МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

На правах рукописи

ЖАРКИХ АНДРЕЙ АНАТОЛЬЕВИЧ

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ФОРМИРОВАНИЯ И КОНТРОЛЯ ЛАТЕНТНЫХ ИЗОБРАЖЕНИЙ

05.13.18 – Математическое моделирование,
численные методы и комплексы программ

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель:
доктор физико-математических наук,
профессор
Шагрова Г. В.

Ставрополь – 2017

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 5
ГЛАВА 1 ОБЗОР СОСТОЯНИЯ ПРОБЛЕМЫ И ПОСТАНОВКА ЗАДАЧИ ИССЛЕДОВАНИЯ 14
1.1 Краткая характеристика латентных изображений 14
1.2 Особенности человеческого восприятия изображений 17
1.3 Математические модели, методы формирования и контроля цифровых изображений, содержащих скрытую информацию 20
1.3.1 Пространственные методы формирования цифровых изображений, содержащих скрытую информацию 21
1.3.2 Частотные методы формирования цифровых изображений, содержащих скрытую информацию 26
1.4 Методы внедрения скрытых изображений в полиграфическую продукцию 29
1.4.1 Методы, основанные на использовании различно-ориентированных растровых структур 30
1.4.2 Методы, основанные на использовании различных линиатур растра 31
1.4.3 Методы, основанные на использовании различных амплитудно-модулированных растровых структур 31
1.4.4 Методы, основанные на фазовом сдвиге растровых структур 35
1.4.5 Физико-химические методы внедрения и контроля скрытой информации36
1.4.6 Программное и аппаратное обеспечение для внедрения скрытой информации в полиграфические изделия и их контроля 40
1.5 Постановка задач исследования 42
1.6 Выводы 44
ГЛАВА 2 РАЗРАБОТКА МАТЕМАТИЧЕСКИХ МЕТОДОВ И АЛГОРИТМОВ ФОРМИРОВАНИЯ ЦИФРОВЫХ ЛАТЕНТНЫХ ИЗОБРАЖЕНИЙ 47
2.1 Метод моделирования латентного изображения 47
2.1.1 Метод моделирования латентного изображения с внедрением цветного изображения 48

2.1.2 Метод моделирования латентного изображения, с внедрением черно-белого изображения 51
2.1.3 Зависимость качества формируемого изображения от выбранного способа растрирования 53
2.2 Математические модели формирования латентных изображений 56
2.2.1 Математическая модель формирования латентного изображения, с внедрением цветного изображения 56
2.2.2 Математическая модель формирования латентного изображения, с внедрением черно-белого изображения 59
2.3 Проверка разработанных моделей формирования латентных изображений 61
2.3.1 Моделирование латентного изображения известным методом 62
2.3.2 Моделирование латентного изображения предложенными методами 65
2.3.3 Проведение испытаний и анализ результатов 66
2.4 Выводы 70
ГЛАВА 3 РАЗРАБОТКА МАТЕМАТИЧЕСКИХ МЕТОДОВ И АЛГОРИТМОВ КОНТРОЛЯ ЦИФРОВЫХ ЛАТЕНТНЫХ ИЗОБРАЖЕНИЙ 71
3.1 Выбор оптимальной вейвлет-функции для выявления скрытого изображения73
3.2 Использование вейвлетов высоких уровней пакетной декомпозиции 81
3.3 Выявление скрытых изображений с помощью вейвлет-преобразований .. 88
3.3.1 Проблема визуализации при совпадающем периоде текстур сюжетного и скрытого изображений 88
3.3.2 Проблема контроля латентного изображения низкого качества 90
3.4 Математическая модель и численный метод контроля латентного изображения 92
3.5 Система формирования и контроля латентных изображений 98
3.6 Выводы 101
ГЛАВА 4 РАЗРАБОТКА ПРОГРАММНЫХ СРЕДСТВ ФОРМИРОВАНИЯ И КОНТРОЛЯ ЛАТЕНТНЫХ ИЗОБРАЖЕНИЙ 103
4.1 Разработка программного модуля формирования латентного изображения103
4.2 Разработка программного комплекса контроля латентных изображений 105

4.2.1 Программный модуль «Конвейерный захват латентных изображений»108
4.2.2 Программный модуль «Построение комбинированных фильтров для выявления латентных изображений» ... 111
4.2.3 Программный модуль «Построение и хеширование контрольного скрытого объекта в латентном изображении» ... 112
4.2.4 Программные модули «Автоматическое определение объекта по распознанному латентному изображению» и «Принятие решения» ... 114
4.3 Тестирование разработанных методов и алгоритмов контроля латентных изображений ... 120
4.4 Выводы ... 130
ЗАКЛЮЧЕНИЕ ... 131
СПИСОК ЛИТЕРАТУРЫ ... 133
ПРИЛОЖЕНИЕ 1. ЛИСТИНГ ПРОГРАММНОГО МОДУЛЯ ФОРМИРОВАНИЯ ЛАТЕНТНОГО ИЗОБРАЖЕНИЯ ... 143
ПРИЛОЖЕНИЕ 2. ЛИСТИНГ ОСНОВНЫХ ФОРМ ПРОГРАММНОГО КОМПЛЕСА КОНТРОЛЯ ЛАТЕНТНОГО ИЗОБРАЖЕНИЯ ... 147
ПРИЛОЖЕНИЕ 3. СВИДЕТЕЛЬСТВА О РЕГИСТРАЦИИ ЭЛЕКТРОННЫХ РЕСУРСОВ ... 166

ВВЕДЕНИЕ

Актуальность проблемы и направление исследований. Основной объем информации, получаемый человеком, воспринимается визуально. Согласно статистическим исследованиям, 83 процента информации воспринимается визуальным образом [1]. Данный факт находит отражение в различных сферах жизнедеятельности.

Производители различных товаров, в основной своей массе, стремятся персонализировать свою продукцию, выпуская красочные упаковки, направленные на определенные психофизические особенности целевого потребителя. При этом производитель стремится создать уникальную упаковку, позиционирующую товар как уникальный.

Нечистые на руку производители, предпочитающие «паразитировать» на уже созданном бренде, или вовсе занимающиеся выпуском контрафактной продукции в первую очередь копируют дизайн упаковки.

Международная торговая палата оценивает объем продаж контрафактной продукции в 390,8 миллиарда евро. Согласно данным таможенных органов, этот показатель еще выше и составляет около 781 миллиарда евро, что соответствует почти 10% объема мировой торговли. Согласно расчетам Института предпринимательства, из-за пиратства и подделки Германия лишилась 70 000 рабочих мест; по всему миру этот показатель составил около 200 000, причем значительная доля приходится на страны ОЭСР. Вследствие растущей глобализации объем подделок за последние годы резко увеличился. Только немецкие таможенные органы в 2006 году конфисковали контрафактной продукции на сумму до 1,2 миллиарда евро, что в пять раз превышает соответствующий показатель, зафиксированный годом ранее. По оценке Немецкой федерации машиностроения, только в этой отрасли ущерб от пиратства и подделки составляет 4,5 миллиарда евро в год. Зачастую, даже экспертам сложно определить факт подделки без требовательных к лабораторному

оборудованию и времени методов экспертизы. Рядовому гражданину отличить подделку, иногда, попросту невозможно [2].

Любое цифровое изображение, созданное будь то с помощью фотоаппарата или в любом графическом редакторе, защищается авторским правом на основании статьи 1265 Гражданского кодекса РФ [3]. Широкий доступ к сети Интернет послужил мощным толчком к неконтролируемому незаконному распространению объектов авторского права, включая цифровые изображения. Определить истинного владельца контента зачастую просто невозможно.

Представление полиграфической и цифровой графической продукции в виде латентных изображений позволяет обеспечить их защиту без нарушения визуальной композиции изображения.

Существующие методы формирования латентных изображений доступны только для квалифицированных специалистов в данной области, поэтому, несмотря на высокую эффективность защиты, латентные изображения используются только для защиты особо ценной полиграфической продукции, такой как банкноты, ценные бумаги и контрольные документы. Поэтому, разработка простых и эффективных методов внедрения скрытых изображений является востребованной на сегодняшний день.

Согласно данных Центробанка за 2014 год количество поддельных российских банкнот составило 0,0012% от общей денежной массы при этом, число ложных отказов автоматических устройств контроля валидных банкнот на сегодняшний день составляет 6% [4].

На текущий момент существует 6 основных методов автоматического контроля полиграфической продукции: контроль габаритов изображения, контроль видимого изображения, контроль инфракрасного изображения, контроль ультрафиолетовых меток, контроль магнитных меток, контроль встроенных электронных носителей.

Латентные изображения, несмотря на то, что, в соответствии с действующим ГОСТом Р 54109-2010 [5], подпадают под требования к машиносчитываемым элементам, на данный момент относятся к

человекочитаемым признакам и не подлежат автоматическому контролю [6]. Разработка методов автоматического контроля латентных изображений позволит увеличить надежность контроля, а также сократить число ложных срабатываний автоматических устройств контроля.

Объект диссертационного исследования – латентные изображения – изображения со скрытой информацией.

Цель диссертационной работы – повышение визуального качества формируемых латентных изображений и сокращение вычислительной сложности контроля скрытой информации, за счет совершенствования методов внедрения и выявления скрытых изображений, для обеспечения возможности автоматизации процесса контроля латентных изображений.

Предмет диссертационного исследования – методы внедрения и выявления скрытой информации, математические модели формирования и контроля латентных изображений, программно-аппаратные комплексы, реализующие эти методы и модели.

Научная задача – разработка методов моделирования и алгоритмов формирования латентных изображений, учитывающих свойства системы человеческого зрения, а также метода автоматического контроля латентного изображения, независящего от способов внедрения скрытой информации, позволяющих улучшить визуальное качество латентного изображения и сократить вычислительную сложность выявления скрытой информации.

Реализация поставленной задачи может быть декомпозирована на следующие **частные задачи**:

− разработка метода моделирования и построение математической модели формирования латентных изображений, учитывающих свойства системы человеческого зрения;

− разработка численного метода контроля латентных изображений на основе новой математической модели выявления скрытых изображений, позволяющей обнаружить скрываемую информацию при неизвестном методе ее внедрения;

− разработка системы формирования и контроля латентных изображений, реализующих предложенные методы;

− разработка алгоритмов формирования и программного модуля для формирования латентного изображения;

− разработка алгоритмов выявления скрытых изображений и программного комплекса для контроля латентных изображений.

Методы исследования. Для решения поставленных в работе научных задач использованы методы математического моделирования и вычислительного эксперимента, вейвлет-анализа, фильтрации изображений, выделения объектов, обработки и сегментирования изображений.

Научная новизна. Разработаны:

1. Метод моделирования латентных изображений и модели формирования латентных изображений с внедрением, как черно-белых, так и цветных скрываемых изображений, отличающиеся от известных, комбинацией частотных и пространственных преобразований исходного изображения, что позволяет повысить визуальное качество латентных изображений на 8,5%;

2. Численный метод контроля латентных изображений, позволяющий сократить число итераций в 6 раз по сравнению с известным методом, на основе математической модели выявления скрытого изображения, вне зависимости от конкретного метода ее внедрения, с помощью вейвлет-преобразования латентного изображения;

3. Система формирования и контроля латентных изображений, на основе предложенных методов, позволяющая, представить процессы от момента формирования до определения подлинности защищенного документа, отличающаяся от известных тем, что предусмотрен контроль латентных изображений, как при известном, так и при неизвестном методах внедрения скрываемых изображений;

4. Алгоритмы и программный модуль для формирования латентного изображения, позволяющие автоматически формировать латентное изображение на основе заданных исходного и скрываемого изображений, отличающиеся от

известных меньшим отклонением меры пикового отношения сигнал/шум от исходного в среднем на 8,5%;

5. Алгоритмы выявления и программный комплекс для контроля латентных изображений, отличающийся от известных возможностью автоматически определять подлинность и вид латентного изображения, полученного с камеры, сетевого или локального хранилищ, что позволило сократить в 6 раз максимальное время анализа латентных изображений.

Практическая значимость результатов работы заключается в том, что разработанные программные средства позволяют:

– формировать латентные изображения с внедрением как черно-белых, так и цветных скрываемых изображений, для защиты объектов интеллектуальной собственности, представленных цифровыми изображениями. От пользователя требуется только определить исходное изображение и внедряемое;

– автоматически определять подлинность и вид латентного изображения, полученного с камеры, сетевого или локального хранилищ в режиме реального времени, в частности при интеграции их с системой технического зрения робототехнического комплекса.

Положения, выносимые на защиту:

1. Метод моделирования латентных изображений и построение математической модели формирования изображений, содержащих скрытое изображение;

2. Численный метод контроля латентных изображений на основе вейвлет-преобразования латентного изображения и математическая модель выявления латентных изображений;

3. Система формирования и контроля латентных изображений, реализующая предложенные методы;

4. Алгоритмы формирования и программный модуль для формирования латентного изображения;

5. Алгоритмы выявления и программный комплекс для контроля латентных изображений.

Достоверность и обоснованность полученных результатов обеспечивается применением современной технологии математического моделирования, корректностью математических постановок задач, и результатами натурных экспериментов.

Авторский вклад в разработку. Основные результаты и выводы диссертационной работы получены лично автором. Авторским вкладом является разработка метода формирования латентных изображений с внедрением как черно-белых, так и цветных скрываемых изображений, разработка метода выявления латентных изображений основанного на вейвлет-преобразовании латентного изображения, проведение экспериментов по формированию и выявлению скрытых изображений, разработка алгоритмов формирования и контроля латентных изображений и разработка программных средств на их основе.

Апробация работы. Основные результаты диссертационной работы докладывались и обсуждались на следующих научных конференциях и семинарах: Научно-техническая конференция по итогам работы профессорско-преподавательского состава СевКавГТУ за 2010 год. (Ставрополь, СевКавГТУ, 2011 г.), Международная конференция «Инфокоммуникационные технологии в науке, производстве и образовании (Инфоком-5)» (Ставрополь, 2012 г.), Международная молодежная конференция «Прикладная математика, управление и информатика» (Белгород, 3-5 октября 2012г.), Региональная научно-практическая конференция «Инновационные идеи молодежи Северного Кавказа – развитию экономики России» (23-24 октября 2012г.), шестая международная научно-техническая конференция «Инфокоммуникационные технологии в науке, производстве и образовании (Инфоком-6)» (Ставрополь, 21-27 апреля 2014г.), IV Международная научно-практическая конференция (Алушта, 2015г.), I Всероссийская научно-техническая конференция молодых ученых, аспирантов и студентов (Ростов-на-Дону, 2015г.), V Международная научно-практическая

конференция (Ставрополь, 2016г.), V Всероссийской научно-технической конференции «Студенческая наука для развития информационного общества» (Ставрополь, 2016г.).

Публикации. По теме диссертации опубликовано 19 работ в журналах и трудах конференций, из них 3 работы в изданиях, рекомендованных ВАК РФ для опубликования научных положений диссертационных работ, 6 работ в трудах международных конференций, получено 6 свидетельств о государственной регистрации программ и программных модулей.

Структура и объем диссертации. Диссертация состоит из введения, четырех глав, заключения, списка литературы из 97 наименований и четырех приложений. Общий объем диссертации 172 страницы.

Во введении обоснована актуальность темы диссертации, определена научная новизна и практическая значимость работы, сформулированы цель и задачи исследования, приведена краткая характеристика полученных результатов и представлены положения, выносимые на защиту.

В первой главе представлен обзор научно-технической и патентной литературы, проведен анализ существующих моделей, методов моделирования и выявления скрытых изображений и программно-аппаратных комплексов для их реализации. Рассмотрены математические методы обработки изображений, обоснована актуальность и сформулированы основные задачи исследования.

Во второй главе разработаны метод моделирования и математическая модель формирования ЛИ, с внедрением цветного изображения, на основе модификации известного метода [37], заключающейся в использовании как частотных, так и пространственных преобразований исходного изображения. Разработаны метод моделирования и математическая модель формирования ЛИ, с внедрением черно-белого изображения, отличающиеся от известного метода [21] изменением растрирования изображения с внедрением скрываемого изображения во все цветовые компоненты без потери визуального качества.

В третьей главе проведен анализ результатов выявления скрытой информации, с помощью различных базисных вейвлетов для построения

математической модели на основе вейвлет-преобразований латентных изображений и разработки численного метода контроля этих изображений. Определены критерии качества вейвлет-функции для решения поставленных задач, обеспечивающие различимость в визуализированных коэффициентах одних элементов при неразличимости других элементов изображения. В результате проведенных вычислительных экспериментов установлено, что биортогональный вейвлет по базису 2.8, по сравнению с другими вейвлетами, имеет более высокую локализацию, как по частоте, так и по времени. Данный вейвлет лучше всего подходит для решения поставленных задач, так как позволяет сегментировать объекты на группы с примерно равным числом элементов и минимальным числом элементов, одновременно входящих в разные группы. Пакетное вейвлет-преобразование позволяет добиться более высокой частотной локализации и сегментировать компоненты латентного изображения на большее число групп, однако для исследования изображений на более высоких уровнях декомпозиции требуется высокое разрешение изображения. Для решения проблем при анализе изображения, связанных с совпадающим периодом текстур фонового и скрытого изображений и низкого качества исследуемого образца, применяется поворот изображения.

Разработаны математическая модель выявления скрытой информации в латентном изображении и численный метод контроля латентных изображений, который не зависит от конкретного способа формирования латентных изображений, сформированных различно ориентированными структурами или образованных структурами с совпадающим периодом фонового и скрытого изображения. Предложенная модель и метод позволяют установить факт наличия скрытого изображения в исследуемом образце, а также в автоматическом режиме установить тип контролируемого латентного изображения.

На основе разработанных методов предложена система формирования и контроля латентных изображений, позволяющая описать соответствующие процессы, для построения алгоритмов и создания программных комплексов.

В четвертой главе разработаны алгоритмы и программный модуль для формирования латентного изображения, позволяющие автоматически формировать латентное изображение на основе заданных исходного и скрываемого изображений, отличающиеся от известных меньшим отклонением меры пикового отношения сигнал/шум от исходного в среднем на 8,5%.

Разработаны алгоритмы выявления скрытой информации на основе вейвлет-анализа изображений, которые позволили сократить в 6 раз максимальное время анализа и программный комплекс для контроля латентных изображений. В программном комплексе реализована возможность оперативного контроля латентных изображений с использованием вейвлет-анализа изображений и базы комбинированных фильтров, представляющих собой оптимальные последовательности фильтров, зависящие от способа формирования латентного изображения, что обеспечивает высокое визуальное качество выявленного скрытого изображения.

Разработанный программный комплекс, в отличие от известных, решает задачу автоматического распознавания латентных изображений, что позволяет использовать такие изображения как машиночитаемый элемент защиты печатной продукции.

В заключении диссертационной работы сформулированы основные теоретические выводы исследований, а также описаны полученные практические результаты.

Библиографический список содержит 97 наименований источников литературы.

В приложения вынесен листинг разработанных программных модулей комплекса, копии актов о внедрении результатов диссертационного исследования, а также копии свидетельств о регистрации алгоритмов и программ.

ГЛАВА 1 ОБЗОР СОСТОЯНИЯ ПРОБЛЕМЫ И ПОСТАНОВКА ЗАДАЧИ ИССЛЕДОВАНИЯ

1.1 Краткая характеристика латентных изображений

Совокупность методов и средств для встраивания информации принято относить к области стеганографии. Набор терминов, используемых в западной литературе по стеганографии, принятый по итогам первого открытого симпозиума в Кембридже, посвященного проблематике сокрытия данных [7] описывает базовый набор терминов стеганографии.

В стеганографии выделяется три основных направления: методы встраивания сообщений в контейнеры, методы извлечения сообщений из контейнера и стегоанализ, представляющий совокупность методов обнаружения, выявления и контроля сообщений в различных контейнерах любой природы [8 – 15].

Появление новых видов передачи и обработки информации привело к становлению нового направления в сфере защиты информации: цифровой стеганографии, одним из методов встраивания сообщений которой являются цифровые водяные знаки (ЦВЗ). Под ЦВЗ понимают скрываемую информацию, которая может встраиваться в аудио-, видео-, графический контейнер [16 – 17].

В работах [18 – 20] вводится понятие текстурированного водяного знака (ТВЗ) – специальной графической информации, скрытой в вариациях каких-либо параметров двумерной квазипериодической текстуры, покрывающей документ или его отдельные части.

В работах авторов [21 – 23], когда сообщение и контейнер представлены двумя изображениями, используются понятия «латентное изображение» (latent image), представляющее из себя заполненный контейнер, при этом под скрытым изображением (hidden image) понимается внедренное сообщение.

Так как под ЦВЗ понимается встраивание не только графической информации, и данный термин является более общим, а в случае с ТВЗ контейнером выступают только текстурированные изображения, в диссертационной работе, для обозначения изображений содержащих скрытую информацию в виде другого изображения используется термин «*латентное изображение*».

К латентным изображениям относятся изображения, основным свойством которых является изменение видимости элементов изображения при изменении условий наблюдения или способа регистрации основного изображения.

В настоящее время латентные изображения широко применяются в различных сферах для защиты документов от копирования. Вместе с тем не существует универсального способа выявления латентных изображений, в связи, с чем затруднен и, как следствие, является дорогостоящим процесс отличия копии от подделки.

Для выявления латентного изображения используются оптические методы неразрушающего контроля, основанные на способности скрытого изображения, становится видимым в косопадающем свете, ультрафиолетовых, инфракрасных, лазерных лучах, а также методы, основанные на цифровой фильтрации [24 – 26].

Методики формирования латентных изображений объединяют три отличительные особенности. Во-первых, скрытые изображения невидимые, во-вторых они неотделимы от латентного изображения, в которое встроены, потому они не могут быть удалены, когда латентное изображение преобразуют или демонстрируют. В-третьих, скрытые изображения подвергаются тем же преобразованиям, что и латентное изображение, что дает возможность узнать о преобразованиях, которые совершались [27]. Таким образом, латентные изображения могут применяться для решения следующих задач:

– *Контроль вещания.* При распределении и трансляции мультимедийная информация может быть подменена, незаконно скопирована, передающие станции могут использовать не все оплаченное заказчиками время и пр. На основе

внедренных скрытых изображений можно определить, где и когда транслировался оригинал. Достоинствами применения латентных изображений для контроля вещания по сравнению с другими методиками являются совместимость, как с аналоговым, так и цифровым вещанием, а также относительная простота.

– *Идентификация владельца.* Закон об авторском праве устанавливает определенные права законного владельца оригинала. Оригинал отмечается уведомлением об авторских правах. Чтобы не ухудшить качество изображения, текст уведомления размещают обычно в каком-нибудь углу, что дает вырезать без большого ущерба для качества. Благодаря невидимости и неотделимости скрытого изображения от латентного изображения имеется возможность идентификации владельца даже при кадрировании [27].

– *Подтверждение права собственности.* Данная задача направлена на предотвращение посягательства на авторство изображения субъектом не являющимся его законным владельцем. Подмена уведомления об авторских правах в латентных изображениях, даже если известна методика его формирования, практически невозможна без потери качества изображения.

– *Отслеживание копий.* Данная задача предполагает возможность в случае установления факта создания нелицензионных копий отследить кто является их распространителем. Внедрение латентных изображений содержащих уникальное для каждой копии скрытое изображение позволяет без потери визуального качества и нарушения художественной композиции решенить данную задачу.

– *Аутентификация содержания.* Для решения задач определения целостности передаваемой информации обычно используется цифровая подпись[28]. Аналогом цифровой подписи могут служить латентные изображения. В данном случае имеется возможность определить какая именно область изображения была изменена, а не только узнать о факте такого изменения. Также в отличае от ЭЦП внедрение ЛИ позволяет решить задачу аутентификации содержания не только для электронных документов но и печатной продукции.

– *Контроль копирования полиграфической продукции.* Данная задача предполагает невозможность копирования изображения в целом или его частей. Латентные изображения могут служить сигналом для копировального устройства о невозможности копирования. Ограниченность такого решения очивидна, поскольку нет закона, который бы обязывал бы снабжать все устройства воспроизведения необходимыми детекторами латентных изображений [27]. Однако в ряде областей данные методики нашли широкое применение. Примерами таких систем является Созвездие Евриона [29], Advanced Access Content System [30].

Также, при копировании, в виду технических особенностей копировальных устройств, скрытое изображение либо становится видимым на копии, либо не воспроизводится на копии [31]. Согласно ГОСТа [5] при изготовлении защищенной полиграфической продукции всех высших уровней защищенности: "А", "Б", "В" обязательно наличие трех различных графических элементов защиты продукции, под которые попадают, в том числе и латентные изображения.

Таким образом, под латентным изображением можно понимать изображение, содержащее скрытую информацию в виде другого изображения, в отличие от ЦВЗ, под которым понимается встраивание не только графической информации, и ТВЗ, у которого контейнером выступают только текстурированные изображения.

1.2 Особенности человеческого восприятия изображений

Современный подход к формированию латентных изображений предполагает встраивание скрытых изображений в существенные области изображений, изменение которых может привести к значительному визуальному искажению исходного изображения [27]. С целью сохранения визуального образа исходного изображения при внедрении скрытого изображения следует учитывать специфику физиологической и психофизиологической перцепции объектов реального мира.

Общее восприятие изображения складывается из совокупности восприятий пространственных, частотных характеристик, а также пространственной ориентации элементов изображения (горизонтальной, вертикальной и диагональной) [32]. При внедрении скрытого изображения следует учитывать особенности восприятия каждой из этих характеристик по отдельности, а также их сочетаний во избежание преждевременного обнаружения скрытого изображения человеческим глазом и больших видимых деформаций исходного изображения. Выделяют следующие свойства человеческого зрения [33]:

1. Большие изменения яркости областей, обладающих малой и средней яркостью, приводят к заметным искажениям изображения. Наиболее эффективно встраивание в области с высокой яркостью.

2. Низкочастотный шум лучше воспринимается человеком, чем высокочастотный, т.е. человеческий глаз более чувствителен к искажениям в низкочастотной области. Предпочтительно выполнять внедрение в высокочастотные области.

3. При одновременном воздействии на глаз двух компонентов изображения со схожими характеристиками эти компоненты воспринимаются как один компонент.

4. Объекты красного цвета являются наиболее заметными по сравнению с объектами других цветов. Объекты синего цвета – наименее заметны [34].

5. Частотная чувствительность горизонтальных и вертикальных элементов выше, чем диагональных.

6. Периодическая структура, состоящая из черных штрихов и пробелов одинаковой ширины неразличима при угле поля зрения между центрами двух штрихов менее 1'30" [35]. Таким образом, минимальное расстояние D_{min}, с которого периодическая структура становится различимой определяется по формуле (1.1)

$$D_{min} = \frac{T}{2} tg 0.0125, \tag{1.1}$$

где T – расстояние между штрихами.

При просмотре изображения с расстояния 40см и более, что является нормальным расстоянием просмотра латентных изображений, расстояние между штрихами периодической структуры, для обеспечения неразличимости, должно быть не более 0,175мм.

В полиграфии для периодических растровых структур используется понятие линиатуры растра равное $1/T$, измеряемое в линиях на дюйм (lpi). Таким образом, для обеспечения неразличимости полиграфического изображения с расстояния 40см, изображение должно быть растрировано с линиатурой не менее 150lpi.

7. Участки с высоким контрастом, а также области с резким перепадом яркости акцентируют на себя больше внимания.

8. Большие однородные участки привлекают больше внимания, чем малые.

9. Длинные и тонкие объекты более воспринимаемы, чем ограниченные и закругленные.

10. При просмотре изображения в первую очередь взгляд падает на центр и на объекты переднего плана.

11. Изображения людей притягивают больше внимания, чем другие объекты.

На сегодняшний день не существует точного математического описания системы человеческого зрения, поэтому объективная оценка качества латентного изображения, с учетом всех особенностей восприятия, также невозможна [33].

Для определения искажений преобразованного изображения по сравнению с исходным используются следующие подходы [36]:

– визуальная оценка латентного изображения и его зрительное сопоставление с оригиналом. Только данный подход, несмотря на его субъективность, позволяет учитывать все основные свойства человеческого зрения.

– Вычисление меры среднеквадратической ошибки (1.2) или ее модификации – меры пикового отношения сигнал/шум (1.3) для полутонового

латентного изображения или для каждой из цветовых компонент полноцветного изображения.

$$\text{СКО} = \frac{1}{N}\sum_{i=1}^{N}(x_i - x_i')^2, \tag{1.2}$$

где N – число пикселов в изображении, x_i, x_i' - значения пикселов исходного и латентного изображения.

$$\text{ПОСШ} = 10log_2\frac{N255^2}{\sum_{i=1}^{N}(x_i - x_i')^2}, \tag{1.3}$$

где N – число пикселов в изображении, x_i, x_i' - значения пикселов исходного и латентного изображения. Латентное изображение считается визуально качественным, если ПОСШ $\geq$ 28 дБ. Недостатком данного подхода является невозможность учесть визуально видимые локальные искажения при отсутствии или низком уровне преобразований остальной части изображения.

В существующих, в настоящее время, методах встраивания скрытой информации в изображения используются либо одно определенное свойство системы человеческого зрения, либо комбинация первых четырех, из перечисленных выше, что сказывается на качестве формируемых латентных изображений, поэтому необходима разработка таких методов, которые максимально возможно учитывали бы эти свойства.

1.3 Математические модели, методы формирования и контроля цифровых изображений, содержащих скрытую информацию

Формирование и выявление цифровых изображений содержащих скрытую информацию производится цифровыми устройствами, передача осуществляется исключительно по цифровым каналам связи.

Учитывая, что передача данного класса изображений происходит исключительно по цифровым каналам связи, уровень их надежности определяется стойкостью к разрушительным процессам сопутствующим такой передаче, а также невозможностью выделения исходного и скрытого изображения в

первоначальном виде. В качестве основных примеров деструктивных воздействий на цифровое изображение можно назвать сжатие, масштабирование, поворот.

На сегодняшний известно очень много алгоритмов для встраивания данных в изображение, однако лишь ограниченный набор методов позволяет использовать в качестве внедряемой информации другое изображение.

Практически все изученные методы формирования цифровых латентных изображений, за исключением способа [37], предполагают внедрение бинарного скрытого изображения $S(x, y) \epsilon \{0,1\}$ в полутоновое исходное изображение или в один из каналов цветного изображения.

Как известно, человеческий глаз имеет наиболее низкое восприятие к синему цвету. Учитывая данную особенность, а также то, что с математической точки зрения, полноцветное изображение в системе RGB представляет собой совокупность из трех матриц одинакового размера, с одинаковыми диапазонами значений $\{0,1, \dots ,255\}$, встраивание обычно производится в синий канал.

1.3.1 Пространственные методы формирования цифровых изображений, содержащих скрытую информацию

Пространственные методы подразумевают внедрение скрытого изображения за счет преобразования яркостей исходного изображения $l(x, y) \in \{1, \dots, L\}$ или цветовых составляющих красного $r(x, y) \in \{0, \dots ,255\}$, зеленого $g(x, y) \in \{0, \dots ,255\}$, и синего $b(x, y) \in \{0, \dots ,255\}$.

Метод наименее значащего бита [38]. Данный метод не относится к числу надежных, однако является показательным для пространственных методов и применяется в ряде других методов.

Каждый элемент матрицы $I(\mathrm{x}, \mathrm{y})$ представляющей полутоновое изображение можно представить в виде 8-битного значения 0bXXXXXXXX. В таком виде матрицу можно разделить на 8 битовых плоскостей, каждая из которых будет соответствовать одному из разрядов двоичных значений:

$$I(x, y) = b_1(x, y) + b_2(x, y) * 2 + \dots + b_k(x, y) * 2^{k-1}, \quad (1.4)$$

где $b_k(x, y) = \{0,1\}, k = 8$.

Наименее значащим битом является последний 0bAAAAAAAX. Изменение данного бита не влечет за собой серьезных визуальных искажений изображения.

Суть данного метода, при внедрении скрытого изображения сводится к замене первой битовой плоскости $b_1(x, y)$ на скрываемое бинарное изображение:

$$L(x, y) = S(x, y) + b_2(x, y) * 2 + \ldots + b_k(x, y) * 2^{k-1}, \quad (1.5)$$

где $b_k(x, y) = \{0,1\}$, $S(x, y) = \{0,1\}$, $k = 8$.

Выявление скрытого изображения, сформированного с помощью данного метода, сводится к разложению латентного изображения на битовые плоскости и их отсечение со 2 по 7.

Для внедрения скрытого изображения могут быть использованы также иные битовые плоскости, однако замена битовых плоскостей, соответствующих более высоким разрядам, приводит к большим искажениям латентного изображения.

Метод замены палитры [39]. В качестве исходного изображения $I(x, y)$ используется палитровое, представляющееся в виде двух матриц: цветовой карты $map(a, 3)$, каждый столбец которой задает интенсивность красного, зеленого и синего цветов соответственно, и управляющей $X(x, y)$, значения которой представляют номер строки a в матрице $map(a, 3)$ [40].

Обобщенный алгоритм данного метода подразумевает следующее:

1. Цветовая карта $map(a, 3)$ упорядочивается таким образом, чтобы смежные цвета стали подобными. Например, значение цвета может быть упорядочено по расстоянию $d(a)$ в RGB-пространстве, вычисляемое как [41]:

$$d(a) = \sqrt{map(a, 1)^2 + map(a, 2)^2 + map(a, 3)^2} \quad (1.6)$$

2. Производится размытие исходного изображения за счет уменьшения общего количества значений цветов вдвое ($a/2$). Элементы палитры дублируются таким образом, чтобы значение цветов для них различалось несущественно, и каждая из пар могла представить бит бинарного 0 или 1 соответственно. Четные строки упорядоченной цветовой матрицы $map'(a, 3)$ соответствуют нулевым значениям пикселя скрытого изображения, а нечетные – единице, или наоборот.

Выявление скрытого изображения сводится к разложению управляющей матрицы $X(x, y)$ палитрового латентного изображения на битовые плоскости и отсечение всех кроме первой. После чего возможно потребуется инвертировать изображение, если четным строкам упорядоченной цветовой матрицы $map'(a, 3)$ соответствуют единичным значениям насыщенности пикселя скрытого изображения, а нечетные – нулевым.

Данный метод не считается надежным, однако достаточно прост в реализации.

Метод квантования изображения [42]. Данный метод основан на межпиксельной зависимости. Размерность внедряемого изображения, в виду специфики метода, меньше исходного $I(x, y)$ на один пиксель по горизонтали или вертикали, т.е. $S(x-1, y)$ или $S(x, y-1)$ соответственно. Алгоритм формирования подразумевает следующее:

1. Формуется матрица из разностей между соседними пикселями, например, по горизонтали:

$$\varepsilon(k, m) = I(x, y) - I(x+1, y) \tag{1.7}$$

2. Формируется стеганоключ $B(\varepsilon, b)$, представляющий матрицу, с соответствующим для каждого из возможных значений разностей ε матрицы $\varepsilon(k, m) \epsilon \{-255, \dots, 255\}$ псевдослучайного значения $b \epsilon \{0,1\}$.

3. Генерируется латентное изображение $L(x, y)$ путем поэлементного сопоставления матрицы разностей $\varepsilon(k, m)$, стеганоключа $B(\varepsilon, b)$ и внедряемого изображения $S(x-1, y)$. Если значение b, соответствующего элементу $\varepsilon(k, m)$, равно значению элемента $S(x-1, y)$, то значение соответствующего элемента $L(x, y)$ заполняется из исходного. Иначе подбирается ближайшее значение ε, которому соответствует значение b равно значению элемента $S(x-1, y)$ и значение элемента $L(x, y)$ заполняется из $I(x, y)$ измененного таким образом, чтобы выполнялось условие: $I(x, y) - I(x+1, y) = \varepsilon$.

Выявление латентного изображения $L(x,y)$ такого типа осуществляется путем сопоставления стеганоключа $B(\varepsilon,b)$ с матрицей разностей латентного изображения $\varepsilon_{L(x,y)}$

Метод Дармтедтера [43]. Существенным недостатком данного метода при создании латентных изображений является ограничение в выборе скрываемого изображения – оно должно быть, как минимум, в 8 раз меньше исходного. Алгоритм формирования подразумевает:

1. Исходное изображение разбивается на блоки 8*8 пикселей, в каждом из которых формируются две группы пикселей с примерно однородными яркостями l_1 и l_2. При этом средняя яркость первой группы меньше чем у второй $l_1 < l_2$.

2. Каждая из групп пикселей делится на две категории, согласно стеганоключа $B(8,8)$, представленного матрицей из нулей и единиц, где 0 обозначает принадлежность к первой категории (А), 1 – ко второй (В). Таким образом, блок разбивается на 4 группы, где $l_{1A} < l_{2A}$, $l_{1B} < l_{2B}$.

3. Для внедрения пиксела скрытого изображения $S(x,y)$ $\epsilon\{0,1\}$ каждый блок исходного изображения модифицируется таким образом, чтобы соблюдалось следующее условие:

$$S(x,y) = \begin{cases} 1, \begin{cases} l'_{1\text{A}} > l'_{2\text{A}}, \\ l'_{1\text{B}} > l'_{2\text{B}}, \end{cases} \\ 0, \begin{cases} l'_{1\text{A}} < l'_{2\text{A}}, \\ l'_{1\text{B}} < l'_{2\text{B}}. \end{cases} \end{cases} \tag{1.8}$$

При этом яркость каждой категории пикселей для обеспечения одномерности должна меняться одинаково.

Алгоритм выявления скрытого изображения требует наличие стеганоключа используемого при внедрении и обратен алгоритму внедрения:

$$S(x,y)' = \begin{cases} 1, \text{если } l'_{1\text{A}} - l'_{1B} < 0 \text{ и } l'_{2\text{A}} - l'_{2B} < 0 \\ 0, \text{если } l'_{1\text{A}} - l'_{1B} > 0 \text{ и } l'_{2\text{A}} - l'_{2B} > 0 \end{cases} \tag{1.9}$$

Метод Лангелаара [44]. Данный метод также подходит для скрываемых изображений как минимум в 8 раз меньше исходного. Алгоритм внедрения следующий:

1. Исходное изображение разбивается на блоки $I_b(x,y)$ размером 8×8 пикселей, в каждом из которых формируются две группы пикселей согласно стеганоключа $B(8,8)$, представленного матрицей из нулей и единиц, формируемой псевдослучайным образом.

2. Для каждой группы пикселей вычисляются средние значения яркости l_0 и l_1.

3. Для внедрения пиксела скрытого изображения $S(x,y)$ $\epsilon\{0,1\}$ выбирается некоторый порог α и каждый блок исходного изображения модифицируется таким образом, чтобы соблюдалось следующее условие:

$$S(x,y) = \begin{cases} 1, l'_0 - l'_1 > \alpha, \\ 0, l'_0 - l'_1 < -\alpha. \end{cases} \quad (1.10)$$

Алгоритм выявления скрытого изображения требует наличие стеганоключа используемого при внедрении. Латентное изображение разбивается на блоки, на основании стеганоключа блоки делятся на группы. По разнице между средними яркостями групп определяется значение бита скрытого изображения:

$$\mathrm{S(x,y)}' = \begin{cases} 1, \text{если } l'_0 - l'_1 > 0 \\ 0, \text{если } l'_0 - l'_1 < 0 \end{cases} \quad (1.11)$$

Метод формирования латентного изображения с внедрением цветного изображения [37].

Латентные изображения, построенные с использованием данного метода не устойчивы к разрушительным процессам, связанным с передачей, однако преимуществом данного метода является возможность внедрения полноцветного скрываемого изображения. Алгоритм формирования подразумевает:

1. Скрываемое изображение $S(x,y)$ растрируется с использованием нескольких растровых структур с получением матрицы $S_r(x,y)$, соразмерной с матрицей $S(x,y)$:

$$S_r(x,y) = k_1 S(x,y) + k_2 R_i(x,y), \quad (1.12)$$

где $R_i(x,y)$ – набор растровых структур; k_1, k_2 – неотрицательные константы.

2. Матрица $S_r(x,y)$ объединяется с инвертированной копией скрываемого изображения $S_{inv}(x,y)$, в результате чего получается матрица $S_{pre}(x,y)$:

$$S_{pre}(x, y) = k_1 S_{inv}(x, y) + k_2 S_r(x, y) \quad (1.13)$$

3. Матрица $S_{pre}(x, y)$ подвергается яркостным преобразованиям. С получением подготовленного к внедрению скрытого изображения S_{adj}:

$$S_{adj}(x, y) = \alpha S_{pre}(x, y) \quad (1.14)$$

4. Формируется латентное изображение $L(x, y)$ путем объединения основного изображения $I(x, y)$ и изображения с измененным контрастом S_{adj}:

$$L(x, y) = k_1 I(x, y) + k_2 S_{adj}(x, y) \quad (1.15)$$

Математическая модель формирования латентного изображения данным методом представлена следующей системой уравнений:

$$\begin{cases} S_r(x, y) = k_1 S(x, y) + k_2 R_i(x, y), \\ S_{pre}(x, y) = k_1 S_{inv}(x, y) + k_2 S_r(x, y), \\ S_{adj}(x, y) = \alpha S_{pre}(x, y), \\ L(x, y) = k_1 I(x, y) + k_2 S_{adj}(x, y), \end{cases} \quad (1.16)$$

Алгоритм выявления скрытого изображения предполагает подбор комбинации линейных фильтров «Усиление границ», «Цветовое тиснение», «Усреднение», «Размытие Барлета», «Слабое выделение границ (модифицированный)», «Размытие», «Цветовое тиснение», «Свечение», «Слабое выделение границ», «Колебание (сверху вниз 45°)». Учитывая, что число возможных комбинаций слишком велико, процесс выявления затруднителен.

1.3.2 Частотные методы формирования цифровых изображений, содержащих скрытую информацию

Метод Жао [45]. Данный метод подходит только для скрываемых изображений как минимум в 8 раз меньше исходного. Алгоритм формирования латентного изображения с использованием данного метода предполагает следующее:

1. Исходное изображение разбивается на блоки $I_b(x, y)$ размером 8×8 пикселей, после чего каждый блок подвергается дискретному косинусному преобразованию (ДКП) с получением матрицы коэффициентов $D_b(u, v)$.

2. Псевдослучайно из каждой матрицы $D_b(u,v)$ выбирается 2 коэффициента $D_b(u_1,v_1)$ и $D_b(u_2,v_2)$ со средними частотами. Из координат этих пар коэффициентов строится стеганоключ B.

3. Для внедрения пиксела скрытого изображения $S(x,y)\epsilon\{0,1\}$ выбирается некоторый порог α и каждая матрица $D_b(u,v)$ модифицируется таким образом, чтобы соблюдалось следующее условие:

$$S(x,y)=\begin{cases}1, |D_b(u_1,v_1)|-|D_b(u_2,v_2)|>\alpha,\\ 0, |D_b(u_1,v_1)|-|D_b(u_2,v_2)|<-\alpha.\end{cases} \quad (1.17)$$

4. Производится обратное ДКП каждой матрицы $D_b(u,v)$, в результате чего получается латентное изображение.

Алгоритм выявления скрытого изображения требует наличие стеганоключа B сформированного при внедрении. Латентное изображение разбивается на блоки, производится ДКП каждого блока, на основании стеганоключа выбираются коэффициенты $D_b{}'(u_1,v_1)$ и $D_b{}'(u_2,v_2)$. По разнице между этими коэффициентами определяется значение бита скрытого изображения:

$$S'(x,y)=\begin{cases}1, \text{если } D_b{}'(u_1,v_1)-D_b{}'(u_2,v_2)>0\\ 0, \text{если } D_b{}'(u_1,v_1)-D_b{}'(u_2,v_2)<0\end{cases} \quad (1.18)$$

Метод Хсу [46,47]. Данный метод подходит только для скрываемых изображений как минимум в 2 раза меньше исходного. Обобщенный алгоритм формирования латентного изображения с использованием данного метода предполагает следующее:

1. Скрываемое изображение $S(x,y)$ подвергается псевдослучайной перестановке, в результате чего получается матрица $RS(x,y)$.

2. Исходное изображение $I(x,y)$ разбивается на блоки $I_b(x,y)$ размером 8×8, матрица $RS(x,y)$ разбивается на блоки $RS_b(x,y)$ размером 4×4.

3. Матрицы блоков $I_b(x,y)$ подвергаются ДКП (получаем матрицы $D_b(x,y)$), из матриц полученных коэффициентов извлекаются 16 относящихся к средним частотам с формированием новых блоков $Dm_b(x,y)$ размерностью 4×4.

4. Вычисляются матрицы $P_b(x,y)\epsilon\{0,1\}$ полярностей разности соседних блоков $Dm_b(x,y)$ и $Dm_{b+1}(x,y)$. В данных матрицах значению 1 будут соответствовать положительные разности, а 0 – отрицательные и равные нулю.

5. Создаются модифицированные матрицы полярностей $P_b{}'(x,y)$ путем сложения по модулю 2 матриц $P_b(x,y)$ и $RS_b(x,y)$.

6. Каждая матрица $Dm_b(x,y)$ модифицируется таким образом, чтобы соблюдалось равенство $P_b'(x,y) = P_b(x,y)$. На выходе получим матрицу $Dm_b{}'(x,y)$.

7. Среднечастотные коэффициенты каждой из матриц $D_b(x,y)$ заменяются на соответствующие $Dm_b{}'(x,y)$.

8. Латентное изображение формируется как обратное ДКП матриц $Dm_b{}'(x,y)$. с их объединением.

Для выявления скрытого изображения необходимо наличие исходного изображения. Алгоритм выявления состоит в поблочном ДКП латентного и исходного изображения и нахождении их разностей. Подобно процессу формирования на основе полярности разностей средних частот определяется бит скрытой информации. На основании извлеченных битов строится соответствующая скрытому изображению матрица.

Метод Чея [48, 49]. Данный метод подходит только для скрываемых изображений как минимум в 4 раза меньше исходного. Алгоритм формирования латентного изображения с использованием данного метода предполагает следующее:

1. Выполняется одноуровневая декомпозиция исходного $I(x,y)$ и скрываемого $S(x,y)$ изображений с применением фильтра Хаара. С получением матриц вейвлет-коэффициентов $I_W(m,n)$ и $S_W(m,n)$ соответственно.

2. Матрицы коэффициентов $I_W(m,n)$ и $S_W(m,n)$ представляются 24-битным числом (включая бит знака) в виде $AAAAAAAABBBBBBBBCCCCCCCC_2$.

3. Коэффициенты $S_W(m,n)$ расширяются до размерности исходного изображения как показано на рисунке 1.1, образуя матрицу $ES(m,n)$.

AAAAAAAABBBBBBBBCCCCCCCC$_2$

AAAAAAAA0000000000000000$_2$	BBBBBBBB0000000000000000$_2$
CCCCCCCC0000000000000000$_2$	AAAAAAAA0000000000000000$_2$

Рисунок 1.1 – Механизм заполнения расширенной матрицы коэффициентов $ES(m,n)$.

4. Матрицы $I_W(m,n)$ и $ES(\ \ ,n)$ поэлементно складываются с использованием весового коэффициента α образуя матрицу $I_W{}'(m,n)$ (1.19). Полученное значение отображается назад к исходной шкале на основе значений минимального и максимального коэффициентов.

$$I_W{}'(m,n) = \alpha I_W(m,n) + ES(m,n) \tag{1.19}$$

5. Для получения латентного изображения $L(x,y)$ производится обратное дискретное вейвлет-преобразование коэффициентов $I_W{}'(m,n)$.

