

Национальный исследовательский
Томский политехнический университет

На правах рукописи

Небаба Степан Геннадьевич

**Технология подготовки изображений лиц к распознаванию личности
в видеопотоке в режиме реального времени
на основе компенсации ракурса и трекинга лиц**

05.13.17 – Теоретические основы информатики

**Диссертация на соискание ученой степени
кандидата технических наук**

Научный руководитель

Захарова Алёна Александровна
доктор технических наук

Томск – 2017

Оглавление

Оглавление	2
Введение.....	4
Глава 1. Аналитический обзор методов подготовки изображений лиц в видеопотоке к распознаванию	14
1.1. Актуальность исследования, обзор и анализ состояния проблемы	14
1.1.1. Трекинг	14
1.1.2. Сравнение изображений	19
1.1.3. Оценка контрастности	20
1.1.4. Оценка резкости	22
1.1.5. Компенсация освещения.....	27
1.1.6. Контроль ракурса	30
1.2. Анализ.....	46
1.3. Выводы по главе 1.	49
Глава 2. Технология подготовки изображений лиц в видеопотоке к распознаванию.....	50
2.1. Выбор методов, используемых в составе технологии.	50
2.1.1. Трекинг	50
2.1.2. Сравнение изображений	54
2.1.3. Методы оценки контрастности и резкости.....	56
2.1.4. Метод коррекции освещения	57
2.1.5. Метод коррекции ракурса	57
2.2. Выбор средств разработки.	72
2.3. Общий вид алгоритма, описывающего технологию подготовки и обработки изображений.	74
2.4. Программная реализация алгоритмов.	84
2.4. Выводы по главе 2.	87
Глава 3. Тестирование алгоритмов и результаты экспериментов.....	88
3.1. Подготовка экспериментов.	88
3.2. Тестирование программной реализации алгоритмов.	89

3.3. Результаты тестирования комплексной технологии подготовки изображений лиц к распознаванию.	98
3.4. Сравнение результатов тестирования предложенной технологии с существующими системами распознавания	112
3.5. Выводы по главе 3.	114
Глава 4. Апробация и внедрение	116
4.1. Апробация результатов работы.....	116
4.2. Внедрение результатов работы.	116
4.3. Результаты тестового контроля.....	118
4.4. Выводы по главе 4.	120
Результаты и выводы	121
Список литературы	123
Публикации по теме диссертации	136
Приложения	139
Приложение 1. Фрагменты исходного кода основных модулей программного комплекса.....	139
Приложение 2. Справка о внедрении результатов научно-исследовательской работы в практику.....	152
Приложение 3. Акт о внедрении научных и практических результатов	153
Приложение 4. Свидетельство о государственной регистрации программы для ЭВМ.....	154

Введение

Обоснование выбора темы исследования:

Работы по решению задачи идентификации личности на основе анализа изображений лица ведутся с самых ранних этапов развития компьютерного зрения [1]. Задача распознавания лиц является одной из первых практических задач, которая послужила стимулом для развития теории распознавания объектов. Данное направление появилось в начале 1980-х годов, но его развитие началось в 1990-х годах в процессе создания информационно-поисковых систем распознавания для идентификации личности [2, 3]. В последнее время в различных сферах деятельности возрастает потребность быстрой и правильной некооперативной идентификации личности человека в видеопотоке в режиме реального времени [4,5,6,7].

В общем случае задача идентификации личности в видеопотоке может быть разделена на несколько этапов:

- 1). Поиск области лица на изображении.
- 2). Слежение за положением обнаруженного лица на последовательности кадров (трекинг).
- 3). Предварительная обработка изображения лица с целью компенсации условий съемки.
- 4). Сравнение изображения лица с изображениями в базе изображений.

Задача поиска лиц на изображении на сегодняшний день успешно решена и применяется во многих технических устройствах [8]. Например, в фототехнике поиск лиц используется для автоматического выбора точки фокусировки [9].

Трекинг является необязательным этапом, но он позволяет соотносить изображения лиц людей на разных кадрах и делать распознавание по наилучшему изображению из последовательности кадров, косвенно повышая точность идентификации. Задача трекинга широко освещена в литературе и имеет множество вариантов применения в задачах компьютерного зрения помимо слежения за лицами людей [10,11,12,13].

Задачи обработки изображений и автоматизированного распознавания (идентификации) лиц являются более сложными, и на данный момент существующие алгоритмы приближаются к точности распознавания лиц человеком только в строго заданных условиях, либо требуют вычислительные ресурсы, недоступные при обработке видеопотока в режиме реального времени.

Развитие теории и практики распознавания объектов во многом было обусловлено достижениями зарубежных ученых: Дж. Филипса (NIST, США) и Г. Векслера (университет Джорджа Мейсона), работающих в этой области уже более 15 лет, а также Такэо Канадэ, М. Турка, П. Виолы, М. Джонса, Б. Лукаса, Дж. Сяо, С. Бэйкера, Д. Команицу, Г. Брадски и других [14,15,16,17]. Разработка методов и алгоритмов распознавания и обработки объектов на изображениях ведется не только зарубежными, но и российскими учеными. Весомый вклад в область анализа изображений внесли Конушин А.С., Вапник В.Н., Журавлев Ю.И., Сойфер В.А., Загоруйко Н.Г., Местецкий Л.М., Алпатов Б.А., Горелик А.Л., Гуревич И.Б., Сергеев В.В., Васин Ю.Г., Корилов А.М., Катаев М.Ю., Князь В.А., Галактионов В.А. и другие [18,19,20,21,22]. В решение проблемы распознавания лиц ценный вклад внесли Желтов С.Ю., Визильтер Ю.В., Крылов А.С., Вежневцев В.П., Шерстобитов А.И., Пахирка А.И., Калайда В.Т., Спицын В.Г. и другие [23,24,25,26,27,28].

В последние годы был предложен ряд различных подходов к обработке, локализации и распознаванию объектов, таких как метод главных компонент (Principal Component Analysis, PCA), нейронные сети, эволюционные алгоритмы, алгоритм AdaBoost (Йоав Фройнд, Роберт Шапире), метод опорных векторов и другие [28,29,30]. Точность, надежность и скорость этих подходов серьезно зависят от шумов на видеопоследовательностях и различных условий съемки. Методы, применяемые для решения задачи распознавания лиц в режиме реального времени, должны обеспечивать приемлемую точность распознавания и высокую скорость обработки видеопоследовательностей.

Число ошибок идентификации личности у всех современных систем распознавания лиц в режиме реального времени имеет явную зависимость от следующих факторов:

- качество изображения лица, находящегося как в базе данных, так и регистрируемого камерами видеонаблюдения;
- перекрытие лица посторонними объектами и другие помехи на изображении;
- ракурс съемки идентифицируемого лица;
- условия освещения.

Изображения анализируемых лиц должны обладать минимально необходимым качеством, чаще всего требования к изображению включают такие пункты, как:

- отсутствие эффекта смазанного изображения, вызванного движением идентифицируемого объекта;
- изображение анализируемых лиц не должно быть меньше минимально необходимого размера (определяется по расстоянию между зрачками).

Для решения этих проблем существует два пути: расширение базы эталонных изображений изображениями с различными условиями съемки (что не всегда возможно сделать), а также моделирование условий съемки на эталонных изображениях при их сравнении с анализируемыми. Примером второго подхода является использование трехмерных (3D) моделей лиц и сравнение отдельно рельефа и текстуры получаемых 3D объектов [31,32,33]. Данный подход предоставляет широкие возможности по моделированию ракурса и освещения, но требует создания 3D модели каждого лица в базе данных, что является ресурсоемкой и нетривиальной задачей.

Во многих сферах деятельности требуется некооперативное распознавание личности человека, например поиск подозрительного человека по короткой видеозаписи плохого качества, на которой он не смотрит прямо в объектив камеры, присутствуют шумы, боковая засветка. Это затрудняет применение как

классических двумерных (2D) методов распознавания, так и методов, использующих 3D модели.

Существующие быстрые 2D методы и алгоритмы достигают хорошей точности только при соблюдении жестких требований по ракурсу изображений лица и условиям съемки, что обусловлено механизмами кластеризации и машинного обучения [34,35].

Если влияние условий съемки, таких как освещение, яркость и контрастность изображения, в большинстве случаев можно компенсировать с помощью существующих методов и алгоритмов обработки изображений, то задача компенсации ракурса решается неэффективно, и часто решения сводятся к использованию оборудования, позволяющего получать 3D модели лиц для дальнейшего использования в распознавании [36,37]. Методы, рассчитанные на построение 3D моделей из видеопотока, обычно не рассматриваются для использования в системах реального времени, так как для создания 3D модели высокого качества в режиме реального времени требуются значительные вычислительные ресурсы [38,39,40].

Условия освещения должны позволять регистрировать изображения лиц, на которых должны быть различимы основные черты лица (достаточный контраст). Для всех современных систем идентификации важно наличие прямого или близкого к нему угла падения лучей света на фронтальную плоскость лица, а также наличие рассеянного источника света (условия приближенные к условиям получения фотографий для официальных документов). Крайне негативное влияние на качество идентификации оказывает наличие боковой засветки лица [41].

Учитывая вышеперечисленные проблемы методов идентификации личности по растровому изображению лица в режиме реального времени, можно сделать вывод о необходимости исследования существующих методов предварительной обработки поступающих изображений как единой технологии. Такая технология должна содержать в себе некоторую совокупность методов и алгоритмов по предварительной обработке и оценке изображений лиц, подобран-

ную под оптимальное соотношение качества и скорости обработки изображений для конкретной решаемой задачи.

Предварительная обработка может проводиться по нескольким критериям, таким как контрастность, резкость, освещение, ракурс, похожесть изображений [42,43]. Технологию предварительной обработки можно представить в виде комплексного алгоритма для задачи обработки видеопотока, в том числе и в режиме реального времени, который должен учитывать все критерии и обладать достаточным быстродействием.

Проблема предварительной обработки изображений чаще всего рассматривается как набор частных задач по улучшению качества, компенсации освещения и ракурса съемки, но не как сложная система взаимозависимых алгоритмов.

Таким образом, необходимость оптимизации всех шагов обработки и анализа изображений лиц под конкретную задачу распознавания, будь то фотографии или видеопоток, создают предпосылки для создания комплексной системы алгоритмов.

Цель работы – повышение точности и скорости распознавания личности в видеопотоке в режиме реального времени посредством технологии подготовки изображений, обеспечивающей компенсацию ракурса и трекинг лиц.

Такая технология должна оперировать сложными исходными данными – изображениями лиц, которые могут обладать низкой контрастностью, резкостью, иметь ракурс, отличный от фронтального, а также иметь производительность, достаточную для ее применения в комплексе с алгоритмом распознавания личности по изображению лица в видеопотоке в режиме реального времени.

Для достижения этих требований необходимо решить следующие **задачи**:

1. Провести обзор существующих методов анализа изображений: оценки схожести, контрастности и резкости, нормирование освещения, и выбрать оптимальные.

2. Исследовать возможность применения 3D моделей лица человека в распознавании лиц по изображению лица в режиме реального времени и разработать алгоритм синтеза моделей лиц, подходящий для этих условий.
3. Разработать новый алгоритм, снижающий влияние ракурса на точность распознавания личности по изображению лица путем корректировки ракурса с помощью синтезированных моделей и применимый для задач реального времени.
4. Предложить и реализовать новый алгоритм трекинга лица в видеопотоке, обеспечивающий эффективное слежение за лицом и обладающий быстрой реакцией, допускающим его применение в системах распознавания личности в режиме реального времени.
5. Разработать новую комплексную технологию автоматического анализа и обработки изображений лиц с целью их подготовки к распознаванию личности в видеопотоке в режиме реального времени и представить ее в виде комплексного алгоритма.
6. Реализовать разработанные алгоритмы в виде программной системы и провести вычислительные эксперименты с целью анализа их качества и эффективности.

Объектом исследования в данной работе является технология распознавания лиц по изображению лица для задач реального времени.

Предмет исследования – процесс предварительной обработки и анализа изображений лиц, используемых в алгоритмах распознавания, а также влияние этого процесса на точность и скорость получения конечного результата распознавания.

Методы исследований. В работе использованы методы теории алгоритмов, теории моделирования, методы планирования численных экспериментов, а также методы теории вероятностей и математической статистики для количественной обработки данных экспериментов.

Научная новизна полученных результатов заключается в следующем:

1. Предложена новая комплексная технология подготовки изображений лиц, обеспечивающая функционирование программных систем распознавания личности в видеопотоке в режиме реального времени в автоматическом режиме, отличительными особенностями технологии являются авторские алгоритмы компенсации ракурса и трекинга лиц (пп. 5, 7, 14 паспорта специальности).
2. Предложена оригинальная модификация метода формирования деформируемой модели лица (Deformable face model, DFM), использующая принципы восстановления ландшафтных поверхностей методом интерполяции Шепарда и синтезирующая 3D модель лица человека в автоматическом режиме (пп. 3, 5, 7 паспорта специальности).
3. Разработан новый алгоритм формирования набора изображений лиц из одного фронтального изображения лица, основанный на применении предложенной модификации метода DFM и синтезировании проекций этой модели на плоскость (пп. 5, 7 паспорта специальности).
4. Разработан новый алгоритм трекинга лиц в видеопотоке, комбинирующий методы background subtraction и Виолы-Джонса, повышающий скорость поиска и трекинга лиц по сравнению с алгоритмом, использующим только метод Виолы-Джонса (пп. 5, 7 паспорта специальности).

Теоретическая значимость результатов диссертационного исследования состоит в том, что разработаны, исследованы и апробированы новые методы и алгоритмы для обработки и анализа изображений лиц в видеопотоке с целью их подготовки к распознаванию в режиме реального времени.

Практическая ценность. Разработанные в диссертации алгоритмические и программные средства предназначены для решения задач предварительного анализа и обработки изображений лиц в видеопоток. Они могут найти применение при создании отечественных программных систем верификации личности и поиска людей, например, при создании мобильных систем идентификации личности на удаленных объектах, функционирующих без доступа к сетевой инфраструктуре и серверам анализа данных.

Программная система, использующая трекинг, может использоваться для анализа массовых моделей поведения, таких как исполнение инструкций на предприятиях со строгим технологическим процессом, планов учений либо эвакуаций.

Результаты исследования методов обработки изображений, полученные в ходе данной работы, могут использоваться в процессе обучения студентов специальностей, связанных с компьютерным зрением, трехмерным моделированием и обработкой изображений.

Реализация результатов работы

Все разработанные алгоритмы были реализованы и протестированы в рамках создания программного комплекса предварительной обработки изображений и распознавания лиц в видеопотоке в режиме реального времени в НУЛ 3D моделирования Института Кибернетики Томского политехнического университета.

Степень достоверности результатов проведённых исследований подтверждается результатами тестирования, апробации и внедрения разработанных алгоритмов и технологии подготовки изображений лиц на собственных и общедоступных базах изображений. Полученные результаты согласуются с результатами, полученными другими авторами.

Внедрение работы

На основе анализа и сравнения методов обработки изображений, проведенного в ходе данной работы, разработан модуль учебной дисциплины по основам обработки изображений и распознаванию образов для студентов специальности Б3.В11 «Интеллектуальные и информационные системы», бакалавриат, направления 09.03.03 «Прикладная информатика» кафедры Программной Инженерии ИК ТПУ.

Тестовый стенд в виде программно-аппаратного комплекса, включающий в себя IP-камеру и персональный компьютер с разработанной программной системой, работает в Кибернетическом центре ИК ТПУ. На полученных данных

были реализованы и протестированы функции обработки изображений и распознавания лиц.

Аналогичный программно-аппаратный комплекс прошел апробацию и внедрен в ООО Научно-производственная компания «Техника дела».

Получены акт о внедрении разработанной программной системы и справка об использовании результатов.

Основные положения, выносимые на защиту:

1. Новая комплексная технология подготовки изображений лиц к распознаванию в видеопотоке в режиме реального времени, внедренная в составе программной системы распознавания личности по изображению лица, основанная на методе классификации k Nearest Neighbours, позволяет повысить точность распознавания на 6-9 %.
2. Предложенная оригинальная модификация метода DFM, использующая метод интерполяции Шепарда, позволяет автоматически синтезировать рельеф лица по единственному плоскому изображению.
3. Разработанный новый алгоритм формирования набора изображений лиц из одного изображения для применения в распознавании личности, основанный на предложенной оригинальной модификации метода DFM, позволяет применять методы классификации лиц в условиях ограниченного набора исходных изображений для задач распознавания в режиме реального времени и повышает точность идентификации на 0,5-4%.
4. Предложенный новый алгоритм трекинга лиц, комбинирующий информацию, полученную методами background subtraction и Виолы-Джонса, работающий в режиме реального времени и снижающий вычислительную нагрузку на программную систему распознавания на 20 и более процентов в сравнении с базовым алгоритмом трекинга, в основе которого метод Виолы-Джонса.

Апробация работы. Результаты диссертационной работы обсуждались и докладывались на следующих симпозиумах, семинарах и конференциях: XII Всероссийская научно-практическая конференция студентов, аспирантов и мо-

лодых ученых «Молодежь и современные информационные технологии» (Томск, 2014), 26-ая Международная научная конференция «GraphiCon 2016».

Публикации. Основные результаты диссертационного исследования изложены в 11 печатных работах, в том числе в 5 статьях из списка рецензируемых журналов, рекомендованных ВАК РФ, в 2 статьях в журналах, индексируемых Scopus, в 1 статье в журнале, индексируемом Web of Science.

На алгоритмы, реализующие предложенные в ходе работы методы обработки изображений, было получено свидетельство о государственной регистрации программы для ЭВМ.

Личный вклад автора. Постановка задач исследования по теме диссертации выполнена автором совместно с научным руководителем, д.т.н. А.А. Захаровой. Обзор существующих методов и алгоритмов анализа и обработки изображений выполнен совместно с С.Ю. Андреевым и М.А. Макаровым. Модификация метода DFM разработана совместно с А.А. Захаровой и С.Ю. Андреевым. Алгоритмы формирования базы изображений лиц и трекинга разработаны совместно с А.А. Захаровой. Комплексная технология подготовки изображений лиц к распознаванию, оптимизированная для использования в режиме реального времени, и результаты экспериментов, представленные в диссертации, получены лично автором.

Структура и объём работы. Диссертация содержит введение, 4 главы, заключение, список литературы, содержащий 121 наименование, и 4 приложения. Общий объём диссертации составляет 154 страницы машинописного текста, иллюстрированного 53 рисунками и 19 таблицами.

Глава 1. Аналитический обзор методов подготовки изображений лиц в видеопотоке к распознаванию

1.1. Актуальность исследования, обзор и анализ состояния проблемы

Практически все частные задачи компьютерного зрения, связанные с предварительной обработкой изображений, такие как трекинг, сравнение изображений, оценка контрастности и резкости, компенсация освещения, контроль ракурса, в том или ином виде рассматривались во многих работах, однако их анализ с точки зрения применимости в системах распознавания личности в режиме реального времени практически никем не поднимался. Это обуславливает необходимость проведения аналитического обзора существующих методов по каждому из этих направлений и определения тех областей, в которых целесообразно воспользоваться ранее предложенными наработками, а также тех, в которых требуется пересмотр подхода с точки зрения автоматизации алгоритмов и повышения скорости их работы.

1.1.1. Трекинг

Поиск лиц на изображении

Существует несколько методов решения данной задачи, однако наибольшее распространение получил алгоритм П. Виолы и М. Джонса, который был опубликован в 2001 году [44]. Алгоритм не специализируется на выделении только лиц, он может быть использован для нахождения и других классов объектов, для чего должен быть использован соответствующий классификатор. Тем не менее, наиболее широкое применение он получил именно для решения задачи обнаружения лиц на растровом изображении [45].

Детектирование лиц использует три ключевых метода:

- свертка функций Хаара с исходным изображением;
- машинное обучение AdaBoost;
- каскадный классификатор для эффективного совмещения множественных функций.

Алгоритм Виолы-Джонса использует вейвлеты Хаара, которые представляют собой прямоугольные волны одинаковой длины (один высокий и один низкий интервал). Для решения поставленной задачи используется ядра свертки, некоторые из которых схематично показаны на рис. 1.

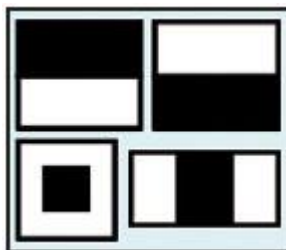


Рис. 1. Примеры вейвлетов Хаара, белому цвету соответствует 1, а черному - -1

Если результат свертки превышает заданный порог, который определяется в процессе обучения классификатора, считается, что на данном участке изображения присутствует функция Хаара. На рис. 2 показаны примеры размещения соответствующих вейвлетов Хаара на изображении лица.

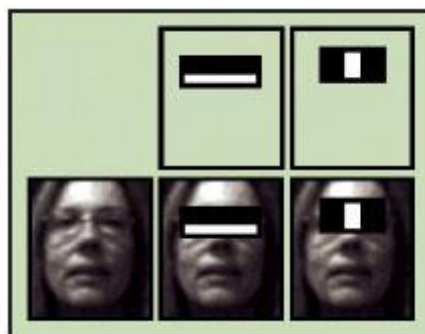


Рис. 2. Примеры размещения соответствия первых двух вейвлетов Хаара на изображении лица

На основе метода машинного обучения AdaBoost, производится подбор вейвлетов Хаара и сопоставления им пороговых величин, сочетание которых характеризует наличие на изображении лица. Размер вейвлетов Хаара задаёт размеры искомых лиц, для поиска всех возможных размеров лиц используется разные масштабы вейвлетов Хаара – пирамида.

В случае использования алгоритма Виолы-Джонса для поставленной задачи на каждом кадре изображения будет найден набор прямоугольных областей, соответствующих найденным лицам.

Слежение за положением лица

Как правило, одно и то же лицо присутствует в поле зрения камеры видеонаблюдения в течение некоторого временного интервала, поэтому его изображение может быть найдено в последовательности кадров. Это позволяет осуществить трекинг искомого лица.

Трекинг (tracking) – это определение местоположения движущегося объекта в видеопотоке. Алгоритм анализирует кадры видеопотока и выдает положение движущихся целевых объектов относительно кадра, позволяя составить последовательность изображений конкретного объекта, которая называется треком (track). Пример трекинга движущихся объектов представлен на рис. 3.

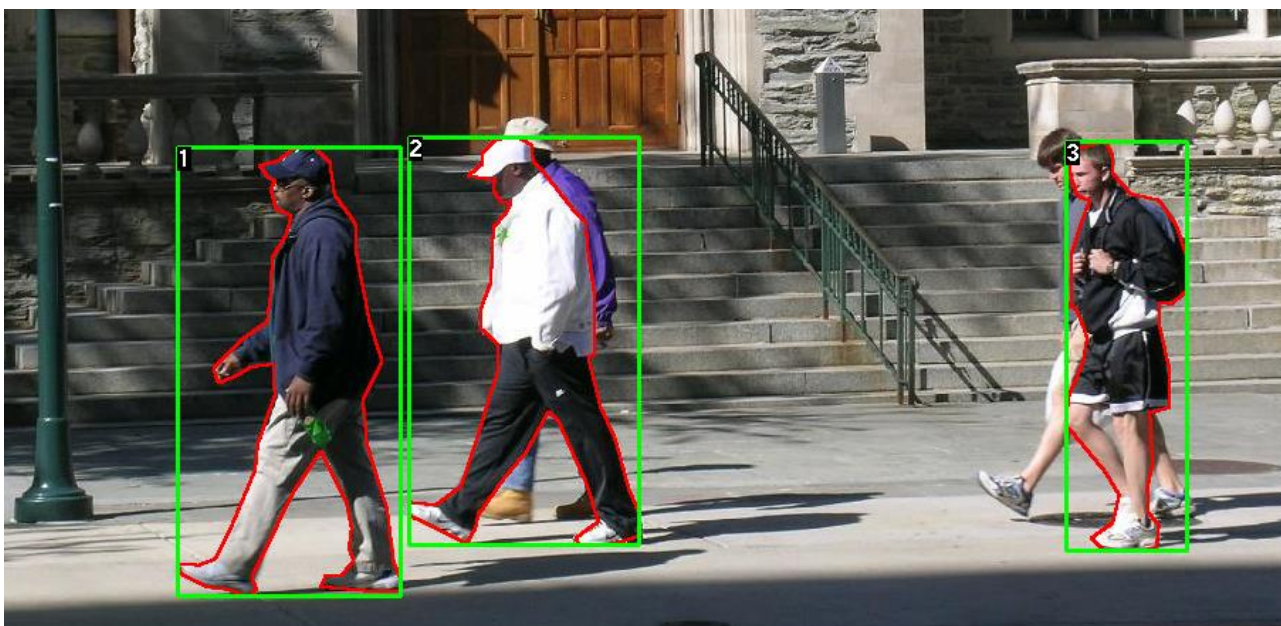


Рис. 3. Пример трекинга движущихся объектов

Движение объекта на видео классифицируется на регулярное (движение объекта по непрерывной траектории) и нерегулярное (разрывы траекторий вследствие заслонения объекта слежения другими объектами, скачка объекта и

других причин). В случае регулярного движения объекта трекер рассматривается как динамическая система [46].

Алгоритмы слежения (трекинга), оценивающие положения одного или нескольких целевых объектов (объектов слежения) на кадрах видеопоследовательности, входят в многочисленные приложения компьютерного зрения в робототехнике и видеонаблюдении [47,48,49]. Нахождение и отслеживание движущихся объектов в видеопотоке является важной задачей для таких приложений, как охранные системы, системы анализа потока машин на автомагистрали, smart room и т.д. [50,51].

Характеристики целевого объекта в реальном видео с течением времени, как правило, изменяются в силу естественных условий видеосъемки, когда, например, во время съемки изменяется освещенность объекта, изменяются относительные размеры объекта в последовательности кадров при удалении/приближении объекта относительно камеры, объект слежения заслоняется другими объектами и т.п.

Часто первым шагом к решению этих задач является выделение переднего плана. Например, в работах [52,53,54] рассматривают методы вычитания фона (background subtraction), применяемые для решения этой задачи. Методы вычитания фона - это методы, которые строят модель заднего плана сцены и на каждом кадре производят попиксельное сравнение текущего значения в каждом пикселе с моделью и в соответствии с результатами сравнения принимают решение, принадлежит пиксель к заднему плану или к переднему. Основные проблемы, с которыми сталкиваются подобные методы – это изменяющийся задний план, изменение освещения, движущиеся тени и шум камеры (рис. 4).

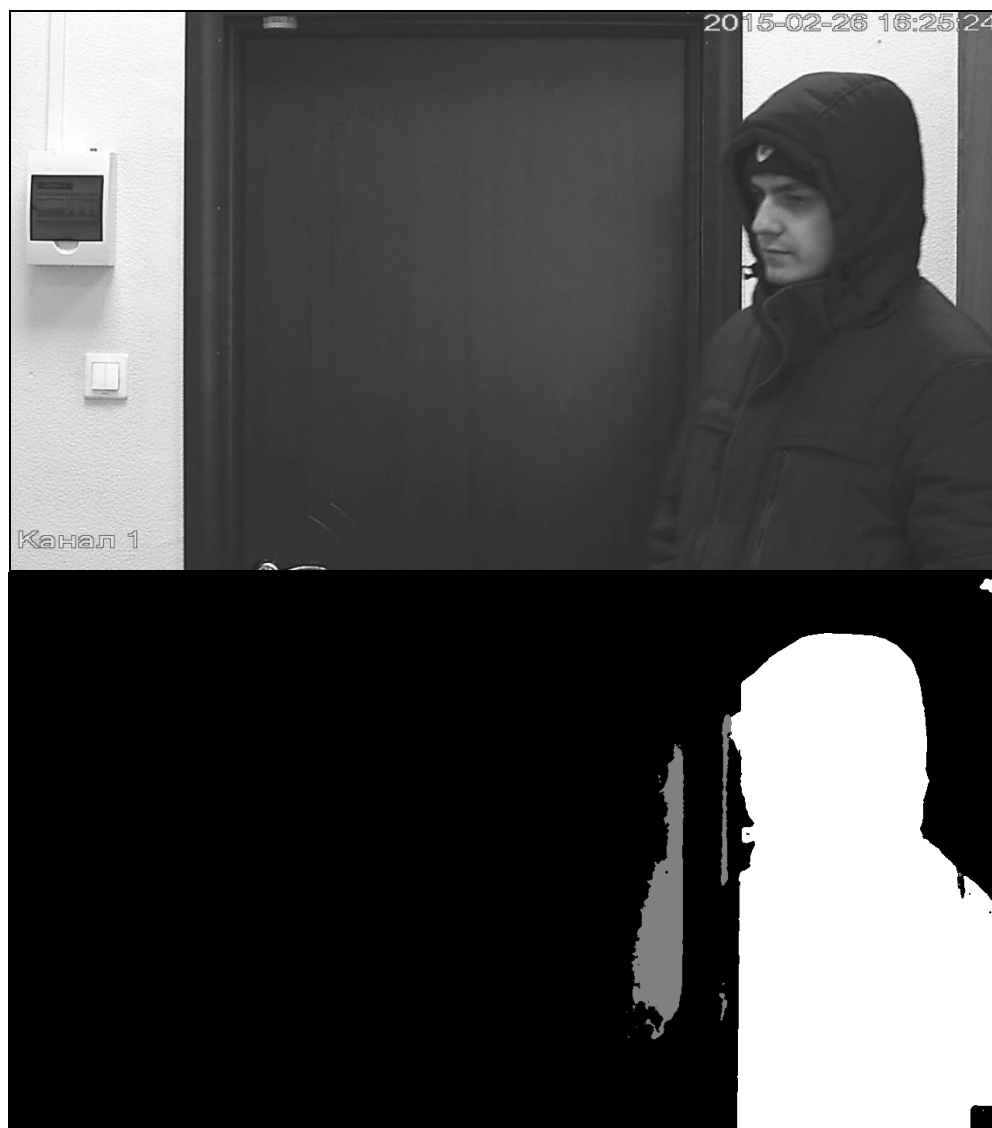


Рис. 4. Пример трекинга методом вычитания фона с подавлением теней

Также существуют алгоритмы, основанные на вероятностных моделях, которые призваны решать часть этих проблем, в частности проблему выделения теней движущихся объектов.

В решаемой задаче слежения за лицом человека большая часть описанных проблем не является актуальной в силу того, что поиск объекта в данном случае осуществляется с помощью хорошо изученного и проверенного алгоритма Виолы-Джонса, для которого разработано множество алгоритмов качественного трекинга [55,56].

Так как трекинг используется исключительно с целью снизить объем вычислений за счёт объединения последовательностей изображений лиц в видео-

потоке (либо с целью получить набор из нескольких изображений одного лица), то к нему не следует выдвигать требования в абсолютной точности и применимости к множеству разных объектов. В этом случае имеет смысл использовать алгоритм, оперирующий с информацией о положении изображений лиц в кадре с добавочным предварительным анализом соседних кадров и фона изображения, что позволит снизить вычислительную нагрузку на систему распознавания детектором лиц и соотносить большую часть изображений лиц на двух соседних кадрах.

1.1.2. Сравнение изображений

В большей части систем видеонаблюдения используется стандартная частота поступления кадров равная двадцати пяти кадрам в секунду. Очевидно, что изображения лица одного человека на двух соседних кадрах практически не должны отличаться друг от друга. Биометрический анализ одинаковых изображений не оправдан с точки зрения загрузки вычислительных ресурсов. В связи с этим целесообразно проводить отбор кадров, на которых изображения одного лица существенно отличаются друг от друга.

Существует несколько основных методов сравнения изображений.

1. Применение хэш-функций к уменьшенной копии изображения и их сравнение. Данный метод обладает хорошим быстродействием, но дает очень грубую оценку, которая напрямую зависит от уменьшения изображения, поэтому может применяться только для поиска отдаленного визуального сходства объектов, например, поиску похожих картинок в коллекции. Метод относится к поиску изображений по содержанию (англ. Content-based image retrieval (CBIR)) [57,58] – разделу компьютерного зрения, решающему задачу поиска изображений, которые имеют требуемое содержание, в большом наборе цифровых изображений.
2. Вычисление коэффициента корреляции по одной из формул. Этот метод достаточно чувствителен к небольшим изменениям изображения,

однако требователен к вычислительным ресурсам. Его применение ограничено специфическими областями работы с изображениями [59,60].

3. Сравнение изображений через построение и анализ SURF-дескрипторов, которые описывают некоторые небольшие области изображения. Исследования показывают, что данный метод показывает наилучшие результаты при сравнении изображений объектов, снятых под разными углами и в разных условиях освещения, однако он требует в несколько раз больше вычислительных ресурсов, чем сравнительно трудоемкий метод вычисления коэффициента корреляции [61]. Существует модификация данного метода (BRISK), которая обладает высоким быстродействием [62], однако данный метод сложно применить к поиску малозаметных отличий в кадрах.

1.1.3. Оценка контрастности

Перед тем, как улучшать изображение, необходимо дать оценку его качеству. Человек, бросив один взгляд на изображение, может сказать яркое оно или тёмное, контрастное или нет, чёткое или размытое и т.д. Алгоритмы же работают детально, анализируя изображение попиксельно или небольшими группами пикселей. Поэтому, на основании работы алгоритма, тяжело дать общую оценку качеству изображения.

Контраст (фр. *contraste*) – в оптике (сенситометрии и фотометрии) разница в характеристиках различных участков изображения, способность фотографического материала или оптической системы воспроизводить эту разницу, а также характеристика чувствительности глаза (зрительной системы) относительно яркости и цвета.

Контрастность (также, в различных контекстах употребляется и само слово контраст и коэффициент контраста) – степень контраста, чаще всего выражается безразмерной величиной, отношением или логарифмом отношений.

Мерой контрастности является величина $k = \frac{B_b - B_s}{B_s}$, где B_b – яркость фона, B_s – яркость предмета или изображения.

Контрастность изображения является показателем его информативности, так как этот показатель учитывает освещённость и различимость объектов на изображении. Этот параметр недостаточен, чтобы можно было признать изображение качественным, так как боковая засветка объектов дает высокую оценку контраста, но затрудняет выделение деталей изображения.

Тем не менее, в случае, если изображение лица удастся нормализовать, контрастность становится более полезной оценкой.

Контрастность изображения бывает яркостная и тоновая.

Яркостная контрастность представляет собой разницу между физической или видимой яркостью отдельных участков изображения. Вообще говоря, вычисление физической или видимой яркости можно рассматривать как конвертацию цветного изображения в ахроматические цвета. Поэтому яркостная контрастность – это сравнение двух участков изображения, приведенных к ахроматическим цветам.

Если проанализировать RGB-гистограммы, то можно сделать вывод, что у контрастного изображения количество тёмных и светлых пикселей должно быть приблизительно одинаковым, разница в их яркости – значительна, а основное место сосредоточения пикселей – возле границ диапазона.

Тоновая контрастность учитывает не только яркостные переходы, но и переходы цвета. Так как при анализе лиц используются чёрно-белые изображения (в градациях серого), то использование тоновой контрастности не имеет смысла.

Хороший критерий оценки яркостной контрастности – дисперсия яркости пикселей изображения [63]:

$$\sigma^2 = \frac{1}{N} \sum_{p=1}^N (Y_p - Y)^2, \quad (1)$$

где Y – среднее значение яркостной контрастности всего изображения, Y_p – значение яркостной контрастности в точке p , N – общее число точек изображения.

Оценка контрастности (1) может быть нормирована путём вычисления отношения среднеквадратического отклонения к максимально возможному значению яркости:

$$C = \frac{2\sigma}{Y_{\max}}. \quad (2)$$

Значение C изменяется в диапазоне $[0;1]$. Значение 0 соответствует одно-тонному изображению, значение 1 — максимально контрастному. Оптимальное значение контрастности (2) зависит от типа объекта, представленного на изображении. На рис. 5 приведены примеры изображений и их оценка контрастности.

