

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(**Н И У « Б е л Г У »**)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК

КАФЕДРА МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ
ИНФОРМАЦИОННЫХ СИСТЕМ

**ВЫСОКОПРОИЗВОДИТЕЛЬНЫЙ АЛГОРИТМ ИДЕНТИФИКАЦИИ
ЛЮДЕЙ ПО РАДУЖНОЙ ОБОЛОЧКЕ ГЛАЗА**

Выпускная квалификационная работа
обучающегося по направлению подготовки 02.04.01 Математика и
компьютерные науки очной формы обучения, группы 07001631
Сельдемирова Владимира Евгеньевича

Научный руководитель
к.т.н., доцент
Михелев В.М.

Рецензент
к.т.н., доцент
Черноморец А.А.

БЕЛГОРОД 2018

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 СОВРЕМЕННЫЕ ПОДХОДЫ И ПРОБЛЕМЫ ИДЕНТИФИКАЦИИ ПО РАДУЖНОЙ ОБОЛОЧКЕ ГЛАЗА.....	7
1.1 Глаз как объект распознавания.....	7
1.2 Обзор исследований.....	8
1.3 Общая схема процесса идентификации.....	10
1.4 Получение изображений.....	12
1.5 Сегментация и нормализация.....	15
2 МАТЕМАТИЧЕСКОЕ ОПИСАНИЕ МЕТОДОВ СЕГМЕНТАЦИИ.....	18
2.1 Оператор Кэнни.....	18
2.2 Интегро-дифференциальный оператор.....	21
2.3 Преобразование Хафа.....	23
2.4 Преобразование Даугмана из декартовых в полярные координаты.....	26
2.5 Расстояние Хэмминга.....	28
3 РАЗРАБОТКА АЛГОРИТМА.....	30
3.1 Предварительная обработка изображения.....	30
3.2 Околокруговой активный контур зрачка.....	31
3.3 Эллиптический контур век.....	35
3.4 Итеративный алгоритм определение лимба и век.....	36
3.5 Контурный метод развёртки радужки.....	38
3.6 Определение и маскирование ресниц на изображении.....	42
3.7 Параллельный подход.....	45
4 ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ.....	50
4.1 Используемые аппаратно-программные средства.....	50
4.2 Критерии точности идентификации.....	53
4.3 Результаты тестирования.....	54
ЗАКЛЮЧЕНИЕ.....	57
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	58

ВВЕДЕНИЕ

Биометрические технологии связаны с распознаванием личности людей на основе их уникальных физических или поведенческих характеристик [1]. Физические характеристики, такие как отпечаток пальца, отпечаток ладони, узоры радужной оболочки глаза или поведенческие характеристики, такие как манера печати на клавиатуре и рукописная подпись, представляют уникальную информацию о человеке, и могут использоваться в приложениях аутентификации.

Достижения науки и техники позволили использовать биометрию в прикладных программах, где требуется установить или подтвердить личность человека. Такие области применения, как контроль пассажиров в аэропортах, контроль доступа в зонах ограниченного доступа, пограничный контроль, доступ к базам данных и финансовым услугам, являются примерами применения биометрических технологий для более надёжной идентификации и проверки.

За последние годы в некоторых странах были выпущены биометрические удостоверения личности и паспорта для улучшения процесса пограничного контроля [2] и упрощения пассажирских перелётов.

В сфере финансовых услуг, биометрические технологии показали большой потенциал в предоставлении большего удобства для клиентов при одновременном повышении их безопасности. Например, банковские услуги и платежи на основе биометрии будут намного безопаснее, быстрее и проще, чем существующие методы, основанные на кредитных и дебетовых картах.

Кроме того, приложения контроля доступа, такие как доступ к базе данных и вход в систему компьютера, также выигрывают от новых предлагаемых технологий. По сравнению с паролями, биометрические технологии предлагают более безопасный и удобный доступ и устраняют такие проблемы, как забывание или взлом паролей.

К преимуществам использования РОГ для идентификации можно отнести:

1. Универсальность

Идентификация по радужной оболочке глаза подходит большинству людей и имеет низкий коэффициент отказа в регистрации (коэффициент, при котором попытки создать шаблон из входных данных безуспешны).

2. Уникальность

Образцы радужной оболочки весьма разнообразны, и считаются уникальными для каждого человека. Узоры формируются случайным образом во время эмбрионального беременности, поэтому даже генетически идентичные близнецы имеют разные узоры радужной оболочки глаза.

3. Постоянство

Радужная оболочка имеет сильное преимущество в том, что она является внутренним органом, и, таким образом, хорошо защищена, но при этом видима извне. Кроме того, радужная оболочка не меняется с возрастом, за исключением повреждения в результате несчастного случая или болезни.

4. Простота регистрации

Форма радужной оболочки почти полностью плоская и, таким образом, достаточно постоянна, гораздо сильнее, чем лицо. Кроме того, изображение может быть снято на расстоянии от 10 см до метра. Поэтому не требуется дорогая 3D-камеры, чтобы делать снимки под разными углами, как,

например, в процессе распознавании лица. Проблема, однако, может заключаться в освещении, потому что глаза являются отражающими поверхностями. Кроме того, радужная оболочка глаза часто частично закрыта веками, но это компенсируется согласно используемого алгоритма.

По этим причинам использование радужной оболочки становится интересным как альтернативный подход к надёжному визуальному распознаванию людей, когда идентификация может быть выполнена на расстоянии менее метра, и особенно при необходимости поиска по очень большой базе данных без каких-либо ложных совпадений.

В целом, будущее биометрической технологии, как полагают, открыто для большего количества инвестиций, основанных на новых услугах, которые она может предложить обществу. Обзор рынка, представленный Международной биометрической группой (IBG), показывает, что доходы биометрической отрасли, постоянно росли с 1538.9 миллионов долларов США в 2005 году до 5749.2 миллиона долларов США в 2010 году [3].

Актуальность. В области цифровой обработки изображений радужной оболочки глаза, было выявлено, согласно наблюдениям, что в существующих методах, окружность не может эффективно моделировать границу радужки, что и является мотивацией для создания точного алгоритма обнаружения радужки, основанного на концепции активных контуров.

Целью исследования является разработка алгоритма, выполняющего сегментацию и нормализацию изображений радужной оболочки глаза. Для достижения поставленной цели нужно решить следующие задачи:

1. Изучить современные методы по теме локализации, сегментации и нормализации радужной оболочки глаза;

2. Разработать математическую модель локализации век при помощи метода эллиптического активного контура и модель локализации и сегментации РОГ при помощи метода околоокружного активного контура;
3. Разработать алгоритм локализации век при помощи метода эллиптического активного контура и алгоритм локализации и сегментации РОГ при помощи метода околоокружного активного контура;
4. Оценить работоспособность разработанного алгоритма на основе вычислительных экспериментов

Объект и предмет исследования. Объект исследования — снимки глаза, выполненные в ближней инфракрасном диапазоне. Предмет исследования — алгоритм сегментации и нормализации изображений радужной оболочки глаза.

Методы исследования. В работе были использованы методы цифровой обработки изображений, геометрии, линейной алгебры.

Научная новизна заключается:

1. Метод нахождения центра и контура РОГ с помощью околоокружного активного контура;
2. Метод нахождения век с помощью эллиптического активного контура;

Практическая значимость определяется возможностью применения алгоритма в системах аутентификации.

Структура и объем. Выпускная квалификационная работа состоит из введения, 4 глав, заключения, списка использованной литературы. Общий объем составляет 61 страница, 13 рисунков, 1 таблица.

1 СОВРЕМЕННЫЕ ПОДХОДЫ И ПРОБЛЕМЫ ИДЕНТИФИКАЦИИ ПО РАДУЖНОЙ ОБОЛОЧКЕ ГЛАЗА

1.1 Глаз как объект распознавания

Узор радужной оболочки образован комбинированными слоями пигментированных эпителиальных клеток, мышцами для управления зрачком, стромальным слоем, состоящим из соединительной ткани, кровеносных сосудов и переднего пограничного слоя [4]. Внешнее строение глаза показано на рисунке 1.1.

Чтобы оценить богатство радужки как объекта для распознавания, полезно рассмотреть её структуру немного подробнее. Радужная оболочка состоит из нескольких слоёв. Её задняя поверхность состоит из сильно пигментированных эпителиальных клеток. Перед этим слоем находятся две совмещённые мышцы, управляющие зрачком. Далее идёт стромальный слой, состоящий из коллагеновой соединительной ткани. Самым передним слоем является передний пограничный слой, отличающийся от стромального слоя более высокой плотностью пигментных клеток, называемыми хроматофорами [5]. Внешний вид радужной оболочки является прямым результатом её многослойной структуры. Цвет радужки обусловлен поглощением света, падающего на пигментированные клетки на переднем слое.

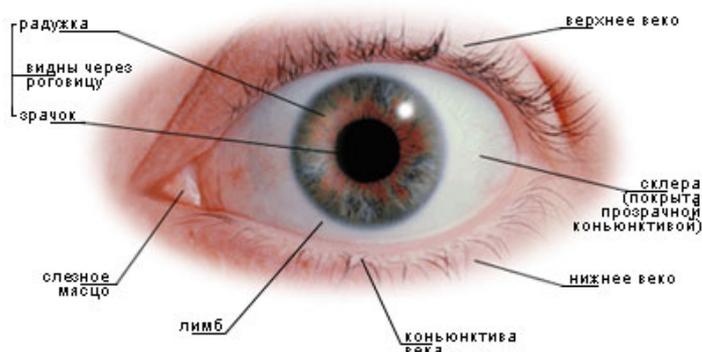


Рис. 1.1. Внешнее строение глаза

Физиологическая сложность этого органа приводит к случайным закономерностям в рисунке радужной оболочки, которые являются статистически уникальными и подходят для биометрических измерений [6]. Кроме того, текстура радужной оболочки стабильна с течением времени, и только незначительные изменения происходят с ней на протяжении всей жизни человека [7]. Это также внутренний орган, расположенный за роговицей и водянистой влагой глаза, и хорошо защищённый от внешней среды. Такие характеристики, как защита от воздействия окружающей среды и более надёжная стабильность с течением времени, по сравнению с другими популярными биометрическими модальностями, хорошо оправдывают проводимые исследования и инвестиции в распознавание радужной оболочки глаза различными исследователями и отраслями по всему миру [8].

1.2 Обзор исследований

История распознавания радужки восходит к середине 19-го века, когда французский врач Альфонс Бертильон изучал использование цвета глаз в

качестве идентификатора [9]. Тем не менее, считается, что основная идея использования узоров радужной оболочки глаза для идентификации, как мы знаем её сегодня, была впервые представлена глазным хирургом, Фрэнком Берчем, в 1936 году [10]. В 1987 году два офтальмолога, Флом и Сафир, запатентовали эту идею [11] и предложили Даугману, профессору Гарвардского университета, изучить возможность разработки алгоритма распознавания по радужной оболочке глаза. После нескольких лет научных экспериментов Даугман предложил и разработал систему распознавания радужной оболочки глаза и опубликовал результаты в 1993 году [12].

Через несколько лет после публикации алгоритма Даугманом другие исследователи разработали новые алгоритмы распознавания радужной оболочки глаза. Системы были представлены Уальдсом [13], Боулзом [14], Тиссэ и др. [15], Жу и др. [16], Лимом и др. [17], Ноем и др. [18].

Хотя алгоритм Даугмана может выполнять распознавание радужки, некоторые сведения об алгоритме не публикуются. Например, для нормализации радужной оболочки глаза на основе полярного преобразования требуется соответствующая точка отсчёта в качестве центра. В этой работе изучается сегментация и нормализация радужки. Оцениваются и сравниваются предыдущие работы по сегментации и нормализации. Кроме того, предлагаются улучшения методов сегментации и нормализации, которые повышают производительность алгоритма распознавания.

