

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

---

Институт кибернетики  
Направление подготовки 09.04.03 Прикладная информатика  
Кафедра программной инженерии

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

Тема работы
<b>Алгоритмическое и программное обеспечение локализации области лёгкого по данным компьютерной томографии</b>

УДК 004.42:616.24:620.179.152.1

Студент

Группа	ФИО	Подпись	Дата
8KM51	Н.В. Григорьева		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент каф. ПИ	С.В. Аксёнов	К.Т.Н.		

**КОНСУЛЬТАНТЫ:**

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
ассистент каф. МЕН	К.А. Баннова	К.Э.Н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент каф. ЭБЖ	М.И. Пустовойтова	К.Х.Н.		

**ДОПУСТИТЬ К ЗАЩИТЕ:**

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
ПИ	М.А. Иванов	К.Т.Н.		

Томск – 2017 г.

## Планируемые результаты обучения по ООП

Код Результата	Результат обучения (выпускник должен быть готов)
<b><i>Профессиональные компетенции</i></b>	
P1	Применять глубокие естественнонаучные и математические знания для решения научных и инженерных задач в области информатики и вычислительной техники.
P2	Применять глубокие специальные знания в области информатики и вычислительной техники для решения междисциплинарных инженерных задач.
P3	Ставить и решать инновационные задачи инженерного анализа, связанные с созданием аппаратных и программных средств информационных и автоматизированных систем, с использованием аналитических методов и сложных моделей.
P4	Выполнять инновационные инженерные проекты по разработке аппаратных и программных средств автоматизированных систем различного назначения с использованием современных методов проектирования, систем автоматизированного проектирования, передового опыта разработки конкурентно способных изделий.
P5	Планировать и проводить теоретические и экспериментальные исследования в области проектирования аппаратных и программных средств автоматизированных систем с использованием новейших достижений науки и техники, передового отечественного и зарубежного опыта. Критически оценивать полученные данные и делать выводы.
P6	Осуществлять авторское сопровождение процессов проектирования, внедрения и эксплуатации аппаратных и программных средств автоматизированных систем различного назначения.
<b><i>Универсальные компетенции</i></b>	
P7	Использовать глубокие знания по проектному менеджменту для ведения инновационной инженерной деятельности с учетом юридических аспектов защиты интеллектуальной собственности.
P8	Осуществлять коммуникации в профессиональной среде и в обществе в целом, активно владеть иностранным языком, разрабатывать документацию, презентовать и защищать результаты инновационной инженерной деятельности, в том числе на иностранном языке.
P9	Эффективно работать индивидуально и в качестве члена и руководителя группы, в том числе междисциплинарной и международной, при решении инновационных инженерных задач.
P10	Демонстрировать личную ответственность и ответственность за работу возглавляемого коллектива, приверженность и готовность следовать профессиональной этике и нормам ведения инновационной инженерной деятельности. Демонстрировать глубокие знания правовых, социальных, экологических и культурных аспектов инновационной инженерной деятельности.
P11	Демонстрировать способность к самостоятельному обучению, непрерывному самосовершенствованию в инженерной деятельности, способность к педагогической деятельности.

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

---

Институт кибернетики  
Направление подготовки 09.04.03 Прикладная информатика  
Кафедра программной инженерии

УТВЕРЖДАЮ:  
Зав. кафедрой ПИ

\_\_\_\_\_ М.А.Иванов \_\_\_\_\_  
(Подпись) (Дата) (Ф.И.О.)

**ЗАДАНИЕ**  
**на выполнение выпускной квалификационной работы**

В форме:

магистерской диссертации
--------------------------

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

Студенту:

Группа	ФИО
8KM51	Григорьева Нина Валерьевна

Тема работы:

Алгоритмическое и программное обеспечение локализации области лёгкого по данным компьютерной томографии	
Утверждена приказом директора (дата, номер)	от 01.03.2017 №1484/с

Срок сдачи студентом выполненной работы:	02.06.2017
--	------------

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

<b>Исходные данные к работе</b>	<p><i>наименование объекта исследования или проектирования: алгоритм аппроксимации контуров изображений, полученных по данным компьютерной томографии;</i></p> <p><i>исходные материалы: алгоритм и программное обеспечение для сегментации изображений, полученных путём компьютерной томографии, библиотека CTImageSegmentation.dll, научные публикации по теме аппроксимации контуров изображений.</i></p> <p><i>требования к результату: получить алгоритм и программное обеспечение для выделения области лёгкого на изображениях компьютерной томограммы, выделить существенные для человеческого восприятия объекты и сегменты.</i></p>
---------------------------------	--

<b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b>	аналитический обзор по литературным источникам этапов обработки изображений, алгоритмов сегментации, методов аппроксимации контуров изображений, обсуждение литературы, обсуждение аналогов программного обеспечения, выбор и реализация алгоритма сегментации и аппроксимации, тестирование работы выбранного алгоритма, обсуждение результатов выполненной работы, написание пояснительной записки, заключение по работе.
<b>Перечень графического материала</b>	Презентация Microsoft Power Point на 15 слайдах

<b>Консультанты по разделам выпускной квалификационной работы</b>	
<b>Раздел</b>	<b>Консультант</b>
<b>Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</b>	К.А. Баннова, ассистент каф. МЕН
<b>Социальная ответственность</b>	М.И. Пустовойтова, к.х.н., доцент каф. ЭБЖ
<b>Раздел, выполненный на иностранном языке</b>	Т.В. Сидоренко, к.п.н., доцент каф. ИЯ ИК
	Е.С. Чердынцев, к.т.н., доцент каф. ПИ
<b>Названия разделов, которые должны быть написаны на русском и иностранном языках:</b>	
1. Обзор литературы	
2. Описание и проектирование выбранной методики	
3. Разработка и реализация алгоритма локализации области лёгкого	
4. Анализ полученных результатов	
5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	
6. Социальная ответственность	
7. Literature review	

<b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b>	08.12.2016
---	------------

**Задание выдал руководитель:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент каф. ПИ	С.В. Аксёнов	к.т.н.		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8KM51	Н.В. Григорьева		

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

Институт кибернетики  
Направление подготовки 09.04.03 Прикладная информатика  
Уровень образования – магистр  
Кафедра программной инженерии  
Период выполнения – осенний / весенний семестр 2016/2017 учебного года

Форма представления работы:

магистерская диссертация

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

Срок сдачи студентом выполненной работы:	02.06.2017
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
10.12.2016	1. Обзор литературы	
15.01.2017	2. Описание и проектирование выбранной методики	
20.04.2017	3. Разработка и реализация алгоритма локализации области лёгкого	
07.05.2017	4. Анализ полученных результатов	
12.05.2017	5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	
18.05.2017	6. Социальная ответственность	
21.05.2017	7. Literature review	

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент каф. ПИ	С.В. Аксенов	К.Т.Н.		

**СОГЛАСОВАНО:**

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
зав. каф. ПИ	М.А. Иванов	К.Т.Н.		

## **Реферат**

Выпускная квалификационная работа содержит 124 с., 35 рис., 21 табл., 31 источник, 3 прил.

Ключевые слова: обработка изображений, сегментация изображений, контурный анализ, аппроксимация контуров изображения, кусочно-линейная аппроксимация.

Объектом исследования является алгоритм аппроксимации контуров изображений, полученных по данным компьютерной томографии.

Цель работы – повышение качества диагностики состояния лёгкого при помощи разработки алгоритмического и программного обеспечения для локализации области лёгкого по данным компьютерной томографии.

Методы проведения работы: исследование путём анализа литературы алгоритмов сегментации изображений и аппроксимации контуров; реализация выбранных методов в виде приложения, позволяющего на компьютерной томограмме выделить границу лёгкого в случае нормы.

Область применения: разработка системы классификации форм и степени развития лёгочных заболеваний на основе характеристик тканей лёгких человека.

Данное исследование носит научный характер и не несет за собой прямой экономической выгоды. Тем не менее, результаты данной работы могут быть применены в дальнейших исследованиях по данной тематике и использованы для разработки медицинского проекта, где увеличение точности постановки диагноза, уменьшение времени приёма пациента и уменьшение неоправданных затрат на лечение больного обуславливает экономический эффект.

## **Определения**

В данной работе применены следующие термины с соответствующими определениями.

**Сегмент изображения** - участок изображения, в котором содержатся однородные по определенному признаку изображения.

**Сегментация изображения** - процесс разделения изображения на сегменты.

**Контур** - это внешние очертания или обвод предмета, а именно граница предмета, которая отделяет объект от фона.

**Контурный анализ** - процесс, позволяющий нахождение, хранение и описание объектов, представленных в форме внешних контуров.

**Аппроксимация контуров** - научный метод, состоящий в нахождении функции близкой к заданной.



## **Обозначения и сокращения**

BMP - Bitmap picture.

DICOM - Digital Imaging and Communications in Medicine.

JPEG - Joint Photographic Experts Group.

ВКР - выпускная квалификационная работа.

КТ - компьютерная томография.

МРТ - магнитно-резонансная томография.

НИР - научно-исследовательская работа.

## Содержание

Введение .....	12
1. Обзор литературы.....	14
1.1.Основные этапы обработки изображений .....	14
1.2.Сегментация изображений .....	20
1.2.1.Понятие сегментации.....	20
1.2.2.Методы, используемые при сегментации медицинских изображений .....	22
1.3.Контурный анализ изображений .....	28
1.4.Программные продукты для постпроцессинговой обработки изображений.....	34
2.Описание и проектирование выбранной методики .....	38
2.1.Библиотека CTImageSegmentation.....	39
2.2.Выделение контуров на сегментированном изображении.....	41
2.3.Аппроксимация линейными сплайнами .....	41
3.Разработка и реализация алгоритма локализации области лёгкого.....	44
3.1.Алгоритм выделения контуров сегментированного изображения .....	44
3.2.Алгоритма кусочно-линейной аппроксимации контуров .....	45
3.3.Программная реализация алгоритма локализации области лёгкого.....	49
4.Анализ полученных результатов .....	51
4.1.Интерфейс разработанной программы.....	51
5.Финансовый менеджмент, ресурсоэффективность и ресурсосбережение .....	65
5.1.Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения .....	65
5.1.1.Потенциальные потребители результатов исследования .....	65
5.1.2.Анализ конкурентных технических решений .....	65
5.1.3.Технология QuaD .....	68

5.1.4.SWOT-анализ .....	69
5.2.Определение возможных альтернатив проведения научных исследований.....	70
5.3.Планирование научно-исследовательских работ .....	71
5.3.1.Структура работ в рамках научного исследования .....	71
5.3.2.Определение трудоемкости выполнения работ .....	73
5.3.3.Разработка графика проведения научного исследования .....	73
5.3.4.Бюджет научно-технического исследования.....	78
5.4.Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования .....	79
6.Социальная ответственность.....	84
6.1.Техногенная безопасность.....	85
6.2.Производственная безопасность.....	85
6.3.Экологическая безопасность .....	93
6.4.Безопасность в чрезвычайных ситуациях .....	94
Заключение.....	96
Список публикаций .....	97
Список использованных источников .....	98
Приложение А.....	102
Приложение Б .....	111
Приложение В.....	121

## **Введение**

Технологии 3D-моделирования данных компьютерной томографии (КТ) в настоящее время являются актуальными и всё активнее применяются врачами в медицинской практике, значительно повышая информативность исследований и оптимизируя этап предоперационного планирования.

Сегментированная 3D- визуализация органов человека с координатной привязкой к пациенту значительно облегчает понимание изображения и ускоряет процесс обучения для начинающих хирургов. 3D-представление сегментированных моделей органов человека является неотъемлемой частью всех хирургических симуляторов, которые имитируют хирургические действия на виртуальных моделях.

Для некоторых клинических задач необходимо понимать трёхмерную структуру данных, что является довольно сложной задачей и во многом зависит от опыта самого врача. Качественная обработка изображения является необходимым этапом решения подобных трудностей. Это повышает актуальность задачи эффективной обработки медицинских данных, заключающуюся в создании программного обеспечения для облегчения определения области патологий.

Предметом исследования данной выпускной квалификационной работы является локализация областей лёгкого на снимках компьютерной томографии пациентов с различными видами лёгочных заболеваний, формата DICOM, с применением сегментации и метода линейной аппроксимации контуров.

Целью работы является повышение качества диагностики состояния лёгкого при помощи разработки алгоритмического и программного обеспечения для локализации области лёгкого по данным компьютерной томографии. Для достижения установленной цели были поставлены и решены следующие задачи:

- исследование и выбор наиболее эффективных методов сегментации, аппроксимации;
- проектирование алгоритма, позволяющего выделять область лёгкого на КТ изображениях;
- разработка программного обеспечения локализации области лёгкого;
- тестирование работы системы при различных входных данных, сравнение результатов и обоснование эффективности работы системы в целом.

Особенностью разработки данного приложения является выявление на поперечном срезе тела пациента не только кости, но и области лёгкого. На обработанном программой изображении выделяется локализация лёгкого в норме, что более точно позволяет находить патологии. Изучение различных аналогов, целью которых также является обработка медицинских изображений, показало, что локализация области лёгкого не доработана ни в одной из систем.

Результаты, представленные в данной работе, могут быть использованы для дальнейших исследований в области обработки медицинских данных. В частности, наиболее перспективным направлением является разработка системы классификации форм и степени развития лёгочных заболеваний. Данный процесс будет осуществляться при помощи построения трёхмерной модели лёгких по снимкам компьютерной томографии. Система значительно повысит эффективность диагностики и лечения различных форм заболеваний.

## **1. Обзор литературы**

### **1.1. Основные этапы обработки изображений**

Одним из приоритетных направлений науки и техники является цифровая обработка изображений. Это объясняется тем, что изображения используются в качестве средства получения визуальной информации. Тот факт, что около 90 процентов информации человек получает при помощи зрения, давно превратился в банальное утверждение, однако не потерял своей значимости. Более того, объемы зрительной информации, которую приходится получать, воспринимать и анализировать, постоянно возрастают. Все большая роль при этом отводится техническим системам обработки и анализа изображений, как с целью преобразования поступающих данных для последующего предоставления человеку, так и с целью автоматического их анализа и извлечения информации, в том числе для выбора последующей реакции.

Наряду со значительным повышением уровня развития техники, самую существенную роль играют методы обработки. Они обеспечивают улучшение изображений для их наилучшего восприятия, а также анализ, распознавание и интерпретацию зрительных образов для принятия решений и управления поведением автономных технических систем.

Число областей, в которых исходные данные поступают в виде изображений, огромно. Здесь и различные системы наблюдения, технического зрения, мониторинга, видеотелефонии, которые регистрируют и передают огромные объемы видеоданных, и различные автономные системы, принимающие решение на основе анализа видеоинформации, и персонализированное телевизионное вещание, и полиграфия со значительно возросшим объемом иллюстраций в печатной продукции, и медицина[12].

Специфика медицинских изображений, как объекта диагностической информации, связана, в первую очередь со способами их получения. Например, ультразвуковые изображения характеризуются наличием мультипликативного

шума, а рентгеновские изображения обладают низким контрастом и динамическим диапазоном. Другой особенностью биомедицинского применения изображений, в отличие от технического зрения, является обязательный визуальный контроль изображения со стороны специалиста, принимающего окончательное решение на основе представленных данных.

Каждая процедура анализа медицинских изображений содержит в своей структуре несколько этапов предварительной обработки для создания условий, повышающих эффективность и качество выделения и распознавания искомых объектов. Методы предварительной обработки зависят от задач исследований, поэтому не ко всем изображениям должны применяться каждый из описанных процессов. Чем выше сложность задачи обработки изображений, тем большее число процессов требуется привлекать для решения этой задачи. На рисунке 1 представлены основные этапы цифровой обработки изображения.

### **Восприятие изображения**

Первый из процессов – восприятие изображения. Основная задача восприятия: выделить ряд проблем, требующих решения. Основные технические сложности, возникающие в данном случае:

- Трудности с восприятием фона изображения;
- Сложная геометрия искомых областей;
- Шум или дезориентирующая информация входных данных;

Для решения определенных на этапе восприятия трудностей переходим ко второму этапу.



Рис. 1. Основные стадии цифровой обработки изображений

### **Предварительная обработка изображения**

Улучшение изображения входит в число наиболее впечатляющих областей предварительной обработки данных. По существу, за методами улучшения изображений стоит идея выявления плохо различимых деталей или просто подчеркивания интересных характеристик на исходном изображении. Известными примерами улучшения является усиление контраста и фильтрация изображения[12].

Важно иметь в виду, что улучшение изображений основано на субъективных предпочтениях человеческого восприятия, которые связаны с тем, что именно считается «хорошим» результатом для конкретной задачи.



- улучшение контраста

КТ изображения часто являются малоконтрастными, это обусловлено ограниченностью диапазона воспроизводимых яркостей. Под контрастом обычно понимают разность максимального и минимального значений яркости. Путем цифровой обработки контраст можно повысить, изменяя яркость каждого элемента изображения и увеличивая диапазон яркостей. Для этого разработано несколько методов.

Например, метод пороговой обработки связан с преобразованием полутонового изображения в бинарное (две градации). Преобразование осуществляется для того, чтобы сократить информационную избыточность изображения, оставить в нем только информацию, которая нужна для решения конкретной задачи. В бинарном изображении должны быть сохранены определенные детали и исключены несущественные особенности (фон).

- фильтрация изображения

Изображения, подлежащие распознаванию, могут быть зашумлены. Наряду с полезной информацией они содержат различные помехи. Причиной тому могут быть искажения, вносимые в изображение предметами, активно отражающими свет (стекло, железо, водные поверхности), неравномерная прозрачность воздушного слоя, пыль, попавшая в объектив, качество используемой аппаратуры и др. Поэтому необходима дополнительная предварительная фильтрация изображения.

- сжатие изображения

Сжатие, как следует из самого названия, относится к методам уменьшения объема памяти, необходимого для хранения изображения, или сужения полосы пропускания канала, требуемой для его передачи. Со сжатием изображений знакомо большинство пользователей компьютеров, встречающих в именах графических файлов определенные расширения; например, jpg

используется в стандарте сжатия изображений, разработанном Объединенной группой экспертов по фотографии (Joint Photographic Experts Group — JPEG).

### **Сегментация изображения**

Сегментация разделяет изображение на составные части или объекты. В целом автоматическая сегментация принадлежит к числу самых трудных задач цифровой обработки изображений. Излишне подробная сегментация уводит процесс решения задачи обработки изображения на сложный путь, если требуется идентифицировать объекты по отдельности. С другой стороны, недостаточно подробная или же ошибочная сегментация почти неизбежно приведет к возникновению ошибок на финальной стадии обработки. В общем, чем точнее сегментация, тем больше шансов на успех при распознавании[13].

- морфологические методы сегментации

Морфологическая обработка связана с инструментами для извлечения таких компонент изображения, которые впоследствии могут быть полезны для представления и описания формы объекта.

- пороговые методы сегментации

Пороговая обработка обладает интуитивно понятными свойствами и проста в реализации. Существуют несколько основных видов пороговой сегментации, но базовыми являются только два: метод с глобальным порогом и метод с адаптивным порогом. Все остальные методы этого класса являются производными от двух упомянутых алгоритмов.

Метод с глобальным порогом является простейшим. После выбора глобального порога, происходит поэлементная проверка всего изображения. Процедура подразумевает разделение изображения на две области: первая относится к объекту, вторая — к фону. В данном случае успешность целиком зависит от того, насколько хорошо диаграмма поддается разделению.

Успешного применения данного метода можно ожидать в условиях контролируемого освещения.

Как только освещение становится неравномерным, исходное изображение можно разбить разделить на подобласти, в каждой из которых для сегментации ищется и используется свой порог. Основной проблемой здесь является задача разбиения изображения на подобласти и выбор для каждой из них своего порога. Поскольку порог зависит от характеристик подобласти изображения, такой порог называют адаптивным.

- методы наращивания областей

Данный метод представляют собой алгоритмы, рекурсивно выполняющие процедуру группировки пикселей в подобласти по заранее заданным критериям. Одним из основных методов здесь является метод водоразделов.

- текстурные методы сегментации

Текстурные методы сегментации опираются при анализе на диффузные (цвет, отражательная способность) свойства поверхности анализируемого объекта. Представленные в этой категории методы являются наборами сложных операторов, которые способны свести процесс распознавания поверхностей к простой задаче различения уровней яркости.

### **Выделение контуров на сегментированном изображении**

Выделение контуров — это выделение некой границы объекта на сегментированном изображении, которая отделяет его от фона и других объектов. Переход к рассмотрению только контуров объектов позволяет уйти от пространства изображения — к пространству контуров, что существенно снижает сложность алгоритмов и вычислений. А именно, дает возможность не рассматривать внутренние точки изображения и фон, тем самым значительно сокращается объем перерабатываемой информации[13].

## **Распознавание объектов**

Суть этапа распознавания объектов - понять, относится ли данные на изображении к классу искомым объектов. Данный этап чаще всего конечный этап обработки, лежащий в основе процессов интерпретации и понимания. Входными для распознавания являются изображения, выделенные в результате сегментации и, частично, отреставрированные.

### **1.2. Сегментация изображений**

#### **1.2.1. Понятие сегментации**

Сегментация является одной из главных задач анализа и обработки изображений. Сегментация – это процесс разбиения исходного цифрового изображения на множество сегментов. Сегменты объединяются по ряду характеристик: по текстуре, по цвету, по близости на изображении, по яркости, по глубине.

Целью сегментации является изменение и упрощение представления изображения для лёгкости и простоты при анализе. В основном, сегментация изображений применяется для выделения границ и объектов, границами служат линии, кривые и т.д. Уточним, что сегментация – это процесс присвоения каждому пикселю изображения таких меток, что пиксели с одинаковыми метками имеют общие визуальные характеристики. На рис. 2 изображён пример сегментации изображения.

