

РЕФЕРАТ

Выпускная квалификационная работа бакалавра состоит из 59 страниц, 43 рисунков, 2 таблиц, 9 использованных источников, 1 приложения.

ОБНАРУЖЕНИЕ ОБЪЕКТОВ, ЦИФРОВАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ, КОНТУРНЫЙ АНАЛИЗ, ОБОБЩЕННОЕ ПРЕОБРАЗОВАНИЕ ХАФА

Целью работы является изучение, реализация и исследование алгоритма обнаружения объектов на изображении, инвариантных к аффинным преобразованиям. В качестве рассматриваемых методов были выбраны алгоритмы контурного анализа, благодаря своим преимуществам перед другими возможными методами.

В работе представлена программная реализация алгоритма обнаружения объектов на изображении, основанная на обобщенном преобразовании Хафа. Также был разработан метод обработки изображения, позволяющий выделять замкнутые контуры сложных форм. Проведен анализ работы этих программ в условиях зашумления изображений.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 ПРАКТИЧЕСКИЕ ЗАДАЧИ ОБНАРУЖЕНИЯ ОБЪЕКТОВ НА ИЗОБРАЖЕНИИ	7
1.1 Области применения методов обработки изображений.....	7
1.2 Способы получения изображений	12
1.3 Дефекты изображения	14
2 ОБЗОР МЕТОДОВ ОБНАРУЖЕНИЯ ОБЪЕКТОВ, ИНВАРИАНТНЫХ К АФФИННЫМ ПРЕОБРАЗОВАНИЯМ	17
2.1 Сравнение методов обнаружения объектов.....	17
2.1.1 Метод фильтрации.....	17
2.1.2 Метод ключевых точек.....	19
2.1.3 Метод с использованием нейронных сетей	20
2.1.4 Методы контурного анализа.....	21
2.2 Выделение контура.....	22
2.3 Преобразование Хафа	28
2.4 Обобщенное преобразование Хафа	31
3 РАЗРАБОТКА И АНАЛИЗ АЛГОРИТМОВ ОБНАРУЖЕНИЯ ОБЪЕКТОВ НА ОСНОВЕ МЕТОДОВ КОНТУРНОГО АНАЛИЗА.....	39
3.1 Разработка алгоритма на основе обобщенного преобразования Хафа	39
3.2 Анализ работы алгоритма при работе с изображениями, инвариантными к аффинным преобразованиям	45
3.3 Анализ работы алгоритма при работе с шумом на изображениях	48
ЗАКЛЮЧЕНИЕ	53
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	56
ПРИЛОЖЕНИЕ А Разработанные алгоритмы.....	57

ВВЕДЕНИЕ

Алгоритмы обнаружения объектов на изображении являются актуальными и востребованными в современном мире, так как они имеют множество применений в различных областях, таких как медицина, производство, биометрия, радиолокация и многие другие. Они помогают автоматизировать и оптимизировать различные процессы, уменьшая или упрощая работу человека. К тому же, такие программы помогают исключить человеческий фактор в некоторых ситуациях.

В настоящее время уже существуют готовые программы для обнаружения объектов на изображении в некоторых областях, однако часть из них является коммерческими и закрытыми, а какая-то часть не имеет открытого исходного кода, что позволило бы менять параметры и вносить изменения для проведения исследований и поиска наилучшего решения. Также существуют программы, которые имеют узкую специализацию, что не позволяет их применять в различных областях. Однако, до сих пор остаются задачи, для которых еще не разработаны программы. Причиной создания собственной реализации алгоритмов обнаружения объектов является необходимость применения их для обработки изображений в различных предметных областях, а также возможность их модернизации с учетом требований к обработке изображений, изменения показателей качества и эффективности работы.

В настоящее время описаны разнообразные алгоритмы, которые могут распознавать объекты на изображении по шаблону. Это корреляционные алгоритмы, которые используют сравнение яркостей пикселей образца и исследуемого изображения. Хотя они и просты в реализации, но не обладают устойчивостью к шуму, а также изменению масштаба и повороту. Следующий тип алгоритмов использует нейронные сети или другие методы машинного обучения для нахождения объектов на изображении. Алгоритмы машинного обучения в некоторых случаях бывают более эффективны, чем корреляционные или другие алгоритмы, но требуют большого количества данных для

обучения. Эффективность таких алгоритмов проявляется тогда, когда отсутствует четкая(аналитическая) информация, с помощью которой можно описать изображение. Еще одним алгоритмом является метод ключевых точек. Этот тип алгоритмов использует определенные признаки объектов (например, форму, текстуру, цвет и так далее) для обнаружения объектов на изображении. Алгоритмы на основе метода ключевых точек более эффективны, чем корреляционные алгоритмы, но требуют большого количества вычислительных ресурсов для обработки признаков.

Еще одним весьма популярным алгоритмом является метод контурного анализа. В нем используются контуры объектов для обнаружения объектов на изображении. Алгоритмы на основе анализа контуров временами более эффективны, чем корреляционные алгоритмы, но требуют большого количества вычислительных ресурсов для обработки данных, однако легко распознают объекты, инвариантные к повороту или изменению масштаба. Методы контурного анализа имеют ряд преимуществ перед другими алгоритмами. Это достаточно высокая точность, то есть количество ложных решений минимально, адаптивность, что говорит о том, что алгоритм может работать с любыми типами объектов, простота реализации. Поэтому работа сфокусирована на исследование именно методов контурного анализа.

Среди алгоритмов контурного анализа есть обобщенное преобразование Хафа, которое позволяет обнаруживать объекты любых форм, благодаря выделению контуров. Также алгоритм хорошо справляется со своей задачей, если на изображении присутствует шум или отсутствует часть данных, что делает данный метод устойчивым и надежным.

В выпускной квалификационной работе бакалавра основное внимание уделяется исследованию обобщенного преобразования Хафа, которое имеет преимущество перед другими методами контурного анализа. Данный алгоритм может обрабатывать изображения объектов любых форм, не требует высокопроизводительного оборудования, работает с изображениями любой яркости и контрастности, а также при наличии шума на изображении.

Целью работы является разработка эффективного алгоритма обнаружения объектов на изображении с использованием методов контурного анализа при влиянии различных дестабилизирующих факторов таких, как шум, поворот и изменение масштаба объекта на изображении.

Решение данной задачи разбилось на два этапа – это выделение контура на изображении и преобразование Хафа для обнаружения выделенного контура. Для того, чтобы получить хорошие результаты на каждом из этапов работы, необходимо предварительно обработать изображение, чтобы удалить шумы, помехи, вызванные искажениями отдельных элементов изображения и так далее.

Сглаживание шума, повышение контрастности и увеличение четкости границ объектов повышают точность при выделении контура. Однако, при использовании стандартной функции Matlab для выделения границ объекта, возникает проблема в виде разрывов этих границ. Ее решение стало еще одной задачей, которая была решена при выполнении выпускной квалификационной работы бакалавра. Данная проблема решилась с помощью применения методов дилатации и эрозии при предобработке изображения.

В работе были разработаны алгоритм выделения замкнутого контура на изображении и алгоритм обнаружения объектов, инвариантных к аффинным преобразованиям, на основе обобщенного преобразования Хафа. Также проведено исследование эффективности работы данных алгоритмов в условиях наличия шумов различной интенсивности.

1 ПРАКТИЧЕСКИЕ ЗАДАЧИ ОБНАРУЖЕНИЯ ОБЪЕКТОВ НА ИЗОБРАЖЕНИИ

1.1 Области применения методов обработки изображений

В работе проводится исследование задач обнаружения объектов на изображениях. Эта задача является довольно актуальной в настоящее время, потому что увеличивается количество программ и приложений, позволяющих дистанционно анализировать какие-либо ситуации, уменьшая или исключая человеческий фактор в данном вопросе.

В основном, системы по распознаванию объектов используются в компьютерном зрении. Цель компьютерного зрения – помочь машинам «увидеть» и «понять» содержание цифровых изображений и применяется оно в таких областях, как медицина, радиолокация, производство, сельское хозяйство, транспортная сфера и так далее. При этом решается довольно большой спектр задач.

Если говорить про медицину, то существуют различные диагностические системы, опирающиеся на обработку изображений. Такие системы помогают врачам интерпретировать медицинские снимки. Визуализация рентгенографии, магнитно-резонансной томографии, эндоскопии и ультразвуковой диагностики и прочих методов исследований дает большой объем информации, который необходимо специалисту оценить и проанализировать за короткое время. Системы распознавания объектов обрабатывают цифровые изображения и видео, выделяя важные участки, такие как возможные заболевания и симптомы, принося вклад в решение врача. Компьютерная медицинская диагностика уже помогает обнаруживать опухоли, проводить профилактические осмотры при маммографии, обнаруживать полипы при колоноскопии и исследовании легких и дыхательных путей. Распознавательные системы обычно не заменяют врача или другого специалиста, а скорее играют вспомогательную роль. Врач, как правило, несет ответственность за окончательную интерпретацию медицинских изображений. Тем не менее, целью некоторых систем является

выявление у пациентов самых ранних признаков аномалий, которые специалисты не могут обнаружить, например, при диабетической ретинопатии, «матовое стекло» на снимках компьютерной томографии легких и прочие нарушения в работе организма.

В одном из Индийских университетов разработан метод, благодаря которому, используя контур, полученный в результате обработки снимка компьютерной томографии легких, и оценку яркости узелков на этом контуре, можно определить наличие раковых опухолей, пример такого исследования представлен на рисунке 1.1.

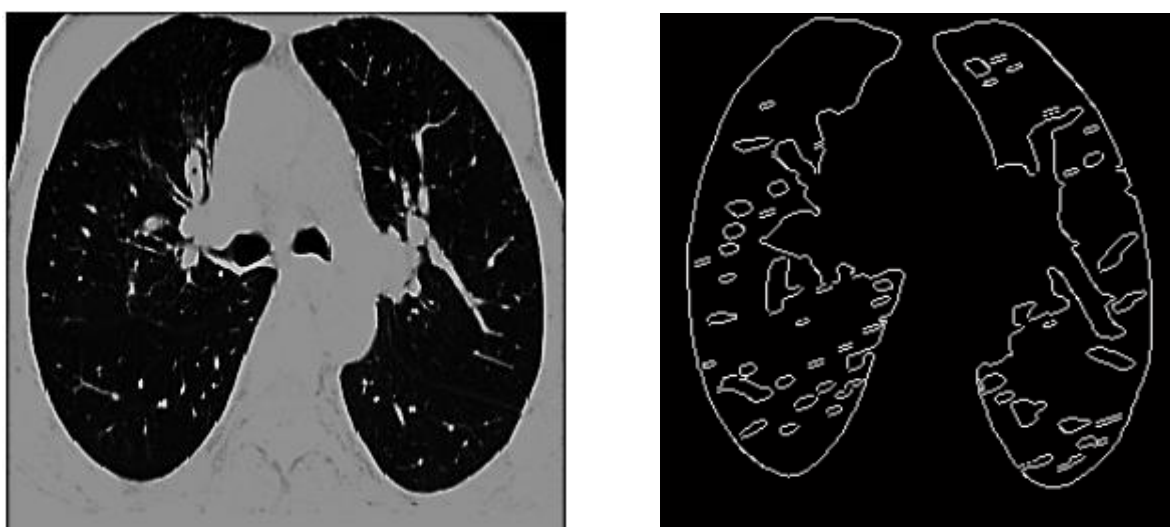


Рисунок 1.1 - Примеры изображений, в которых используется метод выделения контуров объектов

Другой сферой, в которой применяется метод обнаружения объектов, является радиолокация. В этой области решаются такие задачи, как обнаружение целей и их распознавание. Особенно остро эти задачи стоят в радиолокационных системах противовоздушной и противоракетной обороны. Для них существенно различать самолеты, вертолеты, ракеты друг от друга, а также от каких-либо других воздушных объектов вроде птиц, дронов и прочего. Зачастую так же важна информация и о классе, к которому принадлежит объект, то есть является ли он самолетом или вертолетом, транспортный ли это самолет, истребитель или бомбардировщик. В ряде

случаев существенно различие между крылатыми и баллистическими ракетами, боеголовками, беспилотными летательными аппаратами, транспортными самолетами, ракетами.

Пример распознавания корабля в течение нескольких этапов обработки на радиолокационных изображениях приведен на рисунке 1.2.