Общая эффективность системы распознавания радужки зависит от производительности её подсистем. Качество полученных изображений, сегментация, нормализация и извлечение признаков, в целом, определяют производительность системы. Например, Лим сообщает о достижении успеха лишь в 88,2% на более чем 6000 изображениях радужной оболочки на этапе предварительной обработки из-за нежелательных факторов, таких как

закрытие радужки веками, тень от ресниц и шумов в области зрачков. Такое ухудшение этапа предварительной обработки можно рассматривать как ухудшение общей производительности системы [19].

1.3 Общая схема процесса идентификации

Все существующие в настоящее время методы автоматического распознавания радужки глаза по её изображениям реализуют следующую схему (рисунок 1.2.):

- регистрация изображений глаза;
- выделение области радужки на изображении;
- вычисление признаков и формирование из них эталона радужки;
- сравнение двух наборов признаков.

Регистрация изображения глаза производится специализированной камерой, работающей в ближнем инфракрасном диапазоне (от 850 до 900 нм). Камера снабжена полосовым фильтром, гасящим области спектра вне пределов этой полосы и инфракрасным осветителем, имеющим узкий спектральный максимум в этой полосе [20]. В совокупности хорошо подобранная комбинация фильтра и осветителя обеспечивает преобладание в световом потоке, поступающем на сенсор камеры, яркости осветителя над посторонними засветками от иных источников света [21]. В результате блики от посторонних предметов, отражающиеся в роговице и перекрывающие рисунок радужки, подавляются. При этом сохраняется безопасный для здоровья уровень мощности ИК излучения.

Выделение области радужки на изображении (сегментация), по сути, это поиск на изображении относительно тёмного объекта, близкого по форме

к кругу, содержащего внутри себя концентрический более тёмный объект (зрачок). Данная задача может быть решена многими способами, например, поиск концентрических окружностей посредством преобразования Хафа или использования коррелятора для поиска заданной формы с последующим обнаружением контуров зрачка и радужки [22]. Специфичным для задачи сегментации РОГ является наличие век, в большинстве случаев закрывающих верхнюю и нижнюю её части. Некоторые системы выделяют веки явным образом и отбрасывают ложные данные с закрытых участков.

Построение эталона радужки состоит из двух основных этапов: нормирования её изображения и вычисления информативных признаков на нормированном изображении. Нормирование размеров изображения радужки необходимо по двум причинам: различие масштабов снимков и изменение относительного размера и формы зрачка.

Создание записи для базы данных — техническая процедура формирования записи из собственно эталона, времени регистрации и прочих данных, существенных для конкретного приложения.

Сравнение/поиск среди ранее зарегистрированных. Два основных сценария работы биометрической распознающей системы — верификация и идентификация. Верификация — проверка того, что созданный эталон принадлежит конкретному человеку, ранее зарегистрированному в базе данных. Идентификация — поиск персоны, которой может принадлежать эталон, среди ранее зарегистрированных.

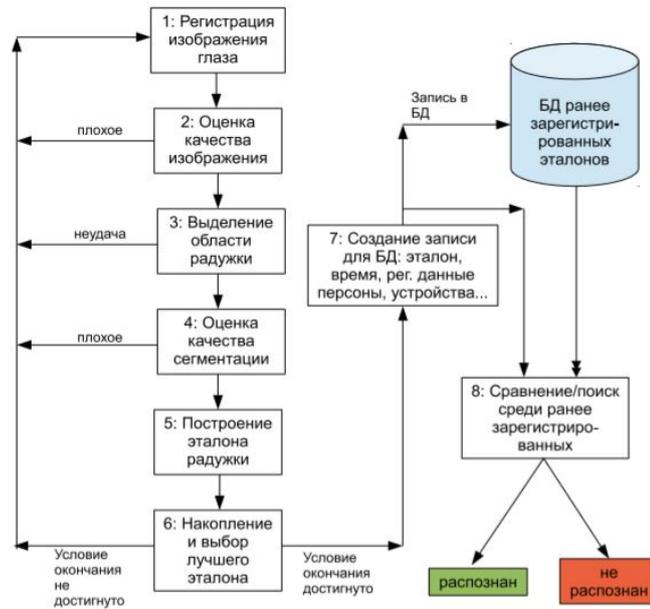


Рис. 1.2. Общая схема процесса идентификации

1.4 Получение изображений

Производительность и результат работы алгоритма сегментации радужной оболочки зависят от изначальных условий съёмки изображения [23]. Это связано с тем, что характеристики изображения радужки могут различаться при различных способах получения изображений. Два основных фактора получения изображения, которые могут влиять на сегментацию радужной оболочки:

1. Внешний вид радужки может варьироваться в зависимости от длины волны, используемой для освещения при получении изображения. Это вызвано реакцией меланина, окрашивающего пигмента, на различные длины волн. Образцы радужной оболочки, полученные при различных длинах волн освещения представлены на рис. 1.3.

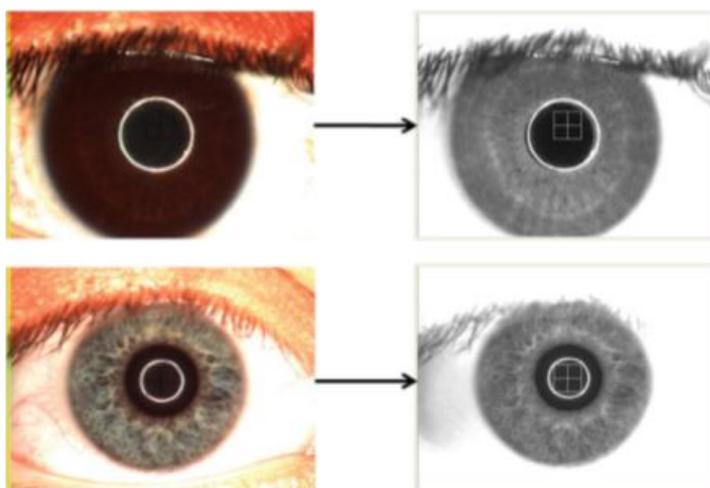


Рис. 1.3. Снимки радужной оболочки при различных длинах волн освещения

2. Вид радужки на изображении также зависит от характеристик используемого прибора (например, разрешения изображения, фокусного расстояния, глубины поля, и конфигурации источника освещения, оптики и поляризационных фильтров, расстояния). На рисунке 1.4. показаны изображения радужной оболочки, полученные с использованием двух разных приборов.

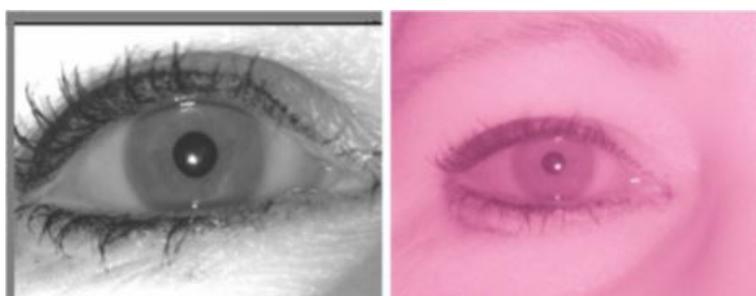


Рис. 1.4. Изображения радужной оболочки, полученные с использованием двух разных приборов

Системы распознавания радужки обычно работают в ближнем

инфракрасном диапазоне. Это связано с тем, что такой диапазон гарантирует, что полученное изображение будет содержать информацию, связанную с текстурой радужной оболочки, а не с её пигментацией. Это свойство, в свою очередь, способствует высокой эффективности распознавания радужной оболочки, снятой в ближнем инфракрасном диапазоне.

Для изображения радужной оболочки, полученного при почти идеальных условиях, сегментация радужной оболочки может быть выполнена с использованием простых методов обработки изображений. Однако, когда изображение радужки получено в неидеальных условиях, сегментация становится сложной задачей. Ниже перечислены некоторые факторы, которые вызывают сложность сегментации радужной оболочки:

1. Оклюзии, вызванные анатомическими особенностями глаза: одной из наиболее распространённых и значительных проблем, возникающих при сегментации радужки, является закрытие части зрачка веками и ресницами. Чтобы получить изображение радужной оболочки, пользователь должен держать глаза открытыми в течение короткого периода времени во время получения изображения. Однако в нормальных условиях незначительная часть человеческого глаза обычно закрывается сверху и снизу, соответственно верхним и нижним веками. На рисунке 1.5. показана окклюзия веками. В таких случаях граница радужки больше не является круговой или эллиптической.

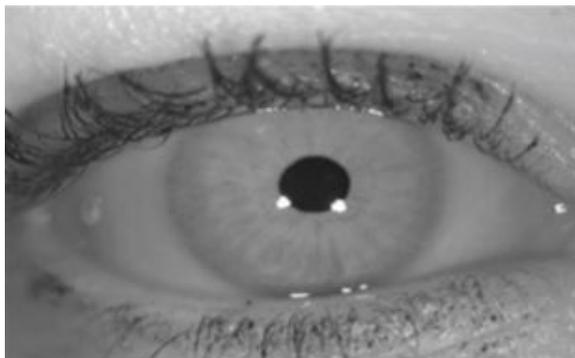


Рис. 1.5. Радужная оболочка прикрыта веками

2. Сегментация изображения радужной оболочки, полученного при плохом освещении, крайне затруднено, поскольку текстура радужной оболочки в таких изображениях может быть недостаточно подчёркнута, что приводит к низкой эффективности распознавания.

3. Блики представляют собой небольшие области на изображении радужной оболочки, характеризующиеся пикселями с высокими значениями интенсивности яркости, которые обычно вызваны неправильной направленностью источника света. Если блики присутствуют на границе радужки, сегментация радужной оболочки становится затруднительной.

В основном большинство исследований ведутся с использованием изображений, содержащихся в базах института CASIA (Chinese Academy of Sciences Institute of Automation). Базы CASIA содержат несколько разделов, но самой часто используемой базой является CASIA-Iris-Interval. Изображения из этой базы получены в ближнем инфракрасном диапазоне с разрешением 320×280 пикселей. Спектр ближнего инфракрасного излучения выделяет особенности структуры радужки, облегчая последующие этапы в процессе идентификации.

1.5 Сегментация и нормализация

При сегментации желательно отделять текстуру радужки от остальной части изображения. Радужка обычно сегментируется путём обнаружения внутренней (зрачок) и наружной границы (лимб). Хорошо известные методы, такие как интегро-дифференциальный оператор, преобразования Хафа — одни из методов обнаружения границ. Ниже приводится описание этих методов.

Правильное определение внутренних и внешних границ текстуры радужки имеет важное значение во всех системах распознавания радужки. Все предыдущие методы сегментации моделируют границы радужки и двух век с помощью простых геометрических моделей. Зрачок и радужка часто моделируются как круги, а два века моделируются как параболические дуги. Однако, согласно наблюдениям, окружность не может эффективно моделировать границу зрачка. Нерегулярная граница зрачка является мотивацией для создания точного алгоритма обнаружения зрачка, основанного на концепции активных контуров. Контурная модель учитывает, что фактические границы зрачка — почти круговой контур, а не идеальный круг. Целью данной работы является демонстрация того, что данный метод может эффективно повысить точность распознавания.

Следующим направлением этой работы является нормализация радужной оболочки глаза. Большинство методов нормализации основаны на преобразовании радужной оболочки в полярные координаты, известном как процесс разворачивания. Граница зрачка и граница радужки, как правило, являются двумя неконцентрическими контурами. Это условие приводит к различному выбору опорных точек для преобразования радужной оболочки в полярные координаты. Правильный выбор опорной точки очень важен там, где радиальная и угловая информация будет определена относительно этой точки.