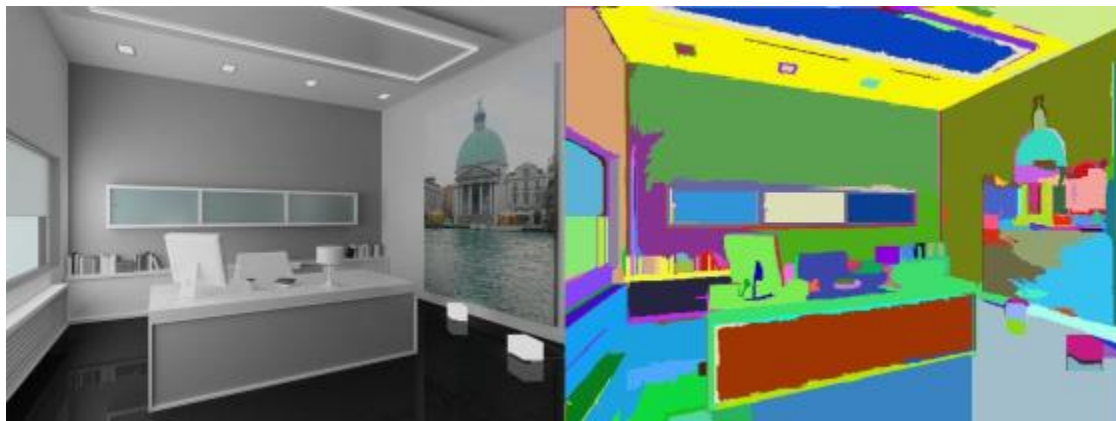


Рис. 2. Пример сегментации изображения

Классификация сегментации изображений:

1) Сегментация, основанная на анализе

- Сегментация, основанная на анализе высокого уровня
- Сегментация, основанная на анализе низкого уровня

2) Сегментация по признаку участия пользователя

- Автоматическая сегментация
- Интерактивная сегментация

Сегментация, основанная на анализе высокого уровня – процесс отделения объектов, находящихся на изображении, от фона, а также друг от друга.

Сегментация, основанная на анализе низкого уровня – это разделение изображения на сегменты схожих между собой пикселей.

Автоматическая сегментация – это сегментация, проводимая без участия пользователя (изображение на входе, а регионы на выходе).

Интерактивная сегментация – это сегментация, проводимая пользователем и требующая/допускающая ввод дополнительной информации (пример представлен на рис 3).



Рис. 3. Пример интерактивной сегментации, Photoshop , Magic Wand

Распознавание лиц, отпечатков пальцев, объектов на снимках со спутников, обнаружение стоп сигналов, машинное зрение и определение видов тканей и патологий на медицинских изображениях являются практическими применениями сегментации изображений. Существуют универсальные методы и алгоритмы сегментации изображений, но для эффективного решения задачи сегментации требуется совмещать базовые алгоритмы со знаниями из предметной области.

### **1.2.2. Методы, используемые при сегментации медицинских изображений**

Существует несколько разных методов сегментации изображений, которые различаются между собой по точности, сложности, возможности использования в автоматическом режиме, применимости для определенных органов, возможности анализа трехмерных изображений и другим параметрам.

Также все методы сегментации можно условно разбить на три поколения, в каждом из которых используются различные алгоритмы [1]. К первому поколению относятся низкоуровневые методы, которые не ссылаются на заранее известную информацию об объекте, что делает их уязвимыми по отношению к некоторым проблемам. Одним из представителей первого поколения является метод краёв, линий и точек, которые несут массу информации о различных областях изображения. Ко второму поколению относится метод нейронных сетей. К третьему поколению относят методы, основанные на использовании знаний более высокого уровня, например, заранее известной информации о структуре объекта, заданных пользователем правил, а также моделей объектов, например, активные контурные модели.

### **Сегментация, основанная на выделении краёв**

Край является границей между двумя различными регионами. На рис. 4 представлены различные типы краев. Линии же могут находиться внутри равномерной однородной области. Точка включена в равномерную однородную

область, и её полутон отличается от значения полутона области, в которой она находится.

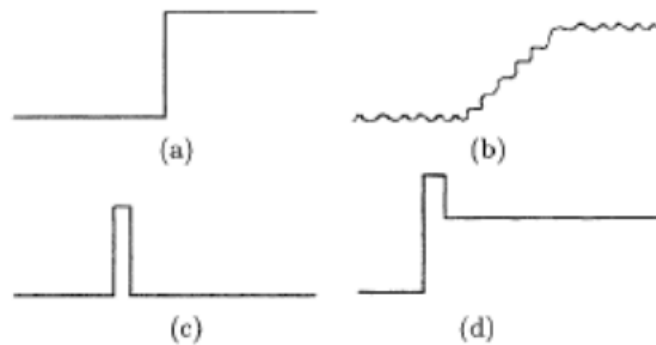


Рис. 4. Различные типы краев: (a) ступень, (b) наклон, (c) линия, (d) ступень-линия

Операция обнаружения края – это операция по выявлению значительных локальных изменений в уровнях интенсивности на изображении. Изменение уровня интенсивности определяется с помощью градиента. Так как функция  $f(x, y)$  является двумерной, то её градиент – вектор:

$$\begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \end{bmatrix} \quad (1)$$

Величину градиента можно вычислить несколькими способами:

$$\left. \begin{aligned} G | f(x, y) | &= \sqrt{G_x^2 + G_y^2}, \\ G | f(x, y) | &= | G_x | + | G_y |, \\ G | f(x, y) | &= \max\{| G_x |, | G_y |\}. \end{aligned} \right\} \quad (2)$$

Направление градиента:  $\theta(x, y) = \arctg(G_y / G_x)$ , где  $\theta$  – угол, который определяется по отношению к оси  $x$ . Градиентные операторы вычисляют

изменения в интенсивности полутонов а также направление, в котором происходит это изменение. Это рассчитывается по разности в значениях интенсивностей соседних пикселей, т.е. производных по оси  $x$  и по оси  $y$ . Данный метод подходит для сегментации изображений, на которых присутствует высокий контраст между различными объектами. К минусам метода можно отнести его восприимчивость к шуму и сложность в нахождении корреляции между интересующей областью и полученными границами, так как выделяются все границы на изображении. Граничные методы сами по себе не являются методами сегментации и, как правило, используются с другими алгоритмами сегментации для решения частной задачи [2].

### **Сегментация методом нейронных сетей**

Искусственные нейронные сети широко используются для сегментации изображений и классификации объектов. Это большой объем взаимосвязанных нейронов, которые сначала обучаются при помощи подходящих для данной цели данных, после чего уже могут использоваться для сегментации других подобных изображений. Нейронные модели зачастую эквивалентны соответствующим методам статистического распознавания образов. Также для сегментации медицинских изображений используются различные комбинации нейронных сетей и алгоритмов нечеткой логики [3]. Основные характеристики нейронных сетей – это устойчивость, чувствительность, оптимальный поиск и подавление шумов. Искусственные нейронные сети могут применяться в сегментации в качестве методов классификации и кластеризации. Методы сегментации, относящиеся ко второму поколению, весьма успешно применяются на практике, однако в общем случае они недостаточно эффективны для полностью автоматизированной и аккуратной сегментации медицинских изображений.



## Активные контурные модели

Активные контурные модели, или «змеи», как их часто называют, являются вариантом деформированных моделей, в которых начальные контуры алгоритмически деформируются в сторону краев изображения. В основном они используются для аппроксимации формы границ объекта в предположении, что границы гладкие. Название «змея» мотивировано поведением таких моделей, которые приспособливают контур между двумя контрольными точками. Активные контурные модели (пример такой сегментации представлен на рисунке 5) опираются на исходный контур, который либо предлагается пользователем, либо получается из априорного знания (касательно геометрических ограничений, ограничений данных, таких как диапазон ожидаемых полутонов, и знания, касающиеся формы объектов). Начиная с исходного контура, функционал энергии минимизируется на основе деформации контура и внешних сил изображения. Процесс оптимизации не может гарантировать, что глобальный минимум на самом деле найден. Вместо этого принимается локальный минимум на основе начального контура. Функционал энергии с параметрическим описанием кривой  $T \ v(s) = (x(s), y(s))$ , где  $x(s)$  и  $y(s)$  представляют собой координаты вдоль кривой  $s \in [0,1]$ , описывается следующим соотношением:

$$E_{\text{contour}} = \int_0^1 [E_{\text{int}}(v(s)) + E_{\text{ext}}(v(s))] ds. \quad (3)$$

Внутренняя энергия  $\text{int } E$  представляет собой гладкость кривой и может быть гибко параметризована с помощью  $\alpha$  и  $\beta$ , чтобы кодировать ожидания относительно гладкости и эластичности контура целевой структуры. Величина значения  $\alpha$ , например, сжимает кривую. Часто  $\alpha$  и  $\beta$  – константы.

$$E_{\text{int}} = \alpha(s) \frac{|dv|^2}{|ds|} + \beta(s) \frac{|d^2v|^2}{|ds^2|} \quad (4)$$

Внешняя энергия  $E_{\text{ext}}$  противодействует внутренней энергии и получается с помощью значений полутонов и градиента изображения в соответствии с уравнением:

$$E_{\text{ext}} = \omega_1 f(x, y) - \omega_2 |\nabla(G_\sigma(x, y) * f(x, y))|^2 \quad (5)$$

где  $\omega_1$  и  $\omega_2$  – веса, которые представляют собой влияние значений полутонов  $f(x, y)$  и градиента  $\nabla(G)$ . Значения полутонов предполагаются нормально распределенными. Значение  $\sigma$  характеризует стандартное отклонение от этого распределения [4].

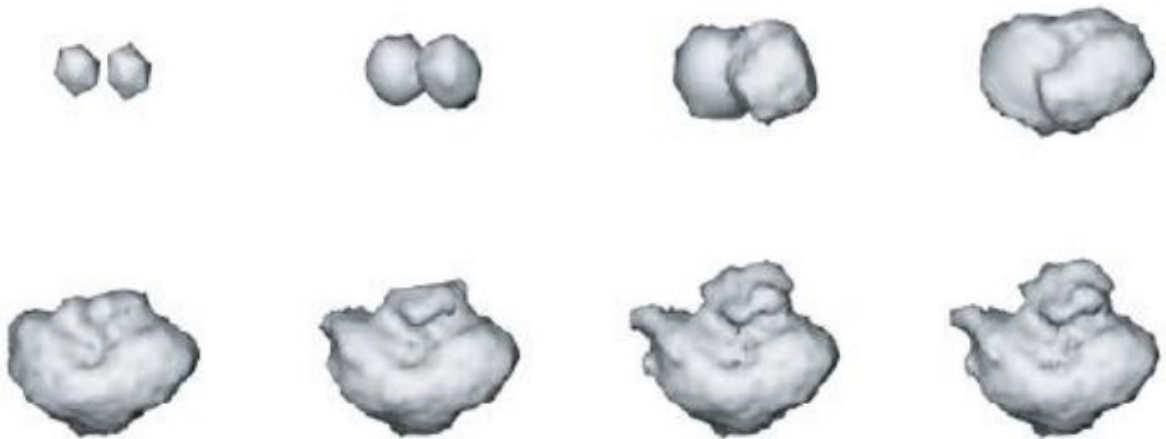


Рис. 5. Сегментация с помощью активных контурных моделей начинается с двух небольших объёмов (левый верхний угол) и через 40 итераций достигает конечного результата (правый нижний угол).

Нейронно-сетевой подход к сегментации изображений имеет преимущества перед традиционными математическими методами. Содержание нейронно-сетевой концепции применительно к задаче визуализации

определяется общими принципами функционирования многослойных перцептронов и включает в себя три этапа:

- а) сбор и подготовку данных;
- б) обучение перцептрона
- в) распознавание.

Следует отметить, что в отличие от стандартных математических процедур реконструкции важным этапом нейронно-сетевых вычислений является первый этап — этап представления и анализа данных измерений. Именно от того, в какой форме данные представлены, как произведены их предварительный отбор и фильтрация в определяющей степени зависит быстроедействие современных алгоритмов обучения, а, в конечном счете, и способность нейронной сети к запоминанию (выделению характерных закономерностей в обучающих данных) и обобщению (адекватной обработке входных сигналов, не использовавшихся для обучения) [5].

Несмотря на то, что сегментация изображений может осуществляться огромным количеством методов, далеко не все из них подходят для автоматической сегментации по различным причинам. В первую очередь, методы, относящиеся к различным поколениям, отличаются между собой по точности и сложности, которые часто обратно пропорциональны друг другу. Во-вторых, некоторые методы могут существовать, как часть какого-то другого алгоритма сегментации и не могут быть использованы в автоматическом режиме. Структура исследуемого объекта также вносит ограничения на возможность применения некоторых методов. Так, например, в случае метода активных контурных моделей ограничение внутренней энергии моделями снижает их гибкость и затрудняет формирование трубчатых и ветвящихся структур или объектов с сильными выступами, что делает их плохо применимыми для сегментации сосудов и бронхиальных деревьев.

### **1.3. Контурный анализ изображений**

Анализ изображения — это процесс выделения нужной информации из изображения с помощью автоматических систем.

Контур - это внешние очертания или обвод предмета, а именно граница предмета, которая отделяет объект от фона. Под контуром понимается пространственно-протяженный разрыв, перепад или скачкообразное изменение значений яркости. Существует ряд проблем при выделении контуров изображения:

- разрывы контура в местах, где яркость меняется не слишком быстро;
- наличие ложных контуров вследствие шума на изображении;
- широкие контурные линии из-за размытости или шума.

Признак изображения — это его простейшая характеристика или свойство. Некоторые признаки являются естественными, они устанавливаются визуальным анализом изображения, а другие, так называемые искусственные признаки, получаются в результате его специальной обработки и измерений. К естественным признакам относятся яркость, текстура различных областей изображения и форма контуров объектов[6].

Контурный анализ является совокупностью методов выделения, описания и преобразования контуров изображений и распознавания зрительных образов. При рассмотрении какого-либо объекта в сознании человека формируется зрительный образ. При восприятии глаз отслеживает линию контура, что приводит к созданию в сознании образа с характерными деталями. Существует мнение, что при восприятии в сознании человека формируются два образа: контура и внутренней части изображения. Такой подход позволяет не рассматривать внутренние точки изображения и тем самым значительно сократить объем перерабатываемой информации. Приведенные данные подчеркивают важную роль контуров при распознавании изображений [7].

Обычно анализ изображения включает в себя получение внешнего контура изображенных объектов и запись координат точек этого контура. Чаще всего требуется получить внешний контур в виде замкнутой кривой или совокупности отрезков дуг. Имеется три общих подхода к представлению границ объекта:

- аппроксимация кривых;
- прослеживание контуров;
- связывание точек перепадов.

### **Аппроксимация кривых**

Существует множество прикладных задач, при решении которых требуется построить кривую, проходящую вблизи экспериментальных точек. Таким образом, возникает задача аппроксимации.

Методы аппроксимации кривых, можно использовать для контурной сегментации путем подбора аналитически заданной кривой к совокупности точек контурного препарата, полученного с помощью детектора перепадов. Если имеется априорная информация относительно ожидаемой формы объекта, например известно, что это прямоугольник или окружность, то в качестве аппроксимирующей кривой нужно взять прямоугольник или окружность. Или же можно попытаться применить аппроксимацию полиномами, используя итеративные методы подбора прямых линий. Метод подбора кривых достаточно хорошо подходит для объектов с простой структурой. Трудности возникают, когда изображение содержит множество перекрывающихся объектов и точки перепадов образуют ветвистые структуры.

### **Прослеживание контуров**

Прослеживание контуров напоминает поведение жука, обходящего препятствие. В случае двоичного изображения, например показанного на рис. 6

(левое изображение) воображаемый жук начинает свой путь на белом поле и движется по направлению к области черных элементов изображения.

После того как жук пересечет черный элемент, он поворачивается налево и переходит к следующему элементу. Если этот элемент черный, жук снова поворачивается налево, если же элемент оказывается белым, то жук поворачивается направо. Эта процедура продолжается до тех пор, пока жук не вернется в исходную точку. Декартовы координаты точек перехода с черного на белое или с белого на черное дают местоположение границы.

На рис. 6 (левое изображение) выступающий элемент в нижнем правом углу объекта, обладающий восьмисвязностью со своим ближайшим соседом, не включен в границу. Заметим, однако, что этот же выступающий элемент на рис. 6 (правое изображение) включен в границу объекта, но здесь начальная точка передвинута. Таким образом, определение границы зависит от точки начала движения.

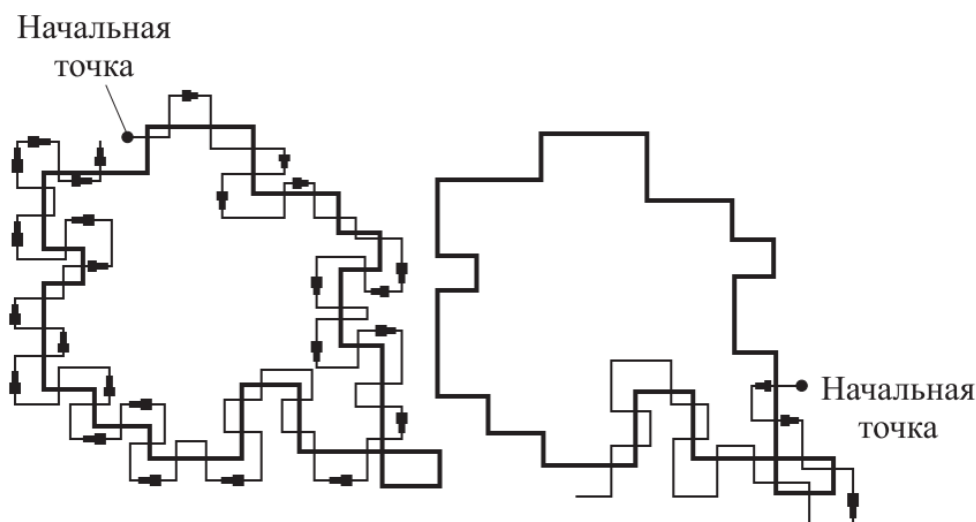


Рис. 6. Иллюстрация метода прослеживания контуров

Другая трудность связана с наличием «дыр» внутри объекта, которые жук может пропустить. Проблемы такого характера можно преодолеть, снабдив жука некоторой памятью и интеллектом, что позволит ему запоминать свои

прошлые шаги и возобновлять прослеживание, если данный путь оказывается ошибочным.

Описанный алгоритм чаще всего применяется к яркостным изображениям, подвергнутым пороговому ограничению, но данную идею легко можно распространить и на многоградационные изображения. Один из подходов состоит в том, что граничная точка отмечается, если разница яркостей соседних элементов изображения достаточно велика. В результате жук формирует контуры по ходу своего движения.

### **Соединение точек перепадов**

Объект или область изображения, которую желательно выделить, обычно отделяется от соседних частей изображения перепадом яркости. Поэтому соединение точек перепадов можно использовать для сегментации.

Метод соединения точек перепадов, разработанный Робертсом [9], основан на тех же основных принципах, что и большинство других методов такого связывания.

В методе Робертса аналоговые значения градиентов (разностей яркости соседних элементов) анализируются блоками размером  $4 \times 4$  элемента. Элемент с наибольшим в блоке значением модуля градиента считается пробной контурной точкой, если это значение больше порога. Затем к значениям градиента около этой пробной точки подбираются линии длиной в пять элементов с ориентацией «север», «восток», «юг» и «запад». Если отношение наилучшей аппроксимации к наихудшей больше второго порога, то пробная контурная точка объявляется действительной и ей приписывается ориентация, соответствующая наилучшей аппроксимации.

Далее к парам контурных точек подбираются прямые линии, если эти точки расположены в смежных блоках размером  $4 \times 4$  элемента и если направление каждой линии находится в диапазоне  $\pm 23^\circ$  относительно

ориентации контура в каждой контурной точке. Точки, не удовлетворяющие критериям соединения, отбрасываются. Типичная граница, полученная на этом этапе, будет содержать, как видно из рис. 7, а, разрывы, и множественные соединения точек. Маленькие треугольники исключаются вычерчиванием наибольшей стороны, а маленькие четырехугольники заменяются их наибольшей диагональю (рис. 7, б). Короткие выступающие линии также уничтожаются. На этом этапе короткие разрывы заменяются мостиками из прямых линий. Этот вид соединения точек перепада можно использовать для широкого класса детекторов перепадов[10].

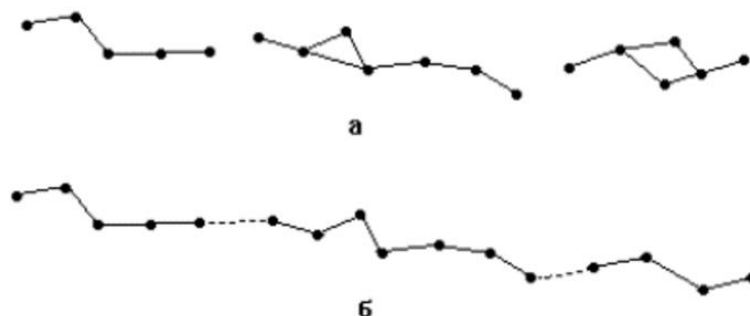


Рис. 7. Примеры соединения точек перепадов по правилу Робертса:

а - соединение точек перепадов;

б - исключение множественных соединений и замена их мостиками из прямых линий.

Подобный метод применен для соединения точек, полученных с помощью детектора перепадов Хьюккеля.

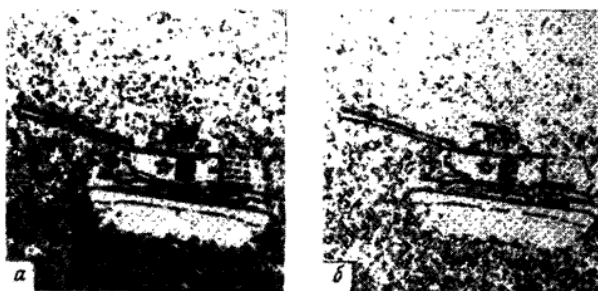


Рис. 8. Пример построения контуров с использованием алгоритма соединения точек перепада: а - без соединения; б — с соединением



Робинсон [9] предложила простой, но эффективный алгоритм соединения, при использовании которого точки перепада, полученные с помощью детектора, дающего возможность определить, ориентацию перепада в системе восьми курсовых направлений, исследуются по блокам из 3x3 элементов (рис. 9). Точка перепада в центре области считается действительной контурной точкой, если ее соседи по соответствующим направлениям принадлежат к перепадам надлежащей ориентации. Рис. 8 иллюстрирует возможность данного способа удалять явно посторонние перепады фона. Применение окон большего размера было бы полезно, однако с увеличением размера окна количество возможных связей действительного контура будет быстро расти.

