей или окружающей природной среде, значительные материальные потери и нарушение условий жизнедеятельности людей [38].

На таком объекте как офисное помещение могут возникнуть такие чрезвычайные ситуации (ЧС) как:

- пожар;
- взрыв;
- природная катастрофа.

В этом разделе рассмотрим наиболее характерную ЧС, такую как пожар в офисном помещении.

Помещение по пожарной и взрывной опасности относится к категории Д (пониженной пожароопасности), характеризующейся отсутствием легковоспламеняющихся веществ и материалов в горячем состоянии [41].

5.4.2. Анализ причин, которые могут вызвать ЧС на производстве при внедрении объекта исследования

Пожар в помещении оператора может возникнуть вследствие причин неэлектрического и электрического характера [42].

К причинам неэлектрического характера относятся халатное и неосторожное обращение с огнем (курение, оставление без присмотра нагревательных приборов).

К причинам электрического характера относятся:

- неработоспособное электрооборудование, неисправности в проводке, розетках и выключателях;
- электрические приборы с дефектами;
- обогрев офисов с помощью оборудования с открытыми нагревательными элементами;
- короткое замыкание.

5.4.3. Обоснование мероприятий по предотвращению ЧС и разработка порядка действий в случае возникновения ЧС

Для того что бы избежать возникновения пожара необходимо проводить следующие профилактические работы, направленные на устранение возможных источников возникновения пожара:

- периодическая проверка проводки;
- проведение инструктажа офисных работников о пожаробезопасности.

Для того что бы увеличить устойчивость офисного помещения к ЧС необходимо устанавливать системы противопожарной сигнализации, реагирующие на дым и другие продукты горения, установка огнетушителей, обеспечить офис и проинструктировать рабочих о плане эвакуации из офиса,

а также назначить ответственных за эти мероприятия. Периодически проводить ложные тревоги, для проверки готовности офиса к ЧС. В ходе осмотра офисного помещения были выявлены системы, сигнализирующие о наличии пожара или задымленности помещения, наличие огнетушителей и средств тушения пожара. Также, ответственные за пожарную безопасность и охрану труда, периодически проводятся инструктажи и учебные тревоги.

В случае возникновения ЧС как пожар, необходимо предпринять меры по эвакуации персонала из офисного помещения в соответствии с планом эвакуации, рисунок 41.



Рисунок 41. План эвакуации при пожаре

При отсутствии прямых угроз здоровью и жизни произвести попытку тушения возникшего возгорания огнетушителем. В случае потери контроля над пожаром, необходимо эвакуироваться вслед за сотрудниками по плану эвакуации и ждать приезда специалистов, пожарников. При возникновении пожара должна сработать система пожаротушения, издав предупредительные сигналы, и передав на пункт пожарной станции сигнал о ЧС, в случае если система не сработала, по каким-либо причинам, необходимо самостоятельно произвести вызов пожарной службы по телефону 01, сообщить место возникновения ЧС и ожидать приезда специалистов.

5.5. Правовые и организационные вопросы обеспечения безопасности

5.5.1. Специальные правовые нормы трудового законодательства

В соответствии с государственными стандартами и правовыми нормами обеспечения безопасности предусмотрена рациональная организация труда в течение смены, которая предусматривает:

- Длительность рабочей смены не более 8 часов;
- Установление двух регламентируемых перерывов (не менее 20 минут после 1-2 часов работы, не менее 30 минут после 2 часов работы);

Обеденный перерыв не менее 40 минут.

Обязательно предусмотрен предварительный медосмотр при приеме на работу и периодические медосмотры.

Каждый сотрудник должен пройти инструктаж по технике безопасности перед приемом на работу и в дальнейшем, должен быть пройден инструктаж по электробезопасности и охране труда.

Предприятие обеспечивает рабочий персонал всеми необходимыми средствами индивидуальной защиты.

Оплата труда, социальные пособия, дополнительные выплаты устанавливаются в соответствии со степенью вредности и опасности выполняемых обязанностей.

5.5.2. Требования к рабочему месту при работе с ПЭВМ

Работа программиста связана с постоянной работой за компьютером, следовательно, могут возникать проблемы, связанные со зрением, также неправильная рабочая поза может оказывать негативное влияние на здоровье. Таким образом, неправильная организация рабочего места может послужить причиной нарушения здоровья и появлением психологических расстройств. На рисунке 42 показано правильная организация рабочего места [31].



Рисунок 42. Организация рабочего места.

Согласно СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы»:

- яркость дисплея не должна быть слишком низкой или слишком высокой;
- размеры монитора и символов на дисплее должны быть оптимальными;
- цветовые параметры должны быть отрегулированы таким образом, чтобы не возникало утомления глаз и головной боли.
- опоры для рук не должны мешать работе на клавиатуре;
- верхний край монитора должен находиться на одном уровне с глазом, нижний – примерно на 20° ниже уровня глаза;
- дисплей должен находиться на расстоянии 45-60 см от глаз;
- локтевой сустав при работе с клавиатурой нужно держать под углом 90° ;
- каждые 10 минут нужно отводить взгляд от дисплея примерно на 5-10 секунд;
- монитор должен иметь антибликовое покрытие;

- работа за компьютером не должна длиться более 6 часов, при этом необходимо каждые 2 часа делать перерывы по 15-20 минут;

ЗАКЛЮЧЕНИЕ

В рамках данной работы были получены следующие результаты:

1. Проведен теоретический обзор существующих методов и алгоритмов оценки движения объектов на видеопоследовательности.
2. Подобраны и обоснованы оптимальные методы, параметры и средства разработки для отслеживания лица человека.
3. Разработан алгоритм отслеживания лица человека на основе применения оптического потока, использующий в качестве локализации области поиска детектор Виолы-Джонса. Предобработка кадра осуществляется с помощью фильтра Гаусса, вычисление потока – пирамидальным методом Лукаса-Канаде.
4. Проведенное тестирование на базе оценки оптического потока показало, что средний показатель ошибки в конечной точке вектора скорости ОП при фильтрации составляет 0,8 пикселя, в то время как без фильтрации средний показатель составляет 1,2 пикселя.
5. Тестирование на базе видеофайлов дало средний показатель обнаружения кадров с лицами 88,04 % при отслеживании на основе детектора Виолы-Джонса и 93,36% для разработанного алгоритма на основе ОП. Средняя длина трека с момента фиксации лица и до момента его потери составляет для детектора ВД 25 кадров, для ОП более 45.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Forsyth, D. Computer vision: a modern approach / D. Forsyth, J. Ponce // Pearson Education Limited. – 2015. – P. 792.
2. Szeliski, R. Computer vision. Algorithms and applications / R. Szeliski // Springer US. – 2010. – P. 957.
3. Maggio, E. Video tracking theory and practice / E. Maggio, A. Cavallaro // Wiley. – 2011. – P. 292.
4. Wu, Y. Tracking appearances with occlusions // Y. Wu, T. Yu, G. Hua // Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2003. – P. 289-295.
5. Mitiche, A. Computer vision analysis of image motion by variational methods / A. Mitiche, J. K. Aggarwal // Springer US. – 2013. – P. 207.
6. Mitiche, A. Computational analysis of visual motion / A. Mitiche // Springer US. – 1994. – P. 235.
7. Harris, C. A combined corner and edge detector / C. Harris, M. Stephens // Proceedings of 4th Alvey Vision Conference. – 1988. – P. 147-151.
8. Fleet, D. Mathematical models for computer vision / D. Fleet, Y. Weiss // Springer US. – 2006. – P. 606.
9. Alatan, A. Image sequence analysis for emerging interactive multimedia services / A. Alatan, L. Onural, M. Wollborn, R. Mech, E. Tuncel, T. Sikora // IEEE Transactions on Circuits and Systems for Video Technology. – 1998. – P. 525-538.
10. Cavallaro, T. Interaction between high-level and low-level image analysis for semantic video object extraction / T. Cavallaro // EURASIP Journal on Advances in Signal Processing. – 2004. – Vol. 2008. – No. 1. – P. 786-797.
11. Bar-Shalom, Y. Tracking and data association / Y. Bar-Shalom, T. Fortmann // New York, Academic Press. – 1988. – P. 353.

12. Sundaresan, A. Multi-camera tracking of articulated human motion using shape and motion cues / A. Sundaresan, R. Chellappa // IEEE Transactions on Image Processing. – 2009. – P. 2114-2126.
13. Xiao, J. Image on 3D AAM from real-time combined 2D+3D active appearance models / J. Xiao, S. Baker, I. Matthews, T. Kanade // Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2004. – P. 535-542.
14. Freeman, W. Orientation histograms for hand gesture recognition / W. Freeman, M. Roth // Proceedings of the Workshop on Automatic Face and Gesture Recognition. – 1995. – P. 296-301.
15. Maggio, E. Hybrid particle filter and mean shift tracker with adaptive transition model / E. Maggio, C. Cavallaro // Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. – 2005. – P. 221-224.
16. Гаганов В. Сегментация движущихся объектов в видеопотоке [Электронный ресурс] / В. Гаганов, А. Конушин // Компьютерная графика и мультимедиа. – 2004. - №2(3). – Режим доступа: <http://cgm.computergraphics.ru/content/view/67> (дата обращения: 01.02.2017).
17. Heikkila, M. A texture-based method for detecting moving objects / M. Heikkila, M. Pietikainen, J. Heikkila // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2006. – P. 657-662.
18. Al-Wassai, F. The IHS transformations based image fusion / F. Al-Wassai, N. Kalyankar, A. Al-Zuky // Journal of Global Research in Computer Science. – 2011. - №5(2). – P. 70-77.
19. Tomasi, C. Good features to track / C. Tomasi, J. Shi // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 1994. – P. 593-600.

20. Arulampalam, M. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking / M. Arulampalam, S. Maskell, N. Gordon, T. Clapp // IEEE Transactions on Signal Processing. -2002. - №2(50). – P. 174-178.
21. Del Bimbo, A. Particle filter-based visual tracking with a first order dynamic model and uncertainty adaptation / A. Del Bimbo, F. Dini // Computer Vision and Image Understanding. – 2011. - №6(115). – P. 771-786.
22. Comaniciu, D. Kernel-based object tracking / D. Comaniciu, V. Ramesh, P. Meer // IEEE Transaction on Pattern Analysis and Machine Intelligence. – 2003. – P. 564-577.
23. Rahimi, L. Reducing drift in differential tracking / L. Rahimi, P. Morency, T. Darrell // Computer Vision and Image Understanding. – 2008. – №2(109). – P. 97-111.
24. Jepson, D. Robust Online Appearance Models for Visual Tracking / D. Jepson, D. Fleet, T. El-Maraghi // IEEE Transaction on Pattern Analysis and Machine Intelligence. – 2003. – P. 1296-1311.
25. Black, M. Eigenttracking: robust matching and tracking of articulated objects using a view-based representation / M. Black, A. Jepson // International Journal of computer vision. – 1998. - №1(26). – P. 63-84.
26. Viola, P. Rapid object detection using a boosted cascade of simple features / P. Viola, M. Jones // Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2001. – P. 511-518.
27. Freund, Y. A short introduction to boosting / Y. Freund, E. Robert // Journal of Japanese Society for Artificial Intelligence. – 1999. - №14(5). – P. 771-780.
28. Shapiro, L. Computer Vision / L. Shapiro, G. Stockman // Prentence Hall. – 2001. – P. 617.