Для извлечения скрытого изображения требуется исходное изображение. Для извлечения вычисляются вейвлет коэффициенты скрытого изображения $S_W{}'(m,n)$ (1.20), по которым стоится само скрытое изображение $S'(x,y)$.

$$S_W{}'(m,n) = \frac{L_{\mathrm{W}}(m,n) - I_W(m,n)}{\alpha I_W(m,n)} \tag{1.20}$$

Обзор существующих методов показал, что они не позволяют внедрять цветные изображения, в качестве скрываемой информации за исключением метода, представленного в работе [37], однако, в сформированных этим методом латентных изображениях проявляется инородная структура, поэтому необходима разработка таких моделей формирования латентных изображений, которые позволяли бы получить более качественные изображения.

1.4 Методы внедрения скрытых изображений в полиграфическую продукцию

Особенностью методов внедрения скрытых изображений в полиграфическую продукцию, в отличие от методов внедрения в цифровые

изображения, является требование устойчивости таких изображений к цифро-аналоговым преобразованиям и аналогово-цифровым преобразованиям [14].

Выявление осуществляется с помощью аналогового устройства или специализированного программного обеспечения [50].

1.4.1 Методы, основанные на использовании различно-ориентированных растровых структур

Для внедрения скрытого изображения с помощью методов, основанных на использовании различно-ориентированных растровых структур [51], выбираются две растровые структуры со схожими или совпадающими частотными характеристиками, с различным ориентированием. Обычно для формирования данного типа латентных изображений выбираются линейчатые растры.

Внедрение скрываемого черно-белого изображения происходит в однородные области исходного изображения. Значимые биты скрываемого изображения растрируются с помощью одной растровой структуры, а остальная область внедрения – другой. Пример такого изображения представлен на рисунке 1.2.

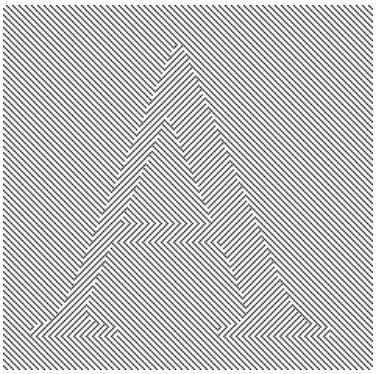


Рисунок 1.2 – Латентное изображение, с различно-ориентированными структурами

Собственные методы и технологии формирования данного типа имеются у компаний многих стран, занимающихся выпуском защищенной полиграфической продукции: Multicolor Latent Image, австрийской фабрики OeBS, LIFT – De La Rue (Великобритания), кипп-эффект – «Гознак» (Россия) [52].

1.4.2 Методы, основанные на использовании различных линиатур растра

Для построения латентного изображения, с помощью методов, основанных на использовании различных линиатур растра [53], выбираются две различные линиатуры со значениями, превышающими порог различения (>150 lpi).

Внедрение скрываемого черно-белого изображения происходит в однородные области исходного изображения. Значимые биты скрываемого изображения растрируются с одной линиатурой, а остальная область внедрения – другой. Примеры таких изображений представлены на рисунке 1.3.

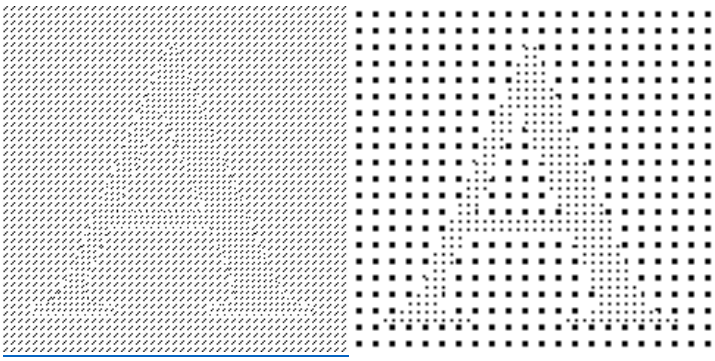


Рисунок 1.3 – Латентные изображения, с растрами различных линиатур

Выявление скрытого изображения происходит при приближении латентного изображения к глазу, за счет появления частотной различимости периодических структур. Скрытое изображение также может проявиться после сканирования или копирования латентного изображения вследствие анизотропии разрешения сканирующих и печатающих устройств. Например, при копировании латентного изображения, построенного растрированием с линиатурами 170 и 190 lpi, при максимальной разрешающей способности принтера и сканера в 180 lpi, можно получить идентичную копию только части изображения, растрированного с линиатурой 170lpi, остальная часть изображения изменит свой контраст.

1.4.3 Методы, основанные на использовании различных амплитудно-модулированных растровых структур

Для построения латентного изображения, с помощью методов, основанных на использовании различных амплитудно-модулированных растровых структур

[21], выбираются две различные амплитудно-модулированные растровые структуры со схожими частотными характеристиками, но имеющие явные структурные различия.

Внедрение скрываемого черно-белого изображения происходит в однородные области исходного изображения. Значимые биты скрываемого изображения растрируются одной растровой структурой, а остальная область внедрения – другой.

Данный метод формирования латентного изображения проиллюстрирован на рисунке 1.4.

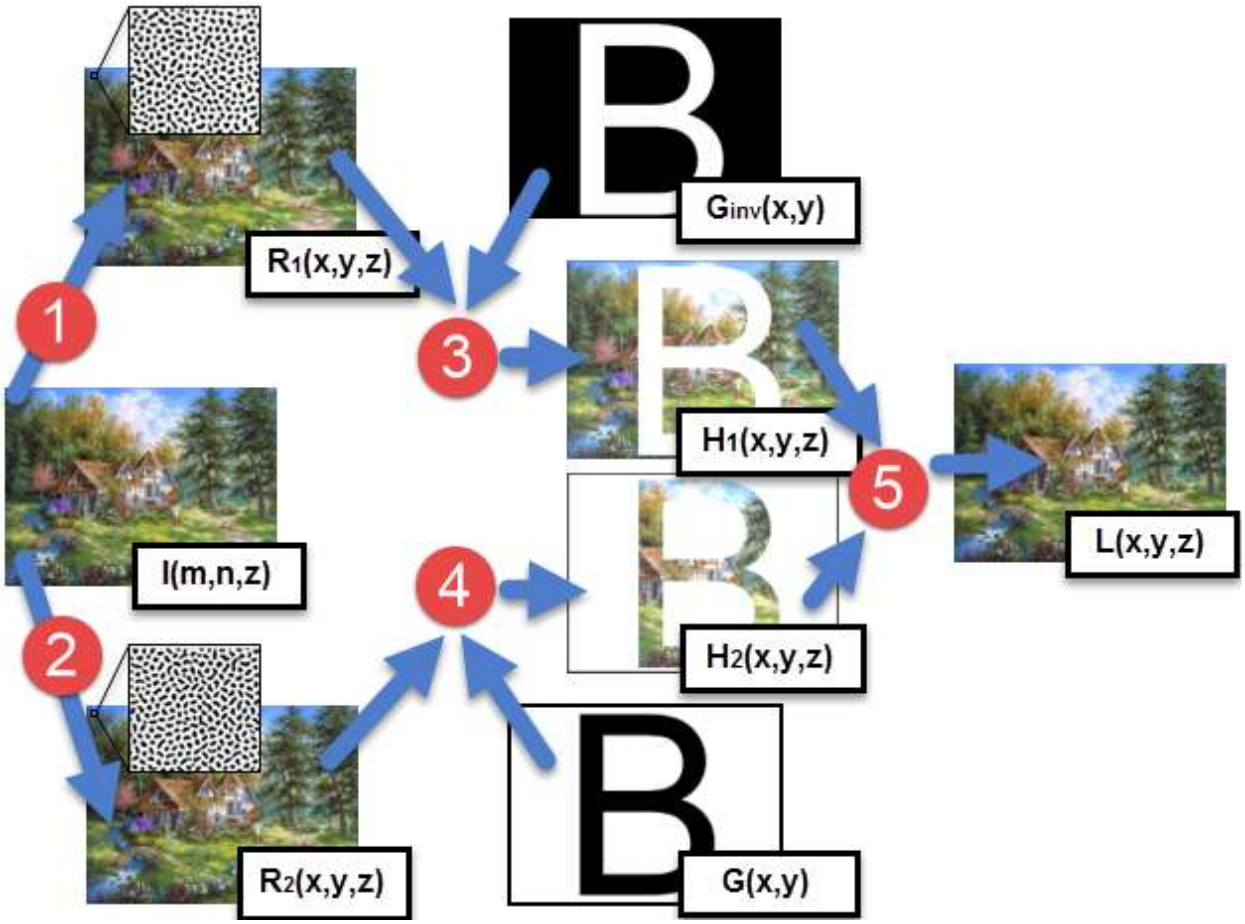


Рисунок 1.4 – Процесс формирования латентного изображения, методом [21]

На рисунке 1.4 представлены следующие элементы:

- $G(x, y)$ – скрываемое изображение;
- $I(m, n, z)$ – исходное изображение;
- $G_{inv}(x, y)$ – инвертированная копия скрытого изображения;

– $R_1(x, y, z)$ – растрированное исходное изображение первой растровой структурой;

– $R_2(x, y, z)$ – растрированное исходное изображение второй растровой структурой;

– $H_1(x, y, z)$ – подготовленное исходное изображение;

– $H_2(x, y, z)$ – подготовленное внедряемое изображение;

– $L(x, y, z)$ – латентное изображение.

Номерами на рисунке 1.4 обозначены следующие операции:

1. Исходное изображение $I(m, n, z)$ растрируется растровой структурой первого типа, с образованием растрированного исходного изображения $R_1(x, y, z)$:

$$R_1(x, y, z) = F_{R1}(I(m, n, z)), \tag{1.21}$$

где $I(m, n, z)$ – исходное изображение;

$R_1(x, y, z)$– растрированное исходное изображение растровой структурой первого типа;

$F_{R1}(I(m, n, z))$ – функция растрирования исходного изображения первого типа.

2. Исходное изображение $I(x, y, z)$ растрируется растровой структурой второго типа, с образованием растрированного исходного изображения $R_2(x, y, z)$:

$$R_2(x, y, z) = F_{R2}(I(m, n, z)), \tag{1.22}$$

где $I(m, n, z)$ – исходное изображение;

$R_2(x, y, z)$– растрированное исходное изображение растровой структурой второго типа;

$F_{R2}(I(m, n, z))$ – функция растрирования исходного изображения второго типа.

3. Из синей цветовой компоненты первого растрированного исходного изображения $R_1(x, y, z)$ вычетается по маске скрываемого изображения $G(x, y)$ область внедрения, с образованием подготовленного исходного изображения $H_1(x, y, 2)$.

$$H_1(x, y, 2) = R_1(x, y, 2) * G(x, y), \quad (1.23)$$

где $H_1(x, y, 2)$ – подготовленный к внедрению синий канал исходного изображения;

$R_1(x, y, 2)$ – синяя цветовая компонента растрированного исходного изображения растровой структурой первого типа;

$G(x, y)$ – скрываемое изображение.

4. Из синей цветовой компоненты второго растрированного исходного изображения $R_2(x, y, 2)$ вычитается по маске инвертированного скрываемого изображения $G_{inv}(x, y)$, вычисляемого по формуле 1.24, область внедрения, с образованием подготовленного скрываемого изображения $H_2(x, y, z)$, вычисляемого по формуле 1.25.

$$G_{inv}(x, y, z) = (2^L - 1) - G(x, y, z) \quad (1.24)$$

где $G(x, y, z)$ – скрываемое изображение;

$G_{inv}(x, y, z)$ – инвертированная копия скрываемого изображения;

L – число разрядов квантования.

$$H_2(x, y, 2) = R_2(x, y, 2) * G_{inv}(x, y), \quad (1.25)$$

где $H_2(x, y, z)$ – подготовленный к внедрению синий канал скрываемого изображения;

$G_{inv}(x, y)$ – инвертированная копия скрываемого изображения.

5. Синий канал латентного изображения формируется путем объединения подготовленных изображений $H_1(x, y, 2)$ и $H_2(x, y, 2)$.

$$L(x, y, 2) = H_1(x, y, 2) + H_2(x, y, 2), \quad (1.26)$$

где $H_1(x, y, 2)$ – подготовленный к внедрению синий канал исходного изображения;

$H_2(x, y, 2)$ – подготовленный к внедрению синий канал скрываемого изображения;

$L(x, y, 2)$ –синий канал латентного изображения.

Выявление скрытого изображения происходит с помощью частотного или пространственного фильтров.

1.4.4 Методы, основанные на фазовом сдвиге растровых структур

С помощью методов, основанных на фазовом сдвиге растровых структур [54], внедрение скрываемого черно-белого изображения происходит в однородные области исходного изображения. Исходное изображение растрируется, затем для внедрения значимого бита скрываемого изображения происходит сдвиг растровой точки в определенном направлении с шагом менее периода растровой структуры. Самый простой пример такого изображения представлен на рисунке 1.5.

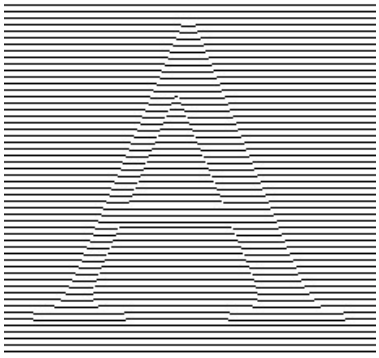


Рисунок 1.5 – Латентное изображение, со сдвигом фазы растровых структур

Выявление скрытого изображения происходит при наложении фильтра с периодом совпадающем при растрировании исходного изображения, при этом происходит усиление контраста в области со скрытым изображением (рисунок 1.6).

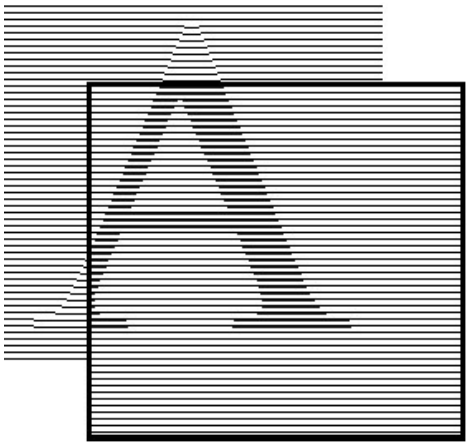


Рисунок 1.6 – Выявленное наложением фильтра скрытое изображение

Фильтр может быть представлен как в цифровом, так и в аналоговом виде. Скрытое изображение также может проявиться после сканирования или копирования латентного изображения вследствие анизотропии разрешения сканирующих и печатающих устройств.

1.4.5 Физико-химические методы внедрения и контроля скрытой информации

Формирование физико-химических латентных изображений происходит непосредственно на защищаемой поверхности либо веществами не воспринимаемыми человеческим глазом при нормальных условиях окружающей среды, но проявляемых при определенном физическом или химическом воздействии, либо наносятся специальными способами, обеспечивающие видимость скрытого изображения под определенным углом. Этапу передачи соответствует весь период существования защищаемого объекта. Выявление таких изображений осуществляется с помощью специальных устройств или невооруженным взглядом при изменении условий наблюдения.

Голограммы. Голограммы представляют собой особый класс изображений, наносимых на фотографическую пластину с помощью лазера, отображающих собой образ трехмерного объекта с сохранением возможности изменения перспективы при изменении точки наблюдения [55]. Изображение на голограмму наносится в виде ряда линий, представляющих тонкие углубления на поверхности, при этом расстояние между ними должно быть соизмеримо с длиной световой волны. Технология предполагает, что на одном миллиметре помещается до нескольких десятков таких линий [56].

Различают следующие виды голограмм:

– Традиционная голограмма, формируемая путем регистрации на фоточувствительном материале отраженного, специальным образом, пучка света от реального объекта.

– Dot-Matrix голограмма, формируемая на специальном оборудовании, генерирующем на основании цифрового изображения матрицу из дифракционных решеток с последующей записью на фоточувствительном материале.

– E-beam голограмма, генерируемая на основании цифрового изображения с использованием компьютерного моделирования и записываемая на фоточувствительном материале методом электро-лучевой литографии [57].

Учитывая, что определенные грани зафиксированного на голограмме объекта становятся видимыми под определенным углом обзора, классическая голограмма сама по себе подпадает под определение латентного изображения. Каждая из граней голограммы представляет собой скрытое изображение. Таким образом, голограмма содержит в себе несколько скрытых изображений.

Помимо скрытых изображений, проявляемых на голограммах при изменении угла обзора, в голограммы, созданные по технологии E-beam, могут встраиваться элементы, считываемые приборами в ультрафиолетовых, инфракрасных или лазерных лучах. На голограмме, области с такими латентными изображениями, образуют матовый фон, представленный хаотичной структурой, видимой только под большим увеличением. Для визуализации необходимо наличие специального прибора [57]. На российском рынке приборов для визуализации данного вида скрытых изображений широкое распространение получило устройство DORS 25 компании «Система». При поднесении устройства к области с латентным изображением на защищенной голограмме, на экране отображается скрытое изображение [58].

Латентные изображения на голограммах, представленные хаотичными структурами могут быть:

– простыми, содержащими одиночные скрытые изображения;

– динамическими, содержащими набор скрытых изображений. При проведении в определенном направлении считывающего прибора над такими латентными изображениями на экране образуется анимированная картинка;

– многоуровневыми, содержащими в одной области несколько скрытых изображений, каждое из которых выявляется при разных углах облучения прибором [57].

Нанесение скрытых изображений на голограмму также возможно с использованием гильоширных узоров [56].

Себестоимость производства голографических изображений, в виду дороговизны изготавливающего оборудования, на сегодняшний день достаточно высока и может быть оправдана только при изготовлении небольших наклеек. При этом, по мнению экспертов, в ближайшее время, широкого распространения данного вида латентных изображений в РФ не предвидится [59].

Защитные краски. На сегодняшний день защитные краски нашли широкое применение как средство технологической защиты печатной продукции. Благодаря сложному составу, изображения, нанесенные этой краской, становятся видимыми или меняют свой цвет в зависимости от условий освещения и наблюдения. В обычных условиях эти краски являются бесцветными.

После нанесения скрытого изображения на любую поверхность, включая печатную продукцию, с помощью защитных красок, визуальный образ, создаваемый данной поверхностью, признается латентным изображением.

Различают следующие виды красок применяемых для создания латентных изображений, по типу воздействия необходимого для выявления срытых изображений:

– Флуорисцентные краски. Для выявления скрытого изображения, нанесенного данным видом красок необходимо, в момент считывания, подвергнуть латентное изображение воздействию УФ-излучения.

– Фотохромные краски. Выявление скрытых изображений, нанесенных данным видом красок возможно только в условиях солнечного света, используемого в момент считывания латентного изображения.

– Иридисцентные краски. Для выявления скрытого изображения, нанесенного данным видом красок необходимо, в момент считывания, подвергнуть латентное изображение воздействию ИК-излучения.

– Термохромные краски. Латентное изображение должно быть подвергнуто тепловому воздействию, в момент считывания, с целью выявления скрытого изображения, нанесенного данным видом красок.

– Оптихромные краски. Данный вид красок содержит специальные компоненты, обеспечивающие преломление света, падающего под разным углом. Для выявления скрытого изображения, нанесенного данным видом красок необходимо латентное изображение, в момент считывания, повернуть к источнику света под определенным углом [60].

Поляризационные пленки со скрытым изображением. Группа способов [61 – 64] предполагает формирование скрытых изображений наложением изготовленной специальным образом прозрачной поляризационной пленки на исходное или другое латентное изображение. Поляризационная пленка создается из прозрачных полимеров с нанесением термическим, химическим или радиационным способом скрытого изображения, визуализируемого при просмотре через поляризатор.

Поляризатор представляет собой устройство с источником поляризационного света и поляризационным фильтром. При наложении поляризатора на латентное изображение через светофильтр становится видимым скрытое изображение [65 – 66].

Поляризационные латентные изображения обладают высокой износостойкостью и достаточно высоким уровнем защиты от копирования. Однако высокая стоимость изготовления и требование к наличию специального поляризатора для контроля данного вида латентных изображений являются существенными факторами, определяющими их не высокое распространение.

Водяные знаки. Латентное изображение с внедренным технологическим водяным знаком представляет собой полиграфическую продукцию, выполненную

на бумаге, для отдельных участков которой характерны различные значения плотности [5]. Нанесение водяного знака осуществляется за счет уменьшения или увеличения толщины бумаги, на которой нанесено исходное изображение, таким образом, чтобы участки с измененной толщиной бумаги представляли образ скрытого изображения.

Контроль изображений, содержащих водяной знак, осуществляется просвечиванием носителя, в результате чего скрытое изображение становится видимым.

Изображения в виде водяного знака могут быть однотоновые, двутоновые, многотоновые и комбинированные скрытые изображения. Однотоновые скрытые изображения образуются за счет либо уменьшения, либо увеличения толщины бумаги. Двутоновые скрытые изображения предполагают, как увеличение, так и уменьшение толщины бумаги. Внедрение многотоновых скрытых изображений осуществляется с использованием различных степеней утончения и утолщения бумаги. Комбинированные латентные изображения содержат несколько тоновых видов скрытых изображений.

Продукция, защищенная водяными знаками, обладает достаточно высоким уровнем защиты от копирования и получила широкое распространение за счет простоты контроля. Однако технологический процесс создания водяных знаков является слишком сложным и требует специального дорогостоящего оборудования. А также водяные знаки имеют техническое ограничение в виде поверхности нанесения – латентные изображения данного класса могут распространяться только на бумажном носителе, что делает их неприменимыми для решения ряда задач защиты промышленной продукции [67].

1.4.6 Программное и аппаратное обеспечение для внедрения скрытой информации в полиграфические изделия и их контроля

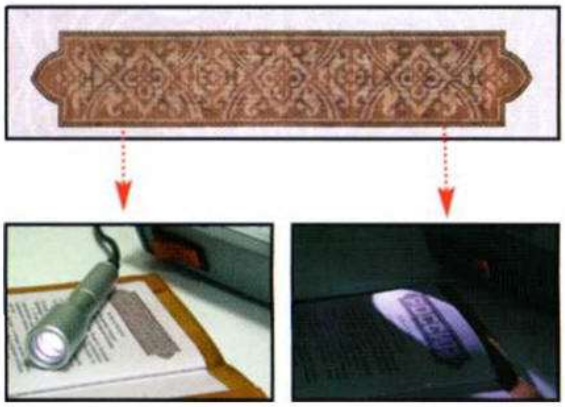
Имеющееся программное обеспечение, предназначенное для массового потребителя весьма ограничено и представлено программными продуктами CERBER компании SecureSoft [68], создающий латентные изображения с

использованием гильоширных узоров с фазовым сдвигом скрываемого изображения, и BBS Designer компании GuardSoft [69], создающий латентные изображения по собственной технологии Ghost on Duty. Для выявления изображений, сформированных по технологии Ghost on Duty, компания GuardSoft предлагает специальные оптические ключи.

Устройства формирования представлены электрографическими печатающими устройствами, поддерживающими технологии Glossmark, SafePaper, TrustMark и DataGlyphs. Данные технологии предусматривают специальные методы растрирования, реализуемые непосредственно аппаратной частью. Для выявления скрытых изображений, сформированных по технологиям SafePaper, TrustMark и DataGlyphs требуется специальный оптический ключ. Для выявления изображений Glossmark достаточно изменить угол обзора напечатанного латентного изображения [67].

В качестве аналоговых распознающих устройств выступают приборы с источниками косопадающего света, например, приборы компаний DORS [70] и «Ультрамаг» [71], представленные на рисунке 1.7.

а



Б

Рисунок 1.7 – Детекторы используемые для визуализации ЛИ. а – Универсальный просмотровый детектор DORS 1300 M2, б – прибор «Ультрамаг-5СЛГ»

Выявление скрытого изображения происходит при повороте латентного изображения, за счет появления частотных различий проекций растровых структур. Скрытое изображение также может проявиться после сканирования латентного изображения вследствие анизотропии разрешения сканирующих устройств.

Программные средства для выявления латентных изображений, такие как Regula Video Scope [72], позволяют только визуализировать латентные изображения.

При внедрении скрытых изображений в полиграфическую продукцию учитываются не только свойства человеческого зрения, но и АЦП, и ЦАП, происходящие при передаче латентных изображений. В случаях, когда факт наличия скрытой информации в печатной продукции не определяется визуально при изменении угла обзора латентного изображения, оперативный контроль такой продукции не возможен и требуется индивидуальный специальный алгоритм для выявления каждого типа латентного изображения. Разработка методов контроля с меньшей вычислительной сложностью контроля скрытой информации позволит обеспечить возможность автоматизацию процесса контроля латентных изображений.

1.5 Постановка задач исследования

Согласно формулам 1.2 и 1.3 визуальное качество преобразованного изображения максимально при минимальном изменении цветовых компонент изображений исходного $I(x, y, z)$ и латентного $L(x, y, z)$. Качество самого латентного изображения $L(x, y, z)$ выше при минимальном искажении выявленного изображения $G'(x, y, z)$ по сравнению со скрытым $G(x, y, z)$. Таким образом, математическая постановка задачи заключается в следующем:

$$\begin{cases} \text{ПОСШ}\left(L(x_i, y_j, z_k), I(x_i, y_j, z_k)\right) & \to max, \\ \text{ПОСШ}\left(G'(x_i, y_j, z_k), G(x_i, y_j, z_k)\right) & \to max, \end{cases} \tag{1.27}$$

где $L(x, y, z)$ – латентное изображение;

$I(x, y, z)$ – исходное изображение;

$G(x, y, z)$ – скрываемое изображение;

$G'(x, y, z)$– выявленное скрытое изображение;

x, y, z – координаты цветовых компонент в системе RGB;

i, j – координаты пикселя над которым производится преобразование;

k – цветовая компонента в системе RGB. $k = 1$ – красная, $k = 2$ – зеленая, $k = 3$ – синяя.

Для сокращения вычислительной сложности визуализации скрытой информации необходимо разработать численный метод и математическую модель для выявления скрытой информации с наименьшим числом преобразований изображения приводящий к гарантированному выявлению. Таким образом, математическая постановка задачи заключается в следующем:

$$\left\{\begin{matrix} F_1\ (L(x, y, z)), \\ \dots \\ F_k\ (L(x, y, z)), \\ G'(x, y, z) = F_k\ (L(x, y, z)) \end{matrix}\right.\Bigg|_{k \to min} \tag{1.28}$$

где $L(x, y, z) -$ латентное изображение;

$F(L(x, y, z))$ – результат преобразования латентного изображения;

$G'(x, y, z)$ – выявленное скрытое изображение;

x, y, z – координаты цветовых компонент в системе RGB;

k – номер последнего преобразования, в результате которого выявилась скрытая информация.

Разработка методов автоматического контроля латентных изображений позволит увеличить надежность контроля, а также сократить число ложных срабатываний автоматических устройств контроля.

Целью диссертационной работы является повышение визуального качества формируемых латентных изображений и сокращение вычислительной сложности контроля скрытой информации, за счет совершенствования методов внедрения и выявления скрытых изображений, для обеспечения возможности автоматизации процесса контроля латентных изображений.

Научная задача – разработка методов формирования латентных изображений, с учетом свойств системы человеческого зрения, что обеспечит повышение визуального качества изображения, а также методов автоматического контроля латентного изображения.

Реализация поставленной задачи разбита на следующие частные задачи:

− разработка метода моделирования и построение математической модели формирования латентных изображений, учитывающих свойства системы человеческого зрения;

− разработка численного метода контроля латентных изображений на основе новой математической модели выявления скрытых изображений, позволяющей обнаружить скрываемую информацию при неизвестном методе ее внедрения;

− разработка системы формирования и контроля латентных изображений, реализующих предложенные методы;

− разработка алгоритмов формирования и программного модуля для формирования латентного изображения;

− разработка алгоритмов выявления скрытых изображений и программного комплекса для контроля латентных изображений.

1.6 Выводы

В диссертационной работе, для обозначения изображений содержащих скрытую информацию в виде другого изображения используется термин «*латентное изображение*». К латентным изображениям относятся изображения, основным свойством которых является изменение видимости элементов изображения при изменении условий наблюдения или способа регистрации основного изображения.

Приведенный обзор литературных источников [21 – 23, 37 – 39, 41 – 49, 51 – 57, 60 – 64] показал, что существующие методы внедрения скрытой информации не учитывают всех свойств человеческого зрения, при внедрении цветных изображений, в латентных изображениях появляется инородная структура,

свидетельствующая о факте внедрения. Поэтому необходимо разработать такие методы формирования латентных изображений, которые бы наиболее полно учитывали свойства человеческого зрения, что позволит обеспечить повышение визуального качества латентного изображения без потери качества выявляемого изображения.

Представление полиграфической и цифровой графической продукции в виде латентных изображений позволяет обеспечить их защиту без нарушения визуальной композиции изображения, однако существующие методы формирования латентных изображений доступны только для квалифицированных специалистов в данной области, поэтому, несмотря на высокую эффективность защиты, латентные изображения используются только для защиты особо ценной полиграфической продукции, такой как банкноты, ценные бумаги и контрольные документы. Поэтому, разработка дешевых и эффективных методов внедрения скрытых изображений в продукцию массового потребления является востребованной на сегодняшний день.

На текущий момент существует шесть основных методов автоматического контроля полиграфической продукции: контроль габаритов изображения, контроль видимого изображения, контроль инфракрасного изображения, контроль ультрафиолетовых меток, контроль магнитных меток, контроль встроенных электронных носителей.

Несмотря на то, что латентные изображения подпадают под требования ГОСТа Р54109-2010 [5] к машиносчитываемым элементам, однако, в настоящее время, автоматический контроль таких изображений, в виду высокой вычислительной сложности, не применяется. Осуществляется только их визуальный контроль с помощью существующих программно-аппаратных средств. Сокращение вычислительной сложности визуализации скрытой информации позволит сделать процесс автоматического контроля целесообразным.

При внедрении скрытых изображений в полиграфическую продукцию учитываются не только свойства человеческого зрения, но и АЦП, и ЦАП,

происходящие при передаче латентных изображений. В случаях, когда факт наличия скрытой информации в печатной продукции не определяется визуально при изменении угла обзора латентного изображения, оперативный контроль такой продукции не возможен и требуется индивидуальный специальный алгоритм для выявления каждого типа латентного изображения. Разработка методов контроля с меньшей вычислительной сложностью контроля скрытой информации позволит обеспечить возможность автоматизацию процесса контроля латентных изображений.

На основе результатов обзора литературных источников определены цель и научная задача. Научная задача была декомпозирована на 5 подзадач. Формулами 1.27 и 1.28 определена математическая постановка задачи.

ГЛАВА 2 РАЗРАБОТКА МАТЕМАТИЧЕСКИХ МЕТОДОВ И АЛГОРИТМОВ ФОРМИРОВАНИЯ ЦИФРОВЫХ ЛАТЕНТНЫХ ИЗОБРАЖЕНИЙ

В отличие от известных методов [21,37] в диссертационной работе предлагается с помощью разработанной математической модели формировать латентное изображение не только за счет комбинации различных растровых структур, образующих основное изображение, области которых определяются на основе взаимно обратных позитивной и негативной масок внедряемого изображения, но и за счет комбинации двух низкочастотных квазипериодических, что позволит разработать модели учитывающие одновременно свойства человеческого зрения 1 – 3, 5 – 6 из перечня, приведенного в п. 1.2.

2.1 Метод моделирования латентного изображения

Как показано в п. 1.3 существующие методы формирования цифровых латентных изображений принято делить на две группы: пространственные и частотные. Суть пространственных методов заключается в преобразованиях яркостей исходного изображения или одной из его цветовых составляющих. Частотные методы предполагают изменение исходного изображения таким образом, чтобы при частотном разложении полученного латентного изображения скрытое изображение оказалось в локализованной частотной области, как правило, низкочастотной.

Как известно, диагональные линии воспринимаются человеческим глазом как низкочастотный сигнал [34]. Учитывая, что частотная чувствительность диагональных структур ниже, чем у горизонтальных и вертикальных [35], можно предположить, что незначительные преобразования в диагональных структурах не приведут к серьезным визуальным искажениям изображения. Принимая также во внимание свойство человеческого зрения неразличимости периодических структур, состоящих из штрихов и пробелов одинаковой ширины, при угле зрения

между центрами двух штрихов менее 1'30"[35], внедрение должно происходит в структуры с минимальными толщинами линий и пробелов между ними.

Разработка комбинированного метода моделирования латентных изображений, основанного на использовании как пространственных, так и частотных преобразований, позволит получить латентные изображения с незначительными искажениями яркости и контраста по сравнению с исходным и с возможностью получения при детектировании достаточно качественного для идентификации выявленного изображения.

2.1.1 Метод моделирования латентного изображения с внедрением цветного изображения

Для внедрения скрываемого изображения $G(x, y, z)$ в низкочастотную составляющую исходного изображения $I(x, y, z)$ производим подготовку скрываемого изображения, преобразовав его в низкочастотный сигнал, представляющий собой внедряемое изображения $H(x, y, z)$. Для этого объединим два изображения: скрываемое изображение $G(x, y, z)$ и его инвертированную копию $G_{inv}(x, y, z)$, чередуя диагональные линии матрицы яркостей изображений $G(x, y, z)$ и $G_{inv}(x, y, z)$. Для реализации такого объединения необходимо растрировать скрываемое изображение, удалив нечетные диагональные линии в $G(x, y, z)$, и инвертированную копию, удалив четные диагональные линии в $G_{inv}(x, y, z)$. Схематично процесс подготовки внедряемого изображения показан на рисунке 2.1.

Внедрение производится путем объединения яркостей исходного изображения $I(x, y, z)$ и подготовленного внедряемого $H(x, y, z)$. Экспериментально определено, что объединение с применением коэффициентов 1 для исходного и 0,1 для скрываемого позволяют получить оптимальный результат согласно формуле (1.27).

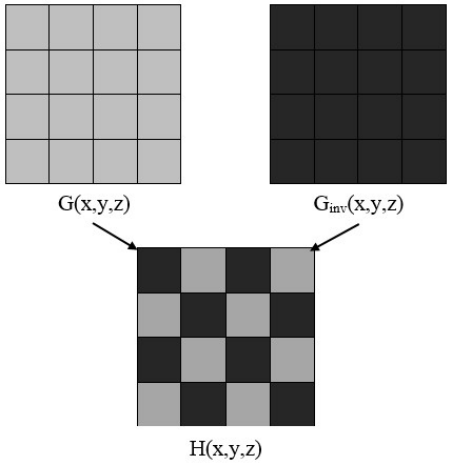


Рисунок 2.1 – Принципиальная схема подготовки внедряемого изображения

Разработанный метод моделирования процесса формирования латентного изображения, с внедрением цветного изображения проиллюстрирован на рисунке 2.2.

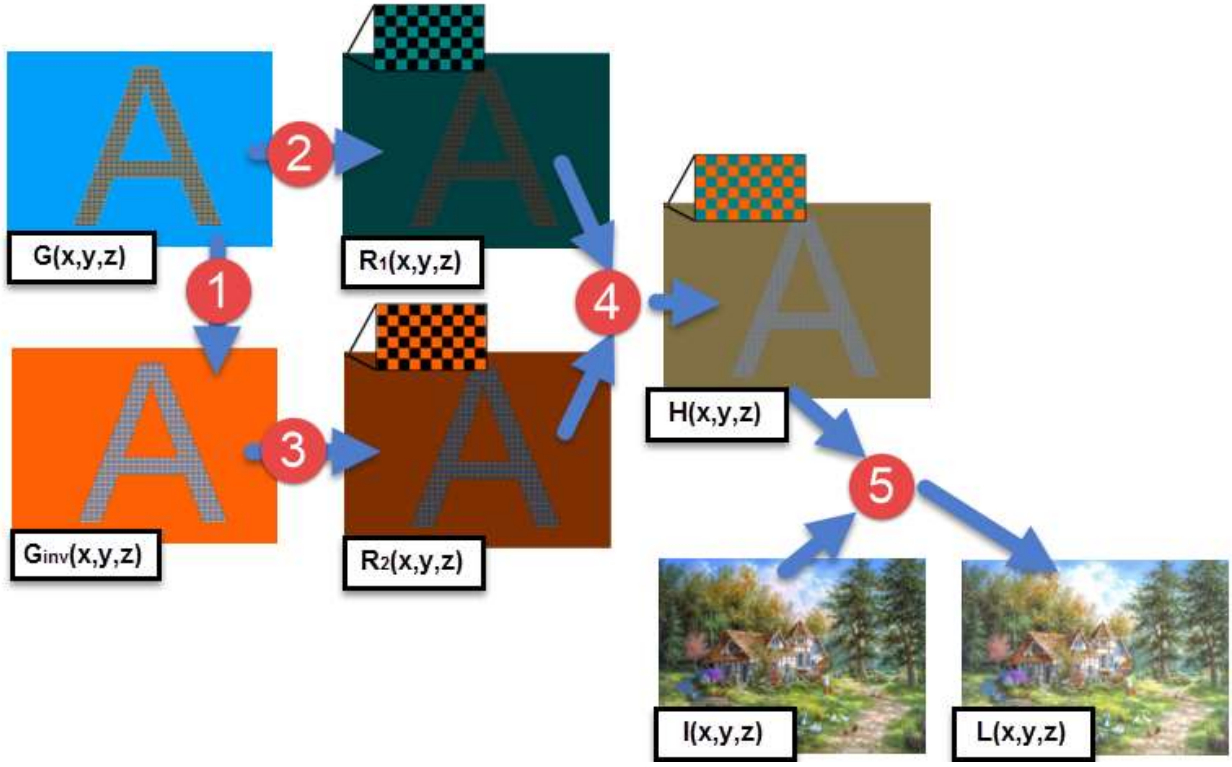


Рисунок 2.2 – Процесс формирования латентного изображения, с внедрением цветного изображения

На рисунке 2.2 представлены следующие элементы:

– $G(x, y, z)$ – скрываемое изображение;

– $I(x, y, z)$ – исходное изображение;

– $G_{inv}(x, y, z)$– инвертированная копия скрытого изображения;

– $R_1(x, y, z)$ – растрированное скрываемое изображение;

– $R_2(x, y, z)$ – растрированная инвертированная копия скрытого изображения;

– $H(x, y, z)$ – подготовленное внедряемое изображение;

– $L(x, y, z)$ – латентное изображение.

Номерами на рисунке 2.2 обозначены следующие операции:

1. Создается инвертированная копия $G_{inv}(x, y, z)$ скрываемого изображения $G(x, y, z)$:

$$G_{inv}(x, y, z) = (2^L - 1) - G(x, y, z) \tag{2.1}$$

2. Скрываемое изображение $G(x, y, z)$ растрируется, с удалением нечетных диагональных линий, образуя первую низкочастотную квазипериодическую структуру $R_1(x, y, z)$:

$$\begin{cases} R_1(x_i, y_j, z_k) = F_{R1}\left(G(x_i, y_j, z_k)\right), \\ R_1(x_{i+1}, y_{j+1}, z_k) = F_{R1}\left(G(x_{i+1}, y_{j+1}, z_k)\right), \end{cases}\Bigg|_{\substack{i=1,3,\dots,N \\ j=1,3,\dots,M \\ k=1,2,3}} \tag{2.2}$$

3. Инвертированная копия скрытого изображения $G_{inv}(x, y, z)$ растрируется, удалением четных диагональные линий, образуя вторую низкочастотную квазипериодическую структуру $R_2(x, y, z)$:

$$\begin{cases} R_2(x_i, y_{j+1}, z_k) = F_{R2}\left(G_{inv}(x_i, y_{j+1}, z_k)\right), \\ R_2(x_{i+1}, y_j, z_k) = F_{R2}\left(G_{inv}(x_{i+1}, y_j, z_k)\right), \end{cases}\Bigg|_{\substack{i=1,3,\dots,N \\ j=1,3,\dots,M \\ k=1,2,3}} \tag{2.3}$$

4. Создается подготовленное внедряемое изображение $H(x, y, z)$ путем объединения квазипериодических структур $R_1(x, y, z)$ и $R_2(x, y, z)$:

$$H(x, y, z) = R_1(x, y, z) + R_2(x, y, z) \tag{2.4}$$

5. Латентное изображение формируется путем встраивания исходного изображения $I(x,y,z)$ и подготовленного внедряемого $H(x,y,z)$ с коэффициентами a и b соответственно:

$$L(x,y,z) = aI(x,y,z) + bH(x,y,z) \quad (2.5)$$

2.1.2 Метод моделирования латентного изображения, с внедрением черно-белого изображения

Для внедрения скрываемого черно-белого изображения $G(x,y)$ в цветное $I(x,y,z)$ предлагается, при внедрении черно-белого изображения, основное изображение $I(x,y,z)$ объединять со своей растрированной копией $I_R(x,y,z)$, на которую наложено по маске скрытое изображение $G(x,y)$.

Разработанный метод моделирования процесса формирования латентного изображения, с внедрением черно-белого изображения проиллюстрирован на рисунке 2.3.

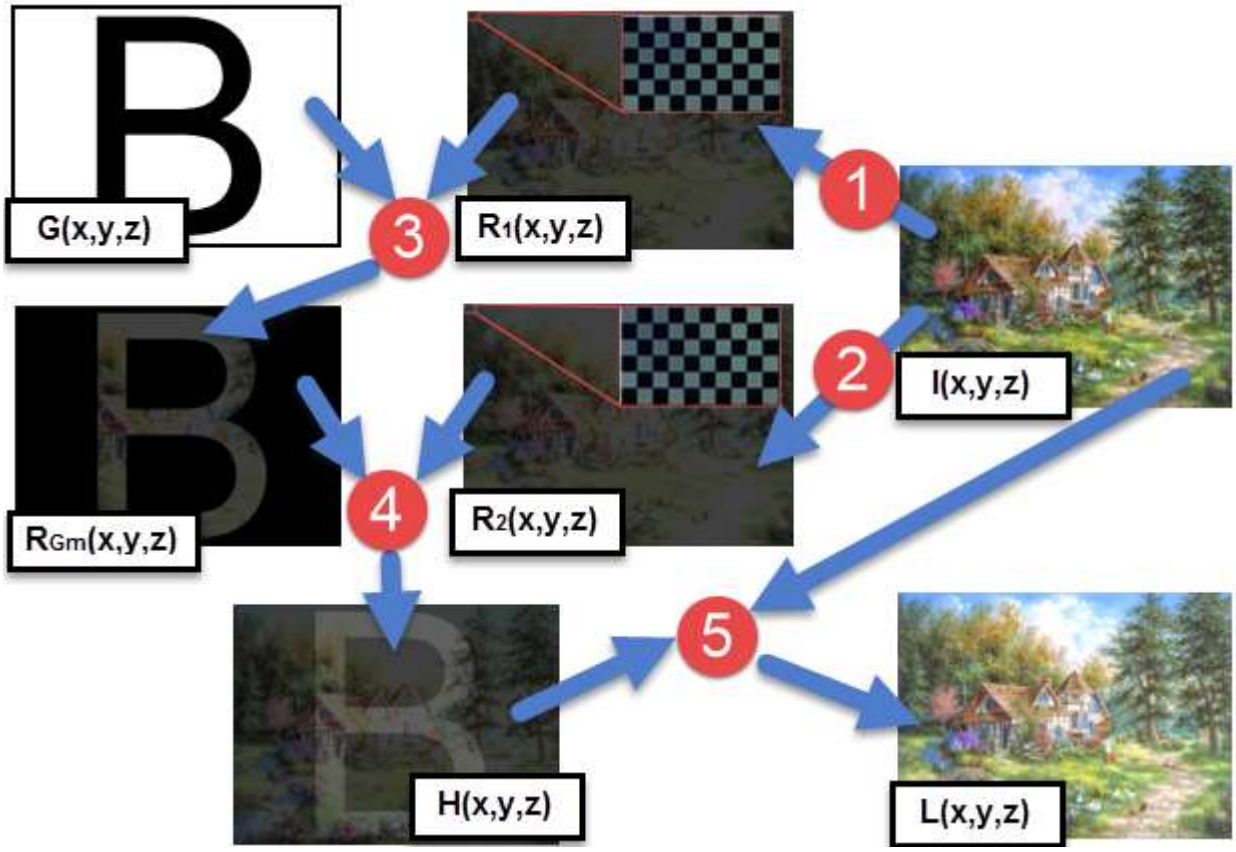


Рисунок 2.3 – Процесс формирования латентного изображения, с внедрением черно-белого изображения

На рисунке 2.3 представлены следующие элементы:

– $G(x, y, z)$ – скрываемое изображение;

– $I(x, y, z)$ – исходное изображение;

– $R_1(x, y, z)$ – растрированное исходное изображение с удалением четных диагональных линий;

– $R_2(x, y, z)$ – растрированное исходное изображение с удалением нечетных диагональных линий;

– $R_{Gm}(x, y, z)$ – растровая структура $R_1(x, y, z)$ маскированая по скрываемому изображению $G(x, y, z)$;

– $H(x, y, z)$ – подготовленное внедряемое изображение;

– $L(x, y, z)$ – латентное изображение.

Номерами на рисунке 2.3 обозначены следующие операции:

1. Исходное изображение $I(x, y, z)$ растрируется, с удалением нечетных диагональных линий, образуя первую низкочастотную квазипериодическую структуру $R_1(x, y, z)$:

$$\begin{cases} R_1(x_i, y_j, z_k) = F_{R1}\left(I(x_i, y_j, z_k)\right), \\ R_1(x_{i+1}, y_{j+1}, z_k) = F_{R1}\left(I(x_{i+1}, y_{j+1}, z_k)\right), \end{cases}\Bigg|_{\substack{i=1,3,\dots,N \\ j=1,3,\dots,M \\ k=1,2,3}} \tag{2.6}$$

2. Копия исходного изображения $I(x, y, z)$ растрируется, удалением четных диагональные линий, образуя вторую низкочастотную квазипериодическую структуру $R_2(x, y, z)$:

$$\begin{cases} R_2(x_i, y_{j+1}, z_k) = F_{R2}\left(I(x_i, y_{j+1}, z_k)\right), \\ R_2(x_{i+1}, y_j, z_k) = F_{R2}\left(I(x_{i+1}, y_j, z_k)\right), \end{cases}\Bigg|_{\substack{i=1,3,\dots,N \\ j=1,3,\dots,M \\ k=1,2,3}} \tag{2.7}$$

3. Из первой квазипериодической структуры $R_1(x, y, z)$ вычитается по маске скрываемого изображения $G(x, y)$ область внедрения, с образованием растровой структуры скрываемого изображения $R_{Gm}(x, y, z)$:

$$R_{Gm}(x, y, z) = G(x, y) * R_1(x, y, z) \tag{2.8}$$

4. Создается подготовленное внедряемое изображение $H(x, y, z)$ путем объединения растровой структуры скрываемого изображения $R_{Gm}(x, y, z)$ и квазипериодической структуры $R_2(x, y, z)$:

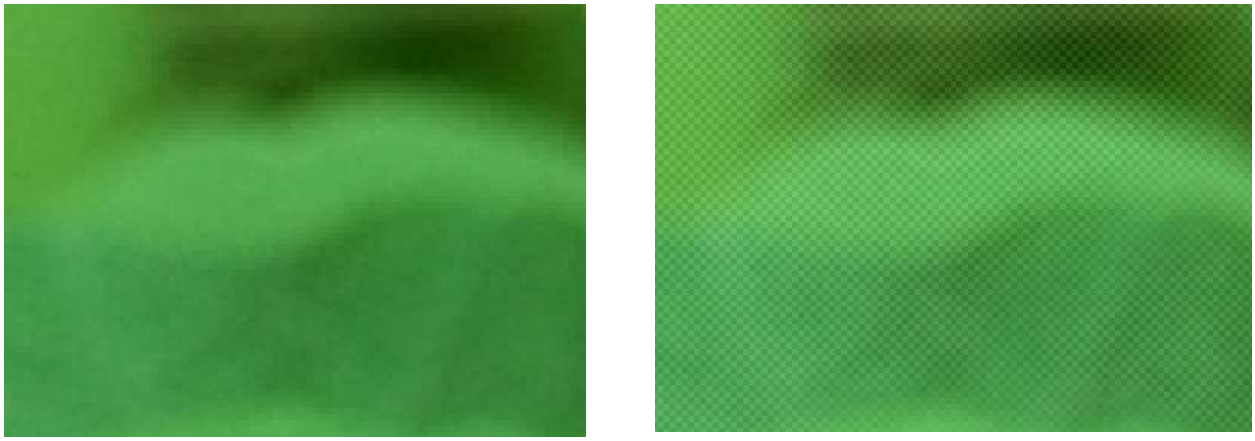
$$H(x, y, z) = R_{Gm}(x, y, z) + R_2(x, y, z) \quad (2.9)$$

5. Латентное изображение формируется путем встраивания исходного изображения $I(x, y, z)$ и подготовленного внедряемого $H(x, y, z)$ с коэффициентами a и b соответственно:

$$L(x, y, z) = aI(x, y, z) + bH(x, y, z) \quad (2.10)$$

2.1.3 Зависимость качества формируемого изображения от выбранного способа растрирования

Несмотря на визуальную неразличимость исходного и латентного изображений, построенного комбинированным методом, при большом увеличении латентного изображения явственно проявляется растровая структура (рисунок 2.4).