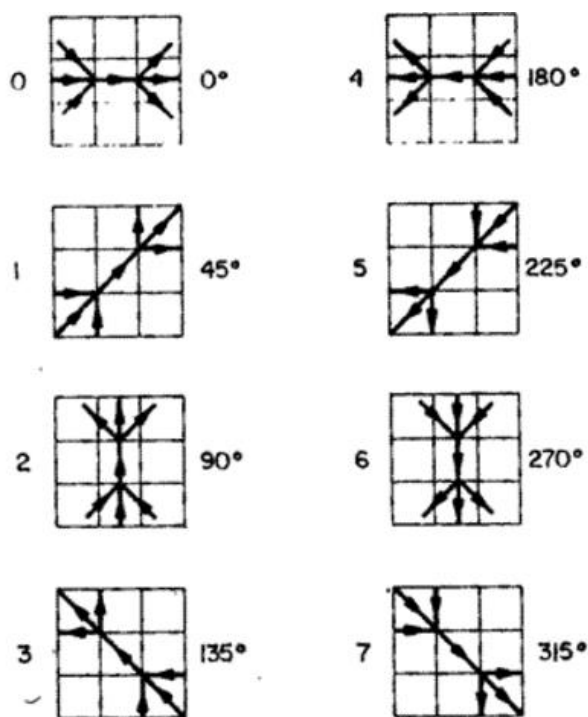


Рис. 9. Правила соединения точек перепадов

#### 1.4. Программные продукты для постпроцессинговой обработки изображений

В настоящее время существует достаточно большое количество продуктов для сегментации изображений. В большинстве своем эти продукты специфичны для различных органов. Ниже приведены примеры программного обеспечения для постпроцессинговой обработки КТ-изображений лёгких:

##### NIH-CIDI Lung Segmentation Tool

Команда разработчиков компании NITRC (Neuroimaging Informatics Tools and Resources Clearinghouse) совместно с медицинской школой университета Массачусетс создали свободно распространяемое программное обеспечение для сегментации КТ-изображений лёгких. Приложение может работать в полностью автоматическом и полуавтоматическом режимах, позволяет пользователю выполнять основные операции фильтрации и ручной коррекции сегментации, если это необходимо, а также выделять цветом необходимые регионы для дальнейшего рассмотрения.



Рис. 10. Результат работы приложения NIH-CIDI Lung Segmentation Tool

## NeuLungCARE

Платная система, основанная на анализе КТ-изображений лёгких. Может самостоятельно работать на ПК, обеспечивая обнаружение узелков лёгких, эмфиземы легких и количественный анализ их функций.

Особенности системы:

- Приложение предоставляет список информации об узелках лёгких, который может быть экспортирован в Excel;
- Доступно автоматическое определение эмфиземы и расчета значений её параметров;

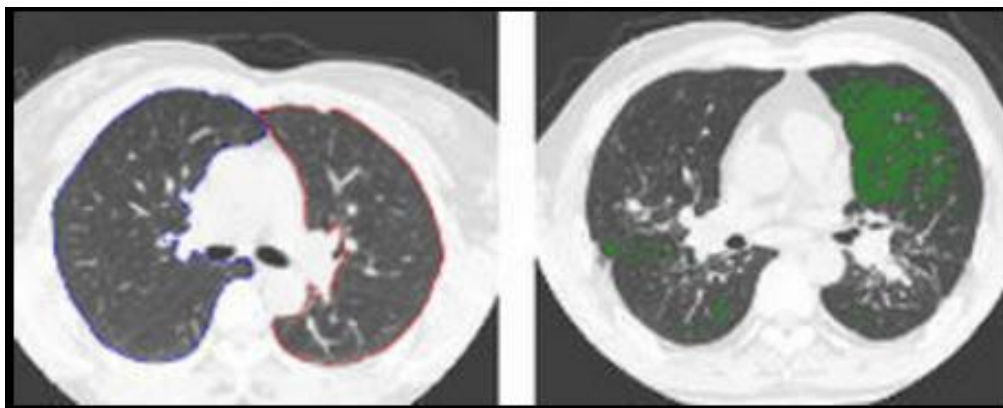


Рис. 11. Результат работы системы NeuLungCARE

## Cloud-Based SafeCT® for Low-Dose CT Lung Cancer Screening

Medic Vision Imaging Solutions - американская компания, разработавшая продукт Cloud-Based SafeCT. С помощью этой услуги изображения, полученные при безопасной низкодозной компьютерной томографии, пересылают по протоколу HTTPS на удаленный облачный сервер для немедленного их преобразования в изображения диагностического качества. Обработанные изображения через безопасное соединение сразу же отправляются обратно заказчику и направляются в систему PACS клиентов, которая доступна любому радиологическому отделению.

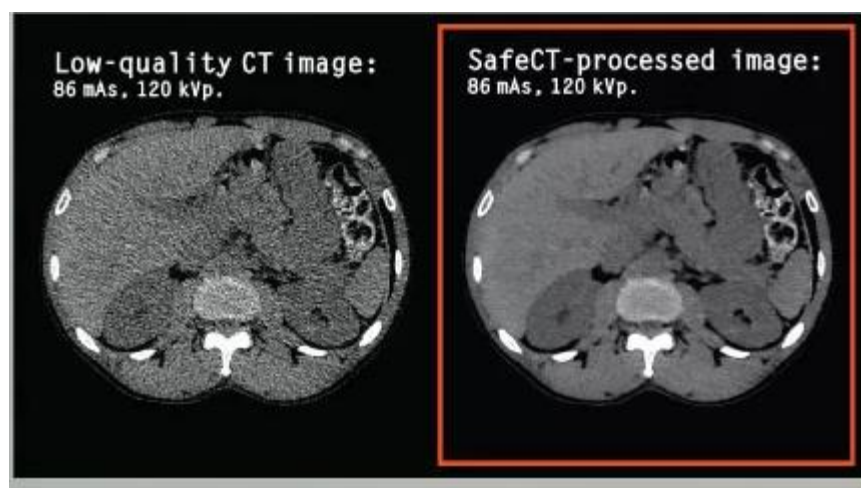


Рис. 12. Результат работы системы Cloud-Based SafeCT

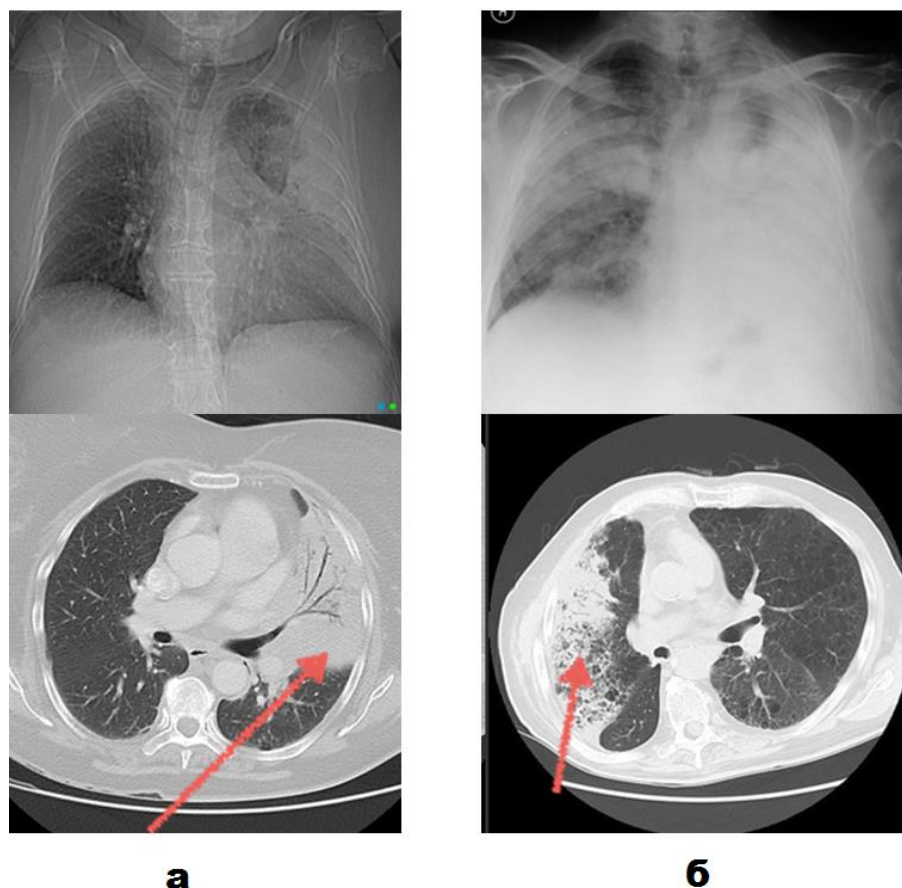


Рис. 13. Рентгенограмма и КТ

(а) пациент с пневмонией; (б) пациент с бронхоальвеолярной карциномой.

Если на рентгенограмме два данных процесса вызывали сомнения у специалистов, то результат компьютерной томографии развеял сомнения. На рис.13(а) у пациента пневмония, а на рис.13(б) рак лёгкого.

При разработке программного обеспечения по локализации области лёгкого выделяются области костей, и восстанавливается граница лёгкого в случае нормы. Это повышает эффективность диагностики, и не позволяет упустить участки патологий (например, уплотнение в лёгких, указанные красной стрелкой на рис. 13 а, б).

Это является преимуществом нашей разработки перед системой NIH-CIDI Lung Segmentation Tool компании NITRC, наиболее близкой по функционалу.

## 2. Описание и проектирование выбранной методики

Для решения задачи по созданию алгоритмического и программного обеспечения локализации области лёгкого были выделены три основных этапа обработки изображений компьютерной томографии.



Рис. 14. Этапы алгоритма локализации области лёгкого

- Сегментация КТ изображений с целью выделения костной ткани;

Сегментация медицинских изображений формата DICOM проводилась на основе динамической библиотеки CTImageSegmentation. Данная библиотека осуществляет пороговую обработку входных изображений.

- Выделение контуров сегментированного изображения

По завершению этапа выделения контуров получаем список белых пикселей, принадлежащих рёбрам.

- Аппроксимация выделенных контуров до получения замкнутой области лёгкого.

Для аппроксимации контуров, принадлежащих рёбрам, был выбран метод аппроксимации линейными сплайнами.

## 2.1. Библиотека **CTImageSegmentation**

Данная библиотека используется для сегментации набора изображений КТ-снимков легких. Название изображений - номера в трехзначном формате: 000, 001, 002, и т.д. В программе указывается номер сегмента, принадлежащего телу на снимке. Данная библиотека реализует сегментацию набора КТ-изображений на языке C++.

Библиотека **CTImageSegmentation** имеет одну функцию сегментации набора КТ-снимков легких *CtBodySegmentation*. Данная функция принимает на вход одномерный массив всех снимков набора КТ-изображений, записанных по строкам матрицы изображения (*byte\* intencity\_input*).

Одномерный массив, в который будет записан результат сегментации (*int\* regions\_output*), а также параметры КТ-изображений

- (*int images\_number*) номер изображения
- (*int image\_height*) высота изображения
- (*int image\_width*) ширина изображения

и параметры обработки

- (*int filter\_width = 9*) ширина фильтра
- (*int intencity\_threshold = 60* порог интенсивности.

Результатом работы функции является одномерный массив, имеющий размерность, совпадающий с размерностью входного одномерного массива

изображений, значениями которого являются индексы регионов, соответствующие пикселям массива изображений.

Те элементы массива, соответствующие пикселям входного набора изображений, которые во входном массиве, например, принадлежали телу, будут иметь один и тот же индекс на всех изображениях - например, индекс 1 принадлежит телу и тогда все значения выходного массива, соответствующие пикселям, принадлежащим телу во входном массиве, будут равны 1.

Также некоторые элементы выходного массива (находятся по краям изображений) равны – 1, данные элементы соответствуют пикселям, исчезнувшим в результате фильтрации. При обработке выходного массива это следует учитывать.

Сама функция возвращает индекс региона, соответствующего телу на изображении (все элементы выходного массива, соответствующие телу, имеют значение равное индексу региона тела), а в массив (*regions\_output*) будет записан массив-результат.

Полученная в результате сегментации матрица используется для отображения сегментов на изображении с помощью цвета. В данном случае сегмент, принадлежащий телу, отображается как белый сегмент. Остальные сегменты отображаются черным цветом.



## 2.2. Выделение контуров на сегментированном изображении

В контурном анализе контур кодируется последовательностью, состоящей из комплексных чисел.

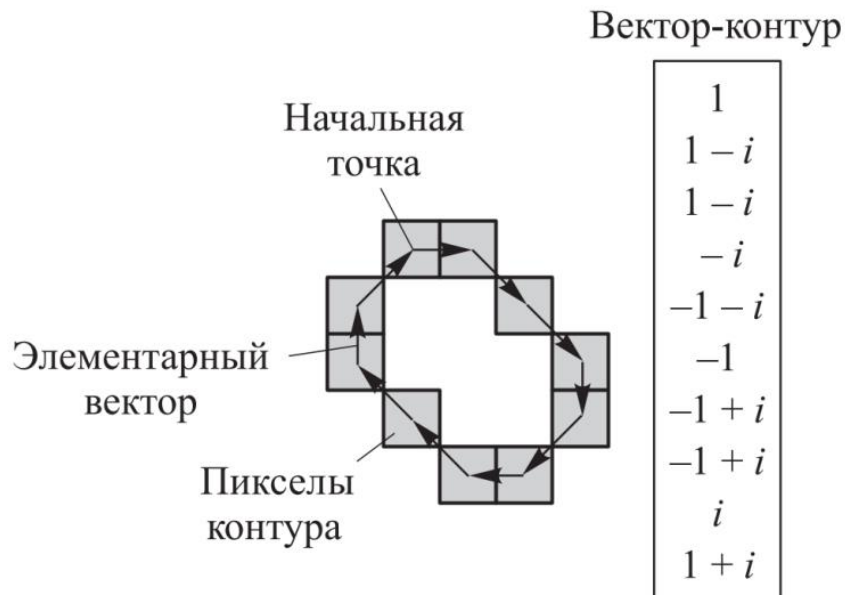


Рис. 15. Кодирование контура

На контуре фиксируется точка, которая называется начальной. Затем контур обходится (например, по направлению движения часовой стрелки), и каждый вектор смещения записывается комплексным числом  $(a + ib)$ , где  $a$  — смещение точки по оси  $x$ ;  $b$  — смещение точки по по оси  $y$ . Смещение берется относительно предыдущей точки (рис. 15) [8].

## 2.3. Аппроксимация линейными сплайнами

Аппроксимацией функции  $f(x)$  называется нахождение такой аппроксимирующей функции  $g(x)$ , полином, которой проходит как можно ближе к каждой из заданных точек и наилучшим образом отображает форму аппроксимируемой функции. Критерии близости функций могут быть различные. В случае если полином один для всей области аппроксимации, говорят, что аппроксимация глобальная. В случае если между различными узлами полиномы различны, говорят о кусочной аппроксимации.

Линейный сплайн - это сплайн, составленный из полиномов первой степени, т.е. из отрезков прямых линий.

На практике для аппроксимации наиболее часто используются кубические сплайны. Кубические сплайны дают высокую точность приближения и гладкость функции. Однако если аппроксимируемая функция имеет сильные перегибы, то в ряде случаев кубический сплайн дает выбросы. Сплайн первой степени в указанной ситуации выбросов не допускает. В некоторых случаях кусочно-линейная аппроксимация функции может оказаться предпочтительнее, чем аппроксимация более высокого порядка. Например, линейный сплайн сохраняет монотонность переданного в него набора точек.

Одним из способов описания выделенных на этапе сегментации контуров является кусочно-линейная аппроксимация.

### **Кусочно-линейная аппроксимация**

Наиболее простым и скоростным методом аппроксимации является кусочно-линейная аппроксимация. В данном случае для каждой пары узлов необходимо определить всего лишь два коэффициента  $a_0$  и  $a_1$ , при этом общее число коэффициентов, подлежащих определению, равно  $2(n-1)$  где  $n$  - общее число узлов[11].

Например, для кусочно-линейной аппроксимации может быть использован итеративный алгоритм подбора конечных точек. На первом этапе работы алгоритма конечные точки контура А и В соединяются прямой линией. Для всех оставшихся точек вычисляются расстояния до прямой АВ. Точка, имеющая наибольшее отклонение от прямой АВ, берется в качестве дополнительного узла. При этом кривая заменяется двумя отрезками АС и СВ (рис. 16). Процедура продолжается до тех пор, пока максимальное значение отклонения точек меньше заданного порога. Точность аппроксимации прямыми линиями определяется величиной порога.

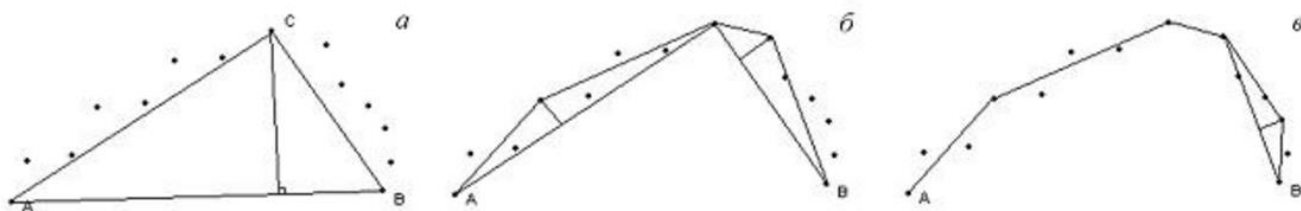


Рис. 16. Итеративный подбор конечных точек: а - первый этап; б- второй этап; в - третий этап

Основным недостатком кусочно-линейной аппроксимации является то, что аппроксимирующая функция не является гладкой (первые производные терпят разрыв в узлах сетки), а также зависимость результатов аппроксимации от исходных экспериментальных данных. Отсутствие гладкости функции не является существенным ограничением, в то время как зависимость результатов аппроксимации от начальных условий (точек А и В) часто не позволяет использовать данный метод аппроксимации для задач отождествления контуров.

### **3. Разработка и реализация алгоритма локализации области лёгкого**

В данном разделе пояснительной записки описываются принципы алгоритма локализации области лёгкого на изображениях компьютерной томографии на этапе выделения и аппроксимации контуров.

В качестве языка разработки программного обеспечения был выбран объектно-ориентированный язык программирования С#, разработка проводилась в интегрированной среде разработки программного обеспечения Microsoft Visual Studio 2015.

#### **3.1. Алгоритм выделения контуров сегментированного изображения**

Так как изображения компьютерной томографии хранятся в формате DICOM, преобразуем КТ изображения каждого пациента в более простой формат BMP.

Затем преобразуем BMP изображение в матрицу пикселей RGB и составим список координат всех точек региона. Регионом будем считать область рёбер, обозначенную белым цветом (рис.17).

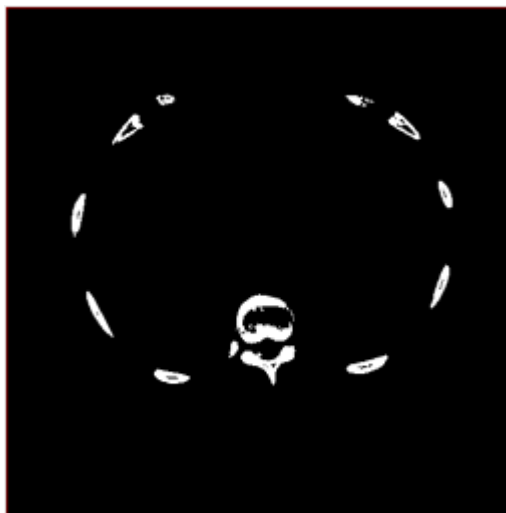


Рис. 17. Сегментированное изображение

Структура RGB модели заключается в трёх слоях red 0, green 1, blue 2, которые при смешении дают итоговый цвет. Условие поиска белых точек региона на сегментированном изображении будет выглядеть так:

$imageMatrix[0, j, i]=255 \ \&\& \ imageMatrix[1, j, i]=255 \ \&\& \ imageMatrix[2, j, i]=255,$

где  $j, i$  - координаты пикселей на изображении.

Координаты каждой точки, удовлетворяющей заданному условию, записываются в список *Contour*.

В итоге, мы получаем список белых пикселей, принадлежащих рёбрам.

### 3.2. Алгоритма кусочно-линейной аппроксимации контуров

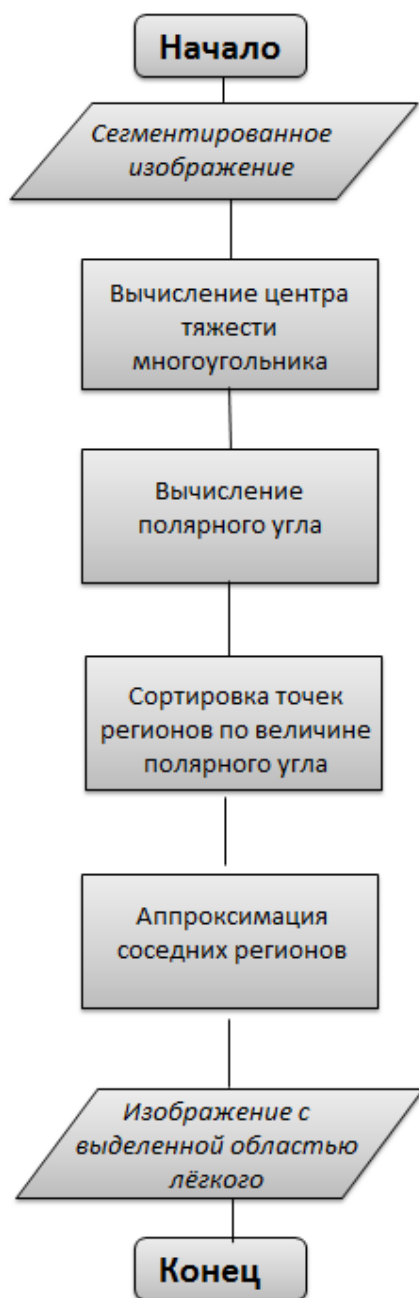


Рис. 18. Этапы алгоритма кусочно-линейной аппроксимации контуров

- **Вычислим центр тяжести многоугольника, ограниченного белыми пикселями**

Представим фигуру в виде многоугольника, которая задается множеством вершин. Каждый белый пиксель – вершина. Понятие "центр тяжести многоугольника" можно интерпретировать различными способами.