В экспериментах рассматриваются характеристики нескольких опорных точек, включая центр зрачка, центр лимба и линейно угаданную центральную точку. Развёртка радужки с помощью центра зрачка предложено Лимом [24] и Болесом [25]. Линейно угаданный центр эквивалентен методике, используемой Юнгом [26]. Цель экспериментов проиллюстрировать поэтапное улучшение результатов распознавания, когда

исходная точка развёртки движется в направлении линейно угаданной центральной точки, начиная с центра лимба. Метод развёртки на основе контуров основан на предположении, что текстуры радужки не будут иметь чрезмерного радиального натяжения от границы зрачка до границы радужки.

Кроме того, большинство подходов к нормализации, основанных на декартово-полярном преобразовании, разворачивают текстуру радужной оболочки в прямоугольный блок фиксированного размера. В качестве примера в методе Лима после нахождения центра зрачка и внутренних и внешних границ радужки текстура преобразуется в полярные координаты с фиксированным разрешением. В радиальном направлении, текстура нормализуется от внутренней границы к внешней границе в 60 пикселей. Этот метод, по-видимому, полезен для приведения изображений радужной оболочки в стандартную форму, что упростит процесс извлечения признаков. Однако, круглая форма радужки подразумевает, что существует различное количество пикселей по каждому радиусу. Преобразование информации с разными радиусами в одно и то же разрешение приводит к разным интерполяциям, а иногда и к потере информации, что может ухудшить производительность системы.

Предлагается новый метод нормализации, который учитывает, что текстуры радужной оболочки имеют разное количество пикселей по каждому радиусу. Полярное разрешение радужки динамически настраивается относительно количества пикселей в каждом радиусе. Экспериментальные результаты показывают, что схема динамической нормализации работает лучше, чем метод фиксированного размера.

2 МАТЕМАТИЧЕСКОЕ ОПИСАНИЕ МЕТОДОВ СЕГМЕНТАЦИИ

2.1 Оператор Кэнни

Детектор контуров Кэнни является оператором обнаружения границ, который использует многоступенчатый алгоритм для обнаружения широкого диапазона контуров в изображениях [27]. Это метод извлечения полезной структурной информации из разных объектов на изображении и резкого уменьшения объёма данных, подлежащих обработке. Он широко применяется в различных системах компьютерного зрения. Кэнни обнаружил, что требования к применению обнаружения контуров в различных системах относительно схожи. Таким образом, обнаружение контуров для решения этих задач может быть реализовано в самых разных ситуациях.

Процесс обнаружения контуров может быть разбит на 5 шагов:

1. Применить гауссовский фильтр для сглаживания изображения, чтобы удалить шум;
2. Найти градиенты интенсивности изображения;
3. Применить подавление немаксимумов, чтобы избавиться от ложного обнаружения краёв;
4. Применение двойную пороговую фильтрацию для определения потенциальных контуров;

5. Трессировка области неоднозначности: обнаружения краёв путём подавления всех других краёв, которые являются слабыми и не связаны с сильными краями.

Фильтр Гаусса

Поскольку результаты обнаружения контура могут быть неправильными из-за шума на изображении, важно отфильтровать шум, чтобы предотвратить ложное обнаружение, вызванное шумом. Для сглаживания изображения применяется Гауссовский фильтр. Этот шаг немного сгладит изображение, чтобы уменьшить влияние шума на детектор контуров. Уравнение для ядра Гауссовского фильтра размера $(2k+1) \times (2k+1)$ задаётся:

$$H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(\frac{-(i-(k+1))^2 + (j-(k+1))^2}{2\sigma^2}\right); 1 \leq i, j \leq (2k+1) \quad (2.1)$$

Важно понимать, что выбор размера ядра сглаживания Гаусса повлияет на производительность детектора. Чем больше размер, тем ниже чувствительность детектора к шуму. Кроме того, ошибка локализации для обнаружения края будет немного увеличиваться с увеличением размера ядра гауссовского фильтра.

Градиент интенсивности

Контурные линии на изображении могут располагаться на различных направлениях, поэтому алгоритм использует четыре фильтра для обнаружения горизонтального, вертикального и диагонального краёв в размытом изображении. Он возвращает значение для первой производной в горизонтальном направлении (G_x) и в вертикальном направлении (G_y). Отсюда можно определить граничный градиент и направление:

$$\begin{aligned} G &= \sqrt{(G_x^2 + G_y^2)} \\ \Theta &= \text{atan2}(G_x, G_y) \end{aligned} \quad (2.2)$$

Угол направления контура округляется до одного из четырёх углов, представляющих вертикальную, горизонтальную и две диагонали (0° , 45° , 90° и 135°).

Подавление немаксимумов

Подавление немаксимумов применяется к "тонкому" контуру. После применения градиентного расчёта край, извлечённый из значения градиента, все ещё довольно размыт. Подавление немаксимумов может помочь подавить все значения градиента (установив их в 0), за исключением локальных максимумов, которые указывают на местоположения с самым резким изменением значения интенсивности. Алгоритм для каждого пикселя в градиентном изображении:

1. Сравните силу контура текущего пикселя с силой края пикселя в положительном и отрицательном направлениях градиента.
2. Если сила контура текущего пикселя больше по сравнению с другими пикселями в маске с тем же направлением (т. е. пиксель, который указывает в направлении y , будет сравниваться с пикселем выше и ниже его по вертикальной оси), значение будет сохранено. В противном случае, значение будет подавлено.

В некоторых реализациях алгоритм классифицирует направления непрерывного градиента на небольшой набор дискретных направлений, а затем перемещает фильтр 3×3 по выходному результату предыдущего шага. На каждом пикселе он подавляет силу края центрального пикселя (устанавливая его значение на 0), если его величина не превышает величину

двух соседних в направлении градиента.

Двойная пороговая фильтрация

После применения немаксимального подавления оставшиеся пиксели обеспечивают более точное представление реальных контуров на изображении. Тем не менее, остаются некоторые пиксели контуров, вызванные шумом и изменением цвета. Чтобы учесть эти ложные результаты, важно отфильтровать пиксели со слабым значением градиента и сохранить краевые пиксели с высоким значением градиента. Это достигается путём выбора высоких и низких пороговых значений. Если значение градиента краевого пикселя выше, чем высокое пороговое значение, оно отмечено как сильный пиксель края. Если значение градиента краевого пикселя меньше высокого порогового значения и больше, чем низкое пороговое значение, оно отмечено как слабый пиксель края. Если значение краевого пикселя меньше значения нижнего порога, оно будет подавлено. Два пороговых значения определяются эмпирически, и их определение будет зависеть от содержания данного входного изображения. Результат обработки изображения оператором Кэнни представлен на рисунке 2.1.

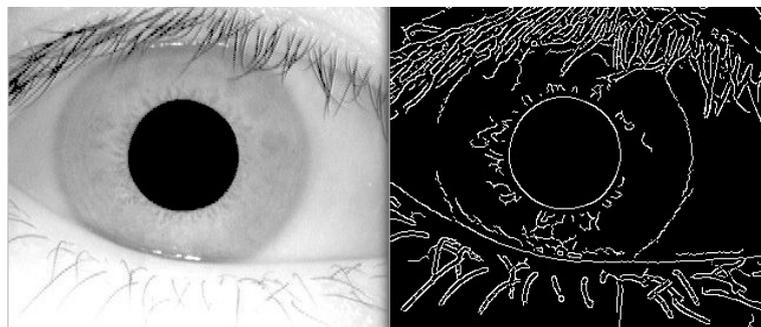


Рис. 2.1. Найденные контуры оператором Кэнни.

2.2 Интегро-дифференциальный оператор

Чтобы локализовать радужку, Даугман предложил интегро-дифференциальный оператор [28]. Оператор предполагает, что зрачок и лимб являются круговыми контурами, и выполняет функцию детектора окружностей. Обнаружение верхних и нижних век также выполняется с помощью интегро-дифференциального оператора путём настройки поиска контура с кругового на дугообразный. Интегро-дифференциальный оператор определяется как:

$$\max(r, x_0, y_0) \left| G_\sigma(r) * \frac{\partial}{\partial r} \int_{(r, x_0, y_0)} \frac{I(x, y)}{2\pi r} ds \right| \quad (2.3)$$

Оператор выполняет попиксельный поиск по всему исходному изображению и получает размытую частичную производную от интеграла по нормализованным круговым контурам с разными радиусами. Ожидается, что границы зрачка и лимба будут максимизировать производную интеграла, где значения интенсивности по круговым границам будут внезапно изменяться.

$G_\sigma(r)$ — сглаживающая функция, управляемая σ , которая сглаживает изображение для более точного поиска.

Как только изменение интенсивности пикселей по границе зрачка будет более выражено, значение σ может быть установлено для грубой свёртки. С другой стороны, когда поиск осуществляется по границе радужки, значение σ устанавливается для более тонкой свёртки. Это связано с изменением интенсивности пикселей на границе радужки.

На изображении радужки, полученном в почти идеальных условиях, обе границы радужки могут быть легко обнаружены с помощью интегро-дифференциального оператора. Однако на изображении, полученном в неидеальных условиях, граница радужки не может быть полностью круговой из-за окклюзий, вызванных веками. Поэтому при поиске границы радужки

угловая дуга интегрирования часто ограничивается левым и правым квадрантами (т. е. вблизи вертикальных краёв радужки). При поиске границы зрачка эта дуга может быть расширена, если окклюзии век относительно малы. После обнаружения обеих границ радужки границы век могут быть обнаружены путём изменения пути интегрирования от кругового к дугообразному. Несмотря на определение границ радужки с помощью интегро-дифференциального оператора, локализованная радужка может быть дополнительно закрыта другими шумными областями, такими как ресницы, тени или блики. Поэтому соответственно генерируется маска, которая регистрирует местоположение этих нежелательных окклюзий.

2.3 Преобразование Хафа

Преобразование Хафа является стандартным инструментом анализа изображений для нахождения кривых, которые могут быть определены в параметрической форме, таких как линии и круги. Определение глобальной формы достигается с помощью локальных шаблонов. Например, распознавание окружности может быть достигнуто путём рассмотрения сильных контуров на изображении в качестве локальных шаблонов и поиска максимального значения кругового преобразования Хафа [29].

Метод локализации, аналогичный методу Даугмана, также основан на первой производной изображения. В предложенном методе Уайлдса карта контуров изображения получается путём нахождения порога величины градиента интенсивности изображения:

$$|\nabla G(x, y) * I(x, y)| \quad (2.4)$$

где $\nabla \equiv (\partial/\partial x, \partial/\partial y)$ и $G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-x_0)^2 + (y-y_0)^2}{2\sigma^2}}$. $G(x, y)$ — функция сглаживания Гаусса с параметром масштабирования σ для выбора правильного масштаба для анализа контуров.