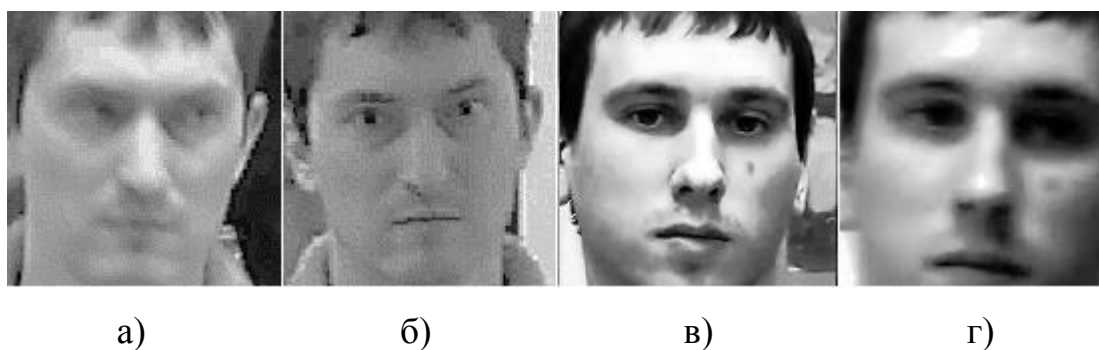


Рис. 5. Примеры оценки контрастности изображений а) $C = 0,42$; б) $C = 0,28$; в) $C = 0,57$; г) $C = 0,52$.

Как можно видеть на рис. 5, контрастность изображения не может являться точным показателем качества изображения, например, изображения а) и г) размытые и менее информативные, но при этом имеют более высокую оценку контрастности, чем изображение б), из-за большого количества точек максимальной и минимальной яркости.

1.1.4. Оценка резкости

Резкость чаще всего трактуется как качество воспроизведения на фото достаточно крупных фрагментов. В этом, условно, ее отличие от четкости изображения, характеризующей различимость мелких деталей.

Повышение резкости – один из самых распространенных способов улучшения фотографий. Обычно всегда требуется в той или иной степени усиливать резкость первоначального изображения.

Объекты на изображении отличаются друг от друга по яркости или оптической плотности, что является причиной их различимости. Граница же между объектами всегда имеет ту или иную степень размытости.

Ширина размытости теоретически резкой границы между светлыми (яркими) и темными участками (объектами) изображения характеризует резкость изображения. Причем это характерно для любых изображений, как цветных, так и черно-белых.

Четкость изображения – это параметр, который характеризует качество полученного изображения и показывает, насколько отчетливо различимы мелкие объекты на изображении и определяется минимальным расстоянием между ними, так, чтобы эти объекты воспринимались как отдельные.

Другими словами, четкость показывает полноту воспроизведения мелких объектов на изображении и насколько у них резкие контуры.

Таким образом, четкость характеризует степень размытия границ вокруг объектов изображения.

Изображение высокой четкости имеет резкие, четко видимые границы.

Резкость может выступать еще одним критерием информативности изображения, она в общем случае может быть определена наличием контуров контрастных переходов. Однозначного метода для определения резкости не существует, но есть несколько алгоритмов, которые позволяют вычислить оценку резкости из различных характеристик изображения.

Главное различие всех предлагаемых методов состоит в формулах, которые определяют количественную оценку контрастности изображения. В целом все методы можно разделить на 3 группы:

- Методы, вычисляющие среднее значение дифференциальных операторов изображения;
- Методы, вычисляющие эксцесс амплитудного спектра изображения;

- Методы, вычисляющие отношение значимых и незначимых точек амплитудного спектра изображения;

Первая группа методов вычисляет разностную характеристику напрямую из исходного изображения, например с помощью оператора Лапласа или оператора Робертса, вторая и третья группы вычисляют частотные характеристики изображения (преобразование Фурье, либо Wavelet-преобразование), и вычисляет количество контрастных переходов из спектра изображения.

Идея метода, использующего дифференциальные операторы, заключается в вычислении среднего значения яркости от преобразованного оператором изображения. Такой метод чувствителен к качеству изображения, а также к характеру изображенных объектов, то есть он может дать высокую оценку резкости для плохого изображения со сложным объектом, и меньшую – для хорошего изображения с малым числом контрастных переходов.

Оператор Лапласа – дифференциальный оператор, сумма вторых частных производных по координатам изображения, эквивалентен операциям градиента и дивергенции:

$$Laplace(f) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}.$$

Оператор Робертса – свертка изображения с ядрами

$$\begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} \text{ и } \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}.$$

Оператор Робертса по сравнению с оператором Лапласа требует меньше вычислений, более точно выделяет тонкие контуры, но очень чувствителен к шуму (рис. 6).



а)

б)

в)

Рис. 6. а) исходное изображение; б) выделение контуров оператором Лапласа; в) выделение контуров оператором Робертса

В работе [64] рассмотрен метод оценки резкости на основе преобразования Фурье, суть которого заключается в вычислении эксцесса изображения в частотной области:

$$\gamma_2 = \frac{\mu_4}{\mu_2^2} = \frac{\mu_4}{\sigma^4}, \quad (3)$$

где μ_2 – второй центральный момент изображения, μ_4 – четвертый центральный момент, σ – стандартное отклонение.

В работе [65] также используется преобразование изображения в частотную область, но в качестве критерия оценки резкости предлагается отношение

$$FM = \frac{T_H}{M \cdot N}, \quad (4)$$

где T_H – число значимых точек амплитудного спектра, $M \cdot N$ – размер изображения амплитудного спектра. Значимые точки определяются как точки, значение в которых превышает некоторую долю максимума амплитуды спектра.

Исследование методов показало, что оценка резкости с использованием оператора Лапласа даёт заметный разброс результатов при схожих условиях освещения и ракурса, что связано отсутствием нормирования результата и способом выделения области лица. Оценка (3) показывает более стабильные результаты, но практически не реагирует на фактическое снижение резкости, полученное размытием по Гауссу. Оценка (4) наиболее точно соответствует изна-

чальной задаче выделения поиска резких изображений, она корректно реагирует на изменения резкости при движении и отсутствие фокуса, и помимо этого имеет строгую нормировку, позволяющую по заданному уровню отсеивать изображения неудовлетворительного качества.

На рис. 7 приведены примеры изображений, их амплитудных спектров и полученных оценок по формулам (3) и (4).

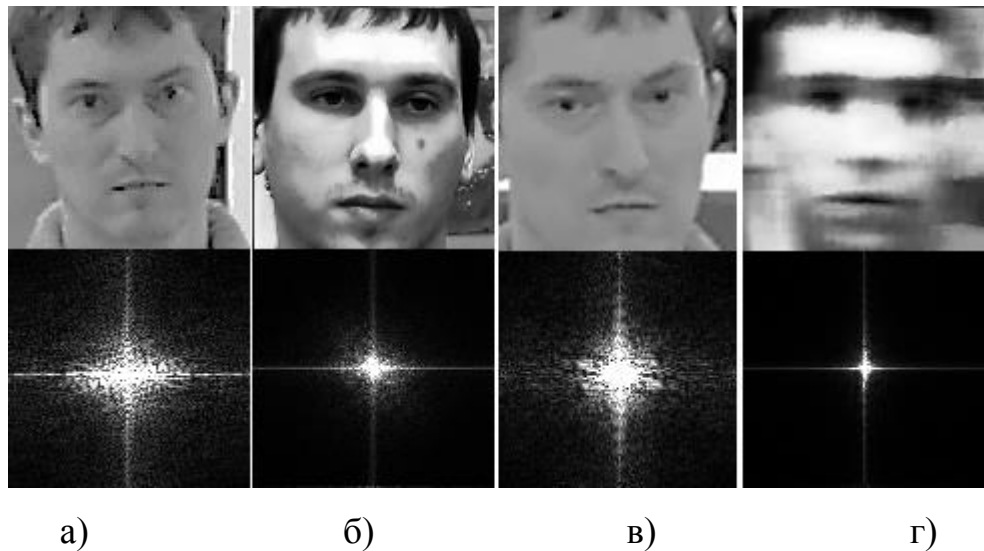


Рис. 7. Примеры оценки резкости изображений:

а) $\gamma_2=0.0035$, $FM=0.1301$; б) $\gamma_2=0.0025$, $FM=0.0388$;

в) $\gamma_2=0.004$, $FM=0.1203$; г) $\gamma_2=0.0062$, $FM=0.0111$.

Несмотря на достоинства оценки (4), она отчасти подвержена тем же проблемам, что и оценка с использованием Лапласиана. Основная причина этого в том, что резкость изображения оценивается количественно, и потому связана с общим числом контрастных переходов, а это число зависит от объекта на изображении. Исходя из этого, можно говорить только об относительной резкости изображения, которая оценивается для одного объекта в треке (серии изображений). При этом можно предположить, что метод даст более точную оценку, если при оценке используются изображения тех участков лица, которые в подавляющем большинстве случаев имеют сходные показатели резкости для всех людей, например глаза, нос, губы.

1.1.5. Компенсация освещения

В случае задачи подготовки изображения к распознаванию лиц проблема разных условий съёмки может быть решена при помощи объединения трекинга лица в видеопотоке и вычисления оценок для серии изображений одного и того же лица, с последующим выделением кадра с наилучшими значениями. Повысить применимость методов (1)-(4) также может компенсация изображения лица по условиям освещения и ракурса.

Для нормирования исходного изображения по условиям освещения существует несколько методов, например DoG-фильтрация [66] и частотная фильтрация полосовым фильтром [67], убирающим низкие и высокие частоты. Оба метода позволяют достаточно эффективно выравнивать изображение при дисбалансе яркости по освещению, например в случае, когда присутствует боковая засветка лица.

DoG обнаруживает края путем применения размытия изображения по Гауссу (Gaussian blur) с указанным значением стандартного отклонения. Результирующее изображение является размытой версией исходного изображения. Далее применяется другое размытие с меньшим значением стандартного отклонения, что дает более резкое изображение, чем в предыдущем случае. Финальное изображение вычисляется путем замены каждого пикселя разностью между двумя размытыми изображениями.

Для нормализации яркости изображения с этим методом часто используется гамма-коррекция.

Частотный полосовой фильтр при нормализации освещения подавляет высокочастотные шумы и сглаживает неравномерное освещение изображений, кодируемое в изображении низкими частотами. Пример полосового фильтра в частотной области представлен на рис. 8.

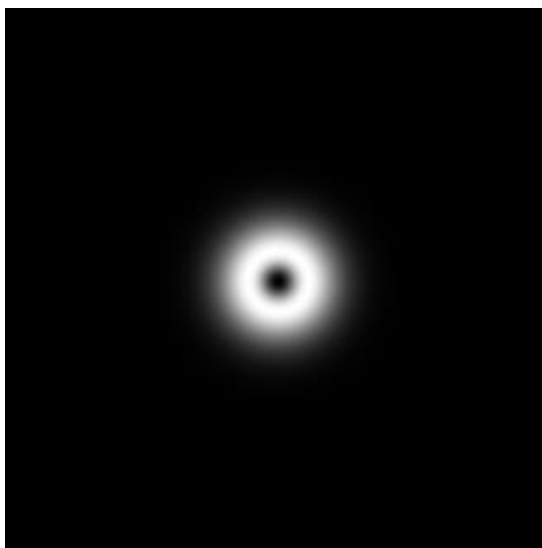


Рис. 8. Пример амплитудного спектра полосового фильтра для изображения

Сравнение этих методов нормирования освещения представлено на рис. 9. Оба метода повышают различимость основных черт лица, но не устраняют полностью влияние неравномерного освещения.



Рис. 9. а) исходное изображение; б) DoG-фильтрация; в) фильтрация полосовым частотным фильтром

Также для повышения контраста изображения применяется адаптивное выравнивание гистограммы с ограничением (CLAHE) [68,69]. Первоначально изображение разбивается на прямоугольные области (блоки). В каждом блоке вычисляется гистограмма распределения яркости. Чтобы избежать перенасыщения монотонных участков изображения, ограничивается наклон функции распределения яркости путем введения ограничения на максимальное значение

частоты яркости. Пусть M – число строк в блоке, N – число столбцов в блоке, α – параметр, задающий ограничение. Предельное значение гистограммы вычисляется в соответствии с формулой:

$$hist_{lim} = hist_{min} + \alpha(MN - hist_{min}),$$

где $hist_{min} = MN/(L+1)$, L – максимальное значение яркости.

После ограничения для каждой области формируется своя функция преобразования яркости на основании выравнивания гистограммы распределения яркости. Для входного изображения вычисляется значение яркостей выходного изображения в соответствии с формулой:

$$LUT(k) = \sum_{j=0}^k p_r(r_j) = \sum_{j=0}^k \frac{n_j}{n},$$

где $LUT(k)$ – значение яркости выходного изображения, соответствующее яркости r_k входного изображения, $k \in [0, L]$; r_j – яркость элемента входного изображения, n_j – число элементов входного изображения со значением r_j , n – общее число элементов изображения; $p_r(r_j)$ – вероятность появления элемента входного изображения с яркостью r_j .

Такое преобразование позволяет равномернее распределить уровни яркости по частоте, в основе преобразования лежит предположение, что в контрастных черно–белых изображениях значения яркости распределены равномерно.

Чтобы разрушить блочную структуру полученного изображения, выходное изображение формируется путем билинейной интерполяции по четырем ближайшим LUT . На рисунке 10 представлен пример обработки изображения с помощью метода CLANE.



Рис. 10. Слева – исходное изображение, справа – изображение, обработанное методом CLANE

1.1.6. Контроль ракурса

Контроль положения лица человека относительно оптической оси камеры является важной проблемой в связи с чувствительностью алгоритмов распознавания к ракурсу. Наиболее известным методом оценки ракурса является POSIT (Pose from Orthography and Scaling with Iterations), он отличается высокой производительностью и быстрой сходимостью.

Алгоритм POSIT вычисляет матрицу поворота и вектор параллельного переноса так, чтобы ошибка между проекцией характерных точек трехмерной модели на плоскость изображения и соответствующими им размеченными точками на изображении была минимальна. Метод POSIT комбинирует два алгоритма. Первый, POS (Pose from Orthography and Scaling), аппроксимирует перспективную проекцию с помощью масштабированной ортографической проекции и вычисляет матрицу поворота и вектор параллельного переноса объекта посредством решения СЛАУ. Второй, POSIT (POS with Iterations), в цикле использует результаты работы POS для нахождения более точной масштабированной ортографической проекции характерных точек, и затем применяет POS к полученным точкам, а не к исходным. В итоге алгоритм POSIT сходится к точному решению за несколько итераций [70].

Проблема использования алгоритма на практике заключается в том, что для его работы необходимо изображение, на котором должны быть отмечены

характерные точки лица, что в общем случае не является автоматизированной процедурой. Существующие методы автоматической расстановки характерных точек допускают серьезные погрешности, но на малых углах поворота позволяют оценить ракурс лица с достаточной точностью [71].

При верной расстановке характерных точек лица алгоритм оценки ракурса лица позволяет получить коэффициенты для задачи коррекции ракурса изображений лиц и вычислить нормированное изображение.

Существует несколько методов автоматической расстановки точек, все они основаны на активных моделях форм (Active Shape Models, ASM).

Активные модели формы - это статистические модели формы объектов, которые могут многократно подвергаться деформациям для наилучшего соответствия объекту, представленному на изображении. Данные модели были предложены Крисом Тейлором и Тимом Кутесом в 1995 году [72]. Модель распределения точек ограничивает форму объекта, поэтому изменение доступно только в пределах, заданных размеченными примерами из обучающей выборки. Форма объекта представлена множеством точек, контролируемым через форму модели. Целью алгоритма ASM является сопоставление модели с новым изображением. В основе алгоритма лежат два чередуемых действия:

- Поиск лучшей позиции для данной точки на изображении вокруг каждой точки;
- Обновление параметров ASM, наилучшим образом соответствующих новым найденным позициям.

Для поиска лучшей позиции в каждой точке можно определять четкие границы на изображении, а можно совместить ожидания для данной точки со статистической моделью. При вычислении лучшей позиции для каждого ориентира точки оригинальный метод предполагает использование расстояния Махаланобиса.

Метод активных моделей форм находит широкое применение в анализе механических узлов, изображений лиц, а также медицинских снимков (как двумерных, так и трехмерных).

Развитие активных моделей форм

Некоторое время математический аппарат активных моделей форм активно развивался. На данный момент выделяются 2 основных подхода к формированию таких моделей: классический, предложенный Кутесом и Тейлором изначально, и подход, основанный на обратной композиции (предложен Бейкером и Мэтьюсом в 2003 году [73]).

У этих двух подходов есть много общего. Активные модели форм используют два типа параметров: связанные с формой объекта (параметры формы), и связанные с текстурой или статистической моделью изображения (параметры внешнего вида). Перед применением модель обучается на множестве изображений с заранее нанесенной разметкой точек. Предварительная разметка изображений чаще всего осуществляется либо полностью вручную, либо в полуавтоматическом режиме, когда сначала с помощью другого более простого алгоритма находится приблизительное расположение меток, а затем оно уточняется экспертом. Каждая из меток имеет свой порядковый номер и определяет характерную точку, которую модель должна будет находить при адаптации к новому изображению. Пример подобной разметки показан на рисунке 11.

На данном изображении лица поставлены 68 меток (особых точек лица), которые образуют форму ASM. Такая форма определяет внешний контур (овал) лица, контуры глаз, рта, бровей и носа. Характер разметки позволяет определить различные параметры лица по его изображению, которые могут использоваться в дальнейшем, в том числе и для последующей обработки с помощью других алгоритмов. К примеру, ASM хорошо совместим с алгоритмами идентификации личности, определения эмоционального состояния субъекта, и даже аудиовизуального распознавания речи.

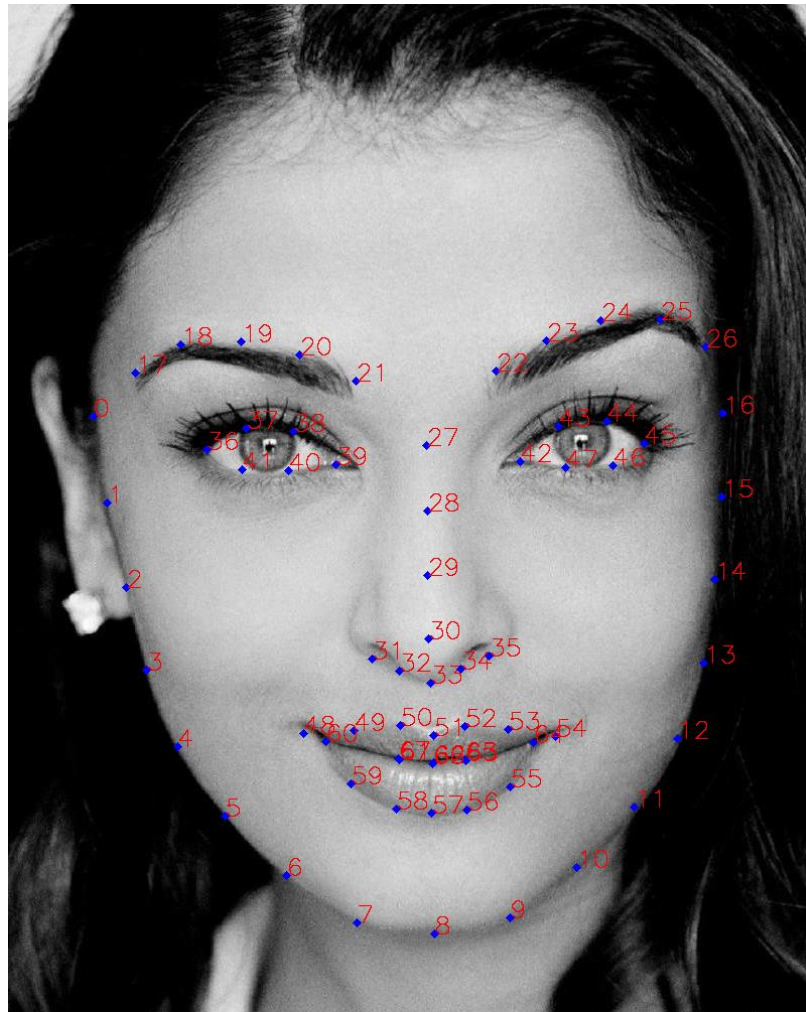


Рис. 11. Пример автоматической разметки лица с помощью алгоритма ASM

Процесс обучения активных моделей форм требует, прежде всего, компенсировать различия в масштабе, смещении и наклоне, поэтому он начинается с нормализации положения всех форм. Основным методом, применяемым для нормализации положения форм - обобщенный Прокрустов анализ [74]. На рисунке 12 показано, как выглядит множество особых точек до и после нормализации (в соответствии с [75]).

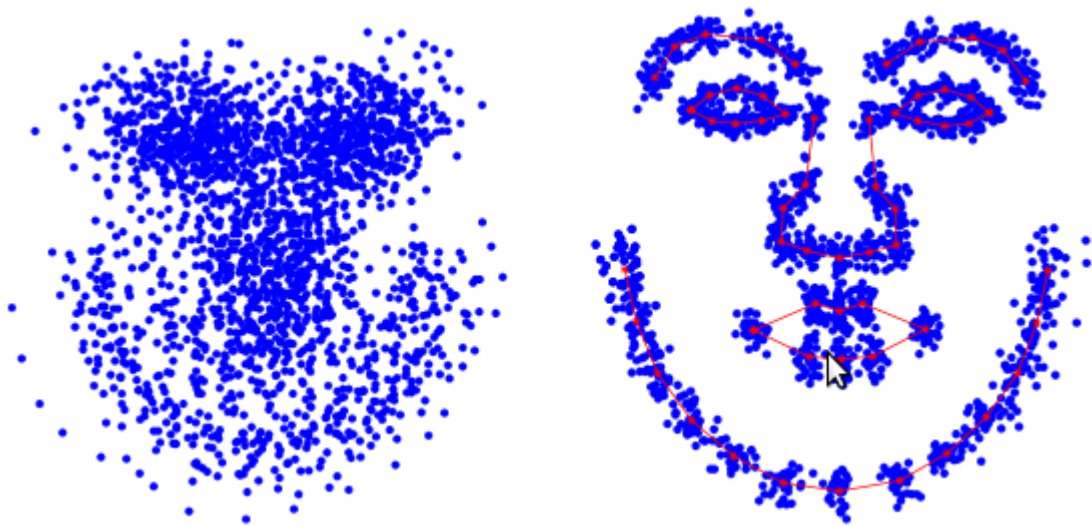


Рис. 12. Множество особых точек до и после нормализации

После нормирования всех форм из точек, составляющих их, проводится формирование матрицы $S = [s_1, s_2, \dots, s_M]$, где $s_M = [x_1^m, \dots, x, y_1^m, \dots, y_N^m]^T$. Выражение для сконструированной формы можно получить путем выделения главных компонент матрицы S :

$$s = s_0 + \Phi_s b_s,$$

где s_0 — базовая форма, то есть форма, усредненная по каждой из реализаций обучающей выборки, Φ_s — матрица главных векторов, b_s — параметры формы. Представленное выше выражение показывает, что форма s может быть выражена в виде суммы базовой формы s_0 с линейной комбинацией содержащихся в матрице Φ собственных форм. Для подгонки формы под реальное изображение можно получать её деформации разного рода путем изменения вектора параметров b_s . На рисунке 13 представлен пример подобной формы [76]. Направления главных компонент показаны красными и синими стрелками.

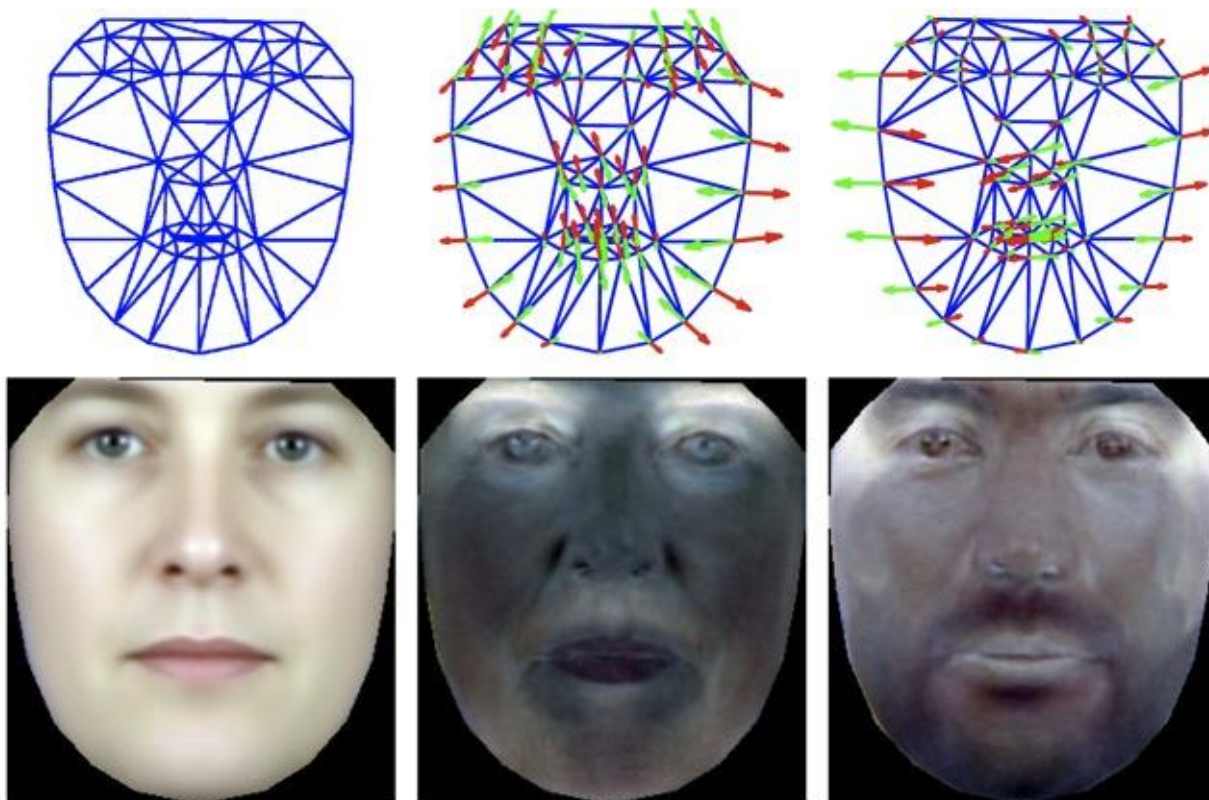


Рис. 13. Получение новой формы лица путем модификации базовой формы

Также все модели активных форм подразделяются на модели с жесткой деформацией и с не жесткой деформацией. К моделям с жесткой деформацией допустимо применять только аффинные преобразования (сдвиг, поворот, масштабирование), а с моделями с не жесткой деформацией допустимо использовать и другие виды деформаций. На практике чаще всего используется некоторая комбинация жестких и не жестких деформаций. При такой комбинационной модели в качестве параметров формы добавляются также параметры расположения (смещение, угол поворота, масштаб или коэффициенты аффинного преобразования).

После вычисления компонентов формы выполняется процедура обучения для компонентов внешнего вида (матрица главных компонент и базовая форма). Процесс обучения делится на три этапа:

- 1). Извлечение из обучающих изображений тех текстур, которые соответствуют базовой форме наилучшим образом. Для этого выполняется триангуляция особых точек базовой формы и формы, которая состоит из особых точек

обучающего изображения. Далее полученные в результате триангуляции регионы обучающего изображения отображаются в соответствующие регионы формируемой текстуры с помощью кусочной интерполяции. На рисунке 14 показан результат такого преобразования на примере одного фронтального изображения.



Рис. 14. Пример извлечения текстуры лица из изображения

2). После формирования всех текстур производится их фотометрическая нормализация, которая предназначена для компенсации различных условий освещения. На данный момент существует множество методов, позволяющих сделать это. Наиболее простой из них — вычитание среднего значения и нормализация дисперсии яркости пикселей.

3). Из текстур формируется матрица в соответствии с правилом, что каждый из столбцов содержит значения пикселей соответствующей текстуры (аналогично матрице S). Для обучения могут быть использованы одноканальные текстуры (градации серого), либо многоканальные (некоторое пространство цветов - RGB или другое). Для случая многоканальных текстур по каждому из каналов формируются отдельные векторы пикселей, а затем выполняется их

конкатенация. Выполнив поиск главных компонент для матрицы текстур, можно получить выражение для синтезированной текстуры:

$$t = t_0 + \Phi_t b_t,$$

где t_0 — базовая текстура, полученная усреднением всех текстур обучающей выборки, Φ_t — матрица собственных текстур, b_t — вектор параметров активно-го внешнего вида. На рисунке 15 показан пример синтезированной текстуры [76].



Рис. 15. Пример синтеза текстуры лица

При практическом использовании метода эффект переобучения модели уменьшают, оставляя в матрицах главных компонент только 95-98% значимых векторов. Этот процент может отличаться для главных компонент внешнего вида и главных компонент формы. Уточнение цифр может быть проведено уже в процессе экспериментальных исследований или с помощью процедуры перекрестной проверки при тестировании модели.

Помимо общей части у разных видов активных моделей форм можно выделить и существенные различия.

Классическая активная модель формы

В классической модели помимо общего обучения необходимо также вычислить вектор комбинированных параметров, который определяется формулой

$$b = \begin{bmatrix} W_s b_s \\ b_t \end{bmatrix} = \begin{bmatrix} W_s \Phi_s^T (s - s_0) \\ \Phi_t^T (t - t_0) \end{bmatrix},$$

где W_s — диагональная матрица весовых значений, позволяющая уравновесить вклад интенсивностей пикселей и расстояний между ними. Для каждого элемента обучающей выборки (пары «форма-текстура») определяется свой вектор b . После этого из полученного набора векторов формируется матрица, для ко-

торой производится поиск ее главных компонент. Затем определяется вектор объединенных параметров текстуры и формы в соответствии с выражением

$$b = \Phi_c c,$$

где Φ_c — матрица главных компонент, соответствующая объединенным параметрам, c — вектор скомбинированных параметров внешнего вида. Из этого выражения могут быть получены новые, описывающие синтезированную форму и текстуру:

$$s = s_0 + \Phi_s W_s^{-1} \Phi_{c,s} c, \quad t = t_0 + \Phi_t \Phi_{c,t} c, \quad \Phi_c = \begin{bmatrix} \Phi_{c,s} \\ \Phi_{c,t} \end{bmatrix}.$$

При использовании на практике посредством удаления шумовых компонент из матрицы Φ_c также снижают эффект переобучения и количество производимых вычислений.

После вычисления параметров формы, внешнего вида и комбинированных параметров необходимо найти матрицу R , называемую матрицей предсказания, которая удовлетворяла бы в смысле минимума среднеквадратичной ошибки линейному уравнению

$$\delta p = R \delta t,$$

где $\delta t = t_{image} - t_{model}$, а δp — возмущение комбинированных параметров внешнего вида и вектора положения. Для решения вышеизложенного уравнения разработаны различные методы.

В общем случае адаптация рассматриваемой активной модели форм к анализируемому изображению происходит по следующему алгоритму:

1. Вычисление всех параметров модели и аффинных преобразований формы на основе начального приближения;
2. Вычисление вектора ошибки δt . С помощью кусочной деформации анализируемого изображения производится извлечение текстуры;
3. Вычисление вектора возмущений $\delta p = R \delta t$;

4. Обновление аффинных преобразований и вектора комбинированных параметров путем суммирования их текущих значений с соответствующими компонентами вектора возмущений;
5. Обновление текстуры и формы;
6. Переход к пункту 2 до тех пор, пока не будет достигнута сходимость.

Общая структура и суть этого алгоритма в любой реализации остаются неизменными, но также для него предложены различные улучшения и модификации.

Применение приведенного выше алгоритма, несмотря на его достаточно высокую эффективность, ограничивается в приложениях реального времени из-за серьезных недостатков: медленный процесс схождения и большой объем вычислений. Новый тип активных моделей форм был предложен в [73] именно для преодоления указанных недостатков.

Активная модель формы обратной композиции

Эффективный в вычислительном плане алгоритм адаптации активной модели внешнего вида был предложен Мэтьюсом и Бейкером. Он зависит только от параметров формы (модель, названная «project-out»). Благодаря этому удалось существенно увеличить быстродействие алгоритма. В основу алгоритма адаптации был положен подход Лукаса-Канаде, а также использование метода Ньютона с целью поиска минимума функции ошибки.

Алгоритм Лукаса-Канаде используется для поиска наилучшего локального соответствия по критерию минимума среднеквадратичной ошибки между реальным изображением и шаблоном. Шаблон при этом подвержен кусочной и/или аффинной деформации, которая задается вектором параметров p и отображает пиксели шаблона на пиксели реального изображения.

Непосредственное нахождение параметров p сводится к задаче нелинейной оптимизации. Для поиска ее решения с помощью линейных методов алгоритм Лукаса-Канаде делает предположение о том, что начальное значение па-

раметров деформации определено. Исходя из этого, алгоритм может итеративно найти приращения параметров δp , на каждой итерации обновляя вектор p .

Активная модель формы обратной композиции для обновления собственных параметров во время процесса адаптации использует аналогичный подход за исключением того, что деформации подвержено анализируемое изображение, а не базовая текстура t_0 .

Активная модель формы обратной композиции на этапе обучения вычисляет особые изображения, называемые «изображения наискорейшего спуска», а также их гессиан. Адаптация модели производится аналогично адаптации классической модели форм, отличие состоит в том, что в данном случае обновляются только параметры формы и, опционально, параметры расположения.

Также стоит отметить, что большое количество возможных вариаций этого метода были предложены еще Мэтьюсом и Бейкером. Каждая из этих модификаций обладает различными свойствами разработанных ими моделей. Современные вариации метода в основном являются комбинацией активных моделей форм с другими алгоритмами [77,78].

В ходе работы были рассмотрены 4 модифицированных алгоритма оригинального ASM:

1. **Классический ASM + метод обратной композиции.** 68 точек, определяет глаза, нос, рот, брови и овал лица. Высокая скорость вычисления. При применении к изображению лица с нефронтальным ракурсом съемки результаты становятся некорректны. Для оценки ракурса подходит плохо. [76]
2. **Stasm-модификация.** 77 точек, определяет глаза, нос, рот, брови и овал лица. Более точный метод, чем классический ASM, но имеет те же проблемы при смене ракурса изображения лица. [79]
3. **SDM facial feature detector.** Использует всего 49 точек, определяет глаза, брови, нос и рот. Данный метод обладает наибольшей точностью и устойчив к изменению ракурса лица на изображении, поэтому хорошо подходит для решения задачи оценки ракурса. [80]

4. **ASM библиотеки dlib.** 68 точек, определяет глаза, брови, нос, рот и овал лица. При использовании стандартного библиотечного детектора лиц этот метод не уступает в точности алгоритму SDM, а при детектировании лица с помощью вейвлетов Хаара обладает хорошим быстродействием при незначительном снижении точности. [81]

В целом можно предположить, что точность и надежность всех методов расстановки точек зависит в большей степени от методики построения обучающей выборки. На рис. 16 представлено сравнение разметки особых точек лица с большим углом поворота относительно объектива камеры для методов STASM, SDM и библиотеки dlib. Из сравнения видно, что метод STASM обучен преимущественно на фронтальных лицах.



Рис. 16. а) STASM-разметка; б) SDM-разметка; в) dlib-разметка

Тестирование алгоритма POSIT (8-точечная модель, уголки глаз + рот) в сочетании с алгоритмом автоматической расстановки особых точек лица dlib показало, что погрешность оценки угла поворота лица не превышает 5 градусов по каждой из осей, что является хорошим результатом для заданных условий.

Методы построения трехмерной модели лица

Для задачи оценки положения лица и автоматической расстановки особых точек лица существуют различные алгоритмы. В то же время, задача коррекции ракурса никогда не рассматривалась в качестве метода, позволяющего улучшить качество работы алгоритмов распознавания по растровому изображению

лица. В первую очередь это связано с проблемой определения трехмерной модели анализируемого лица.

К проблеме получения третьей недостающей координаты можно подойти с разных позиций:

1. Вычисление глубины изображения из особенностей освещения. Данный метод не гарантирует точность и однозначность интерпретации из-за невозможности учесть все возможные особенности освещения в кадре и светоотражающие свойства поверхностей, поэтому не рекомендуется рассматриваться его как метод решения поставленной задачи. Тем не менее, существуют исследования по данному вопросу. [82,83,84,85]
2. Вычисление глубины изображения из характера движения объекта в последовательности кадров. Данный метод подходит только для анализа изображений в видеопотоке, требователен к качеству изображений лица, а также его точность зависит от факторов освещения и свойств детектора лиц. Этот метод используется в ряде исследований, кроме того его вариация со съемкой неподвижного объекта используется в 3D-моделировании. [86,87]
3. Использование заранее подготовленной трехмерной модели лица, которая совмещается с разметкой особых точек на изображении. Этот метод гарантирует точность определения глубины, но требует для работы трехмерные модели каждого обрабатываемого лица, а так же изначальное соответствие модели и изображения, что противоречит поставленной задаче. Кроме этого, получение модели лица для каждого человека требует сложного специального оборудования. Данный метод применяется в основном в сложных системах распознавания с несколькими камерами, вычисляющими трехмерную модель «на лету», в момент детектирования лица. [88,89]

Учитывая выше изложенное, можно заключить, что все рассмотренные методы определения трехмерной модели обладают недостатками и сложно реали-

зуются, что обуславливает необходимость разработки нового метода для получения модели лица.