Рассмотрим вариант интерпретации, если масса находится только в вершинах, причем каждая вершина весит одинаково, в этом случае координаты центра тяжести выражаются по формулам:

$$CenterX = (M1 * X1 + \dots + Mn * Xn) / M \quad CenterY = (M1 * Y1 + \dots + Mn * Yn) / M$$

$(X_i, Y_i)$  - координаты  $i$ -ой вершины многоугольника,

$M_i$  - масса  $i$ -ой вершины.

$M$  - масса всех вершин ( $M = M1 + \dots + Mn$ ).

Таким образом, для нашего частного случая имеем:

$$CenterX = (X1 + \dots + Xn) / n$$

$$CenterY = (Y1 + \dots + Yn) / n$$

$$centerX = sumX / contour.Count;$$

$$centerY = sumY / contour.Count;$$

Таблица 1. Входные параметры а

Входной параметр	Тип	Значение
CenterX, CenterY	integer	Координаты центра тяжести.
sumX, sumY	integer	Сумма координат по X и Y. (X1 + ... + Xn), (Y1 + ... + Yn)
contour.Count	integer	n – количество вершин многоугольника (количество контуров).

- **Вычислим величину полярного угла для каждой точки региона, относительно центра тяжести.**

Чтобы определить полярную систему координат на плоскости, достаточно зафиксировать начало координат **O**, в нашем случае центр тяжести многоугольника, и задать единичный координатный вектор **j**. Точка **O** называется полюсом, а луч **OM**, сонаправленный с вектором **j** – полярной осью. Любая отличная от начала координат точка **R** плоскости однозначно определяется ориентированным углом  $\varphi$  между полярной осью и отрезком **OR**.

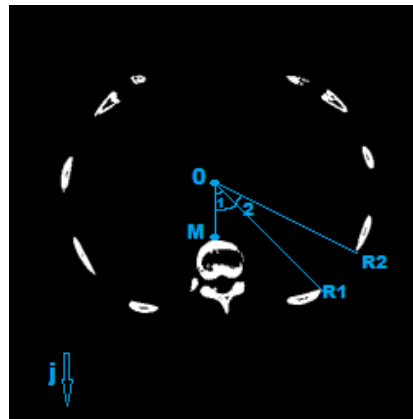


Рис. 19. Изображение полярных углов точек R1 и R2

Из треугольника OMR1 легко найти значение полярного угла  $\varphi$ . Тангенс острого угла треугольника – есть отношение противолежащего катета MR1 к прилежащему катету OM:  $\tan \varphi = \frac{MR1}{OM}$ , следовательно, сам угол:  $\varphi = \arctan \frac{MR1}{OM}$ . Записываем значение полученного угла в список *polarCoord*.

$double\ angle = Math.Atan2(contour[i].Y - centerY, contour[i].X - centerX);$

Таблица 2. Входные параметры б

Входной параметр	Тип	Значение
OM	integer	$contour[i].Y - centerY$
MR1	integer	$contour[i].X - centerX$
angle	double	Полярный угол

- **Отсортируем точки регионов по величине полярного угла**

По величине полярного угла можно определить, какие точки являются соседними. Сортировка необходима для того, чтобы соединять линейным сплайном только ближайшие друг к другу регионы. Отсортируем точки регионов списка *polarCoord*.

*polarCoord.Sort((x, y) => x.Item1.CompareTo(y.Item1));*

И сформируем итоговый отсортированный список точек *Contour*.

- **Соединим крайние точки каждого региона линией в порядке следования точек в отсортированном списке**

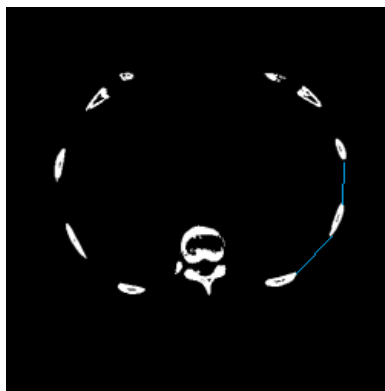


Рис. 20. Пример аппроксимации контуров линиями.

Возьмём за основу упорядоченный список точек *Contour*. Нам необходимо соединить все соседние крайние точки регионов, принадлежащие костям. Для этого используем метод *DrawLine* класса *Graphics*. Данный метод проводит линию, соединяющую две точки, которые заданы парой координат.

*g.DrawLine(pen1, contour[i], contour[i + 1]);*

Таблица 3. Входные параметры в

Входной параметр	Тип	Значение
pen1	Pen	Используемая кисть
contour[i]	List<Point>	Пара координат i-ой точки
contour[i + 1]);	List<Point>	Пара координат[ i+1]-ой точки



### 3.3. Программная реализация алгоритма локализации области лёгкого

Программа является приложением для рабочего стола Windows и имеет простой интерфейс, позволяющий определить локализацию области лёгкого и сохранить результат. Интерфейс представлен на рис.21.

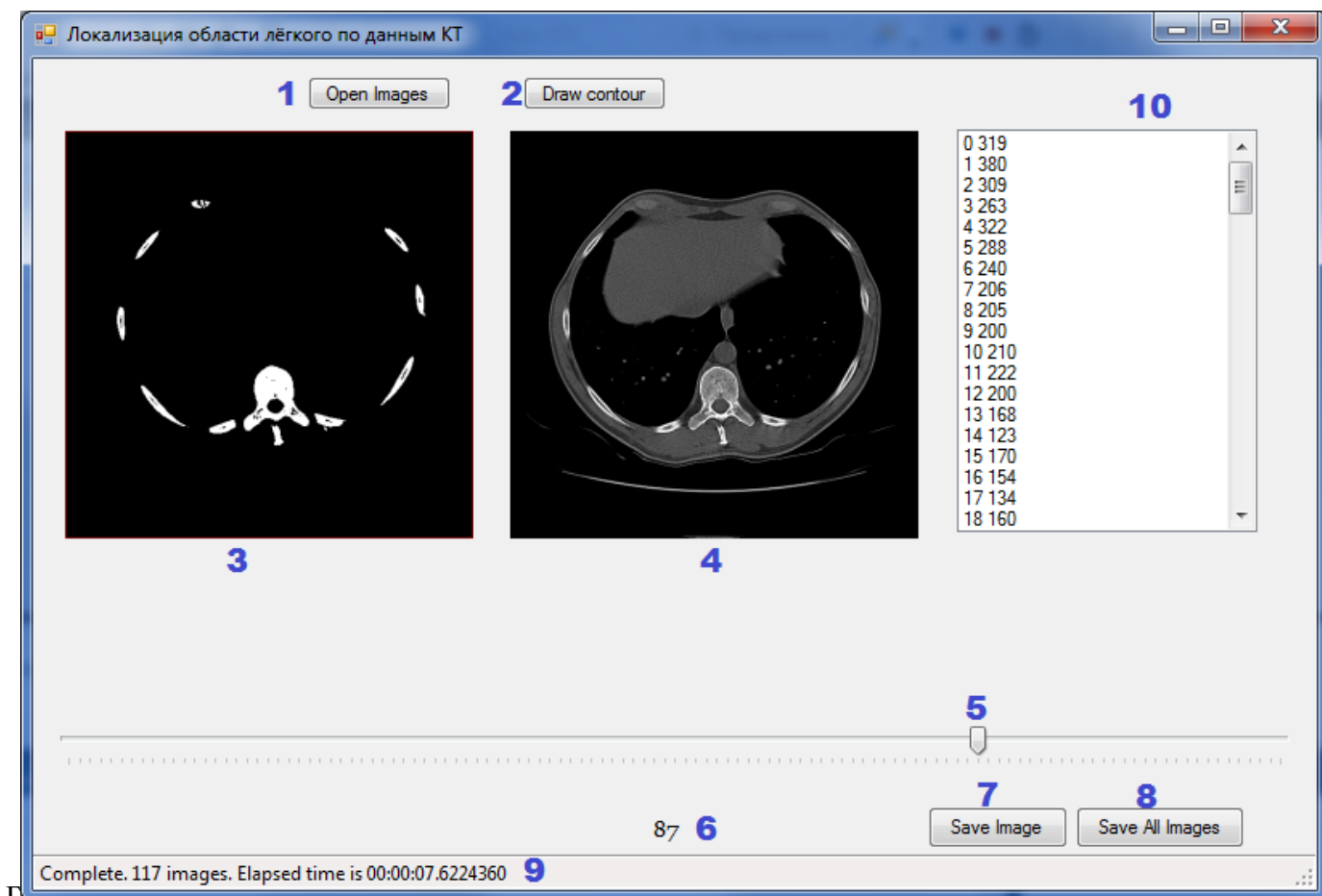


Рис. 21. Интерфейс приложения

Элементы графического интерфейса, обозначенные на рис. 18:

1. Кнопка, позволяющая в диалоговом режиме выбрать изображения для локализации области лёгкого.
2. Кнопка, позволяющая отобразить в левом picture box результат аппроксимации, заменяя при этом сегментированное изображение.
3. Изображение–результат сегментации (до нажатия кнопки Draw Contour) или результат аппроксимации (после нажатия кнопки Draw Contour).

4. Исходное изображение.
5. Полоса прокрутки.
6. Порядковый номер среза пациента, изменяется при движении полосы прокрутки.
7. Кнопка, позволяющая сохранить изображение-результат, отображенное в левом picture box.
8. Кнопка, позволяющая сохранить все изображение-результаты.
9. Отображение времени работы алгоритма локализации области легкого по данным КТ.
10. Список, отображающий порядкового номера изображения(imagesNumber) и числа регионов (subList.Count)

#### **4. Анализ полученных результатов**

В данном разделе пояснительной записки описываются результаты, полученные в ходе тестирования алгоритма локализации области лёгкого по данным КТ. А также приводятся сравнительные результаты работы алгоритма для пяти различных пациентов.

##### **4.1. Интерфейс разработанной программы**

Каждое КТ-изображение передает поперечный срез тела пациента (рис. 22 справа). При продольном перемещении пациента формируется множество таких срезов, которые в совокупности образуют трехмерное представление внутреннего строения тела с продольным разрешением, пропорциональным количеству срезов.

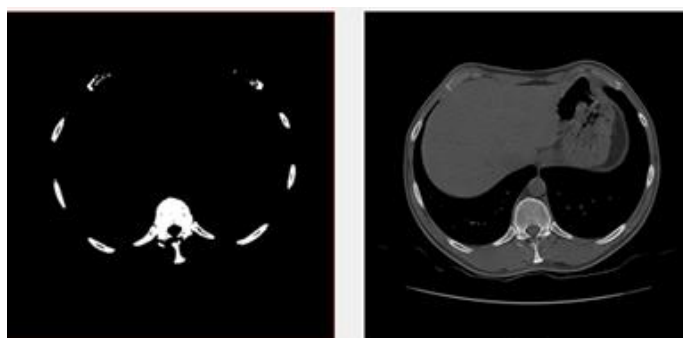


Рис. 22. КТ-изображение справа, сегментированное изображение слева

Кости на изображениях компьютерной томографии отображаются белыми несоединёнными между собой отрезками (рис 22. слева). Это происходит из-за того, что в один поперечный срез тела пациента попадает не одно ребро, а несколько (рис.23).

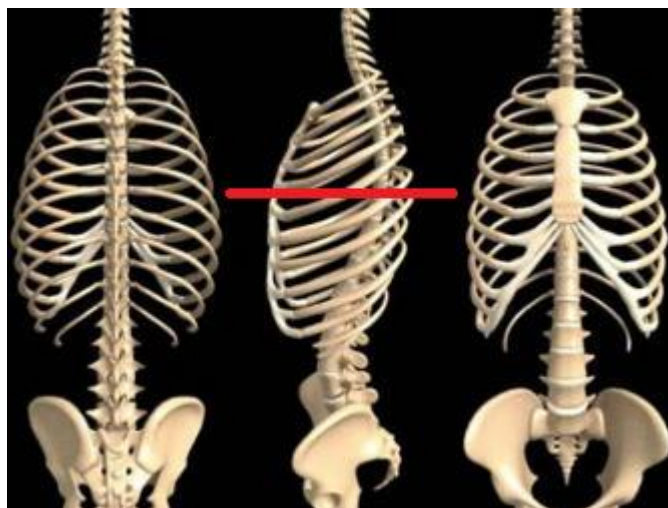


Рис. 23. Грудная клетка человека

После нажатия кнопки *Open Images* открывается диалоговое окно для выбора изображений. Формат входных изображений BMP.

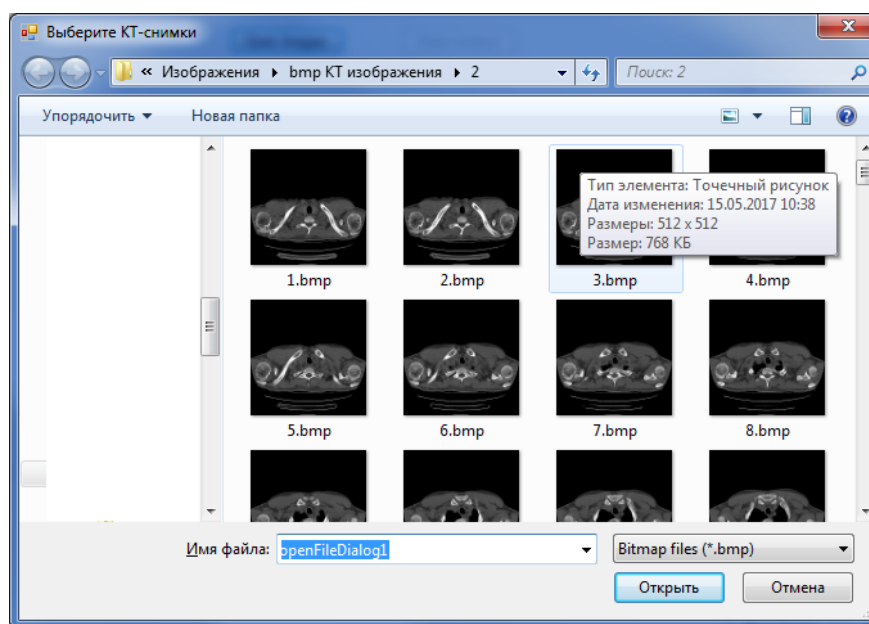


Рис. 24. Диалоговое окно для выбора изображений

## Пациент №1

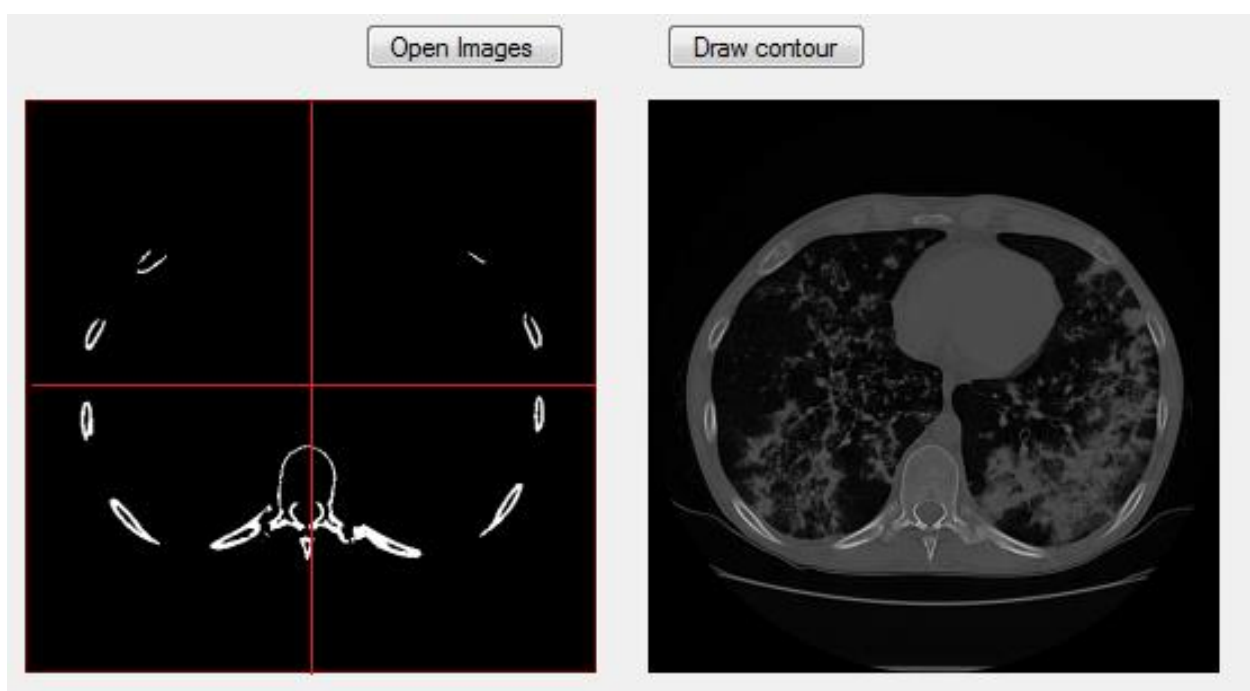


Рис. 25. Результат сегментации КТ изображения Пациента №1

После нажатия кнопки *Draw Contour* в левом picture box сегментированное изображение заменяется изображением результата аппроксимации.



Рис. 26. Результат аппроксимации контуров КТ изображения Пациента №1

Метод кусочно-линейной аппроксимации на тестовых изображениях пациента №1 показал верный результат, а именно выделена замкнутая область лёгкого в норме. Плавность соединения контуров – средняя. Количество обработанных изображений - 144. Время, затраченное на работу алгоритма – 15,114 секунды.

## Пациент №2



Рис. 27. Результат сегментации КТ изображения Пациента №2

После нажатия кнопки *Draw Contour*.

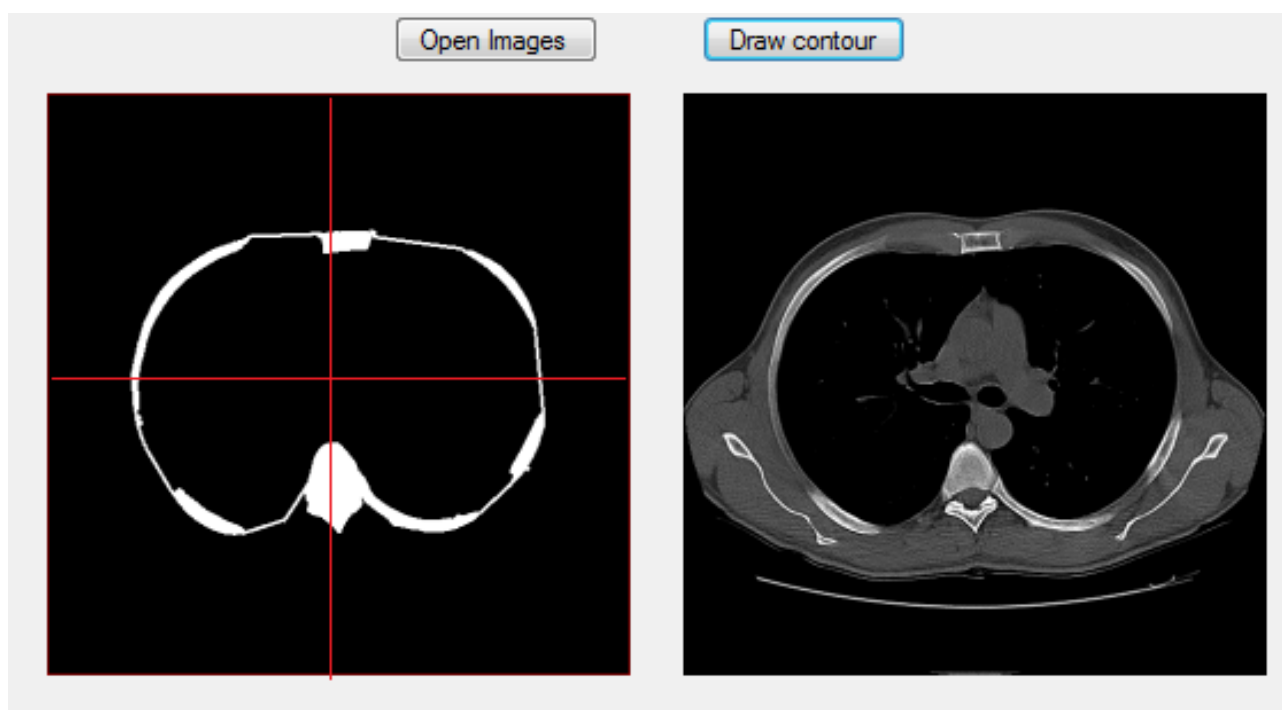


Рис. 28. Результат аппроксимации контуров КТ изображения Пациента №2

Метод кусочно-линейной аппроксимации на тестовых изображениях пациента №2 показал результат, лучше, чем аппроксимация контуров на КТ изображении пациента №1, а именно выделенная замкнутая область здорового лёгкого более округлая. Это связано с выбором другого номера среза и расположением тела пациента. Количество обработанных изображений - 117. Время, затраченное на работу алгоритма – 7,902 секунды. Время работы уменьшилось практически в 2 раза, отчасти из-за меньшего количества входных изображений.



### Пациент №3



Рис. 29. Результат сегментации КТ изображения Пациента №3  
После нажатия кнопки Draw Contour.



Рис. 30. Результат аппроксимации контуров КТ изображения Пациента №3

При помощи метода кусочно-линейной аппроксимации на тестовых изображениях пациента №3 выделена замкнутая область здорового лёгкого, плавность соединения контуров хуже, чем у пациента №2, но лучше, чем у пациента №1. Расположение тела пациента выше центра изображения. На результат работы алгоритма смещение тела человека не сказалось. Количество обработанных изображений - 115. Время, затраченное на работу алгоритма – 12,618 секунды.