Карта контуров затем используется в процессе голосования, чтобы максимизировать определенное преобразование Хафа для желаемого контура [30]. Принимая полученные граничные точки как $(x_j, y_j), j=1, 2, \dots, n$, преобразование Хафа можно записать в виде:

$$H(x_c, y_c, r) = \sum_{j=1}^n h(x_j, y_j, x_c, y_c, r) \quad (2.5)$$

где

$$h(x_j, y_j, x_c, y_c, r) = \begin{cases} 1 & \text{если } g(x_j, y_j, x_c, y_c, r) = 0 \\ 0 & \text{в других случаях} \end{cases} \quad (2.6)$$

Лимб и зрачок моделируются как круги, а параметрическая функция g определяется как:

$$g(x_j, y_j, x_c, y_c, r) = (x_j - x_c)^2 + (y_j - y_c)^2 - r^2 \quad (2.7)$$

Предполагая, что окружность с центром (x_c, y_c) и радиусом r , граничные точки, расположенные над окружностью, приводят к нулевому значению функции. Значение g затем преобразуется в 1 функцией h , которая представляет собой локальный шаблон контура. Локальные шаблоны затем используются в процедуре голосования с использованием преобразования Хафа, H , чтобы найти правильные границы зрачка и лимба. Для обнаружения лимба используется только информация о вертикальном крае. Верхняя и нижняя части, которые содержат информацию о горизонтальном крае, обычно покрываются двумя веками. Информация о горизонтальном крае используется для обнаружения верхних и нижних век,

которые моделируются как параболические дуги вида $x(t)=a_x t^2+b_x t+c_x$ и $y(t)=a_y t^2+b_y t+c_y$ с $0 \leq t \leq 1$. Параметры a_x, b_x, c_x, a_y, b_y и c_y , снова определяются с использованием того же подхода к моделированию на основе гистограммы.

Этот же процесс используется для определения границы зрачка, но со следующими незначительными изменениями:

1. Изображение фильтруется с помощью градиентного краевого детектора, который не настроен по направлению. Это связано с тем, что зрачковая граница менее подвержена окклюзии со стороны век
2. Допустимые значения параметров (x_c, y_c, r) ограничены кругом, описывающим границу радужки.

На практике вводится аккумуляторная матрица для нахождения точки пересечения в пространстве параметров. Во-первых, нам нужно разделить пространство параметров с помощью сетки и создать аккумуляторную матрицу в соответствии с сеткой. Элемент в аккумуляторной матрице обозначает количество кругов в пространстве параметров, проходящих через соответствующую ячейку сетки в пространстве параметров. Этот номер также называется «номером для голосования». Изначально каждый элемент в матрице равен нулю. Затем для каждой точки ребра в исходном пространстве мы можем сформулировать круг в пространстве параметров и увеличить число голосов ячейки сетки, через которую проходит круг. Этот процесс называется голосованием.

После голосования, мы можем найти локальные максимумы в матрице накопителя. Положения локальных максимумов соответствуют центрам окружностей в исходном пространстве.

После обнаружения обеих границ радужной оболочки необходимо

определить расположение верхних и нижних век, которые могут закрывать радужную оболочку. Для выполнения этой операции используется краевой детектор на основе градиента, который настроен на использование горизонтальных кромок.

Представленные методы определения местоположения зрачка и лимба предполагают, что границы являются идеальными кругами. Хотя подходы различны, все эти методы рассматривают зрачок и лимб как круговые кривые, что может привести к неправильному обнаружению границ. На рисунке 2.2. некоторые изображения радужной оболочки глаза из базы данных CASIA показывают, что границы зрачка не являются идеальными кругами.

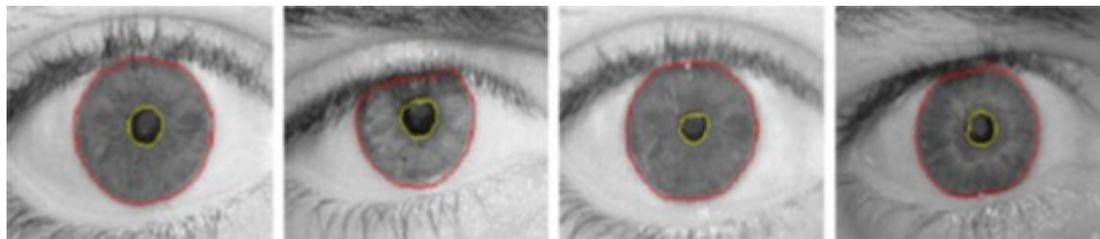


Рис. 2.2. Границы зрачка не являются идеальными кругами

2.4 Преобразование Даугмана из декартовых в полярные координаты

Метод нормализации Даугмана преобразует локализованную текстуру радужной оболочки из декартовой в полярную систему координат. Предложенный способ способен компенсировать нежелательные отклонения, обусловленные расстоянием глаза от камеры и её положением относительно камеры. Декартово-полярное преобразование определяется как:

$$\begin{aligned}
 x(\rho, \theta) &= (1 - \rho) \times x_p(\theta) + \rho \times x_i(\theta) \\
 y(\rho, \theta) &= (1 - \rho) \times y_p(\theta) + \rho \times y_i(\theta)
 \end{aligned}
 \tag{2.8}$$

$$\begin{aligned}
 x_p(\theta) &= x_{p0}(\theta) + r_p \times \cos(\theta) \\
 y_p(\theta) &= y_{p0}(\theta) + r_p \times \sin(\theta) \\
 x_i(\theta) &= x_{i0}(\theta) + r_i \times \cos(\theta) \\
 y_i(\theta) &= y_{i0}(\theta) + r_i \times \sin(\theta)
 \end{aligned}
 \tag{2.9}$$

Процесс по своей сути безразмерен в угловом направлении. Предполагается, что в радиальном направлении текстура изменяется линейно, что называется моделью «резинового листа». Модель резинового листа линейно отображает текстуру радужки в радиальном направлении от границы зрачка к границе лимба в интервал $[0,1]$ и создаёт безразмерное преобразование и в радиальном направлении.

Хотя метод нормализации компенсирует изменения, вызванные масштабом, смещением и расширением зрачка, он по своей сути не инвариантен к вращению радужки. Вращение радужки в декартовых координатах эквивалентно сдвигу в полярных координатах. В этом методе шаблоны радужки смещаются и сравниваются в n разных направлениях, чтобы компенсировать эффекты вращения.

Большинство методов нормализации, используемых в предложенных алгоритмах, преобразуют изображения радужной оболочки из декартовых в полярные координаты. Хотя процесс нормализации, предложенный Даугманом, показал, что он вполне эффективен, существуют некоторые аспекты, которые требуются для подробного изучения и обсуждения. Известно, что предлагаемое псевдополярное преобразование отображает текстуру радужной оболочки в интервал $[0,1]$ в радиальном направлении. Кроме того, в исходной формулировке преобразования каждая точка

текстуры радужной оболочки представлена полярной координатой (r, θ) . Однако вопрос возникает из-за того, что зрачок и лимб не всегда концентричны, а определение полярных координат требует центральной точки как начала полярных координат.

Некоторые алгоритмы, представленные Болесом или Лимом и др. разворачивают текстуру, основываясь на центре зрачка. В более поздних исследованиях, представленных Юнгом и др., используется метод развёртки, который улучшает выравнивание радужки [31]. В этом методе предлагается использовать центр лимба для определения полярных координат точек над границей лимба и центр зрачка для определения полярных координат границы зрачка. Координаты других точек между двумя границами затем линейно рассчитываются в радиальном направлении.

2.5 Расстояние Хэмминга

Для сопоставления в качестве метрики для распознавания используется расстояние Хэмминга, поскольку необходимы битовые сравнения. Используемый алгоритм расстояния Хэмминга также включает маскирование шума, так что при вычислении расстояния Хэмминга между двумя шаблонами радужки используются только значительные биты. При использовании расстояния Хэмминга, в расчёте используются только те биты, которые соответствуют 0 битам в шумовых масках обоих рисунков радужки [32]. Расстояние Хэмминга будет рассчитываться с использованием только битов, генерируемых из истинной области радужной оболочки, и эта формула расстояния Хэмминга задаётся как:

$$HD = \frac{1}{N - \sum_{k=1}^N X_{n_k}(\text{OR})Y_{n_k}} \sum_{j=1}^N X_j(\text{XOR})Y_j(\text{AND})X_{n_j}(\text{AND})Y_{n_j} \quad (2.10)$$

Где X_j и Y_j — два битовых шаблона для сравнения, X_{n_j} и Y_{n_j} — соответствующие маски шума для X_j и Y_j , а N — количество битов, представленных каждым шаблоном.

Хотя, в теории, два шаблона, созданных из одной радужки будут иметь расстояние Хэмминга 0.0, на практике этого не происходит. Нормализация не идеальна, а также будет некоторый шум, который остаётся незамеченным, поэтому некоторые изменения будут присутствовать при сравнении двух внутриклассовых шаблонов радужки.

Поскольку отдельная область радужки содержит признаки с высокой степенью свободы, каждая область радужной оболочки будет создавать битовую диаграмму, которая не зависит от другой, создаваемой другой радужкой, с другой стороны, два шаблона, созданных из одной и той же радужки, будут скоррелированы.

Если два битовых шаблона полностью независимы, например, радужные шаблоны, созданные из разных радужек, расстояние Хэмминга между двумя шаблонами должно быть равно 0,5. Это происходит потому, что независимость подразумевает, что два битовых шаблона будут полностью случайными, поэтому существует 0.5 шанса установить любой бит в 1, и наоборот. Если два шаблона получены из одного и того же радужной оболочки, расстояние Хэмминга между ними будет близко к 0.0, так как они сильно коррелированы, и биты должны согласиться между двумя кодами радужной оболочки

3 РАЗРАБОТКА АЛГОРИТМА

3.1 Предварительная обработка изображения

Выравнивание гистограммы — это метод, который улучшает контраст изображений, и он был признан одним из самых мощных методов улучшения изображения. Шаги для выравнивания гистограммы:

1. Вычислить кумулятивную гистограмму изображения для выравнивания;
2. Нормализовать кумулятивную гистограмму до 255 (это максимальное значение интенсивности в диапазоне (от 0 до 255) изображения в градациях серого);
3. Нормализованная кумулятивная гистограмма используется как функция отображения исходного изображения

Выравнивание гистограммы изображения глаза вызвало изменение значений интенсивности всех пикселей, а также центрального пикселя изображения. Выравнивание гистограммы нормализует значения интенсивности всех пикселей изображения, позволяя областям с более низкой контрастностью иметь более высокую контрастность [33].

Методы повышения контрастности используются для улучшения качества изображения, делая видимые пиксельные детали невидимыми на изображениях с низкой контрастностью. Распределение значений интенсивности пикселей в изображении с низкой контрастностью

распределяется в узком диапазоне по сравнению с распределением значений интенсивности пикселей в изображении с высокой контрастностью. Один из методов повышения контрастности даётся уравнением ниже:

$$T_k(x, y) = a[f_k(x, y) - 127] + 127 \quad (3.1)$$

Где k обозначает количество пикселей, (x, y) — координаты пикселя, $f_k(x, y)$ — исходное изображение. Согласно уравнению, контраст входного изображения уменьшается или увеличивается путём умножения значения каждого пикселя входного изображения на значение a .

3.2 Околокруговой активный контур зрачка

Для того, чтобы эффективно извлечь границу зрачка, необходимо определить характеристики контура, которые система захватывает. В общем случае граница зрачка представляет собой замкнутую, непрерывную и гладкую кривую, которая является почти круговой [34]. Чтобы достичь более высокой производительности на следующих этапах системы распознавания радужки, необходимо зафиксировать этот контур относительно правильной центральной точки и правильного угла [35]. Центральная точка контура определяется как среднее из вершин, которые могут быть записаны как:

$$C = (x_c, y_c) = \frac{1}{N} \sum_{i=1}^N V_i \quad (3.2)$$

где N — общее число вершин, а V_i — i -я вершина.

Угловое разрешение контура является ещё одним важным аспектом,

который следует учитывать. Критерий непрерывности определяется на основе углового разрешения, а не расстояния между вершинами, которые являются общими в моделях активного контура [36]. Это разрешение выбирается исходя из среднего радиуса зрачков в базе данных изображений глаз. Учитывая средний радиус 45 пикселей, периметр круга имеет около 285 пикселей. В этом случае выбирается разрешение 400 углов, чтобы получить непрерывные контуры. Число вершин является постоянным, и каждая вершина представляет определённый угол на протяжении всего процесса. Это условие можно рассматривать как угловые силы, которые приводят вершины в правильное угловое положение относительно обновлённого центра контура.