а *б*

Рисунок 2.4 – Увеличенные изображения: а) исходное, б) латентное

С учетом свойства человеческого зрения, заключающегося в том, что при одновременном воздействии на глаз двух компонентов изображения со схожими характеристиками эти компоненты воспринимаются как один компонент [33], уменьшим данный эффект. Для этого при построении модели формирования предложенным методом моделирования обеспечим неразличимость соседних

пикселей внедряемого изображения $H(x, y, z)$ путем усреднения яркостей соседних пикселей скрываемого изображения $G(x, y, z)$ и его инвертированной копии $G_{inv}(x, y, z)$. Для сохранения размера и разрешения изображения необходимо, чтобы размер растровой точки до и после растрирования оставался неизменным. Перед объединением матриц яркостей скрываемого изображения $G(x, y, z)$ и его инвертированной копии $G_{inv}(x, y, z)$ выполним растрирование этих изображений, позволяющее обеспечить неразличимости соседних пикселей.

Предполагается, что минимальным размером растровой ячейки, позволяющим обеспечить структурное различие является 2×1 или 1×2, объединенных в группы размером 2×2. Интенсивность одного из двух пикселей, при растрировании, приравнивается к средней по растровой ячейке, интенсивность другого пикселя растровой ячейки приравнивается 0. Схематично процесс растрирования описанным способом представлен на рисунке 2.5, уравнение растрирования представлено формулами 2.11 – 2.12.

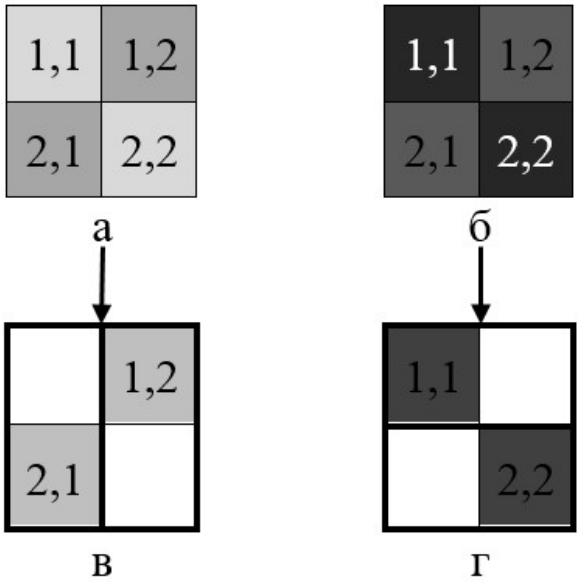
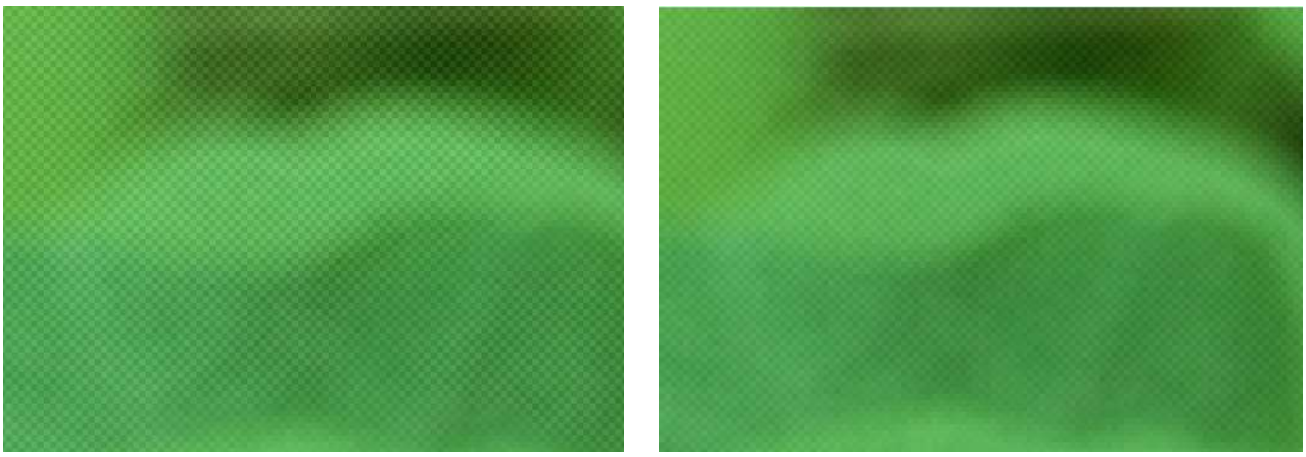


Рисунок 2.5 – Исходные группы растровых ячеек –а, скрываемого и инвертированного изображений – б; растрированные различными методами группы растровых ячеек – в, г.

$$\begin{cases} R_1(x_i, y_j, z_k) = F_{R1}\left(\frac{1}{2}\left(I(x_i, y_j, z_k) + I(x_{i+1}, y_j, z_k)\right)\right), \\ R_1(x_{i+1}, y_{j+1}, z_k) = F_{R1}\left(\frac{1}{2}\left(I(x_i, y_{j+1}, z_k) + I(x_{i+1}, y_{j+1}, z_k)\right)\right), \end{cases}\Bigg|_{\substack{i=1,3,\dots,N \\ j=1,3,\dots,M \\ k=1,2,3}} \quad (2.11)$$

$$\begin{cases} R_2(x_i, y_{j+1}, z_k) = F_{R2}\left(\frac{1}{2}\left(I(x_i, y_j, z_k) + I(x_i, y_{j+1}, z_k)\right)\right), \\ R_2(x_{i+1}, y_j, z_k) = F_{R2}\left(\frac{1}{2}\left(I(x_{i+1}, y_j, z_k) + I(x_{i+1}, y_{j+1}, z_k)\right)\right), \end{cases}\Bigg|_{\substack{i=1,3,\dots,N \\ j=1,3,\dots,M \\ k=1,2,3}} \quad (2.12)$$

Полученное изображение (рисунок 2.6а) обладает слаборазличимой структурой по сравнению с изображением, полученным обычным способом (рисунок 2.6 б), при этом выявленное изображение незначительно теряет качество с полным сохранением сюжетного вида.



а *б*

Рисунок 2.6 – Латентные изображения, полученные с растровыми структурами: *а* – без учета соседних пикселей , *б* – с учетом соседних пикселей

Разработанные методы моделирования латентных изображений позволяют получить модели формирования латентных изображений не только за счет комбинации различных растровых структур, образующих основное изображение, области которых определяются на основе взаимно обратных позитивной и негативной масок внедряемого изображения (черно-белого или цветного

скрываемого изображения), как в известных методах [21, 37], но и за счет комбинации двух низкочастотных квазипериодических структур, что позволило улучшить визуальное качество формируемых латентных изображений.

2.2 Математические модели формирования латентных изображений

На основе предложенных в п. 2.1 методов моделирования разработаны модели.

2.2.1 Математическая модель формирования латентного изображения, с внедрением цветного изображения

На основе предложенного метода моделирования построена математическая модель формирования латентного изображения с внедрением цветного изображения.

В известную математическую модель (формула 1.16) добавлены уравнения растрирования с удалением диагональных линий (формулы 2.2 – 2.3), что позволило исключить из известной модели яркостные преобразования (формула 1.14), введенные для скрытия факта внедрения, но приводящие к существенным искажениям изображения.

Для получения инвертированной копии скрываемого изображения $G_{inv}(x, y, z)$ интенсивность каждого пикселя изображений в координатах x, y, z вычисляется как разница между максимумом диапазона интенсивностей изображения и соответствующего пикселя скрываемого изображения:

$$G_{inv}(x, y, z) = (2^L - 1) - G(x, y, z) \tag{2.13}$$

Для получения первой квазипериодической структуры $G_1(x, y, z)$ скрываемых данных из скрываемого изображения $G(x, y, z)$ удаляются четные диагональные линии, значение интенсивности каждого пикселя нечетных диагональных линий равно среднему значению интенсивности соответствующего пикселя скрываемого изображения и соседнего пикселя в той же строке:

$$\begin{cases} R_1(x_i, y_j, z_k) = \frac{1}{2}\Big(G(x_i, y_j, z_k) + G(x_{i+1}, y_j, z_k)\Big), \\ R_1(x_{i+1}, y_{j+1}, z_k) = \frac{1}{2}\Big(G(x_i, y_{j+1}, z_k) + G(x_{i+1}, y_{j+1}, z_k)\Big), \end{cases}\Bigg|_{\substack{i=1,3,\dots,N \\ j=1,3,\dots,M \\ k=1,2,3}} \tag{2.14}$$

Для получения второй квазипериодической структуры $R_2(x, y, z)$ скрываемых данных из инвертированной копии скрываемого изображения удаляются нечетные диагональные линии, значение интенсивности каждого пикселя четных диагональных линий равно среднему значению интенсивности соответствующего пикселя скрываемого изображения и соседнего пикселя в том же столбце:

$$\begin{cases} R_2(x_i, y_{j+1}, z_k) = \frac{1}{2}\Big(G_{inv}(x_i, y_j, z_k) + G_{inv}(x_i, y_{j+1}, z_k)\Big), \\ R_2(x_{i+1}, y_j, z_k) = \frac{1}{2}\Big(G_{inv}(x_{i+1}, y_j, z_k) + G_{inv}(x_{i+1}, y_{j+1}, z_k)\Big), \end{cases}\Bigg|_{\substack{i=1,3,\dots,N \\ j=1,3,\dots,M \\ k=1,2,3}} \tag{2.15}$$

Для получения подготовленного скрываемого изображения квазипериодические структур $R_1(x, y, z)$ и $R_2(x, y, z)$ объединяются путем сложения интенсивности каждого пикселя изображений в координатах x, y, z:

$$H(x, y, z) = R_1(x, y, z) + R_2(x, y, z) \tag{2.16}$$

Для внедрения скрываемых данных исходное изображение $I(x, y, z)$ объединяется с подготовленным скрываемым изображением $H(x,y,z)$ путем сложения интенсивностей каждого пикселя изображений в координатах x, y, z с коэффициентами 1 и 0,1 соответственно:

$$L(x, y, z) = I(x, y, z) + 0.1H(x, y, z) \tag{2.17}$$

Полученная математическая модель представлена системой уравнений 2.6.

Значение меры ПОСШ для $L(x_i, y_j, z_k)$ относительно $I(x_i, y_j, z_k)$ для известной и разработанной моделей вычислялось по формуле 2.19. Значение меры ПОСШ для $L(x_i, y_j, z_k)$ относительно $I(x_i, y_j, z_k)$ для известной и разработанной моделей вычислялось по формуле 2.20.

Проверка адекватности разработанной модели осуществлялась путем решения задачи (1.27).

$$\begin{cases} G_{inv}(x,y,z) = (2^L - 1) - G(x,y,z) \\ \begin{cases} R_1\left(x_i, y_j, z_k\right) = \frac{1}{2}\left(G\left(x_i, y_j, z_k\right) + G\left(x_{i+1}, y_j, z_k\right)\right), \\ R_1\left(x_{i+1}, y_{j+1}, z_k\right) = \frac{1}{2}\left(G\left(x_i, y_{j+1}, z_k\right) + G\left(x_{i+1}, y_{j+1}, z_k\right)\right), \end{cases} \Bigg|_{\substack{i=1,3,\dots,N \\ j=1,3,\dots,M \\ k=1,2,3}} \\ \begin{cases} R_2\left(x_i, y_{j+1}, z_k\right) = \frac{1}{2}\left(G_{inv}\left(x_i, y_j, z_k\right) + G_{inv}\left(x_i, y_{j+1}, z_k\right)\right), \\ R_2\left(x_{i+1}, y_j, z_k\right) = \frac{1}{2}\left(G_{inv}\left(x_{i+1}, y_j, z_k\right) + G_{inv}\left(x_{i+1}, y_{j+1}, z_k\right)\right), \end{cases} \Bigg|_{\substack{i=1,3,\dots,N \\ j=1,3,\dots,M \\ k=1,2,3}} \\ H(x,y,z) = R_1(x,y,z) + R_2(x,y,z) \\ L(x,y,z) = I(x,y,z) + 0.1H(x,y,z), \end{cases} \tag{2.18}$$

где $G(x,y,z)$ – скрываемое изображение;

$G_{inv}(x,y,z)$ – инвертированная копия скрываемого изображения;

L – число разрядов квантования;

$R_1(x,y,z)$ и $R_2(x,y,z)$ – подготовленные на основе скрываемого изображения низкочастотные квазипериодические структуры;

$H(x,y,z)$ – подготовленное внедряемое изображение;

$L(x,y,z)$ – латентное изображение;

x, y, z – координаты цветовых компонент в системе RGB;

i,j – координаты пикселя над которым производится преобразование;

k – цветовая компонента в системе RGB. $k = 1$ – красная, $k = 2$ – зеленая, $k = 3$ – синяя;

N – ширина изображения в пикселях;

M – высота изображения в пикселях.

$$10log_2 \frac{N255^2}{\sum_{k=1}^{3}\sum_{i=1}^{N}\sum_{j=0}^{M}(L\left(x_i, y_j, z_k\right) - I\left(x_i, y_j, z_k\right))^2}, \tag{2.19}$$

$$10log_2 \frac{N255^2}{\sum_{k=1}^{3}\sum_{i=1}^{N}\sum_{j=0}^{M}(G'\left(x_i, y_j, z_k\right) - G\left(x_i, y_j, z_k\right))^2}, \tag{2.20}$$

где $L(x, y, z)$ – латентное изображение;

$I(x, y, z)$ – исходное изображение;

$G(x, y, z)$ – скрываемое изображение;

$G'(x, y, z)$– выявленное скрытое изображение;

x, y, z – координаты цветовых компонент в системе RGB;

i, j – координаты пикселя над которым производится преобразование;

k – цветовая компонента в системе RGB. $k = 1$ – красная, $k = 2$ – зеленая, $k = 3$ – синяя;

N – ширина изображения в пикселях;

M – высота изображения в пикселях.

Значение ПОСШ$\left(L(x_i, y_j, z_k), I(x_i, y_j, z_k)\right)$ для разработанной модели в среднем составляет 89 против 82 для известной. Значение ПОСШ$\left(G'(x_i, y_j, z_k), G(x_i, y_j, z_k)\right)$ в среднем составляет 37 против 31 для известной. Таким образом, полученный результат соответствует формальной постановке задачи (1.27).

2.2.2 Математическая модель формирования латентного изображения, с внедрением черно-белого изображения

На основе предложенного метода моделирования построена математическая модель формирования латентного изображения с внедрением цветного изображения.

Разработанная математическая модель формирования ЛИ, с внедрением черно-белого изображения, отличающиеся от известного метода [21] заменой формул растрирования 1.20 – 1.21 на формулы 2.6 – 2.7. Внедрение скрываемого изображения при этом происходит во все цветовые компоненты без потери визуального качества, что позволило улучшить качество выявляемого изображения.

Для получения первой квазипериодической структуры $R_1(x, y, z)$ скрываемых данных из копии исходного изображения $I(x, y, z)$ удаляются четные

диагональные линии, значение интенсивности каждого пикселя нечетных диагональных линий равно среднему значению интенсивности соответствующего пикселя копии исходного изображения и соседнего пикселя в той же строке:

$$\begin{cases} R_1(x_i, y_j, z_k) = \frac{1}{2}\Big(I(x_i, y_j, z_k) + I(x_{i+1}, y_j, z_k)\Big), \\ R_1(x_{i+1}, y_{j+1}, z_k) = \frac{1}{2}\Big(I(x_i, y_{j+1}, z_k) + I(x_{i+1}, y_{j+1}, z_k)\Big), \end{cases}\Bigg|_{\substack{i=1,3,\dots,N \\ j=1,3,\dots,M \\ k=1,2,3}} \tag{2.21}$$

Для получения второй квазипериодической структуры $R_2(x, y, z)$ скрываемых данных из копии исходного изображения удаляются нечетные диагональные линии, значение интенсивности каждого пикселя четных диагональных линий равно среднему значению интенсивности соответствующего пикселя копии исходного изображения и соседнего пикселя в том же столбце:

$$\begin{cases} R_2(x_i, y_{j+1}, z_k) = \frac{1}{2}\Big(I(x_i, y_j, z_k) + I(x_i, y_{j+1}, z_k)\Big), \\ R_2(x_{i+1}, y_j, z_k) = \frac{1}{2}\Big(I(x_{i+1}, y_j, z_k) + I(x_{i+1}, y_{j+1}, z_k)\Big), \end{cases}\Bigg|_{\substack{i=1,3,\dots,N \\ j=1,3,\dots,M \\ k=1,2,3}} \tag{2.22}$$

Для получения подготовленного скрываемого изображения квазипериодическая структура $R_1(x, y, z)$ вычитается по маске черно-белого скрываемого изображения $G(x, y)$ и полученное изображение объединяется со второй квазипериодической структурой $R_2(x, y, z)$ путем сложения интенсивности каждого пикселя изображений в координатах x, y, z:

$$H(x, y, z) = G(x, y) * R_1(x, y, z) + R_2(x, y, z) \tag{2.23}$$

Для внедрения скрываемых данных исходное изображение $I(x, y, z)$ объединяется с подготовленным скрываемым изображением $H(x, y, z)$ путем сложения интенсивностей каждого пикселя изображений в координатах x, y, z с коэффициентами 1 и 0,1 соответственно:

$$L(x, y, z) = I(x, y, z) + 0.1H(x, y, z) \tag{2.24}$$

Полученная математическая модель представлена системой уравнений 2.25.

$$\begin{cases} \begin{cases} R_1\left(x_i, y_j, z_k\right) = \frac{1}{2}\left(I\left(x_i, y_j, z_k\right) + I\left(x_{i+1}, y_j, z_k\right)\right), \\ R_1\left(x_{i+1}, y_{j+1}, z_k\right) = \frac{1}{2}\left(I\left(x_i, y_{j+1}, z_k\right) + I\left(x_{i+1}, y_{j+1}, z_k\right)\right), \end{cases}\Bigg|_{\substack{i=1,3,\ldots,N \\ j=1,3,\ldots,M \\ k=1,2,3}} \\ \begin{cases} R_2\left(x_i, y_{j+1}, z_k\right) = \frac{1}{2}\left(I\left(x_i, y_j, z_k\right) + I\left(x_i, y_{j+1}, z_k\right)\right), \\ R_2\left(x_{i+1}, y_j, z_k\right) = \frac{1}{2}\left(I\left(x_{i+1}, y_j, z_k\right) + I\left(x_{i+1}, y_{j+1}, z_k\right)\right), \end{cases}\Bigg|_{\substack{i=1,3,\ldots,N \\ j=1,3,\ldots,M \\ k=1,2,3}} \\ H(x,y,z) = G(x,y) * R_1(x,y,z) + R_2(x,y,z), \\ L(x,y,z) = I(x,y,z) + 0.1H(x,y,z), \end{cases} \tag{2.25}$$

где $G(x, y)$ – скрываемое черно-белое изображение;

$R_1(x, y, z)$ и $R_2(x, y, z)$ – подготовленные на основе исходного изображения низкочастотные квазипериодические структуры;

$H(x, y, z) -$ подготовленное внедряемое изображение;

$L(x, y, z) -$ латентное изображение;

x, y, z – координаты цветовых компонент в системе RGB;

i,j – координаты пикселя над которым производится преобразование;

k – цветовая компонента в системе RGB. k=1 – красная, k=2 – зеленая,

$k = 3$ – синяя;

N – ширина изображения в пикселях;

M – высота изображения в пикселях.

Проверка адекватности разработанной модели осуществлялась аналогичным способом, описанным в п. 2.2.1 для математической модели формирования латентного изображения, с внедрением цветного изображения.

2.3 Проверка разработанных моделей формирования латентных изображений

Проведено сравнение визуального качества латентных изображений, сформированных на основе разработанных моделей, с изображениями, полученными на основе известных методов. Для этого были разработаны модели

в системе MATLAB, позволяющие автоматически формировать латентное изображение методом, предложенным в работе [37] и разработанным методом.

2.3.1 Моделирование латентного изображения известным методом

Для оценки результатов было сформировано латентное изображение известным методом, позволяющим внедрять как черно-белые, так и цветные скрытые изображения [37].

Обобщенный алгоритм формирования данным способом сводится к следующему:

1) скрываемое изображение дублируется;
2) производится инвертирование скрываемого изображения;
3) дублированное скрываемое изображение растрируется заранее подготовленными растровыми структурами;
4) инвертированное скрываемое и растрированное скрываемое изображения объединяются.
5) над полученным подготовленным скрываемым изображением производятся яркостные преобразования;
6) основное и полученное подготовленное скрываемое изображение объединяются, образуя латентное изображение.

К недостаткам данного способа относится то, что технологический процесс предложенного способа весьма трудоемок и требует специального дорогостоящего программного обеспечения, а именно Adobe Photoshop. В связи с этим потоковое внедрение скрытых изображений в основные невозможно.

В рамках диссертационного исследования разработана модель на языке MATLAB [73 – 81], позволяющая автоматически формировать латентное изображение способом, предложенным в работе [37] на основе исходного и скрываемого изображений. Предполагаемое данным способом растрирование осуществляется на основе растровой ячейки $n*n$ загружаемого в программу в виде бинарного изображения, например, как показано на рисунке 2.7.

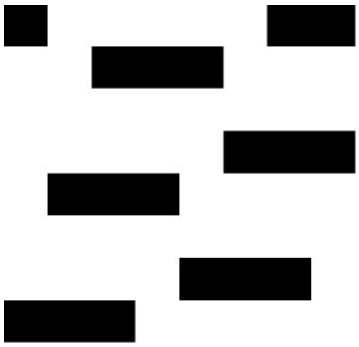


Рисунок 2.7 – Растровая ячейка 8*8

Файлы исходного изображения с именем Base.jpg, скрываемого Concealed.jpg и растровой ячейки Rastr.bmp необходимо поместить в корень диска С.

Загрузка требуемых изображений производится следующими командами:

```
I=imread('C:\Base.jpg');
G=imread('C:\Concealed.jpg');
RCell=imread('C:\Rastr.bmp');
```

После загрузки скрываемое изображение дублируется и инвертируется следующими командами:

```
GCopy=G;
G=255-G;
```

Для создания растровой структуры можно воспользоваться функцией *repmat*(), создающей матрицу из заданного количества копий исходной матрицы. Для создания растровой структуры соразмерной с исходным изображением используем следующие команды:

```
RastrT=repmat(RCeil,ceil(size(G,1)/size(RCell,1)),ceil(size(G,2)/siz
e(RCell,2)));
RastrT(:,(size(G,2)+1):(size(RastrT,2)))=[1];
RastrT((size(G,1)+1):(size(RastrT,1)),:)=[1];
```

Полученная растровая структура представляет собой маску, по которой производится растрирование инвертированного скрытого изображения следующими командами:

```
G(:,:,1)=G(:,:,1).*(uint8(RastrT));
G(:,:,2)=G(:,:,2).*(uint8(RastrT));
G(:,:,3)=G(:,:,3).*(uint8(RastrT));
```

Далее получим негатив маски растровой структуры и наложим ее на копию скрываемого изображения и объединим полученное изображение с растрированным инвертированным скрытым изображением с помощью следующих команд:

```
RastrTInv=1-RastrT;
GCopy(:,:,1)=GCopy(:,:,1).*(uint8(RastrTInv));
GCopy(:,:,2)=GCopy(:,:,2).*(uint8(RastrTInv));
GCopy(:,:,3)=GCopy(:,:,3).*(uint8(RastrTInv));
G=G+GCopy;
```

Над подготовленном для внедрения скрытым изображением проведем яркостные преобразования:

```
G = imadjust (G, [10.4 0.5], [10.4 0.5]);
```

Для непосредственного внедрения скрытого изображения воспользуемся командой imlincomb, с коэффициентом для скрываемого изображения 0.1, что соответствует прозрачности 10%, задаваемой программе Adobe Photoshop:

```
L=imlincomb (0.1, G, 0.9, I);
```

В завершении программа записывает полученное латентное изображение в файл Latent.jpg с помощью следующей команды:

```
imwrite(L,'C:\Latent.jpg','quality',100);
```

Рисунок 2.8 – Латентное изображение, сформированное известным способом

Практическая реализация метода, описанного в работе [37] предлагается автором в системе Adobe Photoshop, что делает проверку данного метода

трудоемким. Получаемое латентное изображение (рисунок 2.8) с помощью предложенной модели идентично изображению, получаемому с помощью Adobe Photoshop, и показывает, согласно формул 2.19 и 2.20, аналогичные результаты при выявлении скрытого изображения.

2.3.2 Моделирование латентного изображения предложенными методами

В рамках диссертационного исследования разработаны модели на языке MATLAB, позволяющие автоматически формировать латентное изображение предложенными методами на основе исходного и скрываемого изображений.

Файлы исходного изображения с именем Base.jpg, скрываемого Concealed.jpg необходимо поместить в корень диска C.

Загрузка требуемых изображений производится также, как и при моделировании известного способа.

После загрузки скрываемое изображение инвертируется следующей командой:

```
GInv=255-G;
```

Для создания для создания квазипериодических структур первого и второго типа будем использовать следующие функции:

```
function R1 = PsevdoR(Orig)
% Функция растрирования PsevdoR
 for i=1:2:(ceil(size(Orig,1)/2))*2 %Y
 for j=1:2:size(Orig,2) %X
 for k=1:1:3
 z(i,j,k)=(Orig(i,j,k)+Orig(i+1,j,k))/2;
 z(i+1,j+1,k)=(Orig(i,j+1,k)+Orig(i+1,j+1,k))/2;
 end;
 end;
 end;
end

function R2 = PsevdoR2(Orig)
%Функция растрирования PsevdoR2
 for i=1:2:(ceil(size(Orig,1)/2))*2 %Y
 for j=1:2:size(Orig,2) %X
 for k=1:1:3
 z(i,j+1,k)=(Orig(i,j,k)+Orig(i,j+1,k))/2;
 z(i+1,j,k)=(Orig(i+1,j,k)+Orig(i+1,j+1,k))/2;
 end;
```

```
 end;
 end;
end
```

Для формирования подготовленного скрываемого цветного изображения необходимо выполнить следующий набор команд:

```
R1 = PsevdoR(G);
R2 = PsevdoR(GInv);
H = R1+ R2;
```

Для формирования подготовленного скрываемого черно-белого изображения необходимо выполнить следующий набор команд:

```
R1 = PsevdoR(I);
R2 = PsevdoR(I);
H = R1.*G + R2;
```

Для непосредственного внедрения скрытого изображения воспользуемся командой imlincomb, с коэффициентом для скрываемого изображения 0.1:

```
L=imlincomb (0.1, H, 1, I);
```

В завершении программа записывает полученное латентное изображение в файл Latent.jpg с помощью следующей команды:

```
imwrite(L,'C:\Latent.jpg','quality',100);
```

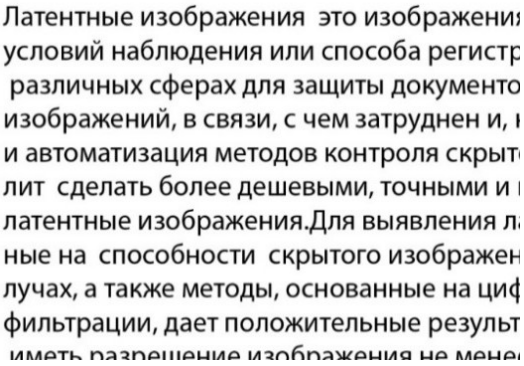
2.3.3 Проведение испытаний и анализ результатов

Для испытаний в окне программного модуля формирования латентных изображений выбраны пути к исходным и внедряемым изображениям последовательно для каждой пары изображений: *а*, *в* и *б, г* (рисунок 2.9) соответственно и сохранены полученные латентные изображения. На рисунке 2.10 показаны пары исходных и латентных изображений. Как видно на латентных изображениях полностью сохранена сюжетная структура исходных изображений, видимых признаков наличия скрытого изображения нет.

а

б

в



г

Рисунок 2.9 – Набор изображений для тестирования (а, б – исходные; в, г – скрываемые)

а

б

в

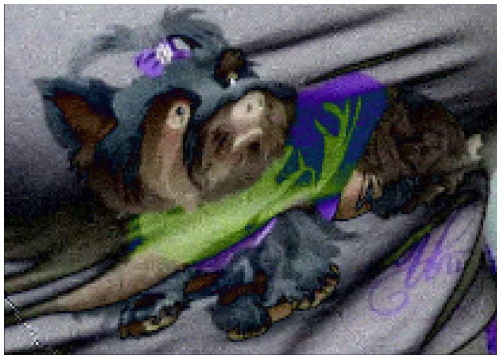
г

Рисунок 2.10 – Набор изображений для визуального сравнения (а, б – исходные; в, г – латентные)

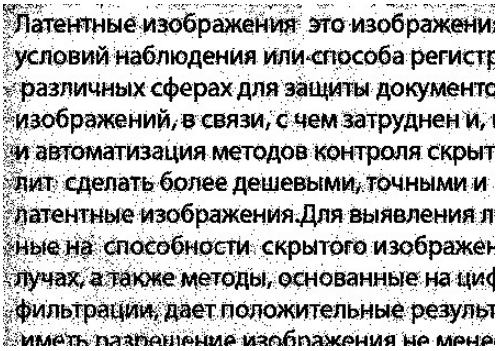
Полученные латентные изображения были проанализированы разработанным методом выявления и сохранены полученные скрытые изображения. На рисунке 2.11 показаны пары скрываемых и выявленных изображений. Как видно, выявленные изображения полностью сохранили структуру скрытых изображений и, в случае с текстом в качестве скрываемого изображения, полностью пригодно для чтения.

а

б



в



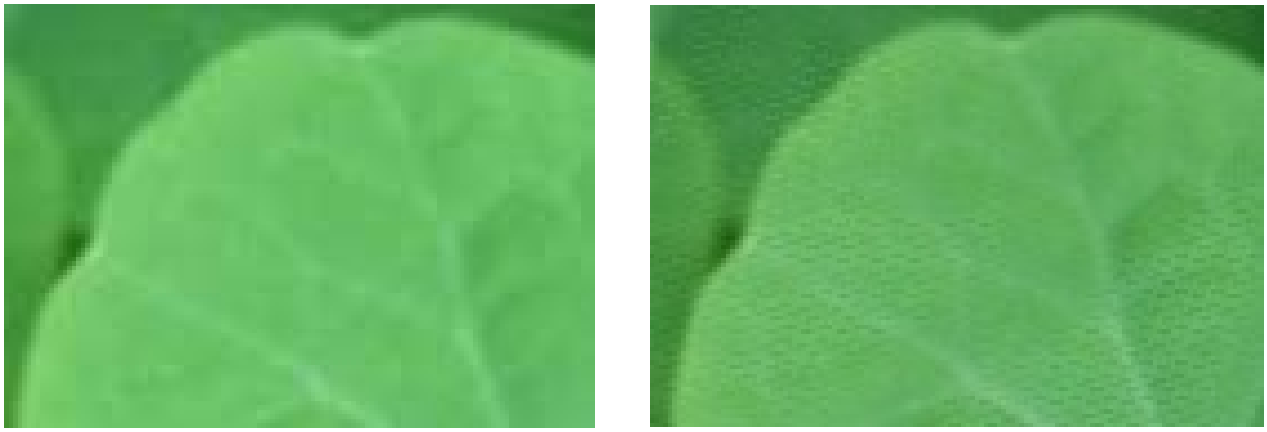
г

Рисунок 2.11 – Набор изображений для визуального сравнения 2 (а, б – скрываемые; в, г – выявленные)

Изображение, полученное комбинированным способом, сравнивалось с изображением, полученным известным способом [37]. Качество изображений оценивалось визуально и путем определения меры пикового отношения сигнал/шум (формула 1.3).

Визуально изображения, полученные с использованием комбинированного способа, отличаются от изображения, сформированного с помощью известного способа более высоким качеством, так как, при увеличении изображений, у

изображения, полученного известным способом, явно проявляется растровая структура (рисунок 2.12).



а *б*

Рисунок 2.12 – Визуальное сравнение изображений, сформированных: комбинированным (*а*) и известным способом (*б*)

Как видно из рисунка 2.12, выявленные изображения полностью сохранили структуру скрытых изображений, в случае с текстом, в качестве скрываемого изображения, он полностью пригоден для чтения. Количественной характеристикой сравнения с известными способами являлась мера пикового отношения сигнал/шум (формула 1.3).

Латентное изображение считается визуально качественным, если ПОСШ ≥ 28 дБ. При этом, чем выше данное значение, тем, считается, выше качество изображения.

Значение ПОСШ для сформированных с использованием разработанных методов в среднем составляет 89Дб, для изображений, полученных методом [37] – в среднем 82Дб. Таким образом, разработанные методы формирования позволили повысить качество сформированного латентного изображения в среднем на 8,5%

Полученные латентные изображения, также были проанализированы разработанным методом выявления скрытых изображений, описанным в главе 3.

На основе полученных моделей разработаны алгоритмы и программный модуль для формирования латентных изображения, приведенные в 4 главе.

2.4 Выводы

Разработаны метод моделирования и математическая модель формирования ЛИ, с внедрением цветного изображения, на основе модификации известного метода [37], заключающейся в использовании как частотных, так и пространственных преобразований исходного изображения.

В известную математическую модель добавлены уравнения растрирования с удалением диагональных линий, что позволило исключить из известной модели яркостные преобразования, введенные для скрытия факта внедрения, но приводящие к существенным искажениям изображения.

Разработаны метод моделирования и математическая модель формирования ЛИ, с внедрением черно-белого изображения, отличающиеся от известного метода [21] заменой формул растрирования на формулы, позволяющие удалять диагональные линии. Внедрение скрываемого изображения при этом происходит во все цветовые компоненты без потери визуального качества, что позволило улучшить качество выявляемого изображения.

Проведена проверка адекватности разработанных моделей согласно условию задачи, описанной формулой 1.27, на основе сравнения данных, полученных в результате вычислительного эксперимента. Для проверки адекватности разработанных методов и моделей были сформированы латентные изображения известным и предложенным методами, для чего были разработаны программы в среде Matlab.

Качество сформированного латентного изображения, с использованием разработанных моделей, по сравнению с известными, повысилось в среднем на 8,5%.

ГЛАВА 3 РАЗРАБОТКА МАТЕМАТИЧЕСКИХ МЕТОДОВ И АЛГОРИТМОВ КОНТРОЛЯ ЦИФРОВЫХ ЛАТЕНТНЫХ ИЗОБРАЖЕНИЙ

Разработан численный метод обнаружения скрытой информации, реализующий вейвлет-преобразования латентного изображения и позволяющий, в отличие от известных методов, выявить скрытое изображение вне зависимости от конкретной модели формирования, на основе предложенной математической модели выявления скрытой информации в латентных изображениях. Предложенная математическая модель позволила сократить вычислительную сложность выявления скрытой информации, что позволило автоматизировать контроль латентного изображения, вне зависимости от способа внедрения скрытой информации.

На основе предложенных методов разработана система формирования и контроля латентных изображений, позволяющая, представить процессы от момента формирования до определения подлинности защищенного документа, отличающаяся от известных тем, что предусмотрен контроль латентных изображений, как при известном, так и при неизвестном методах внедрения скрываемых изображений.

Для выявления латентных изображений используются методы, основанные на Фурье-анализе, вейвлет-анализе, методах фильтрации изображений, а также методы преобразования изображения, основанные на способах внедрения скрытых изображений [83, 84].

К существенным недостаткам первых двух методов можно отнести сложность интерпретации результатов анализа, для первых трех – невозможность отделить скрытое изображение от фонового при совпадающем периоде образующих их текстур. Методы, основанные на способах внедрения, подходят только для решения частных задач контроля. Особенностью всех методов выявления, за исключением методов вейвлет-анализа, является необходимость построения индивидуального фильтра для визуализации латентного изображения.

Для разработки метода контроля, который не будет зависеть от конкретного способа формирования латентных изображений, сформированных различно ориентированными структурами или образованных структурами с совпадающим периодом фонового и скрытого изображения, предлагается произвести сегментацию изображения, используя вейвлет-анализ.

Задачу контроля всякого изобразительного образа полученного с устройства захвата и прошедшего предварительную обработку, включая распознавание скрытого изображения в латентном, можно разделить на следующие подзадачи: сегментация, описание, распознавание, интерпретация. Под сегментацией подразумевается процесс выделения на изображении интересующих объектов, т.е. разбиение поля зрения D на области объектов $D_1,\ldots,D_s$ и область фона D_f. Описание распознаваемых объектов предполагает определение характерных параметров необходимых для выделения требуемого объекта на фоне других, т.е. определение границ классов. Распознавание – идентификация объектов. Интерпретация определяет принадлежность к группе распознаваемых объектов [82].

На рисунке 3.1 представлена совокупность объектов, на которые сегментируется изображение.

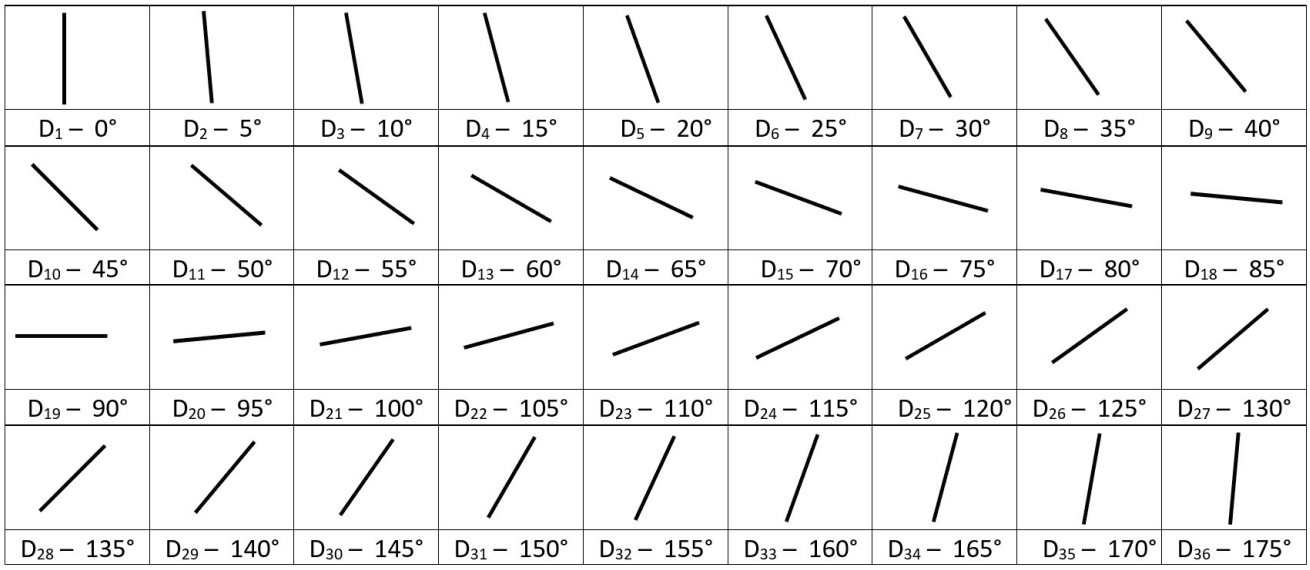


Рисунок 3.1 – Совокупность объектов, подлежащих сегментации

Совокупность объектов $D_1,\ldots,D_s$ на которые необходимо разбить латентное изображение, полученное методом вариации направления линий, представлена линиями различного наклона. Учитывая, что при получении изображения с устройства захвата возможны существенные геометрические деформации изображения, экспериментально установлено, что достаточным шагом вариации наклона линии является 5°.

Представленная на рисунке 3.1 совокупность объектов использовалась для проверки вейвлет-функций на возможность сегментации каждого из объектов.

3.1 Выбор оптимальной вейвлет-функции для выявления скрытого изображения

При вейвлет-разложении изображение $I(x,y)$, где x и y – координаты каждой точки по горизонтали и вертикали, представляется в виде совокупности аппроксимирующих и детализирующих коэффициентов с помощью масштабирующей функции $\varphi(x)$ и вейвлета $\psi(y)$.

Масштабирующая функция для двухмерного сигнала, определяющая аппроксимирующие коэффициенты изображения $I(x,y)$, задается следующим образом:

$$\varphi_j(x,y) = \varphi(x)\varphi(y) \tag{3.1}$$

Детализирующие коэффициенты, в свою очередь, можно представить в виде трех разделимых двухмерных вейвлетов, которые обеспечивают сегментацию изображения:

$$\psi_j^H(x,y) = \psi(x)\varphi(y), \psi_j^V(x,y) = \varphi(x)\psi(y), \psi_j^D(x,y) = \psi(x)\psi(y), \tag{3.2}$$

где $\psi_j^H(x,y)$, $\psi_j^V(x,y)$ и $\psi_j^D(x,y)$ – горизонтальные, вертикальные и диагональные вейвлеты, j - уровень декомпозиции.

Так как лучшую частотную локализацию сигнала можно получить путем декомпозиции вейвлетов (3.2), применяя пакетное вейвлет-преобразование [40], то предлагается производить разложение до третьего уровня декомпозиции, что позволит добиться требуемого уровня сегментации исследуемых объектов.

Для решения поставленной задачи необходимо подобрать вейвлет функцию с высокой локализацией, как по частоте, так и по времени, применение которой позволит определить группы объектов, с примерно равным числом элементов и минимальным числом элементов, одновременно входящих в разные группы. Для определения оптимальной вейвлет функции производилось вейвлет-преобразование контрольного изображения (рисунок 3.1), на основе различных базисных вейвлетов: Хаара (haar), Добеши (*db*), Симлета (*sym*), Коифлета (*coif*), биортогональных вейвлетов (*bior)*, обратно-биортогональных (*rbio*) и вейвлетов Мейера в дискретном варианте (*dmey*).

Для выбора оптимальной вейвлет-функции проанализированы коэффициенты вейвлет-разложения контрольного изображения и латентного, полученных с помощью различных базисных вейвлетов.

В качестве критериев качества вейвлет-функции будем считать различимость в визуализированных коэффициентах одних элементов при неразличимости других элементов изображения.

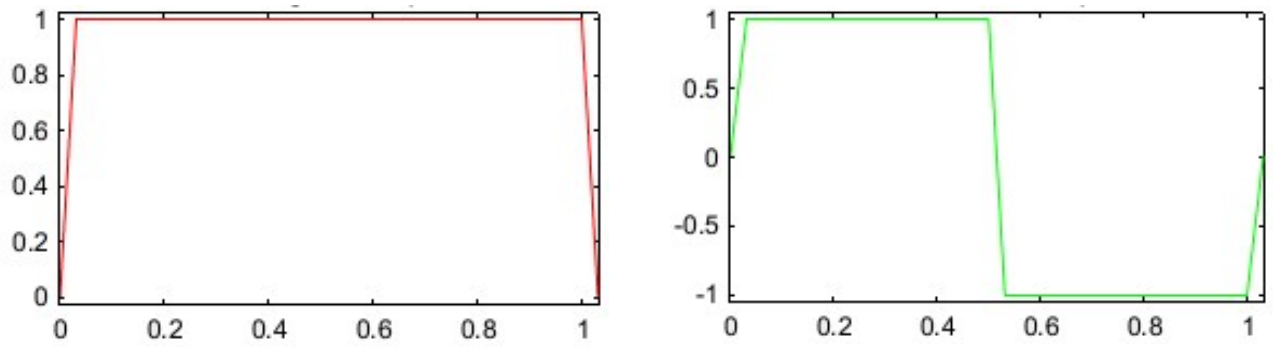
Вейвлет Хаара. Родительская вейвлет-функция $\psi(x)$ вейвлета Хаара с нулевым значением интеграла $\int_{-\infty}^{\infty}\psi(x)d(x) = 0$, определяющая детали сигнала, задается следующим образом:

$$\psi(x) = \begin{cases} 1, & x \in [0,1/2) \\ -1, & x \in [1/2,1) \\ 0\,, & x \notin [0,1) \end{cases} \tag{3.3}$$

Масштабирующая функция $\phi(x)$ с единичным значением интеграла $\int_{-\infty}^{\infty}\phi(x)d(x) = 1$, определяющая грубое приближение сигнала, постоянна:

$$\phi(x) = \begin{cases} 1, & x \in [0,1) \\ 0, & x \notin [0,1) \end{cases} \tag{3.4}$$

Таким образом, графики функций вейвлета Хаара задаются однозначно (рисунок 3.2)

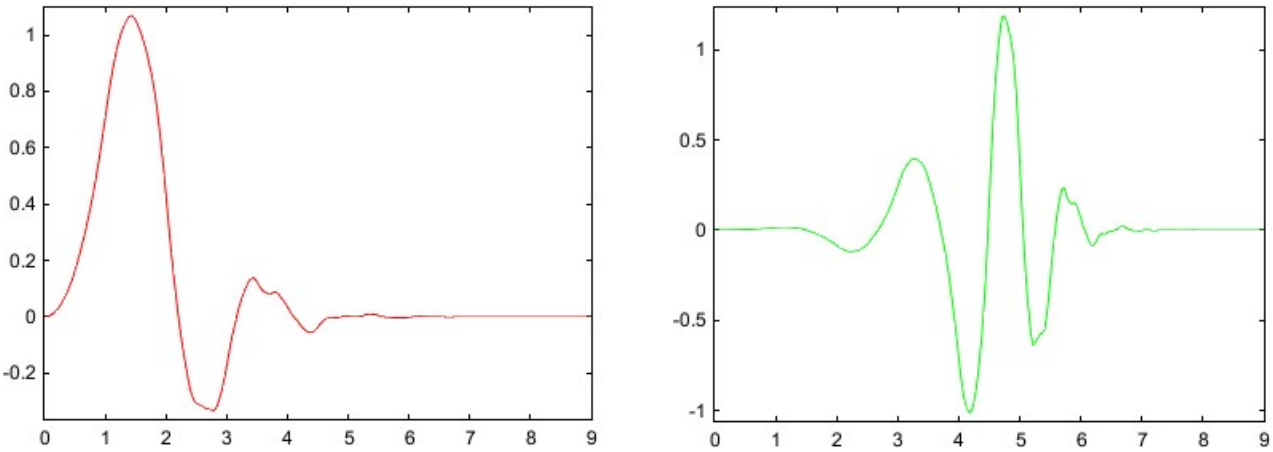


а б

Рисунок 3.2 – Графики функций Хаара: а) масштабирующая функция $\phi(x)$, б) вейвлет-функция $\psi(x)$

Вейвлеты Добеши. Семейство ортогональных вейвлетов с компактным носителем, вычисляемым итерационным путем. Наиболее распространенные фильтры вейвлетов Добеши, строятся с использованием пар коэффициентов в числе от 2 до 45. В ходе диссертационного исследования было установлено, что вейвлет Добеши с высокочастотным и низкочастотным фильтрами, построенными по 5 парам коэффициентов лучше всего подходит для отделения линий различной направленности.