Задача коррекции ракурса может рассматриваться с нескольких возможных позиций, и исходя из поставленной задачи на основе трехмерной модели лица возможно как восстановление фронтального ракурса повернутого лица на изображении, так и создание набора изображений лица, повернутого под разными углами по отношению к оптической оси камеры (рис. 17).

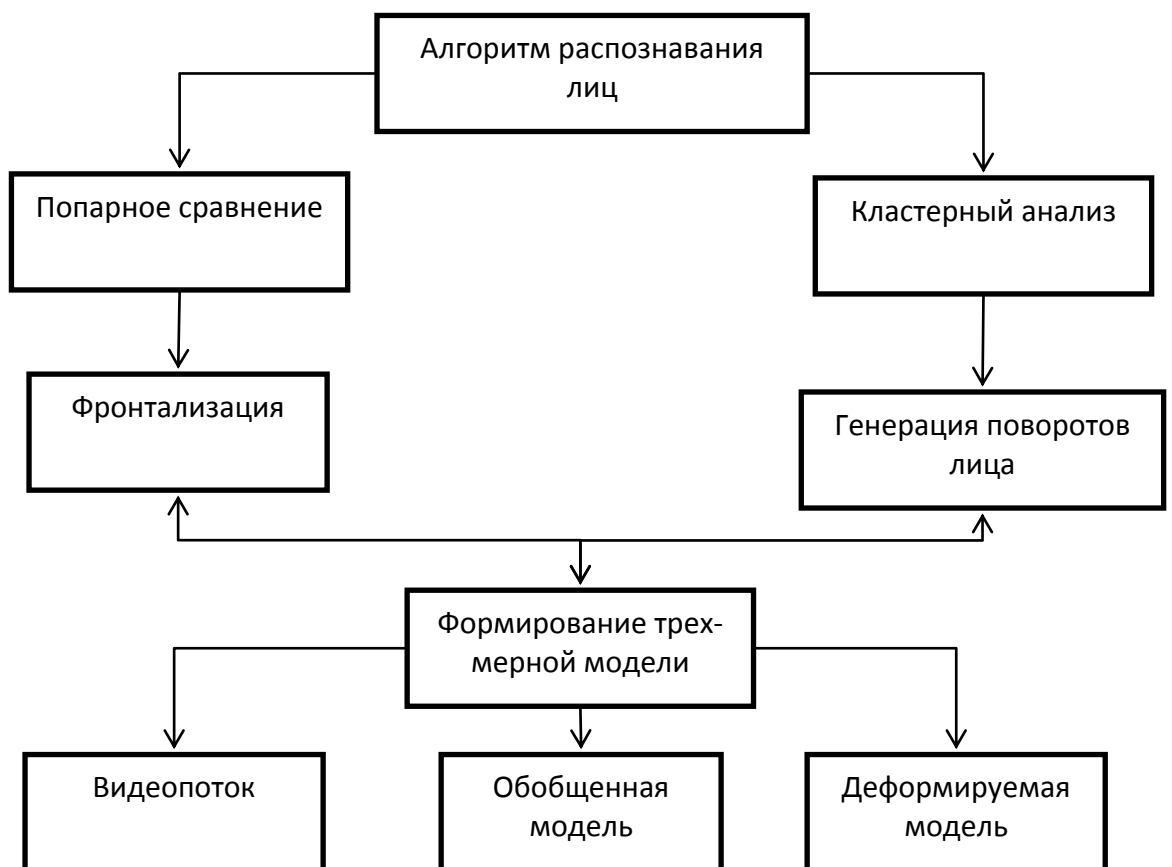


Рис. 17. Варианты решения задачи синхронизации ракурсов для распознавания лиц

В большинстве систем распознавания лиц по двумерному изображению используется принцип построения кластеров из различных изображений одного и того же лица и последующее сравнение изображения лица, полученного из видеопотока, по близости биометрического вектора к одному из кластеров (кластерный анализ) [90]. Также существует вариант распознавания лиц попарным сравнением изображения-эталона с полученными изображениями из ви-

део, но точность такого подхода существенно ниже, так как невозможно полностью устранить влияние освещения, ракурса и качества съемки, а вероятность ошибок повышается [91].

В зависимости от выбора метода распознавания лиц для коррекции ракурса может быть применен как подход приведения всех поступающих для обработки лиц к фронтальному ракурсу (фронтализация) [92], так и подход, использующий генерацию множества изображений лиц под разными углами к оптической оси камеры для формирования кластера, соответствующего конкретному человеку (генерация поворотов лица).

У методов распознавания и применимых к ним методов коррекции ракурса можно выделить достоинства и недостатки:

1). Попарное сравнение (фронтализация):

- + Простой процесс сравнения;
- + Использование минимального количества изображений лиц;
- Сложность процесса оценки ракурса и его низкая точность;
- Сильное влияние условий съемки на результат;
- Дополнительная нагрузка на вычислительные ресурсы в процессе анализа видеопотока.

2). Кластерный анализ (генерация поворотов лица):

- + Оценка угла поворота не влияет на распознавание лиц;
- + Низкая нагрузка на вычислительные ресурсы в процессе анализа видеопотока;
- + Высокая точность распознавания;
- Среднее влияние условий съемки на результат.

В ситуации кластерного анализа метод генерации поворотов лиц может использоваться для получения набора изображений одного лица из единственного изображения с целью построения кластера без ухудшения качества распознавания. Эта задача выглядит более значимой в практическом смысле, чем попытка привести все изображения к единому варианту ракурса. В случае единого ракурса у системы распознавания остается мало информации о рельефе лица.

Эта информация обычно используется для корректировки погрешностей, вносимых освещением или размытием изображения. Кроме этого, метод получения фронтального изображения зависит от точности определения текущего угла поворота [93], и создает дополнительную нагрузку на систему, которой будет необходимо проводить операцию поворота дополнительно к каждому полученному изображению лица.

Исходя из приведенного выше сравнения методов, оптимальным для распознавания лиц в видеопотоке является метод кластерного анализа, так как он обладает такими критичными преимуществами, как низкая вычислительная нагрузка и высокая точность распознавания в варьирующихся условиях съемки.

Для решения задачи формирования трехмерной модели в ситуации, когда есть только набор растровых изображений лица человека, есть три основных подхода. Первый – генерация трехмерной модели из информации, полученной по изменению положения головы человека в видеопотоке («видеопоток» на рисунке 12). Второй – использование схемы совмещения обобщенной трехмерной карты глубины для лица человека с конкретными особыми точками лица на изображении (обобщенная модель). Третий – модификация метода построения обобщенной модели, когда в качестве карты глубины выбирается карта, генерируемая по неявным правилам, полученным в результате машинного обучения системы на базе изображений лиц, для каждого из которых наличествует соответствующая ему трехмерная модель (деформируемая модель) [94,95,96,97,98].

У каждого из рассмотренных выше методов существуют достоинства и недостатки по таким показателям как быстродействие, точность, зависимость от условий съемки. Рассмотрим эти свойства для каждого из методов.

1). Видеопоток:

+ Индивидуальная модель лица;

- Высокая вычислительная сложность;

- Низкая точность;

- Сильная зависимость от разрешения съемки.

2). Обобщенная модель:

- + Минимальные вычисления на этапе построения модели;
- + Минимальная зависимость от условий съемки;
- Точность ниже среднего.

3). Деформируемая модель:

- + Низкая вычислительная сложность;
- + Минимальная зависимость от условий съемки;
- + Средняя точность;
- Сложность процесса обучения.

Среди методов формирования трехмерной модели наименее подходящим для выбранного метода кластерного анализа является метод видеопотока, так как в неидеальных условиях съемки он не обладает никакими преимуществами, но требователен к ресурсам. Обобщенная модель лица выглядит наиболее привлекательным методом из-за легкой реализации и минимальных требований к ресурсам на всех этапах построения. Деформируемая модель имеет единственное преимущество в виде более высокой точности модели, но этот подход сложен в реализации.

1.2. Анализ

Трекинг

Существующие методы трекинга движущихся объектов, оперирующие изменениями в изображении во времени, требуют много ресурсов и зависят от многих параметров, таких как фон (подвижный или неподвижный), характер движения [99,100,101,102].

В связи с тем, что в процессе распознавания лица человека первым этапом всегда является поиск лица на изображении (например, методом Виолы-Джонса), целесообразно воспользоваться этим в процессе трекинга. Зная координаты зафиксированного лица на изображении, становится возможным оперировать не массивом пикселей, а только его относительным изменением положения в последовательности кадров. Сам метод детектирования области лица на изображении достаточно ресурсоемкий и может занимать больше половины

процессорного времени при обработке видеопотока высокого качества на слабых ЭВМ.

В такой постановке задачи может быть предложен новый более эффективный алгоритм трекинга лица, использующий не только информацию о положении лица на изображении, но и быструю предварительную обработку кадров видеопотока с целью уменьшить область, подходящую для использования детектора.

Сравнение изображений

Методы сравнения изображений, основанные на SURF-дескрипторах либо на вычислении хэш-функций, обладают хорошим быстродействием, но плохо подходят к задаче выделения слабых отличий на изображения, в то время как метод вычисления разности двух изображений очень чувствителен к шумам, что является неотъемлемым свойством изображений в видеопотоке.

Единственным методом, который подходит для выполнения поставленной задачи, является метод вычисления коэффициента корреляции, но его быстродействие сильно зависит от исходного размера изображений, поэтому его можно применять только к области изображения с найденным лицом, то есть необходимо проводить сравнение только треков лиц.

Оценка контрастности и резкости

Для оценки контрастности вполне подходит оценка яркостной контрастности, а для оценки резкости – оценка на основе анализа амплитудного спектра изображения. Однако обе эти оценки дают только приближенное представление о качестве изображения, они могут быть использованы для исключения из обработки заведомо плохих изображений либо для оценки применимости методов восстановления контрастности и компенсации освещения.

Компенсация освещения

Каждый из методов компенсации освещения обладает своими преимуществами, и хотя визуально лучшую картинку дает метод CLANE, для различных методов идентификации личности различная обработка снимков может давать совершенно другие результаты. Чтобы сделать выводы о конечной эффектив-

ности методов, необходимо провести тестирование каждого из методов в совокупности с выбранным методом распознавания лиц.

Контроль ракурса

Для задачи распознавания лица может быть применен как подход восстановления фронтального ракурса лица (в этом случае нет строгих требований к начальному набору изображений лица в базе) либо более общий подход получения множества изображений лица с разными углами поворота относительно оптической оси камеры.

В первом подходе достаточно использовать алгоритм автоматической расстановки точек лица ASM и алгоритм POSIT для оценки угла поворота. Далее применяется алгоритм построения неизвестной трехмерной модели лица и выполняется поворот.

Во втором подходе необходимо двигаться от обратной задачи, то есть выполнять расстановку точек на известном фронтальном лице, и уже для него строить трехмерную модель и формировать изображения поворотов. В этом случае исходное изображение должно быть фронтальным или близким к нему, но из процесса построения модели исключается потенциально ненадежный алгоритм оценки ракурса POSIT, зависящий от точности расстановки точек и их взаимного расположения, качества изображения и тому подобного [103].

Первый подход имеет еще один недостаток, а именно плохую применимость к существующим алгоритмам идентификации личности по изображению лица. Наиболее эффективно такие системы работают в условиях, когда имеется набор изображений в разных условиях съемки, по которым строятся биовектора, соответствующие конкретному человеку. Второй же подход позволяет увеличить число изображений в несколько раз, повысив точность идентификации.

Исходя из этих соображений, поворот фронтального изображения для генерации набора биовекторов представляется предпочтительным.

Для формирования трехмерной модели оптимальным представляется использование обобщенной или деформируемой модели лица, так как не налагает

дополнительных требований к оборудованию и процессу подготовки системы распознавания и предоставляет при этом приемлемую точность.

Для выбора между обобщенной и деформируемой моделями необходимо провести дополнительное тестирование качества распознавания при их использовании.

1.3. Выводы по главе 1.

Рассмотрены различные методы оценки и обработки изображений, позволяющие повысить различимость объектов, выделить последовательность кадров, сравнить два изображения.

Выбраны наиболее оптимальные методы оценки контрастности и резкости, методы компенсации освещения и сравнения изображений.

Проанализированы существующие методы трекинга объектов в видеопотоке, сформулированы условия и требования к трекингу лиц в видеопотоке.

Произведен критический анализ методов и алгоритмов расстановки особых точек лица, метода оценки угла поворота POSIT, а также технологии формирования трехмерных моделей лица, на основе которого обоснована разработка алгоритма построения трехмерной модели лица из одного фронтального растрового изображения и создания набора изображений на основе построенной модели.

Глава 2. Технология подготовки изображений лиц в видеопотоке к распознаванию

2.1. Выбор методов, используемых в составе технологии.

Проведенный в главе 1 обзор и анализ существующих методов обработки изображений показал, что для ряда задач предварительной обработки изображений лиц применительно к распознаванию личности не существует эффективных решений, направленных на реализацию программно-аппаратных систем реального времени.

Алгоритмы трекинга, оценки и компенсации ракурса изображений лиц редко рассматриваются как компоненты системы распознавания, что затрудняет их адаптацию к решению рассматриваемой задачи. Разработка методов, оптимальных для автоматического применения в системах реального времени, является критически важной для формирования комплексной технологии предварительной обработки изображений лиц в видеопотоке.

2.1.1. Трекинг

Для разработки алгоритма трекинга за основу взят простейший метод, оперирующий только данными о положении обнаруженных лиц в кадре.

Видеопоток можно представить в виде последовательности кадров (изображений)

$$V_s = [f_1, f_2 \dots f_N],$$

где N – общее число кадров, f_i – текущий кадр. Любое лицо, найденное в кадре, описывается как набор

$$F_{c_{ij}} = \{I_{f_{ij}}, R_{f_{ij}}\},$$

где $I_{f_{ij}}$ – изображение лица, $R_{f_{ij}}$ – его положение в кадре, i – номер кадра, j – номер лица в кадре. При этом $R_{f_{ij}}$ можно представить как набор из двух точек

$$R_{f_{ij}} = \{p_1=(x_1; y_1), p_2=(x_2; y_2)\},$$

где p_1 – верхняя левая точка прямоугольной области, p_2 – нижняя правая точка, x и y – их координаты.

Тогда трек лица k в видеопотоке – это последовательность лиц $F_{c_{ik}}$, соответствующая одному человеку:

$$Tr_k = [F_{c_{1k}}, F_{c_{2k}} \dots F_{c_{Nk}}].$$

Основная задача в трекинге состоит в сопоставлении положений целевого объекта на последовательности кадров. Таким образом, системы трекинга обычно используют модель движения, которая описывает, как может изменяться изображение целевого объекта при всевозможных различных его движениях [52]. В случае слежения за положением лица предполагается, что объекты в следующем кадре меняют свою позицию незначительно. В связи с этим предложен следующий метод трекинга: найденное изображение лица принадлежит треку Tr_k , если точка-центр прямоугольника Rf , $p_c = (\frac{x_1 + x_2}{2}; \frac{y_1 + y_2}{2})$, лица $F_{c_{ij}}$ принадлежит области прямоугольника Rf лица $F_{c_{i-1,j}}$, найденного на предыдущем кадре и принадлежащего треку Tr_k (рис. 18).

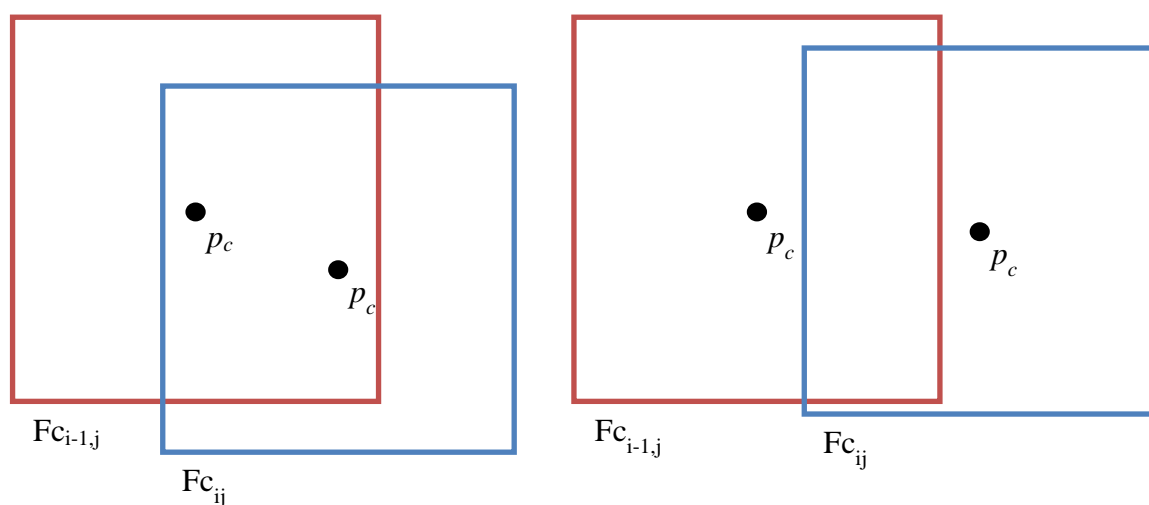


Рис 18. Функционирование простейшего метода трекинга: слева – области обнаруженных лиц принадлежат одному треку, справа – области лиц будут считаться принадлежащими разным людям

Рассмотренный метод не может гарантировать абсолютную точность распознавания трека, но является очень простым с вычислительной точки зрения и часто достаточно эффективен для решения задачи объединения изображений

лиц на видео в последовательности при уточнении результирующей оценки принадлежности изображений лица человеку в системах распознавания. Пример трека лица в видеопотоке, полученного таким методом, представлен на рис. 19.



Рис 19. Пример трека лица в видеопотоке

Данный метод может быть улучшен в двух направлениях – в направлении повышения качества слежения, то есть устранения разрывов в движении из-за несрабатывания детектора, и в направлении повышения скорости вычислений.

Для компенсации недостатка точности этого метода существует ряд решений, направленных на удержание фокуса интереса на областях кадра, в которых детектору не удалось обнаружить лицо, с целью связать его с предыдущим на основании некоторых отличительных особенностей, таких как взаимное расположение точек, их смещение (например, методы SURF, mean shift, KLT) [104]. Однако, с точки зрения применения в системах распознавания, подобный трекинг необходим не всегда. Для задачи распознавания анализ черт лица по изображению, которое искажено в степени, не допускающей его обнаружения детектором, не имеет практического смысла: это может привести к снижению как точности, так и быстродействия системы распознавания. Стоит отметить, что в случае применения алгоритмов распознавания, основанных на анализе 3D образов лиц или частных черт лица, не привязанных к целному объекту, методы улучшения качества трекинга имеют большее значение из-за более высокой вероятности распознать лицо по изображению, содержащему сильные помехи.

В рассматриваемой задаче, связанной с повышением эффективности и скорости обработки видеопоследовательностей системами распознавания личности большее значение имеет быстродействие трекинга. В контексте функци-

онирования системы распознавания реального времени необходимость непрерывного сканирования всей области изображения детектором лиц нивелирует преимущества трекинга лиц существующими методами. Для того, чтобы снизить вычислительную нагрузку, которая может занимать больше половины вычислительных ресурсов в зависимости от архитектуры ЭВМ и параметров ПО, имеет смысл проводить предварительный анализ кадров видео для выделения областей, потенциально содержащих лица. В этом плане наиболее перспективными являются быстрые методы вычитания фона и обнаружения движения, состоящие из простейших операций вычисления разницы между двумя кадрами видеопотока и являющиеся легкими с вычислительной точки зрения [105].

Предлагается алгоритм ускорения базового метода трекинга лиц, основанного на положении лиц в кадре, состоящий из следующих действий:

1. Вычисление разницы между текущим кадром и предыдущим, определяющей наличие движения в потоке видео:

$$f_M = f_i - f_{i-1},$$

где f_i – текущий кадр видеопотока, f_M – матрица, каждый из элементов которой определяет разность соответствующих пикселей f_i и f_{i-1} кадров видеопотока.

2. Вычисление разницы между текущим кадром и первым, определяющей наличие в потоке видео объектов, не являющихся фоном:

$$f_T = f_i - f_0,$$

где f_i – текущий кадр видеопотока, f_T – матрица, каждый из элементов которой определяет разность соответствующих пикселей f_i и f_0 кадров видеопотока.

3. Вычисление бинарных изображений f_T и f_M с порогом, зависящим от качества видеопотока и шумов.
4. Оценка распределения белых пикселей бинарного изображения для f_T и f_M с помощью скользящего окна со стороны, равной половине минимального окна детектора лиц. Те области изображения, в которых сумма белых пикселей превышает выбранный порог, отмечаются как подходящие для

дальнейшего анализа. Результирующие области для изображений f_T и f_M объединяются, что помогает включить в общую область как движущиеся объекты, так и объекты, находящиеся в кадре неподвижно.

5. Детектор лиц применяется в выбранной области изображения, позволяя выделить среди подвижных объектов лица людей.

Такой алгоритм позволит избежать вычисления вейвлетов Хаара для всего кадра целиком, значительно снижая вычислительную сложность трекинга лиц по координатам области изображения лица.

Следует отметить, что предложенный алгоритм способен эффективно работать только в ограниченных условиях: при наличии неподвижного, преимущественно статичного фона, то есть с видеопотоками, полученными от камер видеонаблюдения на стационарных объектах. В случае большой динамики сцен такой способ ускорения обработки кадров окажется неэффективен и даже приведет к замедлению алгоритма поиска лиц на изображении. Медленные или незначительные изменения фона могут быть компенсированы регулярной заменой опорного кадра в моменты, когда суммарная разница соседних кадров в целом меньше некоторого критического порога [106].

Для видеопотока, анализируемого в каждом цветовом канале отдельно, есть возможность еще больше повысить точность выделения областей, что приведет к частичному снижению общей производительности трекинга.

В общем случае выделяемая область с объектами будет меньше кадра целиком, что ведет к снижению числа операций с детектором лиц Виолы-Джонса. В идеальном случае, когда в кадре нет движения и объектов, детектор вовсе не выполняет свои вычисления, что высвобождает дополнительные ресурсы на обработку других кадров видеопотока, находящихся в буфере ввода-вывода.

2.1.2. Сравнение изображений

Из трех рассмотренных подходов к сравнению изображений для решения поставленной задачи наиболее подходящим является метод вычисления коэффициента корреляции. С помощью этого метода можно исключать из трека T_{T_k}

изображения с высоким уровнем корреляции. В общем случае корреляция двух разномасштабных изображений может быть вычислена по следующей формуле:

$$R(x, y) = \frac{\sum_{x', y'} (If_{ik}(x', y') \cdot If_{i+1, k}(x + x', y + y'))}{\sqrt{\sum_{x', y'} If_{ik}(x', y')^2 \cdot \sum_{x', y'} If_{i+1, k}(x + x', y + y')^2}} \quad (5)$$

где If_{ik} – исходное изображение, $If_{i+1, k}$ – изображение для сравнения. Оценка, получаемая по формуле (5), лежит в диапазоне от 0 до 1, где 0 соответствует максимально разным изображениям, а 1 – максимально похожим.

Применяя операцию вычисления $R(x, y)$ по формуле (5) для образца и участка изображения, можно получить изображение в градациях серого, в котором наиболее яркое пятно соответствует наиболее похожему на образец участку изображения (рис. 20).

Для двух изображений одинакового размера матрица $R(x, y)$ будет состоять только из одного элемента, также лежащего в диапазоне [0;1] и характеризующего степень схожести этих изображений.



Рис. 20. Результат вычисления коэффициента корреляции

Этот метод достаточно корректно отображает степень похожести двух изображений, но плохо применим для оценки кадра целиком, так как требова-

телен к вычислительным ресурсам. Однако его можно применять для поиска серии лиц на изображении, принадлежащих одному человеку, и выбору из этой серии только отличающихся друг от друга изображений.

2.1.3. Методы оценки контрастности и резкости

Для оценки контрастности применим метод вычисления нормированной яркостной контрастности изображения по формулам (1)-(2). Применение этой оценки для кадра целиком или области лица на изображении может дать непредсказуемый результат вычислений. В качестве оптимального варианта для вычисления можно использовать только те точки изображения, которые с высокой вероятностью принадлежат области лица, то есть точки, лежащие внутри фигуры, образуемой контуром овала лица и бровей, полученным одной из модификаций алгоритма ASM.

Простейший алгоритм, позволяющий вычислить такие точки, заключается в следующем:

1. Вычисление особых точек лица;
2. Увеличение яркости каждой точки изображение на единицу, таким образом, на изображении не остается нулевых значений яркости;
3. Разбиение области лица на треугольники, образованные ближайшими особыми точками, и сложение изображений треугольников с изображением нулевой яркости.

В результате этого получится изображение, где все не принадлежащие лицу точки изображения обладают нулевой яркостью, и по этому критерию можно исключать их из вычисления оценки контрастности изображения.

Аналогично этот подход можно применить и для оценки резкости, но даже с учетом этого количественная оценка резкости изображения затруднительна. Для вычисления относительной оценки резкости можно воспользоваться формулой (4).

2.1.4. Метод коррекции освещения

Среди методов коррекции освещения невозможно выбрать оптимальный без анализа того, как каждый из методов влияет на конечный результат распознавания. Поэтому для алгоритма подготовки изображений решено протестировать все три метода и выбрать тот, с использованием которого эффективность распознавания будет максимальной.

2.1.5. Метод коррекции ракурса

В качестве основы выбран метод использования готовых трехмерных моделей лица. Для упрощения системы можно сделать допущение, что в качестве модели глубины изображения лица используется некоторая обобщенная модель, предполагая похожий рельеф основных черт, таких как глаза, брови, нос и рот. Такое допущение понизит точность модели, но позволит избежать её вычисления для каждого человека.

В качестве обобщенной трехмерной модели лица была выбрана модель, изображенная на рисунке 21, представляющая собой массив значений глубины 181x122.

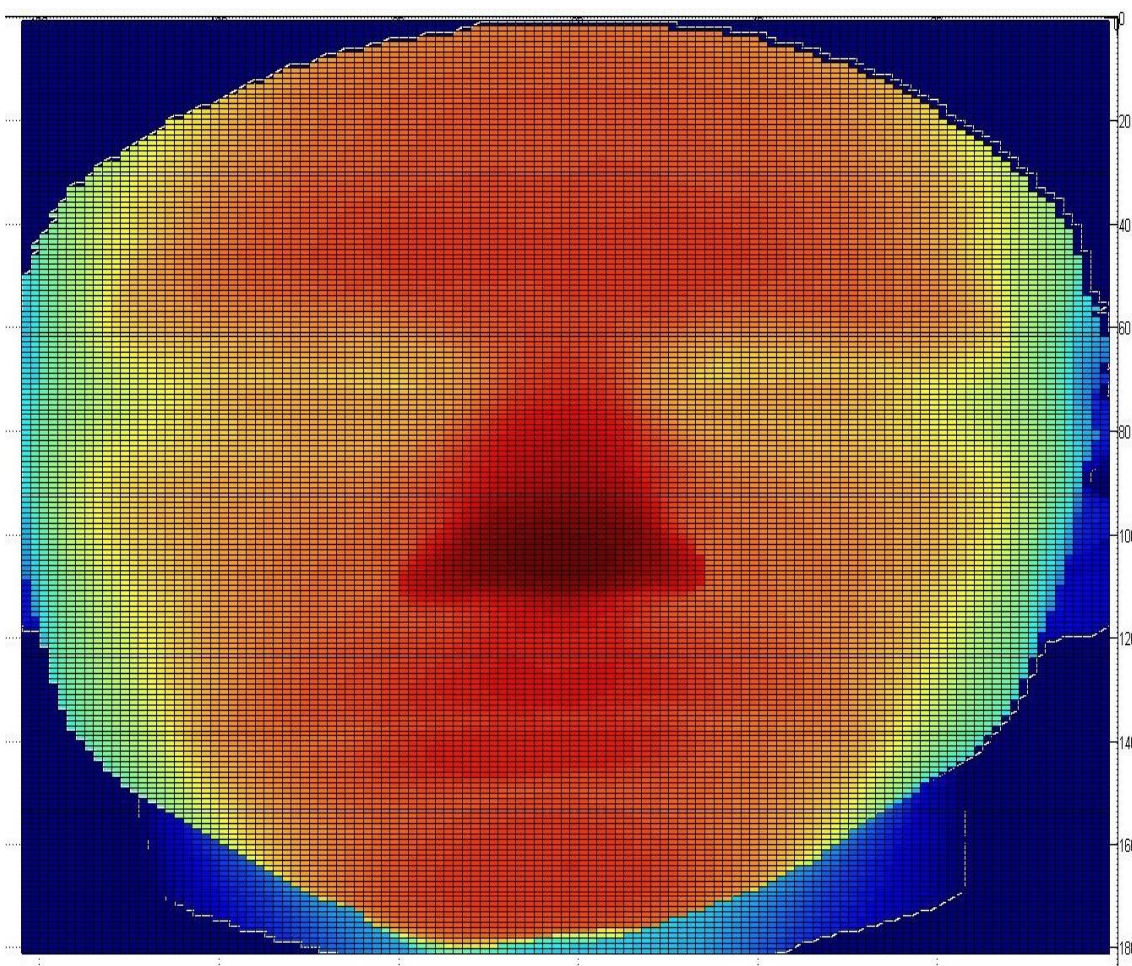


Рис. 21. Изображение карты третьей координаты лица, построенное в среде MATLAB

Для решения задачи коррекции ракурса необходимо произвести следующие операции:

- Совмещение обобщенной модели глубины с особыми точками лица, полученными с помощью одного из вышеописанных алгоритмов;
- Разбиение полученной модели лица на треугольники (триангуляция). В результате этой операции будет получен набор треугольников, описывающий область лица на изображении. Его можно использовать для построения нормированного изображения, но результат получится достаточно грубым (рис. 22);



Рис. 22. а) сетка, наложенная на исходное изображение; б) нормированное изображение

- Дробление треугольников по выбранному критерию, позволяющему снизить размеры используемых треугольников и приблизить используемую треугольную сетку к равномерной. В качестве критерия разбиения можно брать максимальную площадь треугольника либо максимальную длину стороны. Критерий разбиения по максимальной длине позволяет получить более пропорциональную сетку (рис. 23);

Процесс разбиения заключается в циклическом выполнении следующих действий:

- 1). Поиск самой длинной грани треугольника в сетке.
- 2). Создание новой точки ровно посередине самой длинной грани.
- 3). Построение двух новых треугольников вместо одного изначального. Если грань общая для двух треугольников, то смежный треугольник разбивается аналогично.

Процесс повторяется до тех пор, пока суммарное число треугольников не достигнет необходимого значения, либо пока максимальная площадь треугольника не станет меньше заданного порога.



Рис. 23. Сетка из тысячи треугольников: а) полученная с критерием разбиения по максимальной площади; б) полученная с критерием разбиения по максимальной длине стороны

- Каждый из треугольников с помощью операций вращения относительно заданного центра и аффинного преобразования смещается и деформируется в соответствии с величиной угла, определенной с помощью алгоритма POSIT.

Алгоритм POSIT может быть описан следующим образом [70]: пусть задан набор из соответствующих друг другу двумерных точек и трехмерных точек. Требуется найти такие значения матриц поворота и векторов сдвига, называемых параметрами проектирования, которые приведут к максимальному соответствию между проекциями трехмерных точек и двумерными точками.

Опишем камеру-обскуру (классическую модель) с центром проектирования в точке O . Плоскость изображения G этой камеры находится на некотором расстоянии f от центра проектирования O (т.н. фокусное расстояние). Фокусное расстояние предполагается известным изначально., в то время как Оси O_x и O_y направлены вдоль строк и столбцов сенсора камеры, а ось O_z - вдоль оптической оси. Пусть единичные вектора каждой из осей будут определены как i, j и k . Некоторый объект с известными ключевыми точками M_0, \dots, M_n расположен в поле зрения заданной камеры. Точка M_0 интерпретируется как относительная

точка объекта. Все прочие M_i -координаты точек при $i > 0$ могут быть представлены в координатах относительно точки M_0 . Также по условию известны координаты точек $m_i (x_i, y_i)$ на изображении.

В таком случае, для определения величины угла необходимо найти M_i -координаты точек в координатах (X_i, Y_i, Z_i) камеры. Матрица поворота камеры и вектор сдвига будут получены в процессе нахождения этих точек, на основе этого станет известно положение камеры.

Для объекта матрица поворота R – это матрица, строки которой являются в координатной системе камеры координатами единичных векторов i, j, k , представленных в координатной системе объекта (M_0u, M_0v, M_0w) . Матрица поворота используется для преобразования объектных координат в координаты камеры. Проекция вектора M_0M_i на вектор i является из скалярным произведением. Таким образом в координатах камеры определяется разница между X -координатами точек M_0 и M_i .

Для того, чтобы вычислить поворот, нужно определить координаты векторов i и j в координатной системе объекта. Далее, вектор k выражается как векторное произведение $i \times j$. Вектор OM_0 , находящийся между относительной точкой объекта (начало координат в системе объекта) и центром проектирования (камеры) – это вектор сдвига T . Отсюда следует, что $T = \{X_0, Y_0, Z_0\}$. При условии, что относительная точка M_0 задана соответствующей точке m_0 на изображении, длина вектора сдвига определяется через значение длины вектора Om_0 : $T = Z_0/f * Om_0$. Исходя из этого, для нахождения вектора сдвига требуется определить только Z -координату точки M_0 . Можно заключить, что положение объекта целиком и полностью выражается через значения i, j и Z_0 .

Аппроксимация к настоящей перспективной проекции называется масштабированной ортографической проекцией (МОП). В такой аппроксимации для заданного объекта, который находится в зоне видимости камеры, можно сделать предположение о том, что в координатах камеры для всех точек объекта (M_i) Z -координаты значительно друг от друга не отличаются. Следовательно,

каждая из них может считаться равной координате относительной точки объекта Z_0 . Проекция точки объекта M_i в МОП будет иметь координаты:

$$x'_i = \frac{fX_i}{Z_0}, \quad y'_i = \frac{fY_i}{Z_0},$$

а перспективная проекция для этой же точки выразится через координаты

$$x_i = \frac{fX_i}{Z_i}, \quad y_i = \frac{fY_i}{Z_i}.$$

Такое соотношение f/Z_0 названо масштабным множителем МОП. Перспективная проекция совпадает с МОП для относительной точки объекта. Для точки M_i проекция имеет следующий вид:

$$x'_i = \frac{fx_0}{Z_0} + \frac{f(x_i - x_0)}{Z_0} = x_0 + s(X_i - X_0), \quad (6)$$

$$y'_i = y_0 + s(Y_i - Y_0).$$

Настоящая перспективная проекция, а также соотношение неизвестных векторов матрицы поворота i, j и неизвестной Z_0 компоненты вектора сдвига к известным координатам проекций x_i, y_i и известным в системе координат объекта координатам векторов M_0M_i описываются двумя уравнениями, решение которых предоставит достаточно информации для восстановления положения объекта.

$$M_0M_i \cdot \frac{f}{Z_0} i = x_i(1 + \varepsilon_i) - x_0, \quad (7)$$

$$M_0M_i \cdot \frac{f}{Z_0} j = y_i(1 + \varepsilon_i) - y_0. \quad (8)$$

Через ε_i обозначено

$$\varepsilon_i = \frac{1}{Z_0} M_0M_i \cdot k,$$

где k находится как $i \times j$.

Пусть заданы плоскость K , которая параллельна G и проходит через M_0 , и точки M_0, M_i объекта. Исходя из предположения, что луч O из центра проекти-

рования проходит через точку M_i и в точке N_i пересекает плоскость K , можно ввести P_i как обозначение ортогональной проекции точки M_i на плоскость K .