## Пациент №4

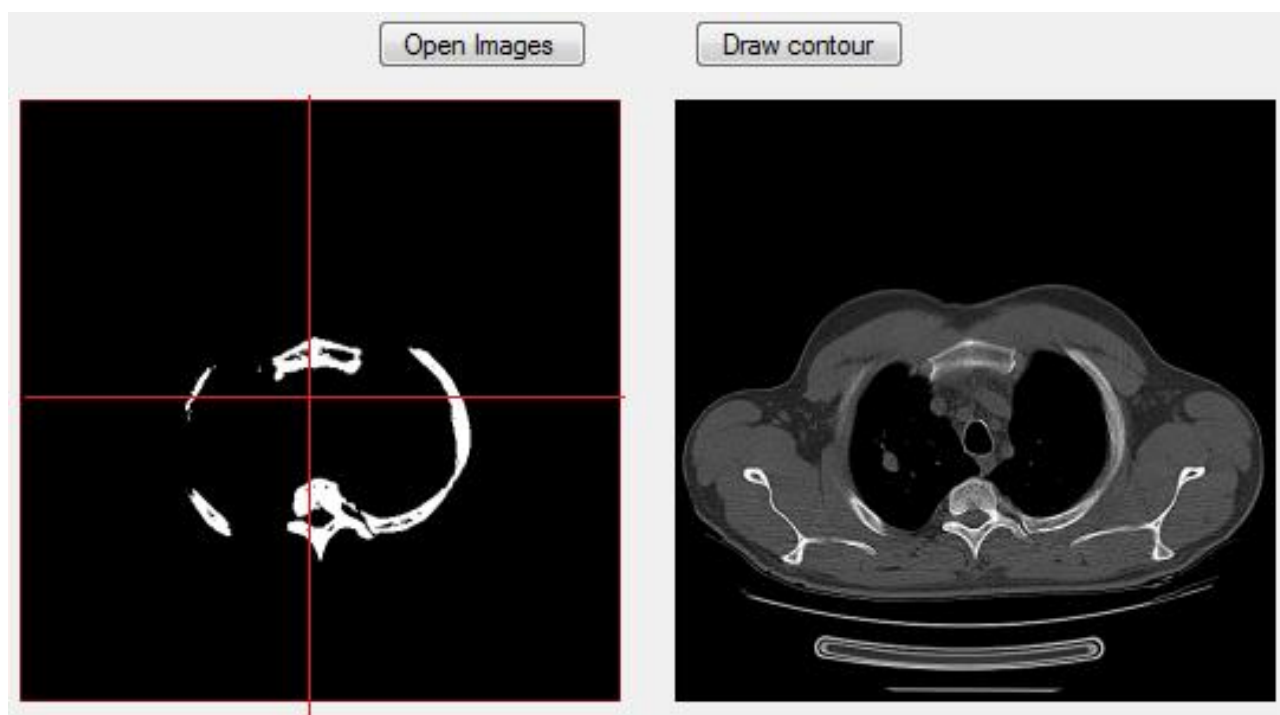


Рис. 31. Результат сегментации КТ изображения Пациента №4

После нажатия кнопки Draw Contour.

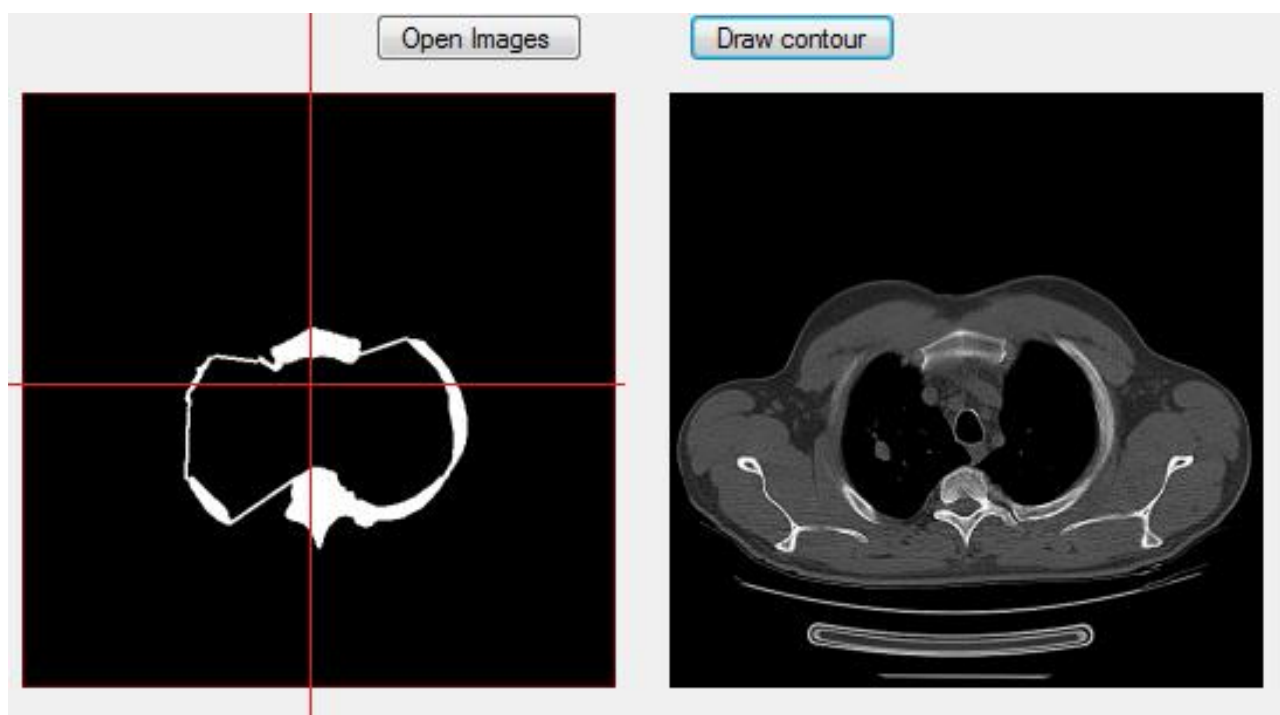


Рис. 32. Результат аппроксимации контуров КТ изображения Пациента №4

Метод кусочно-линейной аппроксимации на тестовых изображениях пациента №4 показал неверный результат, а именно некорректно выделена замкнутая область здорового лёгкого. Для рассмотрения был взят 15 срез, программа некорректно обрабатывает верхние срезы кт изображений тела пациента. Тело пациента немного смещено относительно центра изображения вправо. Количество обработанных изображений - 120. Время, затраченное на работу алгоритма – 12,945 секунды.

## Пациент №5

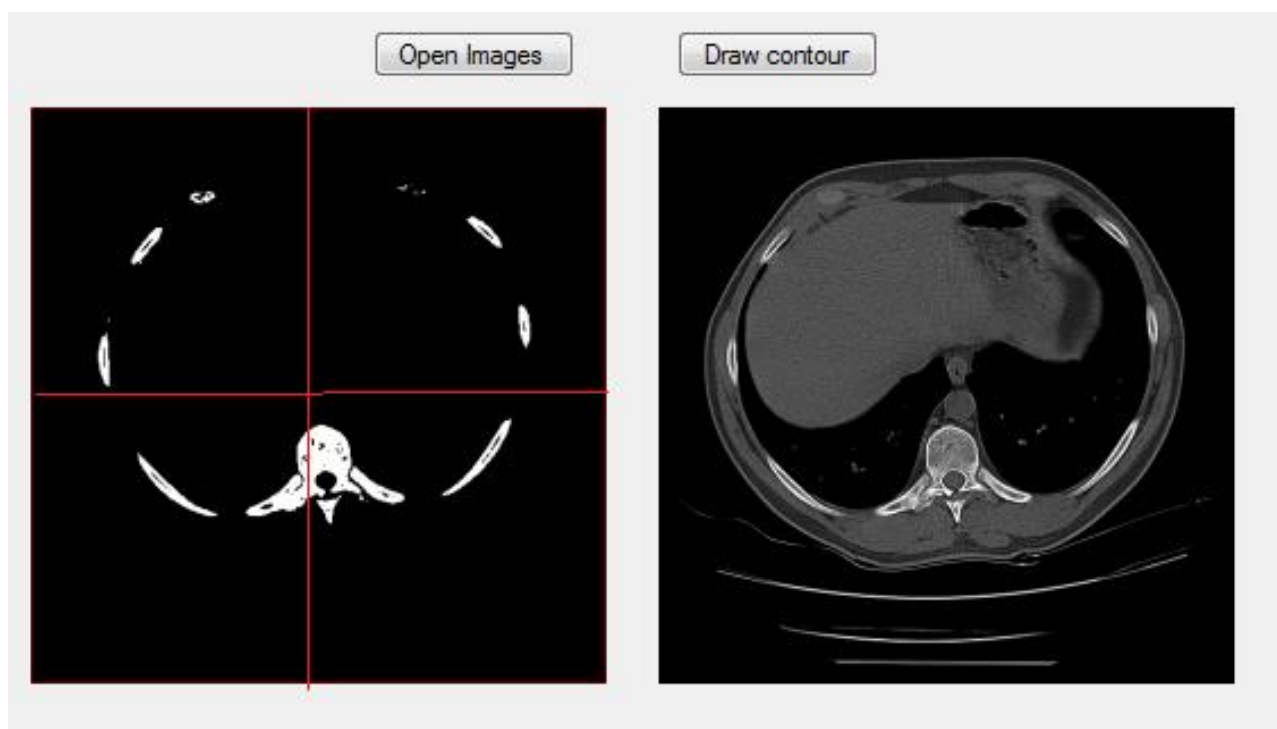


Рис. 33. Результат сегментации КТ изображения Пациента №5  
После нажатия кнопки Draw Contour.



Рис. 34. Результат аппроксимации контуров КТ изображения Пациента №5

Метод кусочно-линейной аппроксимации на тестовых изображениях пациента №5 показал верный результат, а именно выделена замкнутая область здорового лёгкого. Плавность соединения контуров – хорошая. Тело немного смещено относительно центра изображения вправо, но на результат это не повлияло. А также не повлияла на результат одежда, попавшая на КТ изображение. Количество обработанных изображений - 120. Время, затраченное на работу алгоритма – 13,250 секунды.

#### Вывод:

Разработанный алгоритм и программное обеспечение по локализации области лёгкого справляется с поставленной задачей. Шум на фоне не принимается во внимание. Смещение тела, относительно центра изображения также не сказывается на результате. Алгоритм работает некорректно на верхних срезах КТ изображений пациента. Среднее время обработки КТ изображений одного пациента 12,37 секунды.

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

Студенту:

<b>Группа</b>	<b>ФИО</b>
8KM51	Григорьева Нина Валерьевна

<b>Институт</b>	институт кибернетики	<b>Кафедра</b>	программной инженерии
<b>Уровень образования</b>	магистратура	<b>Направление/специальность</b>	09.04.03 прикладная информатика

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

1. <i>Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	<i>На основании информации, представленной в научных статьях и публикациях, аналитических материалах, статистических бюллетенях и изданиях, нормативно-правовых документах, определить методику расчета экономической эффективности.</i>
2. <i>Нормы и нормативы расходования ресурсов</i>	
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

1. <i>Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения</i>	<i>1. Оценка ресурсной, социальной эффективности НИ и потенциальных рисков;</i> <i>2. Планирование этапов разработки программы, определение трудоемкости, формирование бюджета НИИ;</i> <i>3. Сравнительный анализ интегральных показателей эффективности.</i>
2. <i>Планирование и формирование бюджета научных исследований</i>	
3. <i>Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования</i>	

**Перечень графического материала (с точным указанием обязательных чертежей):**

1. <i>Оценка конкурентоспособности технических решений</i>
2. <i>Матрица SWOT</i>
3. <i>Альтернативы проведения НИ</i>

4. <i>Временные показатели проведения НИ</i>
5. <i>Календарный план-график</i>
6. <i>Таблица проведения и бюджет НИ</i>

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

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
ассистент каф. МЕН	К.А. Баннова	к.э.н.		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8KM51	Григорьева Нина Валерьевна		



## **5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение**

Коммерческая привлекательность научного исследования определяется не только превышением технических параметров над предыдущими разработками, но и тем, насколько быстро разработчик сумеет найти ответы на следующие вопросы: будет ли продукт востребован рынком, какова будет его цена, каков бюджет научного проекта, какой срок потребуется для выхода на рынок и т.д.

Целью раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» является проектирование и создание конкурентоспособных разработок, технологий, отвечающих современным требованиям в области ресурсоэффективности и ресурсосбережения.

В рамках данной научной работы исследуется разработка алгоритмического и программного обеспечения локализации области лёгкого по данным компьютерной томографии.

### **5.1. Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения**

#### **5.1.1. Потенциальные потребители результатов исследования**

Осуществим анализ рынка потенциальных потребителей. В связи с тем, работа направлена на решение задачи анализа состояния лёгких человека с помощью исследования снимков компьютерной томографии, то разработанная программа предназначена для использования специалистами по лучевой диагностике, а также первичными потребителями являются лица, занимающиеся научно-исследовательской деятельностью.

#### **5.1.2. Анализ конкурентных технических решений**

В связи с тем, что конкуренты всегда были, есть и будут, то необходимо объективно провести анализ конкурирующих возможностей, чтобы точно

понять основные слабые и сильные стороны программного обеспечения локализации области лёгкого по данным компьютерной томографии.

Основными конкурентами являются:

- NIH-CIDI Lung Segmentation Tool. Свободно распространяемое программное обеспечение для сегментации КТ-изображений лёгких (в таблице присвоен номер «1»);
- NeuLungCARE. Платная система, основанная на анализе КТ-изображений лёгких (в таблице присвоен номер «2»);
- Cloud-Based SafeCT® for Low-Dose CT Lung Cancer Screening (соответственно, номер «3»).

Результаты конкурентного анализа приведены в таблице 1.

Таблица 4. Конкурентный анализ

№	Критерии оценки	Вес	Баллы			Конкурентоспособность		
			Б <sub>1</sub>	Б <sub>2</sub>	Б <sub>3</sub>	К <sub>1</sub>	К <sub>2</sub>	К <sub>3</sub>
	Технические критерии оценки ресурсоэффективности							
1.	Коэффициент загрузки программы	0,25	4	4	4	1	1	1
2.	Максимально возможное число изображений для сегментации	0,05	4	4	4	0,2	0,2	0,2
3.	Текущее число изображений для сегментации	0,05	3	3	3	0,15	0,15	0,15
4.	Время работы программы	0,15	3	4	5	0,45	0,6	0,75
5.	Количество обработанных изображений	0,25	2	4	5	0,5	1	1,25
6.	Количество необработанных изображений	0,15	5	2	2	0,75	0,3	0,3
	Экономические критерии оценки эффективности							
7.	Конкурентоспособность модели	0,05	1	3	5	0,05	0,15	0,25
8.	Законченность проекта	0,05	5	4	4	0,25	0,2	0,2
	Итого:		27	28	32	3,35	3,6	4,1

Поясним рассматриваемые критерии:

- «Коэффициент загрузки программы» - отношение времени загрузки программы ко всему времени сегментирования;
- «Максимально возможное число изображений для сегментации» - это максимальное количество изображений, которое можно сегментировать, то есть при большом количестве изображений, результат будет страдать;
- «Текущее число изображений для сегментации» - количество изображений для сегментации в определенный момент работы программы;
- «Время работы программы» - время, за которое изображение сегментируется, то есть интервал между поступлением изображения и его выходом;

- «Количество обработанных изображений» - то количество изображений, которое успевает обработаться за представленный промежуток времени;
- «Количество необработанных изображений» - то количество изображений, которые не успевают обработаться по времени;

### 5.1.3. Технология QuaD

Оценочная карта для сравнения конкурентных технических решений приведена в таблице 2.

Таблица 5. Оценочная карта для сравнения конкурентных технических решений

№	Критерии оценки	Вес	Баллы	Макс. балл	Отн. знач.	Ср.-взвеш. знач.
	1	2	3	4	5	
Показатели оценки качества разработки						
1.	Коэффициент загрузки программы	0,25	70	100	0,7	0,175
2.	Максимально возможное число изображений для сегментации	0,05	40	100	0,4	0,02
3.	Текущее число изображений для сегментации	0,05	40	100	0,4	0,02
4.	Время работы программы	0,15	40	100	0,4	0,06
5.	Количество обработанных изображений	0,25	90	100	0,9	0,225
6.	Количество необработанных изображений	0,15	90	100	0,9	0,135
Показатели оценки коммерческого потенциала разработки						
7.	Доступность	0,05	80	100	0,8	0,04
8.	Законченность работы	0,05	70	100	0,7	0,035
	<b>Итого:</b>					0,71

Можно видеть, что интегральный показатель конкурентоспособности данной разработки составляет 0,71, что является достаточно благоприятным для продолжения исследований.

#### 5.1.4. SWOT-анализ

Первый этап. Описание сильных и слабых сторон проекта, выявление возможностей и угроз.

Результаты первого этапа представлены в таблице 3.

Таблица 6. Результаты первого этапа SWOT-анализа

	<b>Сильные стороны</b> С1. Экономия времени; С2. Эффективность работы модели; С3. Широкие возможности по масштабированию и улучшению модели С4. Открытость разработки.	<b>Слабые стороны</b> Сл1. Незавершенность проекта; Сл2. Проигрыш во времени работы.
<b>Возможности</b> В1. Тенденция к увеличению популярности применения исследуемой модели в решении задач различного рода; В2. Неравномерность скорости работы у конкурентных спроектированных моделей.	<b>В1С1С2:</b> В результате того, что исследуемая модель весьма эффективна, на ее работу уходит совсем немного времени и она также может быть применима в абсолютно любой отрасли, то ее применение на деле может быть весьма полезным. <b>В2С3С4:</b> Так как другие спроектированные модели имеют не такие хорошие показатели при работе, то эта модель является наиболее перспективной, тем более, что ее итак хорошие данные можно сделать еще лучше.	<b>У1У2Сл1:</b> В случае прекращения исследований по данной теме, разработка не сможет раскрыть своего потенциала и улучшить отстающие параметры по сравнению с конкурентными моделями.
<b>Угрозы</b> У1. Отсутствие мотивации к дальнейшему развитию проекта; У2. Конкуренты могут улучшить модель.		

Итак, исходя из вышеописанного, проект необходимо развивать в направлении наибольшей точности, не останавливаясь на достигнутых

показателях. Но необходимо внимательно следить и по возможности применять новейшие разработки.

Интерактивная матрица проекта представлена в таблицах 4, 5.

Таблица 7.Интерактивная матрица проекта (сильные стороны)

Сильные стороны проекта					
Возможности проекта		C1	C2	C3	C4
	B1	+	+	-	0
	B2	-	-	+	+

Таблица 8.Интерактивная матрица проекта (слабые стороны)

Слабые стороны			
Угрозы проекта		Сл1	Сл2
	У1	+	-
	У2	+	-

## 5.2. Определение возможных альтернатив проведения научных исследований

Поскольку разработка находится на стадии проведения научных исследований, видится логичным применение морфологического подхода для определения возможных альтернатив проведения научных исследований.

В качестве морфологических характеристик в данной работе можно выделить блоки моделирования. Морфологическая матрица приведена в таблице 6.

Таблица 9.Морфологическая матрица для генетического алгоритма

	1	2	3
А. Количество параметров для обработки изображения	Не больше 1	В промежутке от 1 до 3	Больше 3
Б. Размер входного изображения	Маленький	Средний	Большой
В. Размещение изображения	Внешний носитель	Локальный диск	Интернет ресурс

Г. Количество изображений	Не имеет значения	Меньше 35	Больше 35
---------------------------	-------------------	-----------	-----------

В качестве перспективных направлений исследования следует, во-первых, отметить количество параметров для обработки изображения (например, A1B1B1Г2). Также работа модели во многом будет более функциональной и незапутанной при следующих допустимых вариантах (A1B2B1Г1 или A2B2B1Г1).

На данный момент реализовано решение типа A1B1B1Г2.3.

### **5.3. Планирование научно-исследовательских работ**

#### **5.3.1. Структура работ в рамках научного исследования**

Перечень этапов и работ в рамках проведения научного исследования представлен в таблице 7, в которой научный руководитель обозначен «Р», студент-дипломник – это инженер («И»).

Таблица 10. Перечень этапов, работ и распределение исполнителей

Основные этапы	№	Содержание работ	Исполнитель
Разработка технического задания	1	Составление и утверждение технического задания	Р
Выбор направления исследований	2	Подбор и изучение материалов по теме	Р, И
	3	Изучение уже существующих решений в данной области	И
	4	Выбор направления исследований	Р, И
	5	Календарное планирование работ по теме	Р, И
Теоретические и экспериментальные исследования	6	Теоретическая разработка структуры алгоритма, решающего поставленную задачу	И
	7	Построение пробного варианта алгоритма	И
	8	Оценка результатов и, при необходимости, итеративное внесение изменений в структуру алгоритма	Р, И
Обобщение и оценка результатов	9	Сравнение результатов работы конечного варианта алгоритма с другими решениями в данной области	И
	10	Оценка эффективности полученных результатов	И
	11	Оценка целесообразности проведения дальнейших исследований по данной теме	Р, И



### 5.3.2. Определение трудоемкости выполнения работ

Оценим трудоемкость выполнения вышеозначенных работ. Для этого оценим минимальное и максимальное время выполнения каждой работы (таблица 8). Также произведем расчет ожидаемого значения трудоемкости по следующей формуле:

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

Следует заметить, что исполнитель «Магистр» задействован в каждой из перечисленных работ, а потому невозможно ускорение за счет параллельности их выполнения.

### 5.3.3. Разработка графика проведения научного исследования

В таблице 8 приведены временные показатели научного исследования.

В таблице 9 приведен календарный план-график для исполнения №1, где применяется формула:

$$T_{Ki} = T_{pi} * k_{\text{кал}},$$

где  $T_{Ki}$  – продолжительность выполнения  $i$ -й работы в календарных днях;

$T_{pi}$  – продолжительность выполнения  $i$ -й работы в рабочих днях;

$k_{\text{кал}}$  – коэффициент календарности, который определяется по следующей формуле:

$$\frac{T_{\text{кал}}}{T_{\text{кал}} - (T_{\text{вых}} + T_{\text{праздн}})} = k_{\text{кал}} = 1,52,$$

где  $T_{\text{кал}}$  – количество календарных дней в году;

$T_{\text{вых}}$  – количество выходных дней в году;

$T_{\text{праздн}}$  – количество праздничных дней в году.