Графически функции вейвлета Добеши построенные по 5 парам коэффициентов (db5) показаны на рисунке 3.3

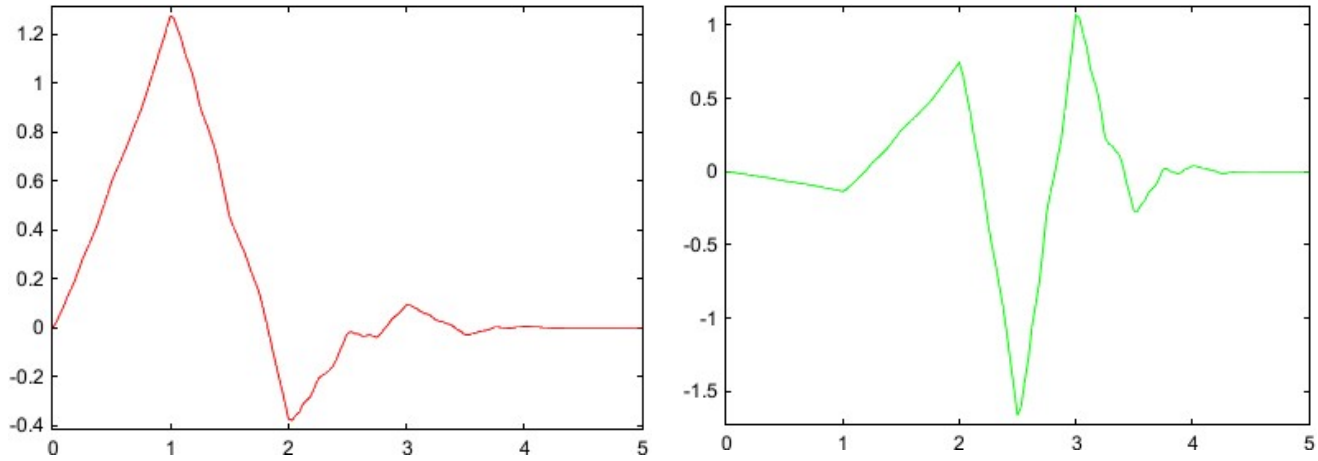


а б

Рисунок 3.3 – Графики функций Добеши 5: а) масштабирующая функция $\phi(x)$, б) вейвлет-функция $\psi(x)$

Симплеты. Симплет-вейвлеты являются частным случаем вейвлетов Добеши с наименьшей асимметрией и компактным носителем. Наиболее распространенные фильтры симплетов, строятся с использованием пар коэффициентов в числе от 2 до 45. Экспериментально было установлено, что симплет с высокочастотным и низкочастотным фильтрами, построенными по 3 парам коэффициентов лучше всего подходит для отделения линий различной направленности.

Графически функции симплета построенные по 3 парам коэффициентов (sym3) показаны на рисунке 3.4



а б

Рисунок 3.4 –Графики функций Симплета 3: а) масштабирующая функция $\phi(x)$, б) вейвлет-функция $\psi(x)$

Койфлеты. Койфлет-вейвлеты являются частным случаем вейвлетов Добеши, у которых нулевые моменты имеются не только у функции $\psi(x)$, но и $\phi(x)$, т.е. удовлетворяет следующим условиям:

$$\begin{cases} \int_{-\infty}^{\infty} x^k \psi(x) d(x) = 0 \Big|_{k=0,\dots,N-1}, \\ \int_{-\infty}^{\infty} \phi(x) d(x) = 1, \\ \int_{-\infty}^{\infty} x^k \phi(x) d(x) = 0 \Big|_{k=0,\dots,N-1} \end{cases} \tag{3.5}$$

Наиболее распространенные фильтры койфлетов, строятся с использованием пар коэффициентов в числе от 1 до 5. Экспериментально было

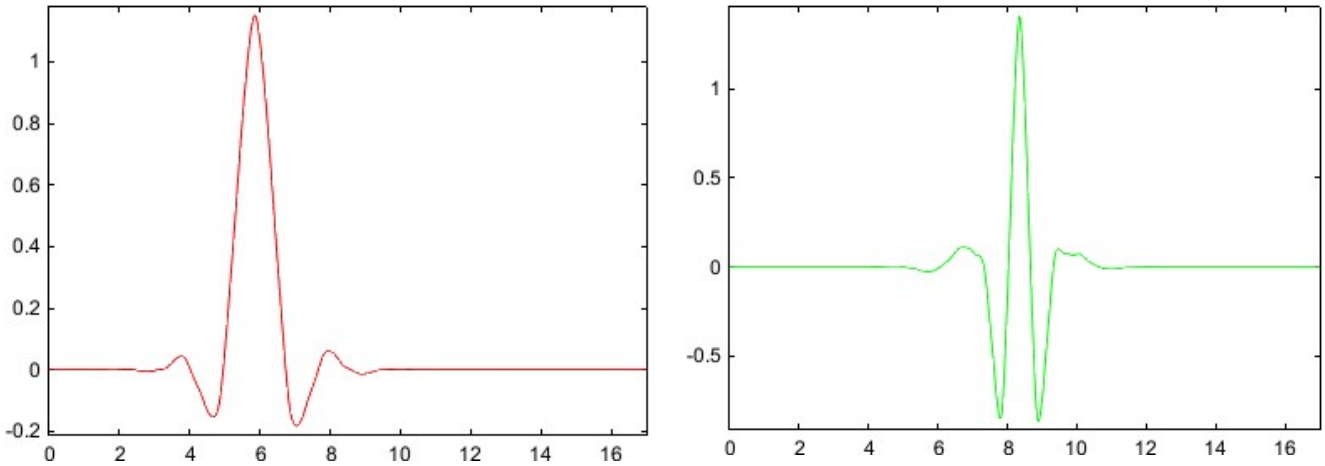
установлено, что койфлет с высокочастотным и низкочастотным фильтрами, построенными по 3 парам коэффициентов лучше всего подходит для отделения линий различной направленности.

Графически функции койфлета построенные по 3 парам коэффициентов (coif3) показаны на рисунке 3.5

Биортогональные вейвлеты. Биортогональные вейвлеты суть пары функций, по одной из которых ведется анализ сигнала, по другой - восстановление. Если один из биортогональных вейвлетов обладает гладкостью *r*, то парный ему вейвлет автоматически приобретает *r* нулевых моментов.

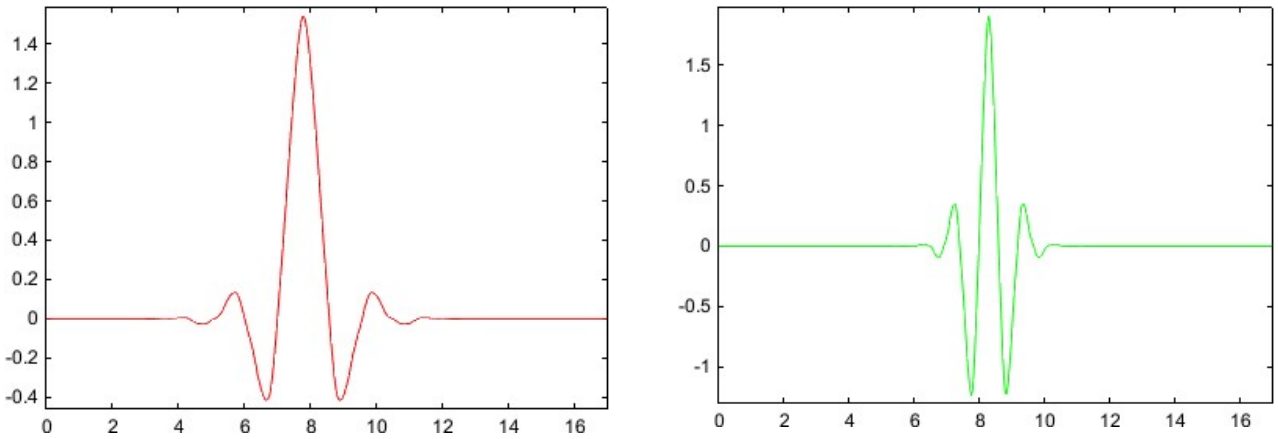
Фильтры биортогональных вейвлетов, строятся по параметрам 1.1, 1.3, 1.5, 2.2, 2.4, 2.6, 2.8, 3.1, 3.3, 3.5, 3.7, 3.9, 4.4, 5.5, 6.8. Экспериментально было установлено, что биортогональный вейвлет с высокочастотным и низкочастотным фильтрами, построенными по параметру 2.8 лучше всего подходит для отделения линий различной направленности.

Графически функции биортогонального вейвлета, построенные по параметрам 2.8 (bior2.8) показаны на рисунке 3.6.



а б

Рисунок 3.5 – Графики функций Койфлета 3: а) масштабирующая функция $\phi(x)$, б) вейвлет-функция $\psi(x)$

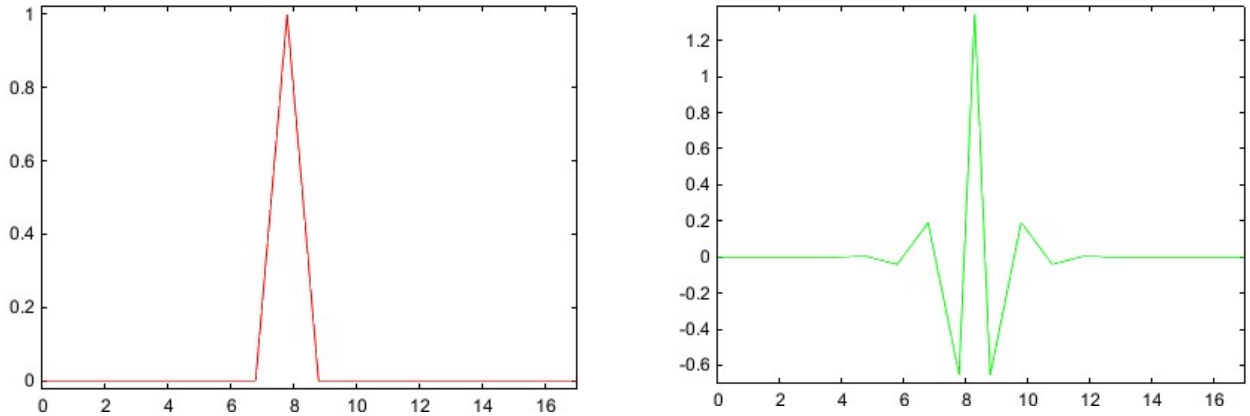


а б

Рисунок 3.6 – Графики функций биортогонального вейвлета 2.8: а) масштабирующая функция $\phi(x)$, б) вейвлет-функция $\psi(x)$

Обратные биортогональные вейвлеты. Обратные биортогональные вейвлеты строятся по аналогии с биортогональными вейвлетами, но пара функций, используемая в биортогональных вейвлетах для разложения сигнала используется для восстановления, а функции восстановления биортогональных вейвлетов – для разложения. Фильтры обратных биортогональных вейвлетов, строятся по параметрам 1.1, 1.3, 1.5, 2.2, 2.4, 2.6, 2.8, 3.1, 3.3, 3.5, 3.7, 3.9, 4.4, 5.5, 6.8.

Графически функции обратного биортогонального вейвлета построенные по параметрам 2.8 (rbio2.8) показаны на рисунке 3.7.

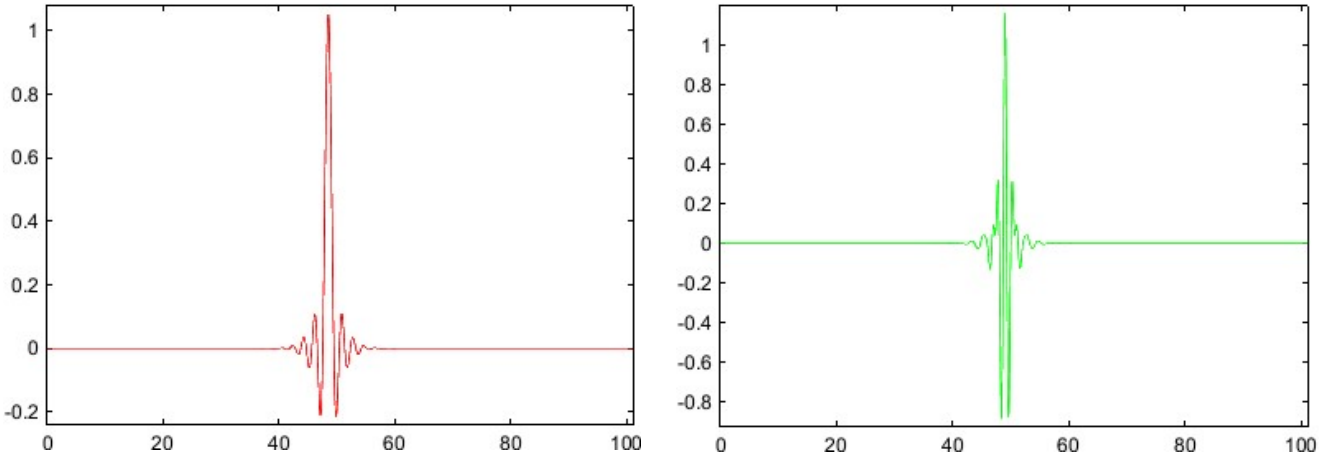


а б

Рисунок 3.7 – Графики функций обратного биортогонального вейвлета 2.8: а) масштабирующая функция $\phi(x)$, б) вейвлет-функция $\psi(x)$

Было установлено, что обратный биортогональный вейвлет с высокочастотным и низкочастотным фильтрами, построенными по параметру 2.8 лучше всего подходит для отделения линий различной направленности.

Дискретный вейвлет Майера. Графически функции дискретного вейвлета Майера показаны на рисунке 3.8



а б

Рисунок 3.8 – Графики функций вейвлета Майера: а) масштабирующая функция $\phi(x)$, б) вейвлет-функция $\psi(x)$

В сводной таблице 3.1 представлены диапазоны групп объектов контрольного изображения (рисунок 3.1) однозначно сегментированные (столбцы со знаком +) и полностью отфильтрованные (столбцы со знаком -) при вейвлет-преобразовании в различных детализирующих коэффициентов (горизонтальных, вертикальных и диагональных) каждой из исследуемых вейвлет-функций.

В таблице 3.2 представлены критерии оценки исследуемых вейвлет-функций для решения задачи сегментации контрольного изображения.

В результате проведенных вычислительных экспериментов установлено, что биортогональный вейвлет по базису 2.8 лучше всего подходят для решения данной задачи, так как позволяет сегментировать объекты на группы с примерно равным числом элементов и минимальным числом элементов, одновременно входящих в разные группы.

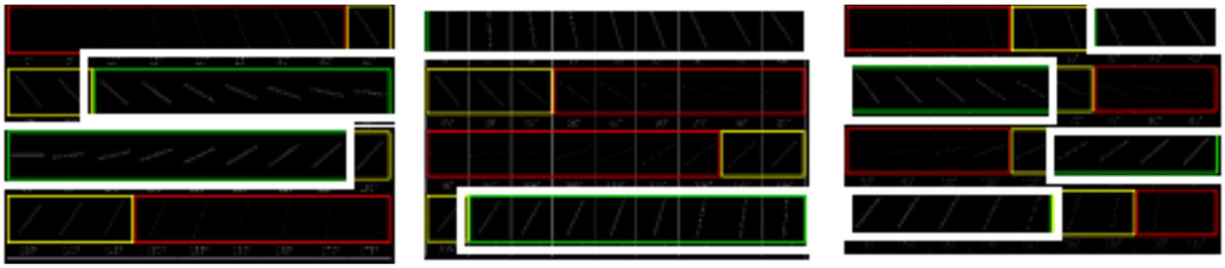
Таблица 3.1 – Группы сегментируемых объектов в детализирующих коэффициентах исследуемых вейвлет-функций

вейвлет-функция	Горизонтальные		вертикальные		диагональные	
	+	-	+	-	+	-
haar	$D_{11} – D_{18}$, $D_{20} – D_{27}$	$D_1 – D_3$, $D_{35} – D_{36}$	$D_1 – D_{13}$, $D_{25} – D_{36}$	$D_{17} – D_{21}$	$D_4 – D_{16}$, $D_{22} – D_{34}$	D_1, D_{19}
db5	$D_{11} – D_{27}$	$D_1 – D_9$, $D_{29} – D_{36}$	$D_1 – D_9$, $D_{29} – D_{36}$	$D_{12} – D_{26}$	$D_6 – D_{14}$, $D_{23} – D_{33}$	$D_1 – D_4$, $D_{16} – D_{20}$, D_{36}
sym3	$D_{11} – D_{27}$	$D_1 – D_9$, $D_{30} – D_{36}$	$D_1 – D_9$, $D_{28} – D_{36}$	$D_{13} – D_{25}$	$D_6 – D_{14}$, $D_{24} – D_{32}$	$D_1 – D_4$, $D_{17} – D_{20}$, $D_{35} – D_{36}$
coif3	$D_{11} – D_{27}$	$D_1 – D_9$, $D_{31} – D_{36}$	$D_1 – D_9$, $D_{29} – D_{36}$	$D_{12} – D_{26}$	$D_7 – D_{13}$, $D_{25} – D_{32}$	$D_1 – D_5$, $D_{15} – D_{23}$, $D_{33} – D_{36}$
Bior2.8	$D_{12} – D_{26}$	$D_1 – D_8$, $D_{31} – D_{36}$	$D_1 – D_9$, $D_{31} – D_{36}$	$D_{13} – D_{25}$	$D_7 – D_{13}$, $D_{24} – D_{32}$	$D_1 – D_4$, $D_{15} – D_{22}$, $D_{35} – D_{36}$
Rbio2.8	$D_{12} – D_{26}$	$D_1 – D_9$, $D_{29} – D_{36}$	$D_1 – D_9$, $D_{29} – D_{36}$	$D_{11} – D_{27}$	$D_6 – D_{14}$, $D_{24} – D_{32}$	$D_1 – D_5$, $D_{15} – D_{23}$, $D_{33} – D_{36}$
dmey	$D_{12} – D_{26}$	$D_1 – D_{10}$, $D_{28} – D_{36}$	$D_1 – D_9$, $D_{30} – D_{36}$	$D_{10} – D_{28}$	$D_7 – D_{13}$, $D_{25} – D_{31}$	$D_1 – D_5$, $D_{15} – D_{23}$, $D_{33} – D_{36}$

Таблица 3.2 – Критерии оценки вейвлет-функций

Критерий	Haar	db5	sym3	coif3	bior2.8	rbio2.8	dmey
Число объектов одинаково классифицируемых во всех дет. коэффициентах	6	0	0	0	0	0	0
Число объектов, одинаково классифицируемых в нескольких дет. коэффициентах	26	18	16	13	10	16	10
Число объектов сегментируемых в горизонтальных коэффициентах	16	15	15	15	13	13	13
Число объектов сегментируемых в вертикальных коэффициентах	25	17	18	17	15	17	16
Число объектов сегментируемых в диагональных коэффициентах	26	18	18	14	14	18	14

На рисунке 3.9 представлены, полученные с помощью этого вейвлета, результаты сегментации контрольного изображения, содержащего разнонаправленные линии.



а *б* *в*

Рисунок 3.9 – Сегментация контрольного изображения на группы объектов, (обозначены белым цветом) визуализированными детализирующими коэффициентами: *а* – горизонтальными, *б* – вертикальными, *в* – диагональными

Как видно из рисунка 3.9 различные визуализированные детализирующие коэффициенты позволяют обозначить группы с четкими границами диапазонов угла наклона объектов входящих в эти группы.

3.2 Использование вейвлетов высоких уровней пакетной декомпозиции

Как известно, пакетное вейвлет-преобразование позволяет раскладывать и низкочастотную, и высокочастотную составляющие части сигнала, что способствует лучшей частотной локализации сигнала [40].

На рисунке 3.10 схематически, в виде дерева, представлено двумерное вейвлет-пакетное разложение сигнала, являющегося изображением $I(x, y)$.

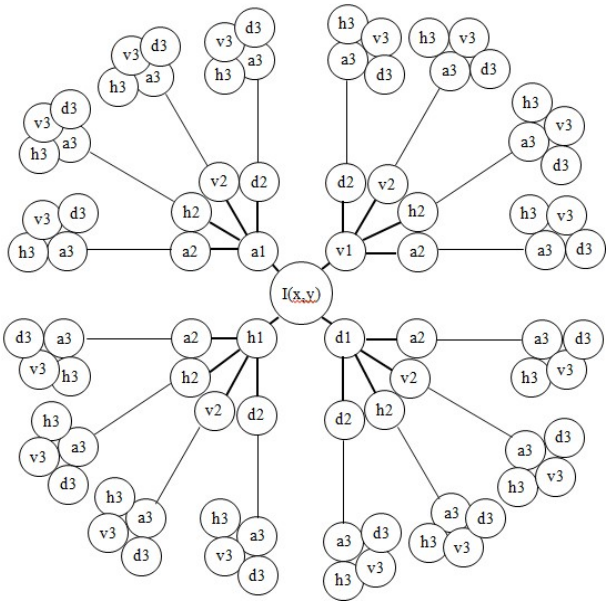


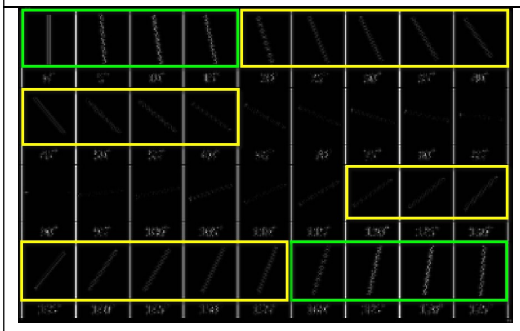
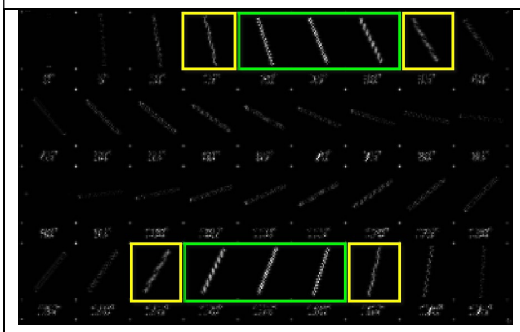
Рисунок 3.10 – Схема пакетного разложения изображения

Буквами *a*, *h*, *v*, *d* обозначаются, соответственно апроксимационные, горизонтальные, вертикальные и диагональные коэффициенты. Цифрами 1 – 3 обозначен уровень разложения.

Исследование проводилось на 2 и 3 уровнях декомпозиции, дальнейшее разложение ведет слишком большому размытию визуализированных коэффициентов. При детальном анализе использовался биортогональный вейвлет 2.8, показавший наилучшие результаты в исследованиях, описанных в главе 3.1. Однако, необходимо отметить, что вейвлеты Хаара, Добеши 5, койфлет 3, симплет 3, обратный биортогональный 2.8, дискретный Майера дают аналогичные результаты, хоть и с различными уровнями дополнительного шума.

Второй уровень разложения изображения позволил сегментировать объекты контрольного изображения на следующие группы, в зависимости от угла наклона (Таблица 3.5): 0° – 15°, 20° – 30°, 25° – 40°, 45° – 65°, 60° – 70°, 75° – 105°, 110° – 120°, 115° – 135°, 140° – 155°, 150° – 160, 165° – 180°.

Таблица 3.5 –Детализирующие коэффициенты 2 уровня разложения

Визуализированные коэффициенты	Уровень разложения	+	+/ –	–
	1v→2a	0° – 15°, 165° – 180°	20° – 60°, 120° – 160°	65° – 115°
	1v→2h	20° – 30°, 150° – 160°	15°,35°, 145°,165°	0° – 10°, 40°-140°, 170°-180°,

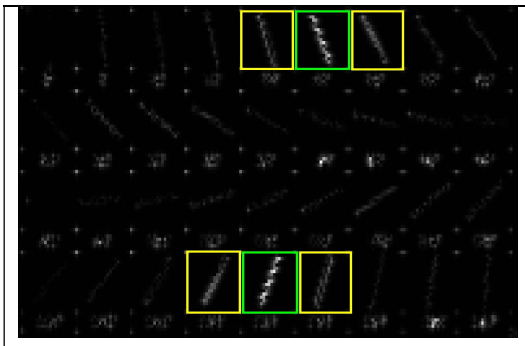
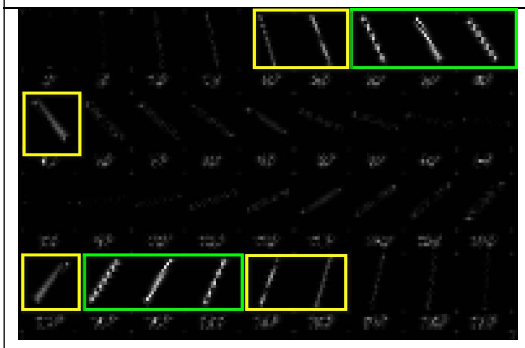
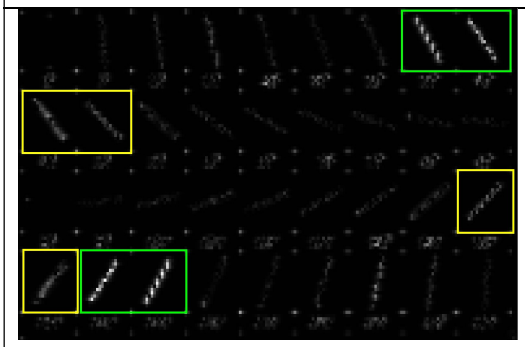
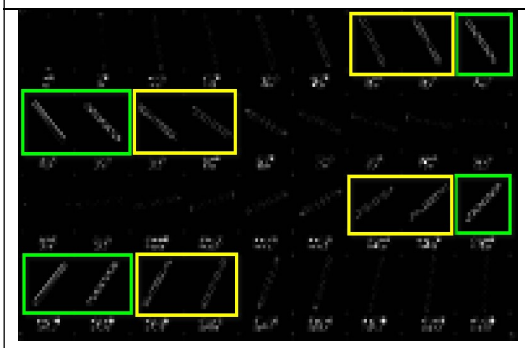
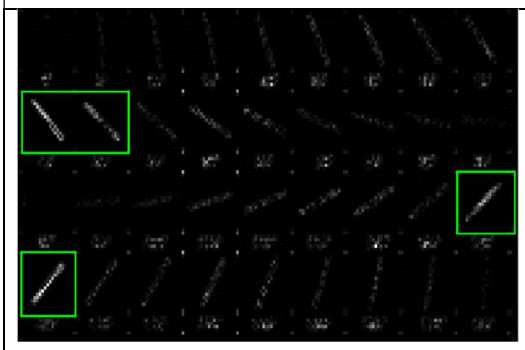
	1d→2h	25° – 40°, 140° – 155°	20°,135°, 160°	0° – 15°, 45°-135°, 165°-180°,
	1h→2d	45° – 65°, 115° – 135°	35° – 40°, 70° – 75°, 105° – 110°, 140° – 145°,	0° – 30°, 80° – 100°, 150° – 180°,
	1h→2v	60° – 70°, 110° – 120°	55°,75°, 105°,125°	0° – 50°, 80° – 100°, 130° – 180°
	1h→2a	75° – 105°	50° – 70°, 110° – 130°	0° – 45°, 135° – 180°

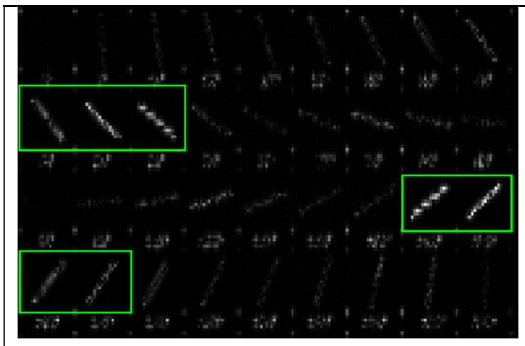
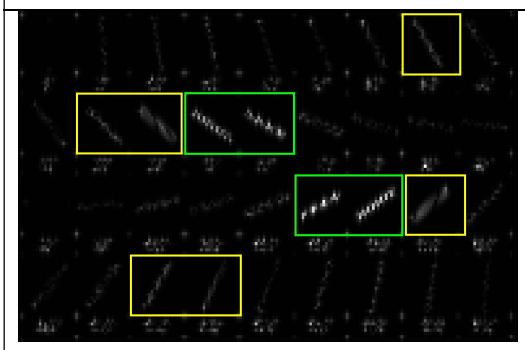
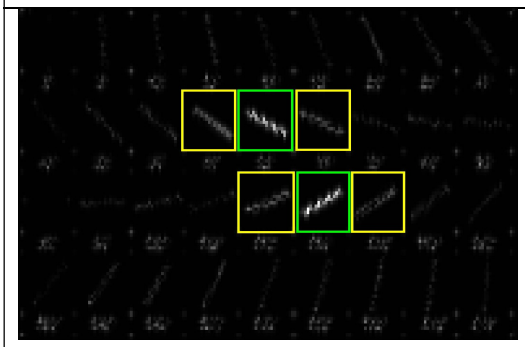
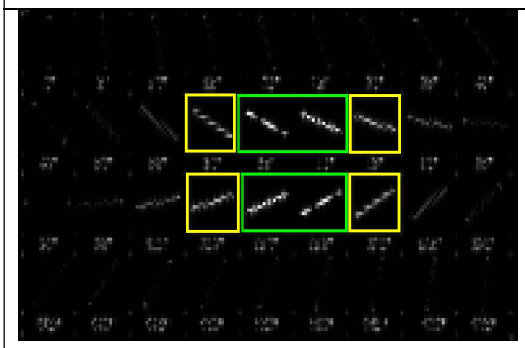
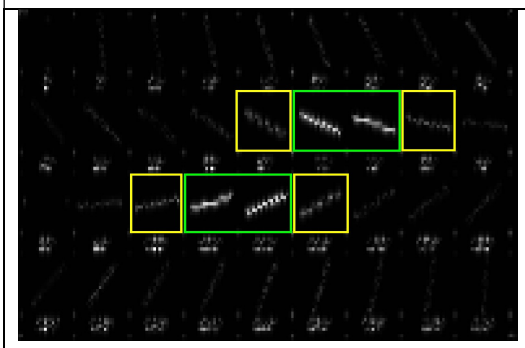
При разложении сигнала до третьего уровня декомпозиции удалось сегментировать объекты на группы с шагом в 5 – 10° (таблица 3.6): 0° – 5°, 10° – 15°, 15° – 20°, 25° – 30°, 30° – 40°, 35° – 40°, 40° – 50°, 50° – 55°, 60° – 65°, 65° – 70°, 70° – 75°, 75° – 80°, 85° – 95°, 100° – 105°, 105° – 110°, 110° – 115°, 115° – 120°, 125° – 130°, 130° – 135°, 130° – 140° 140° – 145°, 150° – 155°, 160° – 165°, 165° – 170°, 175° – 180°, что дает возможность выявлять скрытые изображения, представленные объектами с углом наклона входящим в эти группы.

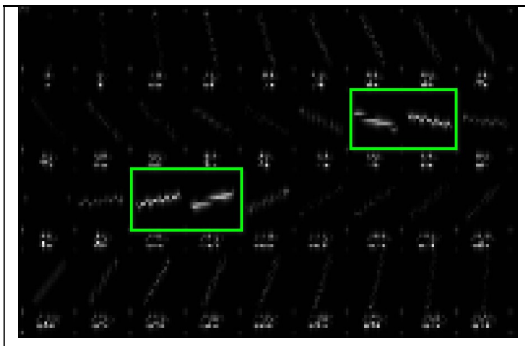
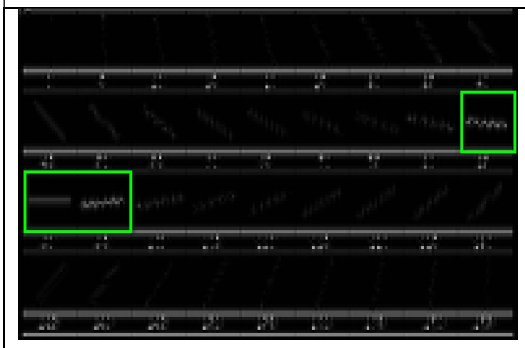
Для выявления скрытого изображения, представленного объектами с углом наклона соответствующим значению столбца «+» таблицы 3.6, необходимо произвести вейвлет-преобразование с соответствующим из таблицы уровнем разложения.

Таблица 3.6 – сегментация на 3 уровне разложения

Визуализированные коэффициенты	Уровень разложения	+	+/ –	–
	v1→a2→a3	0° – 5°, 175° – 180°		10° – 170°
	v1→a2→h3	10° – 15°, 165° – 170°		0° – 5°, 20° – 160°, 175° – 180°
	v1→h2→h3	15° – 20°, 160° – 165°	10°,25°, 155°,170°	0° – 5°, 30° – 150°, 175° – 180°
	d1→h2→a3	25° – 30°, 150° – 155°	20°,35°, 50° – 55°, 120°–125°, 145°, 160°, 170°	0° – 15°, 40° – 45°, 60°– 115°, 130°–140°, 165°, 175° – 180°

	v1→h2→a3	25°,155°	20°,30°, 150°,160°	0° – 15°, 35° – 145°, 165° – 180°
	a1→d2→h3	30° – 40°, 140° – 150°	20° – 25°, 45°,135°, 155° – 160°	0° – 15°, 50° – 130°, 165° – 180°
	d1→a2→h3	35° – 40°, 140° – 145°	45° – 50°, 130° – 135°	0° – 30°, 55° – 125°, 150° – 180°
	a1→d2→d3	40° – 50°, 130° – 140°	30° – 35°, 55° – 60°, 120°– 125°, 145° – 150°	0° – 25°, 65° – 115°, 155° – 180°
	d1→d2→d3	45° – 50°, 130° – 135°		0° – 40°, 55° – 125°, 140° – 180°

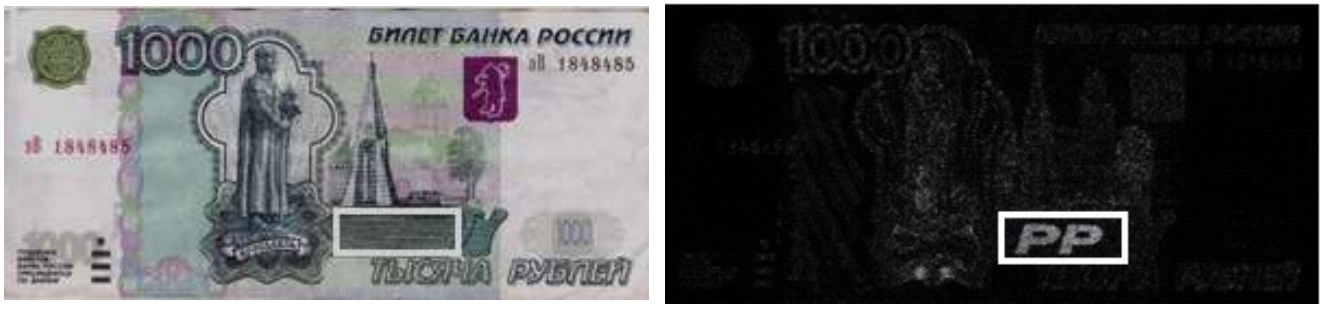
	d1→a2→v3	50° – 55°, 125° – 130°	45°, 135° – 140°	0° – 40°, 60° – 120°, 145° – 180°
	d1→v2→a3	60° – 65°, 115° – 120°	35°, 50° – 55°, 125°, 145° – 150°	0° – 30°, 40° – 45°, 70° – 110°, 130°– 140°, 155° – 180°
	h1→v2→a3	65°, 115°	60°,70°, 110°,120°	0° – 55°, 75° – 105°, 125° – 180°
	h1→h2→v3	65° – 70°, 110° – 115°	60°,75°, 105°,120°	0° – 55°, 80° – 100°, 125° – 180°
	h1→v2→v3	70° – 75°, 105° – 110°	65°,80°, 100°,115°	0° – 60°, 85° – 95°, 120° – 180°

	h1→a2→v3	75° – 80°, 100° – 105°		0° – 70°, 85° – 95°, 110° – 180°
	h1→a2→a3	85° – 95°		0° – 80°, 100° – 180°

Проверка работоспособности предлагаемого метода выполнялась на образцах, содержащих латентные изображения, сформированные с помощью линий, расположенных под определенным углом. Проведено исследование латентных изображений на образцах, представляющих собой сканированные изображения денежных купюр.

Пример сегментации латентного изображения, содержащего скрытую информацию «*PP*», на купюре 1000 российских рублей образца 2004 года, показан на рисунке 3.11. На данном образце имеется скрытое изображение, выполненное с помощью линий, расположенных под углом 110°, что соответствует объекту класса D_{22} контрольного изображения (рисунок 3.1).

Из таблицы 3.6 следует, что для сегментации объекта с углом наклона входящим в диапазон 110° – 115° необходимо выполнить последовательно преобразования до уровня разложения h1→h2→v3 в соответствии с формулами (3.2): $\psi_1^H(x,y) \rightarrow \psi_2^H(x,y) \rightarrow \psi_3^V(x,y)$.



а *б*

Рисунок 3.11 – Пример визуализации скрытой информации на латентном изображении

На полученном в результате преобразований изображении (рисунок 3.11 *б*) явно проявилось скрытое изображение. Стоит отметить, что полученное изображение вполне пригодно для распознавания машинными средствами и может быть использовано для автоматического распознавания с последующим контролем подлинности.

3.3 Выявление скрытых изображений с помощью вейвлет-преобразований

При вейвлет-преобразовании латентных изображений, в которых сюжетное и скрытое изображение представлены текстурами с совпадающим периодом, а также изображений с низким качеством, скрытая информация не выявляется. Для контроля таких изображений предложено перед визуализацией осуществить поворот данного изображения на определенный угол.

3.3.1 Проблема визуализации при совпадающем периоде текстур сюжетного и скрытого изображений

Как видно по результатам исследований (рисунок 3.9) объекты, представленные линиями с совпадающим периодом, при сегментации, попадают в одни и те же группы. Таким образом, если скрытое изображение выполнено линиями зеркальными линиям, из которых построено основное изображение, то выявление становится невозможным.

Если скрытое изображение выполнено линиями зеркальными линиям, из которых построено основное изображение, то выявление становится невозможным, так как объекты, представленные линиями с совпадающим периодом, при сегментации, попадают в одни и те же группы, это видно из результатов, представленных на рисунке 3.9.

Одним из примеров такого изображения является изображение, нанесенное на банкноте РФ номиналом 5000 рублей образца 2006 года. Область латентного изображения выполнена линиями, расположенными под углом 45° (объект D_{10}), скрытое изображение сформировано линиями с углом наклона 135° (объект D_{28}). Попытки выявления данного изображения без дополнительных действий не дают необходимого результата (рисунок 3.12)

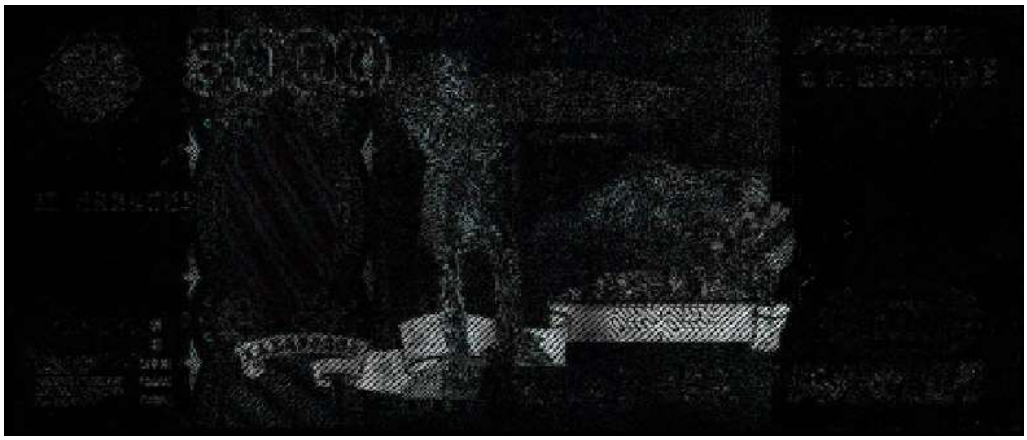


Рисунок 3.12 – Неудачная попытка выявления скрытой информации

Для изменения периода линий изображение необходимо повернуть на несколько градусов. Причем угол поворота может потребоваться разный, в зависимости от угла наклона линий скрытого изображения и уровня разложения на котором исследуется латентное изображение.

Как видно из рисунка 3.13, после поворота изображения на 10 градусов вейвлет-разложение дает положительные результаты выявления скрытого изображения.

Рисунок 3.13 – Результат выявления скрытой информации при повороте на 10°

Учитывая полученные результаты исследований контрольного изображения на различных уровнях в различимости линий разного наклона можно определить следующие углы поворота латентного изображения для обеспечения гарантированного выявления скрытого изображения:

- 1 уровень – 30°;
- 2 уровень – 20°;
- 3 уровень – 10°;

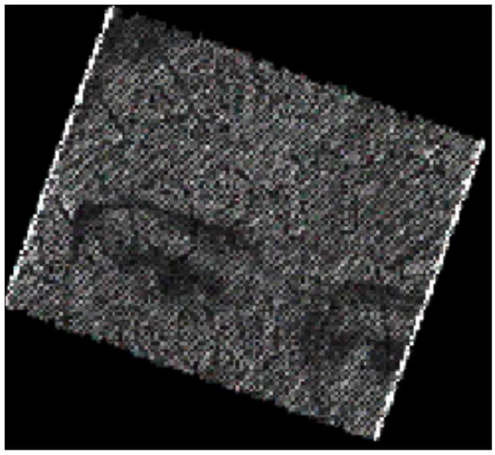
3.3.2 Проблема контроля латентного изображения низкого качества

Размер визуализированного по коэффициентам вейвлет-разложения изображения с каждым уровнем декомпозиции уменьшается в 2 раза, поэтому изображение, сканированное с низким разрешением, на высоких уровнях декомпозиции зачастую теряет свою структуру. При этом, если выявление скрытого изображения по набору структурных линий возможно только на одном из высоких уровней декомпозиции, распознать его становится невозможно.

Возможны также ситуации, когда латентное изображение наносится на рельефную бумагу, образующую ложную структуру на сканированном изображении, что может привести к смешению коэффициентов вейвлет-разложения, описывающих скрытое изображение с описывающими ложную структуру.

Решением данной проблемы может стать поворот изображения таким образом, чтобы угол наклона структурных линий оказался вне диапазона, в котором находятся структурные линии образующие скрытое изображение. Например, латентное изображение, представленное на рисунке 3.14*а* при вейвлет-разложении без дополнительных действий не позволяет выявить скрытое изображение, но при повороте на 30° становится заметным повторяющийся водяной знак в виде текстовой надписи: «AA000217» (рисунок 3.14*б*).

а



б

Рисунок 3.14 – Изображения: *а* – латентное, *б* – диагональных коэффициентов с визуализированной скрытой информацией

Полученные результаты показали, что объекты с совпадающим периодом предложенным методом сегментации классифицируются одинаково. Также из таблицы 3.1 видно, что максимальный диапазон объектов с различным углом наклона, относящихся к одной группе, равен 60°. Если фоновое и скрытое изображение образовано объектами со взаимным расположением друг относительно друга под углом более 30° и области фонового и скрытого изображений при сегментации на первом уровне декомпозиции были отнесены к одной группе, то повернутое на 30° латентное изображение гарантировано будет содержать объекты фонового и скрытого изображений, относящихся к разным классам даже на первом уровне декомпозиции. Таким образом, показана

возможность использования результатов вейвлет анализа латентных изображений для сегментации скрытого изображения.

3.4 Математическая модель и численный метод контроля латентного изображения

Разработана математическая модель контроля латентного изображения на основе вейвлет-преобразований изображения.