В таком случае вектор M_0M_i определяется как сумма:

$$M_0M_i = M_0N_i + N_iP_i + P_iM_i.$$

Пропорция векторов M_0N_i и m_0m_i вычисляется как отношение Z_0 к f . Пропорция пары векторов N_iP_i и Sm_i вычисляется как отношение z -координат других соответствующих векторов в треугольниках Sm_iO и $N_iP_iM_i$, P_iM_i и OC , из свойства подобия этих треугольников. То есть, $N_iP_i / Sm_i = M_0M_i \cdot k / f$. Сумма всех трех векторов определяется следующим выражением:

$$M_0M_i = \frac{Z_0}{f} m_0m_i + \frac{M_0M_i \cdot k}{f} Sm_i + P_iM_i.$$

Скалярное произведение этого равенства на единичный вектор i из системы координат камеры позволяет определить, что $P_iM_i \cdot i = 0$, а $m_0m_i \cdot i = x_i - x_0$, x -координата вектора. Соответственно, координата x_i вектора Sm_i вычисляется как $Sm_i \cdot i$. Обозначив $\varepsilon_i = M_0M_i \cdot k / Z_0$, можно получить уравнение 7. Аналогично, домножением на j , получается уравнение 8.

Другими словами, термы $x_i(1 + \varepsilon_i)$ и $y_i(1 + \varepsilon_i)$ в правой части уравнений 7 и 8 - это координаты точки $p_i (x_i', y_i')$, которая является МОП точки M_i .

Если рассматривать точки M_0 и M_i проекции P_i на плоскость K , а также её образ p_i , то координаты точки p_i на изображении обозначаются как x_i' и y_i' , а вектор $M_0M_i = M_0P_i + P_iM_i$. Пропорция первого из векторов, M_0P_i , и вектора m_0p_i равна отношению Z_0 к f . Отсюда следует, что, умножив скалярно это равенство на i или j , можно получить следующие соотношения:

$$M_0M_i \cdot \frac{f}{Z_0} i = x_i' - x_0,$$

$$M_0M_i \cdot \frac{f}{Z_0} j = y_i' - y_0.$$

Сравнивая полученные уравнения с равенствами 7 и 8 получаем, что

$$x_i' = x_i \cdot (1 + \varepsilon_i), y_i' = y_i \cdot (1 + \varepsilon_i).$$

Равенства 7 и 8 могут быть переписаны в виде

$$M_0 M_i \cdot I = x_i(1 + \varepsilon_i) - x_0, \quad (9)$$

$$M_0 M_i \cdot J = y_i(1 + \varepsilon_i) - y_0, \quad (10)$$

где $I = i * f / Z_0$, а $J = j * f / Z_0$.

Основную идею рассматриваемого метода можно описать следующим образом: если значения ε_i известны, то есть возможность определить координаты I и J из описанных выше уравнений. Затем из этих координат нормализацией могут быть получены вектора i и j , а из нормы I или J может быть получена T_z . Такой алгоритм назван Pose from Orthography and Scaling, или POS. Очевидно, что поиск положения зависим от параметров ε_i . Если ε_i - не точные значения, то решение, получаемое с помощью алгоритма POS, является приближенным. После того, как будут вычислены i и j для некоторых ε_i , появляется возможность получить более точные значения для ε_i , и использовать их повторно. В дальнейшем этот шаг уточнения может быть повторен несколько раз. Подобный алгоритм называется POS with Iterations, или POSIT. Этот алгоритм может найти i, j и Z_0 за несколько итераций.

Первоначально предполагается, что $\varepsilon_i = 0$. Исходя из того, что $\varepsilon_i = 0$, можно также заключить, что $x_i = x_i'$, $y_i = y_i'$, то есть m_i совпадает с p_i . Приемлемый результат будет найден на одной из первых итераций, если отличия z -координат точек объекта минимальны.

На каждой итерации алгоритма POSIT для точек M_i ($i = 1..N$) необходимо решить систему, которая составлена из уравнений вида 9, 10.

В более компактной форме они будут иметь вид:

$$M_0 M_i \cdot I = \xi_i,$$

$$M_0 M_i \cdot J = \eta_i,$$

где

$$I = \frac{f}{Z_0} i, \quad J = \frac{f}{Z_0} j, \quad \xi_i = x_i(1 + \varepsilon_i) - x_0, \quad \eta_i = y_i(1 + \varepsilon_i) - y_0,$$

и где ε_i – это известные параметры, полученные на предыдущем шаге.

В результате получены линейные уравнения, в которых вектора I и J - неизвестные параметры, а известными являются координаты точек m_i, m_0 на изображении: x_i, y_i, x_0, y_0 . Также известны координаты точек в относительной (объектной) системе координат - U_i, V_i, W_i .

Подставив n точек объекта M_1, \dots, M_n и их образы на изображении в уравнение 9, систему уравнений можно представить в виде

$$AI=x', AJ=y',$$

где A – это матрица координат точек объекта M_i в системе координат объекта, x' - вектор, который составлен из элементов ξ_i , а y' - вектор, который составлен из элементов η_i . Если рассматривать общий случай, при хотя бы 3 видимых точках помимо M_0 , и с учетом того, что эти 4 точки не принадлежат одной плоскости, то матрица A будет иметь ранг 3. Тогда решение системы методом наименьших квадратов будет иметь следующий вид:

$$I = Bx', J = By',$$

где B – это матрица, псевдообратная к матрице A . Такая матрица называется объектной. Объектную матрицу можно предрассчитать, если известно взаимное расположение точек объекта. Для этого существуют различные способы: Singular Value Decomposition, декомпозиция матрицы A по сингулярным числам или вычисление по формуле $[A^T A]^{-1} A^T$. Решение с помощью декомпозиции минимизирует значения ошибок $|AI-x'|$ и $|AJ-y'|$. Кроме того, преимущество этого метода заключается в том, что при анализе сингулярных чисел, еще до запуска самого алгоритма, можно определить, принадлежат ли выбранные точки одной плоскости.

Как только найдены решения для I и J с помощью метода наименьших квадратов, посредством нормализации могут быть получены единичные векторы i и j . В соответствии с вышеописанным можно заключить: первая строка матрицы поворота составлена из 3 координат вектора i , вторая - из 3 координат вектора j , а третья - из 3 координат вектора k , который получен как векторное произведение первых двух.

В результате вектор сдвига определяется как вектор OM_0 от точки M_0 (относительной точки объекта) до центра проектирования O . Вектор сдвига равен $Z_0 * Om_0/f$, или Om_0/s . s – это масштабный множитель, он вычисляется как норма вектора I или норма вектора J (либо как среднее между этими векторами). В случае, если параметр ε_i – точное значение, вектор сдвига и матрица поворота определяют точное представление положения объекта относительно камеры. В противном случае значения ε_i можно уточнить и повторить итерацию.

В результате этих операций получается изображение лица, нормированное по углам поворота.

Результат обработки изображения лица показан на рис. 24.



Рис. 24. а) изображение лица, повернутого на 15 градусов по оси X; б) изображение лица, скорректированного на 15 градусов

Рассмотрим другой подход формирования модели лица, который позволяет получить индивидуальную модель на основе растрового изображения лица и набора трехмерных моделей лиц. Для этого предложена следующая модификация метода деформируемой модели лица [107,108].

В качестве основы для формирования модели был взят набор из 11 трехмерных моделей лиц, находящийся в свободном доступе [109]. Каждая трехмерная модель содержит в себе как облако точек в трехмерном пространстве (рис. 25), так и текстурную информацию, позволяющую получить изображение исходного лица во фронтальном ракурсе и, соответственно, найти особые точки (рис. 26).

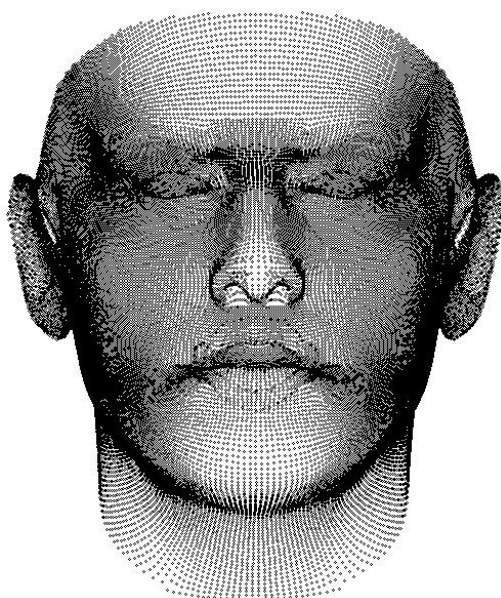


Рис. 25. Облако точек лица в трехмерном пространстве



Рис. 26. Пример фронтального ракурса лица, полученного с помощью наложения текстуры на одну из трехмерных моделей

При наличии достаточно большого числа трехмерных моделей лиц в качестве искомой модели можно взять ближайшую эталонную модель (особые точки модели в среднем наименее удалены от особых точек анализируемого лица), но более точным решением является интерполяция каждой точки трехмерной модели исходя из близости к эталонным моделям.

Для вычисления модели была применена модификация метода интерполяции Шепарда для каждой особой точки [110]:

$$w_i = \frac{\left(\frac{R - h_i}{R \cdot h_i}\right)^2}{\sum_{q=1}^N \frac{R - h_q}{R \cdot h_q}}, \quad (1)$$

где i – номер модели лица, N – число моделей, h_i – расстояние от точки модели до точки изображения на плоскости (X, Y) , R – заданный радиус, $R \geq \min(h_i)$, w_i – коэффициент удельного веса модели при вычислении третьей координаты особой точки лица.

При этом z -координата определяется как сумма произведений z -координат моделей и соответствующих весов, полученных из формулы (1):

$$z_j = \sum_{i=1}^N z_i \cdot w_i, \quad (2)$$

где j – номер особой точки изображения лица.

Данная модификация метода позволяет автоматизировать процесс интерполяции, вычисляя радиус R через другие параметры – размерность пространства, число и плотность точек модели. Радиус R определяется автоматически по формуле:

$$R = \max_{i \in K} |x - x_i|, \quad (3)$$

где $K = N_w < N$ – число ближайших точек. Согласно общим рекомендациям [111] по выбору числа точек, принято значение $N_w = 9$. Данная модификация хорошо подходит для случаев неравномерной неизвестной заранее сетки, а также позволяет использовать вместо константных значений z_j в формуле (2) некоторый функционал, например линейный или квадратичный [112]. В рассматриваемом случае было решено ограничиться константными значениями.

Этот метод позволяет получить уникальную модель анализируемого лица, которая будет более точной, чем любая из эталонных моделей. При таком подходе увеличение числа эталонных моделей аналогично ведет к улучшению точности, но не является критичным для успешности вычисления [113].

Исходя из этого, был предложен следующий алгоритм формирования индивидуальной модели лица:

1. Поиск соответствия между найденными особыми точками и известными трехмерными моделями.

Для того чтобы найти соответствие, необходимо получить фронтальную проекцию множества точек трехмерных моделей лиц $P_i(x, y, z)$. То есть, найти такое множество точек $P_i(x^*, y^*)$, при котором координаты центров глаз будут находиться на строго горизонтальной линии (координата x^*), а серединные точки рта и переносицы – на строго вертикальной линии (координата y^*). [114]

Так как в выбранном для предложенного алгоритма наборе моделей координаты x и y соответствуют координатам x^* и y^* фронтальной проекции, то итоговая проекция получается отбрасыванием третьей координаты z . Далее, из этой проекции можно сформировать фронтальное изображение лица по облаку точек $P_i(x, y)$ и текстуре трехмерной модели.

Для дальнейшего нормирования полученного фронтального изображения модели к фронтальному изображению лица необходимо вычислить набор особых точек $E_j(x, y)$ для изображения, полученного из трехмерной модели, и изображения анализируемого лица с помощью описанного ранее метода ASM.

2. Нормирование особых точек относительно масштаба моделей (например, расстояние между центрами глаз) и центра координат (например, кончик носа).

Для нормирования плоского изображения необходимо определить как минимум две точки, относительно которых будет проводиться масштабирование остальных координат и сведение центров координат. В общем случае координаты изображения лица соответствуют пикселям этого изображения и по определению лежат в плоскости целых положительных чисел, а координаты проекции трехмерной модели могут быть дробными и отрицательными числами.

В рассматриваемой задаче удобнее всего при вычислении коэффициента масштабирования опереться на координаты особых точек, вычисленным алгоритмом ASM. Для этого достаточно взять две устойчивых точки, координаты центров глаз для изображения проекции модели, $E_l(x_l, y_l)$ и $E_r(x_r, y_r)$, а также для изображения анализируемого лица, $E'_l(x'_l, y'_l)$ и $E'_r(x'_r, y'_r)$. Расстояния между этими точками находятся по формулам

$$D = \sqrt{(x_l - x_r)^2 + (y_l - y_r)^2} \quad \text{и} \quad D' = \sqrt{(x'_l - x'_r)^2 + (y'_l - y'_r)^2}.$$

В результате коэффициент масштабирования для изображения анализируемого лица определяется как соотношение $K = D/D'$.

В качестве точки центра координат корректно будет указать точку, которая не является симметричной в модели ASM. Эта точка – кончик носа, $E_n(x_n, y_n)$. Для совмещения центров координат необходимо найти расстояние между координатами точки $E_n(x_n, y_n)$ изображения проекции модели и координатами точки $E'_n(x'_n, y'_n)$ изображения анализируемого лица:

$$D_x = x_n - x'_n, \quad D_y = y_n - y'_n.$$

После вычитания этих расстояний из каждой точки изображения лица и последующего умножения на коэффициент масштабирования K изображения лица и проекции модели становятся нормированными относительно одной и той же системы координат.

3. Вычисление расстояний для каждой особой точки изображения лица и соответствующих особых точек каждой из трехмерных моделей.

Так как полученные координаты особых точек для изображений нормированы по точкам центров глаз E_l и E_r , а также центра носа E_n , то расстояния между остальными точками массивов E и E' характеризуют различия изображения анализируемого лица и трехмерной модели, использованной при формировании фронтальной проекции:

$$F_i = \sqrt{(x_i - x'_i)^2 + (y_i - y'_i)^2}$$

4. Вычисление координаты глубины для особых точек изображения лица путем интерполяции значений в особых точках моделей.

Вычисленное на предыдущем этапе расстояние между соответствующими особыми точками известной модели и заданного изображения лица позволяет судить о близости трехмерных моделей к лицу человека на изображении. После вычисления расстояний для проекции каждой из трехмерных моделей появится возможность применить формулы интерполяции Шепарда, задав радиус R . Таким образом, для каждой особой точки изображения лица находится координата глубины z , позволяющая судить о рельефе анализируемого лица.

5. Определение координаты глубины для всех остальных точек искомой модели постепенным вычислением новых точек между известными особыми точками до тех пор, пока число точек модели не достигнет числа точек эталонных моделей.

Для получения более подробной трехмерной модели лица недостаточно вычисления третьей координаты только в особых точках, поэтому необходимо расширить массив до некоторого предела $Q \leq N$, где N – число точек исходных трехмерных моделей [115].

В этом случае к описанной ранее процедуре дробления треугольников изображения лица добавляется вычисление координаты z для новой точки. Напрямую применять интерполяцию в данном случае некорректно, так как для новой точки изображения лица неизвестны соответствующие точки моделей. Для решения этой проблемы необходимые веса w_i новой точки вычисляются по линейному соотношению весов двух ближайших особых точек. Этим достигается гладкость и непрерывность результирующей трехмерной модели.

В результате этих операций получается индивидуальная трехмерная модель лица, которую можно использовать для алгоритма построения изображения повернутого на заданный угол лица.

2.2. Выбор средств разработки.

К программной реализации разрабатываемых алгоритмов предъявлено несколько требований:

1. Реализация в виде программы, доступной для использования в операционной системе семейства Windows;
2. Выдача результатов работы в формате, пригодном для дальнейшего использования в процессе распознавания;
3. Возможность выбора методов обработки изображений в интерфейсе программы.

В качестве основного языка программирования было решено использовать C++, как наиболее распространённый и имеющий реализации компиляторов для большинства популярных операционных систем. В процессе разработки использовалась среда разработки MS Visual Studio 2013 вместе с Qt (кроссплатформенный инструментарий разработки ПО на языке программирования C++), а также библиотека OpenCV для работы с графическими объектами и Dlib для поиска особых точек лица на изображениях.

Qt позволяет запускать программы в большинстве современных операционных систем путём простой их компиляции для каждой ОС без изменения исходного кода. Включает в себя все основные классы, которые могут потребоваться при разработке прикладного программного обеспечения, начиная от элементов графического интерфейса и заканчивая классами для работы с сетью, базами данных и XML. Qt является полностью объектно-ориентированным, легко расширяемым и поддерживающим технику компонентного программирования.

Отличительная особенность Qt от других библиотек — использование Meta Object Compiler (МОС) — предварительной системы обработки исходного кода. МОС позволяет во много раз увеличить мощь библиотек, вводя такие понятия, как слоты и сигналы. Кроме того, это позволяет сделать код более лаконичным.

Qt комплектуется визуальной средой разработки графического интерфейса «Qt Designer», позволяющей создавать диалоги и формы в режиме WYSIWYG. В поставке Qt есть «Qt Linguist» — графическая утилита, позволяющая упростить локализацию и перевод программы на многие языки; и «Qt Assistant» — справочная система Qt, упрощающая работу с документацией по библиотеке, а также позволяющая создавать кросс-платформенную справку для разрабатываемого на основе Qt ПО. В комплект Qt включена среда разработки «Qt Creator», которая включает в себя редактор кода, справку, графические средства «Qt Designer» и возможность отладки приложений. «Qt Creator» может использовать GCC или Microsoft VC++ в качестве компилятора и GDB в качестве отладчика. Для Windows версий библиотека комплектуется компилятором, заголовочными и объектными файлами MinGW.

Qt распространяется по 3 лицензиям: Qt Commercial, GNU GPL, GNU LGPL. Все они позволяют разрабатывать ПО с собственной лицензией.

С помощью Qt реализовано множество приложений, таких как Autodesk Maya, Skype, Telegram, Медиапроигрыватель VLC, VirtualBox, а также используется такими фирмами, как DreamWorks, Google, HP, Lucasfilm, Panasonic, Philips, Samsung, Siemens, Volvo, Walt Disney Animation Studios и другие.

OpenCV (англ. Open Source Computer Vision Library, библиотека компьютерного зрения с открытым исходным кодом) — библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. Реализована на C/C++, также разрабатывается для других языков программирования. Может свободно использоваться в академических и коммерческих целях — распространяется в условиях лицензии BSD. Основное назначение библиотеки – утверждение общего стандартного интерфейса компьютерного зрения для приложений в этой области, способствование росту числа таких приложений и создания новых моделей использования PC.

Dlib – кросс-платформенная библиотека алгоритмов общего назначения с открытым исходным кодом, написанная на языке программирования C++. Биб-

лиотека представляет собой совокупность независимых программных компонентов.

Dlib включает в себя разнообразные инструменты, такие как программные компоненты для работы с сетью, потоками, графическими пользовательскими интерфейсами, структурами данных, линейной алгеброй, машинным обучением, обработкой изображений, анализом данных, XML и текстовым разбором, численной оптимизацией, байесовскими сетями и многими другими задачами. Большое внимание уделено созданию широкого набора статистических инструментов машинного обучения.

2.3. Общий вид алгоритма, описывающего технологию подготовки и обработки изображений.

Общий вид разработанной технологии подготовки и обработки изображений лиц в видеопотоке представлен схемой на рисунке 27.

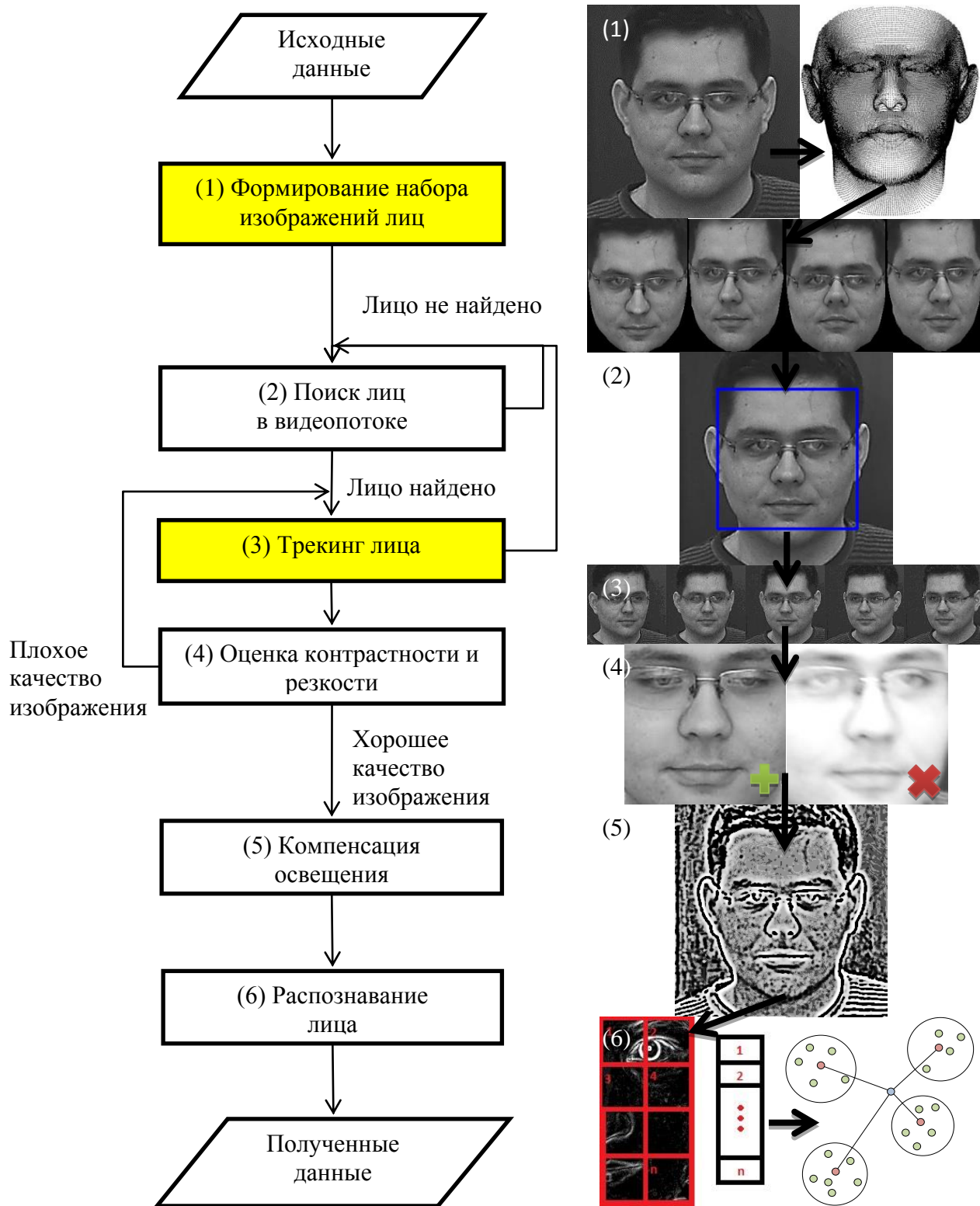


Рис. 27. Схема технологии подготовки изображений лиц к распознаванию

Исходные данные: растровые фронтальные изображения для каждого человека, которого требуется распознавать; набор 3D моделей лиц, видеопоток.

Блок 1. Формирование набора изображений лиц: для каждого изображения лица производится поиск особых точек и модифицированным методом

Шепарда вычисляется индивидуальная 3D модель лица. На основе полученной 3D модели создается набор изображений с разными углами поворота лица, который используется для формирования базы данных по кластерам.

Блок 2. Поиск лица в видеопотоке: с помощью метода Виолы-Джонса производится поиск лица в кадре из видеопотока, если лицо найдено, оно передается на обработку, в ином случае поиск продолжается в следующем кадре.

Блок 3. Трекинг лица: координаты положения изображения лица в двух соседних кадрах сравниваются в соответствии с предложенным алгоритмом, если лицо не найдено, трек прерывается, если найдено, оно рассматривается в совокупности с предыдущим изображением лица в том же треке.

Блок 4. Оценка контрастности и резкости: по результатам этой оценки изображение может быть исключено из обработки, если оно недостаточно резкое или четкое.

Блок 5. Компенсация освещения: изображение приводится к световому диапазону, в котором находятся изображения в базе данных.

Блок 6. Распознавание лица: нормированное по освещению изображение лица передается на анализ алгоритму, который извлекает из него биовектор признаков и сравнивает его с кластерами биовекторов, сформированных из базы данных лиц.

Полученные данные: описанная технология возвращает набор изображений лица в различных ракурсах, а также выбирает кадры из видеопотока и нормирует их по освещению.

Для некоторых блоков технологии, предложенных в рамках данной работы, требуется более детальное описание. На рисунке 28 представлена схема алгоритма формирования индивидуальной 3D модели лица.

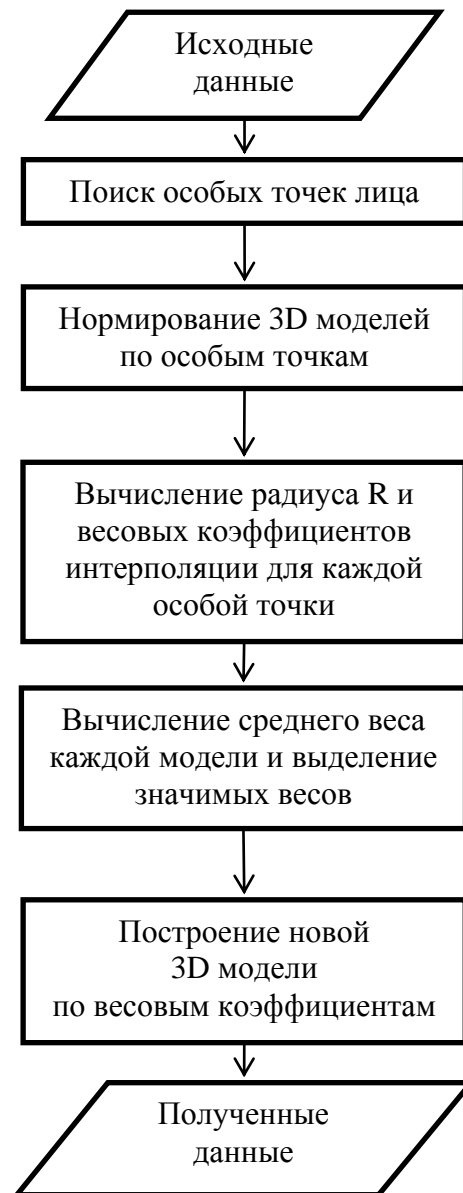


Рис. 28. Схема алгоритма формирования индивидуальной 3D модели лица

Исходные данные: фронтальное изображение лица, набор из N известных 3D моделей лиц.

Блок 1. Поиск особых точек лица: особые точки лица на изображении и для известных 3D моделей ищутся с помощью модификации метода ASM, используемой в библиотеке Dlib.

Блок 2. Нормирование 3D моделей по особым точкам: для каждой 3D модели вычисляется коэффициент масштабирования исходя из расстояния между особыми точками-центрами глаз и точка-начало координат.

Блок 3. Вычисление радиуса и весовых коэффициентов интерполяции для каждой особой точки: задается радиус интерполяции для каждой особой точки как число, принадлежащее диапазону

$$R \in [\min Dist; \max Dist],$$

где $\min Dist$ – минимальное, а $\max Dist$ – максимальное расстояние между особой точкой изображения и одной из 3D моделей. Вычисляются весовые коэффициенты по формуле интерполяции Шепарда.

Блок 4. Вычисление среднего веса каждой модели и выделение значимых весов: вычисляется средний для каждой 3D модели вес по всем особым точкам. Задается уровень, по которому влияние 3D модели считается значимым при формировании новой. Все веса ниже уровня обнуляются.

Блок 5. Построение новой 3D модели по весовым коэффициентам: точки новой 3D модели вычисляются как комбинация точек известных 3D моделей с соответствующими весовыми коэффициентами.

Полученные данные: 3D модель, соответствующая исходному фронтальному изображению.

На рисунке 29 представлена блок-схема алгоритма формирования базы изображений лиц.

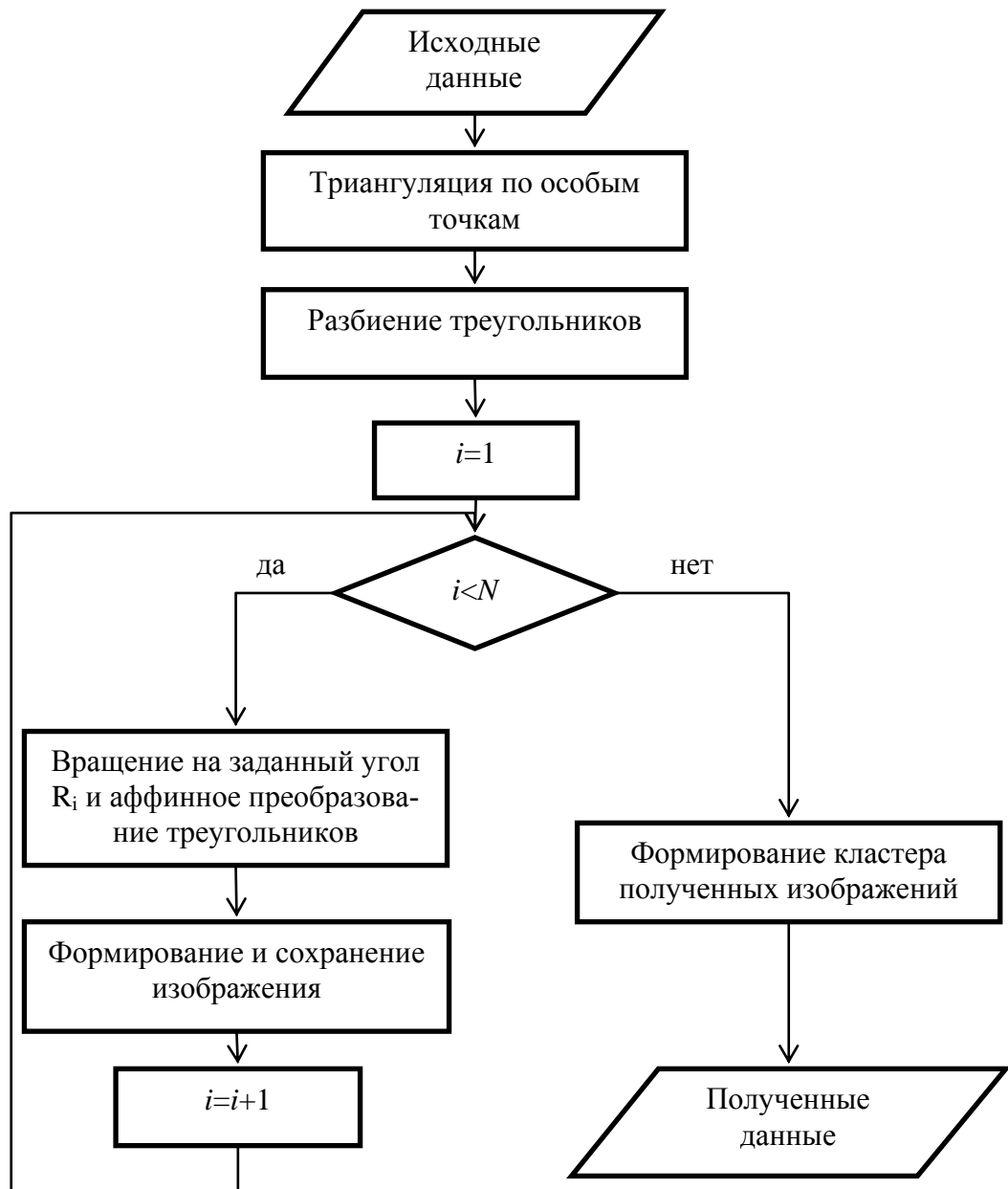


Рис. 29. Схема алгоритма формирования базы данных лиц

Исходные данные: 3D модель лица с известными особыми точками, фронтальное изображение лица, N – число различных вариантов поворота, R – множество значений углов поворота.

Блок 1. Триангуляция по особым точкам: разбиение фронтального изображения лица на треугольники. В результате этой операции будет получен набор треугольников, описывающий область лица на изображении.

Блок 2. Разбиение треугольников: дробление треугольников по выбранному критерию, позволяющему снизить размеры используемых треугольников и приблизить используемую треугольную сетку к равномерной.

Блок 3. Вращение на заданный угол R_i и аффинное преобразование треугольников: каждый из треугольников с помощью операций вращения относительно заданного центра и аффинного преобразования смещается и деформируется в соответствии с величиной угла R_i , заданной условием.

Блок 4. Формирование и сохранение изображения: изображение лица, полученное после преобразований каждого из треугольников, сохраняется как растровое для дальнейшего использования в формировании базы данных.

Блок 5. Формирование кластера полученных изображений: изображения объединяются в единый массив, соответствующий конкретному человеку и использующийся для извлечения биовекторов в процессе обучения системы распознавания.

Полученные данные: набор изображений лица человека с разными ракурсами.

На рисунке 30 представлена схема алгоритма трекинга лиц.

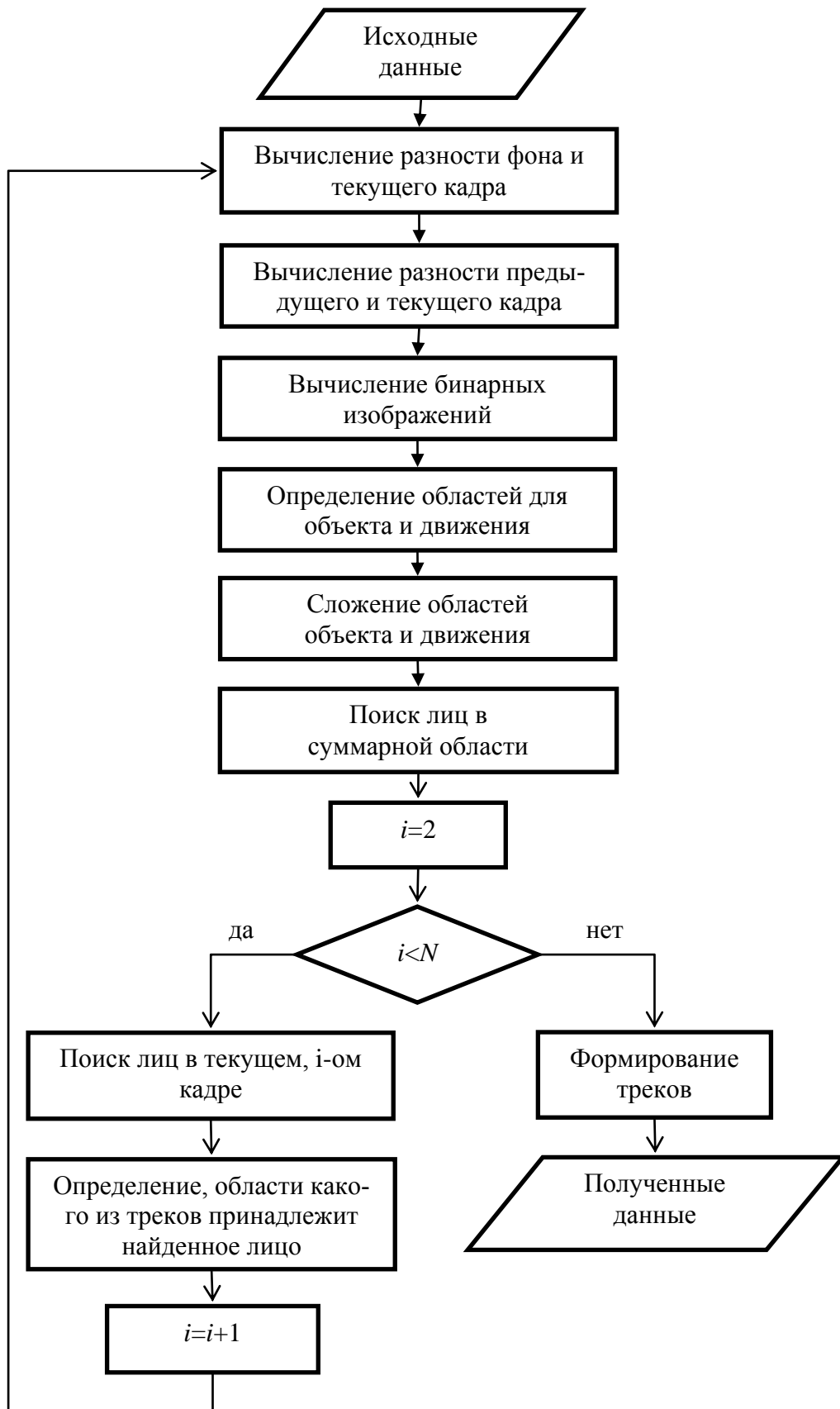


Рис. 30. Схема алгоритма трекинга лиц

Исходные данные: набор кадров видеопотока, N – число кадров.

Блок 1. Вычисление разности фона и текущего кадра $f_T=f_i-f_0$.