Каждая вершина контура представлена вектором, который имеет определённый радиус и направление относительно центральной точки. Чтобы получить гладкую кривую с почти круговой формой, внутренние силы прикладываются в радиальном направлении таким образом, чтобы соседние вершины имели одинаковое значение радиуса в собственном угловом диапазоне, $\Delta\theta$. Величина внутренней силы, приложенной к одной вершине, определяется как:

$$|F_{\text{int},i}| = \frac{1}{N_{\Delta\theta}} \times \left(\sum_{\theta_k = \theta_i - \frac{\Delta\theta}{2}}^{\theta_k = \theta_i + \frac{\Delta\theta}{2}} R_{\theta_k} \right) - R_i \quad (3.3)$$

где $\Delta\theta$ представляет угловой диапазон, $N_{\Delta\theta}$ — количество вершин в заданном угловом диапазоне, а R_i — радиус вершины i . Рисунок 3.1 иллюстрирует определённую внутреннюю силу над вершиной. Внутренние силы предназначены для выталкивания вершин, чтобы иметь тот же радиус, что и средний радиус вершин в определённом угловом диапазоне.

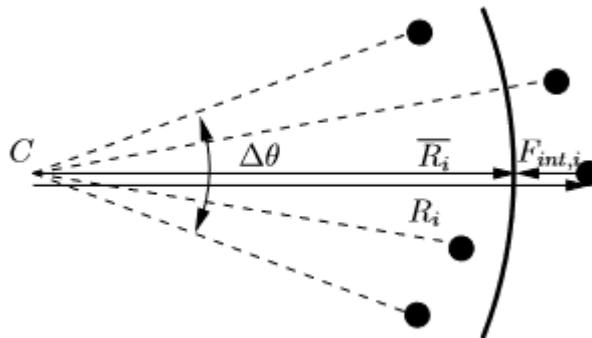


Рисунок 3.1. — Внутренняя сила над вершиной

Внешние силы предназначены для вытягивания кривой в направлении максимального измеренного кругового градиента, и применяются также в радиальном направлении. Круговой градиент изображения вокруг центральной точки получается путём применения интегро-дифференциального оператора к различным углам с соответствующим угловым диапазоном, $\Delta\theta$. Интегро-дифференциальный оператор Даугмана является детектором круговых контуров, определяется как:

$$\max(r, x_0, y_0) \left| G_\sigma(r) * \frac{\partial}{\partial r} \int_{(r, x_0, y_0)} \frac{I(x, y)}{2\pi r} ds \right| \quad (3.4)$$

Исходная формула интегро-дифференциального оператора интегрируется по интервалу $[0, 2\pi]$, который является полным кругом. Однако определение внешних сил основано на получении информации о круговых контурах в меньшем угловом диапазоне. Угловым диапазоном $\Delta\theta = \frac{\pi}{10}$ может быть правильным значением для получения как внутренних, так и внешних сил для приближения к почти круговым границам зрачка. Внешние силы затем определяются радиальным расстоянием каждой вершины от максимального кругового градиента, и могут быть записаны как:

$$|F_{\text{ext},i}| = \operatorname{argmax}_{R_c} \left| \frac{\partial}{\partial r} \int_{\theta_i - \frac{\Delta\theta}{2}}^{\theta_i + \frac{\Delta\theta}{2}} \frac{I(x,y)}{2\pi r} \right| - R_i \quad (3.5)$$

Предложенные внешние силы, основанные на сборе информации о градиенте в радиальных направлениях, имеют преимущество в том, что контур не останавливается в локальном контуре, и возможность пропуска зрачка становится очень низкой. Поиск границы зрачка для получения внутренних и внешних сил осуществляется с учётом поиска интервала вокруг медианного значения полученной радиальной информации зрачка на предыдущей итерации. Этот метод увеличивает скорость алгоритма, ограничивая пространство поиска и отфильтровывая внешние силы, указывающие на сильные контуры, такие как веки и ресницы, вместо границы зрачка. Кроме того, для ускорения алгоритма внешние силы вычисляются только в 128 угловых направлениях, а остальные значения интерполируются на основе предположения о почти круговой границе зрачка в угловом диапазоне $\Delta\theta = \frac{\pi}{10}$. Радиальное перемещение контура основано на композиции внутренних и внешних сил над вершинами контура, и замена каждой вершины в радиальном направлении получается итеративно:

$$R_{V_i}(n+1) = R_{V_i}(n) + F_{\text{int},i} + \alpha \times F_{\text{ext},i} \quad (3.6)$$

где R_{V_i} — радиус вершины i , а $\alpha \in (0,1)$ — вес, контролирующий темп движения контура и задающий состояние равновесия внутренних и внешних сил. Меньшие значения α приводят к сглаживанию контуров, однако контур движется к максимальной области градиента в более медленном темпе. Увеличение α увеличивает эффект внешних сил, и вершины контура быстрее движутся к границе зрачка. Значение α должно быть меньше 1,

чтобы сходиться к границе зрачка. Если α больше 1, контур будет колебаться вблизи границы зрачка из-за определения внешних сил. Окончательное равновесие достигается, когда средний радиус и центр контура становятся такими же, как и в предыдущей итерации.

В этом процессе имеется фиксированное число вершин. Каждая вершина представляет собой определённый угол во всех итерациях относительно обновлённого центра контура. Это условие можно рассматривать как угловые силы, которые отвечают за корректировку углового положения вершин относительно обновлённой центральной точки.

3.3 Эллиптический контур век

Модель контура век основана на предположении, что глазное яблоко имеет сферическую форму. Это предположение приводит к модели века, основанной на открытости глаза. Открытость определяется как угловое положение века относительно центра сферы. Предполагая, что глазное яблоко может быть представлено как сфера:

$$x^2 + y^2 + z^2 = R^2 \quad (3.7)$$

кривая века может быть получена путём пересечения этой сферы с плоскостью, проходящей через ось x с углом ϕ относительно оси Z . Рисунок 3.2. иллюстрирует вид сбоку и вид спереди контура век, учитывая сферическую форму глазного яблока и кривую век в определённой степени открытости глаза. Эта плоскость может быть записана как:

$$\tan(\phi) = \frac{y}{z} \quad (3.8)$$

Кривая пересечения просто становится эллиптической кривой:

$$x^2 = \frac{y^2}{\sin(\phi)^2} = R^2 \quad (3.9)$$

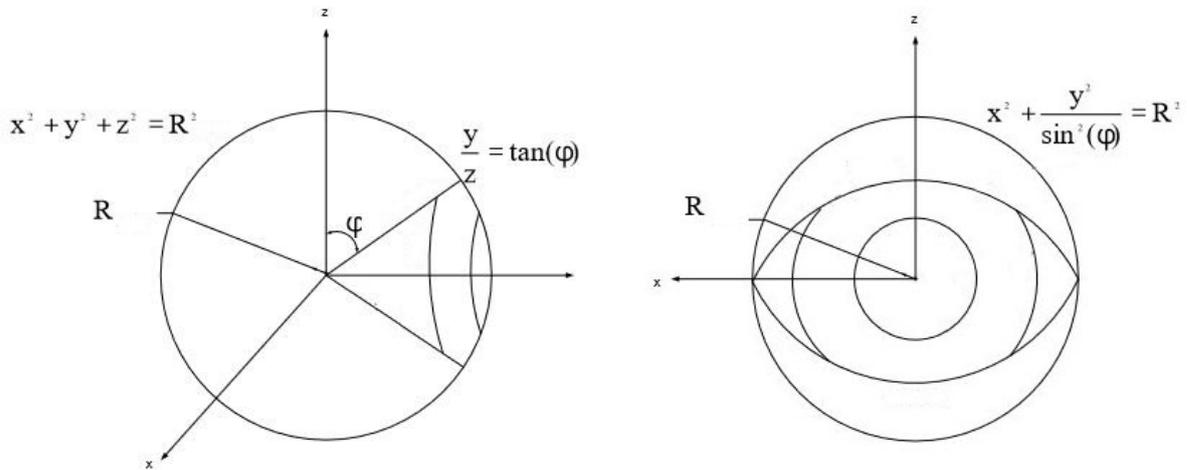


Рис. 3.2. Вид сбоку и вид спереди контура век

Затем эта кривая преобразуется в полярные координаты, чтобы быть более подходящей для нашей системы

$$x = r \sin(\theta), y = r \cos(\theta) \quad (3.10)$$

$$r = \frac{R}{\sqrt{\sin(\theta)^2 + \frac{\cos(\theta)^2}{\sin(\theta)^2}}} \quad (3.11)$$

3.4 Итеративный алгоритм определение лимба и век

Итеративный алгоритм разработан для точного определения границ лимба и век. В методе извлечения используется интегро-дифференциальный

оператор для определения границ. Обнаружение век основано на эллиптических контурах, которые моделируются кривой век в разных степенях открытости глаза.

В изображениях ближнего инфракрасного диапазона, предоставленных Институтом автоматизации китайской Академии наук (CASIA), границы лимба имеют недостаточный контраст, поэтому глобальные методы поиска, такие как интегро-дифференциальный оператор, более подходят для процесса извлечения. Однако даже глобальные методы могут привести к ложному обнаружению из-за таких шумов, таких как сильные границы верхнего и нижнего века.

Алгоритм итеративно ищет границы радужной оболочки и век и исключает обнаруженные области век для следующей итерации. Процесс происходит относительно центра зрачка, как контрольной точки [39], и выполняется путём исключения значений пикселей, где радиус лимба больше радиуса верхнего или нижнего века. Это условие маскирует области, где радужная оболочка покрыта двумя веками, и процесс повторяется до тех пор, пока результат поиска не сойдётся к фиксированному центру и радиусу лимба. На рисунке 3.3. показан пример итеративного алгоритма.

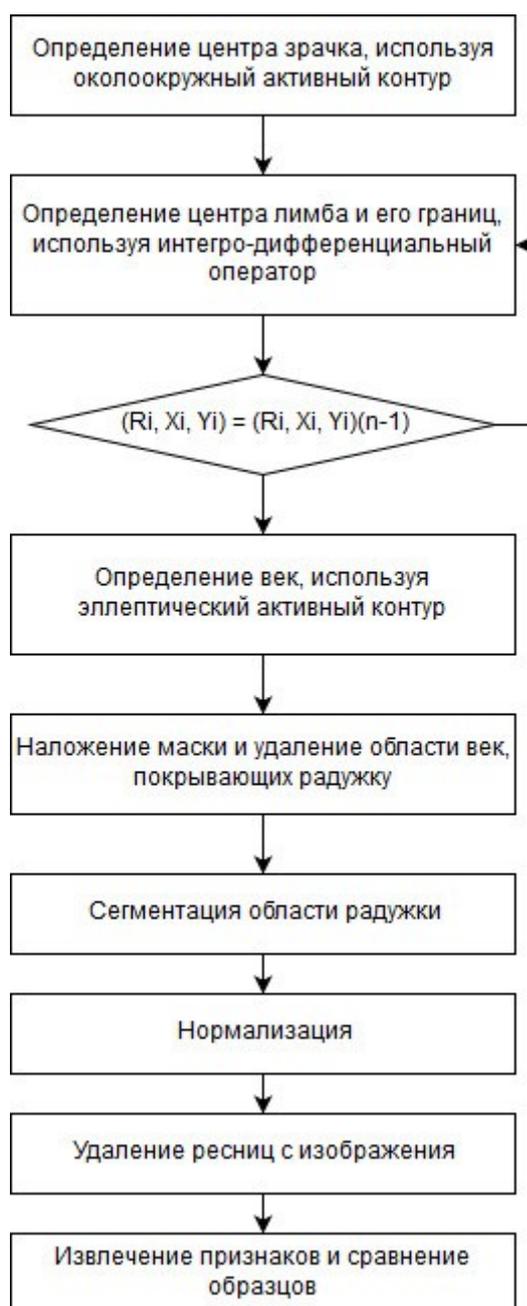


Рис. 3.3. Алгоритм нахождения границы лимба и век

3.5 Контурный метод развёртки радужки

Преобразование текстуры радужки из декартовых в полярные координаты, известное как процесс развёртки, является одним из этапов

алгоритма распознавания радужки и оказывает значительное влияние на общую производительность системы.