Перед выполнением вейвлет-преобразований латентное изображение $L\ (x, y, z)$ приводится к оттенкам серого по формуле:

$$L_{Grey}\ (x, y) = 0.299\, L_R\,(x, y) + 0.587\, L_G\,(x, y) + 0.114\, L_B\,(x, y) \tag{3.6}$$

Производится свертка по строкам с использованием вейвлета $\psi(y)$ полученного латентного изображения в оттенках серого $L_{Grey}(x, y, z)$ и получение детализирующих коэффициентов $W_1(1, m, n)$ последующей сверткой по столбцам с использованием вейвлета $\psi(y)$:

$$\begin{cases} Conv_R = h_\psi(-n) * L_{Grey}(x, y)\,|_{n=2k,k\geq 0} \\ W_1\ (1, m, n) = h_\psi(-m) * Conv_R\ |_{m=2k,k\geq 0} \end{cases} \tag{3.7}$$

Полученные детализированные коэффициенты $W_1(1, m, n)$ визуализируются с образованием визуализированного изображения $E(x, y)$:

$$E(x, y) = W_a\ (1, m, n)\,|_{1\leq a\leq 6} \tag{3.8}$$

Если в полученном визуализированном изображении $E(x, y)$, построенным по коэффициентам $W_1(1, m, n)$, не проявилось скрытое изображение производится вычисление следующих детализированных коэффициентов. Производится свертка по строкам с использованием масштабирующей функции $\varphi(x)$ полученного латентного изображения в оттенках серого $L_{Grey}(x, y, z)$ и получение детализирующих коэффициентов $W_2(1, m, n)$ последующей сверткой по столбцам с использованием вейвлета $\psi(y)$ (система уравнений 3.9). Полученные детализированные коэффициенты $W_2(1, m, n)$ визуализируются с образованием визуализированного изображения $E(x,y)$ по формуле 3.8.

$$\begin{cases} Conv_R = h_\varphi(-n) * L_{Grey}(x,y)|_{n=2k,k\geq 0} \\ W_2\ (1,m,n) = h_\psi(-m) * Conv_R\ |_{m=2k,k\geq 0} \end{cases} \quad (3.9)$$

Если в полученном визуализированном изображении $E(x,y)$, построенным по коэффициентам $W_2(1,m,n)$, не проявилось скрытое изображение производится вычисление следующих детализированных коэффициентов. Производится свертка по строкам с использованием вейвлета $\psi(y)$ полученного латентного изображения в оттенках серого $L_{Grey}(x,y,z)$ и получение детализирующих коэффициентов $W_3(1,m,n)$ последующей сверткой по столбцам с использованием масштабирующей функции $\varphi(x)$ (система уравнений 3.10). Полученные детализированные коэффициенты $W_3(1,m,n)$ визуализируются с образованием визуализированного изображения $E(x,y)$ по формуле 3.8.

$$\begin{cases} Conv_R = h_\psi(-n) * L_{Grey}(x,y)|_{n=2k,k\geq 0} \\ W_3\ (1,m,n) = h_\varphi(-m) * Conv_R\ |_{m=2k,k\geq 0} \end{cases} \quad (3.10)$$

Если в полученном визуализированном изображении $E(x,y)$, построенным по коэффициентам $W_3(1,m,n)$, производится исследование отраженного на угол 30° латентного изображения в оттенках серого $L_{Grey}(x,y)$:

$$L_{Rot30}(x,y) = L_{Grey}(\frac{\sqrt{3}}{2}x_i + \frac{1}{2}x_j, -\frac{1}{2}x_i - \frac{\sqrt{3}}{2}x_j)|_{i\geq 0,j\geq 0} \quad (3.11)$$

Производится свертка по строкам с использованием вейвлета $\psi(y)$ полученного латентного изображения в оттенках серого $L_{Rot30}(x,y)$ и получение детализирующих коэффициентов $W_4(1,m,n)$ последующей сверткой по столбцам с использованием вейвлета $\psi(y)$ (система уравнений 3.12). Полученные детализированные коэффициенты $W_4(1,m,n)$ визуализируются с образованием визуализированного изображения $E(x,y)$ по формуле 3.8.

$$\begin{cases} Conv_R = h_\psi(-n) * L_{Rot30}(x,y)|_{n=2k,k\geq 0} \\ W_4\ (1,m,n) = h_\psi(-m) * Conv_R\ |_{m=2k,k\geq 0} \end{cases} \quad (3.12)$$

Если в полученном визуализированном изображении $E(x,y)$, построенным по коэффициентам $W_4(1,m,n)$, не проявилось скрытое изображение производится

вычисление следующих детализированных коэффициентов. Производится свертка по строкам с использованием масштабирующей функции $\varphi(x)$ изображения $L_{Rot30}(x,y)$ и получение детализирующих коэффициентов $W_5(1,m,n)$ последующей сверткой по столбцам с использованием вейвлета $\psi(y)$ (система уравнений 3.13). Полученные детализированные коэффициенты $W_5(1,m,n)$ визуализируются с образованием визуализированного изображения $E(x,y)$ по формуле 3.8.

$$\begin{cases} Conv_R = h_\varphi(-n) * L_{Rot30}(x,y)|_{n=2k,k\geq 0} \\ W_5\ (1,m,n) = h_\psi(-m) * Conv_R\ |_{m=2k,k\geq 0} \end{cases} \tag{3.13}$$

Если в полученном визуализированном изображении $E(x,y)$, построенным по коэффициентам $W_5(1,m,n)$, не проявилось скрытое изображение производится вычисление следующих детализированных коэффициентов. Производится свертка по строкам с использованием вейвлета $\psi(y)$ изображения $L_{Rot30}(x,y)$ и получение детализирующих коэффициентов $W_6(1,m,n)$ последующей сверткой по столбцам с использованием масштабирующей функции $\varphi(x)$ (система уравнений 3.14). Полученные детализированные коэффициенты $W_6(1,m,n)$ визуализируются с образованием визуализированного изображения $E(x,y)$ по формуле 3.8.

$$\begin{cases} Conv_R = h_\psi(-n) * L_{Rot30}(x,y)|_{n=2k,k\geq 0} \\ W_6\ (1,m,n) = h_\varphi(-m) * Conv_R\ |_{m=2k,k\geq 0} \end{cases} \tag{3.14}$$

Для выполнения подзадачи описания скрытых изображений и построения базы контрольных изображений список характерных параметров таких как цвет, форма, локализация и т.д. целесообразно свести к минимуму до формы объекта. Для этого производится замыкание объектов [76] сегментированного изображения и его бинаризация:

$$B(x,y) = \begin{cases} 1, \text{при } E(x,y) \geq t, \\ 0, \text{при } E(x,y) < t \end{cases} \tag{3.15}$$

Полученное скрытое изображение сравнивается с эталоном из БД контрольных образцов соответствующих данному типу латентного изображения,

одним из методов анализа образов. Если процент соответствия скрытого изображения с эталоном выше порогового, установленного в зависимости от решаемой задачи, исследуемый образец признается подлинным.

Математическая модель выявления латентного изображения на основе вейвлет-преобразования имеет вид:

$$\begin{cases} L_{Grey}(x,y) = 0.299\,L_R(x,y) + 0.587\,L_G(x,y) + 0.114\,L_B(x,y) \\ L_{Rot30}(x,y) = L_{Grey}\left(\frac{\sqrt{3}}{2}x_i + \frac{1}{2}x_j, -\frac{1}{2}x_i - \frac{\sqrt{3}}{2}x_j\right)\Big|_{i\geq 0, j\geq 0} \\ \begin{cases} Conv_R = h_\psi(-n) * L_{Grey}(x,y)\big|_{n=2k,k\geq 0} \\ W_1(1,m,n) = h_\psi(-m) * Conv_R\big|_{m=2k,k\geq 0} \end{cases} \\ \begin{cases} Conv_R = h_\varphi(-n) * L_{Grey}(x,y)\big|_{n=2k,k\geq 0} \\ W_2(1,m,n) = h_\psi(-m) * Conv_R\big|_{m=2k,k\geq 0} \end{cases} \\ \begin{cases} Conv_R = h_\psi(-n) * L_{Grey}(x,y)\big|_{n=2k,k\geq 0} \\ W_3(1,m,n) = h_\varphi(-m) * Conv_R\big|_{m=2k,k\geq 0} \end{cases} \\ \begin{cases} Conv_R = h_\psi(-n) * L_{Rot30}(x,y)\big|_{n=2k,k\geq 0} \\ W_4(1,m,n) = h_\psi(-m) * Conv_R\big|_{m=2k,k\geq 0} \end{cases} \\ \begin{cases} Conv_R = h_\varphi(-n) * L_{Rot30}(x,y)\big|_{n=2k,k\geq 0} \\ W_5(1,m,n) = h_\psi(-m) * Conv_R\big|_{m=2k,k\geq 0} \end{cases} \\ \begin{cases} Conv_R = h_\psi(-n) * L_{Rot30}(x,y)\big|_{n=2k,k\geq 0} \\ W_6(1,m,n) = h_\varphi(-m) * Conv_R\big|_{m=2k,k\geq 0} \end{cases} \\ E(x,y) = W_a(1,m,n)\big|_{1\leq a\leq 6} \\ B(x,y) = \begin{cases} 1, \text{при } E(x,y) \geq t, \\ 0, \text{при } E(x,y) < t \end{cases} \end{cases} \tag{3.16}$$

где $L(x,y,z)$ – латентное изображение;

$L_{Grey}(x,y)$ – латентное изображение, преобразованное к оттенкам серого;

$L_{Rot30}(x,y)$ – латентное изображение в оттенках серого повернутое на 30°;

$Conv_R$ – свертка вейвлет функции по строкам изображения;

$W_a(1,m,n)$ – детализирующие вейвлет-коэффициенты;

$E(x,y)$ – визуализированное скрытое изображение;

$B(x,y)$ – контролируемое изображение;

x,y,z – координаты цветовых компонент в системе RGB;

n – номер строки, m – номер столбца;

t – адаптивный порог бинаризации.

На основе этой модели разработан численный метод контроля латентных изображений, который заключается в том, что:

1. Латентное изображение $L(x, y, z)$ преобразуется к оттенкам серого $L_{Grey}(x, y)$:

$$L_{Grey}(x, y) = 0.299\, L_R(x, y) + 0.587\, L_G(x, y) + 0.114\, L_B(x, y) \tag{3.17}$$

2. Последовательно вычисляются диагональные, вертикальные и горизонтальные вейвлет-коэффициенты $W_a(1, m, n)$ с использованием биортогонального вейвлета с базисом 2.8 путем вейвлет-преобразования изображения $L_{Grey}(x, y)$. Визуализированные вейвлет-коэффициенты $W_a(1, m, n)$ образуют изображение $E(x, y)$.

$$\begin{cases} h_\psi(t) = \{\{0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.3535533905932738, \\ -0.7071067811865476,0.3535533905932738,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0\} \\ h_\varphi = \{\{0.0,0.0015105430506304422, -0.0030210861012608843, \\ -0.012947511862546647, \\ 0.02891610982635418,0.052998481890690945, -0.13491307360773608, \\ -0.16382918343409025,0.4625714404759166,0.9516421218971786, \\ 0.4625714404759166, -0.16382918343409025, -0.13491307360773608, \\ 0.052998481890690945,0.02891610982635418, -0.012947511862546647, \\ -0.0030210861012608843,0.0015105430506304422\} \\ \begin{cases} Conv_R = h_\psi(-n) * L_{Grey}(x, y)|_{n=2k,k\geq 0} \\ W_1\ (1, m, n) = h_\psi(-m) * Conv_R\ |_{m=2k,k\geq 0} \end{cases} \\ \begin{cases} Conv_R = h_\varphi(-n) * L_{Grey}(x, y)|_{n=2k,k\geq 0} \\ W_2\ (1, m, n) = h_\psi(-m) * Conv_R\ |_{m=2k,k\geq 0} \end{cases} \\ \begin{cases} Conv_R = h_\psi(-n) * L_{Grey}(x, y)|_{n=2k,k\geq 0} \\ W_3\ (1, m, n) = h_\varphi(-m) * Conv_R\ |_{m=2k,k\geq 0} \end{cases} \\ E(x, y) = W_a\ (1, m, n)\,|_{1\leq a\leq 3} \end{cases} \tag{3.18}$$

3. Получаем контролируемое изображение $B(x, y)$, путем бинаризации по адаптивному порогу t визуализированного скрытого изображения $E(x, y)$:

$$B(x, y) = \begin{cases} 1, \text{при } E(x, y) \geq t, \\ 0, \text{при } E(x, y) < t \end{cases} \tag{3.19}$$

4. Контрольное изображение $B(x,y)$ сравнивается со скрываемым изображением $S(x,y)$:

$$(x,y) = \begin{cases} \text{выявлено, при } B(x,y) = G(x,y), \\ \text{не выявлено, при } B(x,y) \neq G(x,y) \end{cases} \quad (3.20)$$

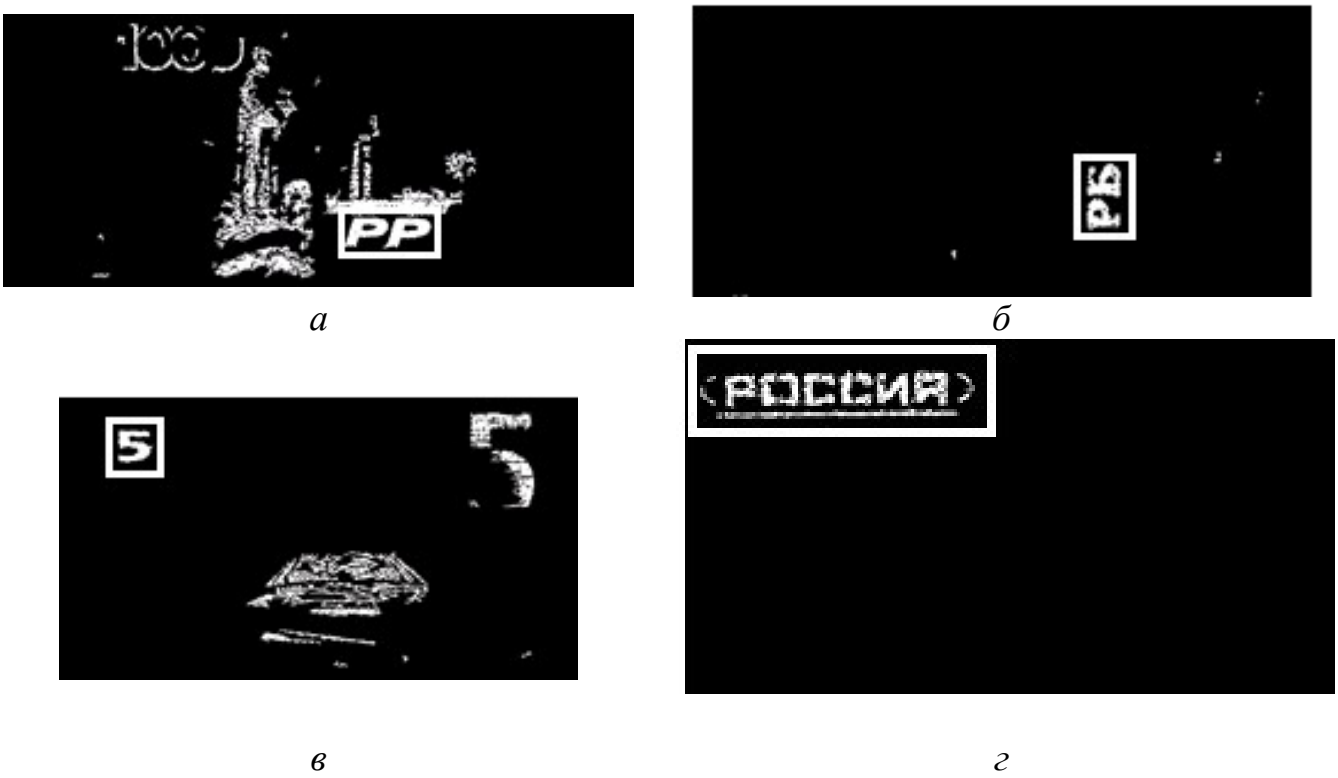
5. Если скрытое изображение не выявлено, осуществляется поворот изображения $L_{Grey}(x,y)$ на 30 градусов и проводится вейвлет анализ изображения $L_{Rot30}(x,y)$:

$$L_{Rot30}(x,y) = L_{Grey}(\frac{\sqrt{3}}{2}x_i + \frac{1}{2}x_j, -\frac{1}{2}x_i - \frac{\sqrt{3}}{2}x_j)\,|_{i\geq 0, j\geq 0} \quad (3.21)$$

6. Выполняется п. 2 для изображения $L_{Rot30}(x,y)$.

7. Выполняются п. 3. 4.

На рисунках 3.15 и 3.16 представлены результаты выявления латентных изображений предложенным и известным методами соответственно.



а *б* *в* *г*

Рисунок 3.15 – Результат выявления латентных изображений предложенным методом: *а* – банкнота номиналом 1000 российских рублей, *б* – банкнота номиналом 50 белорусских рублей, *в* – банкнота номиналом 5 азербайджанских манат, *г* – российский паспорт. Визуализированная скрытая информация помещена в рамку

а *б*

в *г*

Рисунок 3.16 – Результат выявления латентных изображений известным методом: *а* – банкнота номиналом 1000 российских рублей *б* – банкнота номиналом 50 белорусских рублей, *в* – банкнота номиналом 5 азербайджанских манат, *г* – российский паспорт

В отличие от известного метода, основанного на применении последовательности фильтров, зависящей от способа внедрения скрытой информации в латентное изображение, предложенный метод, не зависит от способа формирования латентного изображения и позволяет автоматизировать процесс контроля таких изображений.

3.5 Система формирования и контроля латентных изображений

На основе разработанных методов предложена система формирования и контроля латентных изображений.

Совокупность методов и алгоритмов, позволяющих внедрить одно изображение в другое будем представляет собой систему формирования и контроля латентных изображений (СФКЛИ). СФКЛИ $L = (F, F^{-1}, I, S, P, V)$

является частным случаем стеганографической системы и содержит совокупность исходных I, скрываемых S изображений, способов выявления V и формирования P, а также связывающих их преобразований – прямых F и обратных F^{-1}.

Прямыми преобразованиями являются преобразования, не зависящие от предыдущих трансформаций. К таким преобразованиям относятся: $F(i, s, p) = i_{s,p}$ – описывающее процесс формирования латентного изображения $i_{s,p}$; $F(i_{s,p}) = \widetilde{i_{s,p}}$ – преобразование латентного изображения в ходе передачи; $F(\widetilde{i_{s,p}}, v) = s_{p,v}$ – описывающее процесс выявления скрытого изображения $s_{p,v}$ способом не зависящим от способа формирования. Под обратным преобразованием понимается преобразование $F^{-1}(\widetilde{i_{s,p}}, p) = s_p$, описывающее процесс выявления скрытого изображения s_p на основе способа обратного способу формирования латентного изображения.

Под формированием латентного изображения $i_{s,p}$ с помощью системы L понимается процесс прямого графического преобразования F к конкретным i, s, p. Под выявлением – процесс обратного графического преобразования F^{-1} над конкретными $\widetilde{i_{s,p}}$ и p или прямого F над конкретными $\widetilde{i_{s,p}}$ и v.

Можно выделить 8 состояний СФКЛИ (рисунок 3.17):

1. Выбор исходного i, скрытого s изображений и конкретного способа формирования p.

2. Прямое преобразование $F(i, s, p)$ изображений i, s в соответствии с выбранным методом p.

3. Сформированное латентное изображение $i_{s,p}$, предназначенное для передачи по каналам связи.

4. В процессе передачи по различным каналам с возможными искажениями, связанными как со злым умыслом, так и техническими особенностями устройств передачи информации, латентное изображение $i_{s,p}$ подвергается различным преобразованиям $F(i_{s,p})$.

5. В результате получается модифицированное изображение $\widetilde{i_{s,p}}$. Для контроля этого изображения необходимо знать либо метод формирования ЛИ p, либо метод выявления v.

6. Над изображением $\widetilde{i_{s,p}}$ производится обратное преобразование $F^{-1}(\widetilde{i_{s,p}}, p)$, если известен метод формирования латентного изображения p или прямое преобразование $F(\widetilde{i_{s,p}}, v)$, если метод формирования ЛИ неизвестен, в этом случае необходимо выбрать метод выявления v.

7. Выявленное скрытое изображение s_p, на основе которого осуществляется контроль латентного изображения $i_{s,p}$.

8. Сравнение выявленного скрытого s_p и контрольного s изображений.

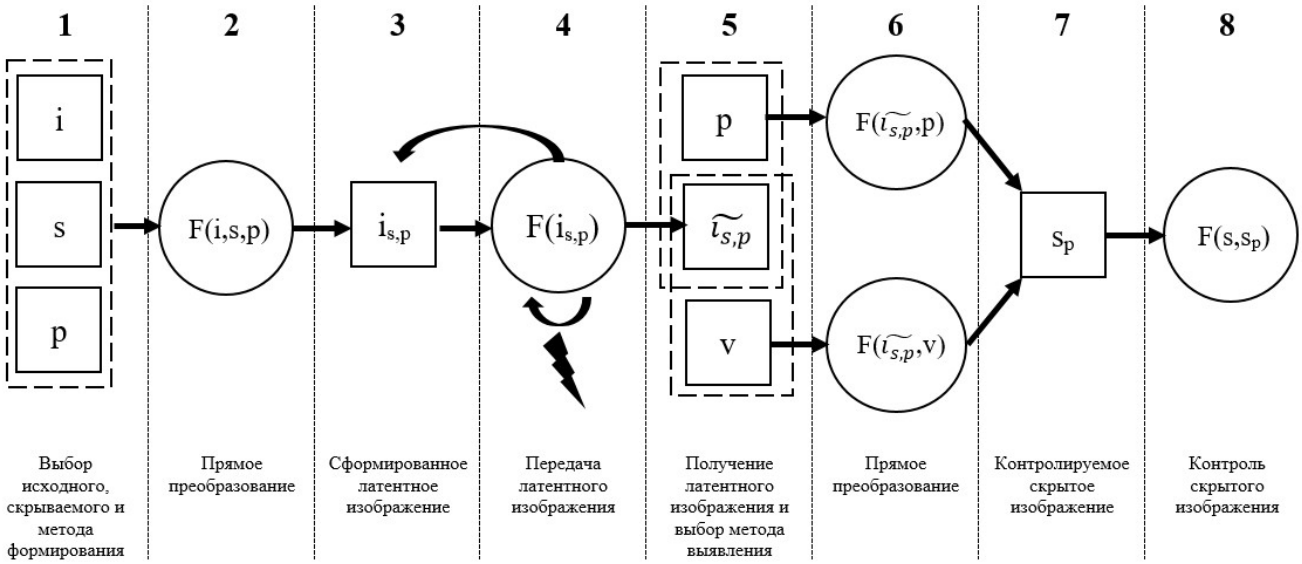


Рисунок 3.17 – Схема процессов формирования и контроля латентных изображений в СФКЛИ

Для реализации процессов формирования и контроля латентных изображений, представленных в структурной схеме СФКЛИ, разработаны соответствующие алгоритмы и программные комплексы, описанные в главе 4.

Разработанная система формирования и контроля латентных изображений, на основе предложенных методов позволяет представить процессы от момента формирования до определения подлинности защищенного документа, в отличие

от известных, в разработанной системе предусмотрен контроль латентных изображений, как при известном, так и при неизвестном методах внедрения скрываемых изображений.

3.6 Выводы

Проведен анализ результатов выявления скрытой информации, с помощью различных базисных вейвлетов для построения математической модели на основе вейвлет-преобразований латентных изображений и разработки численного метода контроля этих изображений.

Определены критерии качества вейвлет-функции для решения поставленных задач, обеспечивающие различимость в визуализированных коэффициентах одних элементов при неразличимости других элементов изображения.

В результате проведенных вычислительных экспериментов установлено, что биортогональный вейвлет по базису 2.8, по сравнению с другими вейвлетами, имеет более высокую локализацию, как по частоте, так и по времени. Данный вейвлет лучше всего подходит для решения поставленных задач, так как позволяет сегментировать объекты на группы с примерно равным числом элементов и минимальным числом элементов, одновременно входящих в разные группы.

Пакетное вейвлет-преобразование позволяет добиться более высокой частотной локализации и сегментировать компоненты латентного изображения на большее число групп, однако для исследования изображений на более высоких уровнях декомпозиции требуется высокое разрешение изображения.

Для решения проблем при анализе изображения, связанных с совпадающим периодом текстур фонового и скрытого изображений и низкого качества исследуемого образца, применяется поворот изображения.

Разработанные математическая модель выявления скрытой информации в латентном изображении и численный метод контроля латентных изображений, который не зависит от конкретного способа формирования латентных

изображений, сформированных различно ориентированными структурами или образованных структурами с совпадающим периодом фонового и скрытого изображения, позволяют установить факт наличия скрытого изображения в исследуемом образце, а также в автоматическом режиме установить тип контролируемого латентного изображения.

Для сегментации каждого объекта $D_1 – D_{36}$ известными методами требуется построение индивидуального фильтра. Таким образом, требуемое число преобразований, согласно формулы 1.27, для гарантированного выявления скрытой информации равно 36. В предложенном методе число преобразований сокращено до шести, что дает шестикратное сокращение числа операций по сравнению с известными методами.

На основе разработанных методов предложена система формирования и контроля латентных изображений, позволяющая описать соответствующие процессы, для построения алгоритмов и создания программных комплексов.

ГЛАВА 4 РАЗРАБОТКА ПРОГРАММНЫХ СРЕДСТВ ФОРМИРОВАНИЯ И КОНТРОЛЯ ЛАТЕНТНЫХ ИЗОБРАЖЕНИЙ

Данная глава содержит описание разработанных программного модуля формирования латентных изображений [85] и программного комплекса контроля латентных изображений, включающего в себя программные модули: «Конвейерный захват латентных изображений» [86], «Построение комбинированных фильтров» [87], «Построение и хеширование контрольного скрытого объекта в латентном изображении» [88], «Автоматическое определение объекта по распознанному латентному изображению» [89, 90], «Принятие решения», а также результаты тестирования разработанных методов и алгоритмов. Разработанные программные средства зарегестрированы в установленном порядке с получением свидетельств о регистрации электронных ресурсов (приложение 4).

4.1 Разработка программного модуля формирования латентного изображения

Для проведения диссертационного исследования был разработан программный модуль на языке MATLAB формирования латентного изображения, позволяющий на основе двух изображений: исходного и скрываемого создать латентное изображение. Программный модуль разработан на основе алгоритма, представленного на рисунке 4.1, построенного на основе модели, описанной в главе 2 диссертационного исследования.

Окно программного модуля создания латентного изображения (рисунок 4.2) включает в себя компоненты для выбора исходного и внедряемого изображений (1), после выбора которых становится активной кнопка формирования латентного изображения (2). Сформированное изображение появляется в окне предварительного просмотра (4). При удовлетворительных результатах –

латентное изображение визуально неотличимо от оригинала, сформированное изображение можно сохранить на диске при помощи кнопки «Сохранить» (3).

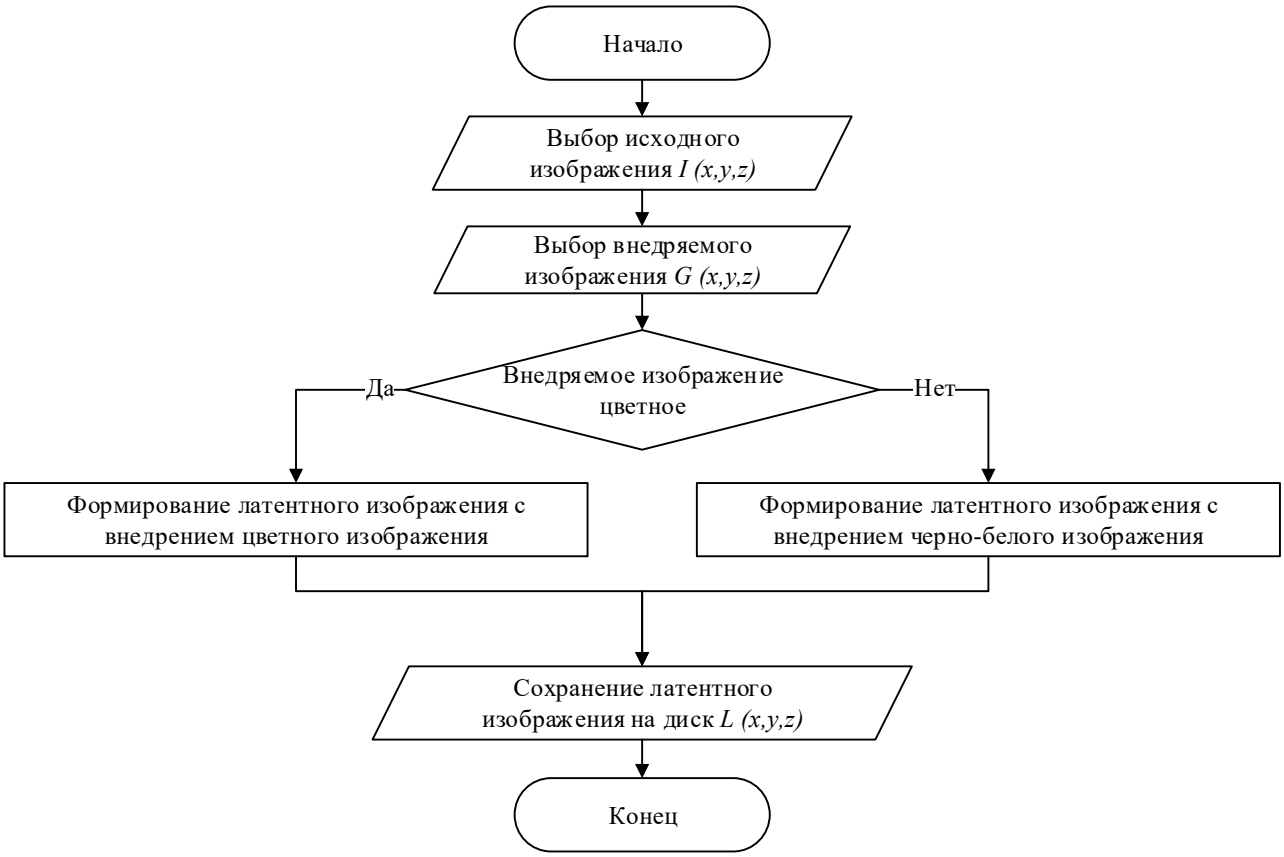


Рисунок 4.1 – Алгоритм формирования латентного изображения

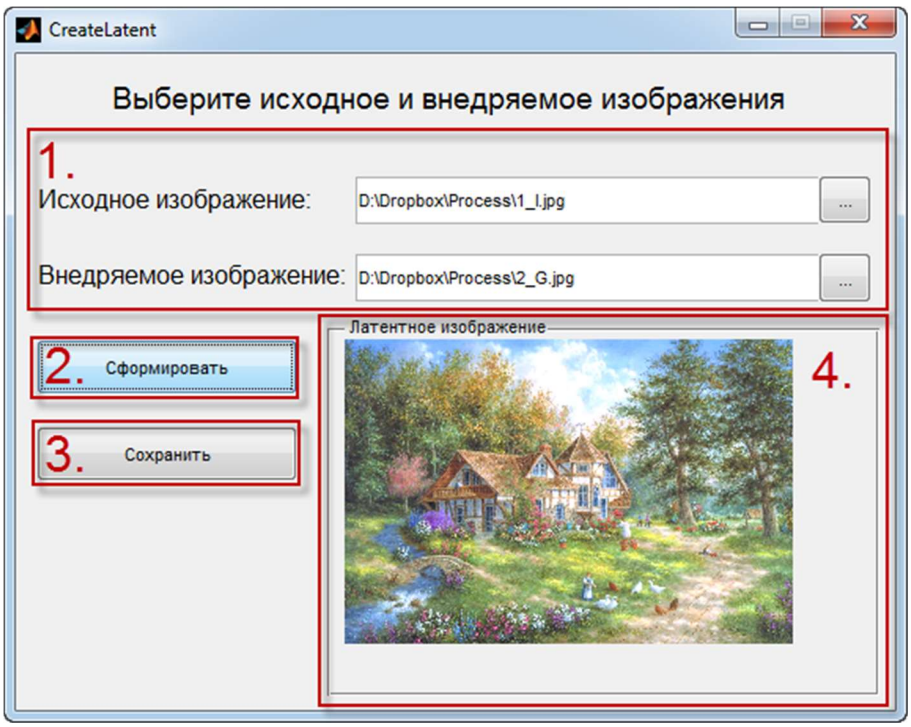


Рисунок 4.2 – Окно программного модуля формирования латентного изображения

Разработанный программный модуль позволяет автоматически формировать латентное изображение на основе загруженного основного и скрываемого изображения. Листинг програмного модуля приведен в приложении 1

4.2 Разработка программного комплекса контроля латентных изображений

Для решения задачи распознавания латентного изображения необходимо было решить следующие подзадачи:

1. Получить пригодное для распознавания латентное изображение. Изображение после АЦП должно сохранить свою структуру.

2. Определить наличие скрытого изображения, и, соответственно, подлинность документа.

3. Принять решение о последующих действиях на основании результата определения подлинности документа.

На рисунке 4.3 представлен обобщенный алгоритм решения задачи:

Рисунок 4.3 – Обобщенный алгоритм решения задачи

На основе разработанных моделей и алгоритмов создан программный комплекс распознавания латентного изображения на языке программирования C# [91-97].

Обобщенная структура программного комплекса представлена на рисунке 4.4.

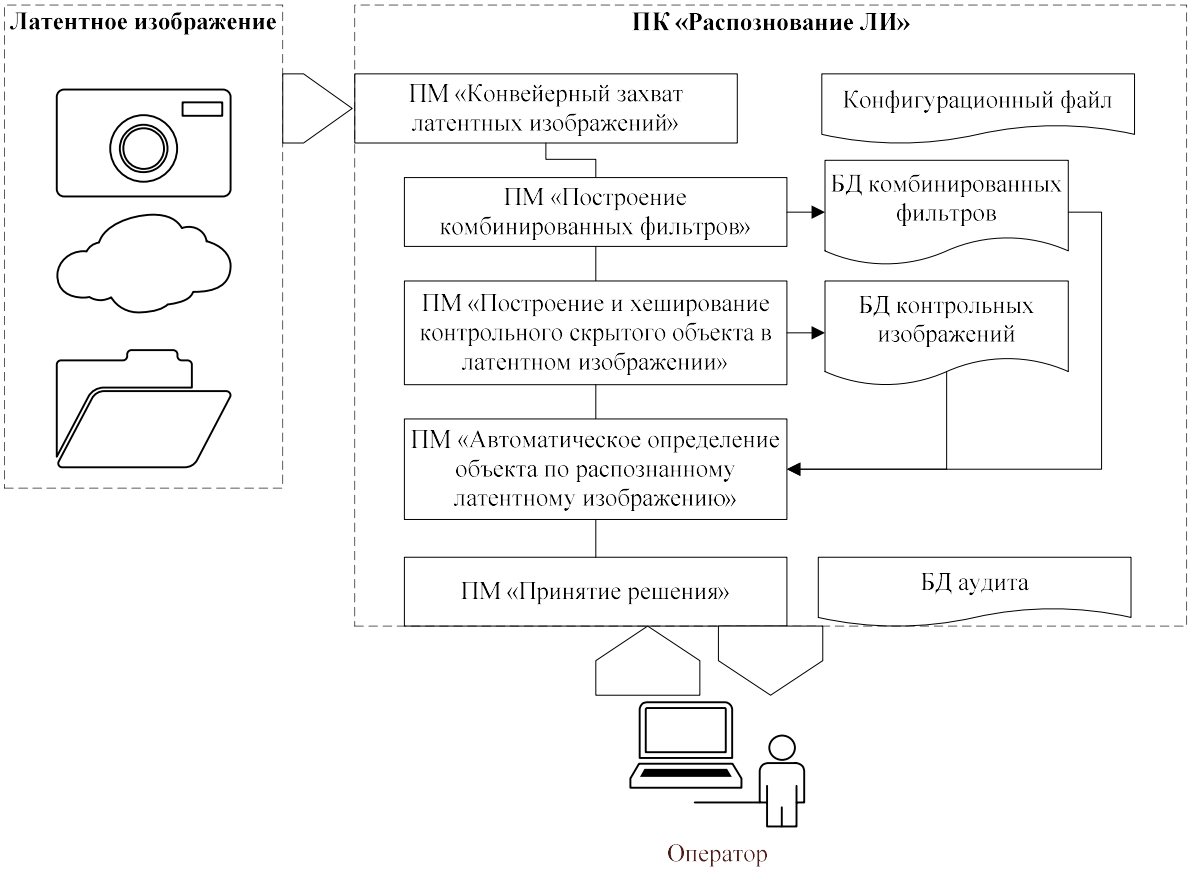


Рисунок 4.4 – Обобщенная структура программного комплекса

Программный комплекс состоит из пяти взаимосвязанных программных модулей:

1. Программный модуль «Конвейерный захват латентных изображений».
2. Программный модуль «Построение комбинированных фильтров».
3. Программный модуль «Построение и хеширование контрольного скрытого объекта в латентном изображении».
4. Программный модуль «Автоматическое определение объекта по распознанному латентному изображению».
5. Программный модуль «Принятие решения».

В состав программного комплекса также входят: конфигурационный файл, файл БД комбинированных фильтров, файл БД контрольных изображений, файл лог-файла и каталог с выявленными изображениями.

Главное окно разработанного программного комплекса, представленное на рисунке 4.5, содержит в своем составе:

– кнопку активации процесса выявления латентных изображений (1);
– кнопку настроек параметров программного комплекса (2);
– рабочую область (3);
– панель отладки (4).

Рисунок 4.5 – Главное окно программного комплекса «Распознавание латентных изображений»

Предусмотрено два режима работы с программным комплексом «Распознавание латентных изображений» автоматический и ручной.

При автоматическом принятии решения программный комплекс без вмешательства оператора принимает решение о подлинности исследуемых образцов.

При ручном режиме – требуется подтверждение оператора, которому на дисплей выводится серия предположительно выявленных скрытых изображений с использованием различных фильтров и результат проверки подлинности. Оператор на основании выведенной на экран информации принимает окончательное решение о подлинности документа. Листинг программного комплекса приведен в приложении 3.

4.2.1 Программный модуль «Конвейерный захват латентных изображений»

Получение латентного изображения для распознавания возможно двумя способами:

1. С камеры.
2. С устройства хранения, локального сетевого ресурса или облачного хранилища.

Благодаря тому, что в IP-камерах нет необходимости передавать аналоговый сигнал в формате PAL или NTSC, в IP-камерах могут использоваться большие разрешения. Для распознавания латентного изображения требуется получение изображение достаточного для различения растровой структуры качества. Поэтому при решении задач получения латентных изображений с устройств видеозахвата, в качестве такого устройства выбирается IP-камера. Также IP-камеры широко используются в составе робототехнических комплексов (РТК).

Получение готового к распознаванию изображения с камеры начинается с активации устройства захвата, происходящей при попадании полностью в область камеры латентного изображения. Алгоритм проверки условия активации устройства захвата, изображенный на рисунке 4.6, состоит из следующих действий:

1. Кадр, полученный с камеры, проверяется с предыдущим полученным кадром. Тем самым проверяется, попал ли в кадр какой-либо предмет, или часть предмета. Если эти два кадра идентичны и, соответственно, новых предметов в области видимости не появилось, то устройство захвата не активируется.

2. Если в кадр попал какой-то предмет, необходимо проверить полностью он попал или нет. Для этого:

2.1 Проверяется, имеется ли полное соответствие цветов пикселей по периметру кадра с цветами пикселей периметра фона. Если такое соответствие имеется, то считаем, что изображение полностью попало в кадр и необходимо активировать устройство захвата, иначе осуществляется дополнительная проверка, предусмотренная действием 2.2.

Проверяется, нет ли полного различия между соответствующими границами кадра и фона. Если такое соответствие имеется, то считаем, что изображение больше размера кадра и полностью не попало в кадр и необходимо отдалить камеру, после чего вновь осуществить проверку 2.1, иначе устройство захвата не активируется.

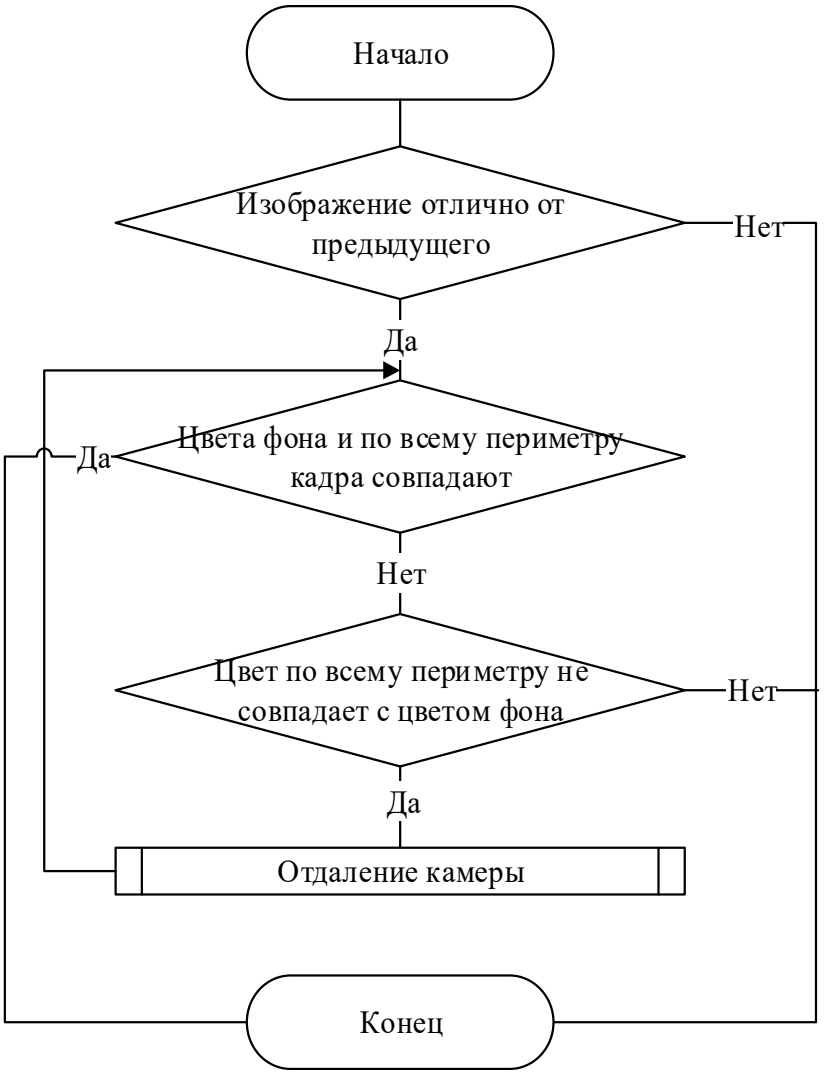


Рисунок 4.6 – Алгоритм проверки необходимости активации устройства захвата

После активации устройства захвата получаем исследуемый кадр. Для проверки «правильности» наводки объектива камеры на резкость используются известные алгоритмы автофокусировки.

Для получения самого латентного изображения необходимо отделить его от фона на исследуемом кадре.

После отделения латентного изображения от фона необходимо произвести коррекцию яркости и контрастности изображения, после чего полученное изображение становится пригодным для исследования на предмет подлинности.

При отсутствии в области видимости устройства захвата пригодных для исследования объектов, программа должна проверить на наличие в целевых каталогах устройств хранения, локальных сетевых ресурсов или облачных хранилищ, записанных с момента предыдущей проверки фалов с расширениями *.jpg, *.jpe, *.jpeg, *.bmp, *.rle, *.png, *.tif, *.tiff.

Алгоритм решения подзадачи получения изображения представлен на рисунке 4.7.

Рисунок 4.7 – Алгоритм получения изображения

4.2.2 Программный модуль «Построение комбинированных фильтров для выявления латентных изображений»

Для решения подзадачи определения подлинности изображения необходима база контрольных скрытых изображений, для наполнения которой создан программный модуль «Построение комбинированных фильтров для выявления латентных изображений».

Интерфейс диалогового окна «Фильтры» программного модуля (рисунок 4.8) содержит в себе следующие элементы:

– выпадающий список «Вид ЛИ», где пользователь определяет вид латентного изображения. В программном комплексе в файле «Filters.ini» по умолчанию добавлены комбинированные фильтры следующих видов латентных изображений: RussianRuble – Российский рубль, BelarusRuble – Белорусский рубль, Japan – Японская иена, DiplomRF – Дипломы РФ, Passport – Паспорт РФ, AzerbaijanManat – Азербайджанский манат, которые представляют собой последовательность линейных фильтров;

– выпадающий список «Наименование фильтра» предоставляет доступ пользователю к такому набору линейных фильтров как: EdgeEnhance, EdgeEnhanceMod, Average, EmbossColorModify, EdgesWeakModify, BlurBarlett, EmbossColor, EdgesWeak, Blur, Glow, Waggle, Null и т.д. (Подробнее линейные фильтры и их маски описаны в таблице – линейные фильтры, этапа «Проработка теоретических основ построения архитектуры, определение алгоритмов, методик, выбор инструментов разработки»);

– область «Проверка комбинированного фильтра», предназначена для предварительного просмотра результата работы выбранного комбинированного фильтра на исследуемом образце;

– кнопка «Загрузить исследуемое изображение» предназначена для выбора исследуемого образца, то есть изображения содержащего скрытую информацию;

– кнопка «Применить» – подтверждает выбор необходимой комбинации фильтров для исследуемого образца. По ее нажатию и в процессе выполнения

команды наименование кнопки меняется на «Идет обработка» в соответствии с текущем статусом выполняемой операции.

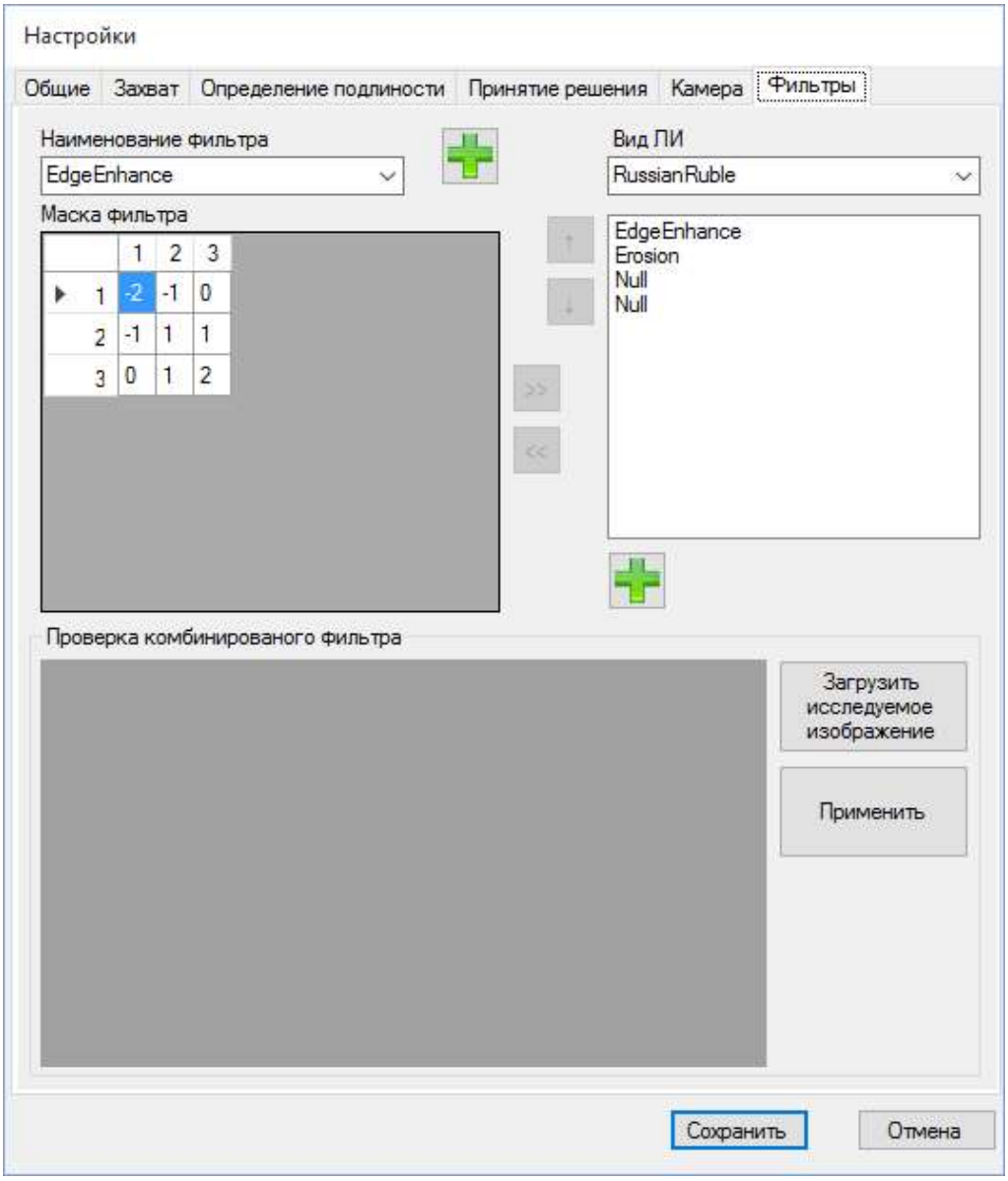


Рисунок 4.8 – Настройки программного комплекса: активная вкладка «Фильтры»

Построение комбинированного фильтра осуществляется методом подбора.

4.2.3 Программный модуль «Построение и хеширование контрольного скрытого объекта в латентном изображении»

Для решения подзадачи определения подлинности изображения необходима база контрольных скрытых изображений, для наполнения которой создан

программный модуль «Построение комбинированных фильтров для выявления латентных изображений».

Для добавления в базу контрольных изображений в программном комплексе существует инструмент «Настройка базы контрольных изображений». По нажатию на кнопку «Настройка базы контрольных изображений» открывается окно, представленное на рисунке 4.9, для занесения контрольных изображений.