Таблица 11. Временные показатели проведения научного исследования

№	Содержание работ	Мин. время выполнения (дн.)			Макс. время выполнения (дн.)			Исполнители			Длительность работ в рабочих днях			Длительность работ в календарных днях		
		И1	И2	И3	И1	И2	И3	И1	И2	И3	И1	И2	И3	И1	И2	И3
1	Составление и утверждение технического задания	7	7	7	14	14	14	Р			9,8	9,8	9,8	15	15	15
2	Подбор и изучение материалов по теме	5	5	10	10	10	15	Р, И			7	7	12	11	11	19
3	Изучение уже существующих решений в данной области	15	15	15	20	20	20	И			17	17	17	26	26	26
4	Выбор направления исследований	7	7	7	10	10	10	Р, И			8,2	8,2	8,2	13	13	13
5	Календарное планирование работ по теме	7	7	7	10	10	10	Р, И			8,2	8,2	8,2	13	13	13
6	Теоретическая разработка структуры алгоритма, решающего поставленную задачу	10	10	15	20	40	25	И			14	31	19	22	48	29
7	Построение пробного варианта алгоритма	25	30	30	25	45	35	И			25	36	32	38	55	49

Продолжение таблицы 8

8	Оценка результатов и, при необходимости, итеративное внесение изменений в структуру алгоритма	30	30	30	50	50	50	Р, И	38	38	38	58	58	58
9	Сравнение результатов работы конечного варианта алгоритма с другими решениями в данной области	7	7	7	14	14	14	И	9,8	9,8	9,8	15	15	15
10	Оценка эффективности полученных результатов	7	7	7	10	10	10	Р, И	8,2	8,2	8,2	13	13	13
11	Оценка целесообразности проведения дальнейших исследований по данной теме	7	7	7	10	10	10	Р, И	8,2	8,2	8,2	13	13	13
Итого:												237	280	263

Таблица 12. Календарный план-график для Исполнения №1

№	Вид работ	Продолжительность выполнения работ																								I, % уд. вес	Н <sub>г</sub> , % на р. гот.	T <sub>k,i</sub>			
		<u>Сент</u>			<u>Окт</u>			<u>Ноябрь</u>			<u>Дек</u>			<u>Февр</u>			<u>Март</u>			<u>Апр</u>			<u>Май</u>						<u>Июнь</u>		
		1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3				1	2	3
1	Составление и утверждение технического задания																												6,4	4	15
2	Подбор и изучение материалов по теме																												18,2	23	11
3	Изучение уже существующих решений в данной области																												7,3	35	26
4	Выбор направления исследований																												1,2	24	13
5	Календарное планирование работ по теме																												2,5	33	13
6	Теор. разработка структ. алгоритма, реш-го поставл. задачу																												19,3	63	22
7	Построение пробного варианта алгоритма																												10,1	78	38

Продолжение таблицы 9

№	Вид работ	Продолжительность выполнения работ																								I, % уд. вес	Н <sub>г</sub> , % нар гот	T <sub>k,i</sub>			
		Сент			Окт			Ноябрь			Дек			Февр			Март			Апр			Май						Июнь		
		1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3						
8	Оценка результатов и, при необходимости, итеративное внесение изменений в структуру алгоритма																												21	89	58
9	Сравнение результатов работы конечного варианта алгоритма с другими решениями в данной области																												4	95	15
10	Оценка эффективности полученных результатов																												5	97	13
11	Оценка целесообразности проведения дальнейших исследований по данной теме																												5	100	13

#### 5.3.4. Бюджет научно-технического исследования

Расчет бюджета НТИ сводится к расчету материальных затрат, затрат на з/п научного руководителя и затрат на з/п дипломника. При этом материальные затраты состоят только из расходных материалов и амортизации оборудования. Обе эти статьи будут учтены при расчете накладных расходов.

Основная заработная плата ассистента ТПУ составляет 18 653 рубля. Оклад дипломника составляет 3 402 руб. Примем во внимание, что руководство над студентом, который выполняет дипломную работу, составляет некоторую долю от всей нагрузки руководителя. Для этого рассчитаем среднедневную заработную плату руководителя:

$$З_{\text{срдн.}} = \frac{18653}{20} = 932,65 \text{ руб.}$$

Если учесть, что в неделю осуществлялось две консультации, по два астрономических часа было отведено на каждую, то в итоге получается, что за месяц было потрачено 16 астрономических часов. Учитывая 8-мичасовой рабочий день, затраты на заработную плату руководителя составляют два среднедневных оклада в месяц, то есть 1 865,3 рубля.

Районный коэффициент по Томской области составляет 1,3, итого:

$$З_{\text{осн}} = 1\,865,3 * 1,3 + 3\,402 * 1,3 = 6\,847,5 \text{ руб.}$$

Тогда дополнительная заработная плата рассчитывается как:

$$З_{\text{доп}} = k_{\text{доп}} * З_{\text{осн}} = 0,13 * 6\,847,5 = 890,2 \text{ руб.}$$

Величину отчислений во внебюджетные фонды определяется как:

$$З_{\text{внеб}} = k_{\text{внеб}} (З_{\text{осн}} + З_{\text{доп}}) = 0,271 * (6\,847,5 + 890,2) = 2\,097 \text{ руб.}$$

Научных и производственных командировок в данном исследовании не производилось. Контрагентные расходы отсутствуют.

Накладные расходы рассчитаем как:

$$З_{\text{накл}} = (З_{\text{внеб}} + З_{\text{доп}} + З_{\text{осн}}) * k_{\text{нр}} = (6\,847,5 + 890,2 + 2\,097) * 0,16 = 1574 \text{ руб.}$$

$$\text{НДФЛ: } (З_{\text{осн}} + З_{\text{доп}}) * 0,13 = (6\,847,5 + 890,2) * 0,13 = 1006 \text{ руб.}$$

Бюджет затрат приведен в таблице 10.

Таблица 13. Бюджет затрат по каждому исполнению НТИ

Наименование статьи	Сумма, руб.		
	Исп1	Исп2	Исп3
1. Материальные затраты НТИ	0	0	0
2. Затраты на спец. оборудование	0	0	0
3. Затраты по основной з/п	59492	67 556	58326
4. Затраты по доп. з/п	9918	11 436	8824
5. Отчисления во внебюджетные фонды	18236	21 748	16452
6. Затраты на научные и производственные командировки	0	0	0
7. Накладные расходы	13181	16 368	12093
8. Бюджет затрат НТИ	100827	117108	95695

#### 5.4. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Интегральный финансовый показатель рассчитывается как:

$$I_{\text{финр}}^{\text{исп.i}} = \frac{\Phi_{\text{р.i}}}{\Phi_{\text{max}}}$$

Используя данные таблицы 10 получаем:

$$I_{\text{финр}}^{\text{исп1}} = 0,86$$

$$I_{\text{финр}}^{\text{исп2}} = 1$$

$$I_{\text{финр}}^{\text{исп3}} = 0,82$$

Интегральный показатель ресурсоэффективности можно определить следующим образом:

$$I_{p,i} = \sum a_i b_i$$

где  $I_{p,i}$  – интегральный показатель ресурсоэффективности для  $i$ -го варианта разработки,

$a_i$  – весовой коэффициент  $i$ -го варианта разработки,

$b_i$  – бальная оценка  $i$ -го варианта исполнения разработки, устанавливаемая экспертным путем по выбранной шкале оценивания,

$n$  – число параметров сравнения.

Расчет интегральных показателей ресурсоэффективности приведен в таблице 11.

Таблица 14. Расчет интегральных показателей ресурсоэффективности

Критерии	Весовой коэф.	Исп.1	Исп.2	Исп.3
1. Коэффициент загрузки программы	0,25	4	4	4
2. Максимально возможное число изображений для сегментации	0,05	4	4	4
3. Текущее число изображений для сегментации	0,05	3	3	3
4. Время работы программы	0,15	3	4	5
5. Количество обработанных изображений	0,25	2	4	5
6. Количество необработанных изображений	0,25	5	2	2
<b>Итого:</b>	<b>1</b>	<b>3,55</b>	<b>3,45</b>	<b>3,6</b>



Сравнительная эффективность разработок приведена в таблице 12.

Таблица 15. Сравнительная эффективность разработок

Показатели	Исп1	Исп2	Исп3
Интегральный финансовый показатель разработки $I_{финр}$	0,86	1	0,82
Интегральный показатель ресурсоэффективности разработки $I_p$	3,55	3,45	3,6
Интегральный показатель эффективности $I$	4,13	3,45	4,39
Сравнительная эффективность вариантов исполнения	1,196495	1	1,272534

Исходя из проведенного анализа, можно отметить, что Исполнение №3 является более предпочтительным в сравнении с исполнением №1 и №2, соответственно. У него более низкая стоимость и показатель ресурсоэффективности более высокий, хоть и на маленькую величину. Исходя из вышеперечисленного, следует, что Исполнение №3, которое реализовано в данной работе, является более эффективным вариантом реализации проекта.

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8KM51	Григорьева Нина Валерьевна

Институт	институт кибернетики	Кафедра	программной инженерии
Уровень образования	магистратура	Направление/специальность	09.04.03 прикладная информатика

Исходные данные к разделу «Социальная ответственность»:	
1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	<p><i>Офисное помещение, в котором есть рабочее место с персональным компьютером. В помещении предусмотрена система вентиляции воздуха. Имеется 9 рабочих персональных компьютеров с жидкокристаллическими экранами.</i></p> <p><i>Вредные факторы: освещенность рабочей зоны, производственный шум, микроклимат помещения, электромагнитное излучение;</i></p> <p><i>Опасные факторы: поражение электрическим током, пожар.</i></p> <p><i>Региональная безопасность: воздействие на атмосферу, литосферу, гидросферу.</i></p> <p><i>Возможны чрезвычайные ситуации в виде пожара либо замыкания проводки</i></p>
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
<b>1. Производственная безопасность</b>  1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения  1.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения	<p><i>1.1. Работы ведутся в помещении с нормальным уровнем освещения (люминесцентные лампы с суммарным уровнем освещенности ниже 215 люкс, что сохраняет допустимое значение в соответствии со СНиП 23-05-95). Уровень шума в пределах нормы. Помещение хорошо вентилируется.</i></p> <p><i>1.2. При выполнении работ на персональном компьютере, возможно короткое замыкание электропроводок. В связи с этим помещение оснащено ручными углекислотными огнетушителями ОУ-2,, а также аптечками первой помощи согласно требованиям ГОСТ Р 51057-01.</i></p>
<b>2. Экологическая безопасность</b>	<i>Негативное воздействие на окружающую природную среду проектируемой производственной</i>

	<i>среды: утилизация оргтехники и компьютеров.</i>
<b>3. Безопасность в чрезвычайных ситуациях</b>	<i>Разработан ряд действий в результате возникновения пожара и мер по ликвидации его последствий. В помещении имеются огнетушители ОУ-2.</i>
<b>4. Правовые и организационные вопросы обеспечения безопасности</b>	<i>Организация рабочих мест с электронно-вычислительными машинами регулируется СанПиНом 2.2.2/2.4.1340 – 03;</i>  <i>Рабочее место при выполнении работ сидя регулируется ГОСТом 12.2.032 – 78.</i>

<b>Дата выдачи задания для раздела по линейному графику</b>	
---	--

**Задание выдал консультант:**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
доцент каф. ЭБЖ	М.И. Пустовойтова	к.х.н.		

**Задание принял к исполнению студент:**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8KM51	Григорьева Нина Валерьевна		

## **6. Социальная ответственность**

В любой проектной либо научно-исследовательской деятельности важную роль занимает безопасность труда и окружающей среды.

В понятие «социальная ответственность» включено следующее: состояние рабочего места, помещения, режим трудовой деятельности и обеспечение мероприятий по защите трудящихся в моменты чрезвычайных ситуаций. регламентируются в соответствии с международным стандартом ICCSR26000:2011 «Социальная ответственность организации» [14]. Целью данного стандарта является принятие проектных решений, исключающих несчастные случаи на производстве и снижение негативных воздействий на окружающую среду.

Согласно данному стандарту такое понятие, как «социальная ответственность», означает ответственность организации за воздействие решений, которые были ею предложены, на общество и окружающую среду.

Раздел включает в себя следующие составляющие: техногенная безопасность, региональная безопасность, организационные мероприятия обеспечения безопасности, особенности законодательного регулирования проектных решений и безопасность в чрезвычайных ситуациях.

Научно-исследовательская работа представляет собой создание алгоритмического и программного обеспечение локализации области лёгкого по данным компьютерной томографии и предполагает большой объем работы с ПК, поэтому важным критерием безопасности является организация рабочего места и режима трудовой деятельности.

## **6.1. Техногенная безопасность**

### **6.1.1. Характеристика объекта исследования**

Выпускная квалификационная работа студента выполнялась в кибернетическом центре ТПУ (четвертый этаж) на кафедре программной инженерии. Параметры учебной аудитории: длина – 6,23 м, ширина – 7,36 м, высота – 3,4 м. Для естественного освещения аудитория оснащена двумя окнами, имеющими одинаковые размеры: ширина – 1,05 м, высота – 2,31 м. Дверь в помещение изготовлена из металла и имеет серый цвет. На подвесном потолке установлена прессованная плитка из полистирола. На стенах поклеены обои белого цвета. Напольным покрытием является линолеум. Площадь аудитории равна 45,85 м<sup>2</sup>, объём – 155,9 м<sup>3</sup>.

Согласно СанПиН 2.2.2/2.4.1340-03 [15], в котором указаны требования к помещениям для работы с ПЭВМ, норма площади рабочего места с персональным компьютером составляет 4,5 м<sup>2</sup>. В рассматриваемой аудитории установлено девять рабочих мест с персональными компьютерами и жидкокристаллическими экранами. Соответственно, на одного человека приходится 5 м<sup>2</sup>, что соответствует вышеуказанным требованиям.

## **6.2. Производственная безопасность**

### **6.2.1. Вредные факторы производственной среды**

#### **6.2.1.1. Освещенность рабочей зоны**

Рабочее (общее) освещение – это основное освещение, обеспечивающее нормальные условия для нахождения человека в помещении [16]. Такое освещение необходимо предусматривать для всех складских, производственных, санитарно-бытовых, жилых и административных помещений.

Освещение в недостаточной степени может привести к напряжению зрения, ослаблению внимания и наступлению преждевременной

утомленности. Слепление, резь в глазах и раздражение могут быть вызваны чрезмерно ярким освещением. Свет на месте труда может создать сильные тени или отблески, а также дезориентировать работающего. Перечисленные причины могут привести к профзаболеваниям.

Труд работника-программиста связан с постоянным использованием персонального компьютера, это значит, что освещение при организации такого рабочего места играет немаловажную роль. Необходимо продумать систему искусственного освещения, которая будет повышать комфортность условий труда.

Одним из основных показателей качества освещения является освещенность  $E$  – поверхностная плотность светового потока. Данный показатель регулируется в соответствии со СНиП 23-05-95 [16]. По характеристике зрительной работы труд программиста относится к 3 разряду зрительной работы, т.е. наименьший размер объекта различения от 0,3 до 0,5 мм (точка). Это значит, что нормативное значение освещенности рабочего места должно быть 200 лк (СНиП 23-05-95) [9].

Рассчитаем фактическую освещенность рассматриваемой учебной аудитории. Длина и ширина аудитории равны соответственно 6,23 и 7,36 м, высота – 3,4 м. Рассчитаем индекс помещения:

$$i = \frac{S}{h \cdot (A+B)}, \quad (1)$$

где  $i$  – индекс помещения;

$S$  – площадь помещения, м<sup>2</sup>;

$h$  – высота помещения, м;

$A$  – длина помещения, м;

$B$  – ширина помещения.

$$i = \frac{45,85}{3,4 * (6 + 7)} = 1$$

Исходя из значения индекса помещения можно определить, что коэффициент использования рассматриваемого светового светильника с люминесцентными лампами равен 43% [17]. Рассчитаем освещенность по формуле, учитывая, что в аудитории 12 светильников по 4 лампы в каждом:

$$E_{\text{факт}} = \frac{N * n * \Phi_{\text{ст}} * \eta}{S * K_3 * Z}, \quad (2)$$

где  $E_{\text{н}}$  – фактическая освещенность;

$N$  – число светильников в помещении;

$n$  – число ламп в светильнике;

$\Phi_{\text{ст}}$  – величина стандартного светового потока, лм;

$\eta$  – коэффициент использования светового потока;

$S$  – площадь помещения;

$K_3$  – коэффициент запаса;

$Z$  – коэффициент неравномерности освещения.

Зная, что  $\Phi_{\text{ст}} = 720$  лм для люминесцентных встроенных ламп для освещения общественных и жилых помещений (ЛВО) [18],  $K_3$  для помещений с малым выделением пыли равен 1,5, а  $Z$  для люминесцентных ламп равен 1 рассчитаем значение фактической освещенности.

$$E_{\text{факт}} = \frac{12 * 4 * 720 * 0,43}{45,85 * 1,5 * 1} = \frac{14860,8}{69} = 215,4 \text{ лк}$$

Данное значение близко к нормативному. Рассчитаем численную оценку разности между фактическим значением освещенности и нормативным.

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

(3)

где  $\Delta E$  – показатель разности между фактической освещенностью и нормативной;

$E_{\text{факт}}$  – фактическое значение освещенности;

$E_{\text{н}}$  – нормативное значение освещенности.

$$\Delta E = \frac{(215,4 - 200)}{200} * 100\% = 7,7\%$$

Зная, что допустимое отклонение освещенности может быть  $\pm 20\%$ , можно сделать вывод, что в аудитории подходящая система освещения.

#### 6.2.1.2. Производственный шум

Люди, которым приходится работать в условиях длительного шума, обычно имеют головные боли, раздражительность, сталкиваются со снижением памяти, повышенной утомляемостью, также у многих понижен аппетит, есть боли в ушах и т. д. Перечисленные факты снижают производительность, работоспособность человека, а также качество труда [19].

Шумовой фон помещения создают десять одновременно работающих компьютеров. Иногда возникает шум, который исходит от принтера или телефонных аппаратов. Также источником шума является система вентиляции или шумы поступающий извне помещения.



Во избежание негативных последствий от производственного шума, его необходимо регулировать в соответствие с нормами, которые указаны в ГОСТ 12.1.003-83 «ССБТ. Общие требования безопасности» [20].

Допустимые уровни звука и звукового давления для рабочего места разработчика-программиста согласно вышеуказанному ГОСТу 12.1.003-83 [13] представлены в таблице 13.

Таблица 16. Предельно допустимые уровни звука

Вид трудовой деятельности/ Частоты	Уровни звука и звукового давления, дБ, в октавных полосах со среднегеометрическими частотами, Гц								
	31,5	63	125	250	500	1000	2000	4000	8000
Научная деятельность, проектирование, программирование, Рабочие места проектно-конструкторских бюро, программистов вычислительных машин и т.д.	86	71	61	54	49	45	42	40	38

Допустимый уровень звукового давления колеблется от 38 дБ до 86 дБ при частоте от 8000 Гц до 31,5 Гц, соответственно.

Для уменьшения воздействий шума можно использовать следующие методы, согласно СНиП 23-03-2003 [21]:

применение звукоизоляции;

экранирование рабочих мест, то есть установка перегородок между рабочими местами;

установка оборудования, производящего минимальный шум.

### 6.2.1.3. Микроклимат помещения

Компьютеры могут привести к увеличению температуры и снижению относительной влажности в помещении. Поэтому необходимо соблюдать параметры микроклимата. В СанПиН 2.2.4.548 – 96 установлены величины параметров микроклимата, создающие комфортные условия [22].

Требования к микроклимату зависят от категории тяжести работ. Работа программиста относится к легкой категории, то есть к категории 1б (СанПиН 2.2.4.548 – 96) [22]. Для каждой категории тяжести работ есть значения показателей микроклимата.

В таблицах представлены данные показатели для теплого периода года (плюс 10 °С и выше) и для холодного периода года.

Таблица 17. Оптимальные величины показателей микроклимата  
(СанПиН 2.2.4.548 – 96) [22]

Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	21-23	20-24	40-60	0,1
Теплый	22-24	21-25		0,1

Таблица 18. Допустимые величины показателей микроклимата (СанПиН 2.2.4.548 – 96) [15]

Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	19-24	18-25	15-75	0,1-0,2
Теплый	20-28	19-29		0,1-0,3

Если температура воздуха отличается от нормальной, то время пребывания в таком помещении должно быть ограничено в зависимости от

категории тяжести работ. Температура в рассматриваемом помещении в холодное время года может опускаться до 15-17 °С, а в теплое время года подниматься до 29-30 °С. Соответствующие нормы представлены в таблицах.

Таблица 19. Рекомендуемое время работы при температуре воздуха ниже допустимых величин (СанПиН 2.2.4.548 – 96) [22]

Температура воздуха, °С	Время пребывания, не более, ч
17	6
18	7

Таблица 20. Рекомендуемое время работы при температуре воздуха выше допустимых величин (СанПиН 2.2.4.548 – 96) [22]

Температура воздуха, °С	Время пребывания, не более, ч
30,0	5
29,5	5,5
29,0	6

К мероприятиям по оздоровлению воздушной среды в производственном помещении относятся правильная организация вентиляции и кондиционирования воздуха, отопление помещений. В рассматриваемой аудитории вентиляция осуществляется естественным и механическим путём. В зимнее время в помещении предусматривается система отопления и установлен кондиционер класса «зима-лето» Carrier 42ADF. Это обеспечивает нормальное состояние здоровья работников в аудитории.

#### 6.2.1.4. Электромагнитное излучение

Электромагнитное излучение - распространяющееся в пространстве возмущение электрических и магнитных полей [23]. Источниками электромагнитного излучения в данном исследовании являются мониторы и системный блок.

На человека, работающего за компьютером, действует электромагнитное излучение. Опасность этого излучения заключается в том, что его влияние негативно сказывается на здоровье работника, а именно: на нервной, эндокринной и сердечно-сосудистой системах.