Основная цель текстуры радужки заключается в том, чтобы контролировать количество света, попадающего в глаз при различных условиях освещения. Текстура радужной оболочки состоит из двух основных мышц, распределенных в угловом и радиальном направлениях. При сильном освещении для уменьшения диаметра зрачка, мышца сокращается, чтобы отрегулировать свет, который попадает в глаз. Однако в условиях, когда свет слабый, мышца расслабляется для увеличения диаметра зрачка, чтобы больше света попадало в глаз.

Два различных условия освещения, сильный свет и слабый свет, приводят к нелинейному поведению текстуры радужки. В области распознавания радужной оболочки глаза изучение поведения радужки является важным, чтобы иметь возможность предложить надлежащую методику нормализации. Чтобы более подробно описать важность метода нормализации, было бы полезно подчеркнуть уникальность радужной оболочки глаза каждого человека и её стабильность с течением времени. Предполагая, что каждая радужная оболочка имеет уникальные узоры, которые не меняются со временем, желательно фиксировать эти узоры каждый раз, когда берётся изображение глаза. Однако, текстура радужки сильно чувствительна к освещению, и текстура деформирует для изменения диаметра зрачка и для того чтобы контролировать количество света попадающего в глаз. Чтобы получить уникальные образцы радужной оболочки, желательно отслеживать изменения радужки для восстановления уникальных узоров.

Следует отметить, что реакция радужки на освещение также варьируется от глаза к глазу из-за различных распределений мышц,

контролирующих зрачок. Поэтому представляется более практичным изучить поведение большого числа радужек, чтобы получить модель нормализации, оптимальную для захвата и отслеживания узоров.

Центр лимба

Центр лимба радужной оболочки можно использовать в качестве опорной точки, чтобы сделать развёртку текстуру радужной оболочки глаза. Возможное преимущество этой точки заключается в том, что контур лимба фиксирован и не изменяется со временем. На рисунке 3.4. показан способ развёртки, использующий центр лимба в качестве контрольной точки.

Центр зрачка

Центр зрачка также может считаться правильным выбором для контрольной точки. Алгоритмы Лима и др., Болеса и др. основаны на центре зрачка в качестве опорной точки. Развёртка радужной оболочки по отношению к этой точке представляется выгодной при извлечении признаков из областей вблизи зрачка, где текстура радужной оболочки имеет более чёткие узоры, и это может привести к приемлемым результатам распознавания. На рисунке 3.4. показан способ развёртки с использованием центра зрачка в качестве контрольной точки.

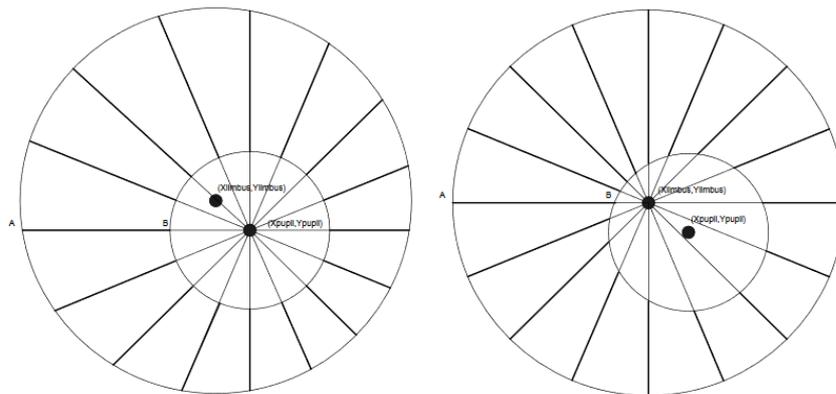


Рис. 3.4. Способ развёртки с использованием центра в качестве контрольной

ТОЧКИ

Минимальное расстояние

Рассмотрим метод развёртки на основе минимального расстояния точек от границы зрачка до границы лимба. В этом методе N точек с одинаковым расстоянием выбираются на границе лимба, а соответствующие точки на границе зрачка выбираются на основе критерия минимального расстояния. Метод приводит к глобальному и локальному минимальному расстоянию между точками над двумя границами. Следует отметить, что метод минимального расстояния может вызвать неравномерное разрешение по границе зрачка из-за различного расположения центра зрачка относительно центра лимба. Кроме того, эксперименты были повторены для условия, при котором N равномерно распределённых точек выбираются по границе зрачка вместо границы лимба, чтобы рассмотреть их потенциальные различия.

Глобальный минимум

Как и метод минимального расстояния, метод развёртки изучается на основе глобального минимума расстояний между границами лимба и зрачка. В этом методе, несмотря на метод минимального расстояния, точки, выбранные над обеими границами, равномерно распределены, а позиционирование двух границ основано на их глобальном расстоянии. Глобальное минимальное расстояние модели, основанной на контуре, определяется как:

$$\operatorname{argmin} \left(\int_0^{2\pi} \sqrt{(x_i - x_p(\theta))^2 + (y_i - y_p(\theta))^2} d(\theta) \right) \quad (3.12)$$

где (x_i, y_i) и (x_p, y_p) являются границами лимба и зрачка соответственно.

θ_o определяется как угловое смещение для границы зрачка, чтобы минимизировать глобальную минимальную формулировку.

Первоначальная идея как минимального расстояния, так и методов на основе контуров возникла из предположения, что текстура радужной оболочки имеет тенденцию находиться в самом расслабленном состоянии в каждом диаметре зрачка. Поэтому было бы целесообразно скорректировать точки над двумя границами таким образом, чтобы текстура не подвергалась чрезмерному радиальному натяжению. Более того, как минимальное расстояние, так и модели на основе контуров имеют преимущество над методами, основанными на центральной точке, в тех случаях, когда зрачки не являются совершенно круговыми. В тех случаях, когда границы зрачка не являются идеальными кругами и были обнаружены активной моделью контура, модели на основе центра могут быть не такими надёжными, как модели на основе контура, из-за чувствительности нормализации к правильному выбору центральной точки.

3.6 Определение и маскирование ресниц на изображении

Один из многих факторов, которые влияют на производительность распознавания радужной оболочки глаза является окклюзия, вызванная ресницами. В то время как некоторые подходы игнорируют области радужки, закрываемые ресницами [40], другие обнаруживают ресницы и маскируют их при кодировании [40]. Напротив, техника пытается восстановить области радужки, которые закрываются ресницами.

Обнаружение ресниц требует правильного выбора функций и

процедуры классификации из-за сложности и случайности шаблонов.

Для выполнения восстановления используется нелинейный метод условной направленной фильтрации. Предложенная методика представляет собой итерационный подход, включающий следующие этапы: обнаружение пикселей, соответствующих окклюзии ресниц, определение направления ресницы, которое вызывает окклюзию, локальная фильтрация области окклюзии в направлении, перпендикулярном реснице, восстановление интенсивности пикселей с помощью фильтра.

Ожидается, что ресница вызовет разрыв по краям. Поэтому, учитывая развёрнутое изображение радужки I , ресница обнаруживается, а её направление оценивается с помощью краевого фильтра. Для этого на изображение наносится фильтр Собеля 3×3 . Фильтр Собеля показан на рисунке 3.5.

-1	-2	-1	z_1	z_2	z_3	-1	0	1
0	0	0	z_4	z_5	z_6	-2	0	2
1	2	1	z_7	z_8	z_9	-1	0	1

Рис 3.5. Фильтр Собеля

Для каждого пикселя градиенты в направлениях x и y $[G_x, G_y]$ могут быть определены следующими уравнениями:

$$\begin{aligned} G_x &= (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \\ G_y &= (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \end{aligned} \quad (3.13)$$

Аналогично, величина градиента в центре маски, $Grad$, вычисляется как:

$$Grad = \sqrt{(G_x^2 + G_y^2)} \quad (3.14)$$

Затем можно определить направление локального градиента, перпендикулярное краю:

$$\theta = \arctan(G_y/G_x) \quad (3.15)$$

Чтобы определить, является ли пиксел закрытым или нет, окно размером $m \times n$ центрируется на пикселе, и отклонение направления градиента для R пикселей, которые лежат в окне и имеют значение градуса выше определённого порога, вычисляется следующим образом:

$$\text{Var}_{Grad} = \frac{1}{r-1} \sum_{i=1}^r (\theta_i - \bar{\theta})^2 \quad (3.16)$$

Сильный контур отмечается, если направление градиента имеет небольшое отклонение, а пиксель классифицируется как затронутый ресницей. Для восстановления такого пикселя вдоль направления применяется 1D медианный фильтр длины L . Этот процесс выдаёт оценку значения изображения с удалённой ресницей.

Чтобы избежать неправильной фильтрации пикселей без ресниц, изменение пикселя выполняется только в том случае, если изменение пикселя превышает определённый порог, связанный с общей дисперсией изображения. Для этой цели параметр $Recover$ вычисляется следующим образом:

$$Recover = Diff - k * Var(Image) \quad (3.17)$$

где $Diff$ представляет разницу в интенсивности между фильтрованным и

нефильтрованным пикселем, $Var(Image)$ представляет дисперсию интенсивности всего нефильтрованного изображения, K обозначает параметр, используемый для настройки порога. Пиксел заменяется фильтруемым значением, только если значение $Recover$ положительное. При таком подходе внешний вид существенно не изменяется, но улучшается производительность распознавания. На рисунке 3.6. показано развёрнутое изображение радужной оболочки глаза до и после применения подхода .

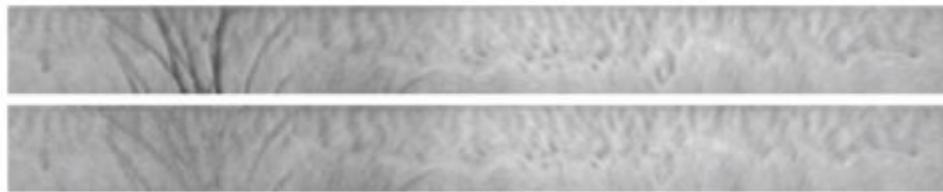


Рис. 3.6. Результат маскирования ресниц

3.7 Параллельный подход

Параллельные вычисления были областью активных исследований в течение последних нескольких лет, и многие технические энтузиасты пытались повысить эффективность приложений посредством параллельных вычислительных технологий и алгоритмов. Он применяется к ряду процессов, начиная от прогнозирования погоды и заканчивая поиском полезных ископаемых, и обработка изображений не является исключением. Все эти приложения требуют большого количества вычислений. С учётом недавнего роста вычислительной мощности машин спрос на более быструю обработку увеличился, как никогда ранее.

Параллельные вычисления направлены на решение сложной задачи

путём разделения задачи на подмножества и решения каждой задачи в виде отдельного потока, который может выполняться параллельно. Некоторые процессы алгоритма могут выполняться параллельно, а некоторые должны выполняться последовательно один за другим. Алгоритм, в котором все задачи могут выполняться параллельно одновременно из-за их независимости от данных, называется параллельным алгоритмом.