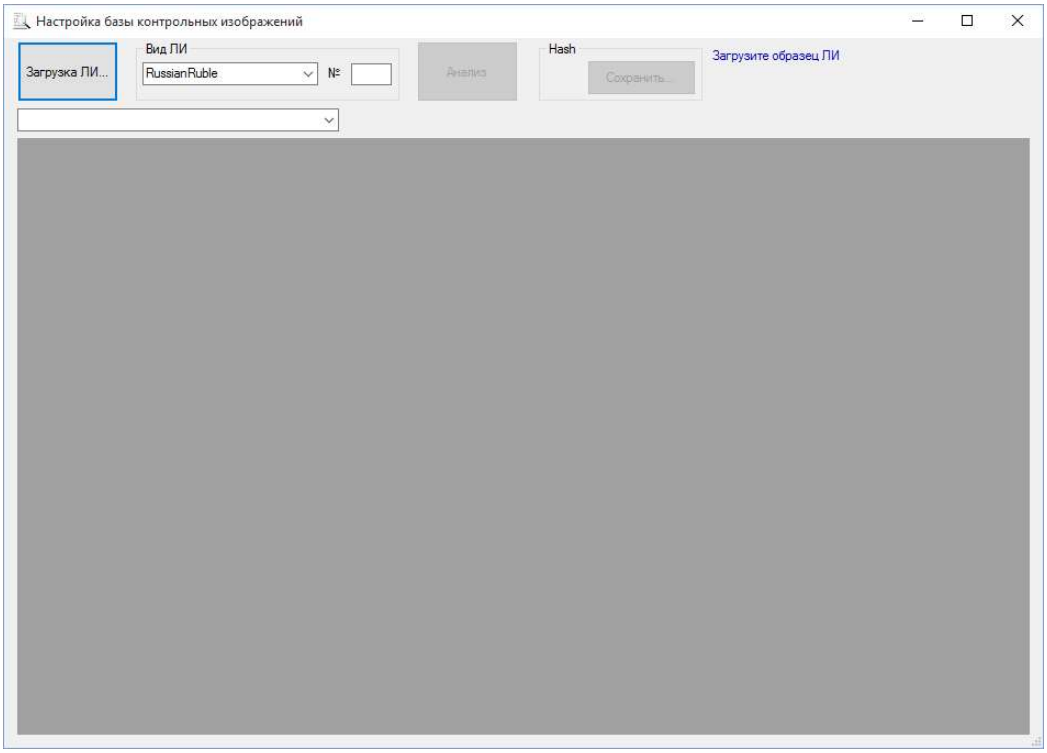


Рисунок 4.9 – Настройка базы контрольных изображений

Контрольные изображения – это объекты, которые выделяются в результате анализа на изображениях, взятых в качестве эталонных, и представляют собой 64-битную двоичную хэш-последовательность, используемую в дальнейшем для сравнения исследуемых изображений с эталонными в модуле «Определение подлинности документа».

Интерфейс диалогового окна «Настройка базы контрольных изображений» (рисунок 4.9) содержит в себе следующие элементы:

– кнопка «Загрузка ЛИ…», предназначенная для загрузки в рабочую область образцов для дальнейшего анализа;

– выпадающий список «Вид ЛИ», где пользователь определяет вид латентного изображения. В программном комплексе в файле «Control.ini» по умолчанию добавлены хэш-последовательности следующих видов латентных изображений: RussianRuble – Российский рубль, BelarusRuble – Белорусский рубль, Japan – Японская иена, DiplomRF – Дипломы РФ, Passport – Паспорт РФ, AzerbaijanManat – Азербайджанский манат;

– поле идентификатора эталонного изображения «№». Идентификатор контрольных изображений генерируется автоматически;

– кнопка «Анализ», предназначенная для применения коэффициентов детализации вейвлет-преобразования на анализируемом образце, а также комбинированных пространственных фильтров;

– окно предварительного просмотра хэш-последовательности изображения, появляется после того как пользователь программного комплекса делает нажатие левой кнопкой мыши по распознанному объекту;

– кнопка «Сохранить», для сохранения хэш-последовательности в файл базы контрольных изображений «Control.ini» с автоматически сгенерированным идентификатором;

– текстовое сообщение со справочной информацией о текущем состоянии процесса настроек базы контрольных изображений, например, «Загрузите образец ЛИ», «Запустите анализ изображения», «Идет вычисление горизонтальных коэффициентов», «Выявление завершено. Укажите выявленное изображение для построения контрольного» и т. д.;

– выпадающий список с перечнем выявленных изображений.

4.2.4 Программные модули «Автоматическое определение объекта по распознанному латентному изображению» и «Принятие решения»

Алгоритм решения подзадачи определения подлинности документа зависит от того известен ли изначально тип исследуемого латентного изображения или нет.

Определение подлинности конкретного типа ЛИ. Для выявления скрытого изображения в латентном из базы комбинированных фильтров автоматически выбирается фильтр, соответствующий указанному в конфигурационном файле типу ЛИ, и производится фильтрация.

Качество полученного предположительно скрытого изображения улучшается путем бинаризации и скрытое изображение сохраняется в рабочей директории.

Скрытое изображение с улучшенным качеством сравнивается с эталоном из БД контрольных образцов соответствующим данному типу латентного изображения. При уровне соответствия скрытого изображения с эталоном выше порогового исследуемый образец признается предположительно подлинным, иначе – предположительно поддельным.

Алгоритм определения подлинности конкретного типа ЛИ приведен на рисунке 4.10.

Рисунок 4.10 – Алгоритм определения подлинности конкретного типа ЛИ

Определение подлинности ЛИ неустановленного типа. Перебор всех возможных вариантов комбинированных фильтров может занять значительное время, поэтому, для ускорения выявления следует воспользоваться универсальным методом быстрого выявления скрытого изображения, основанный на вейвлет-анализе. Данный метод позволяет выявлять скрытую информацию из латентных изображений, полученных с помощью различных способов встраивания скрываемой информации.

Универсальный метод быстрого выявления скрытого изображения заключается в визуализации диагональных коэффициентов, полученных вейвлет-преобразованием с использованием биортогонального фильтра 2.8. Алгоритм данного метода заключается в следующем:

1. Латентное изображение преобразуется в оттенки серого.

2. Производится свертка изображения по строкам через одну с использованием ВЧ-фильтра декомпозиции с использованием биортогонального фильтра *bior2.8*.

Производится свертка полученного изображения по столбцам через один с использованием ВЧ-фильтра декомпозиции биортогонального вейвлета 2.8.

Алгоритм быстрого выявления скрытого изображения приведен на рисунке 4.11.

Рисунок 4.11 – Алгоритм быстрого выявления скрытого изображения

Полученное по средствам вейвлет-анализа предположительно скрытое изображение улучшается путем бинаризации и скрытое изображение сохраняется в рабочей директории.

Скрытое изображение с улучшенным качеством сравнивается с эталоном из БД контрольных образцов соответствующим данному типу латентного изображения. При уровне соответствия скрытого изображения с эталоном выше порогового исследуемый образец признается предположительно подлинным, иначе – начинается последовательная проверка фильтрами из базы комбинированных фильтров.

После наложения каждого фильтра производятся следующие действия:

1. Качество полученного предположительно скрытого изображения улучшается путем бинаризации и скрытое изображение сохраняется в рабочей директории.

2. Скрытое изображение с улучшенным качеством сравнивается с эталоном из БД контрольных образцов соответствующим данному типу латентного изображения.

3. При уровне соответствия скрытого изображения с эталоном выше порогового исследуемый образец признается предположительно подлинным, и проверка заканчивается, иначе выбирается следующий фильтр из базы комбинированных фильтров и накладывается на исследуемый образец. В случае, если достигнут конец базы комбинированных фильтров исследуемый образец признается предположительно поддельным.

Алгоритм определения подлинности конкретного типа ЛИ приведен на рисунке 4.12.

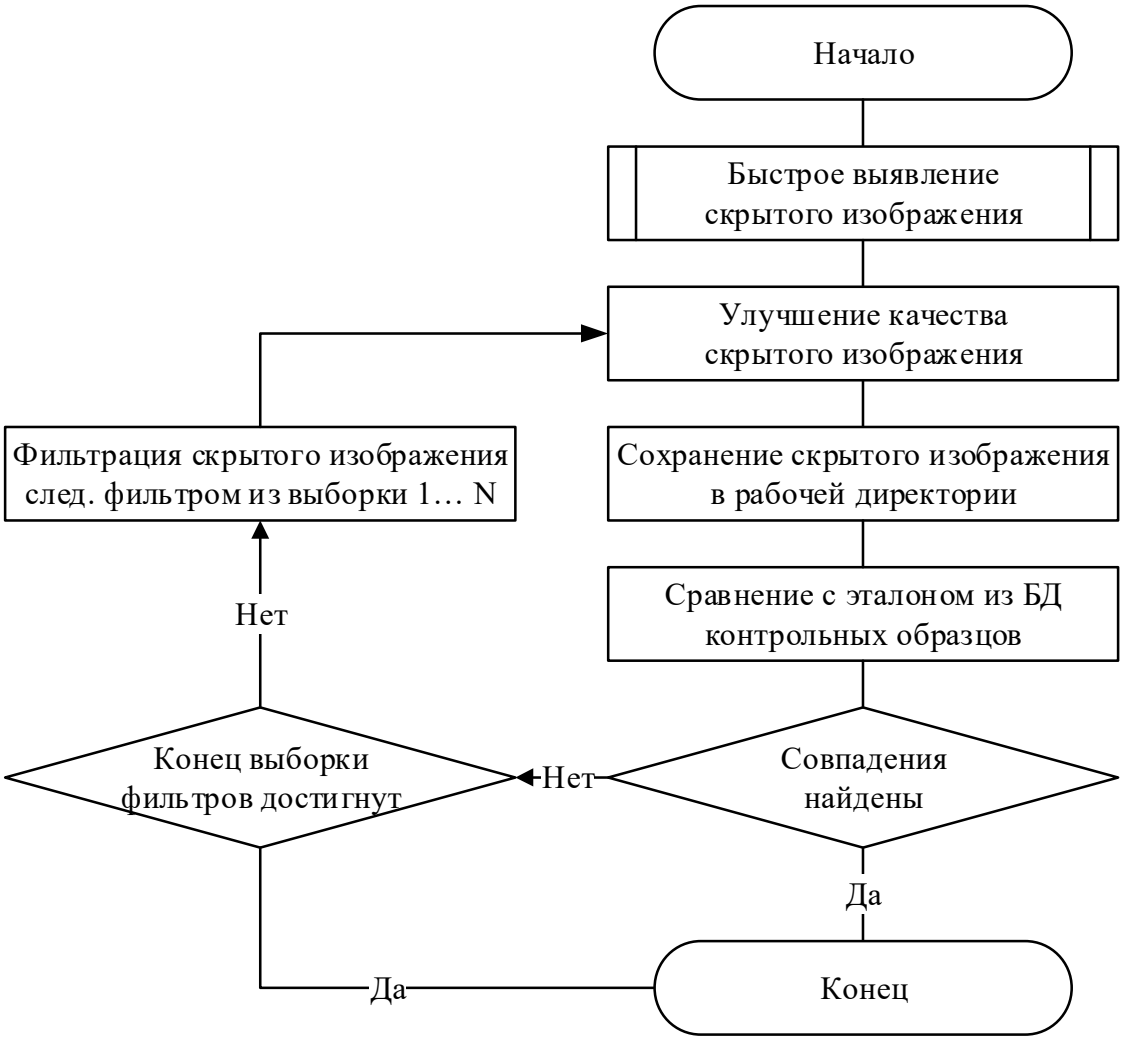


Рисунок 4.12 – Алгоритм определения подлинности ЛИ неустановленного типа

Общий алгоритм определения подлинности документа приведен на рисунке 4.13.

Для решения подзадачи принятия решения о последующих действиях необходима серия предположительно выявленных скрытых изображений с использованием различных фильтров и результат проверки подлинности.

Принятие решения о последующих действиях, выполняемых аппаратным комплексом после определения подлинности документа, возможно двумя способами:

1. Автоматически.
2. С подтверждением оператора.

В случае автоматического принятия решения, оно принимается на основании результата проверки подлинности. Если результат проверки положительный – исследуемый образец признается подлинным, иначе – нет.

В случае принятия решения с подтверждением оператора, оператору на дисплей выводится серия предположительно выявленных скрытых изображений с использованием различных фильтров и результат проверки подлинности. Оператор на основании выведенной на экран информации принимает окончательное решение о подлинности документа.

В зависимости от того, признан ли документ подлинным или нет выполняются заранее определенные последовательности

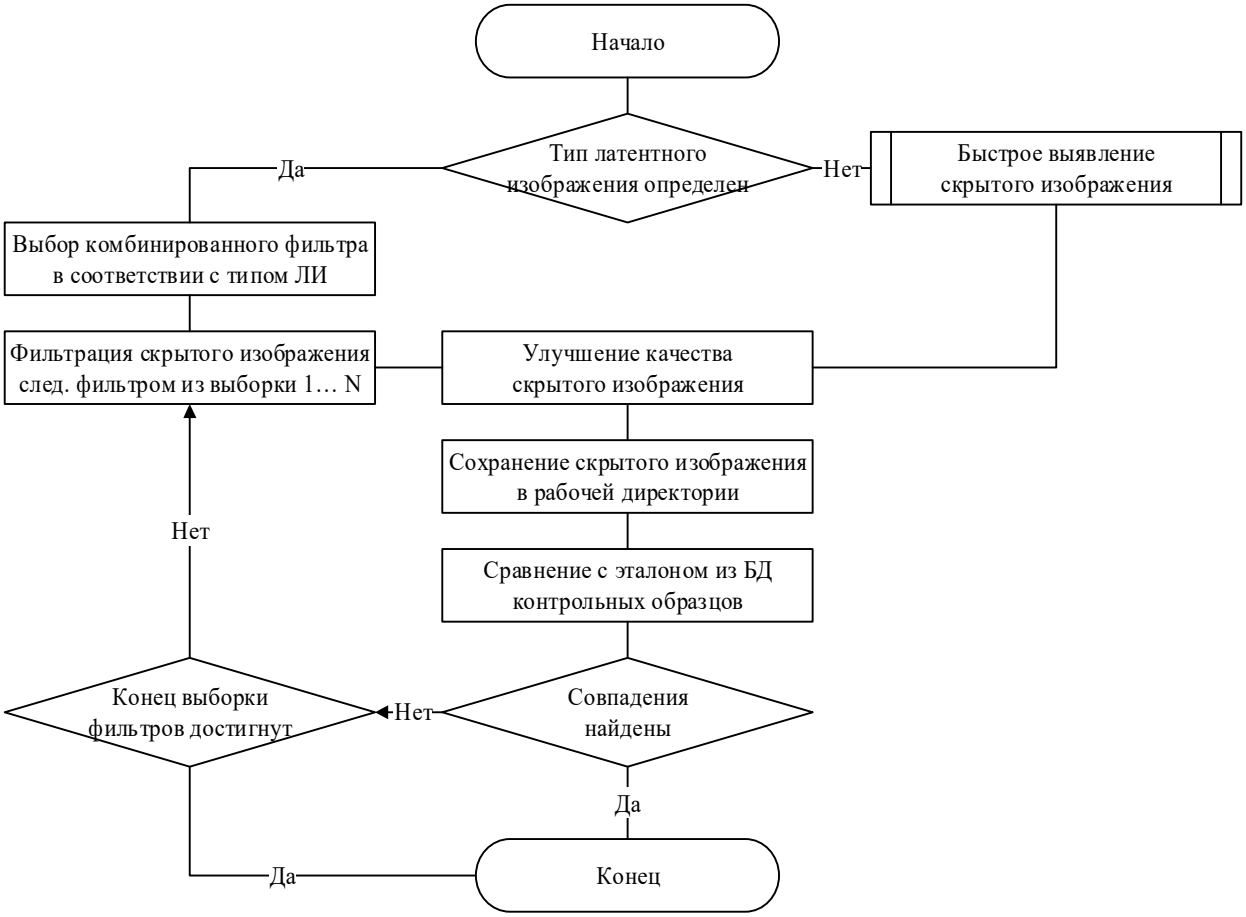


Рисунок 4.13 – Алгоритм определения подлинности документа

действий и результат заносится в БД результатов исследований.

Алгоритм решения подзадачи принятия решения представлен на рисунке 4.14.

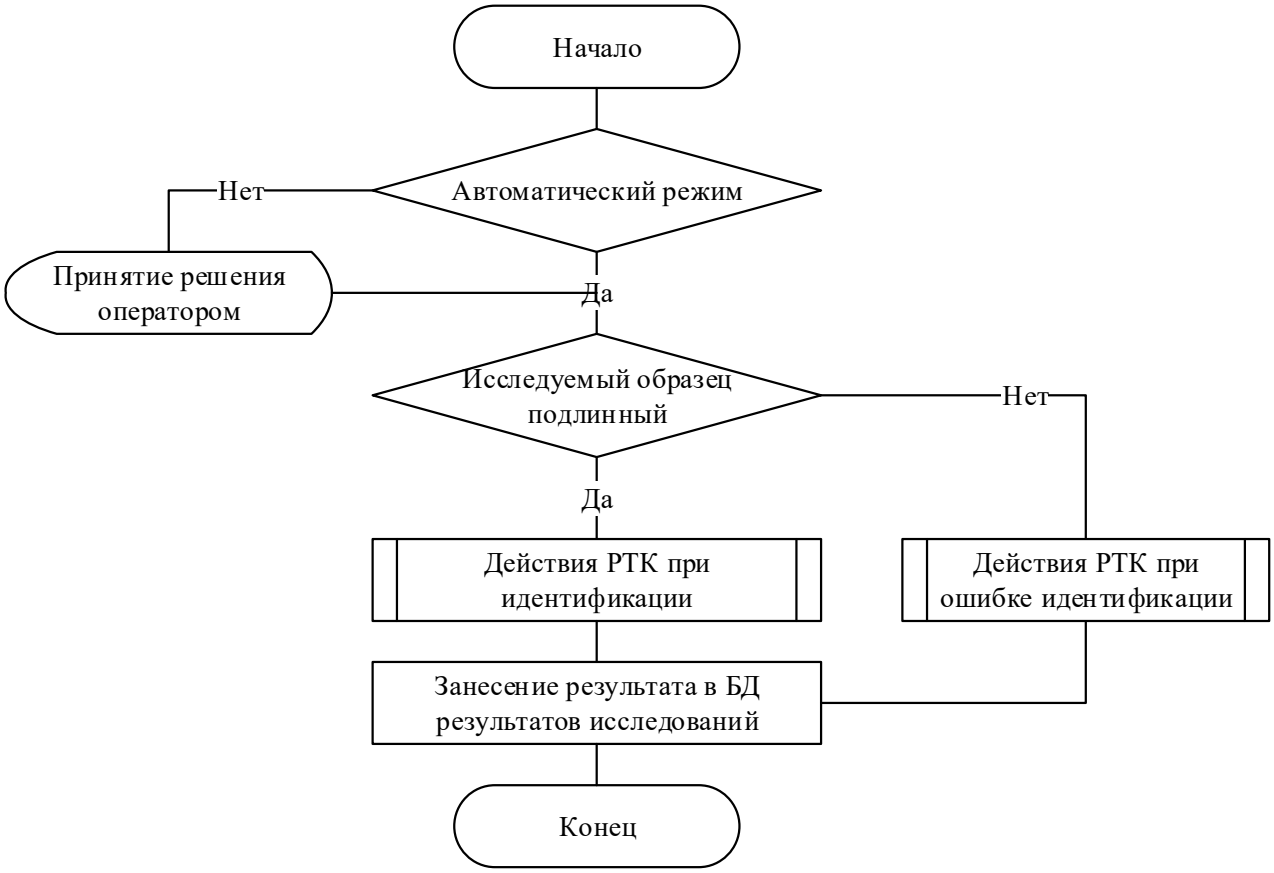


Рисунок 4.14 – Алгоритм принятия решения

4.3 Тестирование разработанных методов и алгоритмов контроля латентных изображений

Для проверки работы программы в режиме реального времени, с проверкой промежуточных результатов оператором, анализировались объекты следующих типов: билеты банка России достоинством в 1000 рублей (модификация 2010г.), полученная из тестового видео, билеты банка России достоинством в 1000 рублей (модификация 2004г.), банкноты банка Азербайджана достоинством в 5 манат, билеты банка Республики Беларусь достоинством в 50 рублей, паспорта граждан Российской Федерации.

После запуска мониторинга начинается захват изображения как описано в подразделе 4.2.1 Программный модуль «Конвейерный захват латентных изображений». Первое полученное для анализа изображение – изображение с

камеры: билет банка России номиналом 1000 рублей – было передано на выявление (рисунок 4.15).

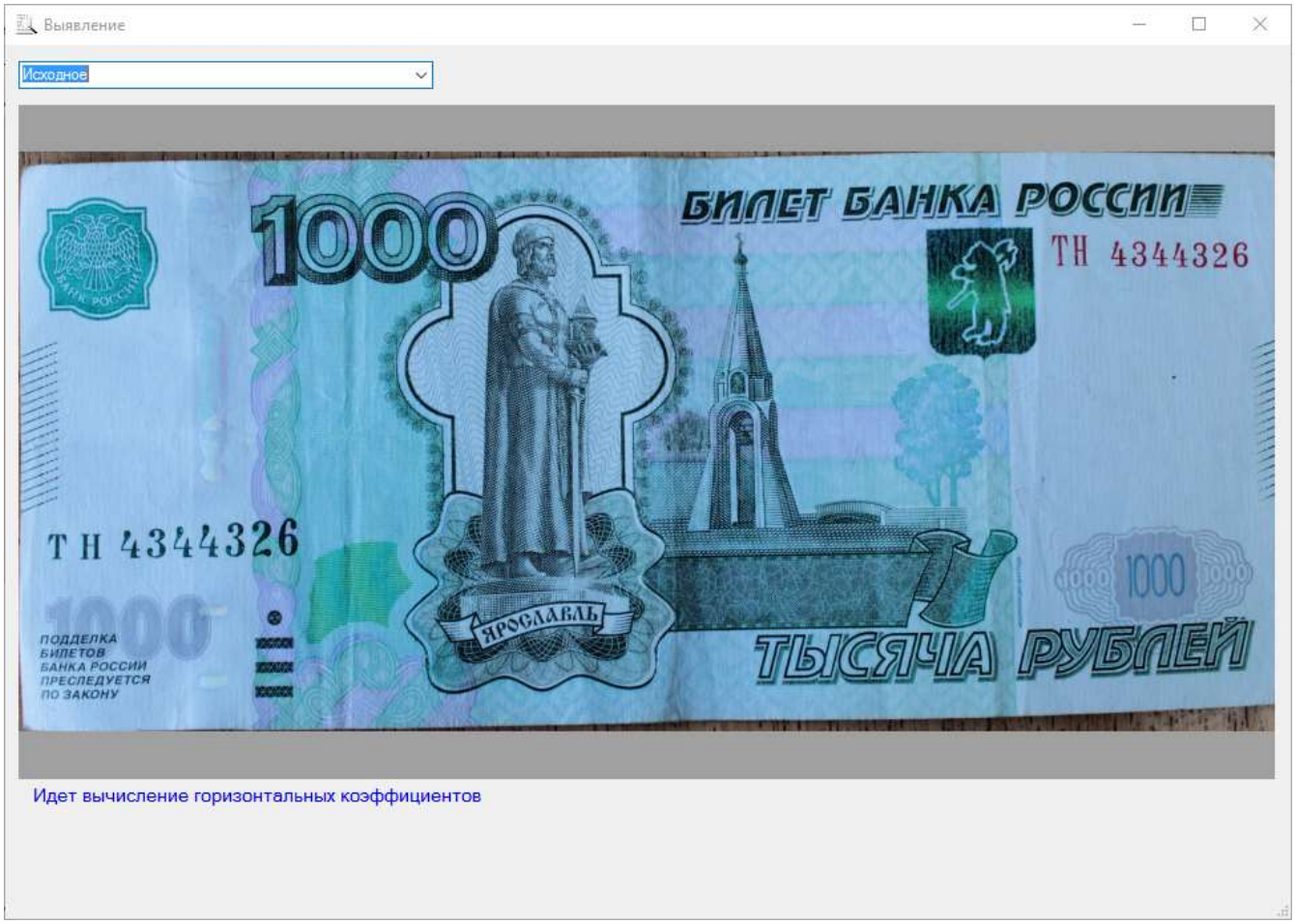


Рисунок 4.15 – Объект, захваченный с камеры анализируется

По завершению анализа система выдала сообщение оператору: «На исследуемом изображении обнаружены скрытые изображения, относящиеся к следующим объектам: RussianRuble» (рисунок 4.16).

У оператора появляется возможность просмотреть обработанные различными способами изображения с помощью селектора над областью с изображением. После чего принять окончательное решение о правильности автоматического распознавания нажатием на кнопку «ДОПУЩЕНО» или неправильности – нажатием на кнопку «НЕ ДОПУЩЕНО» (рисунок 4.17).

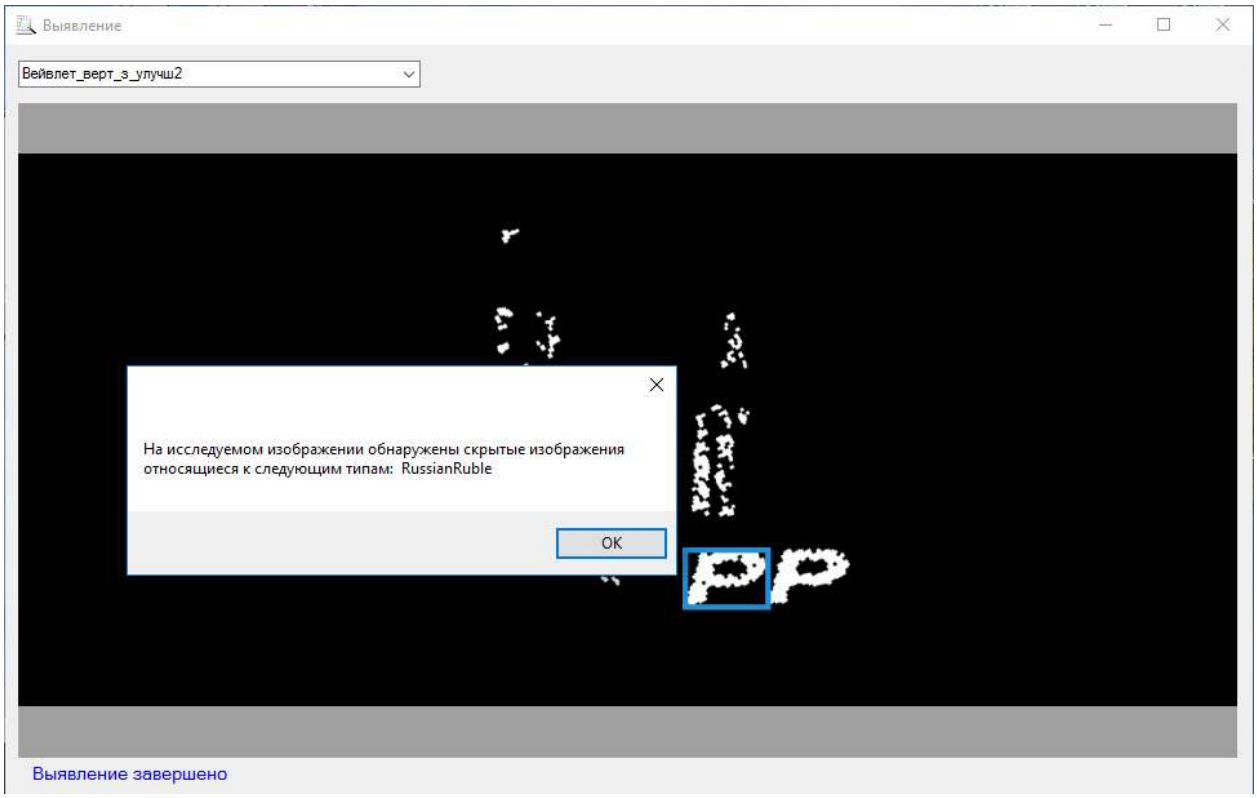


Рисунок 4.16 – Результат автоматического распознавания скрытого изображения

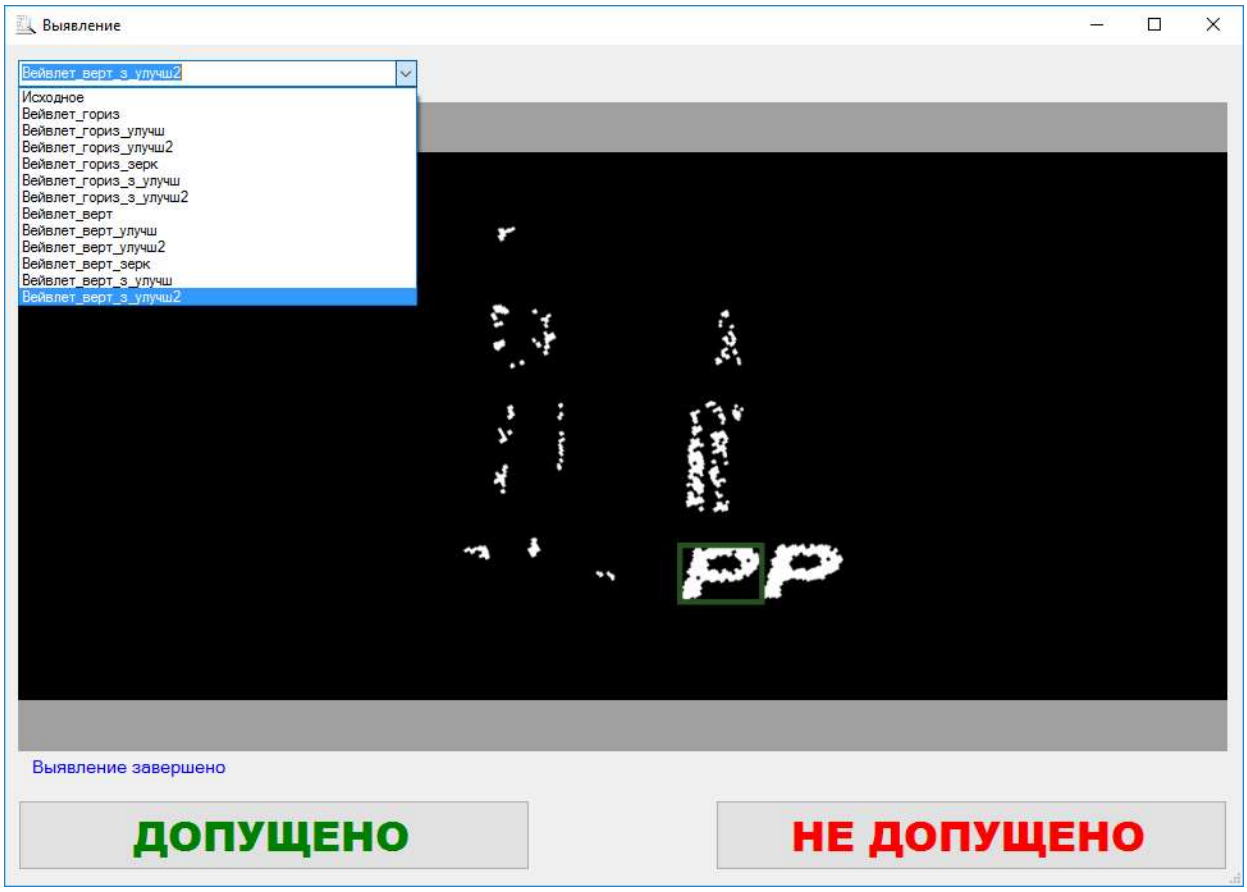


Рисунок 4.17 – Результат автоматического распознавания скрытого изображения

После нажатия на кнопку «ДОПУЩЕНО» осуществился возврат в модуль захвата изображения.

Следующее изображение, поступившее с цифрового хранилища – билет банка России номиналом 1000 рублей (модификация 2004г.). Начался процесс анализа данного образца (рисунок 4.18).

По завершению анализа система выдала сообщение оператору: «На исследуемом изображении обнаружены скрытые изображения, относящиеся к следующим объектам: RussianRuble» (рисунок 4.19).

У оператора появляется возможность просмотреть обработанные различными способами изображения с помощью селектора над областью с изображением. После чего принять окончательное решение о правильности автоматического распознавания аналогично предыдущему образцу. Была нажата кнопка «ДОПУЩЕНО»

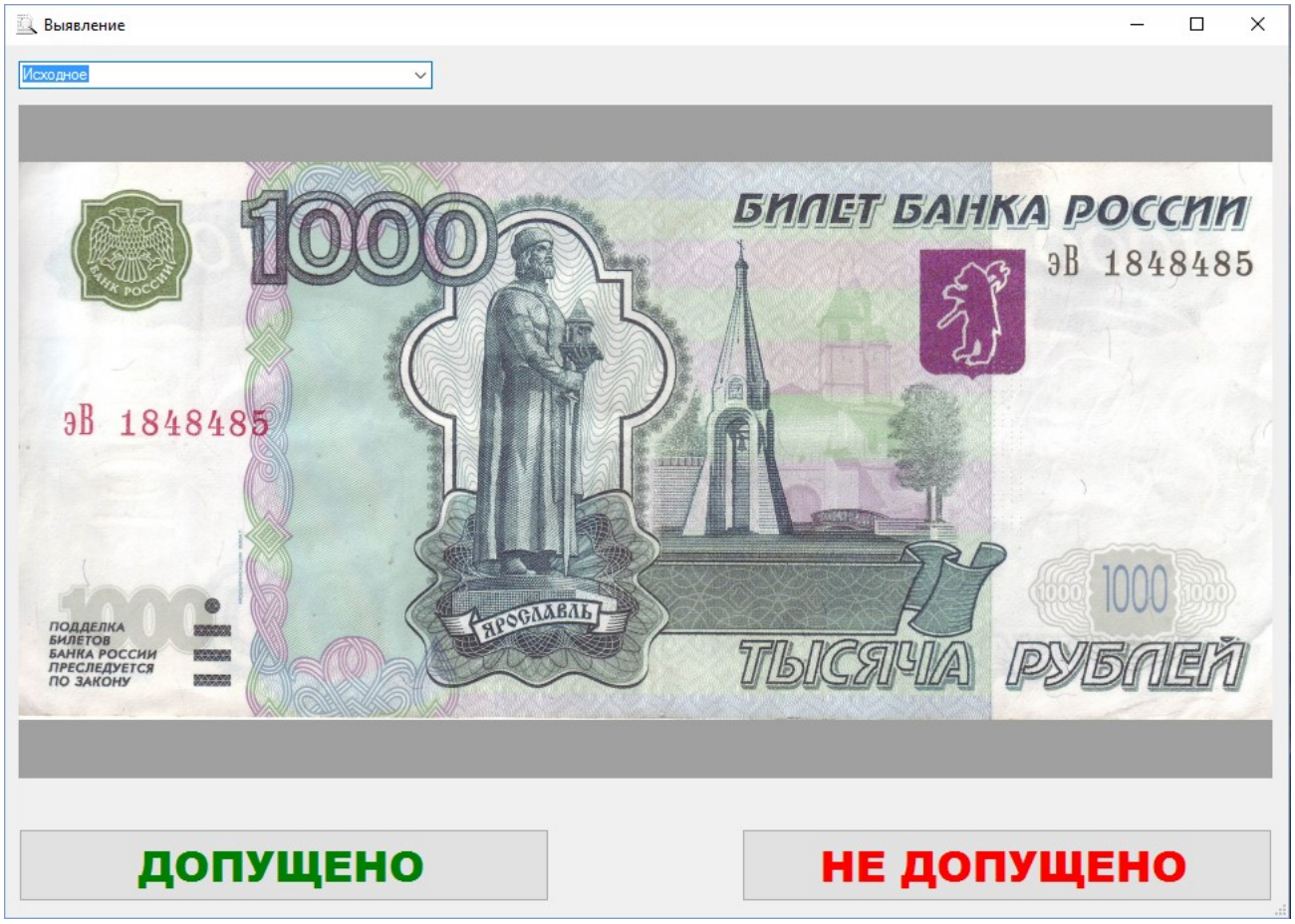


Рисунок 4.18 – Анализ копии банкноты «1000 российских рублей», полученной из цифрового хранилища

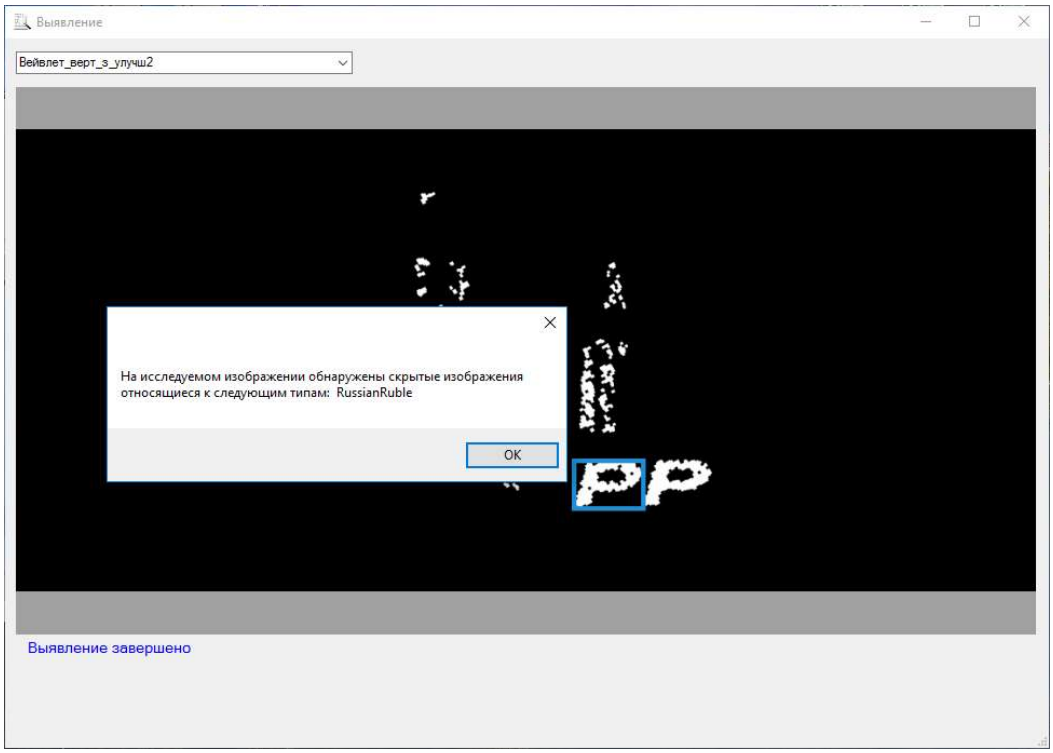


Рисунок 4.19 – Окно с выявленным скрытым изображением на банкноте 1000 российских рублей, полученной из цифрового хранилища

После нажатия на кнопку «ДОПУЩЕНО» осуществился возврат в модуль захвата изображения. Через секунду модуль захвата оказался не активен произошел захват следующего изображения: банкнота Азербайджана номиналом 5 манат. Начался процесс анализа данного образца (рисунок 4.20).

По завершению анализа система выдала сообщение оператору: «На исследуемом изображении обнаружены скрытые изображения, относящиеся к следующим объектам: AzerbayjanManat_5» (рисунок 4.21).

У оператора появляется возможность просмотреть обработанные различными способами изображения с помощью селектора над областью с изображением. После чего принять окончательное решение о правильности автоматического распознавания аналогично предыдущему образцу. Была нажата кнопка «ДОПУЩЕНО».

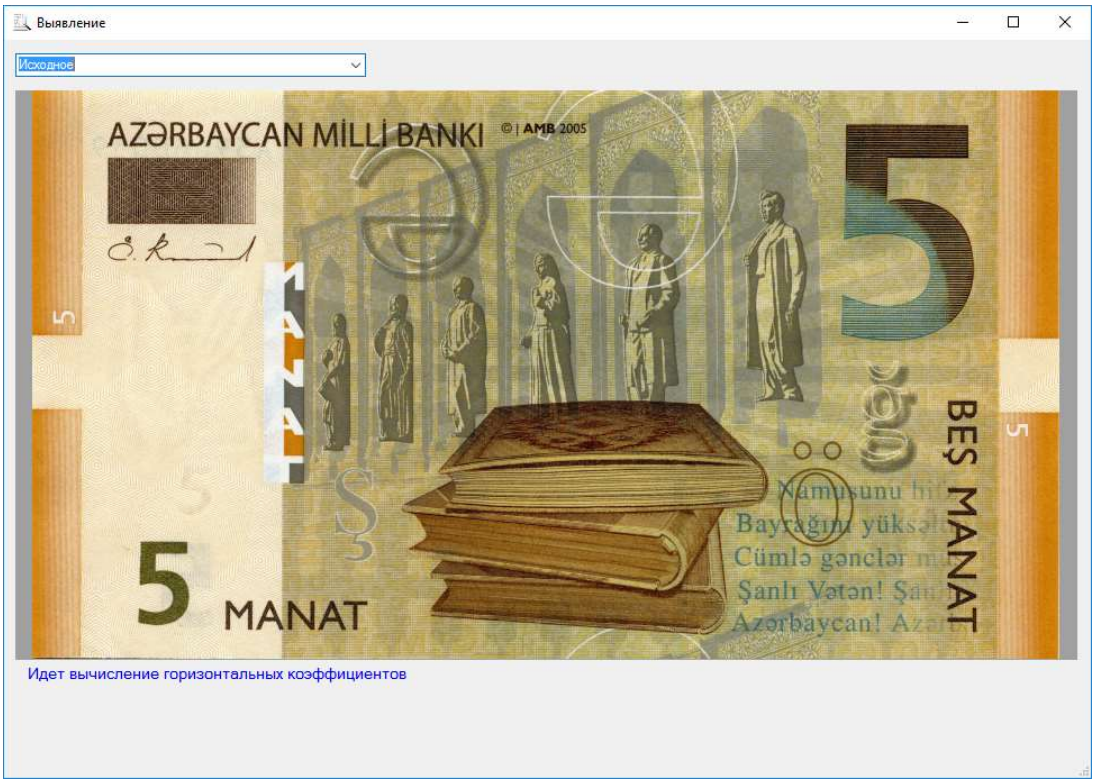


Рисунок 4.20 – Анализ копии банкноты «5 азербайджанских манат», полученной из цифрового хранилища

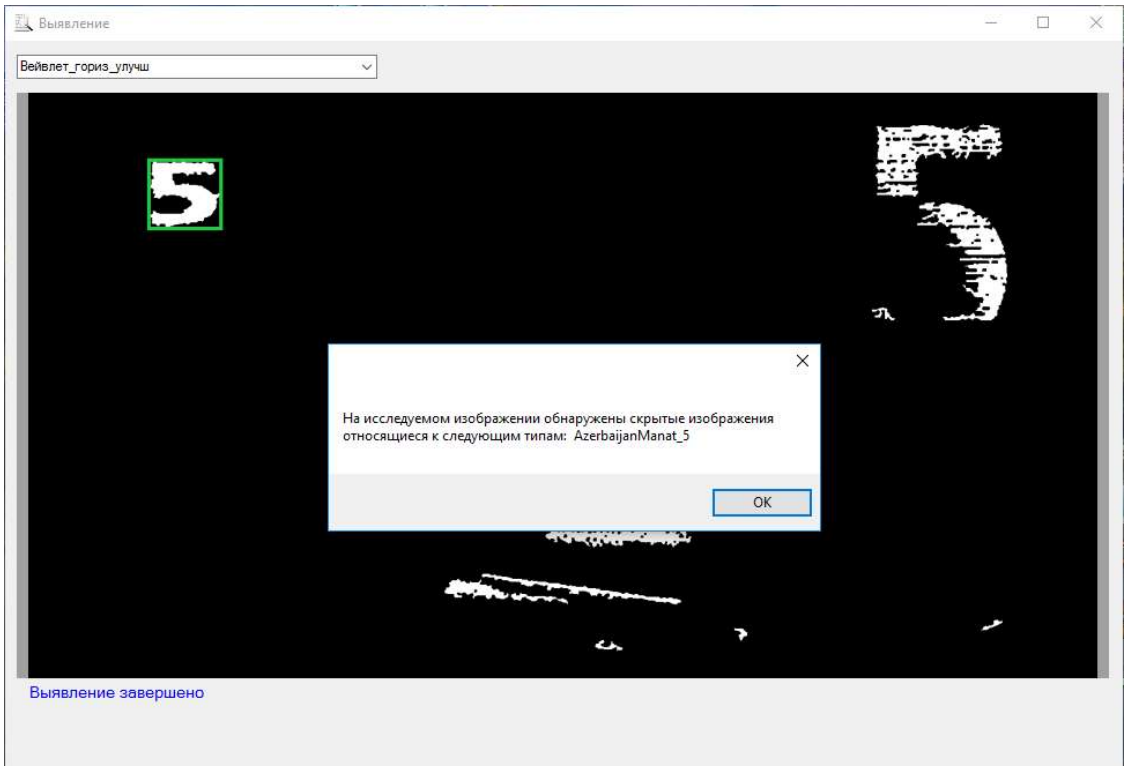


Рисунок 4.21 – Окно с выявленным скрытым изображением на банкноте 5 азербайджанских манат, полученным из цифрового хранилища

После нажатия на кнопку «ДОПУЩЕНО» осуществился возврат в модуль захвата изображения. Через секунду модуль захвата оказался не активен произошел захват следующего изображения: билет банка республики Беларусь номиналом 50 рублей. Начался процесс анализа данного образца (рисунок 4.22).

По завершению анализа система выдала сообщение оператору: «На исследуемом изображении обнаружены скрытые изображения, относящиеся к следующим объектам: BelarusRuble» (рисунок 4.23).

У оператора появляется возможность просмотреть обработанные различными способами изображения с помощью селектора над областью с изображением. После чего принять окончательное решение о правильности автоматического распознавания аналогично предыдущему образцу. Была нажата кнопка «ДОПУЩЕНО».