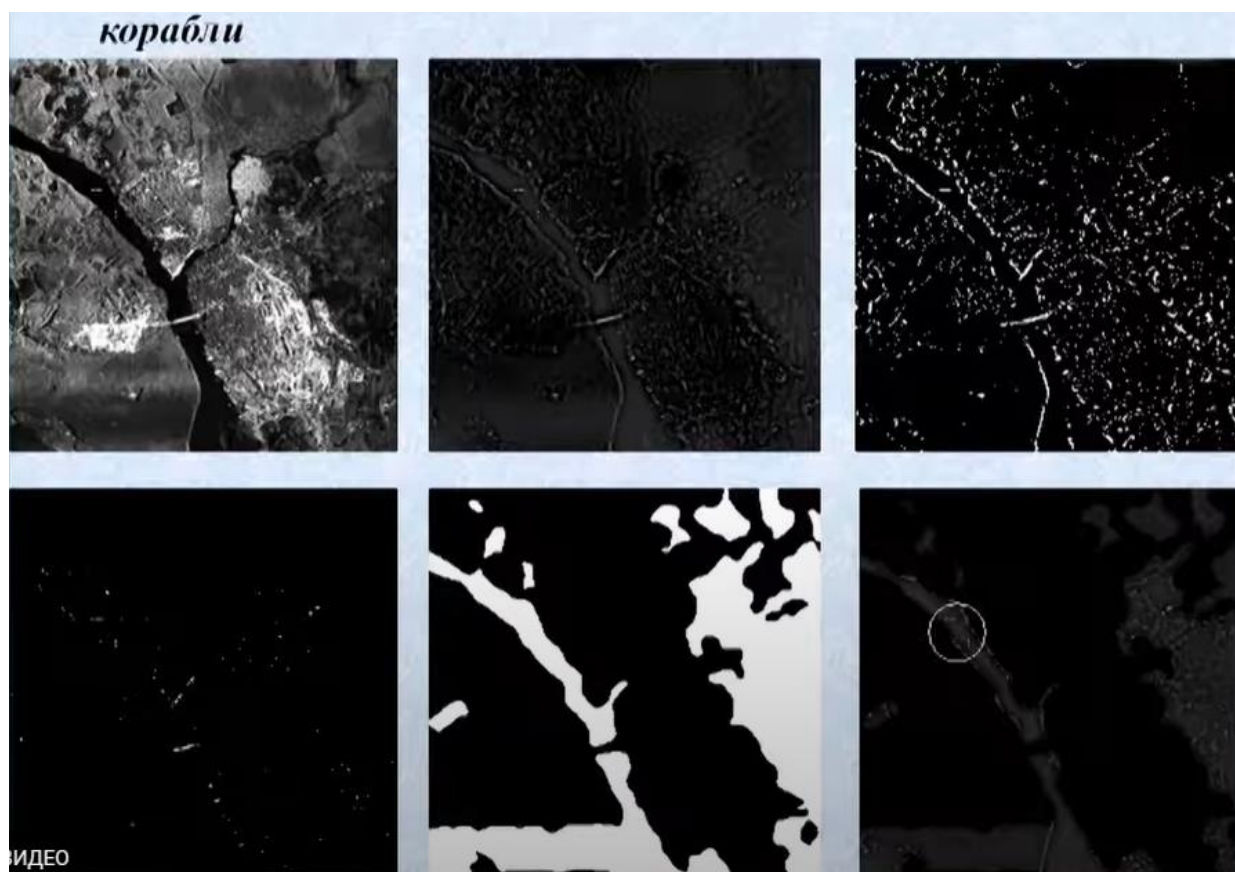


Рисунок 1.2 – Этапы обнаружения корабля на радиолокационном изображении

Также распознавание объектов применяется и на производстве. Такие программы помогают контролировать перемещения сотрудников по опасным зонам, например, зонам разгрузки-погрузки или путям транспортировки, помогают с обеспечением безопасности рабочих, то есть отслеживают наличие касок или спецодежды. Также алгоритмы могут распознавать маркировки и штрих-коды, например, для автоматической рассортировки товаров, но самое их частое применение – контроль качества. Контролировать можно как сам товар на наличие каких-либо физических дефектов, так и этапы

всего процесса производства. Одним из таких примеров может служить управление деревообрабатывающим станком на основе методов контурного анализа изображения. При обработке дерева идентифицируются дефекты по геометрическим формам, что можно увидеть на рисунке 1.3. Далее происходит автоматическое перемещение заготовки и спила дефекта, а после сортировка по размеру получившейся заготовки.

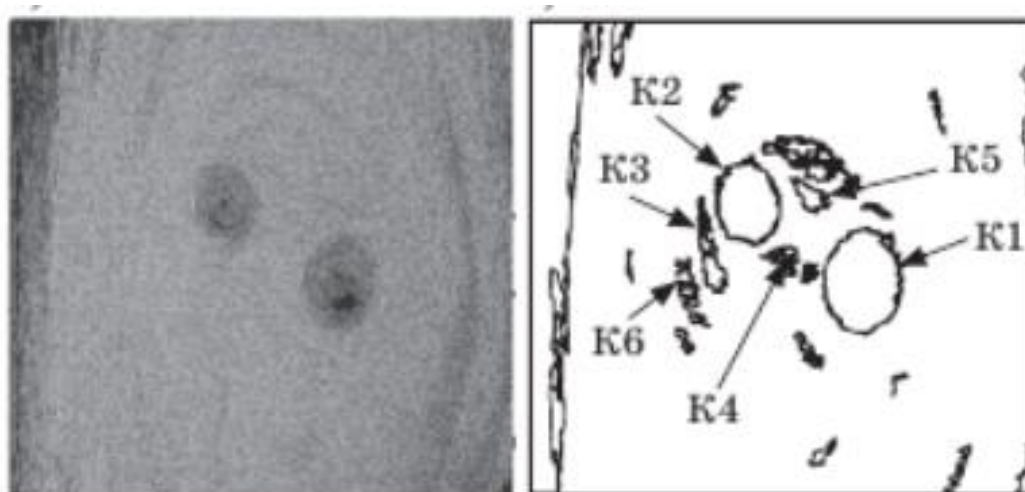


Рисунок 1.3 – Идентификация дефектов древесины с помощью методов контурного анализа

В транспортной сфере системы технического зрения находят широкую область применения. Наибольший интерес проявляется при разработке систем безопасности на дорогах, а также при решении логистических задач. Системы автоматического определения транспортных средств позволяют оценить размер транспортного средства, а по размеру провести классификацию типа транспортного средства. Для определения размера транспортного средства необходимо также знать расстояние до него, используя различные оптические системы наблюдения. При этом существует задача распознавания номерных знаков автомобилей. Такая задача может быть решена различными методами, эффективность которых оценивается в таблице 1.1.

Таблица 1.1 – Результаты тестирования алгоритмов распознавания номерных знаков автомобилей

Алгоритм	Тип распознаваемого номера		
	Нормальные(<30°)	Под углом(>30°)	С дефектом
	Точность/ошибка		
Контурный анализ	0,85 / 0,2	0,69 / 0,38	0,78 / 0,27
Метод Виолы-Джонса	0,89 / 0,42	0,70 / 0,57	0,75 / 0,43
Алгоритм	0,95	0,87	0,86

Одна из основных задач применения современных информационных технологий в области сельского хозяйства заключается в диагностике заболеваний сельскохозяйственных культур и прогнозирования их развития. Компьютерная система диагностирует заболевания плодовых культурных растений на основе обработки изображений их листьев. Применение автоматизированной системы диагностики позволяет использовать объективную диагностическую информацию, осуществлять более раннюю и достаточно точную диагностику заболеваний растений и принимать решения о проведении мероприятий по защите растений. Существует еще одна немаловажная задача в сельском хозяйстве, это отслеживание созревания урожая по снимкам. Она тоже решается с помощью распознавания объектов. На рисунке 1.4 представлен пример выделения контура помидора для дальнейшего его изучения. Система каждый день получает снимки, по которым делает оценку развития плодов, то есть производит сравнение положения и размера овоща по сравнению с образцом.



Рисунок 1.4 – Работа системы для мониторинга роста томатов по изображениям

Для работы всех описанных выше систем необходимы изображения. Они могут быть получены с помощью различных датчиков, которые будут рассмотрены в следующем пункте.

1.2 Способы получения изображений

Для получения изображения используют камеры, разного рода датчики и другие средства захвата изображений. Насколько много устройств используются для сбора данных и насколько они должны быть сложными зависит от задачи. Для простых задач иногда достаточно простой камеры.

Изображения могут быть получены с помощью сканеров, которые отличаются высоким качеством получаемого цифрового изображения, но обладают повышенной требовательностью к условиям эксплуатации. Формирование изображения происходит путем регистрации отраженного от объекта светового потока.

Цифровая фотокамера работает по такому принципу: свет, попадая на фотоэлемент, преобразуется в цифровую форму и записывается в память фотокамеры.

Если говорить об ультразвуковых исследованиях, то происходит посылка импульсов ультразвуковых волн и фиксация отраженного сигнала. Получается 2D-полутонное изображение. Так же можно получить и статичные трехмерные изображения.

Получение рентгеновских изображений основано на ослаблении рентгеновского излучения при его прохождении через различные ткани с последующей регистрацией на рентгеночувствительную пленку. В результате на пленке получается усредненное изображение всех исследуемых объектов.

Томографические изображения получают с помощью устройств двух типов на базе рентгеновского излучения и ядерного магнитного резонанса.

Для дистанционного зондирования используются различные системы, благодаря которым происходит формирование изображения исследуемой области. Это могут быть как фотографические системы, сканирующие системы видимого и инфракрасного диапазона, телевизионные оптические системы, радиолокационные системы бокового обзора, так и сканирующие сверх высокой частоты радиометры.

Снимки, полученные со спутников, могут иметь различное пространственное разрешение, что, в конечном счете, оказывает влияние на выбор метода, реализуемого программной системой распознавания. Высокое разрешение в несколько десятков метров характерно для наиболее широко используемых сканерных снимков с ресурсных спутников и фотографий пилотируемых кораблей, орбитальных станций, автоматических картографических спутников. Такое разрешение удовлетворяет требованиям большинства географических задач.

У каждого датчика есть свои особенности, но для многих изображений характерны расфокусировки и прочие погрешности. Все это говорит о том, что несмотря на то, какие изображения обрабатываются, нужно учитывать появление и дефектов.

1.3 Дефекты изображения

При разработке систем по обнаружению объектов можно столкнуться с определенными проблемами, и как правило, они связаны с качеством обрабатываемых изображений. Входные данные обычно могут иметь большое количество искажений, учитывая, что съемка проводится в неконтролируемых или естественных условиях. Могут быть такие дефекты, как расфокусировка, смазывание, цифровой шум, разладка регистратора, артефакты сжатия, блики, искажения, повреждения объекта обработки и так далее. Такие искажения изображений системы распознавания являются актуальной проблемой и могут приводить к полной потере информации. Современные системы распознавания оказываются неустойчивыми к размытию и шуму.

Шум возникает при регистрации изображений устройствами с фотосенсорами и заметен в виде наложенной маски пикселей случайного цвета и яркости. А также может появляться в результате оцифровки аналогового сигнала в результате теплового шума, ошибок квантования сигнала и переноса заряда на матрице с последующим усилением сигнала, а также из-за загрязнений матрицы, дефектных элементов сенсора и дальнейшего применения алгоритмов сжатия или обработки изображений.

Камеры используют изогнутые линзы для формирования изображения и световые лучи часто слишком сильно или слишком слабо изгибаются по краям этих линз. Это создает эффект, искажающий края изображений, так что линии или объекты кажутся более или менее изогнутыми, чем они есть на самом деле. Это называется радиальным искажением и является наиболее распространенным типом искажений, пример приведен на рисунке 1.5.



Рисунок 1.5 – Радиальное искажение изображения

Другой тип искажения – тангенциальное искажение. Это происходит, когда объектив камеры не выровнен идеально параллельно плоскости изображения, где находится датчик камеры. Из-за этого изображение кажется наклоненным, поэтому некоторые объекты кажутся дальше или ближе, чем они есть на самом деле.

Потеря качества происходит при сжатии изображения и при изменении размера или расширении изображения. В зависимости от использования алгоритмов сжатия могут возникать различные потери информации на изображении. Это может быть как сглаживание границ объекта, появление артефактов, так и незначительные изменения соотношения яркости и контрастности отдельных областей, что может привести к невозможности обнаружения объекта.

В зависимости от расположения камеры и объекта, изображение может быть частично или полностью расфокусировано. Солнце, которое долгое время попадает на фотообъектив, постепенно дает деформацию и приводит к размытости и нарушению контраста. Также движение объекта или же самой камеры может приводить к размытости изображения.

Широкое использование стеклянных стен и окон, различных зеркальных поверхностей в архитектуре и дизайне делает здание более эстетичным, приятным на вид и увеличивает световой эффект интерьера. Но в то же время

использование стекла увеличивает сложность сбора информации. В реальных боевых действиях часто необходимо собирать и идентифицировать информацию о личном составе противника внутри помещения через стеклянные окна, но тонированное стекло фильтрует или добавляет информацию к объекту. Различная степень изгиба стекла или неравномерная толщина приводят к изменению показателя преломления различных областей, вызывая искажение и деформацию исследуемого объекта с информацией о смещенном положении. В то же время наличие отраженных предметов или пятен на поверхности стекла будет искажать информацию об объекте, что приводит к отсутствию или потере информации в частях полученного изображения, что затрудняет достижения точной идентификации противника при таких обстоятельствах. Поэтому различные тени и блики, которые дают перепады яркости и засветы приводят к искажению или потере границ объекта, а иногда и к появлению фальшивых, что затрудняет обработку изображения.

При этом сам объект может иметь какие-либо внешние повреждения, артефакты на поверхности, которые могут повлиять на распознавание.

Также на качество распознавания влияет и контрастность самого объекта и фона. Как правило, реальные изображения, например, спутниковые снимки, отличаются существенной неоднородностью фона. Поэтому для таких изображений необходимо делать предобработку, что требует больших вычислительных затрат.

Погода тоже может влиять на качество распознавания объектов, так, например, современные алгоритмы не обнаруживают объекты, когда к изображению добавляется снег, даже если объекты хорошо видны глазу человека.

Несмотря на эти сложности, выпускная квалификационная работа бакалавра посвящена обнаружению объектов на изображении и для выполнения такой задачи существует много способов, которые будут описаны в следующем разделе.