Блок 2. Вычисление разности предыдущего и текущего кадра $f_M=f_i-f_{i-1}$.

Блок 3. Вычисление бинарных изображений $B(f_T)$ и $B(f_M)$ с заданным порогом, зависящим от шумов на изображении.

Блок 4. Определение областей для объекта и движения скользящим окном: $B(f_T)>C$, $B(f_M)>C$, где C – заданный порог по количеству белых пикселей в скользящем окне.

Блок 5. Сложение определенных областей бинарных изображений $B(f_T)$ и $B(f_M)$.

Блок 6. Поиск лиц в определенной области на исходном изображении методом Виолы-Джонса, все найденные лица формируют начальные треки Tr_k .

Блок 7. Цикл по кадрам видеопотока.

Блок 8. Если лицо на предыдущем кадре найдено, проводится поиск лиц на новом кадре.

Блок 9. Для каждого нового лица проводится проверка, принадлежит ли центр найденного прямоугольника области прямоугольника лица на предыдущем кадре. Если это так, новое лицо добавляется треку, если нет, формируется новый трек.

Блок 10. Все найденные треки передаются для дальнейшей обработки.

Полученные данные: набор треков лиц и отдельных изображений, не входящих в треки (единичное распознавание).

На рисунке 26 представлена схема алгоритма сравнения изображений лиц, которой можно дополнить предложенный алгоритм трекинга в случае анализа медленно движущихся изображений лиц.

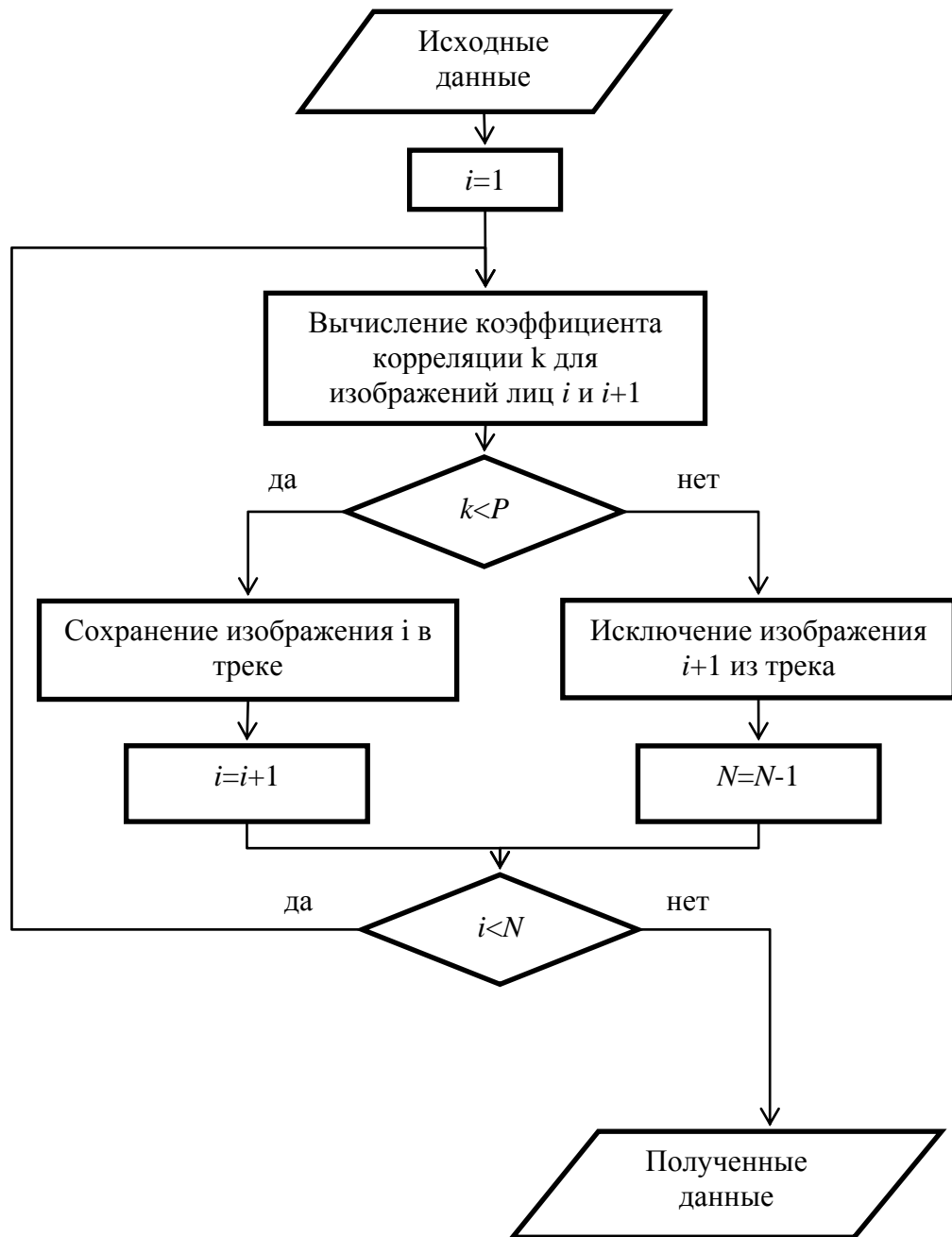


Рис. 31. Схема алгоритма сравнения изображений лиц

Исходные данные: трек изображений лица, N – число изображений, P – пороговый коэффициент корреляции.

Блок 1. Для первых двух лиц вычисляется коэффициент взаимной корреляции.

Блок 2. Если вычисленный коэффициент k больше, чем заданный пороговый коэффициент P , определяющий степень схожести изображений лица, то второе сравниваемое изображение исключается из трека.

Блок 3. Если вычисленный коэффициент k меньше, то первое изображение сохраняется в треке, а второе сравнивается со следующим.

Блок 4. Цикл продолжается до тех пор, пока в треке не останутся только различные изображения лица, либо одно изображение.

Полученные данные: трек различных изображений одного лица.

2.4. Программная реализация алгоритмов.

Разработанные алгоритмы были внедрены в программный комплекс, реализующий захват видеопотока с цифровой камеры, обработку кадров, распознавание личности по изображению лица и трекинг объектов. В рамках данного программного комплекса было проведено тестирование как на базах изображений лиц, так и на видеофайлах.

Созданный программный комплекс состоит из нескольких модулей, осуществляющих основные функции и алгоритмы, применяемые при распознавании: модуль настройки, сохранения и загрузки конфигураций (рисунок 32), модуль запуска и отображения результатов распознавания в видеопотоке (рисунок 33), модуль формирования биометрических векторов из базы изображений лиц (рисунок 34). Каждый из модулей имеет пользовательский интерфейс, написанный с использованием конструктора форм QT Creator.

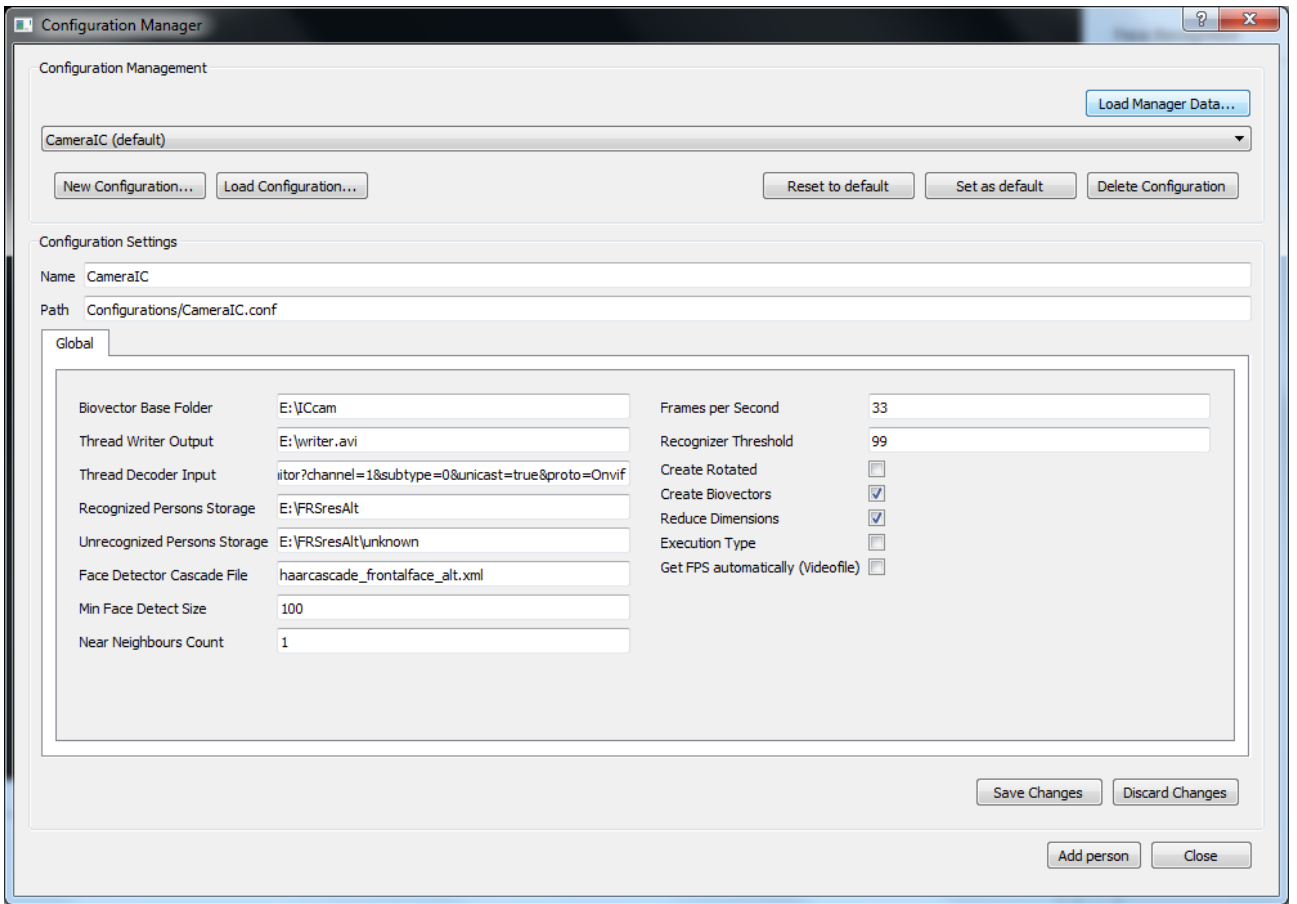


Рис. 32. Модуль менеджера конфигураций

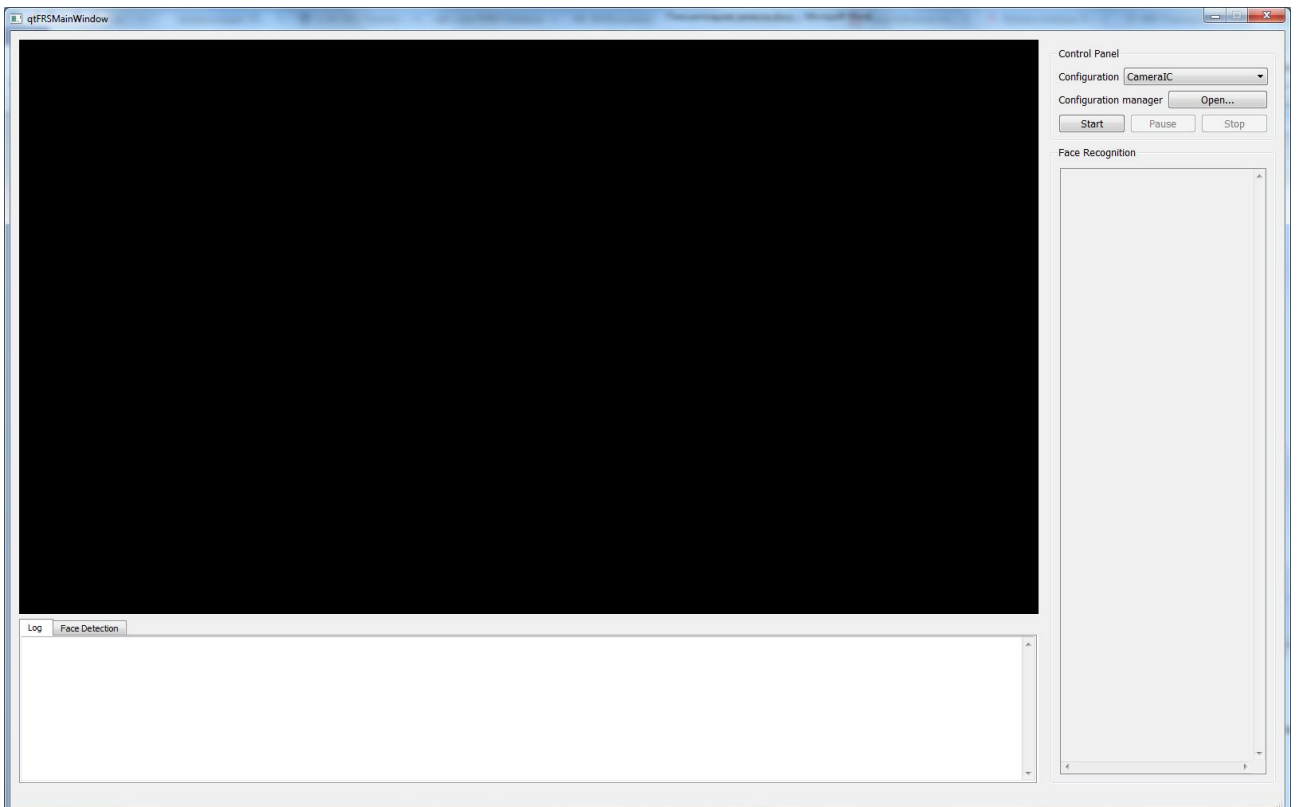


Рис. 33. Модуль основного окна распознавания

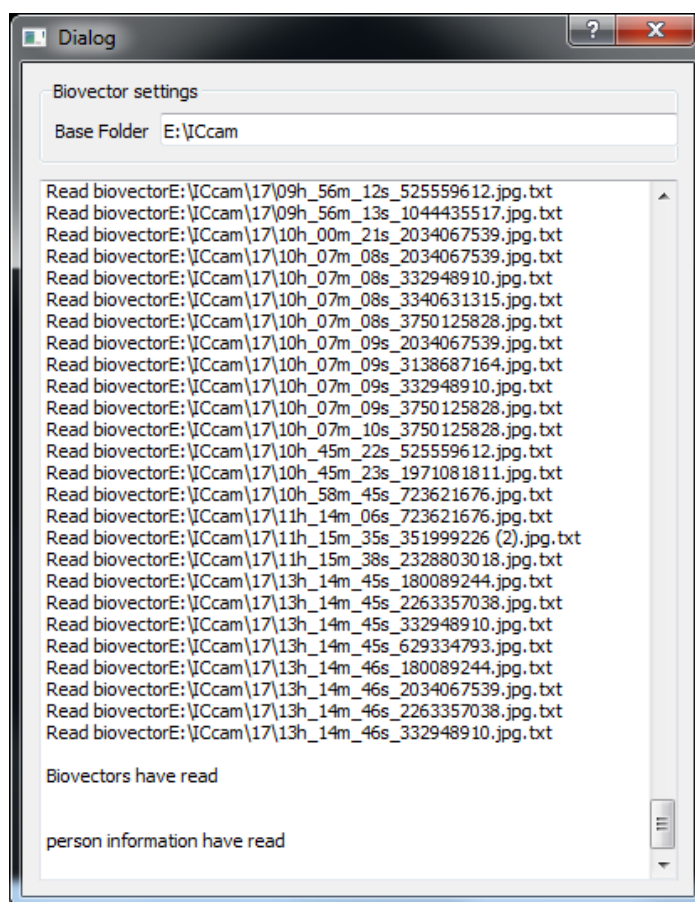


Рис. 34. Модуль формирования базы изображений и биометрических векторов

Применение всех разработанных алгоритмов регулируется посредством модуля настройки конфигураций, после чего становится возможным запуск распознавания лиц в реальном времени как на видео, так и с произвольной камеры, подключенной к персональному компьютеру. Работа в режиме реального времени обеспечивается благодаря параллелизации алгоритма обработки кадров видео с использованием функций современных процессоров.

Для выбора алгоритма вычисления биометрических векторов было проведено тестирование нескольких существующих методов совместно с методами нормирования освещения, в результате которого был выбран алгоритм вычисления стандартного отклонения (STD). В качестве алгоритма снижения размерности биометрических векторов был выбран метод главных компонент (Principal Component Analysis, PCA). В качестве основного алгоритма классификации выбран метод *k* ближайших соседей (*k* Nearest Neighbours, *k*NN).

2.4. Выводы по главе 2.

Предложен алгоритм трекинга лиц в видеопотоке, основанный на методе вычитания фона background subtraction и детекторе лиц Виолы-Джонса. Выбраны методы оценки резкости, контрастности, сравнения изображений, которые подходят для применения в анализе лиц на видео.

Предложена модификация метода построения 3D модели лица с использованием интерполяции Шепарда применительно к обобщенной модели и деформируемой модели.

Предложен алгоритм формирования базы изображений лиц для методов распознавания с использованием 3D моделей лиц. Предложена модификация технологии подготовки изображений лиц в видеопотоке к распознаванию и её алгоритм.

Для всех предложенных алгоритмов и технологии подготовки изображений лиц представлены схемы, алгоритмы внедрены в программный комплекс, реализующий распознавание личности по изображению лица в видеопотоке в режиме реального времени с помощью метода классификации изображений k Nearest Neighbours.

Глава 3. Тестирование алгоритмов и результаты экспериментов

3.1. Подготовка экспериментов.

Предложенная комплексная технология предварительной обработки изображений лиц для подготовки к распознаванию в режиме реального времени, а также входящие в нее алгоритмы компенсации условий съемки (ракурса и освещения) и трекинга требуют экспериментальной проверки эффективности с точки зрения их применимости к задачам распознавания личности по изображению лица, а также сравнения с существующими алгоритмами [116].

Для реализации эксперимента, подтверждающего эффективность разработанного алгоритма, необходимо осуществить выполнение алгоритма распознавания с подключением методов предварительной обработки изображений на базах данных изображений лиц, находящихся в свободном доступе и широко использующихся для тестирования и сравнения методов идентификации личности. В качестве таких баз данных выбраны базы Caltech [117] и FERET [118]. Также в качестве экспериментальных данных получены изображения лиц на реальных видеозаписях и фотографиях.

База данных FERET содержит 14051 8-битное изображение человеческих лиц в оттенках серого с различными ракурсами.

База данных Caltech содержит 450 цветных фронтальных изображений лиц 27 людей с различным освещением, фоном, выражением лица.

Реальные видеозаписи были получены на IP-камере видеонаблюдения RVi-IPC21WDN с объективом вариофокальным мегапиксельным с ИК-коррекцией RVi-1240AIR, а также с помощью веб-камеры Microsoft "LifeCam Studio For Business" 5WH-00002.

Эксперимент проведен в несколько этапов:

1. Деформируемая модель лица: визуальная проверка различимости получаемых трехмерных моделей для лиц с различной геометрией лица.

2. Формирование базы данных лиц: корректность триангуляции и операций переноса изображения.

3. Предварительная обработка изображений лиц:

- а) Оценка эффективности распознавания с фильтрацией изображений по контрастности и резкости;
- б) Оценка эффективности распознавания с выравниванием освещения в кадре различными методами;
- в) Сравнение числа анализируемых объектов при формировании треков и при прямой обработке;
- г) Сравнение эффективности распознавания при различных способах формирования базы лиц (одно изображение, набор изображений, изображения, повернутые с индивидуальной деформируемой моделью).

3.2. Тестирование программной реализации алгоритмов.

База для тестирования алгоритмов

Первоначальное тестирование работоспособности и эффективности методов и алгоритмов проведено на записанных отрезках видео, всего записано 263 коротких ролика, в записи которых участвовало 12 человек (рис. 35).

Всего на данных видео с помощью стандартного детектора лиц было найдено 21735 изображений лиц, включая изображения с недостаточной резкостью, эффектом смазанного изображения и с различными ракурсами относительно оптической оси камеры.



Рис. 35. Пример фронтальных изображений лиц из видеозаписей

Алгоритм построение набора изображений лиц

Для каждого из 12 фронтальных лиц вычислена деформируемая трехмерная модель лица согласно алгоритму, представленному на рисунке 28. С помощью этих моделей получены наборы изображений лиц, повернутых на угол от -20 до +20 градусов по горизонтали относительно оптической оси камеры и от -15 до +15 градусов по вертикали с шагом 5 градусов по каждой оси согласно алгоритму, представленному на рисунке 29. В результате были получены наборы из 15 изображений каждого лица. Примеры сформированных изображений представлены на рисунке 36.



Рис. 36. Пример изображений, полученных с помощью деформируемых моделей лиц

Изображения, представленные на рисунке 36, подтверждают, что деформируемая модель лица корректно сопоставляет точки трехмерной модели точкам плоского растрового изображения и путем аффинных преобразований позволяет получить визуально верные изображения лиц в различных ракурсах.

Для сравнения эффективности распознавания с помощью метода ближайших соседей сделана выборка изображений лиц с разными ракурсами

напрямую из видеопоследовательностей по соотношению 15 изображений на человека.

Также сравнение проводилось с базой, составленной из предположения, что существует лишь одно изображение каждого человека.

Трекинг

Для первоначального сравнительного тестирования, было взято два варианта трекинга: трекинг только с помощью метода Виолы-Джонса и трекинг по предложенному алгоритму с использованием вычитания фона и движения в кадре.

Тестирование проводилось на следующей конфигурации компьютера: процессор Intel Core i7-3770 3,4 GHz, 16 Гб RAM, Windows 7 64-bit.

Так как тестирование алгоритма трекинга возможно только на видео, базы Caltech и FERET в тестировании участвовать не могли.

Собственная база видео состоит из 263 видеопоследовательностей в формате 1280*720 (HD), на каждой из которых присутствует один человек, совершающий разнообразные движения и поворачивающийся относительно оптической оси камеры на углы до 90 градусов. Всего в кадрах детектором лиц на основе метода Виолы-Джонса в реализации библиотеки openCV было обнаружено 21735 изображений лиц.

Тестирование алгоритма, использующего только метод Виолы-Джонса, на собственной базе видео дало следующие результаты: из изначально найденных 21735 изображений лица удалось составить 1846 треков, среднее время обработки одного кадра 16,75 мс.

Алгоритм, использующий предварительный анализ фона и движения, позволил снизить среднее время обработки одного кадра до 2,36 мс, в основном за счет того, что кадры, не содержащие отличий от фона и движущихся объектов, не допускаются до обработки детектором лиц. При этом время обработки одного кадра, содержащего движущиеся объекты, снизилось незначительно – в среднем 12,14 мс. Общее число треков также стало меньше – 1834 – благодаря

тому, что в результате сужения области анализа часть ложных срабатываний детектора лиц была исключена из последующей обработки.

Средняя длина каждого трека составляет 11 кадров, это позволяет исключать из обработки часть кадров, например те, которые обладают недостаточной резкостью или контрастностью, но при этом с высокой вероятностью принадлежат одному человеку.

Алгоритм ускорения поиска лиц применим для ситуаций, в которых ведется стационарная съемка с простым статическим фоном, в случаях же динамического фона этот алгоритм может привести к падению производительности системы на время предварительной обработки с помощью вычитания фона 2-8 мс.

Время обработки каждого кадра зависит от параметров детектора области лица на изображении, в тестовых примерах порядка 10-15 мс, и не требует дополнительных вычислений после обнаружения лица в кадре. Предложенный алгоритм ускоренного трекинга объектов на статичном фоне не мешает применять иные методы повышения точности слежения за объектами на видео, в том числе и те, которые обеспечивают более высокую точность слежения при существенном снижении скорости [119]. Таким образом, скорость предложенного алгоритма трекинга для видео со статическим фоном на 27% выше по сравнению с трекингом на основе выделения области лица, не использующим предварительный анализ фона и движения.

При съемке в реальных условиях возможно снижение средней длины трека, но общая тенденция сохранится, и трекинг как элемент предварительной обработки изображений является состоятельным и эффективным методом, позволяя не обрабатывать повторно лица уже распознанных в видеопоследовательности людей.

Фильтр по резкости и контрастности изображений лиц

Все тестируемые наборы изображений лиц обладают хорошей резкостью и контрастностью, но всё же позволяют провести сравнение влияния отсечения

изображений по нижней границе оценки этих параметров на эффективность распознавания.

Тем не менее, на тестовых наборах не удалось получить какого либо заметного улучшения результатов распознавания с помощью отсеивания изображений по критериям контрастности и резкости. Это связано с незначительным отличием качества изображений в тестовых наборах. При установке порога отсеивания изображений по контрастности и резкости FAR и FRR сохранялись пропорционально общему числу изображений, прошедших отбор. Вероятно, применение этих методов в предложенном виде не подходит для изображений высокого разрешения и высокого качества.

Алгоритм формирования индивидуальной трехмерной модели лица

Для испытания корректности подхода для формирования изображений лица с разными ракурсами была использована трехмерная модель, полученная модифицированным методом Шепарда.

Из одного фронтального изображения (рис. 37) с помощью алгоритма был получен набор изображений лица, повернутого на угол от -20 до +20 градусов относительно оптической оси камеры (рис. 38).



Рис. 37. Исходное фронтальное изображение лица



Рис. 38. Слева – изображения лица, полученные поворотом вдоль вертикальной оси на 20 градусов, справа – вдоль горизонтальной оси

Полученные изображения, использованные для формирования кластера биовекторов в системе идентификации личности, позволили ограничиться использованием одной фотографии человека без снижения эффективности распознавания.

Выравнивание освещения и методы извлечения признаков

Эффективность алгоритмов распознавания может быть оценена двумя вероятностными характеристиками:

1. FAR (False Accept Rate) – вероятность ложного обнаружения, система принимает одного человека, не имеющего фотографий в базе изображений системы распознавания, за другого, зарегистрированного в базе («обозначились»);
2. FRR (False Reject Rate) – вероятность пропуска цели, система не регистрирует совпадения по фотографии зарегистрированного в ней пользователя («не узнали»).

В системах распознавания эти вероятности всегда являются взаимнообратными величинами, в различных системах более важной считается либо первая, либо вторая характеристика.

В качестве усредненной оценки эффективности распознавания обычно берется характеристика EER (Equal Error Rate) – равный уровень ошибок FAR и FRR. Алгоритм с меньшей оценкой EER считается более эффективным [120].

Однако такая оценка не всегда отражает реальную эффективность, так как она зависит не только от минимальных или оптимальных значений оценок FAR и FRR, но и от динамики их изменения, области применения алгоритмов. К примеру, в системах, контролирующих вход людей на объекты ограниченного допуска, вероятность пропуска цели не такая критичная характеристика, как вероятность ложного обнаружения. Исходя из этих соображений, примем конечной оценкой эффективности системы идентификации личности уровень FRR при заданном уровне $FAR \leq 1\%$ от выборки.

Описанные методы выравнивания освещения были скомбинированы в различных вариантах. Тестирование проводилось на базе лиц Caltech Faces, содержащей 450 изображений лиц в разных вариациях освещения. Полученные векторы признаков сравнивались с помощью метрики Евклида. В таблице 1 представлены результаты тестирования рассмотренных алгоритмов. Алгоритмы отсортированы в порядке уменьшения параметра FRR. Кроме FRR в таблице приведены оценки времени формирования вектора признаков, а также длины получаемого вектора признаков. Здесь представлены комбинации методов выравнивания освещения вместе с методами извлечения признаков (для вычисления биометрических векторов). Так, Difference of Gaussians (DoG), вейвлеты Хаара (Haar), фильтр Габора (Gabor) и логарифм от него (Log-Gabor) тестировались совместно с вычислением моментов X_u на основе центральных моментов изображения (HU), гистограммами направленных градиентов (HoG), дискретным косинусным преобразованием (DCT) и стандартным отклонением (STD).

Таблица 1. Результаты тестирования алгоритмов выравнивания освещения и извлечения вектора признаков, отсортированные в порядке убывания EER

Методы	FRR, % (при FAR=1%)	Время обработки изображения, с	Длина вектора признаков
DoG + HU	23,548	0,055	175
Log-Gabor + HU	19,098	0,075	175
Haar + HoG	18,594	0,205	25205
Gabor + HU	16,79	0,095	175
Haar + HU	16,638	0,06	175
Log-Gabor + HoG	15,632	0,215	25205
Gabor + HoG	14,557	0,235	25205
Haar + STD	12,294	0,135	4900
DoG + HoG	12,253	0,205	25205
DoG + DCT	11,835	0,135	50410
Gabor + STD	11,068	0,165	4900
Log-Gabor + DCT	11,059	0,155	50410
DFT + HU	10,764	0,06	175
Haar + DCT	9,74	0,145	50410
DFT + DCT	9,736	0,155	50410
Gabor + DCT	9,561	0,095	250
DFT + HoG	8,628	0,215	25205
DFT + STD	7,03	0,135	4900
Log-Gabor + STD	6,926	0,145	4900
DoG + STD	6,375	0,135	4900
DoG + Log-Gabor + STD	6,301	0,145	4900

Из результатов, представленных в таблице 1 можно сделать вывод, что наиболее эффективным в рамках данного исследования оказался алгоритм, основанный на применении фильтра Difference of Gaussians, фильтрации Log-Gabor и извлечении вектора признаков на основе вычисления стандартного от-

клонения. В сравнении с другими алгоритмами, представленными в таблице 1, данный алгоритм имеет приемлемое время работы и длину получаемого вектора признаков. Также следует отметить, что исключение из этого набора фильтра Log-Gabor слабо влияет на точность распознавания, но при этом снижает время обработки на 10 миллисекунд, таким образом, этот подход может быть рассмотрен для использования в системах распознавания в режиме реального времени вместе с более сложной схемой компенсации условий освещения.

3.3. Результаты тестирования комплексной технологии подготовки изображений лиц к распознаванию.

Наиболее эффективные из описанных методов подготовки изображения были скомбинированы в составе комплексной технологии подготовки изображений лиц к распознаванию. На основе этой комбинации проводилось сравнение эффективности предложенного алгоритма компенсации ракурса на различных базах. Первое тестирование проводилось на базе, составленной из видео. Второе тестирование проводилось на базе лиц Caltech Faces. Третье – на базе лиц FERET 2001 года. Полученные векторы признаков сравнивались с помощью метрики Евклида.

Первое тестирование проводилось на базе, составленной из видео (~21 тысяча изображений лиц, 12 уникальных лиц). В таблицах 2-6 и рисунках 39-42 представлены результаты тестирования системы распознавания с базой из 12 фронтальных изображений лиц, с базой из 15 изображений для каждого лица (всего 180) и с базой изображений, созданных при помощи деформируемых моделей лиц.

Таблица 2. Результаты тестирования системы распознавания лиц на видеофайлах, одно изображение лица на человека

Порог алгоритма распознавания, %	Верно распознанные лица	Ложно опознанные лица	Верно распознанные лица, %	FAR, %	FRR, %
25	2385	0	10,973	0	89,027
30	3883	14	17,865	0,064	82,135
35	5491	97	25,263	0,446	74,736
40	7177	424	33,02	1,951	66,979
45	8792	1333	40,451	6,133	59,549
50	10249	2916	47,154	13,416	52,846

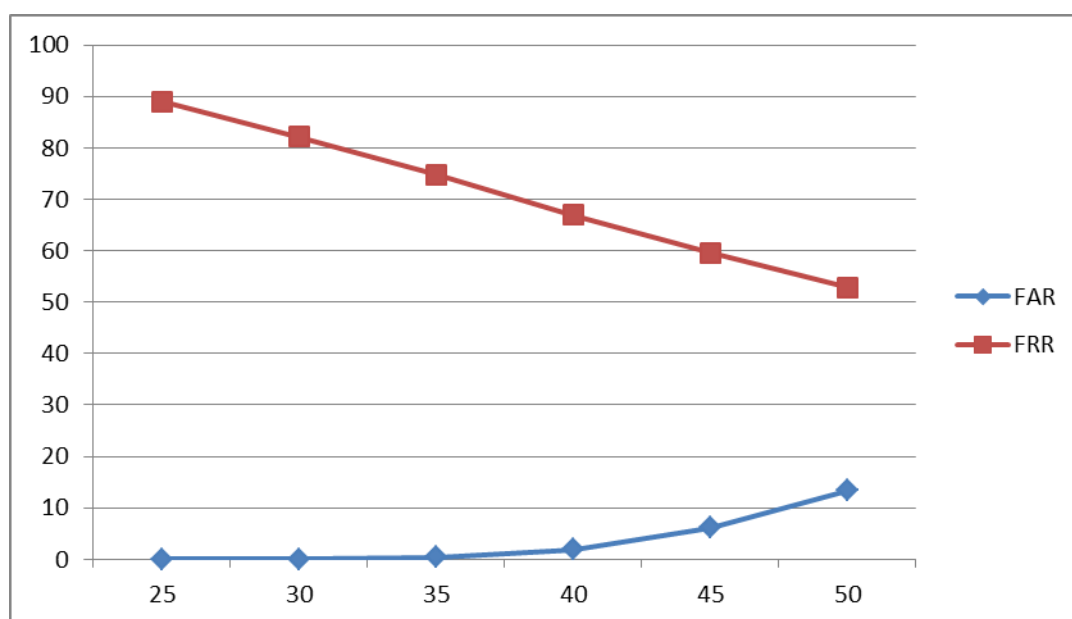


Рис. 39. Ошибки FAR и FRR в зависимости от выбранного порога распознавания, одно изображение лица на человека

Таблица 3. Результаты тестирования системы распознавания лиц на видеофайлах, 15 изображений на человека

Порог алгоритма распознавания, %	Верно распознанные лица	Ложно опознанные лица	Верно распознанные лица, %	FAR, %	FRR, %
50	3959	0	18,215	0	81,785
55	5089	0	23,414	0	76,586
60	6156	18	28,323	0,083	71,677
65	7149	76	32,892	0,35	67,108
70	8066	207	37,111	0,952	62,889
75	8957	445	41,21	2,047	58,79

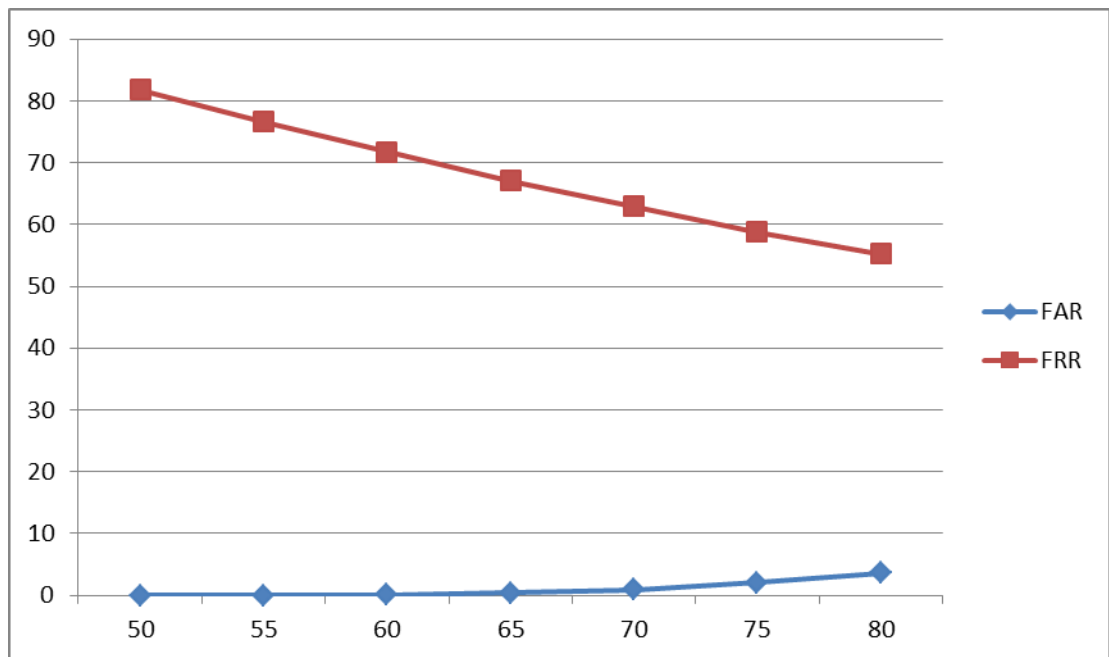


Рис. 40. Ошибки FAR и FRR в зависимости от выбранного порога распознавания, 15 изображений на человека

Таблица 4. Результаты тестирования системы распознавания лиц на видеофайлах, 15 сгенерированных алгоритмом изображений на человека

Порог алгоритма распознавания, %	Верно распознанные лица	Ложно опознанные лица	Верно распознанные лица, %	FAR, %	FRR, %
55	3580	0	16,471	0	83,529
60	4924	1	22,655	0,005	77,345
65	6315	10	29,055	0,046	70,945
70	7632	65	35,114	0,299	64,886
75	8767	156	40,336	0,718	59,664

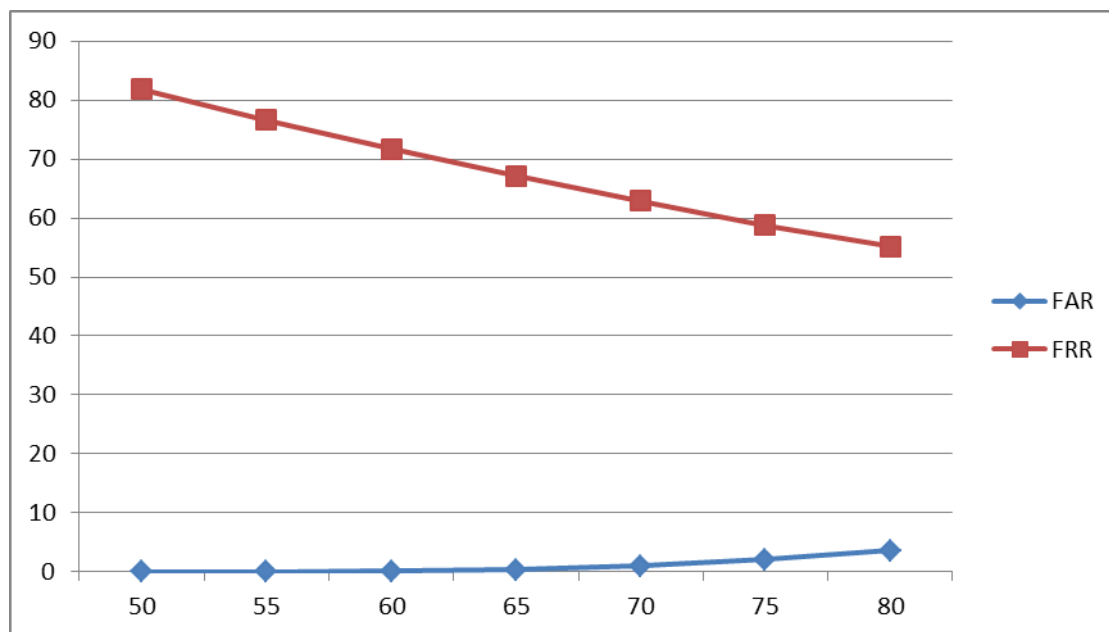


Рис. 41. Ошибки FAR и FRR в зависимости от выбранного порога распознавания, 15 сгенерированных алгоритмом изображений на человека

Полученные при тестировании результаты объединены в виде таблицы взаимозависимости уровней ошибок FAR и FRR, а также в виде графика PR-кривой, которая показывает отношение точности (precision) и полноты (recall). Здесь полнота R – это доля верно распознанных изображений лиц относительно общего числа принятых (верно распознанных и ложно опознанных), выраженная в процентах, а точность P – это доля верно распознанных лиц относительно всех искомых лиц (верно распознанных и ложно отвергнутых), также выраженная в процентах.