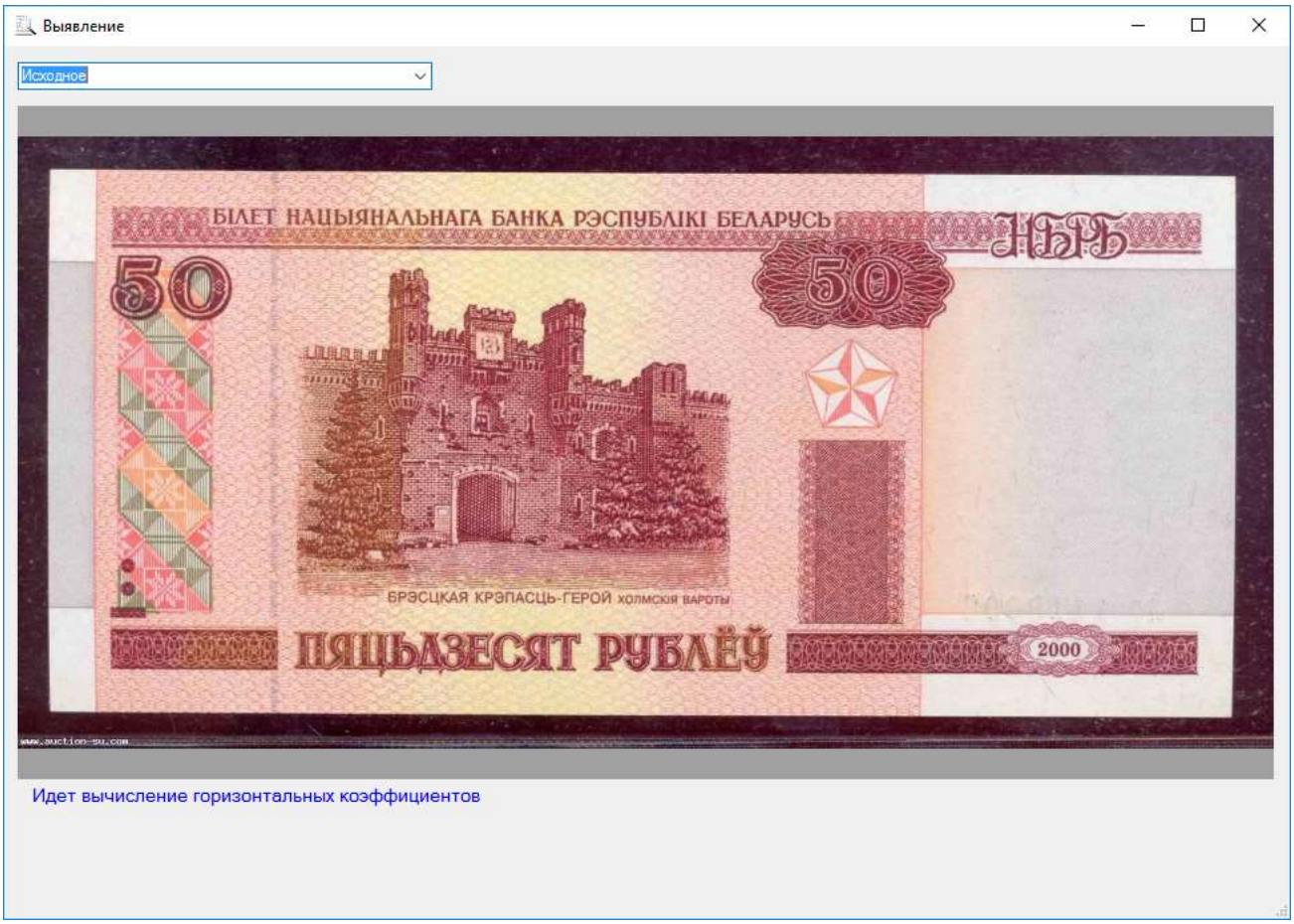


Рисунок 4.22 – Анализ копии банкноты «50 белорусских рублей», полученной из цифрового хранилища

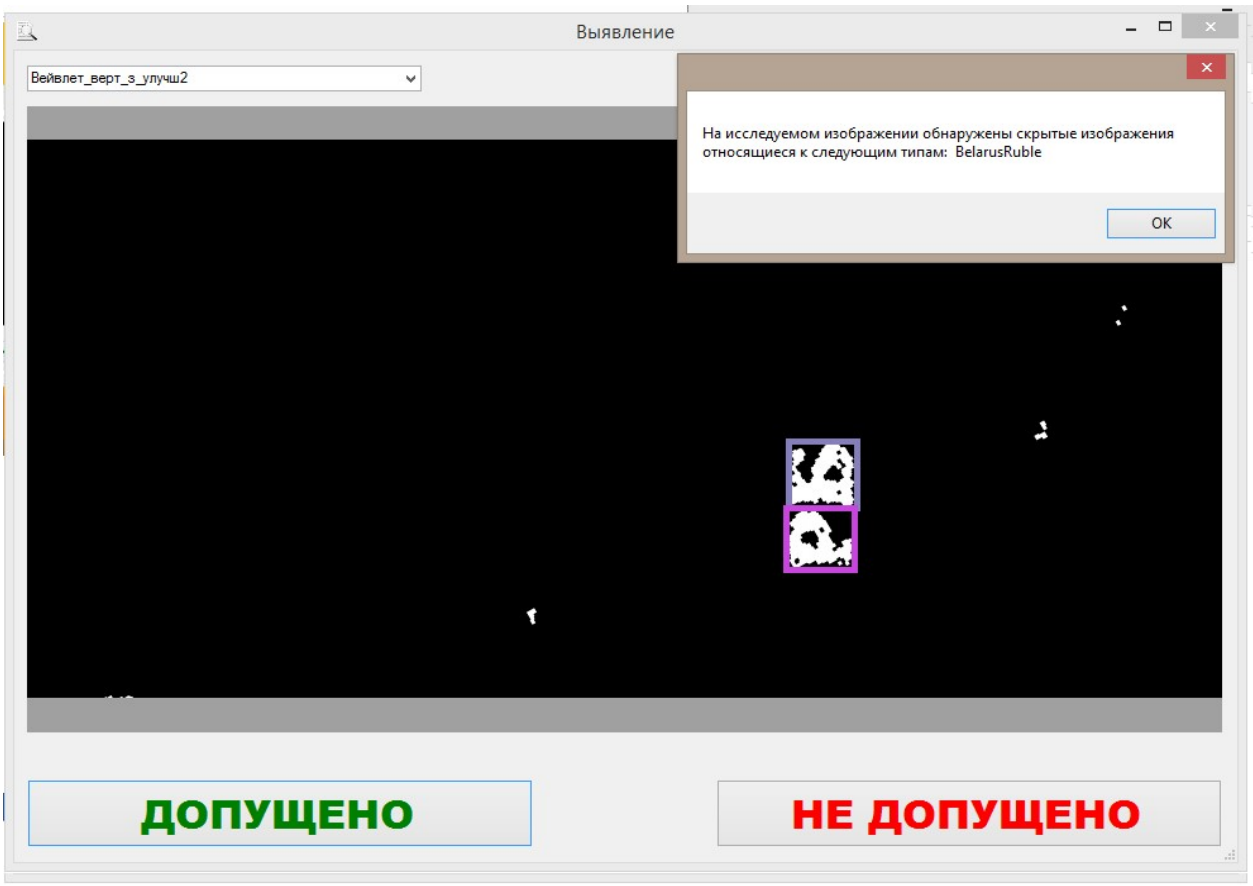


Рисунок 4.23 – Окно с выявленным скрытым изображением на банкноте 50 белорусских рублей, полученным из цифрового хранилища

После нажатия на кнопку «ДОПУЩЕНО» осуществился возврат в модуль захвата изображения. Через секунду модуль захвата оказался не активен произошел захват следующего изображения: паспорт Российской Федерации. Начался процесс анализа данного образца (рисунок 4.24).

По завершению анализа система выдала сообщение оператору: «На исследуемом изображении обнаружены скрытые изображения, относящиеся к следующим объектам: Passport» (рисунок 4.25).

У оператора появляется возможность просмотреть обработанные различными способами изображения с помощью селектора над областью с изображением. После чего принять окончательное решение о правильности автоматического распознавания аналогично предыдущему образцу. Была нажата кнопка «ДОПУЩЕНО».

Рисунок 4.24 – Анализ копии паспорта Российской Федерации, полученной из цифрового хранилища

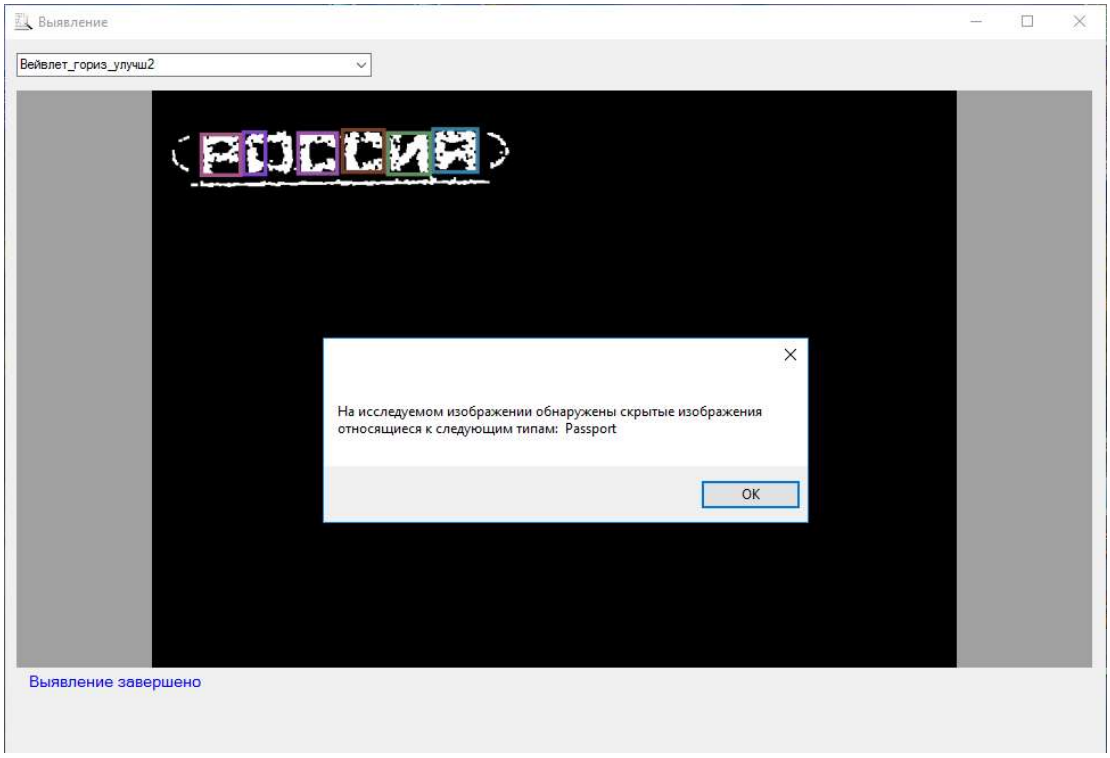


Рисунок 4.25 – Окно с выявленным скрытым изображением на паспорте Российской Федерации, полученным из цифрового хранилища

После нажатия на кнопку «ДОПУЩЕНО» осуществился возврат в модуль захвата изображения. Через секунду модуль захвата оказался не активен, однако образцов входящих цифровых ЛИ в каталоге не осталось, поэтому продолжился мониторинг. Проверка завершена, мониторинг остановлен, завершена работа программы.

Анализ результатов осуществлялся на основании данных из лог-файла и полученных в процессе работы программы изображений, полученных на различных этапах анализа (рисунок 4.26).

LatentDetect.log — Блокнот

Файл Правка Формат Вид Справка

```
13.07.2016 9-55-37  Запуск программы
13.07.2016 9-55-39  Установлен режим работы 2. Работа возможна только с тестовым видео
13.07.2016 9-55-40  Запущен мониторинг
13.07.2016 9-56-23  Мониторинг остановлен
13.07.2016 9-56-23  Изображение с камеры передано на обработку
13.07.2016 9-57-04  Автоматически распознано скрытое изображение типа  RussianRuble
13.07.2016 9-57-09  Оператор определил что изображение подлинное
13.07.2016 9-57-09  Запущен мониторинг
13.07.2016 9-57-14  Мониторинг остановлен
13.07.2016 9-57-14  Изображение с хранилища передано на обработку
13.07.2016 9-57-29  Автоматически распознано скрытое изображение типа   RussianRuble
13.07.2016 9-57-35  Оператор определил что изображение подлинное
13.07.2016 9-57-35  Запущен мониторинг
13.07.2016 9-57-40  Мониторинг остановлен
13.07.2016 9-57-41  Изображение с хранилища передано на обработку
13.07.2016 9-57-51  Автоматически распознано скрытое изображение типа   AzerbaijanManat_5
13.07.2016 9-57-59  Оператор определил что изображение подлинное
13.07.2016 9-57-59  Запущен мониторинг
13.07.2016 9-58-04  Мониторинг остановлен
13.07.2016 9-58-04  Изображение с хранилища передано на обработку
13.07.2016 9-58-23  Автоматически распознано скрытое изображение типа   BelarusRuble
13.07.2016 9-58-29  Оператор определил что изображение подлинное
13.07.2016 9-58-29  Запущен мониторинг
13.07.2016 9-58-34  Мониторинг остановлен
13.07.2016 9-58-34  Изображение с хранилища передано на обработку
13.07.2016 9-58-43  Автоматически распознано скрытое изображение типа   Passport
13.07.2016 9-58-47  Оператор определил что изображение подлинное
13.07.2016 9-58-47  Запущен мониторинг
13.07.2016 9-58-55  Мониторинг остановлен
13.07.2016 9-58-59  Завершена работа программы
```

Рисунок 4.26 – Лог-файл результатами исследования

Из лог-файла видно, что время выявления определения типа образца, в зависимости от разрешения тестового образца и числа проведенных итераций (от

2 до 12) составило от 3 до 41 секунд. Время на выполнение одной итерации 1-3 секунды.

4.4 Выводы

Разработанный программный комплекс системы распознавания латентных изображений позволяет проводить оперативный контроль латентных изображений, обеспечивающий высокое визуальное качество выявленного скрытого изображения, за счет использования вейвлет-анализа изображений и базы комбинированных фильтров, представляющих собой оптимальные последовательности фильтров, зависящие от способа формирования латентного изображения.

Разработанный программный комплекс решает задачу автоматического распознавания латентных изображений, что позволит использовать этот элемент защиты печатной продукции как машиночитаемый.

Разработанный программный модуль формирования латентных изображений позволяет формировать латентные изображения комбинированным способом с внедренным цветным изображением высокого визуального качества в автоматическом режиме со значением ПОСШ выше на 8,5% по сравнению с известным способом.

ЗАКЛЮЧЕНИЕ

В диссертационной работе проведены исследования свойств человеческого зрения учитываемых при формировании латентных изображений. Показано, что существующие методы внедрения скрытой информации не учитывают всех свойств человеческого зрения, при внедрении цветных изображений, в латентных изображениях появляется инородная структура, свидетельствующая о факте внедрения.

Разработан метод моделирования латентных изображений и модели формирования латентных изображений с внедрением как черно-белых, так и цветных скрываемых изображений, позволяющие повысить визуальное качество латентных изображений на 8,5%, отличающаяся от известных, комбинацией частотных и пространственных преобразований исходного изображения.

Проведен анализ результатов выявления скрытой информации, с помощью различных базисных вейвлетов для построения математической модели на основе вейвлет-преобразований латентных изображений и разработки численного метода контроля этих изображений.

Определены критерии качества вейвлет-функции для решения поставленных задач, обеспечивающие различимость в визуализированных коэффициентах одних элементов при неразличимости других элементов изображения.

В результате проведенных вычислительных экспериментов установлено, что биортогональный вейвлет по базису 2.8 лучше всего подходит для решения задачи контроля латентных изображений, так как позволяет сегментировать объекты на группы с примерно равным числом элементов и минимальным числом элементов, одновременно входящих в разные группы.

При вейвлет-преобразовании латентных изображений, в которых сюжетное и скрытое изображение представлены текстурами с совпадающим периодом, а также изображений с низким качеством, скрытая информация не выявляется. Для

контроля таких изображений предложено перед визуализацией осуществить поворот данного изображения на определенный угол.

Предложен численный метод контроля латентных изображений, реализующий вейвлет-преобразование латентного изображения и позволяющий выявить скрытое изображение вне зависимости от конкретного метода её формирования с числом преобразований в 6 раз меньше, чем у известных методов, на основе математической модели выявления латентных изображений.

Разработанная в работе система формирования и контроля латентных изображений, на основе разработанных методов позволяет описать процессы от момента формирования до определения подлинности защищенного документа;

Алгоритмы и программный модуль для формирования латентного изображения, позволяющие автоматически формировать латентное изображение на основе заданных исходного и скрываемого изображений, отличающиеся от известных меньшим отклонением меры пикового отношения сигнал/шум от исходного в среднем на 8,5%;

Разработанный программный комплекс системы распознавания латентных изображений позволяет проводить оперативный контроль латентных изображений, обеспечивающий высокое визуальное качество выявленного скрытого изображения, за счет использования вейвлет-анализа изображений и базы комбинированных фильтров, представляющих собой оптимальные последовательности фильтров, зависящие от способа формирования латентного изображения.

Разработка алгоритмов выявления и программного комплекса для контроля латентных изображений, позволило сократить максимальное время анализа латентных изображений в 6 раз.

Разработанный программный комплекс решает задачу автоматического распознавания латентных изображений, что позволит использовать этот элемент защиты печатной продукции как машиночитаемый.

СПИСОК ЛИТЕРАТУРЫ

1. Леру, П. Визуальные продажи. Использование зрительных образов в продажах и презентациях / П. Леру, П. Корвин. – М.: Добрая книга, 2010. – 304с.

2. Новая концепция борьбы с контрафактной продукцией // Heidelberg News: журнал для клиентов. – 2011. – Вып. 271. – С. 54-56.

3. Гражданский кодекс РФ. Часть четвертая: [федер. закон: принят Гос. Думой: 24 ноября 2006г.: по состоянию на 29.08.2014]. – М.: ФИПС, 2014. – 310с.

4. Список программно-технических средств, прошедших испытание в Банке России и рекомендованных для применения кредитными организациями // Банк России: официальном сайт. Москва, 2000-2017. URL: http://www.cbr.ru/bank-notes_coins/devices/print.asp?file=tested_lockt_bank.htm (дата обращения 13.04.2015).

5. ГОСТ Р 54109 – 2010. Защитные технологии. Продукция полиграфическая защищенная. Общие технические требования. – Введен впервые; 30.06.2011. – М.: Стандартинформ, 2011. – 18 с.

6. Машиночитаемые защитные признаки банкноты Банка России // Банк России: официальном сайт. Москва, 2000-2017. URL: http://cbr.ru/bank-notes_coins/bank-notes/G1997/B5/5R_97_maket.pdf (дата обращения 13.04.2015).

7. Pfitzmann, B. Information Hiding Terminology, in Information Hiding / B. Pfitzmann // Proc. 1st Int. Workshop on Information Hiding. Lecture Notes in Computer Sci. – New-York: Springer-Verlag, 1996. – V. 1174. – P. 347-350.

8. Bender, W. Techniques for data hiding / W. Bender, B. Gruhl, N. Morimoto и др. // Ibm Systems Journal. – 1996. – Vol. 35. – № 3. – P. 313-336.

9. Чваркова, И.Л. Стеганографические методы скрытия информации в аудиоданных / И.Л. Чваркова // Электроника. – 2003. – № 11. – С. 54–56.

10. Грибунин, В.Г. Цифровая стеганография / В.Г. Грибунин, И.Н. Оков, И.В. Туринцев. – М.: Солон-Пресс, 2016. – 262 с.

11. Романцов, А.П. Комплекс лабораторных работ по стеганографии: учебное пособие / А.П. Романцов, В.С. Бугаев, М.А. Фролов; под ред. Заслуженного деятеля науки РФ д.т.н. проф. А. В. Петракова. – М.: РИО МТУСИ, 2005. – 92 с.

12. Завьялов, С.В. Стеганографические методы защиты информации: учеб.пособие / С.В. Завьялов, Ю.В. Ветров. – Спб.: Изд-во Политехн. ун-та, 2012. − 190 с.

13. Аграновский, А.В. Стеганография, цифровые водяные знаки и стеганоанализ / А.В. Аграновский, А.В. Балакин, В.Г. Грибунин и др. – М.: Вузовская книга, 2009. – 220 с.

14. Ramkumar, M. Data Hiding in Multimedia: PhD Thesis / M. Ramkumar. – New Jersey: Institute of Technology, 1999. – 72 p.

15. Гурин, А.В. Технологии встраивания цифровых водяных знаков в аудиосигнал / А.В. Гурин, А.А. Жарких, В.Ю. Пластунов: под общ. ред. А.А. Жарких. – М.: Горячая линия – Телеком, 2015. – 116 с.

16. Cox, I. J. A secure, robust watermark for multimedia / I. J. Cox, J. Kilian, T. Leighton и др. // Information hiding: first international workshop: Lecture Notes in Comp. Science. – 1996. – Vol. 1174. – P. 183–206.

17. Fridrich, J. Steganography in digital media: principles, algorithms, and applications / J. Fridrich. – Cambridge: Cambridge University Press, 2010. – 437 p.

18. Федосеев, В.А. Метод извлечения текстурных водяных знаков, основанных на вариации масштаба / В.А. Федосеев, В.А. Митекин // Проведение научных исследований в области обработки, хранения, передачи и защиты информации: всероссийская конференция (Ульяновск, 15 декабря 2009): сборник трудов. / отв. ред. Ярушкина Н.Г. – Ульяновск: изд-во «Ульяновский государственный технический университет», 2009. – С. 418-426.

19. Митекин, В.А. Метод обнаружения водяных знаков, основанных на визуально незаметном искажении текстур / В.А. Митекин, В.А.Федосеев / Интеллектуализация обработки информации (ИОИ-2010): материалы 8-ой

международной конференции (Кипр, Пафос, 17-24 октября 2010). – Москва: МАКС Пресс, 2010. – С. 363-366.

20. Sergeyev, V. Gabor Filter Based Attack on Printed Documents Protection Methods via Digital Watermarks / V. Sergeyev, V. Fedoseev, V. Mitekin // Intelligent Information Hiding and Multimedia Signal Processing: Eighth International Conference (18-July 2012, Piraeus-Athens, Greece): proceedings. – Piraeus-Athens: IEEE, 2012. – P. 265-268.

21. Шевелев, А.А. Создание латентных изображений с использованием стохастических растровых структур / А.А. Шевелев // Технологія і техніка друкарства. – 2009. – №1-2 (23-24). – С. 226-233

22. Способ формирования латентного изображения: патент на изобретение RUS 2337403: G 06 T 5 00,G 06 K 9 00 / Маккарти Л.Д., Свиджерс Г.Ф.; Правообладатель: Коммонвелс сайнтифик энд индастриал рисеч организейшен; дата регистрации 04.06.2004.

23. Goryaev, M.A. Two models of the latent image formation / M.A. Goryaev // IS&T's: 52nd Annual Conference (Savannah, GA, April 25-28, 1999): proceedings. – Savannah, 1999. – P. 11-13.

24. Неразрушающий контроль: Справочник: в 7 т. / Под общ.ред. В.В. Клюева. – М.: Машиностроение, 2004. – 736 с.

25. Оппенгейм, А. Цифровая обработка сигналов Изд. 2-е, испр. / А. Оппенгейм, Р. Шафер. – М.: Техносфера, 2007. – 856 с.

26. Aleshin, A.A. Software for scanned polygraphic products steganalysis / A.A. Aleshin, D.M. Bogomolov и др. // Pattern Recognition and Image Analysis: New Information Technologies (PRIA-9-2008): 9th International Conference (Nizhni Novgorod, October 15-19, 2008): conference proceedings. – Nizhni Novgorod, 2008. – Vol.1. – P. 3-6.

27. Горбачев, В.Н. Методы цифровой стеганографии для защиты изобразительной информации [Электронный ресурс] / В.Н. Горбачев, Е.М. Кайнарова, А. И. Кулик и др. // Uprint Image Processing Group – Режим доступа: http://uipg.ru/assets/103-Metody-cifrstg-dlya-zashity-2010.pdf, свободный.

28. ГОСТ Р 34.10 – 2012. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи. – Взамен ГОСТ Р 34.10 – 2001; введ. 07.08.2012. – М.: Стандартинформ, 2013. – 17 с.

29. Созведие Евриона и система CDS: первые рубежи защиты купюр // Банкноты стран мира. – 2008. – № 2. – С. 30-33.

30. Advanced Access Content System: Pre-recorded Video Book [Электронный ресурс] / Advanced Access Content System; Intel Corporation, Microsoft Corporation, Sony Corporation, Toshiba Corporation, The Walt Disney Company, Warner Bros. – Режим доступа: http://www.aacsla.com/specifications/specs091/AACS_Spec_Prerecorded_0.91.pdf.

31. Жмакин, М.О. Стеганография и перспективы ее применения в защите печатных документов / М.О. Жмакин // Безопасность информационных технологий. – 2010. – № 3. – С. 74-77.

32. Грегори, Р.Л. Глаз и мозг. Психология зрительного восприятия / Р.Л. Грегори. – М.: Прогресс, 1970. – 223 с.

33. Пэдхем, Ч. Восприятие света и цвета / Ч. Пэдхем, Дж. Сондерс: пер. с англ. Р. Л. Бирновой и М. А. Островского. – М.: изд-во «Мир», 1978. – 255 с.

34. Прокопенко, В.Т. Психология зрительного восприятия: учебное пособие / В.Т. Прокопенко, В.А. Трофимов, Л.П. Шарок. – Спб: СПбГУИТМО, 2006. – 73с.

35. Щеголев, И. О растровом выборе. Часть 2. Патетическая. Линиатура / И. Щеголев // Журнал «Компьюарт». – 2010. – №2. – С. 26-33.

36. Avcibas, I. Statistical evaluating of image quality measures / I. Avcibas, B. Sankur, K. Sayood // Journal of Electronic Imaging. – 2002. – Vol.11. – № 2. – P. 206-223.

37. Топчиев, И. Н. Математическое моделирование и программный комплекс для контроля магнитных полей и латентных изображений: дис. канд. … техн. наук: 05.13.18 / Топчиев Иван Николаевич. – Ставрополь, 2010. – 168 с.

38. Fridrich, J. Steganalysis of LSB Encoding in Color Images / J. Fridrich, R. Du, M. Long // Proceedings of ICME. – New York, 2000. – vol. 3. – P. 1279-1282.

39. Fridrich, J. A new steganographic method for palette-based image / J. Fridrich // Proceedings of ISBT: PISP conference (Savannah, GA, April 1989): proceedings. – Savannah, 1989. – P. 285-289.

40. Смоленцев, Н.К. Основы теории вейвлетов. Вейвлеты в MATLAB / Н.К. Смоленцев. – М.: ДМК Пресс, 2005. – 304с.

41. Конахович, Г.Ф. Компьютерная стеганография: Теория и практика / Г.Ф. Конахович, А.Ю. Пузыренко. – Киев: МК-Пресс, 2006. – 288 с.

42. Matsui, K. Video-steganography: how to secret embed a signature in a picture / K. Matsui, K. Tanaka // IMA intellectual properly project proceeding. – 1994. – Vol.1. – P. 187-205.

43. Darmstaedter, V. Low cost spatial watermarking / V. Darmstaedter, J.-F. Delaigle, J. Quisquater, B. Macq // Computers and graphics. – 1998. – Vol. 5. – P. 417-423.

44. Langelaar, G. Robust labeling methods for copy protection of images / G. Langelaar, R. Lagendijk, J. Biemond // Storage and retrieval for image and video databases: SPIE V (2, San Jose, California, U.S.A., Feb. 1997): proceedings. – San Jose, 1997. – vol. 3022. – P. 298-309.

45. Zhao, J. Embedding robust labels into images for copyright protection / J. Zhao, E. Koch // Intellectual property rights for specialized information, Knowledge and new techniques: int. Congress (Munich-Viena, Verlag, aug.1995): proceeding. – Viena, 1995. – P. 242-251.

46. Hsu, C.-T. DCT-based watermarking for video / C.-T. Hsu, J.-L. Wu // transactions on consumer electronics: IEEE(Feb. 1998). – 1998. – vol. 44. – №1. – P. 206-216.

47. Hsu, C.-T. Multiresolution watermarking for digital images / C.-T. Hsu, J.-L. Wu // transactions on circuits and systems: IEEE. – 1998. – №45(8). – P. 1097-1101.

48. Chae, J.J. A robust embedded data from wavelet coefficients / J.J. Chae, D. Mukherjee, B.S. Manjunath // Storage and retrieval for image and video databases: SPIE: proceedings. – 1998. – Vol. 3312. – P. 308-317.

49. Chae, J. Robust techniques for data hiding in images and video: PhD thesis / J. Chae // department for electrical and computer engineering. – Santa Barbara, CA, USA: University of California, 1999. – 82 p.

50. Шведова, Н.Н. Вопросы методологии криминалистического исследования документов / Н.Н. Шведова // Судебная экспертиза. – 2015. – Т. 1. – № 1 (41). – С. 16-23.

51. Корочкин, Л.С. Материалы и методы защиты специальных бумаг и документов от подделки / Л.С. Корочкин. – Минск: НТУП «Криптотех», 2001. – 264 с.

52. Маресин, В.М. Защищенная полиграфия: Справочник / В.М. Маресин: 2-е изд., стер. – М.: ФЛИНТА, 2014 – 640 с.

53. Глумов, Н.И. Алгоритм извлечения скрытой информации из отсканированных полиграфических изделий / Н.И. Глумов, В.А. Митекин, А.В. Сергеев, В.А. Федосеев // Вестник СГАУ. – 2008. – № 2 (15). – С. 216-220.

54. Описание технологий защиты отпечатков [Электронный ресурс]: компьютерный журнал // КомпьютерПресс: сайт. – М., 2006. – Режим доступа: http://compress.ru/Article.aspx?id=16111, свободный. (Дата обращения: 25.11.2014).

55. Федоров Б.Ф., Цибулькин Л.М. Голография / Б.Ф. Федоров, Л.М. Цибулькин. – М.: Радио и связь, 1989. – 144с.

56. О голографическом торговом знаке [Электронный ресурс] // BH Group: сайт – Режим доступа: http://www.holo.ru/news/doc/9/1.html, свободный. (Дата обращения: 4.11.2014).

57. Элементы дизайна голограмм [Электронный ресурс] // Научно-технический центр «Прикладная компьютерная оптика»: сайт – Режим доступа: http://www.hgr.ru/index.php?show=prod/stickers/orig_design

58. Описание визуализатора DORS 25 [Электронный ресурс] // Компания «Система» : сайт – Режим доступа: http://www.systema.biz/590.html

59. Голография и упаковка [Электронный ресурс] // Информационный портал об упаковке: сайт – Режим доступа: http://www.upakovano.ru/articles/1769, свободный.

60. Валадов Д. О трафаретных защитных красках / Д. Валадов // Компьюарт. – 2011. – №6. – С. 24-27.

61. Increased-security identification card system: patent US 5284364: IC B42D 15/10, G06K 1/00, G06K 1/12, G06K 19/14, G06K 19/08, B42D 015/00 / Jain; Kanti (Briarcliff Manor, NY); Assignee: Anvik Corp.; declared 10.06.1992; published 8.02.1994, Appl. No. 07/897,199

62. Identification system comprising a partially reflective retardation device: patent US 4659112: IC B42D 15/10, G06K 19/14, B42D 015/00 / Daniel T. Reiner, Lawrence Bolt, Philip W. Morlan Jr., Ali Tavasolian; Assignee: Optical Devices, Inc; filed 3.12.1984; published 21.04.1987, Appl. No. 06/677,427

63. Polymer materials with latent images visible in polarized light and methods for their production: patent US 6124970: IC G02B 5/30, G06K 19/14, G02B 027/28, B42D 015/00 / Andrei Karassev, Anatoli Vannikov, Vladimir Kazarinov, Ludmila Karasseva; Assignee: Latents Image Technology Ltd. (Jerusalem, IL); filed 20.10.1997; published 26.09.2000, Appl. No. 08/953,992

64. Process for making photopolymers having varying molecular orientation using light to orient and polymerize: patent US 5389698: IC C08F 2/46, G02F 1/13, G02F 1/1337, G02B 5/30, C08F 002/46, G03C 001/492 / Vladimir Chigrinov, Vladimir Kozenkov, Nicolic Novoseletsky, Victor Reshetnyak, Yuriy Reznikov, Martin Schadt, Klaus Schmitt; Assignee: Hoffmann-La Roche Inc. (Nutley, NJ), Niopic Moscow Research and Production Association (Moscow, RU) ; filed 21.09.1993; published 14.02.1995, Appl. No. 08/125,006

65. Способ удостоверения подлинности предметов: пат. 2165360 Рос. Федерация: МПК^8 B42D15/10, B41M5/00 / Бондарев Л.А.; Куракин С.В.; Одиноков С.Б.; Маклаков В.В.; заявитель и патентообладатель ОАО "Концерн защитные российские технологии"; заявл. 24.02.2000; опубл. 20.04.2000

66. Monitoring of covert marks: patent US 6100967: IC B44C 1/00, B44C 1/02, B44B 7/00, B42D 015/00 / Richard Mark Farrer, Barry Alan Hood; Assignee: SLS Biophile Ltd. (GB) ; filed 05.08.1997; published 8.08.2000, Appl. No. 08/849,737

67. Способы защиты документов [Электронный ресурс] // ООО «ВИЛДИС». – Режим доступа: http://www.bnti.ru/showart.asp?aid=940&lvl=01.03.05.&p=1.

68. Описание программного продукта CERBER [Электронный ресурс] // SecuritySoft Co. Ltd. – Режим доступа: http://www.securesoft.ru/cerber.html, свободный. (Дата обращения: 25.11.2014).

69. Описание программного продукта BBS Designer [Электронный ресурс] // GuardSoft Ltd. – Режим доступа: http://www.guard-soft.com/designer.html, свободный. (Дата обращения: 25.11.2014).

70. Универсальный просмотровый детектор DORS 1300 M2 [Электронный ресурс] / Компания «DORS». – Режим доступа: http://dors.com/catalog/dors-1300/, свободный. (Дата обращения: 03.04.2016).

71. Приборы и системы для определения подлинности банкнот, документов, ценных бумаг [Электронный ресурс] // научно-производственное предприятие «ВИЛДИС». – Режим доступа: http://www.vildis.ru/product/ultramag-s6/, свободный. (Дата обращения: 03.04.2016).

72. Специальное программное обеспечение для работы с изображениями [Электронный ресурс] // Компания «Регула». – Режим доступа: https://regulaforensics.com/ru/products/software/, свободный. (Дата обращения: 03.04.2016).

73. Дьяконов, В.П. MATLAB 6/6.1/6.5 + Simulink 4/5. Основы применения: полное руководство пользователя / В.П. Дьяконов. – М.: СОЛОН-Пресс, 2002. – 768 с.

74. Мэтьюз, Д. Г. Численные методы. Использование MATLAB: пер. с англ. / Д. Г. Мэтьюз, К. Д. Финк. – М.: Изд. дом «Вильямс», 2001. – 713с.

75. Горбаченко, В.И. Вычислительная линейная алгебра с примерами на MATLAB / В.И. Горбаченко. – СПб.: БХВ-Петербург, 2011. – 320с.

76. Гонсалес, Р. Цифровая обработка изображений в среде MATLAB / Р. Гонсалес, Р. Вудс, С. Эддинс. – М.: Техносфера, 2006. – 616 с.

77. Поршнев, С.В. MATLAB 7. Основы работы и программирования: учебник / С.В. Поршнев. – М.: Бином. Лаборатория знаний, 2006. – 320 с.

78. Алексеев, Е.Р. MATLAB 7: Самоучитель. / Е.Р. Алексеев, О.В. Чеснокова. – М.: НТ Пресс, 2006. – 324 с.

79. Курбатова, Е.А. MATLAB 7: Самоучитель. / Е.А. Курбатова. – М.: Вильямс, 2006. – 256 с.

80. Смоленцев, Н.К. Основы теории вейвлетов в MATLAB. / Н.К. Смоленцев. – М.: ДМК, 2005. – 304 с.

81. Глушаков, С.В. Математическое моделирование: Mathcad 2000, MATLAB 5: Учебный курс / С.В. Глушаков, И.А. Жакин, Т.С. Хачиров. – Харьков.: Фолио, 2001. – 524 с.

82. Горелик, А.Л. Методы распознавания / А.Л. Горелик, В.А Скрипкин. – М.: Высшая школа, 1984. – 208с.

83. Шавард, Н. А. Экспресс-анализ подлинности специальных, акцизных, и идентификационных марок / Н. А. Шавард. – М.: «Вилдис»,1999. – 32 с.

84. Стариков, Е. Особенности исследования поддельных денежных билетов в России / Е. Стариков // Multi-Agent Systems and Application: 3rd International Central and Eastern European Conference (Prague, Czech Republic, June 16-18, 2003): proceedings. – Prague, 2003. – P. 97-101.

85. Жарких, А.А. Программный модуль для формирования латентного изображения / А.А. Жарких // Свидетельство о регистрации электронного ресурса №20205, Зарегистрировано в Фонде алгоритмов и программ 11 июля 2014 г.

86. Жарких, А.А. ПМ Конвейерный захват латентных изображений с IP-камеры / А.А. Жарких, Г.В. Шагрова // Свидетельство о регистрации электронного ресурса №22544, Зарегистрировано в Фонде алгоритмов и программ 2 февраля 2017 г.

87. Шагрова, Г.В. ПМ Построение комбинированных фильтров для выявления латентных изображений / Г.В. Шагрова, А.А. Жарких // Свидетельство

о регистрации электронного ресурса №22545, Зарегистрировано в Фонде алгоритмов и программ 2 февраля 2017 г.

88. Шагрова, Г.В. ПМ Построение и хеширование контрольного скрытого объекта в латентном изображении / Г.В. Шагрова, А.А. Жарких, И.Н. Топчиев // Свидетельство о регистрации электронного ресурса №22546, Зарегистрировано в Фонде алгоритмов и программ 2 февраля 2017 г.

89. Жарких, А.А. Программный модуль для контроля латентного изображения / А.А. Жарких // Свидетельство о регистрации электронного ресурса №20206, Зарегистрировано в Фонде алгоритмов и программ 11 июля 2014 г.

90. Жарких, А.А. ПМ Автоматическое определение объекта по распознанному латентному изображению / Г.В. Шагрова, А.А. Жарких, И.Н. Топчиев // Свидетельство о регистрации электронного ресурса №22547, Зарегистрировано в Фонде алгоритмов и программ 2 февраля 2017 г.

91. Шилдт, Г. Полный справочник по C# / Г.Шилдт.; пер.с англ. – М: Издательский дом "Вильямс", 2006. – 752 с.

92. Шилдт, Г. C# 4.0: полное руководство / Г. Шилдт.; пер.с англ. – М.: ООО "И.Д. Вильямс", 2011. – 1056 с.

93. Уотсон, К. Visual C# 2010: полный / К. Уотсон, К. Нейгел, Я. Педерсен [и др.].; пер.с англ. – М.: Диалектика, 2010. – 960 с.

94. Троелсен, Э. Язык программирования C# 2005 и платформа .NET 2.0 /Э. Троелсен; пер.с англ. – М: Издательский дом "Вильямс", 2007. – 1168 с.

95. Понамарев, В. Программирование на C++/C# в Visual Studio .NET 2003 / В. Понамарев. – СПб: БХВ-Петербург, 2004. – 352 с.

96. Вирт, Н. Алгоритмы и структуры данных. Пер. с англ. / Н.Вирт.; пер.с англ. – СПб: Невский Диалект, 2008. – 352 с.

97. Жарков, В.А. Компьютерная графика, мультимедиа и игры на Visual C# 2005 / В.А. Жарков. – М.: Жарков-Пресс, 2005. – 812 с.

ПРИЛОЖЕНИЕ 1. ЛИСТИНГ ПРОГРАММНОГО МОДУЛЯ ФОРМИРОВАНИЯ ЛАТЕНТНОГО ИЗОБРАЖЕНИЯ

```
function varargout = CreateLatent(varargin)
% CREATELATENT MATLAB code for CreateLatent.fig
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
  'gui_Singleton', gui_Singleton, ...
  'gui_OpeningFcn', @CreateLatent_OpeningFcn, ...
  'gui_OutputFcn', @CreateLatent_OutputFcn, ...
  'gui_LayoutFcn', [1] , ...
  'gui_Callback', [1]);
if nargin && ischar(varargin{1})
 gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
 [1varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
 gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before CreateLatent is made visible.
function CreateLatent_OpeningFcn(hObject, eventdata, handles, varargin)
% Choose default command line output for CreateLatent
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = CreateLatent_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in btn_create.
function btn_create_Callback(hObject, eventdata, handles)
global Latent;
% создание окна с пустой полосой прогресса
h = waitbar(0, 'Идет формирование латентного изображения...');
I=imread(get(handles.edt_imageA,'string'));
G = imread(get(handles.edt_imageB,'string'));
waitbar(4/100);
R=PsevdoR(I);
waitbar(40/100);
R2=PsevdoR2(I);
waitbar(60/100);
GCopy=((255-G)-254).*R2;
```

```
waitbar(65/100);
GInvert=255-GCopy;
waitbar(72/100);
GCopy_R = imlincomb (0.5, GCopy, 0.5, R);
waitbar(80/100);
SCR_SN = imlincomb (0.5, GCopy_R, 0.5, GInvert);
waitbar(90/100);
SCR_SN_Lightness = imadjust (SCR_SN, [10.4 0.5], [10.4 0.5]);
waitbar(95/100);
Latent = imlincomb (0.25, SCR_SN_Lightness, 1, I);
waitbar(98/100);
close(h);
cla;
image('CData', Latent);


% hObject handle to btn_create (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% --- Executes on button press in btn_save.
function btn_save_Callback(hObject, eventdata, handles)
% hObject handle to btn_save (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
global Latent;
[1FileName, PathName] = uiputfile('*.jpeg');
% Проверка, был ли выбран файл
if FileName~=0
 % Формирование полного пути к файлу
 FullName = [1PathName FileName];
 % Считывание изображения из графического файла
 imwrite(Latent,FullName,'quality',100);
end

function edt_imageA_Callback(hObject, eventdata, handles)

% hObject handle to edt_imageA (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edt_imageA as text
% str2double(get(hObject,'String')) returns contents of edt_imageA as a double

% --- Executes during object creation, after setting all properties.
function edt_imageA_CreateFcn(hObject, eventdata, handles)
% hObject handle to edt_imageA (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
 set(hObject,'BackgroundColor','white');
end

function edt_imageB_Callback(hObject, eventdata, handles)
% hObject handle to edt_imageB (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edt_imageB as text
% str2double(get(hObject,'String')) returns contents of edt_imageB as a double


% --- Executes during object creation, after setting all properties.
function edt_imageB_CreateFcn(hObject, eventdata, handles)
% hObject handle to edt_imageB (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
 set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in btn_optionA.
function btn_optionA_Callback(hObject, eventdata, handles)
% hObject handle to btn_optionA (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
[1FileName, PathName] = uigetfile('*.bmp;*.jpg;*.tiff;*.jpeg;*.png');
% Проверка, был ли выбран файл
if FileName~=0
 % Формирование полного пути к файлу
 FullName = [1PathName FileName];
 % Считывание изображения из графического файла
 set(handles.edt_imageA,'string',FullName);
 % Вывод изображения на оси
 cla;
end

% --- Executes on button press in btn_optionB.
function btn_optionB_Callback(hObject, eventdata, handles)
% hObject handle to btn_optionB (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
[1FileName, PathName] = uigetfile('*.bmp;*.jpg;*.tiff;*.jpeg;*.png');
% Проверка, был ли выбран файл
if FileName~=0
 % Формирование полного пути к файлу
 FullName = [1PathName FileName];
 % Считывание изображения из графического файла
 set(handles.edt_imageB,'string',FullName);
 % Вывод изображения на оси
 cla;
end

function z = PsevdoR(Orig)
% Функция растрирования PsevdoR
 for i=1:2:(ceil(size(Orig,1)/2))*2 %Y
 for j=1:2:size(Orig,2) %X
 for k=1:1:3
 z(i,j,k)=(Orig(i,j,k)+Orig(i+1,j,k))/2;
 z(i+1,j+1,k)=(Orig(i,j+1,k)+Orig(i+1,j+1,k))/2;
 end;
 end;
 end;
end
```

```
function z = PsevdoR2(Orig)
%Функция растрирования PsevdoR2
 for i=1:2:(ceil(size(Orig,1)/2))*2 %Y
 for j=1:2:size(Orig,2) %X
 for k=1:1:3
 z(i,j+1,k)=(Orig(i,j,k)+Orig(i,j+1,k))/2;
 z(i+1,j,k)=(Orig(i+1,j,k)+Orig(i+1,j+1,k))/2;
 end;
 end;
 end;
end
```

ПРИЛОЖЕНИЕ 2. ЛИСТИНГ ОСНОВНЫХ ФОРМ ПРОГРАММНОГО КОМПЛЕСА КОНТРОЛЯ ЛАТЕНТНОГО ИЗОБРАЖЕНИЯ

Главная форма системы **frmMain**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Imaging;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

using System.IO;
using System.Configuration;
using System.Net.Sockets;
using System.Threading;
using System.Net;

using AForge;
using AForge.Imaging;
using AForge.Imaging.Filters;
using Microsoft.Drawing;

namespace LatentDetect
{
 public partial class frmMain : Form
 {
 bool IsStart = false; // Определяет запущена ли работа модуля
 string ipCam = ConfigurationManager.AppSettings[1"IPCam"]; //IP камеры из файла конфигурации
 int FPS = (Int32.Parse(ConfigurationManager.AppSettings[1"FPS"])); // Определяет частоту
получения кадров
 int ResWidth; // Определяет разрешение получаемого кадра (ширина)
 int ResHeight; // Определяет разрешение получаемого кадра (высота)
 int ActivationStatus = 0; //Статус активации устройства захвата 0-не активно; 1-ожидается
захват; 2-проверка пригодности; 3-объект за границами
 int SecForGetCLI = (Int32.Parse(ConfigurationManager.AppSettings[1"ScanPeriod"]));
 int ProgramMode;//Режим работы модуля захвата 0-захват возможен с дисковых, сетевых хранилищ,
камеры и с каталога с тестовым видео;
 //1-захват возможен только с дисковых, сетевых хранилищ, камеры; 2-захват возможен только с
дисковых, сетевых хранилищ, каталога с тестовым видео;
 //3-захват только с с дисковых, сетевых хранилищ.
 String PathLog = ConfigurationManager.AppSettings[1"LogPath"] + "\\LatentDetect.log";

 Bitmap PrevCadr, CamCadr; // предыдущий и текущий кадры камеры.