2 ОБЗОР МЕТОДОВ ОБНАРУЖЕНИЯ ОБЪЕКТОВ, ИНВАРИАНТНЫХ К АФФИННЫМ ПРЕОБРАЗОВАНИЯМ

2.1 Сравнение методов обнаружения объектов

На сегодняшний день известно, что при разработке практических систем обнаружения объектов по образцу, используются алгоритмы с различными подходами. Это контурный анализ, фильтрация, алгоритмы на нейронных сетях, метод ключевых точек. Каждый из этих подходов имеет как свои плюсы, так и минусы, поэтому круг применения каждого ограничен.

2.1.1 Метод фильтрации

Если говорить про фильтрацию, то для поиска объекта по шаблону используется корреляционный анализ. Основное свойство алгоритма следующее: при наложении исходного изображения и образца их коэффициент корреляции достигает максимального значения, равного сумме квадратов яркостей всех пикселей изображения образца. Функция корреляции используется как средство поиска эквивалента шаблона на изображении. Данный алгоритм хорошо работает при зашумленности изображения, что приводит к снижению контрастности. При этом, если существуют отличия между шаблоном и объектом на исследуемом изображении, тогда дополнительно применяется метод пороговой обработки для обнаружения пиков на изображении с корреляционной оценкой. Однако метод корреляционной обработки является достаточно требовательным к вычислительным мощностям и имеет довольно низкую производительность. Другой его особенностью является отсутствие инвариантности к повороту и изменению масштаба объекта. Данную проблему можно решить, если рассматривать в качестве эталона все возможные повороты, но это еще больше увеличивает вычислительные затраты.

Существует подход, который реализует методы корреляционной обработки в частотной области и это согласованная пространственная фильтрация. В данном подходе используется фильтр, где матрица коэффициентов согласована с эталонным изображением искомого объекта.

Таким образом, при приложении такого фильтра к каждой части изображения, мы получаем значение взаимной энергии двух изображений – эталона и фрагмента. Соответственно, где больше значение энергии, там и располагается искомый объект. Хотя этот алгоритм и аналогичен корреляционному подходу, но есть и существенное отличие. При использовании корреляционной обработки вычисление коэффициента корреляции для каждого фрагмента значения параметров производится отдельно, однако импульсная характеристика согласованного фильтра одинакова для всех фрагментов изображения. Преимуществом метода согласованной фильтрации является маленькое время вычисления, а также отсутствие ложных результатов и хорошая работа даже при зашумленном изображении. Однако этот метод имеет и недостатки. Алгоритм согласованной фильтрации не является инвариантным к повороту изображения или изображению с измененным масштабом.

Таким образом несмотря на то, что методы фильтрации используют довольно небольшое количество ресурсов памяти и времени, они не являются инвариантными к повороту изображения и изменению его масштаба. Это сильно ограничивает применение подобных алгоритмов.

На рисунке 2.1 приведен пример, как с помощью корреляционного анализа происходит поиск здания на спутниковом снимке по шаблону.

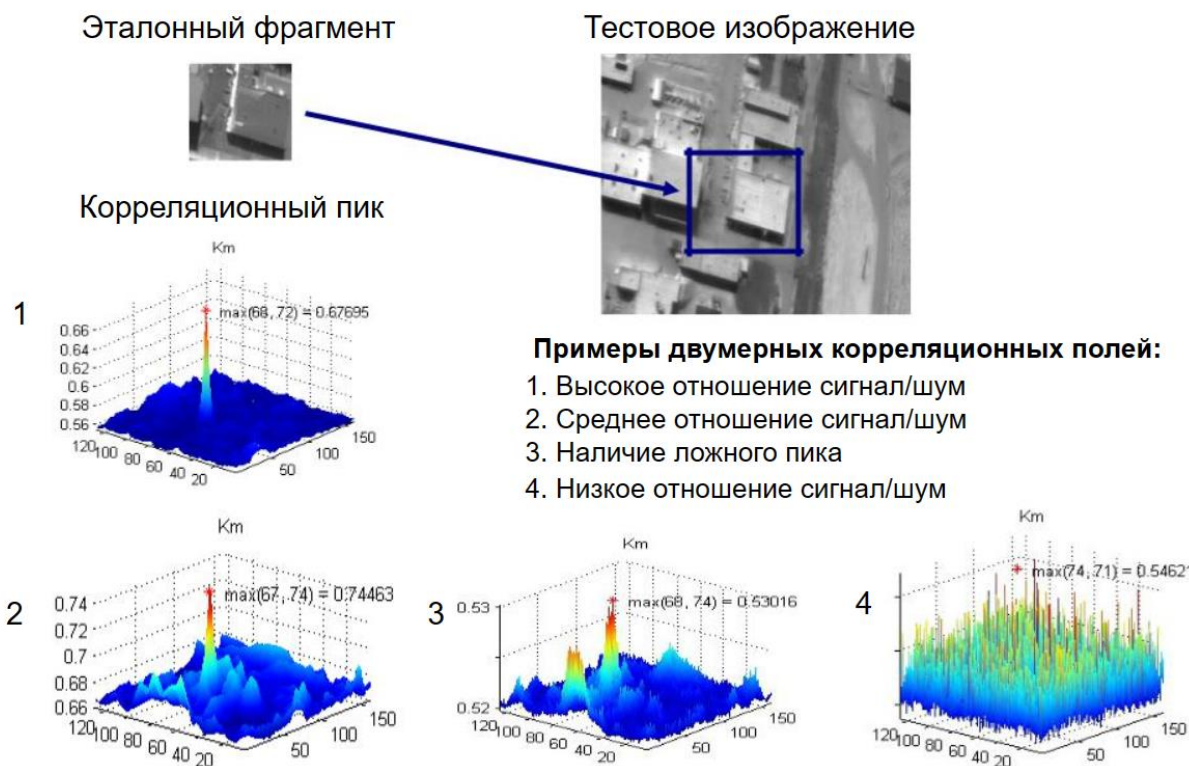


Рисунок 2.1 – Обнаружение объекта на изображении с помощью корреляционного анализа

2.1.2 Метод ключевых точек

Если рассматривать метод ключевых точек, то происходит вычисление абстракций изображения и на нем выделяются ключевые особенности. Данные особенности используются для сравнения двух изображений, благодаря чему можно выявить у них общие составляющие. Как такового определения ключевой особенности картинки нет, поэтому ей может быть как изолированная точка, так и какая-либо кривая или связанная область. Ключевые точки могут находиться различными алгоритмами. Метод лучше всего работает для объектов, которые не содержат большое число повторяющихся элементов, текстур и узоров, что может приводить к совпадению уникальных признаков. Этот метод так же плохо работает для объектов одного цвета. Он может обнаруживать объекты, несмотря на изменение масштаба, поворота на плоскости. Пример работы метода ключевых точек представлен на рисунке 2.2.

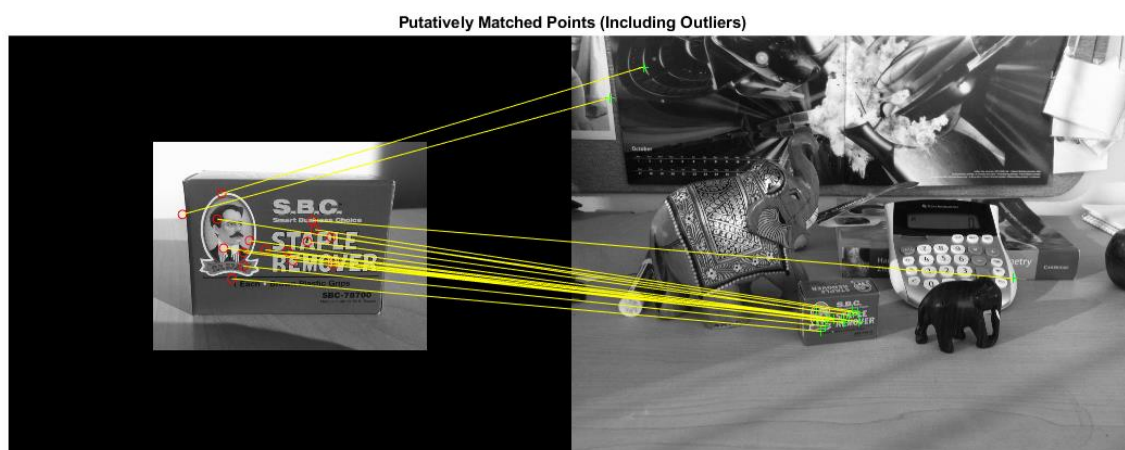


Рисунок 2.2 – Обнаружение объекта на изображении с помощью метода ключевых точек

2.1.3 Метод с использованием нейронных сетей

Говоря об алгоритмах на нейронных сетях, можно выделить один из них, например, каскад Хаара. Данный метод использует функции Хаара для определения вероятности того, что определенная точка является частью объекта. Этот алгоритм требует большого количества ресурсов, для обучения модели, а также чувствителен к повороту более 20 градусов. Однако хорошо справляется с зашумленными изображениями. Пример работы алгоритма представлен на рисунке 2.3.



Рисунок 2.3 – Метод обнаружение объекта на изображении с использованием нейронных сетей

2.1.4 Методы контурного анализа

Существует метод обнаружения объектов как контурный анализ. Он позволяет описывать и находить объекты, которые представлены в виде своих внешних очертаний – контуров. Контур содержит всю необходимую информацию о форме объекта, но внутренние точки объекта не учитываются. Это ограничивает область применения контурного анализа, но позволяет перейти от двумерного пространства изображений к пространству контуров, что позволяет очень сильно снизить алгоритмическую и временную сложность программы. Выделяют несколько методов: цепной код Фримена, обобщенное преобразование Хафа и прочие.

Цепной код Фримена позволяет представить контур в дискретном поле(решетке) с помощью последовательности отрезков прямых, которые представляются набором из восьми стандартных направлений от 0 до 7 в окрестности 3x3. Для построения кода фиксируется первая точка, далее совершается обход контура в выбранном направлении и описывается серией цифр. Благодаря численным методам можно совершать масштабирование, повороты или другие различные преобразования, необходимые для распознавания. Для распознавания контуров могут применяться методы сравнения кодов, которые основаны на корреляционных функциях и гистограммах. Недостатком такого метода является то, что он неточен в условиях наличия шума, потому что происходит накопление ошибок. Также страдает схема корреляции при сопоставлении двух цепочек, что происходит в результате влияния начальной точки, ориентации и масштаба на сам цепочный код.

Еще одним алгоритмом является обобщенное преобразование Хафа. Оно использует информацию о контуре для определения отображения точки контура в исходную точку фигуры. Опорную точку можно рассматривать как начало координат локальной системы для фигуры. Тогда есть простой способ вычисления показателей, которые оценивают вероятность того, что точки на

изображении будут точками формы. Особенностью преобразования является то, что оно будет работать даже тогда, когда граница зашумлена.

Методы контурного анализа позволяют решать основные проблемы распознавания образов – сдвиг, поворот и изменение масштаба изображения объекта. Контур не зависит от смены ракурса, изменений погодных условий, смены датчика и тому подобное. При этом сами алгоритмы не требуют больших вычислительных ресурсов.

Методы контурного анализа являются оптимальными для обнаружения объекта по шаблону, несмотря на неустойчивость к шуму, что исправляется предобработкой изображения. Поэтому в качестве исследуемого в работе метода выбран контурный анализ, а именно обобщенное преобразование Хафа.

2.2 Выделение контура

Для того, чтобы воспользоваться методами контурного анализа, необходимо выделить контур как на эталонном изображении, так и на том, где будет происходить обнаружение. Перед этим нужно выполнить предобработку изображений, чтобы повысить качество выделения контура, потому что реальные изображения не идеальны и имеют искажения или дефекты.

Основная проблема состоит в том, что у объектов чаще всего какая-либо часть границы размыта. Такое может происходить, например, когда объект сливается с фоном. При этом, когда происходит выделение контура, то появляются разрывы, что мешает получить координаты контура. Пример подобной ситуации представлен на рисунке 2.7.

Существует способ, который устраняет разрывы в контурах на изображении с помощью дилатации. Данный метод заключается в том, что происходит перемещение заданного фигурного элемента, например круга, по границе контура объекта. При таком подходе происходит расширение границ объекта, что приводит к потере точности, так как замыкается утолщенная граница, которая не является исходной.

Также существует подобный способ, но имеющий более высокую точность. Метод поочередно применяет операции дилатации и эрозии к изображению. То есть, как и в предыдущем случае, происходит перемещение заданного фигурного элемента, например круга, по границе контура объекта, но только дилатация и эрозия чередуются. Получается, что вначале граница расширяется, а потом сужается, что возвращает границу на первоначальное место. Таким образом происходит замыкание, но при этом контур немного сглаживается. Пример работы данного метода представлен на рисунках 2.8-2.10.

В результате получается замкнутый контур на изображении, который обрабатывается для получения координат границы.

В качестве примера работы алгоритма по выделению контура рассмотрим изображение листа на рисунке 2.4. Алгоритм обрабатывает изображение, что показано на рисунках 2.5 – 2.12. Применение этого же алгоритма, но при наличии шума на изображении представлено на рисунках 2.13 – 2.14.