29. Baker et al [Электронный ресурс]. – Режим доступа: <http://vision.middlebury.edu/flow/> (дата обращения: 17.04.2017).
30. Boston Head Tracking [Электронный ресурс]. – Режим доступа: <https://www.cs.bu.edu/groups/ivc/HeadTracking/Home.html> (дата обращения: 13.03.2017).
31. СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».
32. СанПиН 2.2.4.548 – 96. «Гигиенические требования к микроклимату производственных помещений».
33. СНиП 23-05-95. «Естественное и искусственное освещение».
34. СН 2.2.4/2.1.8.562-96 «Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой, застройки».
35. СанПиН 2.2.4.1191-03. «Электромагнитные поля в производственных условиях».
36. ГОСТ 12.1.009-76 «Электробезопасность. Термины и определения».
37. НПБ 105-95. «Определение категорий помещений и зданий по взрывопожарной и пожарной опасности».
38. СНиП 21-01-97. «Пожарная безопасность зданий и сооружений».
39. СНиП III-42-80. «Охрана окружающей среды».
40. ПУЭ «Правила устройства электроустановок. Издание 7».
41. НПБ 105-03 «Определение категорий помещений зданий и наружных установок по взрывопожарной и пожарной опасности».
42. Федеральный закон от 22.03.2013 г. №123 – ФЗ, Технический регламент о требованиях пожарной безопасности.

Приложение А

2. ALGORITHM OF A HUMAN FACE TRACKING BASED ON THE OPTICAL FLOW

Студент:

Группа	ФИО	Подпись	Дата
8ВМ5А	Гельгинберг Александр Владиславович		

Консультант кафедры ИСТ:

Должность	ФИО	Ученая степень, звание	Подпи сь	Дата
доцент	Мирошниченко Е.А.	к.т.н., доцент		

Консультант – лингвист кафедры ИЯИК:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
старший преподаватель	Рыбушкина С.В.	-		

Human face tracking describes the process of tracking a person through each frame of the video sequence, so it's necessary that the person is detected each shot. The most appropriate detection algorithm is the open source code Viola-Jones, which is reliable not only for face recognition, but also for recognition of other objects. Detection is performed by applying the classifier to several windows inside the image. However, if the face is distorted, the cascade classifier may not work and the detection will suddenly end.

This paper describes a method to overcome this disadvantage by making the face tracker take advantage of the optical flow, which describes the location of the pixels of the local area found on the face in each frame and calculates the values of the velocity between two consecutive frames. As an algorithm for calculating the optical flow, the Lucas-Canada method was chosen for its reliability and flexibility to various modifications. In this paper were used the following modifications to speed up the calculation of the flow: activation of the detector every N frames, calculation of the flow for a set of given points, pyramidal structure and using of Gaussian filtering at the first level of the pyramid image.

The software is developed in C ++ in the development environment of Microsoft Visual Studio 2013, using the Open Computer Vision library and the Qt (version 5.5) framework.

2.1. Scheme of face tracking

Figure 21 shows a simplified scheme of the functioning of the face tracking, divided into several stages:

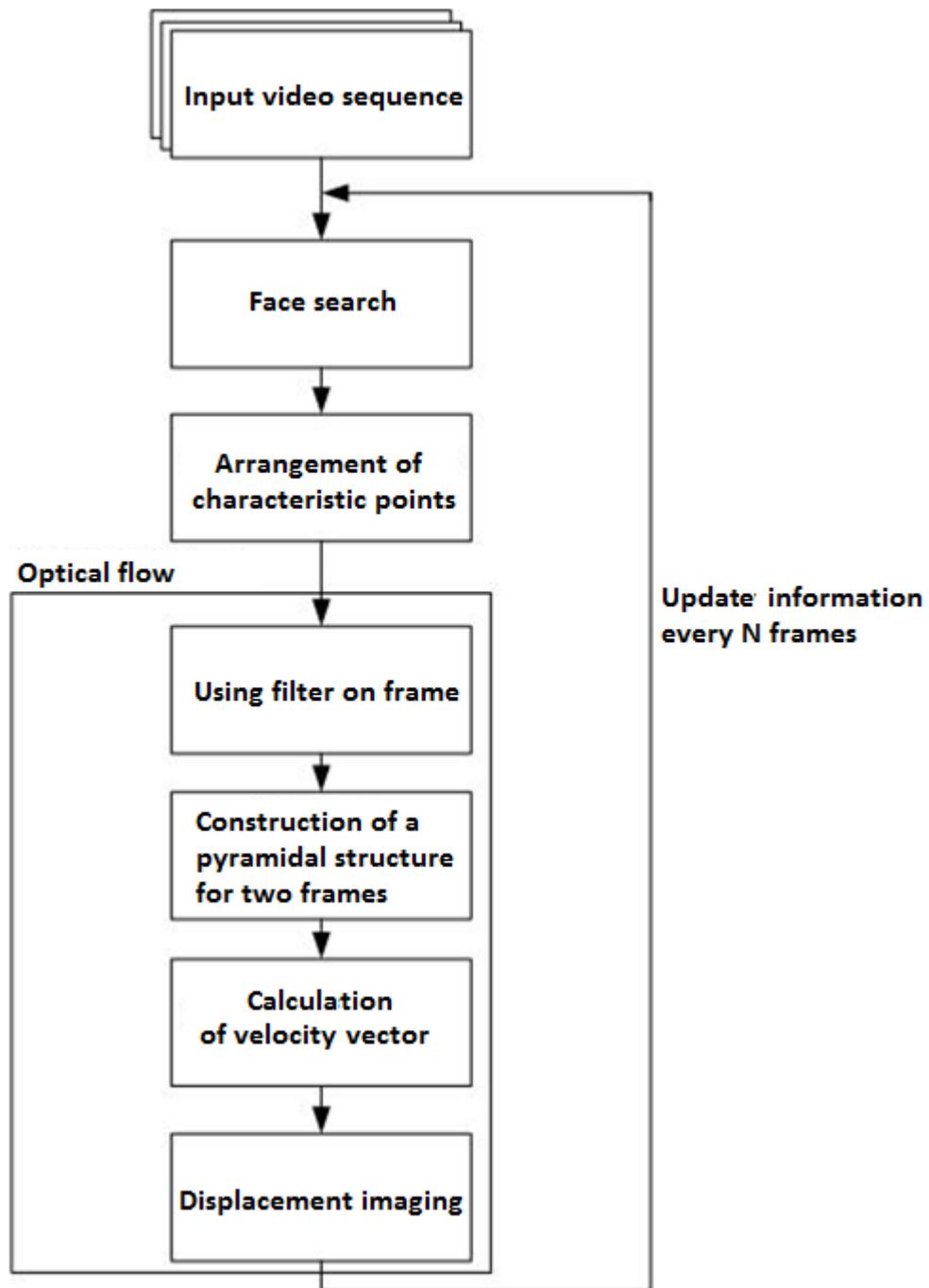


Figure 21. Simplified scheme of work of the face tracking

Input video sequence. As can be seen in figure 21, the algorithm works with streaming video, which is described as a sequence of input frames $I_1, I_2 \dots I_n$.