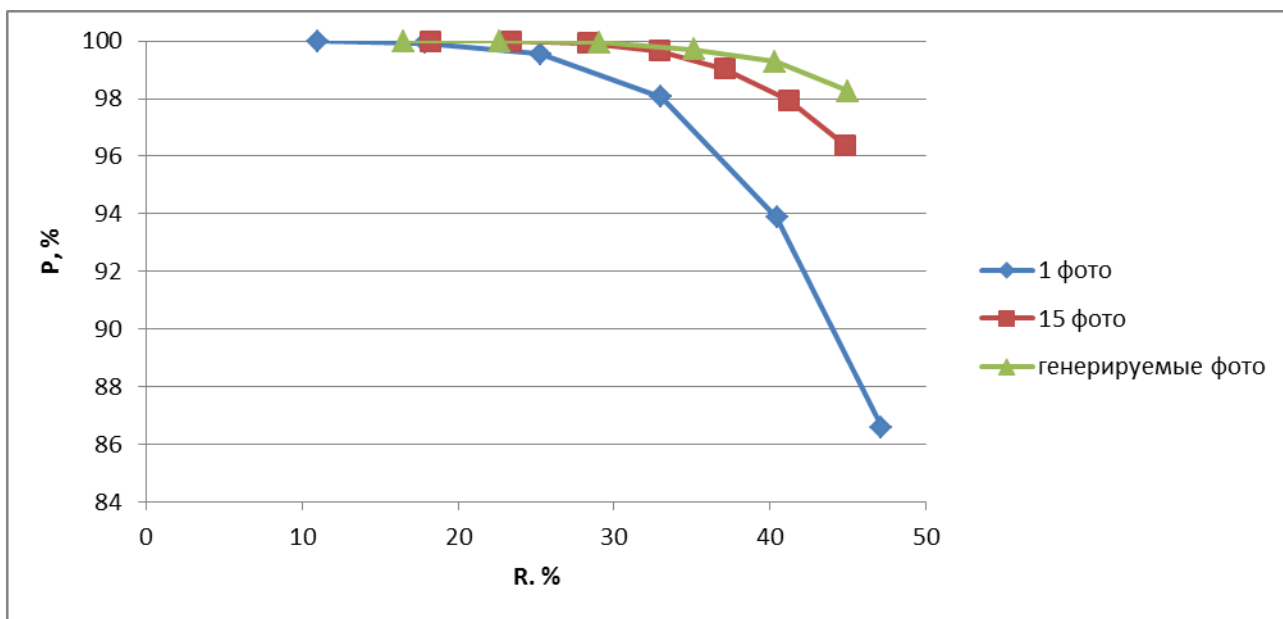


Рис. 42. PR-кривая для тестирования алгоритма на видеофайлах

Сбалансированная F_1 -мера для каждой из баз изображений представлена в таблице 5.

Таблица 5. F_1 -мера тестирования на видеофайлах

Порог распознавания	Одно изображение лица на человека	15 изображений на человека	15 сгенерированных изображений
55	19,776	37,944	28,284
60	30,312	44,135	36,94
65	40,3	49,459	45,022
70	49,403	53,992	51,936
75	56,538	58,013	57,366
80	61,057	61,168	61,737

Таблица 6. Результаты тестирования системы распознавания лиц на видеофайлах, FRR при FAR=1%

База изображений	FRR, % (при FAR=1%)
Одно изображение лица на человека	8,069
15 изображений на человека	4,82
15 сгенерированных алгоритмом изображений на человека	2,179

Как можно видеть из таблиц 5-6, использование сгенерированных изображений в качестве базы изображений лиц для записанных видеофайлов не только не снижает точность распознавания, но и оказывается на 2.6% эффективнее, чем набор из 15 изображений, находящихся в изначальной базе.

Второе тестирование проводилось на базе изображений лиц Caltech Faces [117]. Эта база содержит изображения 26 человек, снятых во фронтальном ракурсе, но с различным качеством и мимикой, общее число изображений 450. Так как число изображений каждого человека в базе разное, при тестировании использовались следующие варианты: 1 изображение на человека, от 3 до 6 изображений на человека и 15 сгенерированных изображений на человека. В таблицах 7-11 и на рисунках 43-46 представлены основные результаты тестирования.

Таблица 7. Результаты тестирования системы распознавания лиц на базе изображений лиц Caltech Faces, одно изображение лица на человека

Порог алгоритма распознавания, %	Верно распознанные лица	Ложно опознанные лица	Верно распознанные лица, %	FAR, %	FRR, %
25	295	0	72,839	0	27,16
30	326	0	80,494	0	19,506
35	354	0	87,407	0	12,593
40	374	1	92,346	0,247	7,654
45	383	2	94,568	0,494	5,432

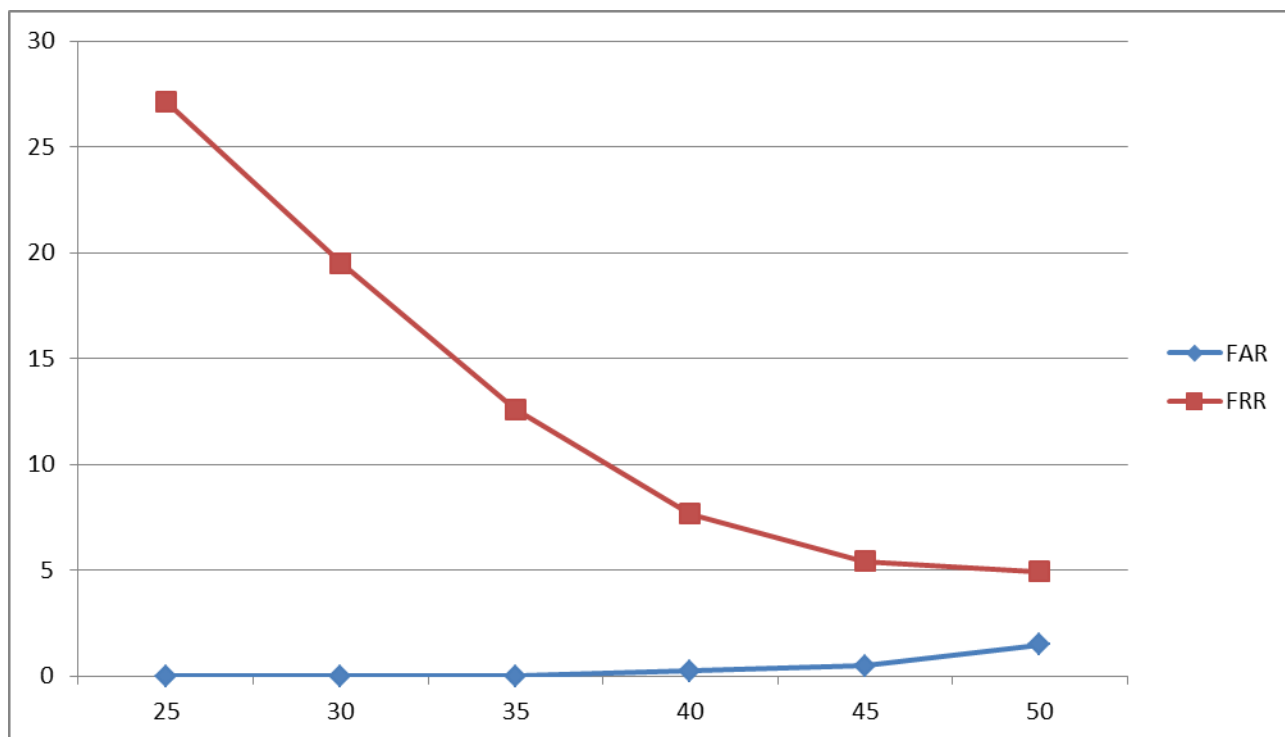


Рис. 43. Ошибки FAR и FRR в зависимости от выбранного порога распознавания, одно изображение лица на человека

Таблица 8. Результаты тестирования системы распознавания лиц на базе изображений лиц Caltech Faces, 3-6 изображений на человека

Порог алгоритма распознавания, %	Верно распознанные лица	Ложно опознанные лица	Верно распознанные лица, %	FAR, %	FRR, %
50	280	0	93,959	0	6,04
55	285	0	95,637	0	4,362
60	287	0	96,309	0	3,691
65	289	0	96,98	0	3,02
70	291	0	97,651	0	2,349
75	291	1	97,651	0,336	2,349
80	291	3	97,651	1,007	2,349

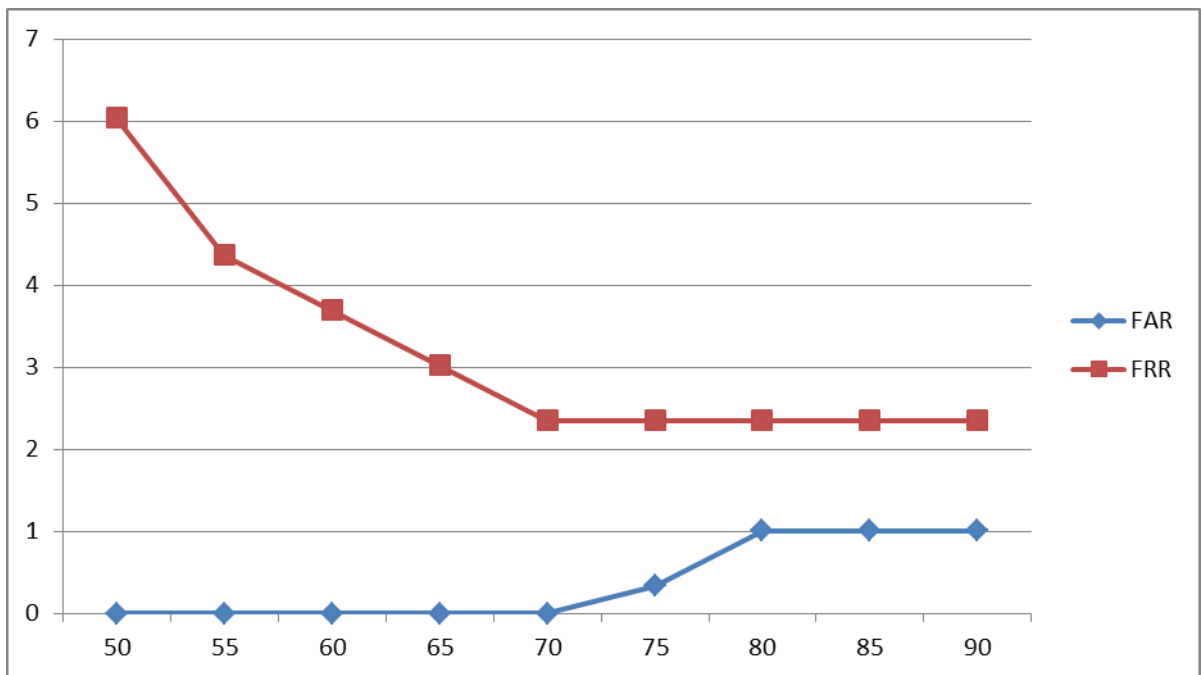


Рис. 44. Ошибки FAR и FRR в зависимости от выбранного порога распознавания, 3-6 изображений на человека

Таблица 9. Результаты тестирования системы распознавания лиц на базе изображений лиц Caltech Faces, 15 сгенерированных алгоритмом изображений на человека

Порог алгоритма распознавания, %	Верно распознанные лица	Ложно опознанные лица	Верно распознанные лица, %	FAR, %	FRR, %
50	320	0	79,012	0	20,988
55	333	0	82,222	0	17,778
60	356	0	87,901	0	12,099
65	365	0	90,123	0	9,876
70	376	0	92,839	0	7,16
75	382	1	94,32	0,247	5,679
80	386	2	95,309	0,494	4,691
85	389	3	96,049	0,741	3,951
95	391	4	96,543	0,988	3,457
99	393	4	97,037	0,988	2,963

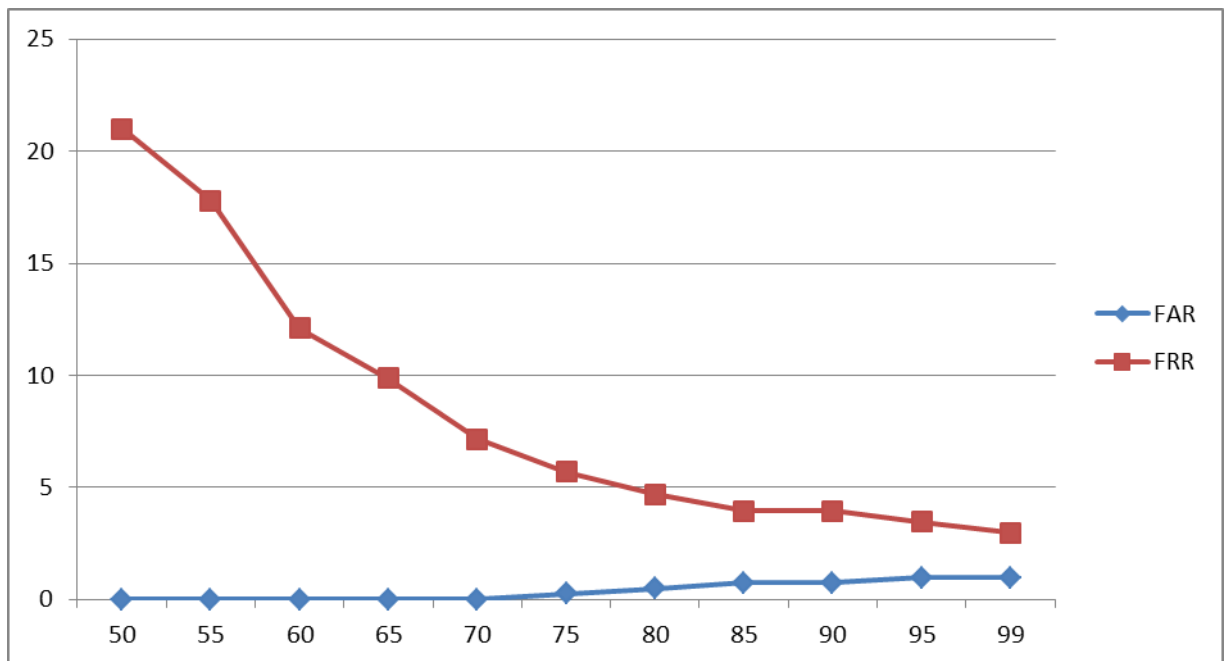


Рис. 45. Ошибки FAR и FRR в зависимости от выбранного порога распознавания, 15 сгенерированных алгоритмом изображений на человека

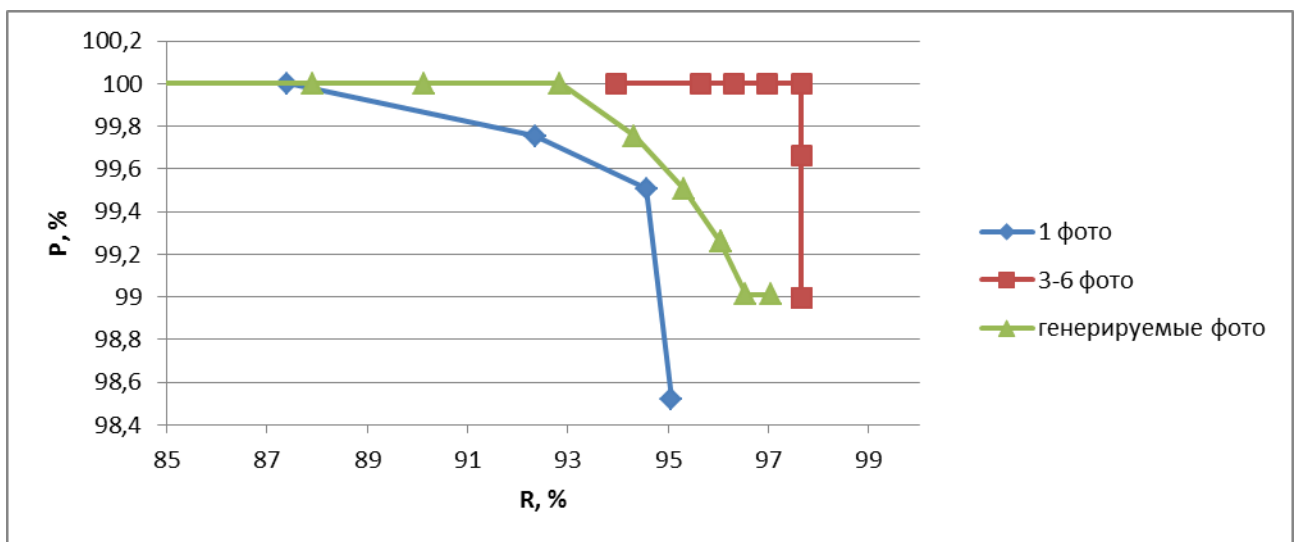


Рис. 46. PR-кривая для тестирования алгоритма на базе Caltech Faces

Таблица 10. F_1 -мера тестирования на базе Caltech Faces

Порог распознавания	Одно изображение лица на человека	15 изображений на человека	15 сгенерированных изображений
75	84,286	98,467	96,961
80	89,193	98,812	97,362
85	93,281	98,647	97,628
90	95,907	98,318	97,628
95	96,974	98,318	97,762
99	96,759	98,318	98,015

Таблица 11. Результаты тестирования системы распознавания лиц на базе изображений лиц Caltech Faces, FRR при FAR=1%

База изображений	FRR, % (при FAR=1%)
Одно изображение лица на человека	5,185
3-6 изображений на человека	2,349
15 сгенерированных алгоритмом изображений на человека	3,457

Как можно видеть из таблиц 10-11, использование сгенерированных изображений оказывается на 1% менее эффективным для базы, состоящей из фронтальных фотографий, чем использование нескольких изображений для каждого человека, но на 1,7% более эффективным, чем использование одного изображения.

Третье тестирование проводилось на базе изображений лиц FERET [118]. Эта база содержит изображения 996 человек, снятых в различных ракурсах (повороты на 0, 45 и 90 градусов по горизонтали относительно оптической оси камеры), с различным качеством, освещением и мимикой, общее число изображений 11297, из них успешно опознаются детектором 5335. Так как число изображений каждого человека в базе разное, при тестировании использовались следующие варианты: 1 изображение на человека, от 2 до 6 изображений на человека и 15 сгенерированных изображений на человека. Для распознавания была составлена база из 30 человек, общее число изображений 181. В таблицах 12-16 и на рисунках 47-50 представлены основные результаты тестирования.

Таблица 12. Результаты тестирования системы распознавания лиц на базе изображений лиц FERET, одно изображение лица на человека

Порог алгоритма распознавания, %	Верно распознанные лица	Ложно опознанные лица	Верно распознанные лица, %	FAR, %	FRR, %
25	58	6	51,381	0	48,619
30	67	26	53,039	0,056	46,961
35	75	216	54,696	0,131	45,304
40	82	865	58,564	0,356	41,436
45	86	1999	61,326	1,818	38,674
50	94	3243	64,641	6,542	35,359
55	99	4206	66,851	15,933	33,149

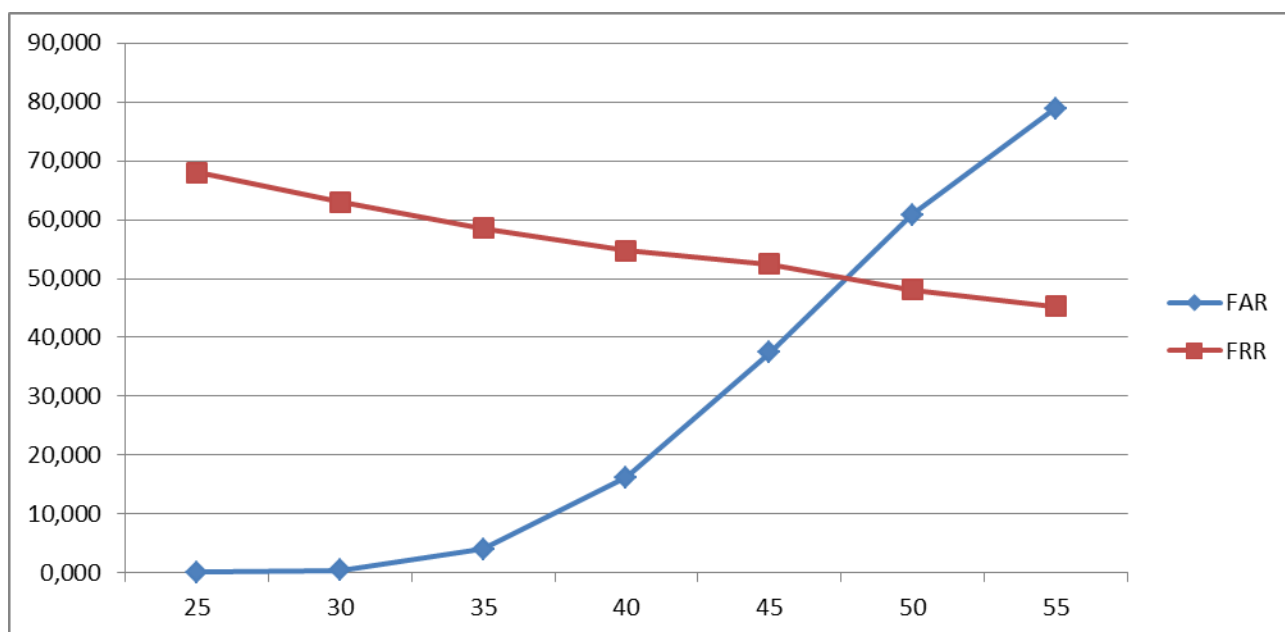


Рис. 47. Ошибки FAR и FRR в зависимости от выбранного порога распознавания, одно изображение лица на человека

Таблица 13. Результаты тестирования системы распознавания лиц на базе изображений лиц FERET, 2-6 изображений на человека

Порог алгоритма распознавания, %	Верно распознанные лица	Ложно опознанные лица	Верно распознанные лица, %	FAR, %	FRR, %
25	93	0	51,381	0	48,619
30	96	3	53,039	0,056	46,961
35	99	7	54,696	0,131	45,304
40	106	19	58,564	0,356	41,436
45	111	97	61,326	1,818	38,674
50	117	349	64,641	6,542	35,359
55	121	850	66,851	15,933	33,149

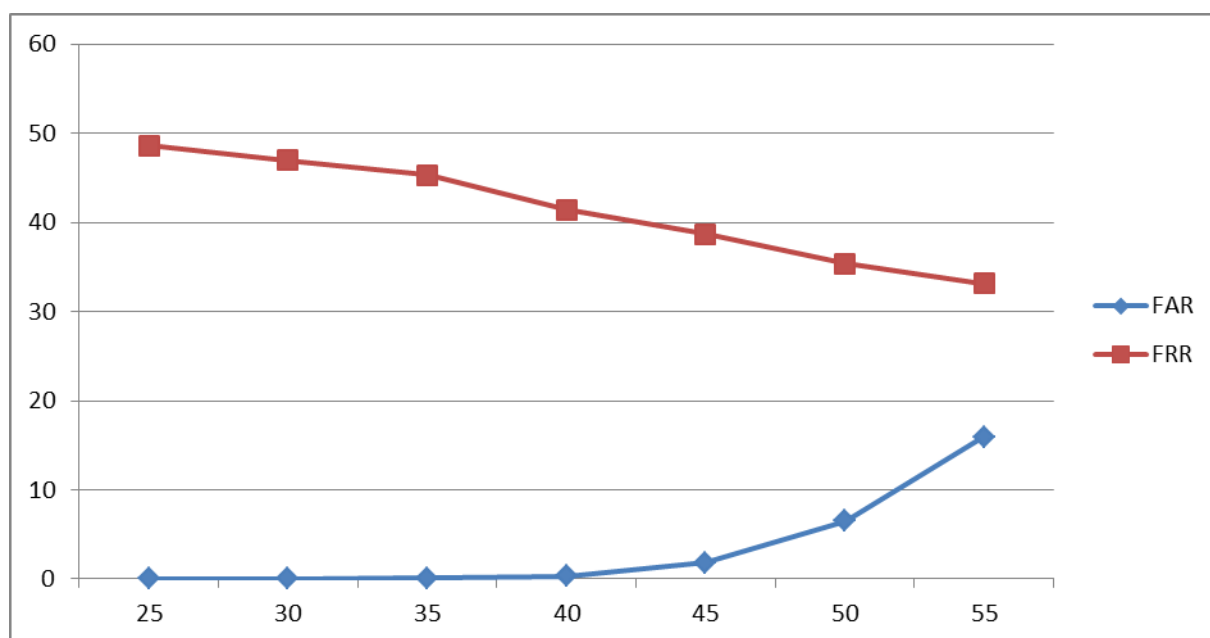


Рис. 48. Ошибки FAR и FRR в зависимости от выбранного порога распознавания, 2-6 изображений на человека

Таблица 14. Результаты тестирования системы распознавания лиц на базе изображений лиц FERET, 15 сгенерированных алгоритмом изображений на человека

Порог алгоритма распознавания, %	Верно распознанные лица	Ложно опознанные лица	Верно распознанные лица, %	FAR, %	FRR, %
50	54	0	29,834	0	70,166
55	59	2	32,597	0,037	67,403
60	65	12	35,912	0,225	64,088
65	71	55	39,227	1,031	60,773
70	72	206	39,779	3,861	60,221
75	80	532	44,199	9,972	55,801
80	86	1023	47,514	19,175	52,486

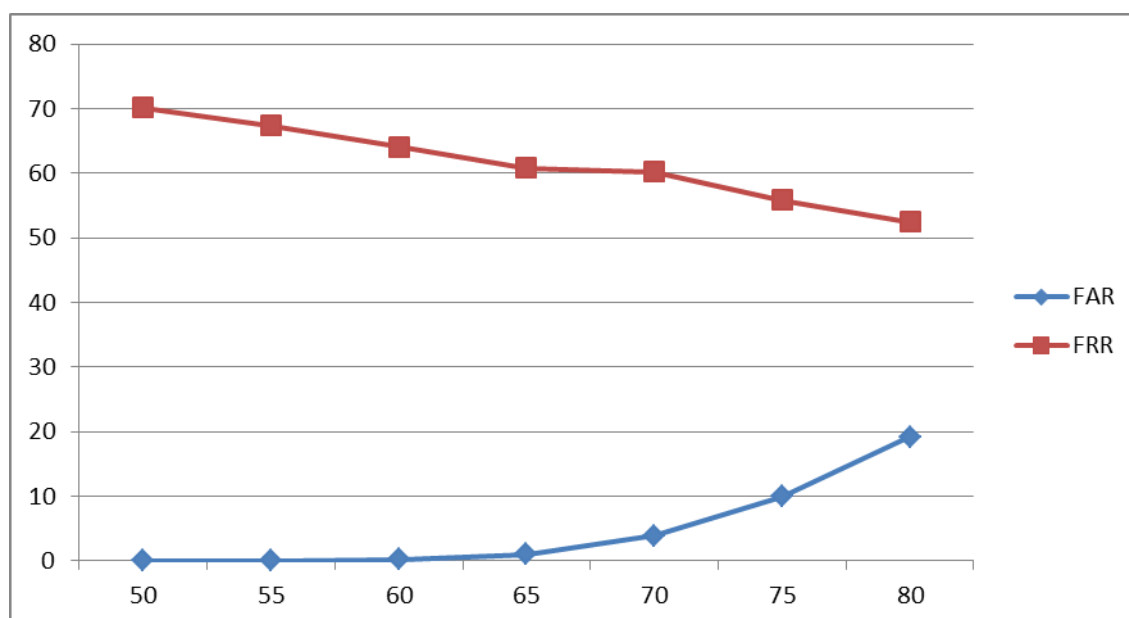


Рис. 49. Ошибки FAR и FRR в зависимости от выбранного порога распознавания, 15 сгенерированных алгоритмом изображений на человека

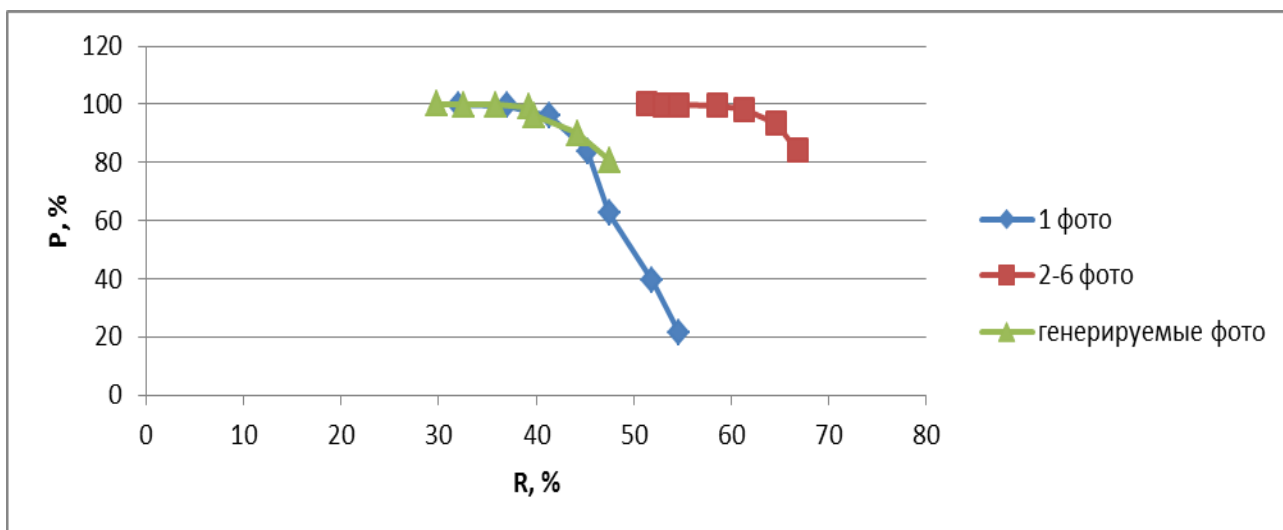


Рис. 50. PR-кривая для тестирования алгоритма на базе FERET

Таблица 15. F_1 -мера тестирования на базе FERET

Порог распознавания	Одно изображение лица на человека	15 изображений на человека	15 сгенерированных изображений
50	48,522	67,883	45,957
55	53,961	69,301	49,162
60	57,878	70,681	52,814
65	58,809	73,77	56,184
70	53,998	75,496	56,274
75	44,685	76,423	59,29
80	30,517	74,477	59,846

Таблица 16. Результаты тестирования системы распознавания лиц на базе изображений лиц FERET, FRR при FAR=1%

База изображений	FRR, % (при FAR=1%)
Одно изображение лица на человека	61,878
2-6 изображений на человека	41,436
15 сгенерированных алгоритмом изображений на человека	61,326

Как можно видеть из таблиц 15-16, использование сгенерированных изображений оказывается на 20% менее эффективным для базы, состоящей из большого числа фотографий, чем использование нескольких изображений для

каждого человека, но всё же на 0,55% более эффективным, чем использование одного изображения.

Полученные векторы признаков сравнивались с помощью метрики Евклида.

Из таблиц 5, 11 и 16 можно сделать вывод, что результаты тестирования по выбранной метрике подтверждают эффективность предложенного подхода: использование алгоритма расширения набора биометрических векторов на собственной базе изображений повышает точность распознавания по сравнению с подходом, когда для формирования биометрических векторов берется набор изображений из самой базы. На базах Caltech Faces и FERET тестирование показало, что набор синтезированных изображений не позволяет достичь точности распознавания с набором естественных изображений, но повышает точность распознавания по сравнению с одиночными изображениями.

Это позволяет утверждать, что использование синтетических изображений в разных ракурсах, полученных с помощью DFM, согласно выбранным критериям оценки эффективности, повышает точность распознавания личности по изображению лица на 0,5-4% в случае для одного изображения лица.

3.4. Сравнение результатов тестирования предложенной технологии с существующими системами распознавания

Кроме сравнения эффективности предложенной технологии и авторских алгоритмов компенсации ракурса с помощью метода кластеризации проведено сравнение результатов, полученных с помощью предложенной технологии подготовки изображений, с мировыми аналогами систем распознавания на базе изображений FERET (информация с сайта Ayonix.com) [121]. На рисунке 51 приведен график характеристик FAR и FRR алгоритма распознавания лиц из программного комплекса Ayonix SDK, а на рисунке 52 – аналогичные результаты распознавания для метода кластеризации kNN с использованием предложенной технологии предварительной подготовки изображений лиц. В таблице 17 показано сравнение точности и скорости 4 алгоритмов распознавания, характеристики которых приведены на сайте Ayonix.com.