Рисунок 2.4 – Изображение используется для применения на нем алгоритма выделения контура



Рисунок 2.5 – Рисунок 2.4 в оттенках серого при отсутствии шума

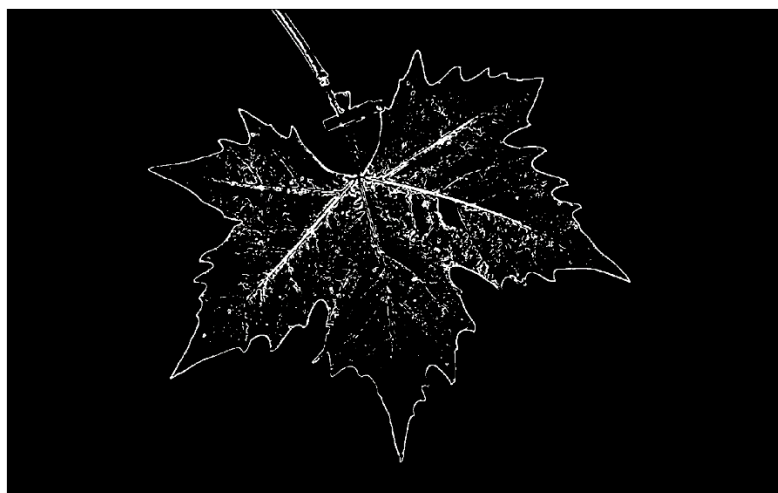


Рисунок 2.6 – Выделение контура с помощью стандартной функции Matlab на рисунке 2.5

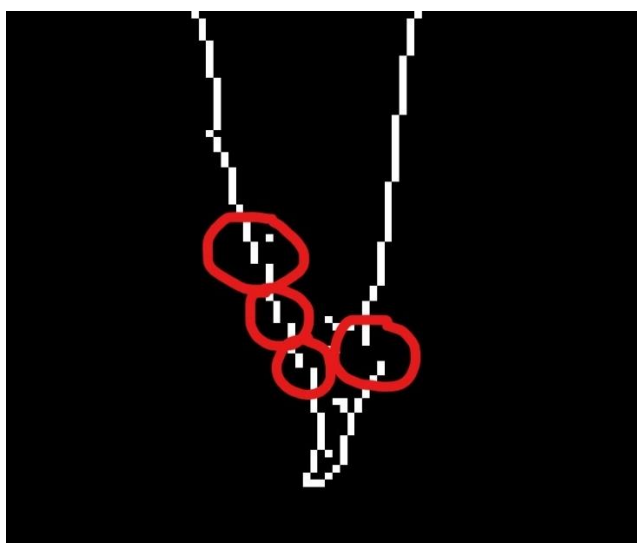


Рисунок 2.7 – Разрывы в контуре на рисунке 2.6

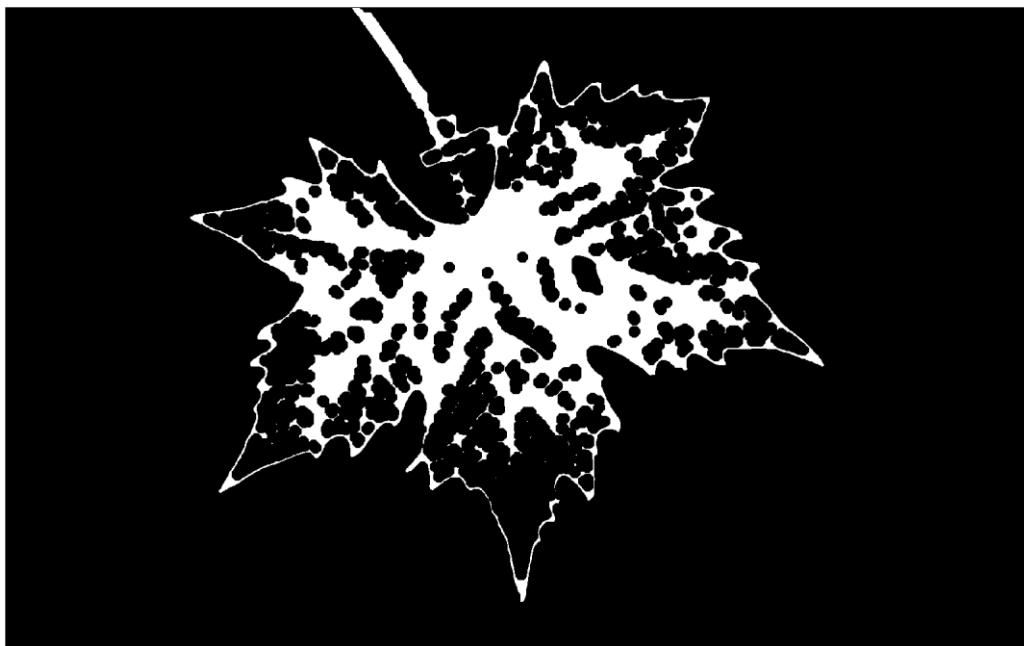


Рисунок 2.8 – Заливка внутри имеющегося контура(только замкнутых частей)



Рисунок 2.9 – Применение дилатации к рисунку 2.8

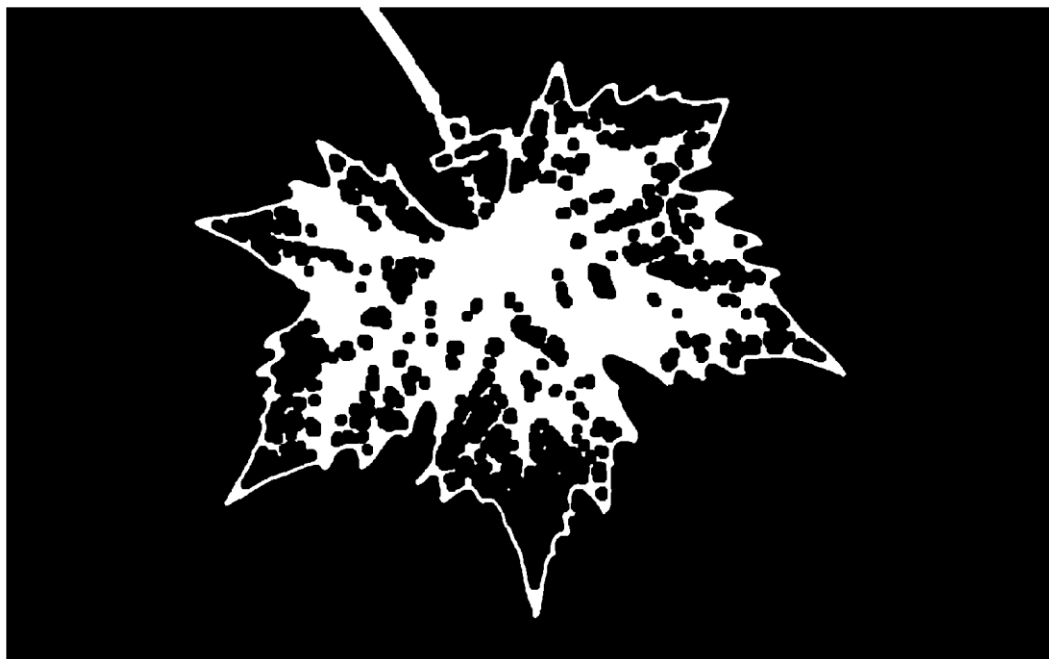


Рисунок 2.10 – Применение эрозии к рисунку 2.9

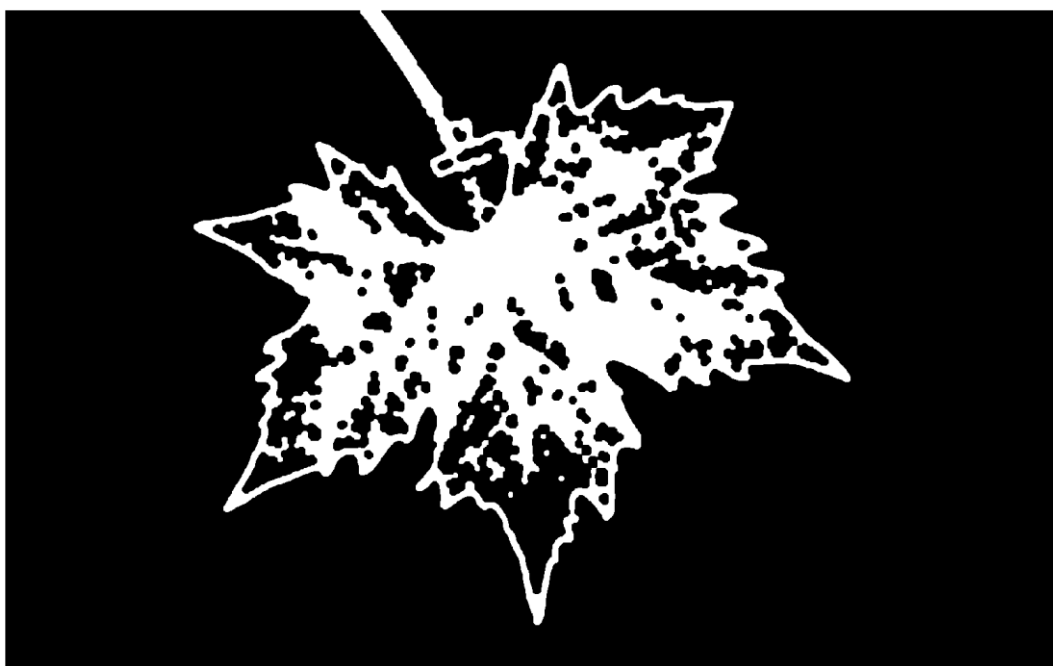


Рисунок 2.11 – Поочередное применение дилатации и эрозии к рисунку 2.10

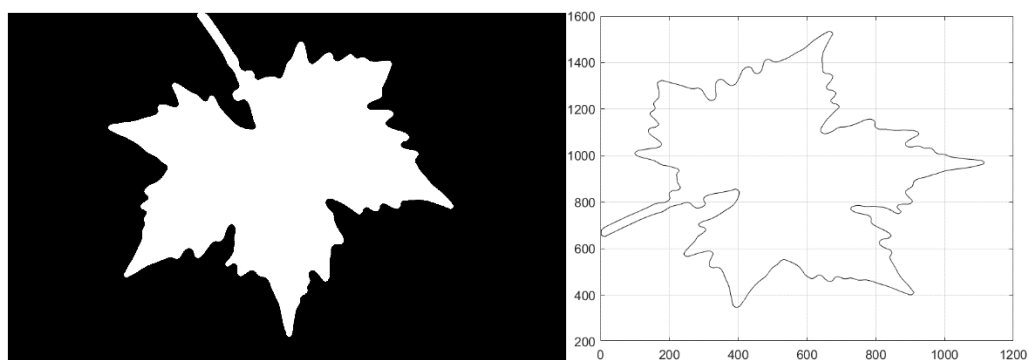


Рисунок 2.12 – Результат выделения контура для рисунка 2.4 при отсутствии шума



Рисунок 2.13 – Рисунок 2.4 при гауссовском шуме с дисперсией, равной 10% от максимального значения яркости

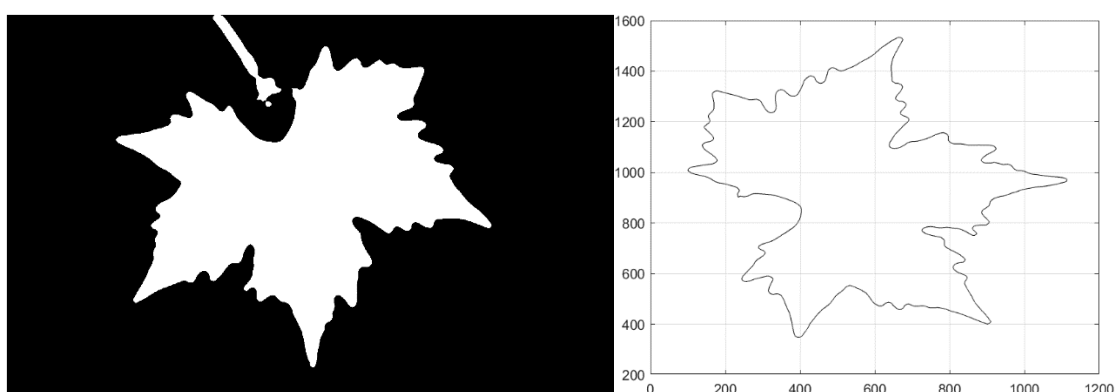


Рисунок 2.14 – Результат выделения контура для рисунка 2.4 при гауссовском шуме с дисперсией, равной 10% от максимального значения яркости

Как можно заметить, для того чтобы получить хороший результат при выделении контура, алгоритм получает точное бинарное изображение объекта, контур которого и выделяется на следующих шагах. При наличии гауссовского шума с дисперсией, равной менее 30% от максимального значения яркости программа справляется хорошо, однако не для всех изображений. Большую роль играет контрастность, потому что при маленьком различии в яркости изображений программе тяжело отделить объект от фона. Также наличие большого количества посторонних объектов может давать большую потерю информации о контуре, что мешает восстановлению целостности объекта. Блики и тени тоже приносят свой вклад в процесс выделения контура.

Не существует идеального и универсального алгоритма, который бы хорошо и точно обрабатывал изображения и выделял замкнутый контур. У фотографий из различных сфер деятельности есть свои особенности, которые влияют на параметры метода, что делает невозможным создание единой программы по обработке изображений.

2.3 Преобразование Хафа

Для того, чтобы разобраться в устройстве обобщенного преобразования Хафа необходимо понять, как работает простое. Изучим подробнее данный алгоритм.

Будем рассматривать аналитические кривые вида:

$$f(x, a) = 0, \quad (2.1)$$

где x – точка изображения, a – вектор параметров.

Чтобы увидеть, как работает преобразование Хафа для таких кривых, предположим, что нас интересует обнаружение круговых границ на изображении.