Оценка величины уровней ЭМП, проведенная по документам компьютера и монитора, выявила их соответствие нормам ТСО–03 и СанПиН 2.2.2/2.4.1340–03 [24]. В таблице 18 приведены нормы уровня ЭМП, которым соответствует техника в кабинете.

Таблица 21. Допустимые уровни ЭМП

Параметр		ВДУ ЭМП
Напряженность электрического поля	5 Гц - 2 кГц	25 В/м
	2 кГц - 400 кГц	2,5 В/м
Плотность магнитного потока	5 Гц - 2 кГц	250 нТл
	2 кГц - 400 кГц	25 нТл
Электростатический потенциал экрана видеомонитора		500 В

Для того, чтобы снизить воздействие таких видов излучения, обычно рекомендуют применять такие мониторы, у которых уровень излучения понижен (MPR-II, ТСО-92, ТСО-99), а также установить защитные экраны и соблюдать режимы труда и отдыха.

## **6.2.2. Опасные факторы производственной среды**

### **6.2.2.1. Поражение электрическим током**

К опасным факторам относят поражение электрическим током согласно ГОСТ 12.0.003-74 [25]. Рабочее место выполняющего работу включает: монитор, клавиатуру, мышь и системный блок, токи статического электричества на корпусах мониторов, системного блока, мыши и клавиатуры могут привести к возникновению разрядов при прикосновении к данным элементам. Такие разряды обычно не представляют опасности для человека, но при возникновении заряда с

большим электрическим потенциалом порождается электрическое поле с повышенной напряженностью, которое может нанести урон человеку.

При продолжительном пребывании человека в таком поле наблюдаются изменения в центральной нервной, сердечно-сосудистой системах. Избыточный электрический заряд способствует выходу компьютера из строя. Для уменьшения величин токов статического электричества используют нейтрализаторы, а также увлажнители воздуха.

#### 6.2.2.2. Пожар

Также к опасным факторам относится и пожарная безопасность (ГОСТ 12.0.003-74 [18]). Пожарная безопасность осуществляется системой пожарной защиты и системой предотвращения пожара.

По взрыво- и пожароопасности все помещения, согласно техническому регламенту НПБ 105-95 [26], делятся на 5 категорий, в зависимости от применяемых на производстве веществ и их количества. Рассматриваемая учебная аудитория относится к пожароопасной категории В, в качестве горючего материала выступает бумага [27].

Основные причины возникновения пожаров:

1. Нарушение правил пожарной безопасности;
2. Перегрузка электросети;
3. Неисправность прибора;
4. Разряд молнии и неисправность молниеотвода.

### 6.3. Экологическая безопасность

Воздействие на литосферу предусматривает под собой утилизацию электронной техники: компьютеров, сканеров и т.п. Утилизация такого оборудования является достаточно сложной, так как они имеют сложную структуру. Непосредственная переработка большей части компонентов включает в себя их сортировку, последующую гомогенизацию и отправку

для повторного использования, т.е. с предварительным помолом или переплавкой.

При рассмотрении влияния персонального компьютера на атмосферу, гидросферу и литосферу выявлены особо вредные выбросы согласно ГОСТ Р 51768-2001 [27]. В случае выхода из строя компьютеров, они списываются и отправляются на специальный склад, который при необходимости принимает меры по утилизации списанной техники и комплектующих. В настоящее время в Томской области утилизацией занимаются две компании: городской полигон и ООО НПП «Экотом». Утилизацией опасных бытовых отходов занимаются компании: ООО «Торем», ООО «СибМеталлГрупп».

#### **6.4. Безопасность в чрезвычайных ситуациях**

Одними из наиболее вероятных видов чрезвычайных ситуаций являются пожар, а также взрыв на рабочем месте.

Всякий работник при обнаружении пожара должен (ППБ 01-03 [28]):

незамедлительно сообщить об это в пожарную охрану;

принять меры по эвакуации людей, каких-либо материальных ценностей согласно плану эвакуации;

отключить электроэнергию, приступить к тушению пожара первичными средствами пожаротушения.

Учебные аудитории кибернетического центра ТПУ оснащены ручными углекислотными огнетушителями ОУ-2 по одному на аудиторию, а также аптечками первой помощи согласно требованиям ГОСТ Р 51057-01 [29].

При возникновении пожара должна сработать система пожаротушения, передав на пункт пожарной станции сигнал о ЧС. В случае если система не сработала, то необходимо самостоятельно произвести вызов пожарной службы по телефону 101, сообщить место возникновения ЧС и ожидать приезда специалистов.

Рабочее место располагается в кибернетическом центре ТПУ, 212 аудитория. На рисунке 11 представлен план эвакуации второго этажа кибернетического центра.

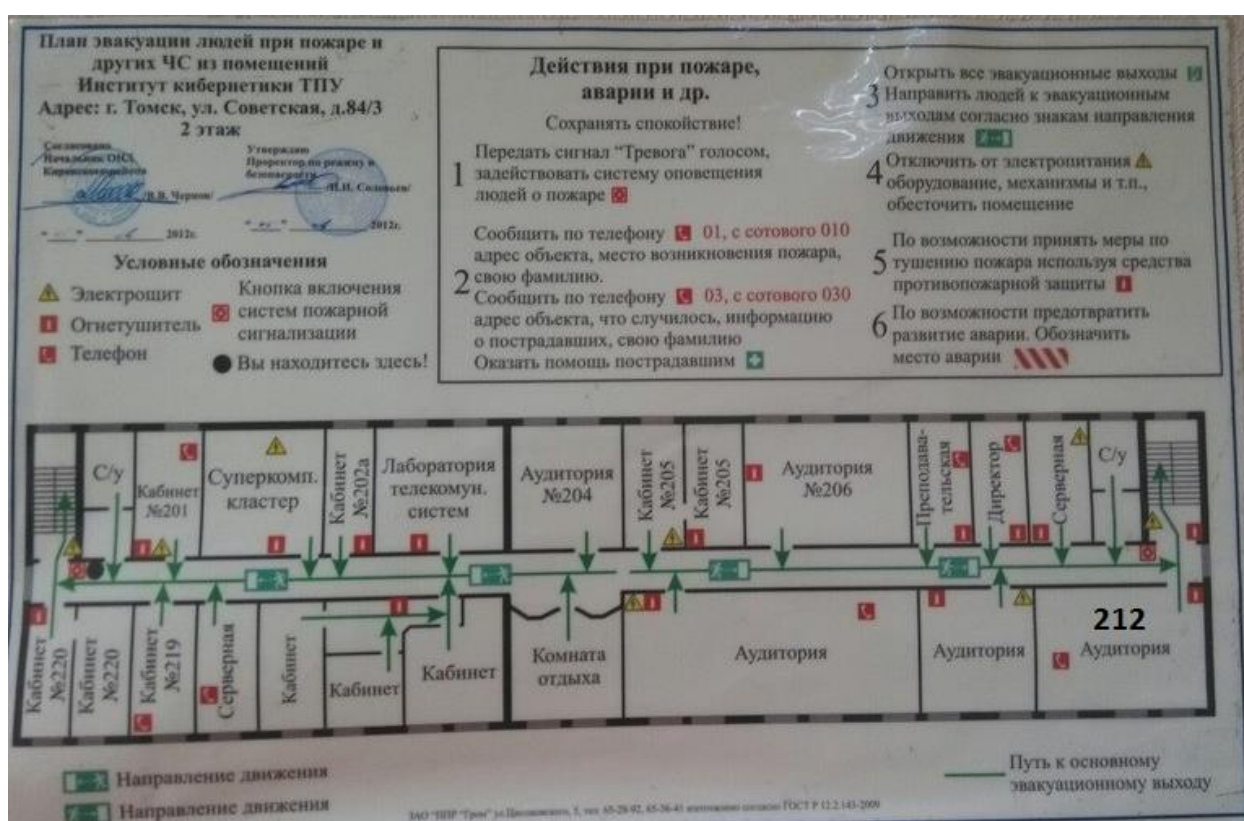


Рис. 35. План эвакуации при пожаре и других чрезвычайных ситуациях

## **Заключение**

В ходе выполнения выпускной квалификационной работы был разработан алгоритм и программное обеспечение для линейной аппроксимации контуров сегментированных изображений, которые позволяют повысить качество диагностики состояния лёгкого по данным компьютерной томографии.

Алгоритм был реализован на языке программирования C# и протестирован на большом количестве снимков различного качества. Программа позволяет с достаточно высокой точностью выделять, как четкие, так и размытые контуры. Система была протестирована на изображениях компьютерной томографии пяти реальных пациентов. Тестирование показало, что результат работы алгоритма линейной аппроксимации контуров сегментированного изображения не зависит от положения тела пациента и зашумлённого фона.

При разработке приложения по локализации области лёгкого выделяется не только костная ткань, но и восстанавливается граница лёгкого в норме. Это повышает эффективность диагностики, и не позволяет упустить участки патологий, а также является преимуществом нашей разработки перед другими системами.

Полученные результаты исследований могут использоваться в широком спектре задач. Одним из наиболее перспективных направлений применения результатов исследований является разработка системы классификации и диагностирования форм степени заболеваний лёгких по данным компьютерной томографии. Данный процесс будет осуществляться при помощи построения трёхмерной модели. Система применима к использованию специалистами по лучевой диагностике для описания состояния лёгкого по данным КТ.



### Список публикаций

1. A. A. Alekseev, V. V. Osipova, M. A. Ivanov, A. Klimentov, N. V. Grigorieva, H. S. Nalamwar. Efficient data management tools for the heterogeneous big data warehouse [Electronic resource] // Physics of Particles and Nuclei Letters : Scientific Journal. September 2016. Vol. 13, iss. 5. P. 689–692. Режим доступа: <http://dx.doi.org/10.1134/S1547477116050022>
2. Григорьева Н.В. Анализ основных существующих методов сегментации изображений компьютерной томографии [Электронный ресурс] // Инновационные научные исследования: сборник статей VIII международной научно-практической конференции, Пенза, 20 марта 2017 года. – МЦНС «Наука и просвещение», 2017 – С. 28-31. Режим доступа: <https://elibrary.ru/item.asp?id=29138467>
3. Григорьева Н.В.. Исследование доступного программного обеспечения для проведения сегментации изображения лёгких по данным КТ. [Электронный ресурс] // журнал «Вестник современных исследований», Омск, 12 мая 2017 года.–Соловьёв Вадим Анатольевич, 2017– С. 135-138. Режим доступа: <http://elibrary.ru/contents.asp?titleid=63748>

### **Список использованных источников**

1. Магонов Е.П., Трофимова Т.Н. Автоматическая сегментация МРТ-изображений головного мозга: методы и программное обеспечение // ИМЧ РАН. Лекции и обзоры. 2012. Т. 3. № 3. С. 35–40.
2. Марусина М.Я. Инвариантный анализ и синтез в моделях с симметриями. СПб.: СПбГУ ИТМО, 2004. 144 с.
3. Acharya T., Ray A.K. Image processing. Principles and applications. 2005. 428 p.
4. Dougherty G. Digital image processing for medical applications. Cambridge University Press, 2009. 447 p.
5. Марусина М.Я. Повышение качества измерений на основе теоретико-группового анализа и синтеза измерительных систем // Дисс. на соискание ученой степени доктора техн. наук. – СПб: ИТМО, 2005. – 340с.
6. Сирота А.А., Соломатин А.И. Статистические алгоритмы обнаружения границ объектов на изображениях. Вестник ВГУ. Сер. Системный анализ и информационные технологии, 2008, № 1, с. 58–64.
7. Heikkila M., Pietikainen M. A texture-based method for modeling the background and detecting moving objects. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, vol. 28, no. 4, pp. 657–662. doi: 10.1109/TPAMI.2006.68.

8. Чудовская А.К. Возможности распараллеливания алгоритмов выделения контура по технологии CUDA. Сб. докл. IV Межд. науч.-практич. конф. «Современная информационная Украина: информатика, экономика, философия». Донецк, 2010, с. 67–70.
9. Прэтт У. Цифровая обработка изображений: Пер. с англ. — М.: Мир, 1982. — Кн. 2 — 480 с.
10. Роженцов, А.А. Формирование 3D изображений при проведении трансуретральной резекции предстательной железы / А.А. Роженцов, В.Н. Дубровин, А.А. Баев, А.С. Наумов // Вестник Поволжского государственного технологического университета. – 2008. – № 3. – С. 45-50. – ISSN 2306-2819.
11. Выделение и описание контуров URL: [http://wiki.technicalvision.ru/index.php/Выделение\\_и\\_описание\\_контуров](http://wiki.technicalvision.ru/index.php/Выделение_и_описание_контуров) [Дата обращения 18.03.2017]
12. Гонсалес Р., Вудс Р. Цифровая обработка изображений, .Издание 3-е, испр. и доп. /При финансовой поддержке ФА по печати и массовым коммуникациям в рамках Федеральной целевой программы «Культура России», 2012. -32 с.
13. Dougherty G. Digital image processing for medical applications. Cambridge University Press, 2009. 447 p.
14. Международный стандарт ICCSR26000:2011 «Социальная ответственность организации».

15. СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы. – М.: Информационно-издательский центр Минздрава России, 2003. – 54 с.
16. СНиП 23-05-95. Естественное и искусственное освещение. – М.: Центр проектной продукции в строительстве, 2011. – 70 с.
17. ГОСТ 6825-91. Лампы люминесцентные трубчатые для общего освещения. – М.: Издательство стандартов, 1992. – 242 с.
18. Борьба с шумом на производстве: Справочник / Е.Я. Юдин, Л.А. Борисов; Под общ. ред. Е.Я. Юдина – М.: Машиностроение, 1985. – 400с., ил.
19. ГОСТ 12.1.003-83. ССБТ. Общие требования безопасности. – М.: Издательство стандартов, 2002. – 13 с.
20. СНиП 23-03-2003. Защита от шума. – М.: Госстрой России, 2004. – 34 с.
21. СанПиН 2.2.4.548 – 96. Гигиенические требования к микроклимату производственных помещений. – М.: Информационно-издательский центр Минздрава России, 1997. – 20 с.
22. Безопасность жизнедеятельности. /Под ред. Н.А. Белова - М.: Знание, 2000 - 364с.
23. СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы. – М.: Информационно-издательский центр Минздрава России, 2003. – 54 с.

24. ГОСТ 12.0.003-74 ССБТ. Опасные и вредные производственные факторы. Классификация. - М.: Издательство стандартов, 2001. – 4 с.
25. НПБ 105-95. Определение категорий помещений и зданий по взрывопожарной и пожарной опасности. / Шебеко Ю.Н. – М.: ВНИИПО, 1998. – 119 с.
26. СП 12.13130.2009. Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности. – М.: Проспект, 2010. – 32 с.
27. ГОСТ Р 51768-2001. Ресурсосбережение. Обращение с отходами. Методика определения ртути в ртутьсодержащих отходах. Общие требования. – М: Издательство стандартов, 2001. - 13 с.
28. ППБ 01-03. Правила пожарной безопасности в Российской Федерации. – М.: ФГУ ВНИИПО МЧС России, 2003. – 111 с.
29. ГОСТ Р 51057-01. Огнетушители переносные. Общие технические требования. Методы испытаний. – М.: Издательство стандартов, 2001. – 48 с.
30. Выделение и описание контуров URL:  
[http://wiki.technicalvision.ru/index.php/Выделение\\_и\\_описание\\_контуров](http://wiki.technicalvision.ru/index.php/Выделение_и_описание_контуров)  
[Дата обращения 18.03.2017]
31. Сегментация  
URL: [https://ru.wikipedia.org/wiki/Сегментация\\_\(обработка\\_изображений\)](https://ru.wikipedia.org/wiki/Сегментация_(обработка_изображений))  
[Дата обращения 18.03.2017]

## Приложение А

### Раздел 7 Literature review

Студент:

Группа	ФИО	Подпись	Дата
8KM51	Н.В. Григорьева		

Консультант кафедры ПИ

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент	Е.С. Чердынцев	К.Т.Н.		

Консультант – лингвист кафедры ИЯИК

Должность	ФИО	Ученая степень, звание	Подпись	Дата
зав. каф. ИЯ ИК	Т.В. Сидоренко	К.П.Н.		

## **7. Segmentation algorithm for CT images**

Medical images have become essential in medical diagnosis and treatment. These images play a substantial role in medical applications because doctors exhibit interest in exploring the internal anatomy [1]. Many techniques have been developed based on X-ray and cross-sectional images like Computed Tomography (CT) or Magnetic Resonance Imaging (MRI), or other tomographic modalities.

Over the years, medical image processing has contributed a lot in medical applications; for example, the use of image segmentation, image registration, and image guided surgery is so common in medical surgery. CT is medical imaging technique which uses X-ray in imaging internal body organs and structure. It produces a number of parallel slices of each organ by passing X-ray pulses through the body.

The most important part of medical image processing is image segmentation. Image segmentation is a procedure for extracting the region of interest (ROI) through an automatic or semi-automatic process. Many image segmentation methods have been used in medical applications to segment tissues and body organs. Some of the applications consist of border detection in angiograms of coronary, surgical planning, simulation of surgeries, tumor detection and segmentation, brain development study, functional mapping, blood cells automated classification, mass detection in mammograms, image registration, heart segmentation and analysis of cardiac images, etc.

In medical research, segmentation can be used in separating different tissues from each other, through extracting and classifying features. One of the efforts is classifying image pixels into anatomical regions which may be useful in extracting bones, muscles, and blood vessels. For example, the aim of some brain research works is to partition the image into different region colors such as white and different grey spectra which can be useful in identifying the cerebrospinal fluid in brain

images, white matter, and grey matter. This process can also prove useful in extracting the specific structure of breast tumors from MRI images [20].

This chapter presents the latest region-based methods. The advantages and disadvantages of this method are described besides examination of algorithm with its application in Magnetic Resonance Imaging and Computed Tomography image analysis. Algorithm is explained separately with its ability and features for the analysis of grey-level images.

## **7.1. Region-based methods**

A region is composed of some pixels which two by two are neighbors and the boundary is made from differences between two regions. Most of the image segmentation methods are based on region and boundary properties. Here we explain two most popular region based approaches:

-thresholding:

-region growing.

### **7.1.1. Thresholding**

Thresholding is one of the simplest and fastest segmentation methods based on the assumption that images are formed from regions with different grey levels. The histogram of images has different peaks and valleys which can divide images into different parts [21]. Threshold is a value in a histogram that divides intensities into two parts: the first part is the “foreground” having pixels with intensities greater than or equal to the threshold and the second part is the “background” having pixels with intensities less than the threshold. Therefore,

$$g(x, y) = \begin{cases} \text{foreground} & \text{if } f(x, y) \geq T \\ \text{background} & \text{if } f(x, y) < T \end{cases} \quad (1)$$



where  $f(x, y)$  is the pixel intensity in the  $(x, y)$  position and  $T$  is the threshold value. An inappropriate threshold value leads to poor segmentation results. To separate more than one object with different grey levels, more than one threshold is used which is called multi thresholding. Figure 2 shows the application of thresholding in CT images. In this case, thresholding is applied in the CT image of legs to extract the bone area from the background.

Thresholding segmentation usually does not take into account the spatial information of images which leads to sensitivity to noise and intensity in homogeneities.

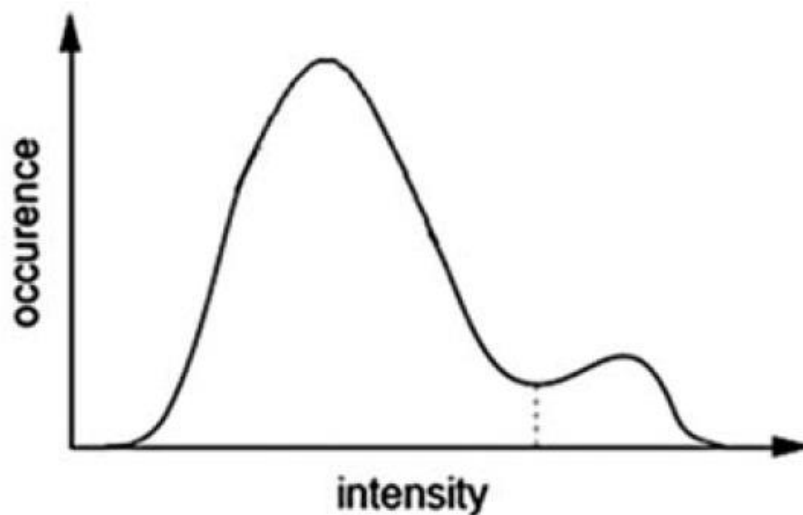


Figure 1. Histogram function with a valley which is useful

These problems may occur in MRI images which fundamentally destroy the histogram and make partitioning more complex [31].

Global thresholding works on the idea that an image has a bimodal histogram and the object can be separated from the background using a threshold value. In the following, local (adaptive) thresholding that uses a local threshold value and Otsu's thresholding that uses an automatic threshold value are described.

## Local Thresholding

Global thresholding does not provide satisfactory results for some type of images such as images which do not have a constant background and have diversity across the object. For this kind of images, thresholding provides a good result in one region but fails in other parts of images [25]. In order to find different threshold values for different parts of images, the local thresholding method divides images into subimages and then calculates the threshold value for each part. The results of thresholding for each part of an image are then merged. In this method, an image is divided into vertical and horizontal lines, whereas each part includes a region of both the background and the object. Finally, an interpolation is needed to produce appropriate results. Figure 2 shows the application of thresholding in CT images.

Different statistical methods are used to select the threshold value for each subimage, for example, mean, standard deviation, mean and standard deviation together, and mean of maximum and minimum.

Local thresholding needs more time to segment an image compared to global thresholding. This method is more useful in the case of images with varying backgrounds.