Face search. The Viola-Jones detector at this stage records the detected face and its area.

Arrangement of characteristic points. Within the region detected by the detector, characteristic points are fixed. In this algorithm, the midpoint of the face described by the square is chosen as the characteristic point, its coordinates and intensity are remembered.

Using filter on frame. To eliminate noise, a Gaussian filter is used with a standard deviation value of $\sigma = 1$.

Construction of a pyramidal structure for two frames. Two consecutive frames are scaled, forming 4 resolution levels.

Calculation of the velocity vector. The Lucas-Canada method calculates the velocity vector at the levels of both frames - from the smaller to the larger.

Displacement imaging. Because of the previous action, the characteristic point on the next frame is smoothly shifted in the vicinity of the fixed area.

Update information every N frames. By updating the information, it is meant to reactivate the detector after a predetermined number of frames. This is necessary in order to update the position of the characteristic points on the faces of people already present in the frame, as well as those who just appeared in the field of view of the observer. In addition, the activation of the detector on each N frame allows you to save computing resources.

2.2. Choice of development tools

For the software implementation the following basic requirements are presented:

1. The algorithm should be implemented as a program for operating systems of the Windows family;
2. The presence of a convenient graphical user interface (GUI) with a visual result output and the ability to specify its parameters.

Since the issue of performance is not unimportant, it was decided to use C++ as the programming language. The choice is due to the fact that most of the available computer vision libraries are written using this high-level language. Such

a bundle will provide the most convenient interaction of the connection and setting of the necessary tools, as well as a good speed of execution.

Among the libraries of computer vision with open source, the most popular are AForge.Net, ROS (Robot Operation System) and OpenCv. The first two libraries are not suitable for this implementation, since AForge.Net is a C # framework, and ROS is used to create robot software. In this paper, the OpenCV library was chosen. Currently, it is the most famous and most developed library of computer vision. It is distributed under the BSD license with open source, as a result of which it is allowed to be used both for academic purposes and for commercial purposes. It consists of modules: image processing, motion analysis, object detection, machine learning methods, etc. All this contributes to the fact that many other projects and libraries of computer vision are based on the implementation of algorithms from OpenCV.

Based on the above choices, as the development environment Microsoft Visual Studio 2013 was chosen. IDE has a huge set of tools and capabilities, allowing one to develop both console applications and applications with a graphical interface. Also, the environment allows a user to add third-party plug-ins and libraries, which allows to significantly expand its functionality.

The GUI is designed using the Qt library, which is a cross-platform toolkit for quickly developing application interfaces in C ++. Also, Qt provides the developer with a complete framework with its own set of class libraries, which can almost completely replace the system-dependent code, which significantly reduces the frequency of such hard-to-catch errors as memory leaks, unhandled exceptions, etc. Using Qt allows you to write much less code than using other libraries. The code itself looks simpler, more logical and understandable, it is easier to maintain and develop.

Qt provides three types of licenses: Qt Commercial, GNU GPL and GNU LGPL. They all allow you to create software with a proprietary license, while the source code of the library is the same for each type of distribution.

2.3. Face search

The Viola-Jones method described in the previous section has an open implementation in the OpenCV computer vision library. The library implements the `CascadeClassifiers` class, and also has a Haar set of Haar attributes `haarcascade_frontalface.xml` and `haarcascade_frontalface_alt.xml` already matched for individuals.

To perform the search for a person at the input of the method of the classifier `detectMultiScale ()`, it is necessary to submit a frame, a vector for the areas in which the faces were detected, the sliding window magnification factor when entering a new pass (by default this coefficient is 1.1), the number of neighboring windows in which the face must be detected, the minimum and maximum size of the scanning window must also be defined (Figure 22).



Figure 22. The area of the detected face

Based on the methods described above, the *FaceDetector.h* class was created in the program with the basic methods of *FaceDetector::detect ()*, where the classifier is implemented, as well as the method for displaying the *bool FaceDetector::getFace () rectangle*.

The N frame count is implemented by the method in the class currently responsible for processing frames at the current time, *FrameProcessingRunnable.h*. The cycle counts the specified number N, then turns on the detector's methods to update / retrieve the information for 1 frame and again transfers control to the optical stream.

2.4. Arrangement of characteristic points

As a characteristic point, it was decided to use the center of the detected face. Since the optical stream considers the displacement vector in the vicinity of a certain region, it is quite enough to use information about one point p_c in the hypothetical center of the face (Figure 23) to fix it as high as possible for N frames.

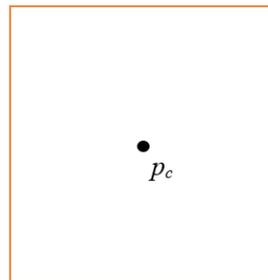


Figure 23. The center of the detected area

The coordinates of the center of the region are found from the formula (36):

$$p_c = \left(\frac{x_1+x_2}{2}; \frac{y_1+y_2}{2} \right), \quad (40)$$

where x_1 , y_1 and x_2 , y_2 are the coordinates of the upper left and lower right corner of the rectangular area.

Figure 24 shows a frame with a fixed characteristic point.