В декартовых координатах окружность имеет вид:

$$(x - a)^2 + (y - b)^2 = r^2 \quad (2.2)$$

Предполагается, что изображение было преобразовано в краевое представление, так что известна только величина локальных изменений интенсивности. Пиксели, величина которых превышает некоторый порог – краевые пиксели. Для каждого краевого пикселя можно задать вопрос: если этот пиксель лежит на окружности, то каково геометрическое место параметров этой окружности? Ответ: правильный круглый конус как на рисунке 2.15. Это можно увидеть из уравнения окружности, если рассматривать x и y как фиксированные значения, а a , b и r изменяемые параметры.

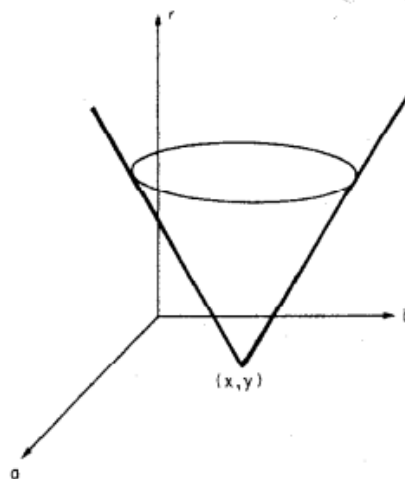


Рисунок 2.15 – Геометрическое место параметров без информации о направлении

Если использовать информацию о направлении, связанную с ребром, то это уменьшит геометрическое место параметров, что можно увидеть на рисунке 2.16. Это связано с тем, что центр окружности для точки (x, y) должен располагаться на r единиц вдоль направления градиента. Окружность имеет три параметра, но используя уравнение окружности вместе с его производной, количество параметров сократится до одного. Уравнение

$$\frac{df(x,a)}{dx} = 0 \quad (2.3)$$

вводит значение $\frac{dy}{dx}$, которое известно из

$$\frac{dy}{dx} = \operatorname{tg} \left[\varphi(x) - \frac{\pi}{2} \right], \quad (2.4)$$

где $\varphi(x)$ – направление градиента. Это предполагает следующий алгоритм.

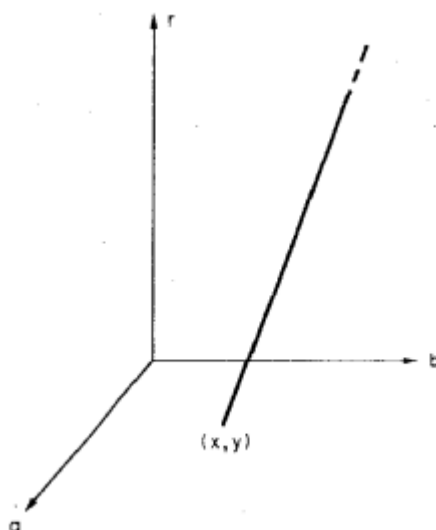


Рисунок 2.16 – Геометрическое место параметров с информацией о направлении

Для конкретной кривой, описываемой формулой (2.1), с вектором параметров \mathbf{a} формируется массив накопления $A(\mathbf{a})$, первоначально заполненный нулями. Затем для каждого краевого пикселя x вычисляются все \mathbf{a} такие, что должно выполняться уравнение (2.1) и

$$\frac{df(x,a)}{dx} = 0 \quad (2.5)$$

и увеличиваются соответствующие элементы массива, то есть

$$A(\mathbf{a}) = A(\mathbf{a}) + 1 \quad (2.6)$$

После рассмотрения каждого краевого пикселя \mathbf{x} локальные максимумы в массиве A соответствуют кривым $f(\mathbf{x}, \mathbf{a})$ на изображении.

2.4 Обобщенное преобразование Хафа

Ключом к обобщению алгоритма на сложные формы является применение информации о направлении. В качестве примера рассмотрим поиск круговой формы с фиксированным радиусом r_0 на произвольной форме, представленной на рисунке 2.17. Для каждой точки \mathbf{x} с направлением градиента φ нужно увеличить значение для точки на $\mathbf{x} + \mathbf{r}$, то есть $\mathbf{a} = \mathbf{x} + \mathbf{r}$. Таким образом теперь $\mathbf{r} = \mathbf{a} - \mathbf{x}$, которое будет меняться по величине и направлению в зависимости от граничной точки.

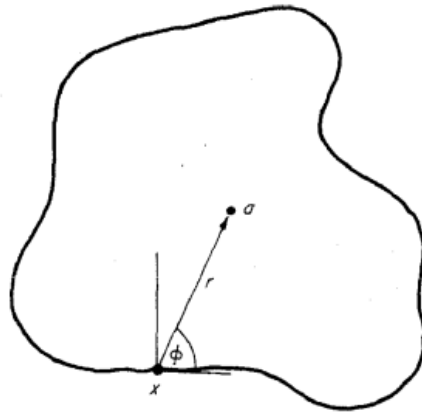


Рисунок 2.17 – Пример для обобщенного преобразования Хафа

Тот факт, что \mathbf{r} изменяется произвольным образом, означает, что обобщенное преобразование Хафа для произвольной формы лучше всего представляется в виде таблицы, которая называется R-таблицей.

R-таблица выстраивается путем изучения граничных точек формы. Выбирается опорная точка формы \mathbf{y} . Для каждой граничной точки \mathbf{x} вычисляется направление градиента $\varphi(\mathbf{x})$ и $\mathbf{r} = \mathbf{y} - \mathbf{x}$. \mathbf{r} сохраняется как функция φ . Отображение, которое представляет таблица, имеет векторное

значение и φ может иметь несколько возможных значений r . Схематичное представление R-таблицы можно увидеть в таблице 2.1.

Таблица 2.1 – Описание формата R-таблицы

i	φ_i	R_{φ_i}
0	0	$\{r a - r = x, x \in B, \varphi(x) = 0\}$
1	$\Delta\varphi$	$\{r a - r = x, x \in B, \varphi(x) = \Delta\varphi\}$
2	$2\Delta\varphi$	$\{r a - r = x, x \in B, \varphi(x) = 2\Delta\varphi\}$
...

R-таблица используется для обнаружения экземпляров формы S на изображении следующим образом. Рассматривается каждый краевой пиксель x в изображении, для которого увеличиваются все соответствующие точки $x + r$ в массиве аккумуляторов A , где r является записью в таблице с индексированным φ , то есть $r(\varphi)$. Максимумы в массиве A соответствуют возможным экземплярам формы S . Некоторые простые формы инвариантны к вращению, то есть записи в таблице приращений являются инвариантными функциями направления градиента φ .

Этот пример является подходящим только для форм с соответствующей ориентацией и соответствующим масштабом, то есть массив аккумуляторов является двумерным. Для поиска фигур произвольной ориентации θ и масштаба s в описание необходимо добавить эти параметры. Таким образом, массив аккумуляторов состоит из четырех значений, которые соответствуют параметрам (y, s, θ) .

Обозначим конкретную R-таблицу для формы S как $R(\varphi)$. Простые преобразования самой этой таблицы помогают обнаруживать масштабированные или перевернутые экземпляры одной и той же формы. Например, если форма имеет масштаб s , то преобразование обозначается как T_s , то есть

$$T_s[R(\varphi)] = sR(\varphi), \quad (2.7)$$

таким образом все векторы масштабируются по s . Также, если объект повернут на θ , то преобразование обозначается как T_θ , то есть

$$T_\theta[R(\varphi)] = \text{Rot}\{R[(\varphi - \theta) \bmod 2\pi], \theta\}, \quad (2.8)$$

так все индексы изменяются на $-\theta$ по модулю 2π , находятся соответствующие векторы \mathbf{r} и поворачиваются на θ .

Инверсии интенсивности фона фигуры также можно учитывать с помощью простой модификации R-таблицы. Индексы в таблице изменяются с φ на $(\varphi + \pi) \bmod 2\pi$ и

$$T_{fg}\{T_{fg}[R(\varphi)]\} = R(\varphi), \quad (2.9)$$

где T_{fg} – преобразование интенсивности фона фигуры.

Еще одно свойство, которое будет полезно при описании композиции преобразований это изменение точки отсчета. При выборе новой точки отсчета \mathbf{y}' такой, что $\mathbf{y} - \mathbf{y}' = \mathbf{r}$, модификация R-таблицы задается, как $R(\varphi) + \mathbf{r}$, то есть \mathbf{r} добавляется к каждому вектору в таблице.

Если рассматривать составную форму S , состоящую из частей S_1 и S_2 . Эту форму можно определить с помощью R-таблиц для S_1 и S_2 . Если $\mathbf{y}, \mathbf{y}_1, \mathbf{y}_2$ опорные точки для форм S, S_1 и S_2 соответственно, то можно вычислить $\mathbf{r}_1 = \mathbf{y} - \mathbf{y}_1$, $\mathbf{r}_2 = \mathbf{y} - \mathbf{y}_2$. Тогда составное обобщенное преобразование Хафа $R_S(\varphi)$ задается выражением

$$R_S(\varphi) = [R_{S_1}(\varphi) + \mathbf{r}_1] \cup [R_{S_2}(\varphi) + \mathbf{r}_2], \quad (2.10)$$

что означает, что для каждого значения индекса φ к $R_{S_1}(\varphi)$ добавляется r_1 , к $R_{S_2}(\varphi)$ добавляется r_2 и объединение этих множеств хранится в $R_S(\varphi)$.

Таким образом, мы можем определить фигуры как разницу между таблицами с общими записями, то есть

$$R_S = R_{S_1} - R_{S_2} \quad (2.11)$$

обозначает форму S , определяемую формой S_1 с удаленными общими элементами S_2 . Аналогично определяется операция пересечения. Основное использование операции объединения заключается в обнаружении фигур, которые являются составными частями более простых фигур. Однако операция разности также выполняет полезную функцию. Используя ее, можно построить R-таблицы, которые явно различают два похожих типа фигур. Примером может служить различие между шайбами и шестиугольниками.

Однако, при использовании такого преобразования можно потерять точность в зависимости от выбора опорной точки. При удалении опорной точки от формы небольшие угловые ошибки приводят к большим. Одним из решений этой проблемы является использование таблицы для каждой подформы с ее собственной лучшей контрольной точкой и сглаживание результирующего массива аккумулятора с помощью составного шаблона сглаживания.

Обозначим шаблон сглаживания как $h_i(y_i)$ для опорной точки y формы S , тогда шаблон свертки задается формулой

$$H(y) = \sum_{i=1}^N h_i(y - y_i) \quad (2.12)$$

Рассмотрим обобщение алгоритма для обнаружения фигуры S , состоящей из частей S_1, \dots, S_N . Для каждой точки края с направлением

градиента φ и для каждого значения масштаба s и ориентации θ , увеличить значения для точек $\mathbf{x} + \mathbf{r}$ в массиве A , где

$$R_S(\varphi) = T_S \left\{ T_\theta \left[\bigcup_{k=1}^N R_{S_k}(\varphi) \right] \right\} \quad (2.13)$$

Максимумы в $A_S = A \cdot H$ соответствуют возможным экземплярам формы S .

Если используется стратегия увеличения массива аккумуляторов на единицу, то содержимое массива аккумуляторов будет примерно пропорционально периметру фигуры, обнаруживаемой на изображении.

Эта стратегия направлена на поиск форм, где можно обнаружить большую часть периметра. Доступны несколько различных стратегий приращения, в зависимости от различного качества данных изображения. Если обнаружены более короткие, очень выступающие части периметра, как это может быть в случае с частично закрытыми объектами, то альтернативная стратегия увеличения значения модуля градиента может быть более успешной, т. е.

$$A(\mathbf{a}) := A(\mathbf{a}) + g(\mathbf{x}) \quad (2.14)$$

Так же можно комбинировать стратегии, например

$$A(\mathbf{a}) := A(\mathbf{a}) + g(\mathbf{x}) + c, \quad (2.15)$$

где c - константа.

Другой возможностью является использование информации о локальной кривизне в функции приращения. Используя эту стратегию, проверяются соседние краевые пиксели для вычисления приблизительной кривизны K . Это требует более сложного оператора, чем рассмотренные нами

краевые операторы, и усложняет таблицу. Теперь вместе с каждым значением r должны храниться соответствующие значения кривизны. Затем приращение взвешивает «информативные» краевые пиксели с высокой локальной кривизной следующим образом:

$$A(a) := A(a) + K \quad (2.16)$$

При определенных обстоятельствах может появиться необходимость взвесить локальную информацию, которая непротиворечива. Например, при поиске границы этого объекта связанный набор ребер, соответствующий объекту, может быть более важным, чем набор несвязанных ребер. Например, это может возникнуть в ситуациях с очень зашумленными данными. Простая стратегия для обработки этого случая состоит в том, чтобы явно записывать контрольные точки для каждого краевого пикселя во время первого прохода. Затем при втором проходе краевые пиксели могут увеличиваться более чем на единицу, если соседние краевые пиксели увеличивают одну и ту же опорную точку.

Более сложной стратегией является поиск связанных сегментов кривой в пространстве изображения, которые имеют совместимые параметры. Но этот алгоритм основан на динамическом программировании и описан Баллардом и Склански. В алгоритме динамического программирования на каждом шаге итерации можно построить более длинные совместимые кривые из всех точек ребра. Таким образом, функция приращения для точки x будет представлять собой самую длинную совместимую кривую из этой точки. В параллельной реализации этого алгоритма содержимое массива аккумуляторов может динамически изменяться. Первоначально содержимое будет отражать глобальную информацию, но при последующих итерациях содержимое будет взвешиваться в пользу согласованной локальной информации.