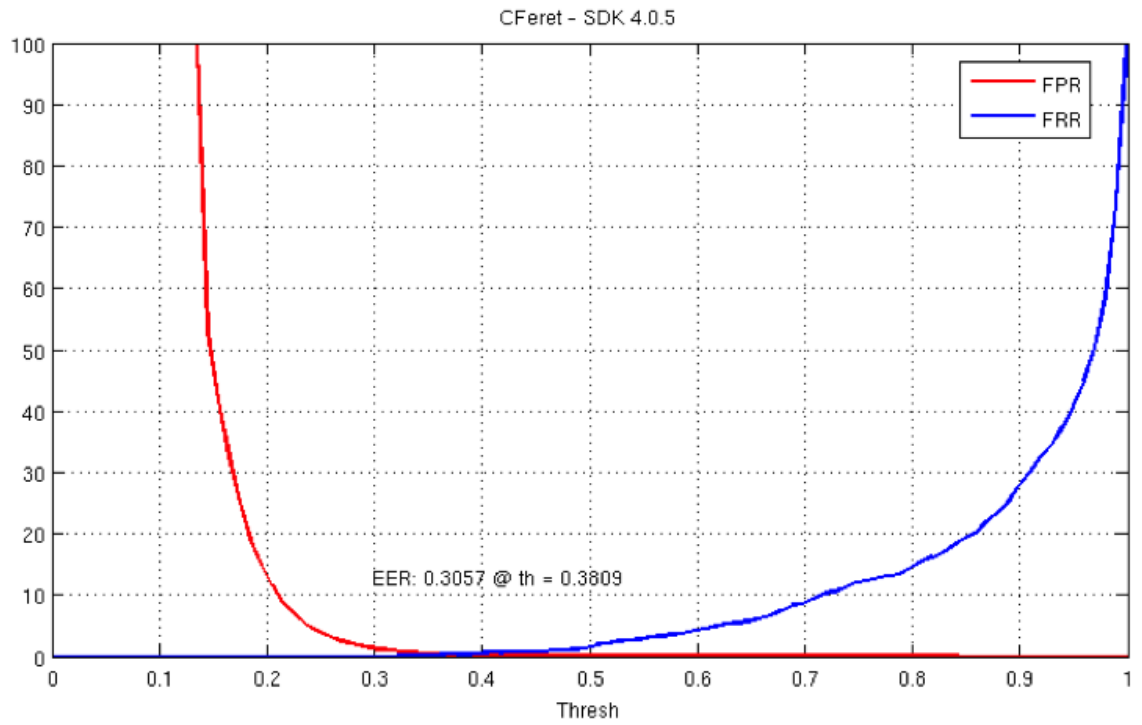


Рис. 51. График FAR и FRR Ayonix SDK

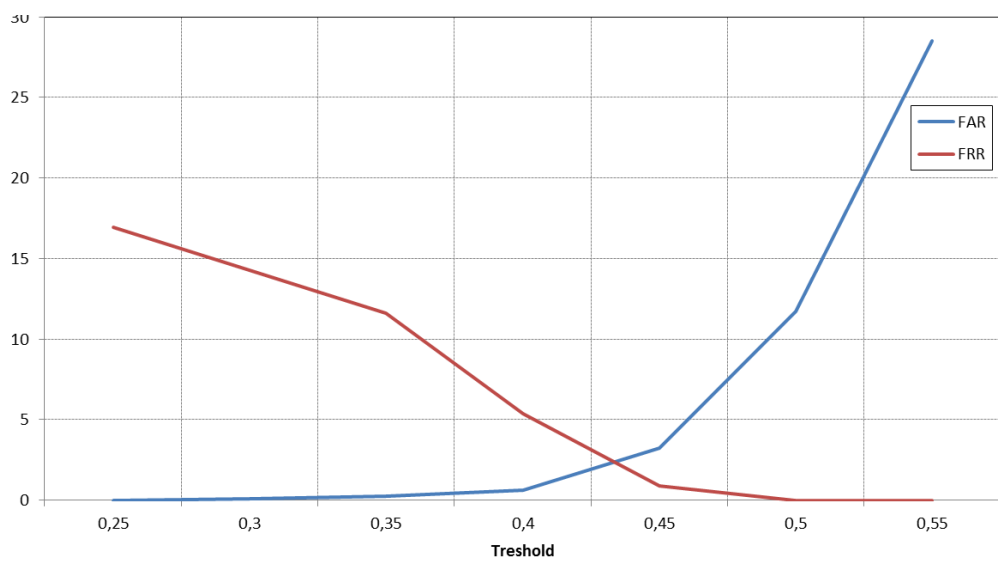


Рис. 52. График FAR и FRR собственной реализации

Таблица 17. Сравнение результатов распознавания существующих систем и метода классификации kNN с использованием предложенной технологии

Критерий	Germany Company	Japan Company	Ayonix v4.0 3D Release	kNN с предложенной технологией
ЕЕР, %	97,9	97,9	99,6	97,4
Скорость поиска, мс	870	1200	468	152

Из таблицы 17 очевидно, что метод классификации kNN с применением предложенной технологии предварительной подготовки изображений уступает в точности ведущим мировым алгоритмам распознавания лиц, но заметно быстрее производит поиск по базе изображений, что подтверждает эффективность применения технологии для задач распознавания лиц в режиме реального времени.

3.5. Выводы по главе 3.

Получены результаты тестирования предложенных алгоритмов и выбранных методов в рамках технологии подготовки изображений лиц к распознаванию.

Предложенный алгоритм трекинга успешно объединяет последовательности изображений лиц в видеопотоке, увеличивая скорость работы детектора лиц Виолы-Джонса более чем на 20%.

Алгоритмы оценки контрастности и резкости не продемонстрировали заметного влияния на алгоритм распознавания, что отчасти связано с качеством тестовых изображений лиц. Для окончательного вывода по этим методам требуется тестирование в сложных условиях съемки.

Результаты экспериментов с алгоритмами выравнивания освещения подтвердили эффективность методов DoG и Log-Gabor в повышении точности распознавания, а также высокую скорость обработки изображений с их помощью.

Алгоритмы построения деформируемой трехмерной модели лица и формирования на их основе изображений лиц в различных ракурсах показали эффективность на тестовых изображениях, а также не оказали негативного эффекта в критических ситуациях, при сравнении изображений лиц с углами поворота 45 градусов и больше.

Сравнение полученных результатов с аналогичными системами распознавания, чьи характеристики представлены в открытом доступе, показало, что предложенный алгоритм предварительной подготовки изображений лиц к распознаванию в комбинации с методом классификации kNN демонстрирует несколько меньшую точность распознавания при значительном увеличении скорости обработки изображений лиц.

В целом можно заключить, что все рассмотренные в рамках технологии обработки изображений лиц алгоритмы и методы эффективны на тестовых данных, следовательно, эта технология может быть применена на реальных видеозаписях и при обработке видеопотока в режиме реального времени для дальнейшего тестирования и анализа результатов.

Глава 4. Апробация и внедрение

4.1. Апробация результатов работы.

На программную реализацию оригинальных авторских алгоритмов формирования индивидуальных трехмерных моделей и набора изображений лиц в различных ракурсах для систем распознавания получено свидетельство о регистрации программы для ЭВМ.

На основе анализа и сравнения методов обработки изображений составлен учебный модуль по основам обработки изображений и распознаванию образов для студентов специальности Б3.В11 «Интеллектуальные и информационные системы», бакалавриат, направления 09.03.03 «Прикладная информатика» кафедры Программной Инженерии ИК ТПУ.

Результаты диссертационной работы обсуждались и докладывались на 2 международных научных конференциях, в том числе на конференции «GraphiCon 2016». По результатам работы опубликовано 11 статей, в том числе 5 статей в журналах, включенных в список ВАК, 2 статьи в журналах, индексируемых Scopus, 1 статья в журнале, индексируемом Web of Science.

4.2. Внедрение результатов работы.

Разработанный программный комплекс внедрен в ООО «Научно-производственная компания «Техника дела»», г. Томск, где используется для формирования входных данных для системы распознавания лиц в виде набора синтезированных изображений лиц в различных ракурсах, а также сравнительного анализа методов обработки цифровых изображений. Получен акт внедрения.

С помощью IP-камеры RVi-IPC21WDN, установленной на входе Института кибернетики Томского политехнического университета на видео фиксируются лица сотрудников и студентов, заходящих в корпус. Этот видеопоток обрабатывается разработанным программным комплексом непрерывно в режиме реального времени. Также на двух видеозаписях общей длительностью около 8 часов, полученных в этих условиях, был проведен тестовый контроль.

Эти видеозаписи сделаны в сложных условиях: ракурс съемки лиц далек от фронтального (камера на высоте 2 метров от пола), освещение меняется в зависимости от времени суток, так как в помещение попадает солнечный свет с улицы через окна и стекло дверей (рисунок 53).

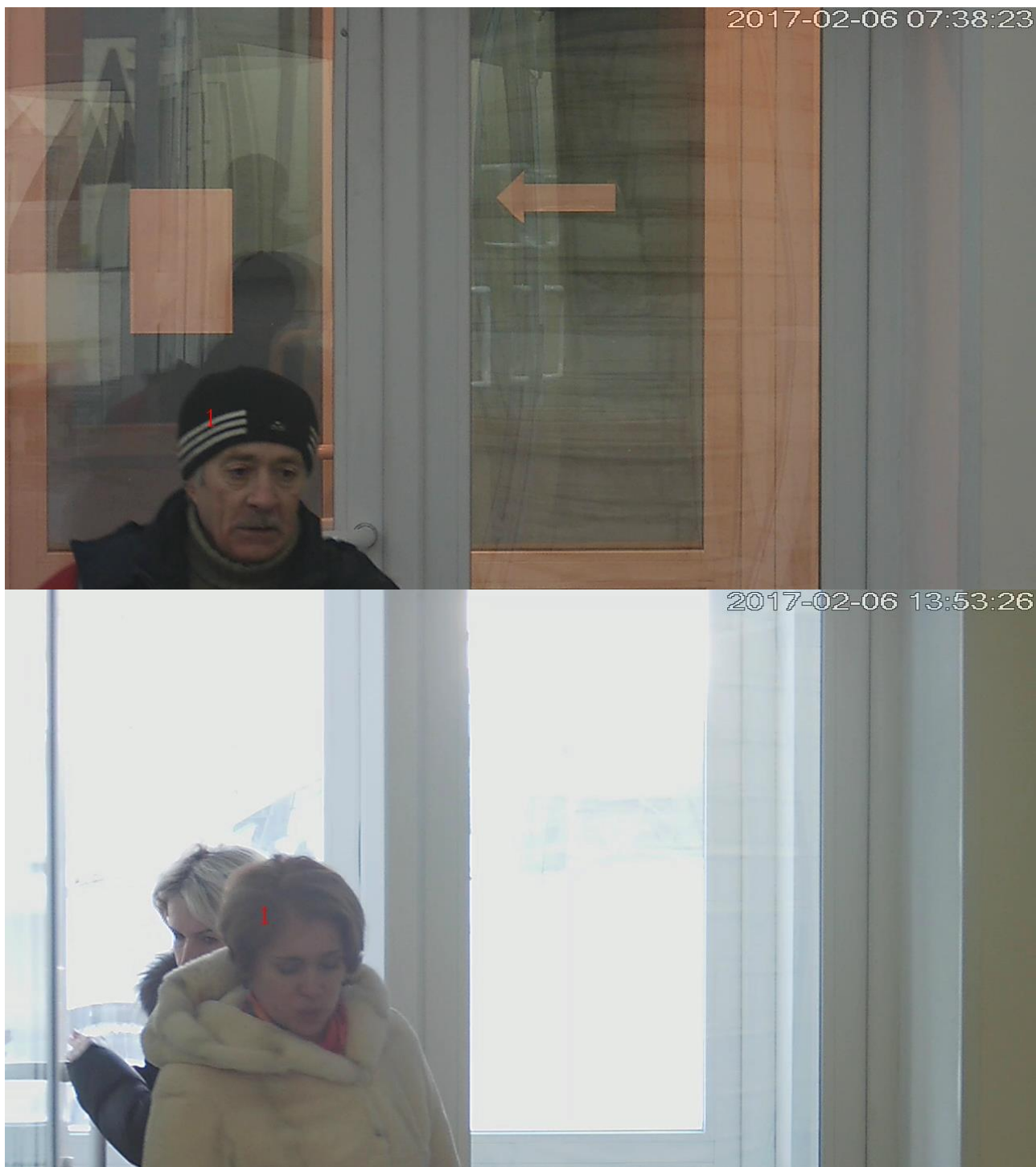


Рис. 53. Пример изменения условий съемки в зависимости от времени суток

4.3. Результаты тестового контроля.

Общее число лиц, обнаруженных стандартным детектором на этих видеозаписях – 7676 и 9827 на первом и втором видео соответственно. Технология подготовки лиц к распознаванию протестирована следующим образом: составлена база изображений лиц из сотрудников института Кибернетики общей численностью 40 человек, присутствующих на видеозаписях, распознавание лиц на кадрах проведено последовательно с каждым из алгоритмов обработки изображений, предложенных в составе комплексной технологии (метод компенсации ракурса DFM; методы компенсации освещения DoG, Log-Gabor; фильтры контрастности и резкости изображений), а также без них (алгоритмом kNN при $k=1$). Результаты этого эксперимента представлены в таблицах 18 и 19.

На первой видеозаписи стандартным детектором на основе вейвлетов Хаара зафиксировано 7676 изображений лиц, из них верно определенных – 6663, остальные приходятся на ложные срабатывания детектора и отсеиваются в дальнейшем анализе при попытке расстановки особых точек лица. Среди всех найденных лиц число лиц людей, содержащихся в базе сотрудников ИК – 638. Ввиду большого числа неизвестных лиц и сложных условий съемки пороговый уровень FAR взят равным 10%.

Таблица 18. Результаты применения отдельных компонентов технологии предварительной обработки изображений лиц в системе распознавания на первой видеопоследовательности

Показатель	Верно распознанные лица	Ложно опознанные лица	Верно распознанные лица, %	FAR, %	FRR, %	Среднее время обработки изображения, мс
Алгоритм kNN	12	693	1,881	11,502	98,119	10
DFM	15	600	2,351	9,959	97,649	10
DoG	55	750	8,621	12,448	91,379	11
DoG + Log-	14	667	2,194	11,07	97,806	25

Gabor				1		
Фильтр контрастности и резкости	34	503	5,329	7,549	94,671	10
DFM + DoG + контрастность и резкость	52	550	8,15	9,129	91,85	11

На второй видеозаписи стандартным детектором на основе вейвлетов Хаара зафиксировано 9823 изображений лиц, из них верно определенных – 8788, остальные приходятся на ложные срабатывания детектора и отсеиваются в дальнейшем анализе при попытке расстановки особых точек лица. Среди всех найденных лиц число лиц людей, содержащихся в базе сотрудников ИК – 555.

Таблица 19. Результаты применения отдельных компонентов технологии предварительной обработки изображений лиц в системе распознавания на второй видеопоследовательности

Показатель	Верно распознанные лица	Ложно опознанные лица	Верно распознанные лица, %	FAR, %	FRR, %	Время обработки изображения, мс
Алгоритм kNN	3	953	0,54	10,844	99,459	10
DFM	5	803	0,9	9,137	99,1	10
DoG	13	844	2,342	9,604	97,658	11
DoG + Log-Gabor	6	988	1,081	11,243	98,919	26
Фильтр контрастности и резкости	7	784	1,261	8,921	98,739	10
DFM + DoG + контрастность и резкость	10	626	1,802	7,123	98,198	11

Как можно видеть из таблиц 18 и 19, оптимальное сочетание алгоритмов позволяет добиться существенного улучшения результатов распознавания по соотношению ошибок FAR/FRR, а также не оказывает существенного влияния на скорость обработки изображений, что свидетельствует об эффективности предложенной технологии обработки изображений.

4.4. Выводы по главе 4.

На основе анализа и сравнения методов обработки изображений составлен учебный модуль по основам обработки изображений и распознаванию образов для студентов специальности БЗ.В11 «Интеллектуальные и информационные системы», бакалавриат, направления 09.03.03 «Прикладная информатика» кафедры Программной Инженерии ИК ТПУ.

Программно-аппаратный комплекс был внедрен в ООО Научно-производственная компания «Техника дела» для формирования входных данных для системы распознавания лиц в виде набора синтезированных изображений лиц в различных ракурсах, а также сравнительного анализа методов обработки цифровых изображений. Получены акт о внедрении разработанной программной системы и справка об использовании результатов работы.

Реализованный программный комплекс был внедрен в составе программно-аппаратной системы в ИК ТПУ в режиме постоянного функционирования. На базе этой системы проведен тестовый контроль и анализ оптимального сочетания алгоритмов обработки изображений для конкретной задачи, показана состоятельность предложенной комплексной технологии и алгоритмов для решения реальных задач обработки, анализа и распознавания образов.

Результаты и выводы

В ходе выполнения диссертационной работы были получены следующие основные научные и практические результаты.

1. Выбраны наиболее подходящие с точки зрения эффективности и быстродействия методы анализа и обработки изображений в рамках технологии подготовки изображений лиц к распознаванию личности в режиме реального времени по результатам оценки различных существующих методов и алгоритмов.
2. Произведен анализ существующих алгоритмов расстановки особых точек лица, оценки угла поворота, а также синтеза 3D моделей. Обоснована необходимость модификации существующих алгоритмов с целью повышения их скорости работы и автоматизации.
3. Разработан новый алгоритм трекинга лиц в видеопотоке, комбинирующий информацию, полученную методом background subtraction и методом Виолы-Джонса, и вычисляющий трек лица в видеопотоке со статичным фоном на 20 и более процентов быстрее алгоритма трекинга, основанного на детекторе Виолы-Джонса.
4. Разработана оригинальная модификация метода синтезирования DFM, использующая принципы восстановления ландшафтных поверхностей методом интерполяции Шепарда, позволяющая автоматически синтезировать рельеф лица по единственному плоскому изображению.
5. Разработан новый алгоритм формирования базы изображений лиц из одного фронтального изображения, основанный на предложенной модификации метода DFM и позволяющий применять методы классификации лиц в условиях ограниченного набора исходных изображений, повышая точность программных систем распознавания личности, использующих методы классификации.
6. Предложена новая комплексная технология подготовки изображений лиц к распознаванию личности, основанная на авторских методах компенса-

ции ракурса и трекинга лиц и рассчитанная на использование в программных системах распознавания в видеопотоке в режиме реального времени.

7. Протестированы авторские методы и алгоритмы в составе новой комплексной технологии обработки изображений лиц для программной системы распознавания личности на различных базах изображений лиц, как собственных, так и находящихся в открытом доступе.
8. Проведена апробация рассмотренных алгоритмов и методов в рамках комплексной технологии обработки изображений лиц, подтверждающая их быстроедействие и демонстрирующая повышение точности программных систем распознавания личности по изображению лица в видеопотоке в режиме реального времени.
9. Разработанное алгоритмическое и программное обеспечение внедрено в научно-образовательной и промышленной организациях, а результаты проведенной работы - в учебный процесс, что подтверждено соответствующими документами.

Список литературы

1. A.J. Goldstein, L.D. Harmon, and A.B. Lesk, "Identification of Human Faces", Proc. IEEE, May 1971, Vol. 59, No. 5, 748-760.
2. M.A. Turk and A.P. Pentland, "Face Recognition Using Eigenfaces", Proc. IEEE, 1991, 586-591.
3. Adini Y. Face recognition: The problem of compensating for changes in illumination direction / Y. Adini, Y. Moses, S. Ullman // IEEE Transactions on Pattern Analysis and Machine Intelligence. - 1997. – 19. - P. 721–732.
4. Zhao W., Chellappa R., Phillips P.J., Rosenfeld A. Face recognition: A literature survey // ACM computing surveys (CSUR). 2003. vol. 35. no. 4. pp. 399-458.
5. Заварикин Д.Н., Кадейшвили А.А., Коробкова С.В., Соколов А.Ю., Степаненко О.В. Система некооперативной биометрической идентификации людей // Вестник МГТУ им. Н.Э. Баумана. Сер. "Приборостроение", 2012. С. 40-45.
6. Zhou H., Mian A., Wei L., Creighton D., Hossny M., Nahavandi S. Recent advances on singlemodal and multimodal face recognition: a survey // Proceedings of IEEE Transactions on Human-Machine Systems. 2014. vol. 44. no. 6. pp. 701-716.
7. Taigman Y., Yang M., Ranzato M.A., Wolf L. Deepface: Closing the gap to human-level performance in face verification // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014. pp. 1701-1708.
8. Tomasi C., Kanade T. Detection and Tracking of Point Features. Carnegie Mellon University Technical Report CMU-CS-91-132, 1991.
9. DCRP Review: Canon PowerShot S5 IS. Режим доступа: http://www.dcresource.com/reviews/canon/powershot_s5-review (Дата обращения: 25.03.2017).

10. Назипов А.З., Шлеймович М.П. Трекинг кисти руки в системе распознавания динамических жестов // Сборник научных трудов по итогам международной научно-практической конференции «Актуальные вопросы науки и техники», Изд.: Инновационный центр развития образования и науки, КНИТУ-КАИ им. А.Н. Туполева, г. Казань, 2015 С. 27-28.
11. Кривцов О.А. Отслеживание головы человека на видеокдрах: состояние проблемы / О.А. Кривцов, А.М. Кориков // Журнал «Информационные технологии» – М.: Изд-во «Новые технологии», 2010. – № 7. – С. 2–9.
12. Пастушков А.В., Калайда В.Т. Программная система слежения за объектом в видеопотоке // Южно-Сибирский научный вестник. 2013. Т. 3, № 1. С. 90-91.
13. Watada J., Musa Z., Jain L.C., Fulcher J., Human tracking: a state of art survey // Proceedings of the 14th International Conference on Knowledge-based and Intelligent Information and Engineering Systems, 2010, pp. 454–463.
14. Rizvi S.A., Phillips P.J., Moon H. The FERET verification testing protocol for face recognition algorithms // Automatic Face and Gesture Recognition, Proceedings on Third IEEE International Conference. – IEEE, 1998. – С. 48-53.
15. Wu Z., Song S., Khosla A., Yu F., Zhang L., Tang X., Xiao J. 3D ShapeNets: A deep representation for volumetric shapes // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2015. – С. 1912-1920.
16. Comaniciu D., Ramesh V. Robust Detection and Tracking of Human Faces with an Active Camera. // Proceedings on IEEE Visual Surveillance, 2011.
17. Bradski G., Kaehler A. Learning OpenCV: Computer vision with the OpenCV library // O'Reilly Media, Inc., 2008.

- 18.Сергеев А.Е., Конушин А.С., Конушин В.С. Подавление ложноположительных обнаружений лиц в видеопотоках систем видеонаблюдения // Компьютерная оптика. 2016. Т. 40. № 6. С. 958-967.
- 19.Zagoruiko N.G., Kutnenko O.A.. Recognition Methods Based on the AdDel Algorithm // Pattern Recognition and Image Analysis. 2004. Vol. 14. № 2. pp.198-204.
- 20.Chernyshov V.G., Mestetskii L.M. Mobile Computer Vision System for Hand-Based Identification // Pattern Recognition and Image Analysis. 2015. Vol. 25, № 2, pp. 209-214.
- 21.Желтов С.Ю., Князь В.А, Салтыкова Е.Б. Автоматическая трехмерная реконструкция наблюдаемой сцены по стереовидеопоследовательностям // Вестник Компьютерных И Информационных Технологий. ISSN 1810-7206, 2006, № 3. С. 39-43.
- 22.Галактионов В.А., Дерябин Н.Б., Денисов Е.Ю. Объектно-ориентированный подход к реализации систем компьютерной графики // «Информационно-измерительные и управляющие системы», № 6, 2009, с. 96-108 («Интеллектуальные и адаптивные роботы», № 1-2, 2009, с. 96-108).
- 23.Vizilter Yu.V., Zheltov S.Yu. Similarity measures and comparison metrics for image shapes // Journal Of Computer And Systems Sciences International. 2014. Vol. 53. pp 542–555.
- 24.Krylov A.S., Nasonov A.V. 3d image sharpening by grid warping // Lecture Notes in Computer Science. 2015. Vol. 9242. pp. 441–450.
- 25.Шерстобитов А.И., Федосов В.П., Приходченко В.А., Тимофеев М.В. Распознавание лиц на групповых фотографиях с использованием алгоритмов сегментации // Известия ЮФУ. Технические науки. 2013. Т. 148. С. 66-73.
- 26.Зотин А.Г., Пахирка А.И., Дамов М.В., Савчина Е.И. Улучшение визуального качества изображений, полученных в сложных условиях

- освещенности на основе инфракрасных данных // Программные продукты и системы. 2016. Т. 115. № 3. С.109-120.
- 27.Калайда В. Т., Губанов Н. Ю. Идентификация лица человека методом опорной гиперплоскости // Вычислительные технологии. – 2007. – №1. – С.96–101.
- 28.Bui T.T.T., Phan N.H., Spitsyn V.G., Bolotova Y.A., Savitsky Y.V. Development of algorithms for face and character recognition based on wavelet transforms, PCA and neural networks // Proceedings of IEEE International Siberian Conference on Control and Communications (SIBCON'2015). 2015. pp. 1-6.
- 29.Афонасенко А.В., Елизаров А.И. Обзор методов распознавания структурированных символов // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2008. – №. 2-1 (18).
- 30.Korikov A.M. Neural network technologies for image classification [Electronic resources] / A.M. Korikov, A.V. Tungusova // Proceedings of SPIE. — 2015. — Vol. 9680 : Atmospheric and Ocean Optics: Atmospheric Physics.
- 31.Bronstein A., Bronstein M., Kimmel R. Expression-invariant 3D face recognition // Proceedings of Audio and Video-based Biometric Person Authentication (AVBPA). Lecture Notes in Computer Science 2688, Springer. 2003. pp. 62–69.
- 32.Paysan P. Knothe R., Amberg B., Romdhani S., Vetter T. A 3D Face Model for Pose and Illumination Invariant Face Recognition // Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). 2–4 Sept. 2009. P. 296–301.
- 33.Щеголева Н. Восстановление 3D поверхности лица. Сборник научных трудов 23-ей Международной научной конференции «GraphiCon2013» (г. Владивосток, 16–20 сентября 2013 г.) Владивосток: Институт автоматизации и процессов управления ДВО РАН, 2013. С. 135-139.

34. Gorodnichy D., Granger E., Radtke P. Survey of commercial technologies for face recognition in video // CBSA, Border Technology Division, Tech. Rep. 2014. Vol. 22.
35. Zhang H., Berg A.C., Maire M., Malik J. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition // Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. – IEEE, 2006. – Т. 2. – С. 2126-2136.
36. Манолов А.И., Соколов А.Ю., Степаненко О.В., Тумачек А.С., Тяхт А.В., Цискаридзе А.К., Заварикин Д.Н., Кадейшвили А.А.. Некооперативная биометрическая идентификация по 3D-моделям лица с использованием видеокамер высокого разрешения // Сборник научных трудов 19-ой Международной научной конференции «GraphiCon2009» (г. Москва, 5–9 октября 2009 г.) М.: МГУ им. М.В. Ломоносова, 2009. С. 210-214.
37. Klehm O., Rousselle F., Papas M., Bradley D., Hery C., Bickel B., Beeler, T. Recent advances in facial appearance capture // Computer Graphics Forum. 2015. vol. 34. no. 2. pp. 709-733.
38. Song H., Lv J., Wang Y. Rapid 3D Face Modeling from Video // Pacific Rim Conference on Multimedia. 2015. pp. 201-211.
39. Garrido P., Zollhöfer M., Casas D., Valgaerts L., Varanasi K., Pérez P., Theobalt C. Reconstruction of Personalized 3D Face Rigs from Monocular Video // ACM Transactions on Graphics (TOG). 2016. vol. 35. no. 3. p. 28.
40. Kemelmacher-Shlizerman I., Seitz S.M. Face reconstruction in the wild // Computer Vision (ICCV), IEEE International Conference on 2011. – С. 1746-1753.
41. Mutch J., Lowe D.G. Multiclass object recognition with sparse, localized features // Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. – IEEE, 2006. – Т. 1. – С. 11-18.
42. Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов. – М.: Мир, 1989. – 448 с.

- 43.Оппенгейм А.В. Цифровая обработка сигналов / А.В. Оппенгейм, Р.В. Шафер. – М.: Техносфера, 2006. – 848 с.
- 44.P. Viola and M.J. Jones, «Rapid Object Detection using a Boosted Cascade of Simple Features», proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2001), 2001.
- 45.Местецкий Л.М., «Математические методы распознавания образов», МГУ, ВМиК, Москва, 2002–2004., с. 42 – 44.
- 46.Вражнов Д.А. О качестве работы алгоритмов слежения за объектами на видео / Д.А. Вражнов, А.В. Шаповалов, В.В. Николаев // Компьютерные исследования и моделирование. – 2012. - Т. 4, № 2, С. 303–313.
- 47.Abtahi S., Omidyeganeh M., Shirmohammadi S., Hariri B. YawDD: A Yawning Detection Dataset // Proceedings of ACM Multimedia Systems, Singapore, pp. 24-28, March 2014.
- 48.Kumar S., Yadav J.S. Video object extraction and its tracking using background subtraction in complex environments // Perspectives in Science. – 2016. – Т. 8. – С. 317-322.
- 49.Mandellos N.A., Keramitsoglou I., Kiranoudis C.T. A background subtraction algorithm for detecting and tracking vehicles // Expert Systems Application, 2011. № 38 (3), pp. 1619—1631.
- 50.Обухова Н.А. Векторы оптического потока в задачах сегментации и сопровождения подвижных объектов // Изв. вузов России. Радиоэлектроника. – 2006. – Вып. 2. – С. 42–51.
- 51.Mithun N.C., Howlader T., Rahman S.M.M. Video-based tracking of vehicles using multiple time-spatial images // Expert Systems with Applications. – 2016. – Т. 62. – С. 17-31.
- 52.Гаганов В. Сегментация движущихся объектов в видеопотоке / В. Гаганов, А. Конушин // Компьютерная графика и мультимедиа. - 2004. - №2(3). Режим доступа: <http://cgm.computergraphics.ru/content/view/67>, свободный. (Дата обращения: 25.03.2017)

53. Abbott R.G., Williams L.R. Multiple target tracking with lazy background subtraction and connected components analysis // *Machine Vision and Applications*. – 2009. – Т. 20. – №. 2. – С. 93-101.
54. Bouwmans, T. Traditional and recent approaches in background modeling for foreground detection: an overview // *Computer Science Review*, 2014. № 11, pp. 31—66.
55. Boda R., Priyadarsini M.J.P. Face Detection And Tracking Using KLT And Viola Jones // *ARPN Journal of Engineering and Applied Sciences*, Vol. 11, No. 23, December 2016, pp. 13472-13476.
56. Mikhaylov D., Samoylov A., Minin P., Egorov A. Face Detection and Tracking from Image and Statistics Gathering // *IEEE Conference on Signal-Image Technology and Internet-Based Systems*, 2014.
57. Charles E. Jacobs A. Finkelstein D.H. Sales in Fast Multiresolution Image Querying // *Department of Computer Science and Engineering, University of Washington, Seattle, Washington 98195*.
58. Datta R., Joshi D., Li J., Wang J.Z. Image Retrieval: Ideas, Influences, and Trends of the New Age // *The Pennsylvania State University*.
59. Корреляционное сопоставление изображений – Техническое зрение. Режим доступа: http://wiki.technicalvision.ru/index.php/Корреляционное_сопоставление_изображений (Дата обращения: 25.03.2017).
60. Альтман Е.А., Захаренко Е.И. Быстрый алгоритм вычисления двумерной корреляции для видеобработки // *Доклады ТУСУР*. 2015. №2 (36) С.119-124.
61. Bay H., Ess A., Tuytelaars T., Van Gool L. SURF: Speeded up robust features // *Computer Vision and Image Understanding*. – 2008. – V. 110. – P. 346–359.
62. Leutenegger S. BRISK: Binary Robust invariant scalable keypoints / S. Leutenegger, M.Chli, R.Y. Siegwart // *Proceedings of the 2011 International*

- al Conference on Computer Vision (ICCV '11), 6 November 2011, P. 2548-2555.
63. Фисенко В.Т. Компьютерная обработка и распознавание изображений: учебное пособие / В.Т. Фисенко, Т.Ю. Фисенко - СПб: СПбГУ ИТМО, 2008. – 192 с.
64. Мониц Ю.И. Оценки качества для анализа цифровых изображений / Ю. И. Мониц, В. В. Старовойтов. - Минск: Государственное научное учреждение ОИПИ НАН Беларуси, 2008.
65. Kanjar D. Image Sharpness Measure for Blurred Images in Frequency Domain / D. Kanjar, V. Masilamani // International Conference on Design and Manufacturing. - Procedia Engineering, 2013. - P. 149 – 158.
66. Tan X. Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions / X. Tan, B. Triggs // IEEE Transactions on image processing, Vol. 19, № 6, June 2010. – P. 1635-1650.
67. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс // Москва: Техносфера, 2012. – 1104 с.
68. Zuiderveld K. Contrast limited adaptive histogram equalization // Graphics gems IV, Academic Press Professional, Inc. San Diego, CA, USA, 1994. - P. 474-485.
69. Фисенко Т.Ю. Исследование и разработка методов улучшения подводных изображений / Т. Ю. Фисенко, В. Т. Фисенко – Материалы международной конференции «Прикладная оптика», 15-19 октября 2012. - С. 294-298.
70. DeMenthon D.F. Model-based object pose in 25 lines of code / D.F. DeMenthon, L.S. Davis // International Journal of Computer Vision, Vol. 15, Issue 1-2, June 1995. - P. 123-141.
71. Milborrow S. Multiview Active Shape Models with SIFT Descriptors for the 300-W Face Landmark Challenge / S. Milborrow, T.E. Bishop, F. Nicolls // The IEEE International Conference on Computer Vision (ICCV) Workshops, 2013, P. 378-385.

72. Cootes T.F., Taylor C.J., Cooper D.H., Graham J. Active shape models - their training and application // Computer Vision and Image Understanding, Vol. 61, January 1995, P. 38-59.
73. S. Baker, R. Gross, and I. Matthews. Lucas-Kanade 20 years on: A unifying framework: Part 3. Technical Report CMU-RI-TR-03-35, Carnegie Mellon University Robotics Institute, 2003.
74. Procrustes analysis - Wikipedia. Режим доступа: https://en.wikipedia.org/wiki/Procrustes_analysis. (Дата обращения: 25.03.2017).
75. Stegmann M.B. Analysis and Segmentation of Face Images using Point Annotations and Linear Subspace Techniques. Technical report IMM-REP-2002-22, Informatics and Mathematical Modelling, Technical University of Denmark, 2002.
76. Matthews I., Baker S. Active Appearance Models Revisited. International Journal of Computer Vision, Vol. 60, No. 2, November, 2004, pp. 135 — 164.
77. Kazemi V., Sullivan J. One millisecond face alignment with an ensemble of regression trees // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2014. – С. 1867-1874.
78. Jingu H., Savvides M. In between 3D active appearance models and 3D Morphable Models // Proceedings of the CVPR Workshops, 2009, pp. 20–25.
79. Milborrow S. Active Shape Models with SIFT Descriptors and MARS / S. Milborrow, F. Nicolls // The 9th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VIS-APP), 2014, P. 119-133.
80. Xuehan Xiong. Supervised Descent Method and its Applications to Face Alignment / Xuehan Xiong, F. De la Torre // The IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 23-28 June 2013, P. 532-539.

81. Davis E. King. Dlib-ml: A Machine Learning Toolkit // Journal of Machine Learning Research, №10, 2009, P. 1755-1758.
82. Zimmermann J. SilSketch: Automated Sketch-Based Editing of Surface Meshes / J. Zimmermann, A. Nealen, M. Alexa // EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling, 2007, P. 23-30.
83. Vatahska T., Bennewitz M., Behnke S. Feature-based head pose estimation from images // Humanoid Robots, 2007. 7th IEEE-RAS International Conference on. – IEEE, 2007. – C. 330-335.
84. Lee M., Choi C.H. Real-time facial shape recovery from a single image under general, unknown lighting by rank relaxation // Computer Vision and Image Understanding. – 2014. – T. 120. – C. 59-69.
85. Georgios P., Panagiotis P., Theoharis T. Using facial symmetry to handle pose variations in real-world 3D face recognition // IEEE Transactions On Pattern Analysis and Machine Intelligence, № 33 (10). 2011. pp. 1938–1951.
86. Agarwal S. Building Rome in a Day / S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, R. Szeliski // Communications of the ACM, Vol. 54 No. 10, P. 105-112.
87. Breuer P., Kim K., Kienzle W., Scholkopf B., Blanz V. Automatic 3D Face Reconstruction from Single Images or Video // In FG, 2008.
88. Shahriar Hossain M. Inexpensive Construction of a 3D Face Model from Stereo Images / M. Shahriar Hossain, M. Akbar, J. Denbigh Starkey // 10th international conference on Computer and information technology (ICCIT), 27-29 Dec. 2007, P. 1-6.
89. Asthana A., Marks T.K., Jones M.J., Tieu K.H., Rohith M.V. Fully automatic pose-invariant face recognition via 3D pose normalization // Computer Vision (ICCV), 2011 IEEE International Conference on. – IEEE, 2011. – C. 937-944.