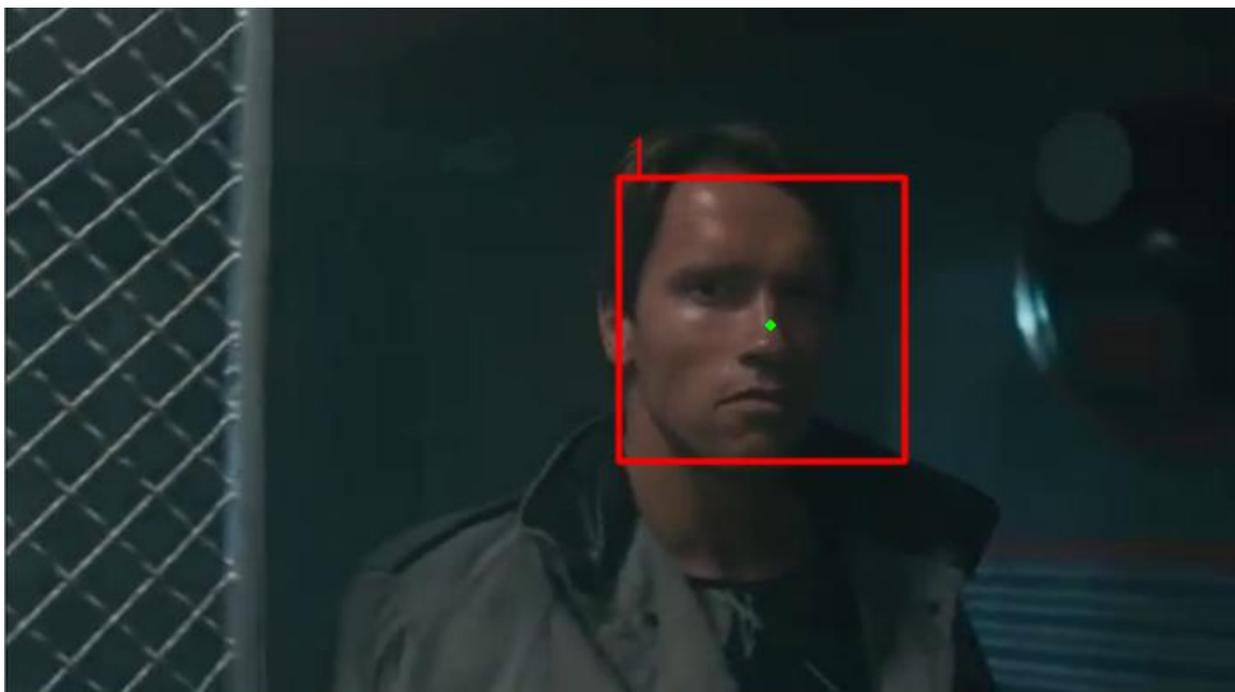


Figure 24. Characteristic point

The *int OpticalFlow::putSpecialPoint ()* method of the *OpticalFlow* class is responsible for the arrangement of the characteristic point.

2.5. Using filter on frame

The weak point of the algorithm for calculating the optical flow is that on some fragments of the image the intensity can change unevenly without explicit regularities, which is due to the presence of boundaries in these areas or due to noise in the frame. Contours and other sharp changes are a component of the high-frequency part of the Fourier spectrum. The use of a low-pass Gaussian filter allows to weaken high frequencies and to pass low ones. The resulting image will contain fewer sharp details, resulting in a slightly blurred (smoothed) image. Such a result makes it easier to find a flow at given characteristic points, since the intensity in a certain neighborhood is normalized.

To construct a Gaussian spatial filter (which is a low-pass filter), a convolution matrix is given. In this paper, the construction of a 3x3 matrix is implemented in the class *GaussianFiltering.h* in the code below:

```

int k =3;
float div = 1.0f/(2.0f*(float)Math.PI*sigma*sigma);
float[,] A = new float[k,k];
for (int i = 0; i < k; i++)
{
    for (int j = 0; j < k; j++)
    {
        A[i, j] = div * (float)Math.Exp(-(Math.Pow(k-i-
1,2)+Math.Pow(k-j-1,2))/(2*sigma*sigma));
    }
}

```

Figure 25 shows an example of an image for different standard deviation values.