При поиске составного объекта разные части могут иметь разное значение. Это легко приспособить, связав вес w_i с каждой таблицей R_{S_i} , чтобы

каждая запись в R_{S_i} увеличивалась на коэффициент w_i , а не на единицу. Составной объект можно искать последовательным образом. Последовательное применение таблицы может значительно повысить эффективность вычислений за счет ограничения областей для последующих подходящих приращений. Кроме того, стандартные методы могут быть использованы для остановки процесса после того, как форма будет определена с желаемым уровнем достоверности. Возможны и более сложные стратегии.

В работе был описан метод обнаружения экземпляров формы S на изображении, который является обобщением преобразования Хафа. Это преобразование является отображением краевого пространства в пространство-аккумулятор, так что экземпляры S создают локальные максимумы в пространстве-аккумуляторе. Это отображение удобно описать как таблицу соответствия опорных точек ориентации ребер, называемую R-таблицей. Этот метод имеет следующие свойства:

- 1) Изменения масштаба, повороты, инверсии фигуры и фона и перенос точки отсчета S могут быть обнаружены прямыми преобразованиями в R-таблице.
- 2) По заданной границе формы ее R-таблица может быть легко построена и это требует пропорционального количеству граничных точек количества операций.
- 3) Фигуры хранятся как канонические формы: экземпляры фигур обнаруживаются, исходя из знаний о преобразовании канонической формы в экземпляр. Если это преобразование неизвестно, необходимо попробовать все возможные преобразования.
- 4) Если фигура S рассматривается как составная часть нескольких форм S_1, \dots, S_N , то R-таблица обобщенного преобразования Хафа для S может быть просто построена путем объединения R-таблиц для S_1, \dots, S_N .
- 5) Составная форма S может быть эффективно обнаружена последовательным образом путем постепенного добавления R-

таблиц для частей S к алгоритму обнаружения, пока не будет достигнут желаемый уровень достоверности.

- 6) Значения таблицы могут быть взвешены с точки зрения локально согласованной информации.
- 7) Важность подформы S можно регулировать, связывая вес w_i с R -таблицей.

3 РАЗРАБОТКА И АНАЛИЗ АЛГОРИТМОВ ОБНАРУЖЕНИЯ ОБЪЕКТОВ НА ОСНОВЕ МЕТОДОВ КОНТУРНОГО АНАЛИЗА

3.1 Разработка алгоритма на основе обобщенного преобразования Хафа

В качестве изображения, на котором будет происходить обнаружение, рассматривается фотография с напечатанным текстом, которая представлена на рисунке 3.1. В качестве распознаваемого объекта выбрана буква «а», показанная на рисунке 3.2.



Рисунок 3.1 – Исходное изображение, на котором выполняется поиск объектов



Рисунок 3.2 – Шаблон объекта, поиск которого будет производиться

Для того, чтобы произвести обнаружение объекта, необходимо выделить контура на исходном изображении, на изображении с шаблоном. Выделение контура текста представлено на рисунке 3.3. Далее, алгоритм обрабатывает изображения и получается массив аккумуляторов, который представлен на рисунке 3.4. По нему происходит поиск максимума, значения которого показывают место, где находится искомый объект на изображении.

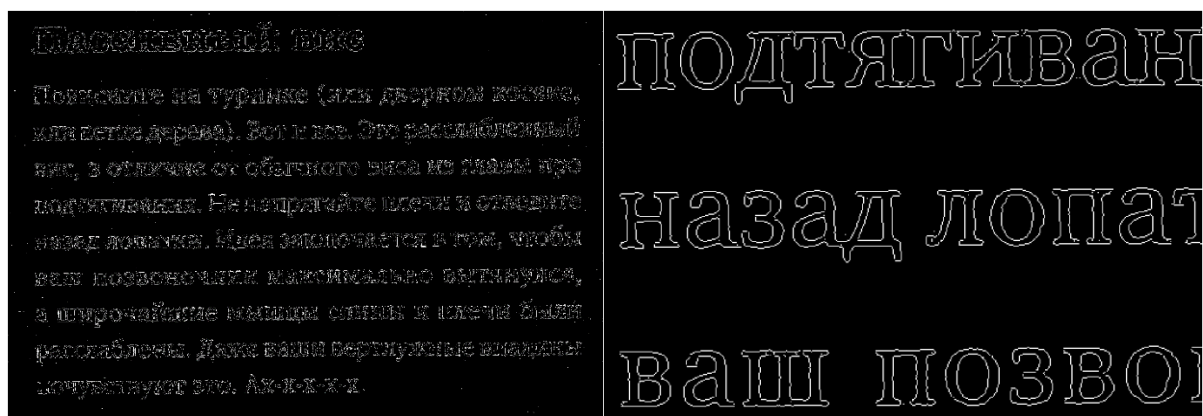


Рисунок 3.3 – Выделенный контур на рисунке 3.1

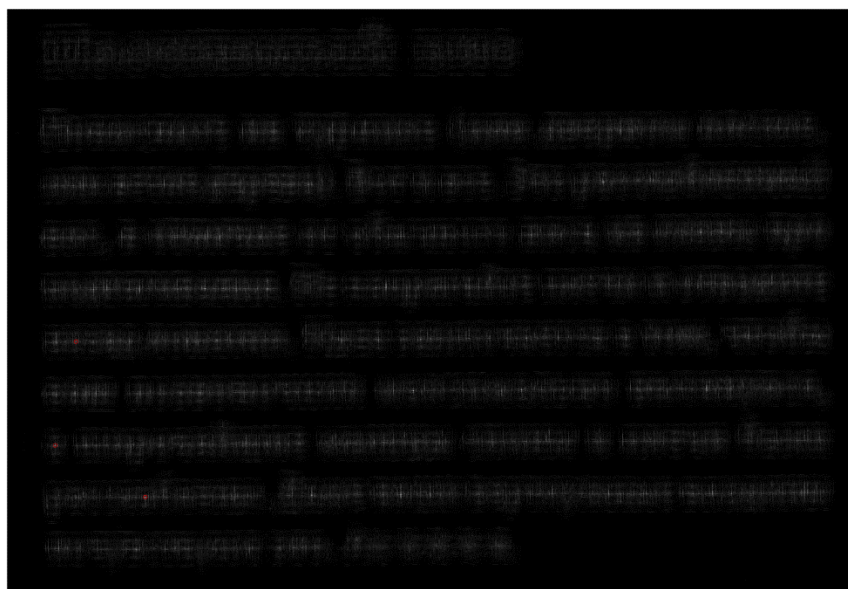


Рисунок 3.4 – Отображение аккумуляторов

На рисунке 3.5 представлены результаты по обнаружению буквы «а» на изображении. Программой были показаны несколько первых наилучших совпадения. Как можно заметить, что все решения расположены в левой части изображения, это произошло из-за того, что правая часть фотографии немного размытая, что дает менее точный контур, поэтому и точность совпадения объектов из правой части и шаблона ниже.

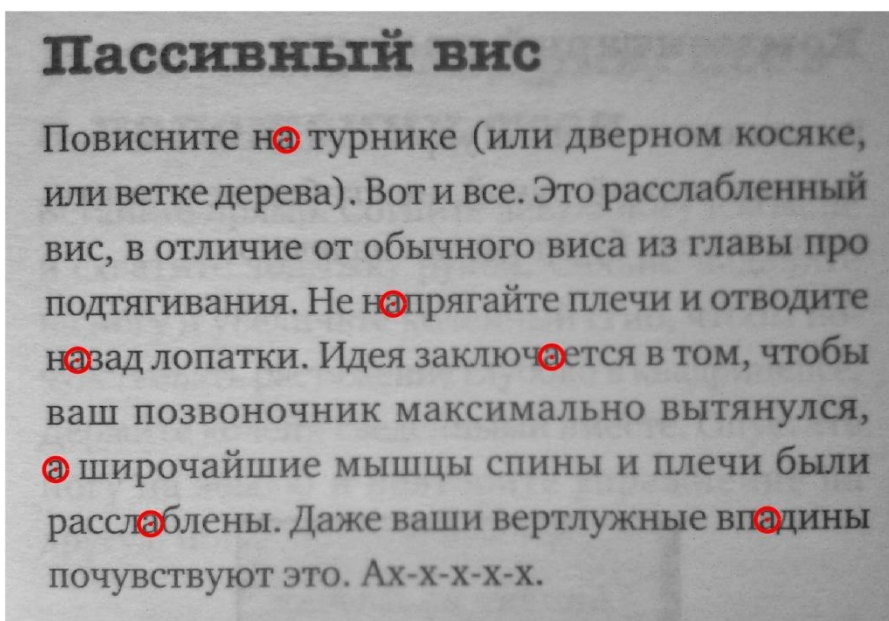


Рисунок 3.5 – Результат работы алгоритма, отображаются несколько наилучших результатов

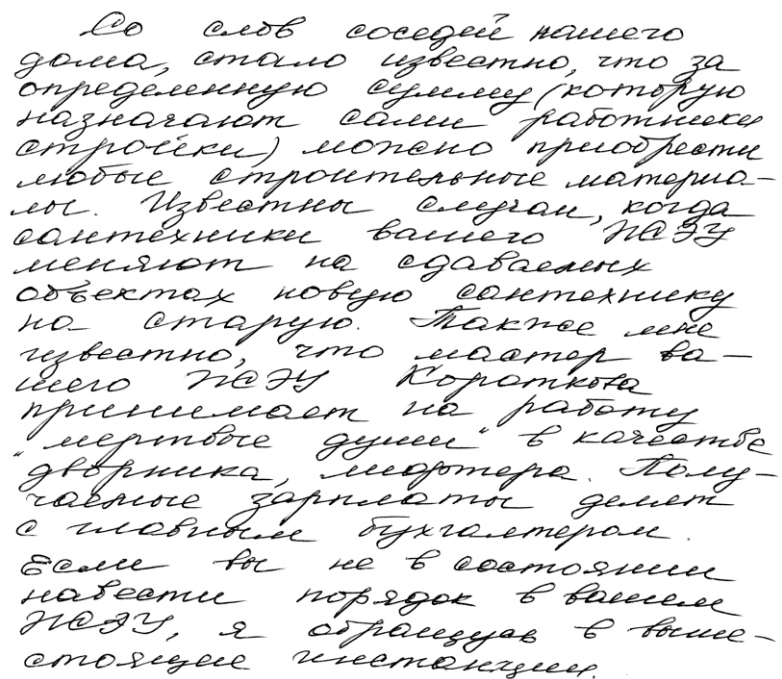
Для того, чтобы убедиться в том, что результат зависит от качества изображения, а не от алгоритма, можно воспользоваться «идеальным» изображением, представленном на рисунке 3.6.



Рисунок 3.6 – Применение алгоритма к «идеальному» изображению

Печатные буквы являются довольно простыми объектами для обнаружения, поэтому можно рассмотреть рукописные буквы. Эта задача намного сложнее, потому что человек не пишет одинаковыми буквами, в отличии от машины. Каждый символ, выведенный рукой человека, является

уникальным, потому что может меняться наклон, размер буквы, толщина линии. В качестве примера использовалось отсканированное изображение рукописного текста, показанное на рисунке 3.7.



Со слов соседей нашего
дома, стало известно, что за
определенную сумму (которую
называют сами работники
стройки) можно приобрести
любые строительные материа-
лы. Известно также, когда
сантехники вашего № 33
меняют на сдаваемых
объектах новую сантехнику
на старую. Также же
известно, что мастер Ва-
шего № 33 Короткого
принимает на работу
"мертвые души" в качестве
дворника, моторера. Полу-
чаемые зарплату денег
с машинным бухгалтером.
Если вы не в состоянии
навести порядок в вашем
№ 33, я образую в ваше-
м месте и место.

Рисунок 3.7 – Изображение с рукописным текстом,
где обнаруживается буква «с»

Результат работы алгоритма можно увидеть на рисунке 3.8. Как можно заметить, выделенные буквы являются очень похожими, но не идентичными. Они наиболее приближены к тому образцу, который использовался в качестве шаблона.

Со слов соседей нашего дома, стало известно, что за определенную сумму (которую называют одни работники строики) можно приобрести любые строительные материалы. Известно также, когда сантехники вашего НИИ меняют на сдвигаемых объектах новую сантехнику на старую. Также мне известно, что мастер вашего НИИ Короткого принимает на работу "мертвые души" в качестве дворника, мототера. Получившие зарплату деньги с машинным бухгалтером. Если вы не в состоянии навести порядок в вашем НИИ, я обращаюсь в вышестоящие инстанции.

Рисунок 3.8 – Результат обнаружения, отображены 3 наилучших совпадения

Можно рассмотреть более сложные формы, например птиц. На рисунке 3.9 представлена фотография, по которой будет происходить обнаружение птиц, имеющих похожее положение, как на рисунке 3.10. Результат обнаружения можно увидеть на рисунке 3.11. Алгоритм нашел не только птицу, изображенную на шаблоне, но и других, которые имеют очень похожее расположение, взмах крыльев. Количество и точность обнаруживаемых объектов зависит от порога, по которому фильтруется результат работы алгоритма. Таким образом, можно обнаруживать не только конкретный объект, а еще и очень близкий по форме.



Рисунок 3.9 – Фотография птиц, на которой будет производиться обнаружение



Рисунок 3.10 – Шаблон, по которому будет производиться обнаружение



Рисунок 3.11 – Результат работы алгоритма

Программа хорошо справилась с распознаванием букв, как машинных, так и рукописных. Она без проблем справляется с распознаванием сложных форм за довольно короткий промежуток времени и при небольших затратах памяти. Эти факторы являются довольно большим преимуществом перед другими алгоритмами при распознавании объектов.