 //!!!!!Вспомогательные переменные для имитации камеры
 int Frame,NumFrames;
 bool Direction;

 HttpWebRequest req;
 WebResponse resp;
 Stream stream;
 byte[1] buffer = new byte[11000000];

 public frmMain()
 {
```

```
String LogText = DateTime.Now.ToString().Replace(':', '-') + " Запуск программы\r\n";
System.IO.File.AppendAllText(PathLog, LogText);

frmInitial fi = new frmInitial();
fi.ShowDialog();
InitializeComponent();
}
private void frmMain_Load(object sender, EventArgs e)//Действия при загрузке программы
{
Frame = 0;
NumFrames = NumTestFrames();
Direction = true;
tmrFPS.Interval = 1000 / FPS;
}

private void frmMain_Activated(object sender, EventArgs e)
{
ProgramMode = (Int32.Parse(ConfigurationManager.AppSettings[1"ProgramMode"]));
if (ProgramMode == 2)
{
chkDebug.Checked = true;
}
else
{
chkDebug.Checked = false;
}
if (ProgramMode == 0)
{
chkDebug.Enabled = true;
}
else
{
chkDebug.Enabled = false;
}
if (ProgramMode != 3)
{
btnStartStop.Enabled = true;
}
else
{
btnStartStop.Enabled = false;
}


var Resolution = ConfigurationManager.AppSettings[1"Resolution"];
ResWidth = Convert.ToInt32(Resolution.Substring(0, Resolution.LastIndexOf('*')));
ResHeight = Convert.ToInt32(Resolution.Substring(Resolution.IndexOf('*') + 1));
ipCam = ConfigurationManager.AppSettings[1"IPCam"];
// SecForGetCLI = (Int32.Parse(ConfigurationManager.AppSettings[1"ScanPeriod"]));
PathLog = ConfigurationManager.AppSettings[1"LogPath"] + "\\LatentDetect.log";
}

private void btnConfig_Click(object sender, EventArgs e)//Открываем окно настроек
{
StopMonitor();
frmConfig f = new frmConfig();
f.ShowDialog();
}

private void SearchCLI_Click(object sender, EventArgs e)//!!!!!Тестовая кнопка получения
следующего цифрового ЛИ
{
var NextCLI = GetCLI();
if (NextCLI != "ПапкаПуста")
```

```
{
StopMonitor();
tabImage.SelectedTab = tpgPrepImage;
Stream file = File.Open(NextCLI, FileMode.Open);
CamCadr = (Bitmap)Bitmap.FromStream(file);
file.Close();
file.Dispose();
pctPrepImage.Image = CamCadr;
var PathSave = ConfigurationManager.AppSettings[1"ResultsPath"] + "\\Input\\" +
DateTime.Now.ToString().Replace(':','-') + Path.GetExtension(NextCLI);
File.Move(NextCLI, PathSave);
BeginDetect(false);
String LogText = DateTime.Now.ToString().Replace(':', '-') + " Изображение с хранилища
передано на обработку\r\n";
System.IO.File.AppendAllText(PathLog, LogText);
}
else
{
MessageBox.Show("Папка с цифровыми ЛИ пуста");
}
}

private void btnStartStop_Click(object sender, EventArgs e)//Запуск монитора входящих
изображений
{
if (IsStart) // Действия при остановке монитора
{
StopMonitor();
}
else // Действия при запуске монитора
{
StartMonitor();
}
}

private void tmrFPS_Tick(object sender, EventArgs e)//Хронометр камеры
{
if (chkDebug.Checked) //!!!!!Если в режиме отладки без камеры
{
CamCadr = getTestImage();
}
else
{
CamCadr = getCamImage(ResWidth, ResHeight, 0);
}
pctCamImage.Image = CamCadr;
if (ActivationStatus <= 1 || ActivationStatus == 3 || ActivationStatus == 5 ||
ActivationStatus == 6)
{
Activation();
}
else if (ActivationStatus == 2)
{
CheckBorders();
}
else if (ActivationStatus == 4)
{
Activation();
ActivationStatus = 5;
lblBLOB.Text = "\nТекущий объект\nнеобходимо убрать\nдля получения следующего\nобъекта\n";
lblBLOB.BackColor = Color.Red;
}
}
```

```
private void tmrMonitor_Tick(object sender, EventArgs e)//Хронометр монитора цифровых ЛИ
{
if (SecForGetCLI > 1)
{
SecForGetCLI--;
lblTimerCLI.Text = "Цифровое ЛИ будет извлечено\nчерез: " + SecForGetCLI + " сек.";
}
else if (ActivationStatus == 0 || ActivationStatus == 5)
{
var NextCLI = GetCLI();
if (NextCLI != "ПапкаПуста")
{
 StopMonitor();
 tabImage.SelectedTab = tpgPrepImage;
 Stream file = File.Open(NextCLI, FileMode.Open);
 CamCadr = (Bitmap)Bitmap.FromStream(file);
 file.Close();
 file.Dispose();

 pctPrepImage.Image = CamCadr;
 var PathSave = ConfigurationManager.AppSettings[1"ResultsPath"] + "\\Input\\" +
DateTime.Now.ToString().Replace(':', '-') + Path.GetExtension(NextCLI);
 File.Move(NextCLI, PathSave);
 BeginDetect(false);
 String LogText = DateTime.Now.ToString().Replace(':', '-') + " Изображение с хранилища
передано на обработку\r\n";
 System.IO.File.AppendAllText(PathLog, LogText);

}
SecForGetCLI = (Int32.Parse(ConfigurationManager.AppSettings[1"ScanPeriod"]));
lblTimerCLI.Text = "Цифровое ЛИ будет извлечено\nчерез: " + SecForGetCLI + " сек.";
}
}

# region Анализ изображения с камеры
public void Activation()//Функция определения необходимости активации устройства захвата
{
ThresholdedDifference TDfilter = new ThresholdedDifference(120);
TDfilter.OverlayImage = (Bitmap)PrevCadr;
ConnectedComponentsLabeling CCLfilter = new ConnectedComponentsLabeling();
Erosion Efilter = new Erosion();
var temp = CCLfilter.Apply(Efilter.Apply(TDfilter.Apply(CamCadr)));
PrevCadr = CamCadr;
if (CCLfilter.ObjectCount > 0 && (ActivationStatus==0||ActivationStatus==3))
{
ActivationStatus = 1;
lblBLOB.Text = "\n\n\nОжидается захват\n\n\n\n";
lblBLOB.BackColor = Color.Yellow;
}
else if (CCLfilter.ObjectCount == 0 && ActivationStatus == 1)
{
ActivationStatus = 2;
lblBLOB.Text = "\n\n\nПроверка пригодности\n\n\n\n";
lblBLOB.BackColor = Color.Blue;
}
else if (CCLfilter.ObjectCount > 0 && ActivationStatus == 5)
{
ActivationStatus = 6;
lblBLOB.Text = "\n\n\nУбираем объект\n\n\n\n";
lblBLOB.BackColor = Color.Orange;
}
else if (CCLfilter.ObjectCount == 0 && ActivationStatus == 6)
{
ActivationStatus = 0;
```

```
lblBLOB.Text = "\n\n\nОбъект убран. Не активно\n\n\n\n";
lblBLOB.BackColor = Color.LightGray;
}
}

public void CheckBorders()//Функция проверки полного вхождения в кадр
{
Rectangle rect = GetObjectCoordinates(CamCadr);
if (rect.X < 5 || rect.Y < 5 || (CamCadr.Width - rect.Width) < 5 || (CamCadr.Height -
rect.Height) < 5)
{
ActivationStatus = 3;
lblBLOB.Text = "\n\n\nОБЪЕКТ ЗА КАДРОМ\n\n\n\n";
lblBLOB.BackColor = Color.Red;
}
else
{
StopMonitor();
CropNSave(rect);
}
}

public Rectangle GetObjectCoordinates(Bitmap Image)//Функция определения границ входящего
изображения
{
FiltersSequence Morph = new FiltersSequence();
Morph.Add(new HistogramEqualization());
Morph.Add(Grayscale.CommonAlgorithms.BT709);
Morph.Add(new BayerDithering());
Morph.Add(new Erosion3x3());
Morph.Add(new HomogenityEdgeDetector());
Image = Morph.Apply(Image);
BlobsFiltering BFfilter = new BlobsFiltering(50, 50, 1000, 1000, true);
BFfilter.ApplyInPlace(Image);
UnmanagedImage UMTemp = UnmanagedImage.FromManagedImage(Image);
int minX = UMTemp.Width;
int minY = UMTemp.Height;
int maxX = 0;
int maxY = 0;
int offset = UMTemp.Stride - UMTemp.Width;
// find rectangle which contains something except color to remove
unsafe
{
byte* src = (byte*)UMTemp.ImageData.ToPointer();
for (int y = 0; y < UMTemp.Height; y++)
{
 for (int x = 0; x < UMTemp.Width; x++, src++)
 {
 if (*src != 0)
 {
 if (x < minX)
 minX = x;
 if (x > maxX)
 maxX = x;
 if (y < minY)
 minY = y;
 if (y > maxY)
 maxY = y;
 }
 }
 src += offset;
}
}
return new Rectangle(minX, minY, maxX - minX, maxY - minY);
```

```
 }
 # endregion
 public void CropNSave(Rectangle rect)//Функция кадрирования
 {
 Crop filter = new Crop(rect);
 ActivationStatus = 4;
 lblBLOB.Text = "\n\nОбъект готов к\ndальнейшей обработке\n\n\n";
 lblBLOB.BackColor = Color.Green;
 tabImage.SelectedTab = tpgPrepImage;
 CamCadr=filter.Apply(CamCadr);
 pctPrepImage.Image = CamCadr;
 var PathSave = ConfigurationManager.AppSettings[1"ResultsPath"] + "\\Input\\" +
DateTime.Now.ToString().Replace(':', '-') + ".bmp";
 pctPrepImage.Image.Save(PathSave, System.Drawing.Imaging.ImageFormat.Bmp);
 BeginDetect(true);
 String LogText = DateTime.Now.ToString().Replace(':', '-') + " Изображение с камеры передано
на обработку\r\n";
 System.IO.File.AppendAllText(PathLog, LogText);
 }

 public Bitmap getCamImage(int resx, int resy, int comp)//Функция получения изображения с
камеры
 {
 int read, total = 0;
 string sourceURL = "http://" + ipCam + "/axis-cgi/jpg/image.cgi?resolution=" +
resx.ToString() + "x" + resy.ToString() + "&compression=" + comp.ToString();
 // create HTTP request
 req = (HttpWebRequest)WebRequest.Create(sourceURL);
 // get response
 resp = req.GetResponse();
 // get response stream
 stream = resp.GetResponseStream();
 // read data from stream
 while ((read = stream.Read(buffer, total, 1000)) != 0)
 {
 total += read;
 }
 // get bitmap
 return (Bitmap)Bitmap.FromStream(new MemoryStream(buffer, 0, total));
 }

 # region Обеспечение имитации камеры
 public static string NextTestImage(int NumberCadr)//!!!!!Функция имитации камеры
 {
 var dir = ConfigurationManager.AppSettings[1"TestVideoPath"];
 var files = Directory.EnumerateFiles(@dir, "*.*").
 Where(s => s.EndsWith(".bmp") || s.EndsWith(".jpg") || s.EndsWith(".jpe") ||
  s.EndsWith(".jpeg") || s.EndsWith(".rle") || s.EndsWith(".png") ||
  s.EndsWith(".tif") || s.EndsWith(".tiff")).ToArray();
 return files[1NumberCadr];
 }
 public Bitmap getTestImage()//Функция получения изображения имитирующего работу камеры
 {
 if (Direction == true & Frame < NumFrames-1) { Frame += 1; }
 else if (Direction == true & Frame == NumFrames - 1) { Direction = false; Frame -= 1; }
 else if (Direction == false & Frame > 0) { Frame -= 1; }
 else if (Direction == false & Frame == 0) { Direction = true; Frame += 1; }
 // get bitmap
 return (Bitmap)Bitmap.FromFile(NextTestImage(Frame));
 }
 public static int NumTestFrames()//!!!!!Функция имитации камеры
 {
 var dir = ConfigurationManager.AppSettings[1"TestVideoPath"];
 var files = Directory.EnumerateFiles(@dir, "*.*").
```

```
Where(s => s.EndsWith(".bmp") || s.EndsWith(".jpg") || s.EndsWith(".jpe") ||
 s.EndsWith(".jpeg") || s.EndsWith(".rle") || s.EndsWith(".png") ||
 s.EndsWith(".tif") || s.EndsWith(".tiff")).ToArray();
return files.Length;
}
# endregion

public static string GetCLI()//Функция получения адреса следующего цифрового ЛИ
{
var dir = ConfigurationManager.AppSettings[1"DigitalLIPath"];
var files = Directory.EnumerateFiles(@dir, "*.*").
Where(s => s.EndsWith(".bmp") || s.EndsWith(".jpg") || s.EndsWith(".jpe") ||
 s.EndsWith(".jpeg") || s.EndsWith(".rle") || s.EndsWith(".png") ||
 s.EndsWith(".tif") || s.EndsWith(".tiff")).ToArray();
if (files.Length > 0)
{
return files[10];
}
else
{
return "ПапкаПуста";
}
}

public void StartMonitor() // Действия при запуске монитора
{
if (!IsStart)
{
String LogText = DateTime.Now.ToString().Replace(':', '-') + " Запущен мониторинг\r\n";
System.IO.File.AppendAllText(PathLog, LogText);
}
btnStartStop.Image = LatentDetect.Properties.Resources.pause_48x48;
IsStart = true;
tabImage.SelectedTab = tpgCamIm;
if (chkDebug.Checked) //!!!!!Если в режиме отладки без камеры
{
CamCadr = getTestImage();
}
else
{
CamCadr = getCamImage(ResWidth, ResHeight, 0);
}
PrevCadr = CamCadr;
tmrFPS.Start();
tmrMonitor.Start();
}

public void StopMonitor() // Действия при остановке монитора
{
if (IsStart)
{
String LogText = DateTime.Now.ToString().Replace(':', '-') + " Мониторинг остановлен\r\n";
System.IO.File.AppendAllText(PathLog, LogText);
}
btnStartStop.Image = LatentDetect.Properties.Resources.play_48x48;
IsStart = false;
tmrFPS.Stop();
tmrMonitor.Stop();
}

public void BeginDetect(bool IsScaned) // Запуск распознавания изображения
{
frmDetect f = new frmDetect(CamCadr, IsScaned);
f.ShowDialog();
```

```
 StartMonitor();
 }

 private void frmMain_FormClosed(object sender, FormClosedEventArgs e)
 {
 String LogText = DateTime.Now.ToString().Replace(':', '-') + " Завершена работа
программы\r\n";
 System.IO.File.AppendAllText(PathLog, LogText);
 }

 }
}
```

Форма проверки подлинности документа и принятия решения **frmDetect**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

using System.Configuration;
using System.IO;

using AForge;
using AForge.Imaging;
using AForge.Imaging.Filters;
using Microsoft.Drawing;

namespace LatentDetect
{
 public partial class frmDetect : Form
 {
 Bitmap[1] Image = new Bitmap[1100];
 Bitmap Hash;
 IniFile FiltersFile, ControlFile;
 List<string> listControl = new List<string>();
 List<string> listDetect = new List<string>();
 String MessageDetect, PathSave;
 int NumDetect;
 String PathLog = ConfigurationManager.AppSettings[1"LogPath"] + "\\LatentDetect.log";

 double AuthKoef;
 string TypeLI;
 bool IsAuto;

 public frmDetect(Bitmap CamCadr, bool IsScaned)
 {
 InitializeComponent();
 PathSave = ConfigurationManager.AppSettings[1"ResultsPath"] + "\\Detect\\" +
DateTime.Now.ToString().Replace(':', '-') + "\\";
 Directory.CreateDirectory(PathSave);
 NumDetect = 0;
 Image[10] = CamCadr;
 FiltersFile = new IniFile(ConfigurationManager.AppSettings[1"CombinedFiltersBase"] +
"\\Filters.ini");
 ControlFile = new IniFile(ConfigurationManager.AppSettings[1"ControlImagesBase"] +
"\\Control.ini");
 if (IsScaned)
```

```
{
AuthKoef = Convert.ToDouble(ConfigurationManager.AppSettings[1"AuthenticityScanedLI"])/100;
TypeLI=ConfigurationManager.AppSettings[1"TypeScanedLI"];
}
else
{
AuthKoef = Convert.ToDouble(ConfigurationManager.AppSettings[1"AuthenticityDigitalLI"])/100;
TypeLI = ConfigurationManager.AppSettings[1"TypeDigitalLI"];
}

var FullListControl = ControlFile.IniReadKey("Control");
if (TypeLI != "*ALL")
{
foreach (String HashType in FullListControl)
{
 if (HashType.IndexOf(TypeLI) > -1)
 {
 listControl.Add(HashType);
 }
}
}
else
{
listControl = FullListControl;
}

if (ConfigurationManager.AppSettings[1"ModeDecision"] == "0")
{
IsAuto = false;
}
else
{
IsAuto = true;
}
}
private void frmDetect_Load(object sender, EventArgs e)
{
cmbImage.Items.Clear();
cmbImage.Items.Add("Исходное");
cmbImage.SelectedIndex = 0;
pctImage.Image = Image[10];
}

private void frmDetect_Shown(object sender, EventArgs e)
{
DetectImage DI;
RotateBicubic RotateF = new RotateBicubic(330, true);
RotateBicubic ReRotateF = new RotateBicubic(30, true);
lblStatus.Text = "Идет вычисление горизонтальных коэффициентов";
this.Refresh();
Image[11] = wavelet(Image[10], 'h');
cmbImage.Items.Add("Вейвлет_гориз");
Image[11].Save(PathSave + "horiz.bmp", System.Drawing.Imaging.ImageFormat.Bmp);
Image[12] = qualityUp(Image[11]);
Image[12].Save(PathSave + "horiz_up.bmp", System.Drawing.Imaging.ImageFormat.Bmp);
DI = CheckHashBlobsOnImg(Image[12]);
Image[12] = DI.Image;
cmbImage.Items.Add("Вейвлет_гориз_улучш");
pctImage.Image = Image[12];
cmbImage.SelectedIndex = 2;
if (!DI.IsDetected)
{
this.Refresh();
Image[13] = qualityUp2(Image[11]);
```

```
Image[13].Save(PathSave + "horiz_up2.bmp", System.Drawing.Imaging.ImageFormat.Bmp);
DI = CheckHashBlobsOnImg(Image[13]);
Image[13] = DI.Image;
cmbImage.Items.Add("Вейвлет_гориз_улучш2");
pctImage.Image = Image[13];
cmbImage.SelectedIndex = 3;
}
if (!DI.IsDetected)
{
this.Refresh();
Image[14] = RotateF.Apply(Image[10]);
Image[14] = wavelet(Image[14], 'h');
cmbImage.Items.Add("Вейвлет_гориз_зерк");
Image[14].Save(PathSave + "horizM.bmp", System.Drawing.Imaging.ImageFormat.Bmp);
Image[15] = qualityUp(Image[14]);
Image[15] = ReRotateF.Apply(Image[15]);
Image[15].Save(PathSave + "horizM_up.bmp", System.Drawing.Imaging.ImageFormat.Bmp);
DI = CheckHashBlobsOnImg(Image[15]);
Image[15] = DI.Image;
cmbImage.Items.Add("Вейвлет_гориз_з_улучш");
pctImage.Image = Image[15];
cmbImage.SelectedIndex = 5;
}
if (!DI.IsDetected)
{
this.Refresh();
Image[16] = qualityUp2(Image[14]);
Image[16] = ReRotateF.Apply(Image[16]);
Image[16].Save(PathSave + "horizM_up2.bmp", System.Drawing.Imaging.ImageFormat.Bmp);
DI = CheckHashBlobsOnImg(Image[16]);
Image[16] = DI.Image;
cmbImage.Items.Add("Вейвлет_гориз_з_улучш2");
pctImage.Image = Image[16];
cmbImage.SelectedIndex = 6;
}

if (!DI.IsDetected)
{
lblStatus.Text = "Идет вычисление вертикальных коэффициентов";
this.Refresh();
Image[17] = wavelet(Image[10], 'v');
cmbImage.Items.Add("Вейвлет_верт");
Image[17].Save(PathSave + "vert.bmp", System.Drawing.Imaging.ImageFormat.Bmp);
Image[18] = qualityUp(Image[17]);
Image[18].Save(PathSave + "vert_up.bmp", System.Drawing.Imaging.ImageFormat.Bmp);
DI = CheckHashBlobsOnImg(Image[18]);
Image[18] = DI.Image;
cmbImage.Items.Add("Вейвлет_верт_улучш");
pctImage.Image = Image[18];
cmbImage.SelectedIndex = 8;
}
if (!DI.IsDetected)
{
this.Refresh();
Image[19] = qualityUp2(Image[17]);
Image[19].Save(PathSave + "vert_up2.bmp", System.Drawing.Imaging.ImageFormat.Bmp);
DI = CheckHashBlobsOnImg(Image[19]);
Image[19] = DI.Image;
cmbImage.Items.Add("Вейвлет_верт_улучш2");
pctImage.Image = Image[19];
cmbImage.SelectedIndex = 9;
}
if (!DI.IsDetected)
{
```

```
this.Refresh();
Image[110] = RotateF.Apply(Image[10]);
Image[110] = wavelet(Image[110], 'v');
cmbImage.Items.Add("Вейвлет_верт_зерк");
Image[110].Save(PathSave + "vertM.bmp", System.Drawing.Imaging.ImageFormat.Bmp);
Image[111] = qualityUp(Image[110]);
Image[111] = ReRotateF.Apply(Image[111]);
Image[111].Save(PathSave + "vertM_up.bmp", System.Drawing.Imaging.ImageFormat.Bmp);
DI = CheckHashBlobsOnImg(Image[111]);
Image[111] = DI.Image;
cmbImage.Items.Add("Вейвлет_верт_з_улучш");
pctImage.Image = Image[111];
cmbImage.SelectedIndex = 11;
}
if (!DI.IsDetected)
{
this.Refresh();
Image[112] = qualityUp2(Image[110]);
Image[112] = ReRotateF.Apply(Image[112]);
Image[112].Save(PathSave + "vertM_up2.bmp", System.Drawing.Imaging.ImageFormat.Bmp);
DI = CheckHashBlobsOnImg(Image[112]);
Image[112] = DI.Image;
cmbImage.Items.Add("Вейвлет_верт_з_улучш2");
pctImage.Image = Image[112];
cmbImage.SelectedIndex = 12;
}

if (!DI.IsDetected)
{
lblStatus.Text = "Идет вычисление диагональных коэффициентов";
this.Refresh();
Image[113] = wavelet(Image[10], 'd');
cmbImage.Items.Add("Вейвлет_диаг");
Image[113].Save(PathSave + "diag.bmp", System.Drawing.Imaging.ImageFormat.Bmp);
Image[114] = qualityUp(Image[113]);
Image[114].Save(PathSave + "diag_up.bmp", System.Drawing.Imaging.ImageFormat.Bmp);
DI = CheckHashBlobsOnImg(Image[114]);
Image[114] = DI.Image;
cmbImage.Items.Add("Вейвлет_диаг_улучш");
pctImage.Image = Image[114];
cmbImage.SelectedIndex = 14;
}
if (!DI.IsDetected)
{
this.Refresh();
Image[115] = qualityUp2(Image[113]);
Image[115].Save(PathSave + "diag_up2.bmp", System.Drawing.Imaging.ImageFormat.Bmp);
DI = CheckHashBlobsOnImg(Image[115]);
Image[115] = DI.Image;
cmbImage.Items.Add("Вейвлет_диаг_улучш2");
pctImage.Image = Image[115];
cmbImage.SelectedIndex = 15;
}
if (!DI.IsDetected)
{
this.Refresh();
Image[116] = RotateF.Apply(Image[10]);
Image[116] = wavelet(Image[116], 'd');
cmbImage.Items.Add("Вейвлет_диаг_зерк");
Image[116].Save(PathSave + "diagM.bmp", System.Drawing.Imaging.ImageFormat.Bmp);
Image[117] = qualityUp(Image[116]);
Image[117] = ReRotateF.Apply(Image[117]);
Image[117].Save(PathSave + "diagM_up.bmp", System.Drawing.Imaging.ImageFormat.Bmp);
DI = CheckHashBlobsOnImg(Image[117]);
```

```
 Image[117] = DI.Image;
 cmbImage.Items.Add("Вейвлет_диаг_з_улучш");
 pctImage.Image = Image[117];
 cmbImage.SelectedIndex = 17;
 }
 if (!DI.IsDetected)
 {
 this.Refresh();
 Image[118] = qualityUp2(Image[116]);
 Image[118] = ReRotateF.Apply(Image[118]);
 Image[118].Save(PathSave + "diagM_up2.bmp", System.Drawing.Imaging.ImageFormat.Bmp);
 DI = CheckHashBlobsOnImg(Image[118]);
 Image[118] = DI.Image;
 cmbImage.Items.Add("Вейвлет_диаг_з_улучш2");
 pctImage.Image = Image[118];
 cmbImage.SelectedIndex = 18;
 }
 if (!DI.IsDetected)
 {
 this.Refresh();
 var FullListCombined = FiltersFile.IniReadKey("Combined");
 if (TypeLI == "*ALL")
 {
  int count=19;
  foreach (String CF in FullListCombined)
  {
  if (!DI.IsDetected)
  {
  lblStatus.Text = "Идет наложение комбинированного фильтра" + CF;
  this.Refresh();
  Image[1count] = Combine(Image[10], CF);
  cmbImage.Items.Add("Комб_" + CF);
  Image[1count].Save(PathSave + "comb" + CF + ".bmp", System.Drawing.Imaging.ImageFormat.Bmp);
  count += 1;
  Image[1count] = qualityUp(Image[1count-1]);
  Image[1count].Save(PathSave + "comb" + CF + "up.bmp",
System.Drawing.Imaging.ImageFormat.Bmp);
  DI = CheckHashBlobsOnImg(Image[1count]);
  Image[1count] = DI.Image;
  cmbImage.Items.Add("Комб_улучш_" + CF);
  pctImage.Image = Image[1count];
  cmbImage.SelectedIndex = count;
  count += 1;
  }
  if (!DI.IsDetected)
  {
  this.Refresh();
  Image[1count] = qualityUp2(Image[1count-2]);
  Image[1count].Save(PathSave + "comb" + CF + "up2.bmp",
System.Drawing.Imaging.ImageFormat.Bmp);
  DI = CheckHashBlobsOnImg(Image[1count]);
  Image[1count] = DI.Image;
  cmbImage.Items.Add("Комб_улучш2_" + CF);
  pctImage.Image = Image[1count];
  cmbImage.SelectedIndex = count;
  count += 1;
  }
  if (count > 97 || DI.IsDetected) break;
  }
 }
 else
 {
  if (FullListCombined.IndexOf(TypeLI) > -1)
  {
```

```
  if (!DI.IsDetected)
  {
  lblStatus.Text = "Идет наложение комбинированного фильтра" + TypeLI;
  this.Refresh();
  Image[119] = Combine(Image[10], TypeLI);
  cmbImage.Items.Add("Комб_" + TypeLI);
  Image[119].Save(PathSave + "comb" + TypeLI + ".bmp",
System.Drawing.Imaging.ImageFormat.Bmp);
  Image[120] = qualityUp(Image[119]);
  Image[120].Save(PathSave + "comb" + TypeLI + "up.bmp",
System.Drawing.Imaging.ImageFormat.Bmp);
  DI = CheckHashBlobsOnImg(Image[120]);
  Image[120] = DI.Image;
  cmbImage.Items.Add("Комб_улучш_" + TypeLI);
  pctImage.Image = Image[120];
  cmbImage.SelectedIndex = 20;
  }
  if (!DI.IsDetected)
  {
  this.Refresh();
  Image[121] = qualityUp2(Image[119]);
  Image[121].Save(PathSave + "comb" + TypeLI + "up2.bmp",
System.Drawing.Imaging.ImageFormat.Bmp);
  DI = CheckHashBlobsOnImg(Image[121]);
  Image[121] = DI.Image;
  cmbImage.Items.Add("Комб_улучш2_" + TypeLI);
  pctImage.Image = Image[121];
  cmbImage.SelectedIndex = 21;
  }
  }
 }
 }

 this.Refresh();
 lblStatus.Text = "Выявление завершено";
 foreach (String LD in listDetect)
 {
 MessageDetect += " " + LD;
 }

 if (MessageDetect!=null)
 {
 String LogText = DateTime.Now.ToString().Replace(':', '-') + " Автоматически распознано
скрытое изображение типа "+ MessageDetect+"\r\n";
 System.IO.File.AppendAllText(PathLog, LogText);
 }
 else
 {
 String LogText = DateTime.Now.ToString().Replace(':', '-') + " Анализ ЛИ завершен.
Автоматически распознать скрытое изображение не удалось\r\n";
 System.IO.File.AppendAllText(PathLog, LogText);
 }
 if (IsAuto)
 {
 this.Close();
 }
 else
 {
 if (MessageDetect != null)
 {
  MessageBox.Show("На исследуемом изображении обнаружены скрытые изображения относящиеся к
следующим типам:" + MessageDetect);
 }
 else
```

```
{
 MessageBox.Show("Скрытых изображений не обнаружено");
}
btnAllowed.Visible = true;
btnNotAlowed.Visible = true;
}
}

private void cmbImage_SelectedIndexChanged(object sender, EventArgs e)
{
pctImage.Image = Image[1cmbImage.SelectedIndex];
}

private void btnAllowed_Click(object sender, EventArgs e)
{
this.Close();
String LogText = DateTime.Now.ToString().Replace(':', '-') + " Оператор определил что
изображение подлинное\r\n";
System.IO.File.AppendAllText(PathLog, LogText);
}

private void btnNotAlowed_Click(object sender, EventArgs e)
{
this.Close();
String LogText = DateTime.Now.ToString().Replace(':', '-') + " Оператор не смог определить
изображение как подлинное\r\n";
System.IO.File.AppendAllText(PathLog, LogText);
}

#region Вейвлет-анализ
//Функция получения вейвлет-коэффициентов
Bitmap wavelet(Bitmap OrigImage, char CoefType)
{
// CoefType: a - апроксимационные коэффициенты
// h - горизонтальные коэффициенты
// v - вертикальные коэффициенты
// d - диагональные коэффициенты

// Объявляем высокочастотный и низкочастотные фильтры для вейвлета bior2.8 для строк (X) и
столбцов (У)
double[1,] CLowX = {{0.0},{0.0015105430506304422},{-0.0030210861012608843},{-
0.012947511862546647},{0.02891610982635418},
{0.052998481890690945},{-0.13491307360773608},{-
0.16382918343409025},{0.4625714404759166},{0.9516421218971786},
{0.4625714404759166},{-0.16382918343409025},{-
0.13491307360773608},{0.052998481890690945},{0.02891610982635418},
{-0.012947511862546647},{-0.0030210861012608843},{0.0015105430506304422}};
double[1,] CHighX = {{0.0},{0.0},{0.0},{0.0},{0.0},{0.0},{0.0},{0.3535533905932738},{-
0.7071067811865476},{0.3535533905932738},{0.0},
{0.0},{0.0},{0.0},{0.0},{0.0},{0.0},{0.0}};
double[1,] CLowY = {{0.0,0.0015105430506304422,-0.0030210861012608843,-
0.012947511862546647,0.02891610982635418,
0.052998481890690945,-0.13491307360773608,-
0.16382918343409025,0.4625714404759166,0.9516421218971786,
0.4625714404759166,-0.16382918343409025,-
0.13491307360773608,0.052998481890690945,0.02891610982635418,
-0.012947511862546647,-0.0030210861012608843,0.0015105430506304422}};
double[1,] CHighY = {{0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.3535533905932738,-
0.7071067811865476,0.3535533905932738,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0}};
int KernelSize = CHighX.GetLength(0);
// Приводим к оттенкам серого
Grayscale filter = new Grayscale(0.2125, 0.7154, 0.0721);
Bitmap WaveRows = filter.Apply(OrigImage);
```

```
 // Расширяем изображение перед сверткой
 WaveRows = ExtentionImageMirrow(WaveRows, KernelSize);
 //Сворачиваем по строкам
 if (CoefType == 'a' || CoefType == 'h')
 WaveRows = conv2(WaveRows, CLowX);
 else if (CoefType == 'v' || CoefType == 'd')
 WaveRows = conv2(WaveRows, CHighX);
 //Прореживаем строки
 WaveRows = decImage(WaveRows, false);
 //Обрезаем лишние столбцы
 Crop CropFilterX = new Crop(new Rectangle(KernelSize / 2, 0, WaveRows.Width - KernelSize,
WaveRows.Height));
 Bitmap WaveCols = CropFilterX.Apply(WaveRows);

 //Сворачиваем по столбцам
 if (CoefType == 'a' || CoefType == 'v')
 WaveCols = conv2(WaveCols, CLowY);
 else if (CoefType == 'h' || CoefType == 'd')
 WaveCols = conv2(WaveCols, CHighY);
 //Прореживаем столбцы
 WaveCols = decImage(WaveCols, true);
 //Обрезаем лишние строки
 Crop CropFilterY = new Crop(new Rectangle(0, KernelSize / 2, WaveCols.Width, WaveCols.Height
- KernelSize));
 Bitmap OutImage = CropFilterY.Apply(WaveCols);
 return OutImage;
 }

 //Функция зеркального расширения изображения
 Bitmap ExtentionImageMirrow(Bitmap Image, int Size)
 {
 int UpLeftSize = (int)Size / 2;
 int DownRightSize = Size - UpLeftSize;
 Bitmap temp = new Bitmap(Image.Width + UpLeftSize, Image.Height + UpLeftSize);
 TextureBrush tBrush = new TextureBrush(Image);
 tBrush.WrapMode = System.Drawing.Drawing2D.WrapMode.TileFlipXY;
 Graphics tg = Graphics.FromImage(temp);
 tg.FillRectangle(tBrush, 0, 0, temp.Width, temp.Height);
 tg.Dispose();
 temp.RotateFlip(RotateFlipType.Rotate180FlipNone);
 Bitmap OutImage = new Bitmap(temp.Width + DownRightSize, temp.Height + DownRightSize);
 TextureBrush oBrush = new TextureBrush(temp);
 oBrush.WrapMode = System.Drawing.Drawing2D.WrapMode.TileFlipXY;
 Graphics g = Graphics.FromImage(OutImage);
 g.FillRectangle(oBrush, 0, 0, OutImage.Width, OutImage.Height);
 g.Dispose();
 OutImage.RotateFlip(RotateFlipType.Rotate180FlipNone);
 return OutImage;
 }

 // Функция свертки изображения
 Bitmap conv2(Bitmap Image, double[1,] kernel)
 {
 int rows = Image.Width;
 int cols = Image.Height;
 int kRows = kernel.GetLength(0);
 int kCols = kernel.GetLength(1);
 int kCenterX = kCols / 2;
 int kCenterY = kRows / 2;
 Bitmap outImage = new Bitmap(Image);
 var tempIn = new FastBitmap(Image);
 var tempOut = new FastBitmap(outImage);

 for (int i = 0; i < rows; ++i)  // Строки
```

```
{
for (int j = 0; j < cols; ++j)  // Колонки
{
 double NewColor = 0;
 for (int m = 0; m < kRows; ++m) // Строки матрицы-свертки
 {
 int mm = kRows - 1 - m; // переварачиваем строки матрицы-свертки

 for (int n = 0; n < kCols; ++n) // Колонки матрицы-свертки
 {
 int nn = kCols - 1 - n; // переварачиваем строки матрицы-свертки

 // Вычисляем индексы изображения для проверки не выхода за границы
 int ii = i + (m - kCenterY);
 int jj = j + (n - kCenterX);

 // если не выходит за границы - считаем
 if (ii >= 0 && ii < rows && jj >= 0 && jj < cols)
 NewColor += ((tempIn.GetColor(ii, jj).R) * kernel[1mm, nn]);
 }
 }
 // отсекаем значения <0 и >255
  if (NewColor < 0) NewColor = 0;
  if (NewColor > 255) NewColor = 255;
 //Присваиваем новое значение
 tempOut.SetColor(i, j, Color.FromArgb((int)(NewColor), (int)(NewColor), (int)(NewColor)));
}
}
tempIn.Dispose();
tempOut.Dispose();
return outImage;
}

//Функция прореживания изображения
Bitmap decImage(Bitmap Image, bool RowsCols)
{
// RowsCols: false - строки
// true - колонки
if (RowsCols == false)
{
Bitmap OutImage = new Bitmap(Image.Width, Image.Height / 2);
var tempIn = new FastBitmap(Image);
var tempOut = new FastBitmap(OutImage);
for (int i = 0; i < OutImage.Width; i++)
{
 for (int j = 0; j < OutImage.Height; j++)
 {
 int NewColor = tempIn.GetColor(i, j * 2).R;
 tempOut.SetColor(i, j, Color.FromArgb(NewColor, NewColor, NewColor));
 }
}
tempIn.Dispose();
tempOut.Dispose();
return OutImage;
}
else
{
Bitmap OutImage = new Bitmap(Image.Width / 2, Image.Height);
var tempIn = new FastBitmap(Image);
var tempOut = new FastBitmap(OutImage);
for (int i = 0; i < OutImage.Width; i++)
{
 for (int j = 0; j < OutImage.Height; j++)
 {
```

```
 int NewColor = tempIn.GetColor(i * 2, j).R;
 tempOut.SetColor(i, j, Color.FromArgb(NewColor, NewColor, NewColor));
 }
}
tempIn.Dispose();
tempOut.Dispose();
return OutImage;
}
}
#endregion

Bitmap qualityUp(Bitmap Image)
{
FiltersSequence Morph = new FiltersSequence();

Morph.Add(Grayscale.CommonAlgorithms.BT709);
Morph.Add(new JarvisJudiceNinkeDithering());
Morph.Add(new Dilatation());
Morph.Add(new OilPainting());
Morph.Add(new BlobsFiltering(15,15,1200,1200));
Morph.Add(new Closing());
Bitmap OutImage = Morph.Apply(Image);

return OutImage;
}
Bitmap qualityUp2(Bitmap Image)
{
FiltersSequence Morph = new FiltersSequence();

Morph.Add(Grayscale.CommonAlgorithms.BT709);
Morph.Add(new Dilatation());
Morph.Add(new OilPainting());
Morph.Add(new JarvisJudiceNinkeDithering());
Morph.Add(new Closing());
Morph.Add(new BlobsFiltering(15, 15, 1200, 1200));
Morph.Add(new Opening());
Bitmap OutImage = Morph.Apply(Image);

return OutImage;
}

#region Сверка с контрольными изображениями

public struct DetectImage
{
public Bitmap Image;
public bool IsDetected;
}

private DetectImage CheckHashBlobsOnImg(Bitmap Image)
{
DetectImage DI;
Bitmap OutIm = new Bitmap(Image);
DI.Image = OutIm;
DI.IsDetected = false;
String HashCheck,HashControl;

BlobCounter blobCounter = new BlobCounter(Image);
Rectangle[1] rects = blobCounter.GetObjectsRectangles();
foreach (Rectangle rc in rects)
{
FiltersSequence HashIm = new FiltersSequence(
new Crop(new Rectangle(rc.Left, rc.Top, rc.Width, rc.Height)),
new ResizeBicubic(8, 8)
```

```
 );
 Hash = HashIm.Apply(Image);
 HashCheck = GetHashString(Hash);

  foreach (String HC in listControl)
  {
  HashControl = ControlFile.IniReadValue("Control", HC, null);
  int simil=0;
  if (HashControl.Length==64)
  {
  for (int i = 0; i < 64; i++)
  {
  if (HashControl.ElementAt(i) == HashCheck.ElementAt(i))
  {
   simil += 1;
  }
  }
  }

  if (simil > 64*AuthKoef)
  {
  NumDetect += 1;
  DI.IsDetected = true;
  string TypeDetect;
  if (HC.LastIndexOf('#') > -1) TypeDetect=HC.Substring(0, HC.LastIndexOf('#'));
  else TypeDetect=HC;
  if (!listDetect.Contains(TypeDetect)) listDetect.Add(TypeDetect);
  Bitmap tmpOK = Image.Clone(rc,System.Drawing.Imaging.PixelFormat.Format24bppRgb);
  tmpOK.Save(PathSave + "Detect_"+NumDetect+".bmp", System.Drawing.Imaging.ImageFormat.Bmp);

  Random rand = new Random();
  Color color = Color.FromArgb(rand.Next(992, 7168) / 32, rand.Next(992, 7168) / 32,
rand.Next(992, 7168) / 32);
  Pen myPen = new Pen(color,5);
  Graphics g = Graphics.FromImage(OutIm);
  g.DrawRectangle(myPen, rc);
  myPen.Dispose();
  g.Dispose();
  break;
  }
  }
 }
 return DI;
 }

 private String GetHashString(Bitmap Hash)
 {
 String HashString = "";
 var tempIm = new FastBitmap(Hash);
 for (int i = 0; i < Hash.Height; i++)
 {
 for (int j = 0; j < Hash.Width; j++)
 {

  if (tempIm.GetColor(j, i).R < 128)
  {
  HashString += "0";
  }
  else HashString += "1";
 }
 }
 tempIm.Dispose();

 return HashString;
```

```
 }
#endregion

 private Bitmap Combine(Bitmap Image, String CombF)
 {

 Rectangle cloneRect = new Rectangle(0, 0, Image.Width, Image.Height);
 Bitmap OutImage = Image.Clone(cloneRect, System.Drawing.Imaging.PixelFormat.Format24bppRgb);
 Grayscale Gfilter = new Grayscale(0.2125, 0.7154, 0.0721);
 Bitmap grayImage = Gfilter.Apply(OutImage);

 List<String> Filters = FiltersFile.IniReadValue("Combined", CombF, null).Split(',').ToList();
 foreach (String F in Filters)
 {
 if (F == "Erosion")
 {
  Erosion Efilter = new Erosion();
  Efilter.ApplyInPlace(OutImage);
 }
 else if (F == "Dilatation")
 {
  Dilatation Dilfilter = new Dilatation();
  Dilfilter.ApplyInPlace(OutImage);
 }
 else
 {
  int[1,] filter = FiltersFile.IniReadFilter("Filters", F, null);
  if (filter != null)
  {
  Convolution Cfilter = new Convolution(filter);
  Cfilter.ApplyInPlace(OutImage);
  }
  else
  {
  String LogText = DateTime.Now.ToString().Replace(':', '-') + " Ошибка: применяемый фильтр
\"" + F + "\" включенного в комбинированный фильтр \"" + CombF + "\"не найден в базе\r\n";
  System.IO.File.AppendAllText(PathLog, LogText);
  }
 }
 }
 return OutImage;
 }
 }
}
```

ПРИЛОЖЕНИЕ 3. СВИДЕТЕЛЬСТВА О РЕГИСТРАЦИИ ЭЛЕКТРОННЫХ РЕСУРСОВ

ГОСУДАРСТВЕННАЯ АКАДЕМИЯ НАУК
РОССИЙСКАЯ АКАДЕМИЯ ОБРАЗОВАНИЯ
ИНСТИТУТ НАУЧНОЙ И ПЕДАГОГИЧЕСКОЙ ИНФОРМАЦИИ
ОБЪЕДИНЕННЫЙ ФОНД ЭЛЕКТРОННЫХ РЕСУРСОВ "НАУКА И ОБРАЗОВАНИЕ"

СВИДЕТЕЛЬСТВО О РЕГИСТРАЦИИ ЭЛЕКТРОННОГО РЕСУРСА

№ 20205

Настоящее свидетельство выдано на электронный ресурс, отвечающий требованиям новизны и приоритетности:

Программный модуль для формирования латентного изображения

Дата регистрации: **11 июня 2014 года**

Автор: **Жарких А.А.**

Директор ИПИПР РАО,
д.п.н., профессор А.М. Романов

Руководитель ОФЭРНиО, советник
работник науки и техники А.Н. Галинина

Дата выдачи 11.06.2014

ГОСУДАРСТВЕННАЯ АКАДЕМИЯ НАУК
РОССИЙСКАЯ АКАДЕМИЯ ОБРАЗОВАНИЯ

ИНСТИТУТ НАУЧНОЙ И ПЕДАГОГИЧЕСКОЙ ИНФОРМАЦИИ

ОБЪЕДИНЕННЫЙ ФОНД ЭЛЕКТРОННЫХ РЕСУРСОВ "НАУКА И ОБРАЗОВАНИЕ"

СВИДЕТЕЛЬСТВО О РЕГИСТРАЦИИ ЭЛЕКТРОННОГО РЕСУРСА

№ 20206

Настоящее свидетельство выдано на электронный ресурс, отвечающий требованиям новизны и приоритетности:

Программный модуль для контроля латентных изображений

Дата регистрации: **11 июня 2014 года**

Автор: **Жарких А.А.**

Директор ИНИПИ РАО,
д.п.н., профессор _____ А.М. Резников

Руководитель ОФЭРР(и), Архивно-
библиотечных и технологий _____ А.Н. Гуляев

Дата выдачи 11 06 2014

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное научное учреждение

ИНСТИТУТ УПРАВЛЕНИЯ ОБРАЗОВАНИЕМ РОССИЙСКОЙ АКАДЕМИИ ОБРАЗОВАНИЯ

ОБЪЕДИНЕННЫЙ ФОНД ЭЛЕКТРОННЫХ РЕСУРСОВ "НАУКА И ОБРАЗОВАНИЕ"

(основан в 1991 году)

СВИДЕТЕЛЬСТВО О РЕГИСТРАЦИИ ЭЛЕКТРОННОГО РЕСУРСА

ИУО РАО
ОФЭРНиО

№ 22544

Настоящее свидетельство выдано на электронный ресурс, отвечающий требованиям новизны и приоритетности:

ПМ «Конвейерный захват латентных изображений с IP-камеры»

Дата регистрации: **02 февраля 2017 года**

Авторы: **Жарких А.А., Шагрова Г.В.**

Директор ФГБНУ ИУО РАО,
доктор экономических наук _____ С.С. Неустроев

Руководитель ОФЭРНиО, почетный
работник науки и техники России _____ А.И. Галкина

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное научное учреждение

ИНСТИТУТ УПРАВЛЕНИЯ ОБРАЗОВАНИЕМ РОССИЙСКОЙ АКАДЕМИИ ОБРАЗОВАНИЯ

ОБЪЕДИНЕННЫЙ ФОНД ЭЛЕКТРОННЫХ РЕСУРСОВ "НАУКА И ОБРАЗОВАНИЕ"

(основан в 1991 году)

СВИДЕТЕЛЬСТВО О РЕГИСТРАЦИИ ЭЛЕКТРОННОГО РЕСУРСА

ИУО РАО ОФЭРНиО

№ 22545

Настоящее свидетельство выдано на электронный ресурс, отвечающий требованиям новизны и приоритетности:

ПМ «Построение комбинированных фильтров для выявления латентных изображений»

Дата регистрации: **02 февраля 2017 года**

Авторы: **Шагрова Г.В., Жарких А.А., Топчиев И.Н.**

Директор ФГБНУ ИУО РАО,
доктор экономических наук С.С. Неустроев

Руководитель ОФЭРНиО, почетный
работник науки и техники Росии А.И. Галкина

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное научное учреждение

ИНСТИТУТ УПРАВЛЕНИЯ ОБРАЗОВАНИЕМ РОССИЙСКОЙ АКАДЕМИИ ОБРАЗОВАНИЯ

ОБЪЕДИНЕННЫЙ ФОНД ЭЛЕКТРОННЫХ РЕСУРСОВ "НАУКА И ОБРАЗОВАНИЕ"

(основан в 1991 году)

СВИДЕТЕЛЬСТВО О РЕГИСТРАЦИИ ЭЛЕКТРОННОГО РЕСУРСА

ИУО РАО ОФЭРНиО

№ 22547

Настоящее свидетельство выдано на электронный ресурс, отвечающий требованиям новизны и приоритетности:

ПМ «Автоматическое определение объекта по распознанному латентному изображению»

Дата регистрации: **02 февраля 2017 года**

Авторы: **Жарких А.А., Шагрова Г.В., Топчиев И.Н.**

Директор ФГБНУ ИУО РАО,
доктор экономических наук С.С. Неустроев

Руководитель ОФЭРНиО, почетный
работник науки и техники России А.И. Галкина

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное научное учреждение

ИНСТИТУТ УПРАВЛЕНИЯ ОБРАЗОВАНИЕМ РОССИЙСКОЙ АКАДЕМИИ ОБРАЗОВАНИЯ

ОБЪЕДИНЕННЫЙ ФОНД ЭЛЕКТРОННЫХ РЕСУРСОВ "НАУКА И ОБРАЗОВАНИЕ"

(основан в 1991 году)

СВИДЕТЕЛЬСТВО О РЕГИСТРАЦИИ ЭЛЕКТРОННОГО РЕСУРСА

ИУО РАО ОФЭРНиО

№ 22546

Настоящее свидетельство выдано на электронный ресурс, отвечающий требованиям новизны и приоритетности:

ПМ «Построение и хеширование контрольного скрытого объекта в латентном изображении»

Дата регистрации: **02 февраля 2017 года**

Авторы: **Шагрова Г.В., Жарких А.А.**

Директор ФГБНУ ИУО РАО,
доктор экономических наук _____ С.С. Неустроев

Руководитель ОФЭРНиО, почетный
работник науки и техники России _____ А.И. Галкина