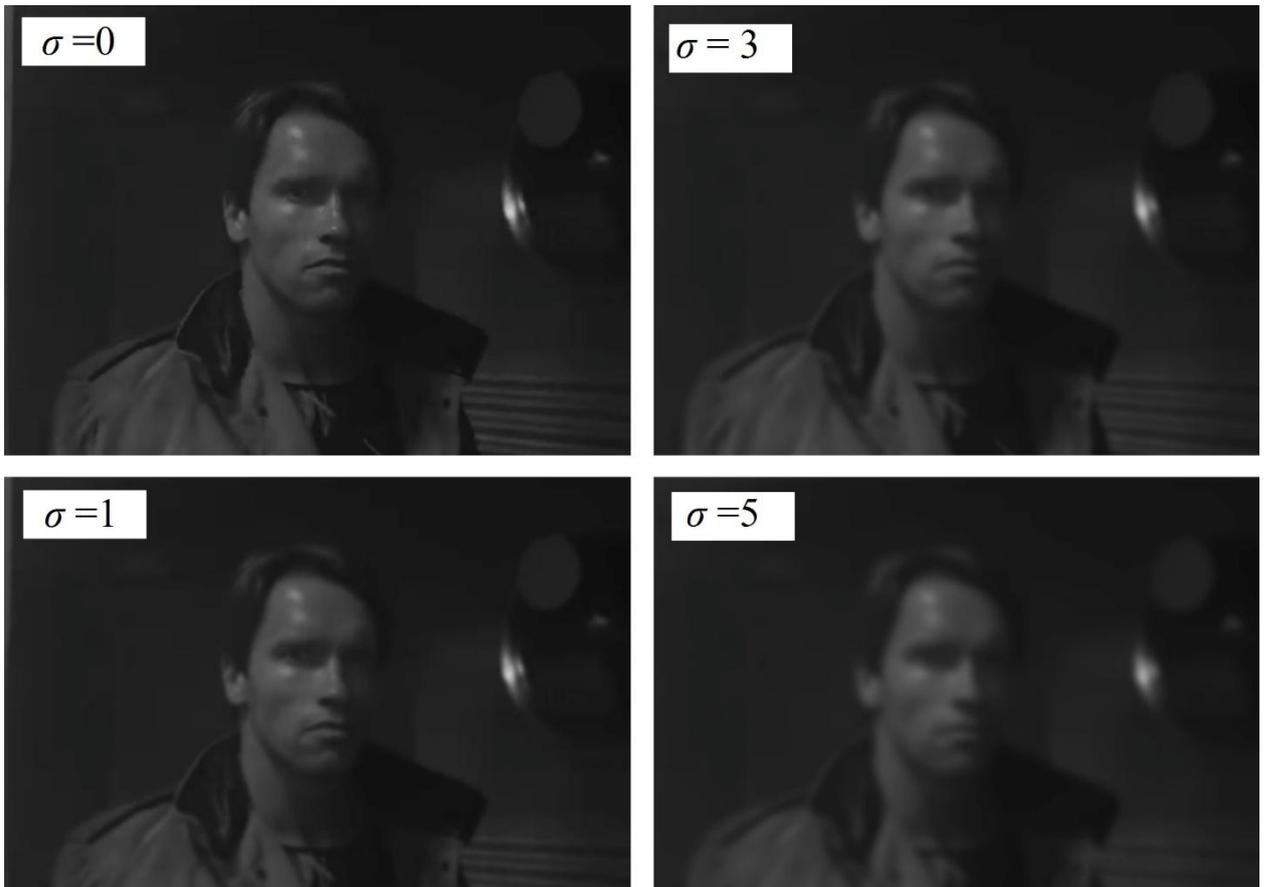


Figure 25. The image after filtering with different values of the standard deviation

As can be seen from Figure 25, σ at high values gives an unacceptable result with a strong blur. Therefore, in this paper it was decided to choose $\sigma = 1$.

2.6. Finding the velocity vector

At this stage, multi-scaling is used, the construction of a pyramid of images of different scale. In this paper, a pair of frames are scaled 2 times along each axis to obtain 4 levels (Figure 26).

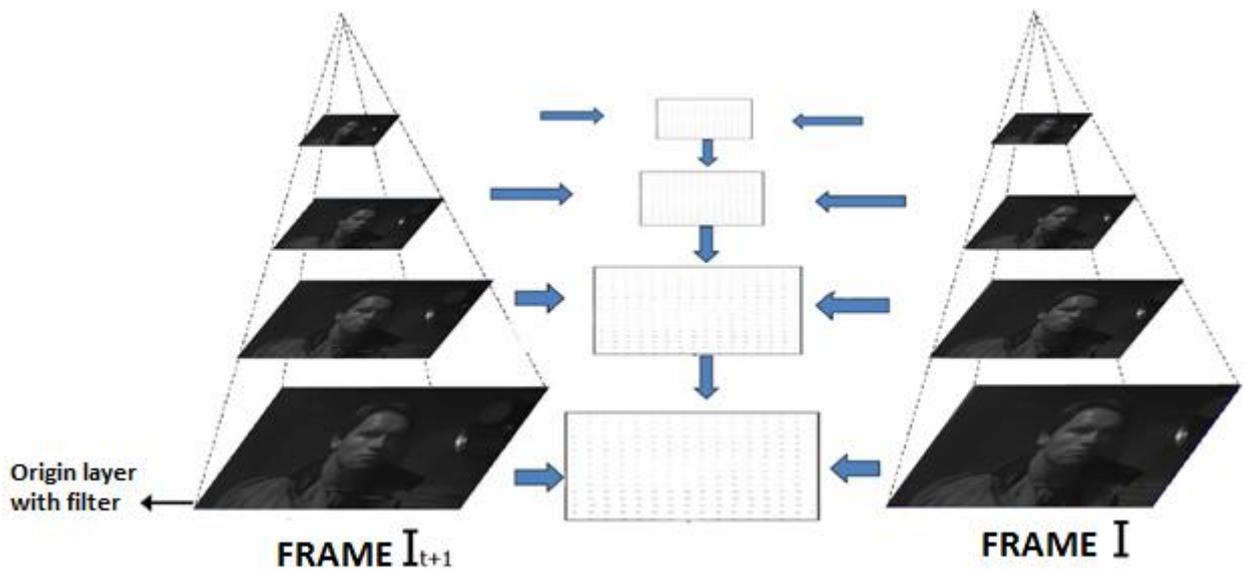


Figure 26. Construction of a pyramid in two frames

This method allows you to detect movement in large spatial scales, starting analysis from a smaller image to a larger one. The shift in the small image will correspond to the shift in the large image.

The function *cvCalcOpticalFlowPyrLk()* is used to implement the pyramidal calculation of the optical flow through the Lucas-Canada method. This function has the following set of input parameters:

```
void cvCalcOpticalFlowPyrLk(  
    const CvArr*      imgA  
    ,const CvArr*      imgB  
    ,CvArr*           pyrA  
    ,CvArr*           pyrB  
    ,CvPoint2D32f*     featuresA  
    ,CvPoint2D32f*     featuresB  
    ,int               count  
    ,CvSize            winSize  
    ,int               level  
    ,char*             status  
    ,float*            track_error  
    ,CvTermCriteria    criteria
```

```

        ,int          flags
    );

```

The first two arguments are the current and next images (8-bit, single-channel) sequences. The next two arguments are the buffers allocated to store image pyramids. The featuresA argument contains the coordinates of the characteristic points that you want to find, and featuresB is a similar array that contains the coordinates of the new calculated points from the featuresA array. The count parameter is the number of points in the featuresA array. The winSize parameter specifies the size of the local area in which the move value is calculated. Parameter level - specifies the depth of the image pyramid, if set to 0, then the pyramid is not used. The track_error parameter corresponds to the difference of numbers between the coordinates of a specific monitored point on the current and the next frame. The argument criteria specifies the conditions for completing the calculations. This structure is used in many algorithms to generate the necessary stop condition:

```

    cvTermCriteria(
        int type // CV_TERMCRIT_ITER, CV_TERMCRIT_EPS
        ,int max_iter
        ,double epsilon
    );

```

The first argument CV_TERMCRIT_ITER or CV_TERMCRIT_EPS informs the algorithm of the termination time - either after a certain number of iterations or when the convergence metric reaches a certain small value, and the next two parameters (max_iter, epsilon) specify the stop criteria themselves. The last argument (flags) is intended to control the internal organization of the function.