3.2 Анализ работы алгоритма при работе с изображениями, инвариантными к аффинным преобразованиям

Используемый алгоритм обладает возможностью обрабатывать повернутые изображения, изображения с измененным масштабом. В качестве примера будут использоваться преобразованные изображения шаблона, показанные на рисунках 3.12 - 3.14.



Рисунок 3.12 – Рисунок 3.10, повернутый на угол 90 градусов



Рисунок 3.13 – Рисунок 3.10, увеличенный в масштабе



Рисунок 3.14 – Рисунок 3.10, увеличенный в масштабе и повернутый на 90 градусов

Результаты работы алгоритма представлены на рисунках 3.15-3.17 соответственно. В качестве решения отображены три наилучших результата работы программы.



Рисунок 3.15 – Результат поиска шаблона, показанного на рисунке 3.12



Рисунок 3.16 – Результат поиска шаблона, показанного на рисунке 3.13



Рисунок 3.17 – Результат поиска шаблона, показанного на рисунке 3.14

На рисунках выше были рассмотрены шаблоны, к которым применялись геометрические преобразования, то есть повернуто и увеличено было само эталонное изображение. На рисунке 3.18 представлен результат работы алгоритма, когда изображение, на котором будет происходить обнаружение, перевернуто на 180 градусов. В качестве шаблона использовалось изображение, показанное на рисунке 3.14. Программа отобразила три наилучших полученных результата.



Рисунок 3.18 – Результат поиска шаблона, показанного на рисунке 3.14

Как можно заметить, алгоритм находит объект, несмотря на поворот и изменение масштаба. Это является большим преимуществом перед другими методами и дает возможность применения к обширному числу областей.

3.3 Анализ работы алгоритма при работе с шумом на изображениях

В качестве примера для исследования работы программы с шумом на изображении рассматривается рисунок 3.1. В качестве шума использовался гауссовский, наложение производилось стандартной функцией Matlab.

Для того, чтобы добиться хорошего результата при поиске на зашумленном изображении, необходимо предварительно его обработать, то есть сгладить изображение, не сильно потеряв при этом четкость контура. Однако, если шум имеет большую дисперсию, то даже хорошая обработка не сможет достаточно точно восстановить контур. На рисунках 3.19 и 3.20 представлен результат поиска объекта на зашумленных изображениях. Уровень дисперсии шума на втором изображении значительно выше, чем на первом, что делает контур букв более расплывчатым и менее точным.

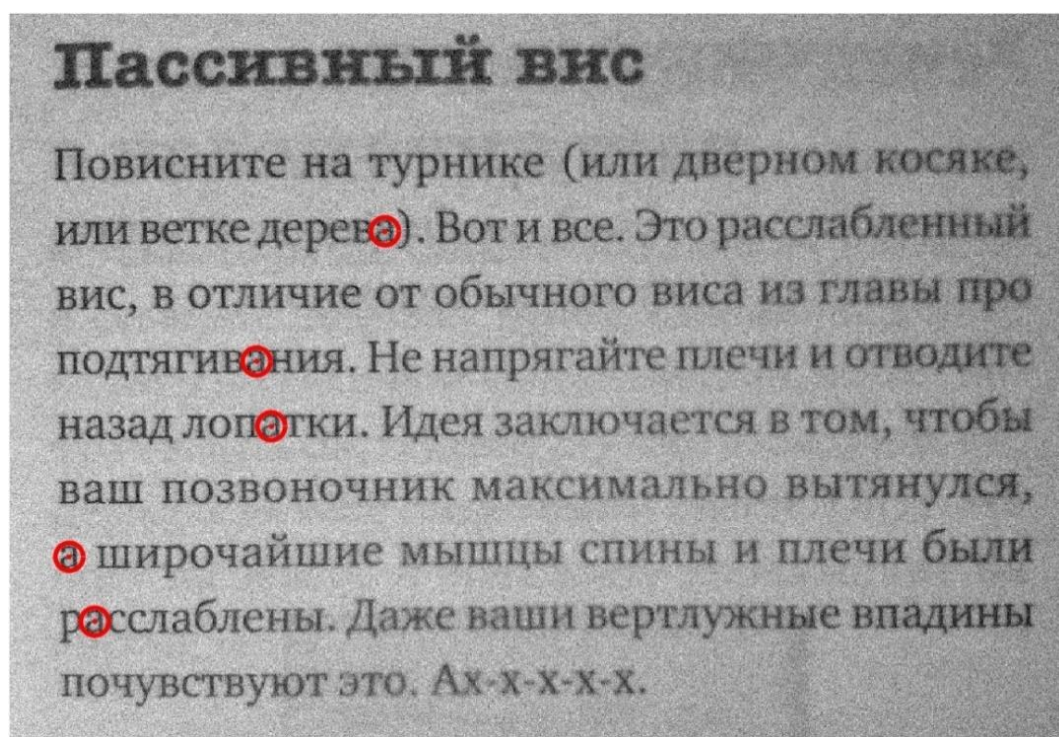


Рисунок 3.19 – Результат работы алгоритма при наложения гауссовского шума с дисперсией 0,01 на рисунок 3.1

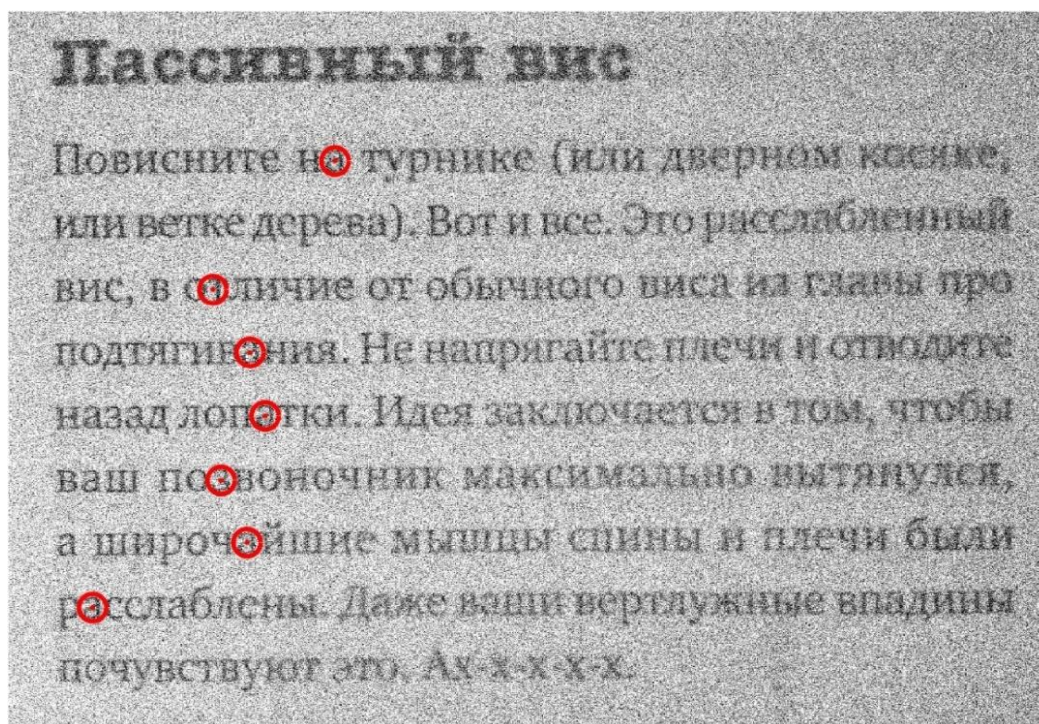


Рисунок 3.20 – Результат работы алгоритма при наложении гауссовского шума с дисперсией 0,07 на рисунок 3.1

На рисунках 3.21, 3.22 представлено выделение контура в случае, если не используется предварительная обработка изображения и, если производится фильтрация соответственно. Как можно заметить, при наличии шума, алгоритм не может четко выделить контур, происходит частичная потеря данных, а также появление ложных. При высокой дисперсии шума происходит полная потеря данных, как показано на рисунке 3.23. Это значительно влияет на дальнейшую работу алгоритма. В качестве обработки, представленной на рисунке 3.22, использовались различные методы фильтрации. Это медианная фильтрация и фильтр с гауссовым ядром сглаживания. Для этого использовались стандартные функции в Matlab. Чем больше дисперсия шума, тем более тщательная предобработка изображения требуется для достижения хороших результатов. При сильном зашумлении методы фильтрации размывают границы объектов, что приводит к потере точности, в разы ухудшая результат обнаружения.

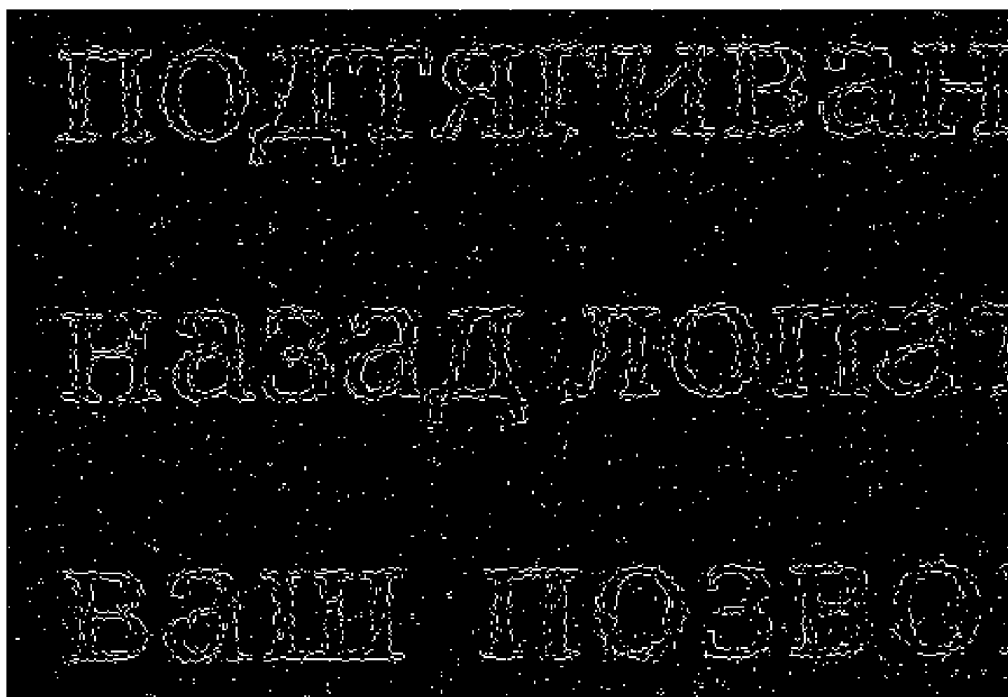


Рисунок 3.21 – Выделение контура на рисунке 3.19 без предварительной обработки изображения

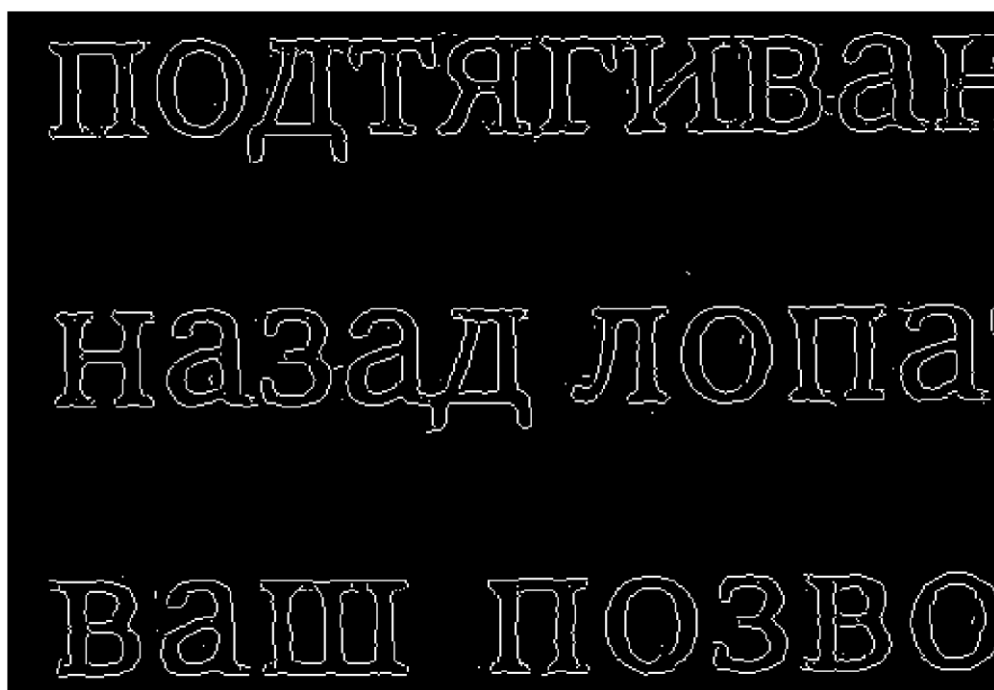


Рисунок 3.22 – Выделение контура на рисунке 3.19 с предобработкой изображения



Рисунок 3.23 – Выделение контура на рисунке 3.20 без предварительной обработки изображения

Несмотря на наличие шума, алгоритм справляется с обнаружением объектов, но не без предварительной обработки изображения. Как можно заметить, на рисунке 3.18 алгоритм распознал не только необходимые объекты, но и несколько лишних. Это происходит, потому что шум влияет на контур букв, делая его менее похожим на исходный. Шум размывает границы объекта, тем самым меняя его форму, поэтому алгоритм находит ложные объекты. Качество работы данного метода зависит от дисперсии шума и фильтрации изображения.