90. Marin-Jimenez M.J., de la Blanca N.P. Empirical study of multi-scale filter banks for object categorization // Pattern Recognition, 2006. ICPR 2006. 18th International Conference on. – IEEE, 2006. – Т. 1. – С. 578-581.
91. Sagonas C., Panagakis Y., Zafeiriou S., Pantic M. Face frontalization for Alignment and Recognition // arXiv preprint arXiv:1502.00852. – 2015.
92. Агарков А.В., Нюнькин К.М. Восстановление фронтального вида лица человека по одному изображению // Искусственный интеллект. – 2005. – №. 1. – С. 4-12.
93. Martins P., Batista J. Monocular head pose estimation // International Conference Image Analysis and Recognition. – Springer Berlin Heidelberg, 2008. – С. 357-368.
94. Liao H.B., Chen Q.H., Zhou Q.J., Guo L. Rapid 3D face reconstruction by fusion of SFS and Local Morphable Model // Journal of Visual Communication and Image Representation. – 2012. – Т. 23. – №. 6. – С. 924-931.
95. Gao Y., Dai Q., Zhang N., 3D model comparison using spatial structure circular descriptor // Pattern Recognition № 43 (3). 2010. pp. 1142–1151.
96. Jiang D.L., Hu Y.X., Yan S.C., Zhang L., Zhang H., Gao W. Efficient 3D reconstruction for face recognition // Pattern Recognition, № 6. 2005. С. 787–798.
97. Lee M., Choi C.H. Fast facial shape recovery from a single image with general, unknown lighting by using tensor representation // Pattern Recognition. – 2011. – Т. 44. – №. 7. – С. 1487-1496.
98. Kemelmacher-Shlizerman I., Basri R., Nadler B. 3D Face Reconstruction from Single Two-Tone and Color Images // Shape Perception in Human and Computer Vision. – Springer London, 2013. – С. 275-284.
99. Kalal Z., Mikolajczyk K., Matas J. Forward-Backward Error: Automatic Detection of Tracking Failures // International Conference on Pattern Recognition, 2010.

100. Bouguet J.Y. Pyramidal implementation of the Affine Lucas Kanade Feature Tracker Description of the algorithm // Intel Corporation. – 2001. – T. 5. – №. 1-10. – C. 4.
101. Nikolov B., Kostov N. Motion detection using adaptive temporal averaging method // Radioengineering, № 23 (2), 2014. C. 652—658.
102. Sardari F., Moghaddam M.E. An object tracking method using modified galaxy-based search algorithm // Swarm and Evolutionary Computation. – 2016. – T. 30. – C. 27-38.
103. Kemelmacher-Shlizerman I., Basri R. 3D face reconstruction from a single image using a single reference face shape // IEEE transactions on pattern analysis and machine intelligence. – 2011. – T. 33. – №. 2. – C. 394-405.
104. Anitha C., Venkatesha M.K., Adiga B.S. High speed face detection and tracking // International Journal of Soft Computing and Artificial Intelligence, ISSN: 2321-404X, Volume-3, Issue-2, Nov-2015, pp. 84-90.
105. Aggarwal A., Biswas S., Singh S., Sural S., Majumdar A.K. Object tracking using background subtraction and motion estimation in MPEG videos // Asian Conference on Computer Vision. – Springer Berlin Heidelberg, 2006. – C. 121-130.
106. Zamalieva D., Yilmaz A. Background subtraction for the moving camera: A geometric approach // Computer Vision and Image Understanding. – 2014. – T. 127. – C. 73-85.
107. Blanz V., Vetter T. Face recognition based on fitting a 3D morphable model // Proceedings of IEEE Transactions on pattern analysis and machine intelligence. 2003. vol. 25. no. 9. pp. 1063-1074.
108. Zhu X., Yan J., Yi D., Lei Z., Li S.Z. Discriminative 3D morphable model fitting // Proceedings of 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG). 2015. vol. 1. pp. 1-8.

109. Basel Face Model database. Access at:
<http://faces.cs.unibas.ch/bfm/main.php?nav=1-1-0&id=details>. (Дата обращения: 25.03.2017).
110. Renka R.J. Multivariate Interpolation of Large Sets of Scattered Data // ACM Transactions on Mathematical Software (TOMS). 1988. Vol. 14. №. 2. pp. 139-148.
111. ALGLIB User Guide. Интерполяция/аппроксимация по обратному средневзвешенному расстоянию. Режим доступа:
<http://alglib.sources.ru/interpolation/inversedistanceweighting.php#tuning>
(Дата обращения: 16.03.2017).
112. Renka R.J. Algorithm 790: CSHEP2D: Cubic Shepard method for bivariate interpolation of scattered data // ACM Transactions on Mathematical Software (TOMS). 1999. Vol. 25. №. 1. pp. 70-73.
113. Antonakos E. Zafeiriou S. Automatic construction of deformable models in-the-wild // Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 1813–1820. IEEE, 2014.
114. ГОСТ Р ИСО/МЭК 19794-5-2006. Автоматическая идентификация. Идентификация биометрическая. Форматы обмена биометрическими данными // М.: Госстандарт России. 2006.
115. Климов А.С., Розалиев В.Л., Орлова Ю.А. Автоматизация построения объемной модели головы человека // Известия Волгоградского государственного технического университета. – 2014. – Т. 22. – №. 25 (152).
116. Neo J. Three-dimensional generic elastic models for two-dimensional pose synthesis and face recognition : дис. – Carnegie Mellon University, 2010.
117. Computational Vision: Archive. Режим доступа:
www.vision.caltech.edu/html-files/archive.html (Дата обращения: 25.03.2017).

118. Phillips P.J., Moon H., Rauss P.J., Rizvi S. The FERET evaluation methodology for face recognition algorithms // Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence. 2000. vol. 22. pp. 1090-1104.
119. Лычков И.И., Алфимцев А.Н., Девятков В.В. Отслеживание движущихся объектов для мониторинга транспортного потока // Информационные технологии и системы: материалы 34-й конф. молодых ученых и специалистов ИППИ РАИ, 2-7 окт. – 2011. – С. 31.
120. Кухарев Г.А., Каменская Е.И., Матвеев Ю.Н., Щеголева Н.Л. Методы обработки и распознавания изображений лиц в задачах биометрии / Под ред. М.В. Хитрова. СПб: Политехника. 2013. 388 с.
121. Ayonix Vendor Tests and Comparison Charts. Режим доступа: <http://ayonix.com/technology/test-results/> (Дата обращения: 28.05.2017).

Публикации по теме диссертации

Статьи в журналах, которые включены в перечень российских рецензируемых научных журналов и изданий для опубликования основных научных результатов диссертаций:

1. Небаба С.Г. Автоматическая оценка ракурса лица в кадре и приведение изображения к нулевым углам поворота / С.Г. Небаба, А.А. Захарова, С.Ю. Андреев // Научная визуализация, 7 (4), 2015.
2. Небаба С.Г. Алгоритм формирования индивидуальной трехмерной модели лица человека из растрового изображения лица и набора частных трехмерных моделей лица / С.Г. Небаба, А.А. Захарова, С.Ю. Андреев // Информационное общество, №5, 2015.
3. Савицкий Ю.В. Анализ методов извлечения вектора признаков из изображения лица человека для задачи идентификации / Ю.В. Савицкий, С.Г. Небаба, В.Г. Спицын, С.Ю. Андреев, М.А. Макаров // Научная визуализация, 8 (2), 2016.

4. Небаба С.Г. Алгоритм построения деформируемых 3D моделей человеческого лица и обоснование его применимости в системах распознавания личности / С.Г. Небаба, А.А. Захарова // Труды СПИИРАН. 2017. Вып. 52. С. 157-179.
5. Небаба С. Г. Тестирование технологии подготовки изображений лиц к распознаванию личности в видеопотоке в режиме реального времени // Современная наука: актуальные проблемы теории и практики. Серия «Естественные и технические науки», 2017. №3-4. С. 73-77.
6. Небаба С.Г. Трекинг изображений лиц в видеопотоке применительно к алгоритму распознавания лиц / С.Г. Небаба, А.А. Захарова // Кибернетика и программирование (принято к печати)

Scopus и Web of Science:

7. Nebaba S.G. Methods of Automatic Face Angle Recognition for Life Support and Safety Systems / S.G. Nebaba, A. A. Zakharova, T.V. Sidorenko, V.R. Viitman // The European Proceedings of Social & Behavioural Sciences EpSBS (January 2017), vol. XIX, pp. 735-744.

Другие публикации:

8. Небаба С.Г. Подготовка изображений лиц в видеопотоке к распознаванию и фильтрация неинформативных изображений / С.Г. Небаба, С.Ю. Андреев, М.А. Макаров, А.А. Захарова // Сборник трудов XII Международной научно-практической конференции студентов, аспирантов и молодых учёных «Молодёжь и современные информационные технологии». – Томск, 2014.
9. Небаба С. Г. Подготовка изображений лиц к распознаванию и фильтрация неинформативных изображений / С.Г. Небаба, С.Ю. Андреев, М.А. Макаров // Проблемы информатики. 2014. № 4. С. 53-60.
10. Nebaba S.G. Preparation of the Face Images in a Video Stream for Recognition and Filtering of Non-Informative Images / S.G. Nebaba, S.Y. Andreev, M.A. Makarov // Applied Mechanics and Materials, Sensors, Measurement, Intelligent Materials and Technologies III, 2015, Vols. 738-739, P. 648-655.

11. Небаба С.Г. Применение алгоритма формирования индивидуальной трехмерной модели человеческого лица в системе распознавания личности по изображению лица / С.Г. Небаба, А.А. Захарова // Труды 26-ой Международной научной конференции «GraphiCon2016», С. 310-313.
12. Небаба С.Г. Технология подготовки изображений лиц к распознаванию личности в видеопотоке в режиме реального времени // Материалы XX Юбилейной Международной конференции по вычислительной механике и современным прикладным программным системам (ВМСППС'2017), 24-31 мая 2017 г., Алушта. – М.: Изд. МАИ, 2017. – 816 с. С. 600-602.

Свидетельства о регистрации программ для ЭВМ:

13. Свидетельство о регистрации программы для ЭВМ №2017611032 «Создание набора изображений лиц для алгоритмов распознавания». Авторы: Небаба С.Г., Захарова А.А. Дата регистрации в Реестре программ для ЭВМ 19 января 2017 г.

Приложения

Приложение 1. Фрагменты исходного кода основных модулей программного комплекса

facemodel.h

```
#ifndef _F_MODEL_H_
#define _F_MODEL_H_

#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>

#include "Triangle.h"
#include <math.h>

using namespace trl;

// класс модели триангуляции лица
namespace fml{
    class MFace
    {
    public:
        std::vector <Triangle> faceTriangles; // вектор треугольников, составлен-
ных из особых точек лица
        double Model3DM[123][182]; // матрица глубины для модели лица
        cv::Point2i coordsM[20000]; // массив координат для соотнесения особых точек
лица с их глубиной
        double coefM; // коэффициент масштабирования для матрицы глубины
        int typeOfModel = 0;
    public:
        MFace(); // конструктор
        MFace(cv::Point3f *fPoints, cv::Point2i* coords, double** Model3D, double coef);

        void MFace::drawFace(cv::Mat img); // рисование треугольной сетки на за-
данном изображении
        void MFace::sortByZ(); // сортировка (не используется)
        int MFace::maxST(); // номер треугольника с максимальной площадью
        int MFace::maxLN(); // номер треугольника с максимальной стороной
        void MFace::break2Trs(int num); // разбиение одного треугольника на 2 по
наибольшей стороне
        void MFace::breakATrs(int count); // цикл разбиений треугольников с макси-
мальной площадью до заданного общего числа треугольников в модели

        void MFace::rotation(cv::Point3f centerP, double alpha, double beta); // поворот
на заданный угол по оси X и по оси Y с центром в заданной точке
        void MFace::rotationZ(cv::Point3f centerP, double alpha, double beta); // поворот
координаты Z на заданный угол по оси X и по оси Y с центром в заданной точке
    };
} //namespace fml
```

```
#endif
```

facemodel.cpp

```
#include "FaceModel.h"
```

```
using namespace fml;
```

```
MFace::MFace()
```

```
{  
}
```

```
MFace::MFace(cv::Point3f *fPoints, cv::Point2i* coords, double** Model3D, double coef)
```

```
{
```

```
    // сохранение матрицы глубин и координат особых точек
```

```
    for (int i = 0; i < 123; i++)
```

```
        for (int j = 0; j < 182; j++)
```

```
            Model3DM[i][j] = Model3D[i][j];
```

```
    coefM = coef;
```

```
    for (int i = 0; i < 71; i++)
```

```
        coordsM[i] = coords[i];
```

```
    // формирование вектора треугольников
```

```
    int i, j, k;
```

```
    i = 0; j = 1; k = 36;
```

```
    faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
```

```
        j = 17;
```

```
    faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
```

```
    i = 1; j = 2; k = 41;
```

```
    faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
```

```
        j = 36;
```

```
    faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
```

```
    i = 2; j = 3; k = 41;
```

```
    faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
```

```
    i = 3; j = 4; k = 48;
```

```
    faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
```

```
        j = 41;
```

```
    faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
```

```
    i = 4; j = 5;
```

```
    faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
```

```
    i = 5; j = 6;
```

```
    faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
```

```
    i = 6; j = 7; k = 59;
```

```
    faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
```

```
        j = 48;
```

```
    faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
```

```
    i = 7; j = 58;
```

```
    faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
```

```
        j = 8; k = 58;
```

```
    faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
```

```
    i = 8; j = 9; k = 56;
```

```
    faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
```

```

        j = 56; k = 57;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
        j = 57; k = 58;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 9; j = 10; k = 55;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
        j = 55; k = 56;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 10; j = 11; k = 54;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
        j = 54; k = 55;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 11; j = 12; k = 54;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 12; j = 13;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 13; j = 14; k = 46;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
        j = 46; k = 54;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 14; j = 15; k = 46;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 15; j = 16; k = 45;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
        j = 45; k = 46;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 16; j = 26; k = 45;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 17; j = 18; k = 37;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
        j = 36;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 18; j = 19;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 19; j = 20; k = 38;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
        j = 37;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 20; j = 21;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 21; j = 38; k = 39;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
        j = 27;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
        j = 22; k = 27;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 22; j = 23; k = 43;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
        j = 42;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
        j = 27; k = 42;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));

```

```

i = 23; j = 24; k = 43;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 24; j = 25; k = 44;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
    j = 43;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 25; j = 26;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 26; j = 44; k = 45;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 27; j = 28; k = 39;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
    k = 42;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 28; j = 29; k = 39;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
    k = 42;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 29; j = 30; k = 32;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
    k = 34;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
    j = 31; k = 32;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
    k = 39;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
    j = 34; k = 35;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
    j = 35; k = 42;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 30; j = 32; k = 33;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
    j = 34;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 31; j = 39; k = 48;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
    j = 32; k = 49;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
    j = 48;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 32; j = 33; k = 50;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
    j = 49;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 33; j = 34; k = 52;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
    j = 51;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
    j = 50; k = 51;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 34; j = 35; k = 53;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));

```

```

        j = 52;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 35; j = 53; k = 54;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
        j = 42;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 36; j = 37; k = 41;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 37; j = 38; k = 40;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
        j = 40; k = 41;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 38; j = 39; k = 40;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 39; j = 40; k = 48;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 40; j = 41;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 42; j = 43; k = 47;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
        j = 47; k = 54;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 43; j = 44; k = 47;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 44; j = 45; k = 46;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
        j = 46; k = 47;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 46; j = 47; k = 54;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 48; j = 49; k = 60;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
        j = 59;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 49; j = 50; k = 61;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
        j = 60;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 50; j = 51; k = 62;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
        j = 62; k = 61;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 51; j = 52; k = 62;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 52; j = 53; k = 63;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
        j = 63; k = 62;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 53; j = 54; k = 64;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
        j = 64; k = 63;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));

```

```

i = 54; j = 55; k = 64;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 55; j = 56; k = 63;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
    j = 63; k = 64;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 56; j = 57; k = 62;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
    j = 62; k = 63;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 57; j = 58; k = 62;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 58; j = 59; k = 61;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
    j = 61; k = 62;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 59; j = 60; k = 61;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));

```

// добавочные треугольники

```

i = 68; j = 69; k = 24;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 69; j = 70; k = 19;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 21; j = 22; k = 69;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 0; j = 17; k = 70;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 17; j = 18; k = 70;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 18; j = 19; k = 70;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 19; j = 20; k = 69;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 20; j = 21; k = 69;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 22; j = 23; k = 69;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 21; j = 22; k = 69;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 22; j = 23; k = 69;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 23; j = 24; k = 69;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 24; j = 25; k = 68;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 25; j = 26; k = 68;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));
i = 26; j = 16; k = 68;
faceTriangles.push_back(Triangle(fPoints[i], fPoints[j], fPoints[k], i, j, k));

```

}


```

void MFace::drawFace(cv::Mat img)
{
    for (int i = 0; i < faceTriangles.size(); i++)
        faceTriangles[i].draw(img, cvScalar(0, 255, 0));
}

void MFace::sortByZ()
{
    //sort(faceTriangles.begin(), faceTriangles.end(), &trl::Triangle::compareByZ);
}

int MFace::maxST()
{
    int maxSTind = 0;
    double maxST = faceTriangles[0].sT;

    for (int i = 1; i < faceTriangles.size(); i++)
        if (faceTriangles[i].sT > maxST)
        {
            maxST = faceTriangles[i].sT;
            maxSTind = i;
        }

    return maxSTind;
}

int MFace::maxLN()
{
    int maxLNind = 0;
    double maxLN = faceTriangles[0].l12;

    for (int i = 0; i < faceTriangles.size(); i++)
    {
        if (faceTriangles[i].l12 > maxLN)
        {
            maxLN = faceTriangles[i].l12;
            maxLNind = i;
        } else
        if (faceTriangles[i].l13 > maxLN)
        {
            maxLN = faceTriangles[i].l13;
            maxLNind = i;
        } else
        if (faceTriangles[i].l23 > maxLN)
        {
            maxLN = faceTriangles[i].l23;
            maxLNind = i;
        }
    }

    return maxLNind;
}

```

```

void MFace::break2Trs(int num)
{
    // точки выбранного треугольника
    cv::Point3f ptN1 = faceTriangles[num].pt1;
    cv::Point3f ptN2 = faceTriangles[num].pt2;
    cv::Point3f ptN3 = faceTriangles[num].pt3;
    int ixN1 = faceTriangles[num].ix1;
    int ixN2 = faceTriangles[num].ix2;
    int ixN3 = faceTriangles[num].ix3;

    // новая точка
    cv::Point3f newPT;
    int newIX = faceTriangles.size() + 1;

    // индексы точек для поиска смежного треугольника
    int ixNad1, ixNad2;
    double mLength;

    // поиск наибольшей стороны
    if ((faceTriangles[num].l12 > faceTriangles[num].l13) && (faceTriangles[num].l12 > face-
Triangles[num].l23))
    {
        // если сторона 1-2
        mLength = faceTriangles[num].l12;
        ixNad1 = ixN1;
        ixNad2 = ixN2;
        newPT.x = (ptN1.x + ptN2.x) / 2;
        newPT.y = (ptN1.y + ptN2.y) / 2;
        coordsM[newIX].x = (double)(coordsM[ixN1].x + coordsM[ixN2].x) / 2;
        coordsM[newIX].y = (double)(coordsM[ixN1].y + coordsM[ixN2].y) / 2;
        newPT.z = Model3DM[coordsM[newIX].x][coordsM[newIX].y] * coefM;
        faceTriangles.push_back(Triangle(ptN1, ptN3, newPT, ixN1, ixN3, newIX));
        faceTriangles.push_back(Triangle(ptN2, ptN3, newPT, ixN2, ixN3, newIX));
    }
    else
    if (faceTriangles[num].l13 > faceTriangles[num].l23)
    {
        // если сторона 1-3
        mLength = faceTriangles[num].l13;
        ixNad1 = ixN1;
        ixNad2 = ixN3;
        newPT.x = (ptN1.x + ptN3.x) / 2;
        newPT.y = (ptN1.y + ptN3.y) / 2;
        coordsM[newIX].x = (double)(coordsM[ixN1].x + coordsM[ixN3].x) / 2;
        coordsM[newIX].y = (double)(coordsM[ixN1].y + coordsM[ixN3].y) / 2;
        newPT.z = Model3DM[coordsM[newIX].x][coordsM[newIX].y] * coefM;
        faceTriangles.push_back(Triangle(ptN1, ptN2, newPT, ixN1, ixN2, newIX));
        faceTriangles.push_back(Triangle(ptN2, ptN3, newPT, ixN2, ixN3, newIX));
    }
    else
    {

```

```

// если сторона 2-3
mLength = faceTriangles[num].l23;
ixNad1 = ixN2;
ixNad2 = ixN3;
newPT.x = (ptN3.x + ptN2.x) / 2;
newPT.y = (ptN3.y + ptN2.y) / 2;
coordsM[newIX].x = (double)(coordsM[ixN3].x + coordsM[ixN2].x) / 2;
coordsM[newIX].y = (double)(coordsM[ixN3].y + coordsM[ixN2].y) / 2;
newPT.z = Model3DM[coordsM[newIX].x][coordsM[newIX].y] * coefM;
faceTriangles.push_back(Triangle(ptN1, ptN3, newPT, ixN1, ixN3, newIX));
faceTriangles.push_back(Triangle(ptN1, ptN2, newPT, ixN1, ixN2, newIX));
}
// удаление исходного треугольника
faceTriangles.erase(faceTriangles.begin() + num);

// поиск смежного треугольника
for (int i = 0; i < faceTriangles.size(); i++)
{
    if (faceTriangles[i].ix1 == ixNad1 || faceTriangles[i].ix2 == ixNad1 || faceTriangles[i].ix3 == ixNad1)
        if (faceTriangles[i].ix1 == ixNad2 || faceTriangles[i].ix2 == ixNad2 || faceTriangles[i].ix3 == ixNad2)
        {
            ptN1 = faceTriangles[i].pt1;
            ptN2 = faceTriangles[i].pt2;
            ptN3 = faceTriangles[i].pt3;
            ixN1 = faceTriangles[i].ix1;
            ixN2 = faceTriangles[i].ix2;
            ixN3 = faceTriangles[i].ix3;

            // поиск общей стороны
            if (fabs(mLength - faceTriangles[i].l12) < 0.0001)
            {
                // сторона 1-2
                faceTriangles.push_back(Triangle(ptN1, ptN3, newPT, ixN1, ixN3,
newIX));
                faceTriangles.push_back(Triangle(ptN2, ptN3, newPT, ixN2, ixN3,
newIX));
            }
            else
            if (fabs(mLength - faceTriangles[i].l13) < 0.0001)
            {
                // если сторона 1-3
                faceTriangles.push_back(Triangle(ptN1, ptN2, newPT, ixN1, ixN2,
newIX));
                faceTriangles.push_back(Triangle(ptN2, ptN3, newPT, ixN2, ixN3,
newIX));
            }
            else
            {
                // если сторона 2-3

```

```

        faceTriangles.push_back(Triangle(ptN1, ptN3, newPT, ixN1, ixN3,
newIX));
        faceTriangles.push_back(Triangle(ptN1, ptN2, newPT, ixN1, ixN2,
newIX));
    }
    // удаление исходного треугольника
    faceTriangles.erase(faceTriangles.begin() + i);
    break;
}
}
}

void MFace::breakATrs(int count)
{
    int i = 0;
    while (faceTriangles.size() < count)
    {
        break2Trs(maxLN());
        i = faceTriangles.size();
    }
}

void MFace::rotation(cv::Point3f centerP, double alpha, double beta)
{
    // смещение начала координат в точку-центр поворота
    for (int i = 0; i < faceTriangles.size(); i++)
    {
        faceTriangles[i].pt1.x = faceTriangles[i].pt1.x - centerP.x;
        faceTriangles[i].pt2.x = faceTriangles[i].pt2.x - centerP.x;
        faceTriangles[i].pt3.x = faceTriangles[i].pt3.x - centerP.x;

        faceTriangles[i].pt1.y = faceTriangles[i].pt1.y - centerP.y;
        faceTriangles[i].pt2.y = faceTriangles[i].pt2.y - centerP.y;
        faceTriangles[i].pt3.y = faceTriangles[i].pt3.y - centerP.y;

        faceTriangles[i].pt1.z = faceTriangles[i].pt1.z - centerP.z;
        faceTriangles[i].pt2.z = faceTriangles[i].pt2.z - centerP.z;
        faceTriangles[i].pt3.z = faceTriangles[i].pt3.z - centerP.z;
    }

    // Поворот по x
    for (int i = 0; i < faceTriangles.size(); i++)
    {
        cv::Point3f rotP;
        rotP.x = faceTriangles[i].pt1.x;
        rotP.y = faceTriangles[i].pt1.y * cos(beta) - faceTriangles[i].pt1.z * sin(beta);
        rotP.z = faceTriangles[i].pt1.y * sin(beta) + faceTriangles[i].pt1.z * cos(beta);
        faceTriangles[i].pt1 = rotP;

        rotP.x = faceTriangles[i].pt2.x;
        rotP.y = faceTriangles[i].pt2.y * cos(beta) - faceTriangles[i].pt2.z * sin(beta);
        rotP.z = faceTriangles[i].pt2.y * sin(beta) + faceTriangles[i].pt2.z * cos(beta);
    }
}

```

```

        faceTriangles[i].pt2 = rotP;

        rotP.x = faceTriangles[i].pt3.x;
        rotP.y = faceTriangles[i].pt3.y * cos(beta) - faceTriangles[i].pt3.z * sin(beta);
        rotP.z = faceTriangles[i].pt3.y * sin(beta) + faceTriangles[i].pt3.z * cos(beta);
        faceTriangles[i].pt3 = rotP;
    }

    // Поворот по y
    for (int i = 0; i < faceTriangles.size(); i++)
    {
        cv::Point3f rotP;
        rotP.x = faceTriangles[i].pt1.x * cos(alpha) + faceTriangles[i].pt1.z * sin(alpha);
        rotP.y = faceTriangles[i].pt1.y;
        rotP.z = faceTriangles[i].pt1.x * (-sin(alpha)) + faceTriangles[i].pt1.z * cos(alpha);
        faceTriangles[i].pt1 = rotP;

        rotP.x = faceTriangles[i].pt2.x * cos(alpha) + faceTriangles[i].pt2.z * sin(alpha);
        rotP.y = faceTriangles[i].pt2.y;
        rotP.z = faceTriangles[i].pt2.x * (-sin(alpha)) + faceTriangles[i].pt2.z * cos(alpha);
        faceTriangles[i].pt2 = rotP;

        rotP.x = faceTriangles[i].pt3.x * cos(alpha) + faceTriangles[i].pt3.z * sin(alpha);
        rotP.y = faceTriangles[i].pt3.y;
        rotP.z = faceTriangles[i].pt3.x * (-sin(alpha)) + faceTriangles[i].pt3.z * cos(alpha);
        faceTriangles[i].pt3 = rotP;
    }

    // Координаты в пиксели
    for (int i = 0; i < faceTriangles.size(); i++)
    {
        faceTriangles[i].pt1.x = faceTriangles[i].pt1.x + centerP.x;
        faceTriangles[i].pt1.y = faceTriangles[i].pt1.y + centerP.y;
        faceTriangles[i].pt1.z = faceTriangles[i].pt1.z + centerP.z;

        faceTriangles[i].pt2.x = faceTriangles[i].pt2.x + centerP.x;
        faceTriangles[i].pt2.y = faceTriangles[i].pt2.y + centerP.y;
        faceTriangles[i].pt2.z = faceTriangles[i].pt2.z + centerP.z;

        faceTriangles[i].pt3.x = faceTriangles[i].pt3.x + centerP.x;
        faceTriangles[i].pt3.y = faceTriangles[i].pt3.y + centerP.y;
        faceTriangles[i].pt3.z = faceTriangles[i].pt3.z + centerP.z;
    }
}

void MFace::rotationZ(cv::Point3f centerP, double alpha, double beta)
{
    // смещение начала координат в точку-центр поворота
    for (int i = 0; i < faceTriangles.size(); i++)
    {
        faceTriangles[i].pt1.x = faceTriangles[i].pt1.x - centerP.x;
        faceTriangles[i].pt2.x = faceTriangles[i].pt2.x - centerP.x;

```

```

        faceTriangles[i].pt3.x = faceTriangles[i].pt3.x - centerP.x;

        faceTriangles[i].pt1.y = faceTriangles[i].pt1.y - centerP.y;
        faceTriangles[i].pt2.y = faceTriangles[i].pt2.y - centerP.y;
        faceTriangles[i].pt3.y = faceTriangles[i].pt3.y - centerP.y;

        faceTriangles[i].pt1.z = faceTriangles[i].pt1.z - centerP.z;
        faceTriangles[i].pt2.z = faceTriangles[i].pt2.z - centerP.z;
        faceTriangles[i].pt3.z = faceTriangles[i].pt3.z - centerP.z;
    }

    // Поворот по x
    for (int i = 0; i < faceTriangles.size(); i++)
    {
        cv::Point3f rotP;
        rotP.x = faceTriangles[i].pt1.x;
        rotP.y = faceTriangles[i].pt1.y;
        rotP.z = (faceTriangles[i].pt1.y * sin(beta) + faceTriangles[i].pt1.z * sin(beta) *
sin(beta)) / cos(beta) + faceTriangles[i].pt1.z * cos(beta);
        faceTriangles[i].pt1 = rotP;

        rotP.x = faceTriangles[i].pt2.x;
        rotP.y = faceTriangles[i].pt2.y;
        rotP.z = (faceTriangles[i].pt2.y * sin(beta) + faceTriangles[i].pt2.z * sin(beta) *
sin(beta)) / cos(beta) + faceTriangles[i].pt2.z * cos(beta);
        faceTriangles[i].pt2 = rotP;

        rotP.x = faceTriangles[i].pt3.x;
        rotP.y = faceTriangles[i].pt3.y;
        rotP.z = (faceTriangles[i].pt3.y * sin(beta) + faceTriangles[i].pt3.z * sin(beta) *
sin(beta)) / cos(beta) + faceTriangles[i].pt3.z * cos(beta);
        faceTriangles[i].pt3 = rotP;
    }

    // Поворот по y
    for (int i = 0; i < faceTriangles.size(); i++)
    {
        cv::Point3f rotP;
        rotP.x = faceTriangles[i].pt1.x;
        rotP.y = faceTriangles[i].pt1.y;
        rotP.z = (faceTriangles[i].pt1.z * sin(alpha) * sin(alpha) - faceTriangles[i].pt1.x *
sin(alpha)) / cos(alpha) + faceTriangles[i].pt1.z * cos(alpha);
        faceTriangles[i].pt1 = rotP;

        rotP.x = faceTriangles[i].pt2.x;
        rotP.y = faceTriangles[i].pt2.y;
        rotP.z = (faceTriangles[i].pt2.z * sin(alpha) * sin(alpha) - faceTriangles[i].pt2.x *
sin(alpha)) / cos(alpha) + faceTriangles[i].pt2.z * cos(alpha);
        faceTriangles[i].pt2 = rotP;

        rotP.x = faceTriangles[i].pt3.x;
        rotP.y = faceTriangles[i].pt3.y;
    }

```

```

        rotP.z = (faceTriangles[i].pt3.z * sin(alpha) * sin(alpha) - faceTriangles[i].pt3.x *
sin(alpha)) / cos(alpha) + faceTriangles[i].pt3.z * cos(alpha);
        faceTriangles[i].pt3 = rotP;
    }

    // Координаты в пиксели
    for (int i = 0; i < faceTriangles.size(); i++)
    {
        faceTriangles[i].pt1.x = faceTriangles[i].pt1.x + centerP.x;
        faceTriangles[i].pt1.y = faceTriangles[i].pt1.y + centerP.y;
        faceTriangles[i].pt1.z = faceTriangles[i].pt1.z + centerP.z;

        faceTriangles[i].pt2.x = faceTriangles[i].pt2.x + centerP.x;
        faceTriangles[i].pt2.y = faceTriangles[i].pt2.y + centerP.y;
        faceTriangles[i].pt2.z = faceTriangles[i].pt2.z + centerP.z;

        faceTriangles[i].pt3.x = faceTriangles[i].pt3.x + centerP.x;
        faceTriangles[i].pt3.y = faceTriangles[i].pt3.y + centerP.y;
        faceTriangles[i].pt3.z = faceTriangles[i].pt3.z + centerP.z;
    }
}

```

Приложение 2. Справка о внедрении результатов научно-исследовательской работы в практику



Ministry of Education and Science of the Russian Federation
Federal State Autonomous Educational Institution of Higher Education
"National Research Tomsk Polytechnic University" (TPU)
30, Lenin ave., Tomsk, 634050, Russia
Tel. +7-3822-606333, +7-3822-701779,
Fax +7-3822-563865, e-mail: tpu@tpu.ru, tpu.ru
ОКПО (National Classification of Enterprises and Organizations):
02069303,
Company Number: 1027000890168,
VAT / KPP (Code of Reason for Registration)
7018007264/701701001, BIC 046902001

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский
Томский политехнический университет» (ТПУ)
Ленина, пр., д. 30, г. Томск, 634050, Россия
тел.: +7-3822-606333, +7-3822-701779,
факс +7-3822-563865, e-mail: tpu@tpu.ru, tpu.ru
ОКПО 02069303, ОГРН 1027000890168,
ИНН/КПП 7018007264/701701001, БИК 046902001

СПРАВКА

о внедрении результатов научно-исследовательской работы в практику

На основе полученных результатов научно-исследовательской работы «Технология подготовки изображений лиц из видеопотока к распознаванию личности в режиме реального времени» соискателя Небабы С.Г. разработан модуль учебной дисциплины по основам обработки изображений и распознаванию образов для студентов специальности Б3.В11 «Интеллектуальные и информационные системы», бакалавриат, направления 09.03.03 «Прикладная информатика» кафедры Программной Инженерии ИК ТПУ.

Доцент
Кафедры Программной инженерии
Института кибернетики ТПУ, к.т.н.

С.В. Аксенов

И.о. руководителя
Кафедры Программной инженерии
Института кибернетики ТПУ, к.т.н.

М.А. Иванов

Начальник
учебно-методического отдела
Института кибернетики ТПУ, к.т.н.

О.М. Гергет

Приложение 3. Акт о внедрении научных и практических результатов

УТВЕРЖДАЮ:

Директор



ООО «Научно-производственная компания «Техника дела»

Ю.П. Груздев

06 2017 г.

АКТ

о внедрении научных и практических результатов

г. Томск

14.06.2017 г.

Мы, нижеподписавшиеся: Андреев Сергей Юрьевич, ведущий инженер; Немтинов Дмитрий Валентинович, главный инженер - подтверждаем, что алгоритмическое и программное обеспечение системы «RTV Face Classifier» (на ключевые алгоритмы которой получено свидетельство о государственной регистрации программ для ЭВМ «Создание набора изображений лиц для алгоритмов распознавания») и техническая документация в электронном виде внедрены в ООО «Научно-производственная компания «Техника дела».

Программное обеспечение «RTV Face Classifier» использовалось в ООО «Научно-производственная компания «Техника дела» для построения набора ракурсов на основе одного фронтального изображения лица. Внедренный программный продукт позволил формировать входные данные для системы распознавания лиц в виде набора изображений с вариациями углов отклонения фронтальной плоскости лица в диапазоне 20 градусов по углу места и азимуту. Применение сформированных изображений в системе распознавания позволило повысить точность идентификации личности, снизив среднюю ошибку EER системы с 2,4% до 1,9%, а алгоритм трекинга позволил повысить частоту кадров, обрабатываемых в режиме реального времени, с 15 к/с до 20 к/с.

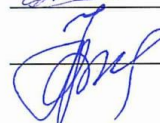
Научный руководитель проекта: д.т.н. Захарова А.А.

Исполнитель проекта: Небаба С.Г.

Ведущий инженер

 С.Ю. Андреев

Главный инженер

 Д.В. Немтинов

Приложение 4. Свидетельство о государственной регистрации программы для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2017611032

**Создание набора изображений лиц для алгоритмов
распознавания**

Правообладатель: *федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический
университет» (RU)*

Авторы: *Небаба Степан Геннадьевич (RU),
Захарова Алена Александровна (RU)*

Заявка № **2016663098**
Дата поступления **30 ноября 2016 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ **19 января 2017 г.**

Руководитель Федеральной службы
по интеллектуальной собственности

 **Г.П. Ивлиев**