As a result of the implementation of the algorithm, a working prototype for tracking a person's face was obtained, working on the principle of "detector-optical flow", illustrated in Figure 27.

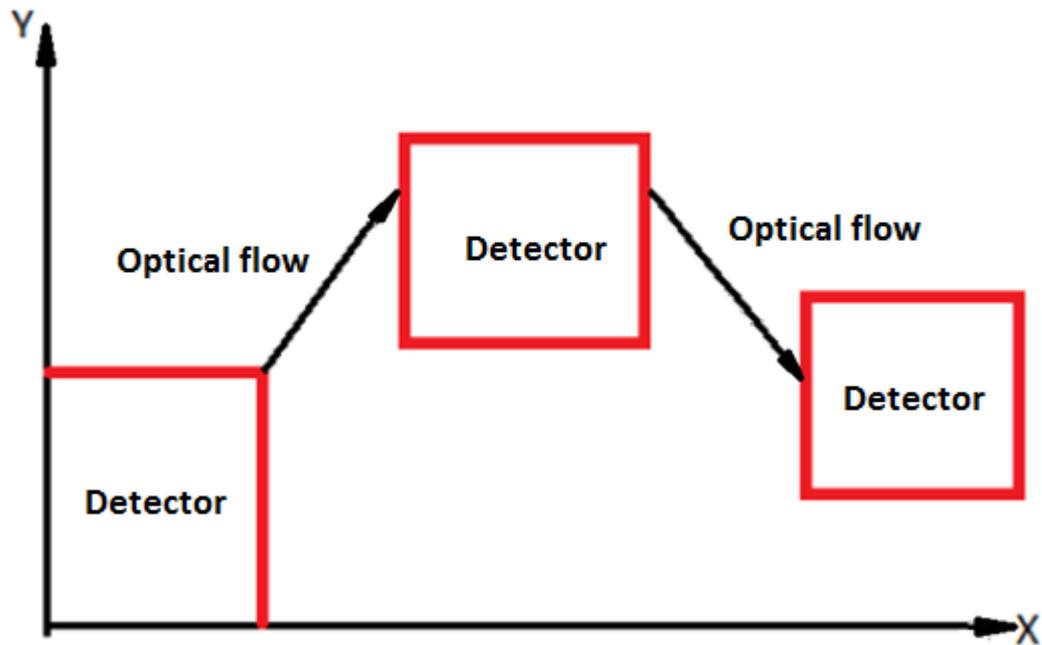


Figure 27. Diagram of the sequence of motion of a person in a frame

In the images are recorded frames of the video sequence with the value of the previously found characteristic point, which is subject to tracking features. After the function is executed, the state of the array is checked to determine the points that could be tracked, and then to check the found new locations of the featuresB data points.

2.7. Interface

To test the algorithm, a graphical interface was developed using Qt library tools. The main feature when connecting a library to a C ++ project was the availability of an additional preprocessor, the meta-object compiler (MOC). It is necessary for processing Qt classes before passing the program to the C ++ compiler.

The developed program consists of several modules that perform the basic functions and algorithms used in the method: the module for configuring, saving and loading configurations (Figure 28), the module for starting and displaying the results of the tracking (Figure 29). In accordance with the Qt GUI templates for the interface, the following files are generated:

- main.cpp - main program file;

- qtfrsmainwindow.h - header file of the main window class;
- qtfrsmainwindow.cpp - window class implementation file;
- qtfrsmainwindow.ui - a file describing the class of the window image;
- qtfrsconfigurationmanager.h - configuration manager class file;
- qtfrsconfigurationmanager.cpp - configuration manager configuration class file;
- qtfrsconfigurationmanager.ui - the configuration file of the configuration manager image class.