ЗАКЛЮЧЕНИЕ

В выпускной квалификационной работе бакалавра рассмотрена задача разработки и анализа алгоритмов обнаружения объектов на изображениях. Такие методы используются в задачах распознавания символов в тексте, определения состояния объектов на видеоизображениях, обнаружения объектов на снимках, полученных с помощью оптических, радиолокационных, инфракрасных датчиков, обработки рентгеновских, снимков магнитно-резонансной томографии.

При выборе тех или иных методов обнаружения объектов обращают внимание на разные показатели качества. С одной стороны, необходимо, чтобы методы работали хорошо при наличии помех, чтобы методы были инвариантны к возможным поворотам и сдвигам искомым изображений, а с другой стороны, такие свойства методов приводят к усложнению алгоритмов, в результате чего, они работают очень долго. Зачастую долгое время работы таких алгоритмов уменьшает их актуальность. Существуют различные критерии, позволяющие производить выбор тех или иных алгоритмов. Эти критерии характеризуют как качественные показатели, так и эксплуатационные. Именно с этой точки зрения рассматривались различные алгоритмы обнаружения объектов на изображении в работе.

В работе основное внимание уделено методам контурного анализа, то есть методам обработки изображений, которые используются для определения формы объекта. Они основаны на анализе контура, то есть линии, которая определяет границы объекта на изображении. Контуром называется граница, характеризующаяся перепадом значений яркости между внутренней и внешней частями объекта. Для описания также могут использоваться различные подходы, например, цепные коды, координаты, кривизна и прочие. Причиной выбора методов именно контурного анализа является то, что они имеют преимущества в сравнении с другими методами, благодаря небольшим затратам ресурсов и способности обрабатывать повернутые и масштабированные объекты. При описании объекта на изображении с

помощью контура получается инвариантность описания этого самого контура при повороте объекта или изменении масштаба. Таким образом, разработка алгоритма была разделена на два этапа: выделение контура и обнаружение объектов. Для выделения границ объектов применяются различные математические алгоритмы, которые сводятся к фильтрации.

Выделение контура может производиться с помощью таких методов, как Собела, Кэнни, Робертса, Превитта. В их основе лежит обработка с помощью масок, которые реализуют выделение градиента или лапласиана. Использовались стандартные алгоритмы, которые уже реализованы в Matlab. Однако, при применении этих операторов к сложным формам, возникают разрывы в контуре, то есть граница формы оказывается разомкнутой. Причиной этому могут послужить дефекты на изображениях или на носителе, помехи при передаче изображении по каналу связи и так далее. Все это приводит к тому, что выделить четко контур не удастся, он имеет разрывы.

Для устранения этой проблемы был реализован метод выделения замкнутого контура, позволяющий сохранить как можно больше информации о фигуре, что непосредственно влияет на дальнейшую точность обнаружения объектов. Алгоритм выделения замкнутого контура использует такие приемы, как эрозия и дилатация. Дилатация расширяет шкалу полутонов, используя элемент структурирования, то есть в двоичном изображении пикселю присваивается значение 1, если какой-либо из соседних пикселей имеет значение 1. Эрозия разрушает шкалу полутонов, используя элемент структурирования, то есть в двоичном изображении пикселю присваивается значение 0, если какой-либо из соседних пикселей имеет значение 0. То есть происходит расширение границ, которое замыкает разрывы, а затем сужение, что возвращает границы в первоначальный вид, устраняя пропуски в контуре. В результате получается гладкий контур без каких-либо дефектов. Данный метод может работать и при наличии шума на изображении, что увеличивает возможные области его применения.

В качестве алгоритма обнаружения объекта был реализован метод, основанный на обобщенном преобразовании Хафа. Само преобразование было предложено Хафом еще в 1962 году, однако работало оно только для аналитических кривых. Позднее Баллард предложил обобщенный вариант для сложных форм. Алгоритм заключается в том, что происходит вычисление параметров для каждой точки контура. Анализируя границы на самом изображении, вносятся изменения в массив накопителей, где максимум указывает на соответствие искомой формы. Исследование этого алгоритма показали, что он хорошо работает при распознавании символов и букв на печатном и рукописном тексте. При этом текст может иметь искажения в виде шума или размытия, с которыми метод справляется.

Однако, результат будет более точным, если совершается предварительная обработка изображения, то есть увеличивается контрастность между объектом и фоном, увеличивается четкость границ объектов. В качестве такой обработки может выступать фильтрация шумов, повышение резкости изображения и так далее. Эффективность данного алгоритма зависит от зашумленности и от точности выделения контура как на эталонном изображении, так и на исследуемом.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Ballard D. H. Generalized the Hough transform to detect arbitrary shapes // Pattern Recognition, Volume 13. - 1980. - №2. - С. 111-122.
2. Гонсалес Р., Вудс Р. Цифровая обработка изображений. - 3-е изд. - М.: Техносфера, 2012. - 1105 с.
3. Дьяконов В.П. MATLAB. Полный самоучитель. - 1-е изд. - М.: ДМК Пресс, 2012. - 770 с.
4. Сойфер В. А. Компьютерная обработка изображений, Часть 2. Методы и алгоритмы // Соросовский образовательный журнал. - 1996. - №3. - С. 110-121.
5. Болгов А.Н. Применение преобразования Хафа и его модификаций для нахождения графических примитивов // Актуальные проблемы авиации и космонавтики. - 2019. - №2. - С. 352.
6. Способ замыкания контуров объектов на матрице полутонового растрового изображения // Патентный поиск URL: <https://findpatent.ru/patent/245/2450355.html> (дата обращения: 29.01.2023).
7. Computer-aided diagnosis // Wikipedia URL: https://en.wikipedia.org/wiki/Computer-aided_diagnosis (дата обращения: 02.02.2023).
8. О задаче поиска объекта на изображении. Часть 1: Базовые методы // ДОМ-СТРАНИЦА Евгения Сергеевича Борисова URL: <http://mechanoid.su/cv-image-detector.html?ysclid=lfmlptzfhx729768593> (дата обращения: 25.02.2023).
9. Обнаружение и локализация объектов на радиолокационных изображениях // Air-Navigation URL: <https://air-navigation.org/video/xvwbzs> (дата обращения: 17.03.2023).

ПРИЛОЖЕНИЕ А

Разработанные алгоритмы

Программа 1:

```
clear all; close all;
A = imread('list1.png');
A = im2double(rgb2gray(A));
figure(20); imshow(A,[]); pause;
%% A = A + 0.1*randn(size(A));
A = A/max(A(:));
%% A = adapthisteq(A);
figure(21); imshow(A,[]); pause;
A = adapthisteq(A);
A = 1 - A;
A = imgaussfilt(A, 1.2);
[Gx,Gy]= imgradientxy(A, 'sobel');
%% figure(1); imshowpair(Gx, Gy, 'montage'); pause;
G = sqrt(Gx.^2 + Gy.^2);
%% figure(2); imshow(G); pause;
G = 1 - G;
%% figure(2); imshow(G);
[mA nA] = size(G);
A_sr = sum(G(:))/(mA*nA);
%G(G>0.2)=1;
BW = edge(G,'sobel');
G = 1 - G;
se = strel('disk', 11);
I = imdilate(G,se);
figure(3); imshow(I);
I = imerode(I, se);
figure(4); imshow(I); pause;
[counts,x] = imhist(I,16);
stem(x,counts);
T = otsuthresh(counts);
I = imbinarize(G,T);
I = filter2(fspecial('average',3),I);
I = medfilt2(I, [3 3]);
I(I<0.7)=0;
T = adaptthresh(I, 0.1);
I = imbinarize(I,T);
figure(14); imshow(I); pause;
I = imclose(I, se);
% I(I<0.7)=0;
se = strel('disk', 12);
I = imdilate(I,se);
se = strel('disk', 10);
I = imerode(I, se);
se = strel('disk', 6);
I = imdilate(I,se);
se = strel('disk', 4);
I = imerode(I, se);
I = imfill(I, 'holes');
I(I>0.5)=1;
figure(5); imshow(I); pause;
I = medfilt2(I,[23 23]);
figure(15); imshow(I); pause;
L = zeros(mA+2, nA+2);
L(2:mA+1, 2:nA+1)=I;
BW = edge(L,'sobel');
figure(6); imshow(BW);
```



```

row = 185; col = 547;
BOUND = bwtraceboundary(BW,[row, col],'N');
B      = complex(BOUND(:,1),BOUND(:,2));
B      = B - mean(B);
nB     = length(BOUND(:,1));
figure(7); plot(BOUND(:,1),BOUND(:,2),'k'); grid on;
%% figure(8); plot(1:nB,BOUND(:,1),'b',1:nB,BOUND(:,2),'r'); grid on;
nfb    = length(B);
FBND   = abs(fft([B' zeros(1,nfb)]));
FBND   = abs(fft([B']));
nit    = length(FBND);
nit4   = round(nit/4);
FB0    = FBND;
% FB0 = FBND(2:end-1);
FB0    = FB0/max(FB0);
%% figure(9); stem(FB0(1:100)); grid on;

```

Программа 2:

```

clear all; close all;
img = imread('text.jpg');
img = im2gray(img);
temp = imread('pp.jpg');
temp = im2gray(temp);
I = img;
canny_I = edge(img,'sobel',0.024);
canny_temp = edge(temp,'canny');
% canny_temp = edge(temp,'canny', 0.75);
dy_I=imfilter(double(img),[1; -1],'same');
dx_I=imfilter(double(img),[1 -1],'same');
theta_I = atan2(dy_I,dx_I)*90/pi();
dy_temp=imfilter(double(temp),[1; -1],'same');
dx_temp=imfilter(double(temp),[1 -1],'same');
theta_temp = atan2(dy_temp,dx_temp)*90/pi();
refPointy = round(size(canny_temp,1)/2);
refPointx = round(size(canny_temp,2)/2);
%Generate the R-table
[y,x] = find(canny_temp > 0);
pointsRange = round(size(x,1));
angleRange = 50;
Rtable = zeros(angleRange, pointsRange,360,20,2);
binCount = zeros(181,1);
%Write to Rtable
for i = 1:pointsRange
    fi = theta_temp(y(i),x(i)) + 90;
    fi = round(fi/180*(angleRange-1) )+1;
    binCount(fi) = binCount(fi) + 1;
    h = binCount(fi);
    for scale = 5:20
        delta_x = scale/10*(refPointx - x(i));
        delta_y = scale/10*(refPointy - y(i));
        for angle = 1:360
            Rtable(fi, h,angle,scale, 1) = round(delta_x*cosd(angle) -
            delta_y*sind(angle));
            Rtable(fi, h,angle,scale, 2) = round(delta_x*sind(angle) +
            delta_y*cosd(angle));
        end
    end
end
%Accumulator:
[y_I,x_I] = find(canny_I > 0);

```

```

pointRange_I = size(x_I,1);
[M,N] = size(canny_I);
count = zeros(M,N);
maxeach = zeros(360,20,1);
for scale = 5:20
    for angle = 1:360
        for i=1:pointRange_I
            fi = theta_I(y_I(i),x_I(i)) + angle;
            fi = asind(sind(fi)) + 90;
            fi = round(fi/180*(angleRange-1) )+1;
            for j = 1:binCount(fi)
                n_x = x_I(i) + Rtable(fi, j,angle,scale, 1);
                n_y = y_I(i) + Rtable(fi, j,angle,scale, 2);
                if (n_y>=1) && (n_y<=M) && (n_x>=1) && (n_x<=N)
                    count(n_y, n_x) = count(n_y, n_x)+1;
                end
            end
        end
        maxpoint = max(count(:));
        maxeach(angle,scale) = maxpoint;
        count = zeros(M,N);
    end
end
bestvoting = max(maxeach(:));
[bestangle, bestscale] = find(maxeach >= bestvoting);
for i=1:pointRange_I
    fi = theta_I(y_I(i),x_I(i)) + bestangle;
    fi = asind(sind(fi)) + 90;
    fi = round(fi/180*(angleRange-1) )+1;
    for j = 1:binCount(fi)
        n_x = x_I(i) + Rtable(fi, j,bestangle,bestscale, 1);
        n_y = y_I(i) + Rtable(fi, j,bestangle,bestscale, 2);
        if (n_y>=1) && (n_y<=M) && (n_x>=1) && (n_x<=N)
            count(n_y, n_x) = count(n_y, n_x)+1;
        end
    end
end
count = mat2gray(count);
figure;
imshow(count);
maxpoint = max(count(:));
xx = [];
yy = [];
for i=1:M
    for j=1:N
        if (count(i,j) > 0.2 * maxpoint)
            xx = [xx i];
            yy = [yy j];
        end
    end
end
figure;
I = uint8(I);
imshow(I);
hold on;
sita=0:0.05:2*pi;
r=30;
a = size(xx);
plot(yy(1), xx(1), 'r*', 'LineWidth', 2, 'MarkerSize', 3);
plot(yy(1)+r*cos(sita),xx(1)+r*sin(sita), 'r', 'LineWidth', 3);
for i=1:a(2)

```

```

if ((i>1) && ((yy(i-1)+10<yy(i)) || (yy(i-1)-10>yy(i))) && ...
    ((xx(i-1)+5<xx(i)) || (xx(i-1)-5>xx(i))))
    plot(yy(i), xx(i), 'r*', 'LineWidth', 2, 'MarkerSize', 3);
    plot(yy(i)+r*cos(sita),xx(i)+r*sin(sita),'r','LineWidth', 3);
end
end
hold off;

```