Были предприняты значительные усилия для разработки параллельных алгоритмов с целью распознавания радужной оболочки глаза, и это все еще область активных исследований, и это было источником мотивации для проекта. Другой источник мотивации — последние достижения в развитии графических процессоров, которые широко используются в исследовательских проектах для ускорения приложений, требующих значительного количества времени на обычных системах. Поскольку распознавание радужной оболочки глаза является вычислительным приложением для интенсивной обработки изображений, оно служит хорошим кандидатом для применения параллельной обработки. Локализация является важной частью процесса распознавания радужки, поэтому применение параллельной обработки для ускорения процесса улучшает общий процесс распознавания радужки.

Первоначальное наблюдение за процессом распознавания радужной оболочки глаза заключалось в том, что каждый этап процесса связан с большим объёмом обработки и, таким образом, занимает значительное количество времени. До сих пор большинство методов, разработанных для распознавания радужной оболочки глаза, носили последовательный характер. Этот фактор в сочетании с ростом параллельных инфраструктур, таких как графические процессоры, был заманчивым фактором. Локализация радужки является важным этапом в процессе обнаружения радужки и занимает

значительное количество времени при её последовательной реализации.

Классификация параллельных систем

SISD (Single Instruction Single Data) — Один последовательный процессор выполняет набор инструкций для работы с данными, хранящимися в единой памяти. Как правило, инструкции выполняются последовательно, но они могут перекрываться на этапах выполнения, что означает, что мы используем конвейеризацию в такой системе для ускорения. Эти типы системы показали очень низкий параллелизм или его отсутствие.

SIMD (Single Instruction Multiple Data) — Несколько процессоров одновременно выполняют командную инструкцию, и каждый процессор имеет собственную память и собственный поток данных. Каждый процессор имеет связанную с ним память данных, так что каждая команда выполняется на различных наборах данных различными процессорами.

MISD (Multiple Data Single Instruction) — Последовательность потока данных передается в набор процессоров один за другим, каждый из которых выполняет другой поток команд. Выход одного процессора работает как вход для следующего процессора. Такого рода структуры не реализуются коммерчески и не существуют в физическом мире.

MIMD (Multiple Instruction Multiple Data) — Машины, использующие архитектуру MIMD, имеют множество процессоров, работающих асинхронно и независимо. Каждый процессор имеет свои локальные данные и выполняет на нем свои собственные инструкции. Многоядерные процессоры и многопоточные процессоры основаны на этой модели.

Модель коммуникации

Передача данных между параллельными ядрами является основой параллельной обработки, поскольку в большинстве случаев требуется передача данных из одного ядра в другое, и это происходит ещё чаще, если элементы данных взаимозависимы. Эффективность параллельного алгоритма зависит от времени, затрачиваемого на передачу различных фрагментов данных между элементами обработки, и фактически ограничена. Иногда также случается, что время, затрачиваемое на передачу данных между элементами обработки, становится больше времени, необходимого для параллельного расчёта результатов. В этом случае преимущества параллельного алгоритма будут полностью потеряны. Таким образом, возникает необходимость в разработке подходящих методов для передачи данных и оптимизации количества потоков данных между процессорными ядрами. Ускорение параллельной программы сильно зависит от взаимодействия между процессорами. В некоторых случаях, когда связь между процессорами больше, чем вычисление, тогда производительность системы ухудшается даже после распараллеливания. Существует два основных вида обмена данными между параллельными задачами, которые обращаются к пространству разделяемой памяти и обмена сообщениями между системами:

MessagePassing — В этом механизме каждый процессор имеет свою память и адресное пространство. Взаимодействие между ядрами обработки достигается за счёт передачи сообщений между ядрами обработки. Интерфейсы передачи сообщений используются для поддержки таких взаимодействий. Обмен сообщениями используется для передачи данных, работы, и для синхронизации между процессами. В самом общем виде

парадигмы передачи сообщений поддерживают выполнение различных программ на каждом из p узлов. Основными операциями в этой парадигме программирования являются отправка и получение сообщений. Также должен существовать метод присвоения уникальной идентификации каждому из процессов, выполняющих параллельное приложение.

Shared Address Space — В этом методе общая память доступна для всех процессорных ядер. Адресное пространство общей памяти может быть доступно равномерно или неравномерно. В Uniform Memory Access (UMA) время, затраченное процессором на доступ к любому местоположению памяти, является тем же, тогда как в Non Uniform Memory Access (NUMA) время, затраченное процессором на доступ к различным местоположениям памяти, может отличаться. Таким образом, каждая память имеет равные права доступа к любой памяти.

На стадиях оценки по топологии и геометрии выполняется обработка наборов данных, содержащих частные признаки изображения и связи между ними, причем каждый шаг вычислений выполняется независимо от других. Организуем параллельную обработку данных на этих этапах с помощью технологии OpenMP. При оценке степени подобия по геометрии параллельные вычисления можно использовать при подсчете количества связей, совпадающих с базовыми точками. Для этого при помощи директивы `omp for` выполняется обработка набора связей, каждый из параллельных процессов работает с собственным набором данных.

Технология OpenMP в данной работе является базисом для дальнейшего распараллеливания алгоритма. На ее основе выполнялись реализации сегментации РОГ, а также производился сравнительный анализ быстродействия.

4 ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ

4.1 Используемые аппаратно-программные средства

База данных изображений CASIA

Название: Chinese Academy of Science Institute of Automation Iris Image Database. Это несколько разнородных баз данных. Были использованы: CASIA-Iris-Interval. Это набор изображений радужки, снятых собственным устройством Института Автоматизации. Разрешение 320×280 пикселей, число персон — 249, число зарегистрированных глаз — 395, общее количество изображений — 2639. Число изображений на один глаз от 1 до 26. Все изображения высокого качества.

Язык программирования C++

C ++, произносится как «Си-плюс-плюс», это язык программирования, который был построен на языке C. Синтаксис C ++ почти идентичен C, но он имеет объектно-ориентированные функции, которые позволяют программисту создавать объекты внутри кода. Это упрощает программирование, повышает эффективность, а некоторые даже говорят, что это больше удовольствия. Из-за гибкости языка большинство программ сегодня написаны на C ++. Он компилируется непосредственно на собственный код машины, позволяя ему быть одним из самых быстрых языков в мире. Язык предлагает много вариантов парадигм. C++ предлагает замечательную поддержку процедурных, универсальных и объектно-

ориентированных парадигм программирования, а также многих других парадигм.

IDE Qt Creator

Qt Creator — это кросс-платформенная интегрированная среда разработки C++, JavaScript и QML, которая является частью SDK для платформы разработки приложений QT GUI, включает в себя визуальный отладчик и встроенный графический интерфейс и конструктор форм. Функции редактора включают подсветку синтаксиса и автозаполнение. Qt Creator использует компилятор C++ из коллекции компиляторов GNU в Linux и FreeBSD. В Windows он может использовать MinGW или MSVC с установкой по умолчанию, а также может использовать отладчик консоли Microsoft при компиляции из исходного кода.

Фреймворк Qt

Qt — это кросс-платформенный фреймворк приложений и инструментарий виджетов для создания классических и встроенных графических пользовательских интерфейсов, а также приложений, которые работают на различных программных и аппаратных платформах с небольшим или нулевым изменением кодовой базы, при этом все еще являясь приложением с большими возможностями и скоростью. Qt в настоящее время разрабатывается как компанией Qt, публично зарегистрированной компанией, так и проектом Qt с участием отдельных разработчиков и фирм, работающих над продвижением Qt. Qt используется для разработки графических пользовательских интерфейсов (GUI) и мультиплатформенных приложений, которые работают на всех основных

настольных платформах и большинстве мобильных или встроенных платформ. Большинство графических программ, созданных с помощью Qt, имеют собственный интерфейс, в этом случае Qt классифицируется как widget toolkit. Также могут быть разработаны программы без графического интерфейса, такие как средства командной строки и консоли для серверов. Qt поддерживает различные компиляторы, включая компилятор GCC C++ и Visual Studio.

Библиотека OpenCV

OpenCV (Open Source Computer Vision Library) представляет собой библиотеку программного обеспечения для компьютерного зрения с открытым исходным кодом и компьютерного обучения. OpenCV была создана для обеспечения общей инфраструктуры приложений для компьютерного зрения и ускорения использования восприятия машины в коммерческих продуктах. В библиотеке более 2500 оптимизированных алгоритмов, которые включают в себя полный набор классических и современных алгоритмов компьютерного зрения и машинного обучения. Эти алгоритмы могут использоваться для обнаружения и распознавания лиц, идентификации объектов, классификации действий человека в видео, отслеживания движения камеры, отслеживания движущихся объектов, извлечения 3D-моделей объектов, создания 3D-облаков точек из стереокамер, сшивания изображений вместе для получения высокого разрешения изображение всей сцены, найти похожие изображения из базы данных изображений, удалять красные глаза с изображений, сделанных с помощью вспышки, следить за движениями глаз, распознавать декорации и устанавливать маркеры, чтобы накладывать их на дополненную реальность и т. д. OpenCV имеет более 47 тысяч пользователей сообщества и

предполагаемое количество загрузок, превышающих 14 миллионов. Библиотека широко используется в компаниях, исследовательских группах и правительственных органах. OpenCV работает с C ++, Python, Java и MATLAB и поддерживает Windows, Linux, Android и Mac OS. OpenCV ориентируется в основном на приложения в режиме реального времени и использует преимущества команд MMX и SSE, когда они доступны. В настоящее время активно развиваются полнофункциональные интерфейсы CUDA и OpenCL. OpenCV написана на языке C ++ и имеет шаблонный интерфейс, который работает без проблем с контейнерами STL.

Параметры тестовой системы

Количество процессов — 4, тип процессора — Intel Atom 1,6 ГГц, оперативная память 2 Гб.

4.2 Критерии точности идентификации

При проверке возможны два типа ошибок: первого и второго рода. Вероятности этих двух ошибок обратно связаны и определяются уровнем значимости и мощностью теста. Поэтому, прежде чем находить их вероятности, следует определить, какая ошибка имеет более серьёзные последствия для конкретной ситуации.

Ошибка первого рода (ложное срабатывание) — это неправильное отклонение истинной нулевой гипотезы. Обычно ошибка первого рода приводит к выводу, что предполагаемый эффект или связь существуют, когда на самом деле это не так. Вероятность совершения ошибки первого рода равна α , что является уровнем значимости, установленным для проверки гипотезы. Значение α равное 0.05 означает, что мы готовы принять 5%

вероятности того, что ошибаемся, когда отклоняем нулевую гипотезу. Для снижения этого риска необходимо использовать меньшее значение α . Тем не менее, использование меньшего значения для α означает, что мы с меньшей вероятностью обнаружим истинную разницу, если она действительно существует.

Ошибка второго рода (пропуск события) — это отказ отклонить ложную нулевую гипотезу. Когда нулевая гипотеза ложна и мы не отклоняем её, то совершаем ошибку второго рода. Вероятность совершения ошибки второго рода равна β , что зависит от мощности теста. Мы можем уменьшить риск совершения ошибки второго рода, если тест имеет достаточную мощность, то есть размер выборки достаточно велик, чтобы обнаружить практическую разницу, когда она действительно существует.

FAR (False Acceptance Rate, коэффициент ложного пропуска, вероятность ложной идентификации) — вероятность того, что система биоидентификации по ошибке признает подлинность пользователя, не зарегистрированного в системе.

FRR (False Rejection Rate, коэффициент ложного отказа доступа) — вероятность того, что система биоидентификации не признает подлинность шаблона зарегистрированного в ней пользователя.