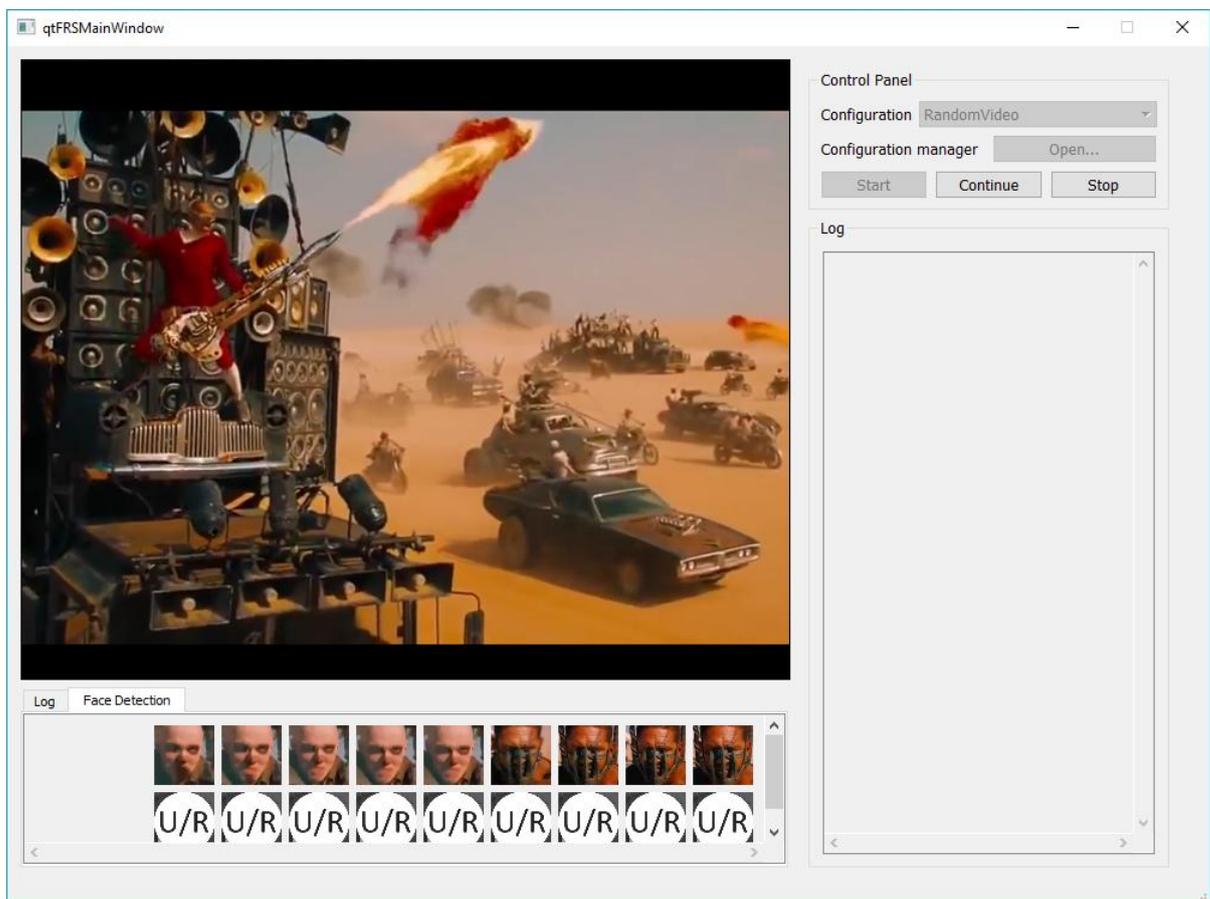


Figure 28. Main window module

The main window contains the main screen for displaying the video (Dashboard), a tab for displaying faces detected by the Detection, tabs for displaying auxiliary information that is activated when necessary (Log) and Control Panel (Control Panel) with the buttons for managing the video stream

(Start, Pause / Continue, Stop), a drop-down list of Configuration and Configuration Manager.

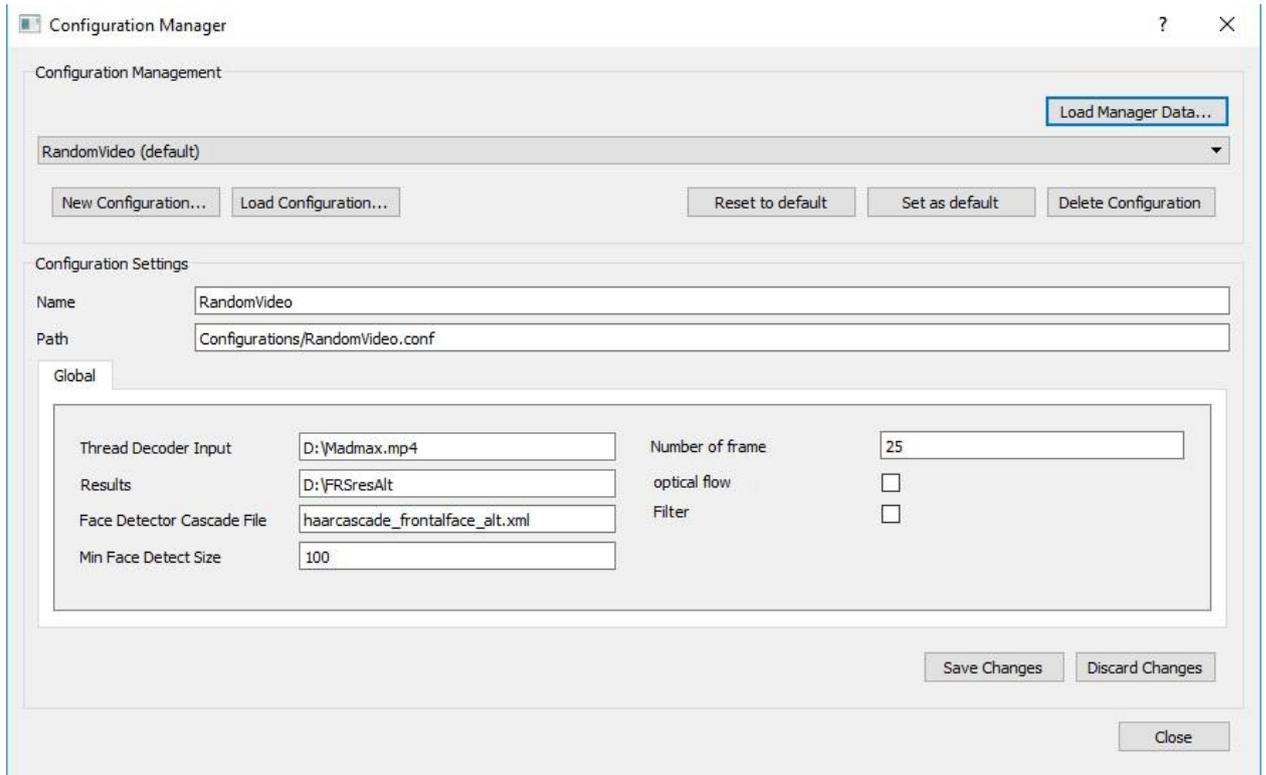


Figure 29. Configuration Manager Module

To quickly change the settings, use the Configuration Manager menu. The menu has two sections: Configuration Management and Configuration Settings. At the top of the menu, the following buttons are available for switching:

- Load Manager Data ... - loads the default parameters from the file FRSSConfigManagerData.conf;
- The drop-down list for selecting the available configurations in the project folder;
- New Configuration - create a new configuration;
- Load Configuration - loads third-party configurations into the project folder;
- Reset to default - setting the current configuration parameters is similar to the configuration with the label "default";

- Set as default - set the configuration to "default". If the project is run again the default configuration will be selected;
- Delete Configuration.

The following parameters are entered at the bottom of the menu:

- Name - the name of the configuration;
- Path - configuration location;
- Thread Decoder Input - the path to the input video sequence;
- Result - the location of the text file, in which the result is entered in the form of numerical indicators;
- Face Detector Cascade File - path to the file with a set of Haar attributes;
- Min Face Detect Size - the minimum possible size of the found face, faces less than this value - are ignored;
- Number of Frame - a value that indicates the frame interval between which the detector is activated. If empty - then the detector is off, if 0 - then it works on every frame;
- Optical flow - when enabled, activates the methods for calculating the optical flow;
- Filter - when activated, activates the frame filter;
- Save Changes;
- Discard Changes - undo changes;
- Close - close the configuration manager window.