4.3 Результаты тестирования

Для тестирования сегментации радужной оболочки глаза были использованы следующие алгоритмы:

1. Предложенный алгоритм на основе методов околоокружного и

эллиптического активных контуров.

2. IFPP, реализация USIT алгоритма сегментации радужной оболочки на основе преобразования Фурье.
3. САИТ, реализация USIT преобразования Хафа с поправкой на контраст.
4. OSIRIS, система с открытым исходным кодом для распознавания радужной оболочки глаза.

Для проверки эффективности предложенной реализации была проведена серия вычислительных экспериментов, направленных на исследование работы алгоритма. Для тестирования использовалась система, с конфигурацией: количество процессов — 4, тип процессора — Intel Atom 1,6 ГГц, оперативная память 2 Гб.

Во время сегментации радужки, желательно избегать ошибок сегментации зрачковых и лимбических границ. Желательна особая устойчивость к умеренным шумовым факторам, например, расфокусировка, размытость изображения, различное просветление, направление движения вне оси, высокая дисперсия размера ирисовой диафрагмы или нелинейные искажения при применении несовершенных моделей радужной оболочки. Точность оценивается путём оценки влияния сегментации на достоверность распознавания. Скорость, то есть соблюдение ограничений в реальном времени является фактором качества, который часто игнорируется в сравнительных исследованиях, но этот фактор является наиболее критичным в приложениях. Среднее время сегментации оценивается для каждого из алгоритмов. При проведении экспериментов время работы алгоритма замерялось без учёта времени, затрачиваемого на чтение данных с диска и размещение их в оперативной памяти. Результаты представлены в таблице 4.1 и на рисунке 4.1.

Таблица 4.1.

Процент ошибок идентификации и скорость работы алгоритмов

	Ошибки идентификации (%)	Время сегментации (с)
Предложенный алгоритм	0.74%	0.18 с
IFFP	7.05%	0.21 с
CAHT	1.20%	1.02 с
OSIRIS	2.33%	0.84 с

На рисунке 4.1. представлены результаты тестирования в виде графика сравнений:

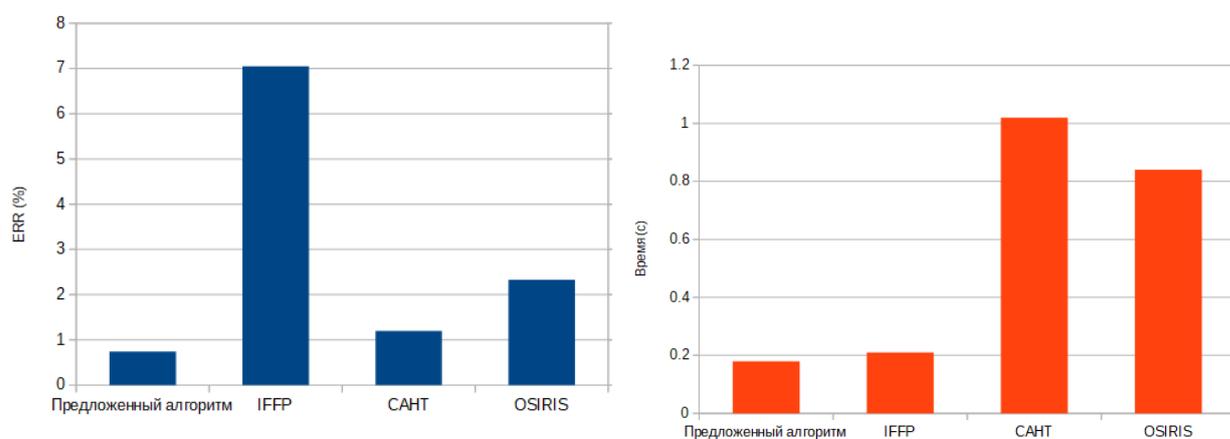


Рис. 4.1. Графики ошибок идентификации и скорости работы

ЗАКЛЮЧЕНИЕ

Разработан алгоритм, объединяющий методы в единый комплекс, осуществляющий сегментацию изображений РОГ. Отличительной особенностью работы системы является определение параметров радужки на нескольких шагах (начальное приближение, уточнение). Последовательность определения параметров радужки отличается от общепринятой и оптимизирована для достижения максимальной на каждом этапе доли изображений, на которых проведена сегментация. Представлены результаты работы этого комплекса для баз данных, находящихся в открытом доступе. В качестве направлений дальнейших исследований можно выбрать реализацию алгоритма сравнения РОГ на основе топологических векторов для графических процессоров, а также реализацию различных стратегий балансировки загрузки процессов в подходе параллелизма по данным и применение других технологий параллельного программирования в сочетании с OpenMP.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. R. Wildes, "Iris Recognition: an Emerging Biometric Technology." Proceeding of the IEEE, Vol. 85, No. 9, 1997.
2. Синицын И.Н., Новиков С.О., Урмаев О.С. Развитие технологий интеграции биометрической информации // Системы и средства информатики. 2004. N.14. P.5.
3. A. Safir, International Biometrics Group Report, 2010, 8-17.
4. Потебня Г.П., Лисовенко Г.С., Кривенко В.В. Клиническая и экспериментальная иридология. Киев: Наукова думка, 1995.
5. Соколов Е.Н., Измайлов Ч.А. Цветовое зрение. – М.: Издательство Московского университета, 1984. – 175 с.
6. Kansky J.J. Clinical Ophthalmology: a Systematic Approach, Elsevier. London, 2003.
7. Вельховер Е.С., Шульпина Н.Б., Алиева З.А., Ромашов Ф.Н. Иридодиагностика. – М.: Медицина, 1988. – 240 с.
8. Старовойтов В.В., Монич Ю.И. Распознавание человека по изображению радужной оболочки глаза: проблемы и достижения // Искусственный интеллект. 2011. N.3. P.278–284.
9. A. Bertillon, "a couleur de liris." Rev. Sci., vol. 36, no. 3, pp. 6573, 1885
10. J.G Daugman. "Iris recognition." American Scientist, July-Aug. 2001
11. L. Flom and A. Safir, "Iris recognition system." U.S. Patent 4 641 349, 1987.
12. J.G Daugman, "High Confidence Visual Recognition of Persons by a Test of Statistical Independence." IEEE Transactions On Pattern Analysis and Machine

Intelligence, Vol. 15, No.11, pp.1148-1161,Nov. 1993.

13. R. Wildes, J.C. Asmuth, G.L. Green, S.C. Hsu, R.J. Kolczynski, J.R. Matey and S.E. McBride, "A system for automated iris recognition." In Proceedings of the IEEE Workshop on Applications of Computer Vision, pp. 121-128, 1994

14. W.W. Boles, "A wavelet transform based technique for the recognition of the human iris." In Proceedings of the International Symposium on Signal Processing and its Application, ISSPA, Gold Coast, Australia, pp. 25-30, August, 1996.

15. S. Lobregt and M. A. Viergever, "A Discrete dynamic contour model." IEEE Transactions on Medival Imaging, 14(1), 1995, 12-24.

16. Y. Zhu, T. Tan and Y. Wang "Biometric personal identification based on iris pattern." Proceeding of. 15th International Conference on Pattern Recognition, vol. 2, pp. 801804, 2000.

17. S. Lim K. Lee O. Byeon and T. Kim, "Efficient Iris Recognition through Improvement of Feature Vector and Classifier." ETRI J., vol. 23, no. 2, pp. 61-70, 2001.

18. S. Noh, K. Pae, C. Lee, J. Kim. "Multiresolution independent component analysis for iris identification." The 2002 International Technical Conference on Circuits/Systems, Computers and Communications, Phuket, Thailand, 2002.

19. Форсайт Д.А., Понс Ж. Компьютерное зрение. Современный подход.: Пер. с англ. – М.: Издательский дом "Вильямс" 2004. – 928 с.

20. Синицын И.Н., Новиков С.О., Ушмаев О.С. Развитие технологий интеграции биометрической информации // Системы и средства информатики. 2004. N.14. P.5.

21. Методы компьютерной обработки изображений / Под. ред. В.А.Сойфера. – 2-е изд., испр. – М.: ФИЗМАТЛИТ, 2003. – 784 с.

22. Розенфельд А. Распознавание и обработка изображений с помощью вычислительных машин. М.:Мир, 1972. – 230 с.
23. J. Zuo, N. Kalka, N. Schmid, A robust IRIS segmentation procedure for unconstrained subject presentation, in Aug 2006, pp. 1–6
24. S. Lim K. Lee O. Byeon and T. Kim, “Efficient Iris Recognition through Improvement of Feature Vector and Classifier.” ETRI J., vol. 23, no. 2, pp. 61-70, 2001.
25. W.W. Boles and B. Boashash, “A human identification technique using images of the iris and wavelet transform.” IEEE Transactions on Signal Processing, vol. 46, no. 4, pp. 1185-1188, 1998.
26. B. J. Joung, J. O. Kim, C. H. Chung, Key Seo Lee, Wha Young Yim, Sang Hyo Lee, “On Improvement for Normalizing Iris Region for a Ubiquitous Computing.” In Proceedings of International Conference on Computational Science and Its Applications ICCSA 2005, Singapore, May 9-12, 2005.
27. Canny J. A computational approach to edge detection // IEEE TPAMI. 1986. V.8. N.6. P.679–698
28. J. Daugman, How iris recognition works, in vol. 1 (2002).
29. J. Illingworth, J. Kittler, A survey of the Hough transform. Comput. Vis. Graph. Image Process. 44(1), 87–116 (1988).
30. Фурсов В.А., Бибииков С.А. Локализация контуров объектов на изображениях при вариациях масштаба с использованием преобразования Хафа // Компьютерная оптика. 2013. N4. Т37. с.496-502.
31. B. J. Joung, J. O. Kim, C. H. Chung, Key Seo Lee, Wha Young Yim, Sang HyoLee, “On Improvement for Normalizing Iris Region for a Ubiquitous Computing.” In Proceedings of International Conference on Computational

Science and Its Applications ICCSA 2005, Singapore, May 9-12, 2005.

32. Павельева Е.А. Метод проекционной фазовой корреляции в ключевых точках радужной оболочки глаза // Proc. 22nd International Conference on Computer Graphics and Vision, GraphiCon-2012. P.128-132.

33. Яне Б. Цифровая обработка изображений. М.: Техносфера, 2007. – 584с.

34. Смирнов Д.А., Матвеев И.А. Определение границ век на изображении глаза методом активных контуров // Труды ИСА РАН, Динамика неоднородных систем. 2006. N. 25(2). с.200-207

35. Матвеев И.А. Оптимизация кругового пути как метод выделения и уточнения границ радужки на изображении глаза // Известия РАН. Теория и системы управления. 2011. N.5. с.78-84.

36. S. Lobregt and M. A. Viergever, “A Discrete dynamic contour model.” IEEE Transactions on Medical Imaging, 14(1), 1995, 12-24.

37. Матвеев И.А. Поиск центра глаза на изображении // Труды ИСА РАН. Динамика неоднородных систем. 2006. N.10(2). с.206-211.

38. J. Illingworth, J. Kittler, A survey of the Hough transform. Comput. Vis. Graph. Image Process. 44(1), 87–116 (1988).

39. W.W. Boles, “A wavelet transform based technique for the recognition of the human iris.” In Proceedings of the International Symposium on Signal Processing and its Application, ISSPA, Gold Coast, Australia, pp. 25-30, August, 1996.

40. Сеницын И.Н., Новиков С.О., Урмаев О.С. Развитие технологий интеграции биометрической информации // Системы и средства информатики. 2004. N.14. P.5.