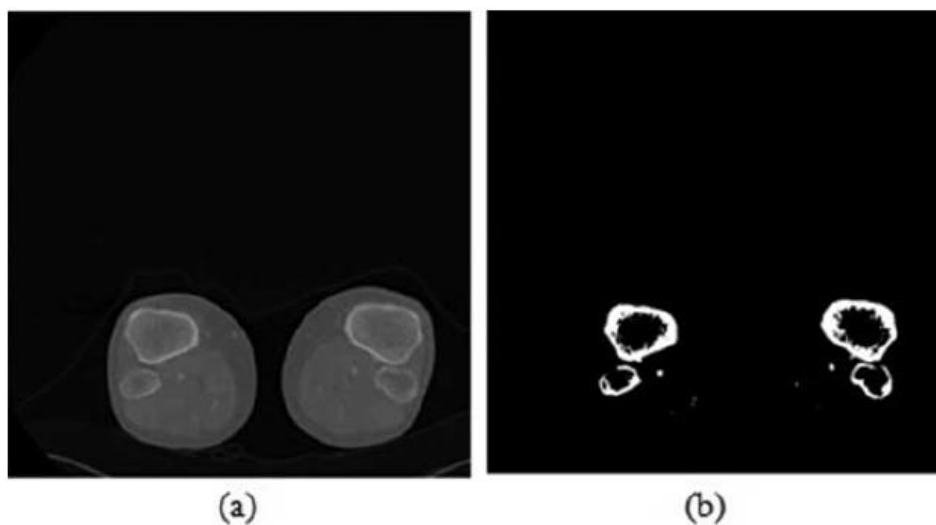


Figure 2. Applying thresholding on the CT image. (a) Original CT images. (b) Extracting the bone by using thresholding

## Otsu's Thresholding

Threshold value is usually selected visually which leads to problems and may even lead to poor results. To automate the selection of threshold value, some methods have been presented such as Otsu's method. The goal of Otsu's thresholding is to find an optimal value for global thresholding. In this method, it is assumed that an image has two pixel classes or has a biomodal histogram [36,37]. It chooses the threshold to minimize the intra-class variance (the variance within the class) of black and white cluster pixels.

The intra-class variance could be define using weighted equation of variances of each cluster:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t), \quad (2)$$

where the weights  $q_i$  are the probability for each class and estimated as

$$q_1(t) = \sum_{i=1}^t P(i),$$
$$q_2(t) = \sum_{i=t+1}^I P(i), \quad (3)$$

and the class means are given by

$$\mu_1(t) = \sum_{i=1}^t \frac{iP(i)}{q_1(t)},$$
$$\mu_2(t) = \sum_{i=t+1}^I \frac{iP(i)}{q_2(t)}. \quad (4)$$

Finally, the individual class variances are given by

$$\sigma_1^2(t) = \sum_{i=1}^t [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)},$$

$$\sigma_2^2(t) = \sum_{i=t+1}^I [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}.$$
(5)

The process can be stopped here and the algorithm continued by applying the within-class variance for all threshold values  $t$  and selecting the value that minimizes  $\sigma_w^2(t)$ . To achieve faster calculation, we can use the relation between the within-class and between class variances. Otsu explains that minimizing the intra-class variance is similar to maximizing the between-class variance. After some calculation we have

$$\sigma_b^2(t) = \sigma - \sigma_w^2(t) = q_1(t)q_2(t)[\mu_1(t) - \mu_2(t)]^2.$$
(6)

The threshold value that maximizes the between-class variance is the optimal threshold value. Figure 3 shows the application of Otsu's thresholding in cell images.

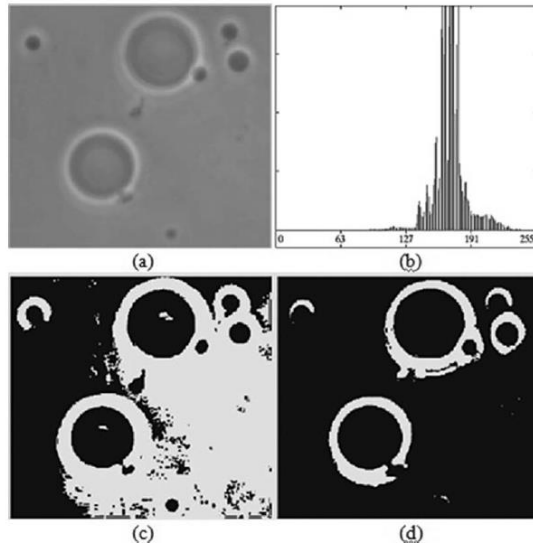


Figure 3. Otsu's thresholding. (a) Original image. (b) Histogram of an image. (c) Global thresholding. (d) Otsu's

### 7.1.2. Region Growing

Region growing is an interactive segmentation method which requires some seed points to be initialized and start the process. This technique separates a region of images based on some predefined law according to intensity information. In the simplest form, region growing requires one seed point and the region will be grown based on its homogeneity properties according to neighboring pixels [41]. There are some region-based methods which have differences in homogeneity criterion definition. A general region algorithm for extracting one object is as given below:

**Algorithm:** Input (seed point)

(1) Region  $r \leftarrow \{\text{seed}\}$

(2) While  $r.\text{neighbours} \neq \{\}$

(a) For each voxel  $x$  in  $r.\text{neighbours}$ , if  $P(x,r) \geq \text{true}$ , then add  $x$  to  $r$

(b ) End while

(3) Return  $r$

In the above algorithm,  $r$  is a region that we want to extract. Based on homogeneity criteria, some region growing-based methods have been presented. In this algorithm, the fundamental region-growing method has been explained by evaluating the distance between voxel  $x$  and the mean of region which is presented by the function  $P$  [42].  $P$  is expressed as

$$P(x, r) = |f(x) - \mu_r| < T, \quad (7)$$

Where  $\mu_r$  is the region's mean of  $r$  and  $T$  is a threshold. The threshold can be selected manually or using an automated method.

In Figure 4, region growing has been applied on the MRI image of the knee. In this figure, two seed points

have been selected from two different regions and the region-growing process has been applied.

The disadvantage of region growing is that the result of this technique significantly depends on the seed point selection. Selecting a seed point depends on human ability; thus, the extracted shape considerably depends on the user. Although noise sensitivity in this method is less than thresholding, but it can make a hole in the extracted shape or produce a disconnected area. Region growing has been widely used in mammograms in order to extract the potential lesion from its background [49].

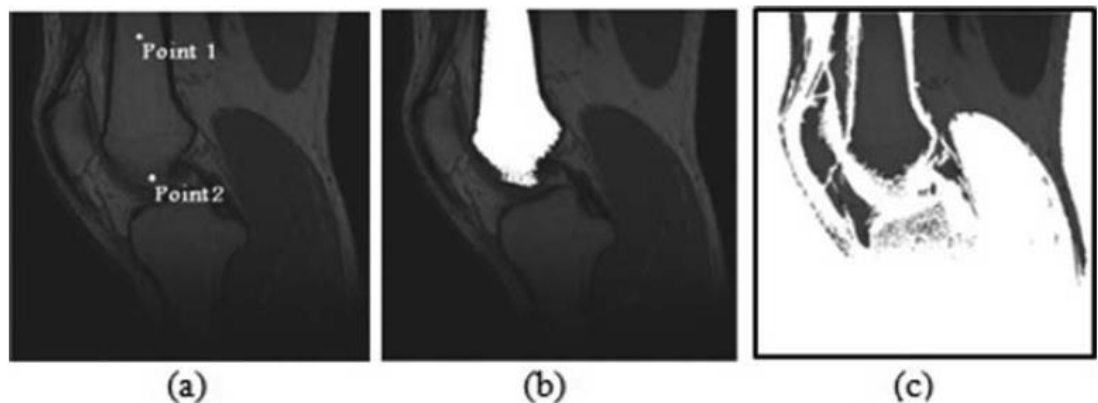


Figure 4. Performing region growing on an MRI bone image.

(a) Original images with two seed points. (b) Region growing (c) Region growing result for seed point 2.

Thresholding (global, local, Otsu) and region growing upon which region-based approaches are based are also explained. This category is very sensitive to noise, but they are simple to implement. They do not have good results in MRI without preprocessing but are effective for CT images which have less noise.

## Приложение Б

Исходный код класса Form1.cs

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Drawing.Imaging;
using System.Runtime.InteropServices;
using System.Windows.Forms;

namespace BodySegmentationApp
{
    public partial class MainForm : Form
    {
        //импорт библиотеки

        [DllImport("CTImageSegmentation.dll", CallingConvention =
CallingConvention.Cdecl)]
        unsafe public extern static int CtBodySegmentation(byte* intencity_input, int*
regions_output,
            int images_number, int image_height, int image_width, int filter_width = 9,
int intencity_threshold = 60);

        private string[] filenames;
        private static byte[][][] ctImages;
        private static int[][][] ctRegions;
        private static int imagesNumber;
        private static int imageHeight;
        private static int imageWidth;
        private static int filterWidth = 3;
        private static int intensityThreshold = 95;
        private static int bodyRegionIndex;

        private static int defaultImageShowIndex = 0;

        ColorFactory cFactory;
        //конструктор, срабатывающий при создании главной формы

        public MainForm()
        {
            InitializeComponent();
            cFactory = new ColorFactory();
        }
    }
}
```

```

private void button_browse_Click(object sender, EventArgs e)
{
//метод класса MainForm, который является событием клика кнопки name
"button_browse"
    filenames = null;
    ctImages = null;
    ctRegions = null;
    imagesNumber = 0;
    imageHeight = 0;
    imageWidth = 0;
    bodyRegionIndex = -1;

    pictureBox_original.Image = null;
    pictureBox_segmented.Image = null;
    trackBar.Maximum = 0;
    trackBar.Value = 0;
    toolStripStatusLabel.Text = "Loading and analysis of images";

    openFileDialog.Filter = "Bitmap files (*.bmp)|*.bmp|(*.BMP)|*.BMP";

    if (openFileDialog.ShowDialog() == DialogResult.OK)
    {
        if (openFileDialog.FileNames.Length > 0)
        {
            filenames = new string[openFileDialog.FileName.Length];
            filenames = openFileDialog.FileNames;
        }
        else
        {
            filenames = null;
            ctImages = null;
            ctRegions = null;
            imagesNumber = 0;
            imageHeight = 0;
            imageWidth = 0;
            bodyRegionIndex = -1;

            pictureBox_original.Image = null;
            pictureBox_segmented.Image = null;
            trackBar.Maximum = 0;
            trackBar.Value = 0;
            toolStripStatusLabel.Text = " ";
            return;
        }
    }
}

```



```

    }
}
else
{
    filenames = null;
    ctImages = null;
    ctRegions = null;
    imagesNumber = 0;
    imageHeight = 0;
    imageWidth = 0;
    bodyRegionIndex = -1;

    pictureBox_original.Image = null;
    pictureBox_segmented.Image = null;
    trackBar.Maximum = 0;
    trackBar.Value = 0;
    toolStripStatusLabel.Text = " ";
    return;
}

```

```

DateTime begin = DateTime.Now;

```

```

// Загрузка изображений

```

```

ctImages = ImageProcessing.GetGrayscaleImagesData(filenames, ref
imagesNumber, ref imageHeight, ref imageWidth);

```

```

// Выделить память для массива

```

```

ctRegions = new int[imagesNumber][[]];
for (int i = 0; i < imagesNumber; i++)
{
    ctRegions[i] = new int[imageHeight][[]];
    for (int j = 0; j < imageHeight; j++)
        ctRegions[i][j] = new int[imageWidth];
}

```

```

bodyRegionIndex = BodySegmentation();

```

```

DateTime end = DateTime.Now;

```

```

Bitmap firstOriginalImage =
ImageProcessing.GetBitmapFrom32Matrix(ctImages[defaultImageShowIndex],
imageHeight, imageWidth);

```

```

        Bitmap firstRegionsImage =
ImageProcessing.GetBitmapFromRegions(ctRegions[defaultImageShowIndex],
        imageHeight, imageWidth, bodyRegionIndex, cFactory);

        trackBar.Maximum = filenames.Length - 1;
        trackBar.Value = defaultImageShowIndex;
        pictureBox_original.Image = firstOriginalImage;
        pictureBox_segmented.Image = firstRegionsImage;
        label_trackBarValue.Text = defaultImageShowIndex.ToString();
        toolStripStatusLabel.Text = string.Format("Complete. {0} images. Elapsed
time is {1}", filenames.Length, end - begin);
        getIndices();

        button_draw_contour.Enabled = true;
    }

    private void getIndices()
    {
//метод вывода в listbox 1)порядкового номера изображения(imagesNumber)
2)числа регионов (subList.Count)

        List<List<int>> newList = new List<List<int>>();
        for (int k = 0; k < imagesNumber; k++)
        {
            List<int> subList = new List<int>();
            subList.Add(ctRegions[k][0][0]);
            for (int i = 0; i < imageHeight; i++)
                for (int j = 0; j < imageWidth; j++)
                {
                    int inList = 1;
                    for (int p = 0; p < subList.Count; p++)
                        if (subList[p] == ctRegions[k][i][j])
                        {
                            inList = 0;
                            break;
                        }

                    if (inList == 1)
                    {
                        subList.Add(ctRegions[k][i][j]);
                    }
                }
            listBox1.Items.Add(k + " " + subList.Count);
            newList.Add(subList);
        }
    }

```

```

    }
}

unsafe static int BodySegmentation()
{
//метод BodySegmentation сегментации кости
// Подготовить данные для использования функции CtBodySegmentation на
языке C

    int imageSize = imageHeight * imageWidth;
    byte[] intensityInput = new byte[imagesNumber * imageSize];
    int[] regionsOutput = new int[imagesNumber * imageSize];

    for (int k = 0; k < imagesNumber; k++)
        for (int i = 0; i < imageHeight; i++)
            for (int j = 0; j < imageWidth; j++)
                intensityInput[(k * imageSize) + (i * imageHeight) + j] =
ctImages[k][i][j];

    // Использование функции на языке C

    int bodyRegionIndex = -1;
    fixed (byte* intensityInput_ptr = intensityInput)
    {
        fixed (int* regionsOutput_ptr = regionsOutput)
        {
            bodyRegionIndex = CtBodySegmentation(intensityInput_ptr,
regionsOutput_ptr, imagesNumber,
            imageHeight, imageWidth, filterWidth, intensityThreshold);

            // Запись результатов, полученных функцией CtBodySegmentation на
языке C++ в переменные C #

            for (int k = 0; k < imagesNumber; k++)
                for (int i = 0; i < imageHeight; i++)
                    for (int j = 0; j < imageWidth; j++)
                        ctRegions[k][i][j] = regionsOutput_ptr[(k * imageSize) + (i *
imageHeight) + j];
        }
    }

    return bodyRegionIndex;
}

```

```

private void trackBar_Scroll(object sender, EventArgs e)
{
// метод trackBar_Scroll для полосы прокрутки - выводит номер элемента

    TrackBar bar = (TrackBar)sender;
    if (bar == null)
        return;

    label_trackBarValue.Text = bar.Value.ToString();
}

private void trackBar_ValueChanged(object sender, EventArgs e)
{
//метод trackBar_ValueChanged обновляет данные при смещении полосы
прокрутки(номера и сами изображения)

    TrackBar bar = (TrackBar)sender;
    if (bar == null)
        return;

    int currentIndex = bar.Value;
    Bitmap ctBitmap =
ImageProcessing.GetBitmapFrom32Matrix(ctImages[currentIndex], imageHeight,
imageWidth);
    Bitmap regionsBitmap =
ImageProcessing.GetBitmapFromRegions(ctRegions[currentIndex], imageHeight,
imageWidth,
        bodyRegionIndex, cFactory);

    pictureBox_original.Image = ctBitmap;
    pictureBox_segmented.Image = regionsBitmap;
    label_trackBarValue.Text = currentIndex.ToString();
}

private void button_saveImage_Click(object sender, EventArgs e)
{
//метод-обработчик события клика на кнопку name
"button_saveImage"сохраняющий одно изображение

    toolStripStatusLabel.Text = "Saving selected image with regions";

    if (pictureBox_segmented.Image == null)
    {

```

```

        MessageBox.Show("Нечего сохранять", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    saveFileDialog.Filter = "Bitmap (*.bmp)|*.bmp";
    if (saveFileDialog.ShowDialog() == DialogResult.OK)
    {
        pictureBox_segmented.Image.Save(saveFileDialog.FileName,
System.Drawing.Imaging.ImageFormat.Bmp);
    }

    toolStripStatusLabel.Text = "Image is saved";
}

private void button_saveAllImages_Click(object sender, EventArgs e)
{
    //метод-обработчик события клика на кнопку name
    "button_saveAllImages"сохраняющий все изображения

    toolStripStatusLabel.Text = "Saving images with regions";

    if (pictureBox_segmented.Image == null)
    {
        MessageBox.Show("Нечего сохранять", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    if (folderBrowserDialog.ShowDialog() ==
System.Windows.Forms.DialogResult.OK)
    {
        if (folderBrowserDialog.SelectedPath != null)
        {
            for (int i = 0; i < filenames.Length; i++)
            {
                string regionsImageName = folderBrowserDialog.SelectedPath + "\\\"
+ (i + 1) + ".bmp";
                Bitmap regionsBitmap =
ImageProcessing.GetBitmapFromRegions(ctRegions[i], imageHeight, imageWidth,
bodyRegionIndex, cFactory);
                regionsBitmap.Save(regionsImageName,
System.Drawing.Imaging.ImageFormat.Bmp);
            }
        }
    }
}

```

```

    }
}

toolStripStatusLabel.Text = "Images are saved";
}

private void button_draw_contour_Click(object sender, EventArgs e)
{ //функция обводит контур по сегментированным точкам

    //преобразуем bmp изображение в матрицу пикселей RGB

    Bitmap bmp = (Bitmap)pictureBox_segmented.Image;
    byte[,] imageMatrix = BitmapToByteRgbQ(bmp);

    //составляет список координат всех точек региона
    List<Point> contour = new List<Point>();
    for (int i = 0; i < bmp.Height; ++i)
        for (int j = 0; j < bmp.Width; ++j)

            //структура rgb модели заключается в трёх слоях 0, 1, 2(которые при
совмещении дают итоговый цвет)
            //j, i - координаты пикселей на изображении
            if (imageMatrix[0, j, i] == 255 && imageMatrix[1, j, i] == 255 &&
imageMatrix[2, j, i] == 255)
                contour.Add(new Point(i, j));

    //найдем центр тяжести многоугольника

    int centerX, centerY, sumX = 0, sumY = 0;
    for (int i = 0; i < contour.Count; ++i)
    {
        sumX += contour[i].X;
        sumY += contour[i].X;
    }
    centerX = sumX / contour.Count;
    centerY = sumY / contour.Count;

    //отсортируем точки по полярному углу относительно центра тяжести
многоугольника

    List<Tuple<double, Point>> polarCoord = new List<Tuple<double,
Point>>();
    for (int i = 0; i < contour.Count; ++i)
    {

```

```

        double angle = Math.Atan2(contour[i].Y - centerY, contour[i].X - centerX);
//вычисляем полярный угол относительно центра тяжести

        polarCoord.Add(new Tuple<double, Point>(angle, contour[i]));
    }
    polarCoord.Sort((x, y) => x.Item1.CompareTo(y.Item1));
    contour.Clear();

    // формируем итоговый отсортированный список точек
    for (int i = 0; i < polarCoord.Count; ++i)

        contour.Add(new Point(polarCoord[i].Item2.X, polarCoord[i].Item2.Y));

    contour.Add(new Point(polarCoord[0].Item2.X, polarCoord[0].Item2.Y));
//еще раз добавили первую точку

    //дорисовываем изображение
    //SolidBrush brush = new SolidBrush(Color.LightCyan); //просто раскрасить
    весь регион
    Pen pen1 = new Pen(Color.White); //используемая кисть

    pen1.Width = 3;

    //Pen pen2 = new Pen(Color.LightBlue); //для сплайна
    //pen2.Width = 3; //для сплайна

    Graphics g = Graphics.FromImage bmp);

    for (int i = 0; i < contour.Count - 1; ++i)
    {
        //g.DrawRectangle(pen, contour[i].X, contour[i].Y, 1, 1); //просто
    раскрасить весь регион
        g.DrawLine(pen1, contour[i], contour[i + 1]); //аппроксимация линиями
    }

    //g.DrawCurve(pen2, contour.ToArray(), 0.1f); //аппроксимация сплайнами

    pictureBox_segmented.Image = bmp; //вывод полученного изображения в
pictureBox
    }

    private unsafe static byte[,] BitmapToByteRgbQ(Bitmap bmp)
    { //функция осуществляет наиболее быстрое преобразование Bitmap в
byte[,]

```

```

    int width = bmp.Width,
        height = bmp.Height;
    byte[,] res = new byte[3, height, width];
    BitmapData bd = bmp.LockBits(new Rectangle(0, 0, width, height),
ImageLockMode.ReadOnly,
    PixelFormat.Format24bppRgb);
    try
    {
        byte* curpos;
        fixed (byte* _res = res)
        {
            byte* _r = _res, _g = _res + width * height, _b = _res + 2 * width *
height;

            for (int h = 0; h < height; h++)
            {
                curpos = ((byte*)bd.Scan0) + h * bd.Stride;
                for (int w = 0; w < width; w++)
                {
                    *_b = *(curpos++); ++_b;
                    *_g = *(curpos++); ++_g;
                    *_r = *(curpos++); ++_r;
                }
            }
        }
    }
    finally
    {
        bmp.UnlockBits(bd);
    }
    return res;
}
}
}

```



## Приложение В

Исходный код класса ImageProcessing.cs

```
using System.Drawing;
using System.Drawing.Imaging;

namespace BodySegmentationApp
{
    public class ImageProcessing
    {
        private static byte[][] GetGrayscaleImageData(Bitmap image) // метод
        GetGrayscaleImageData преобразует входное изображение в массив байтов в
        градациях серого
        { //byte[][] - то, что получаем на выходе
            //(Bitmap image) - то, что получаем на входе, Bitmap- класс, а image -
            название объекта класса
            int imageHeight = image.Height;
            int imageWidth = image.Width;
            byte[][] imageData = new byte[imageHeight][]; // изображение
            представленное в виде двумерного массива байтов
            for (int i = 0; i < imageHeight; i++)
                imageData[i] = new byte[imageWidth];

            // преобразование входного изображения в массив байтов, bitmapData -
            это временная переменная для хранения байтов

            BitmapData bitmapData = image.LockBits(new Rectangle(new Point(0, 0),
                image.Size), ImageLockMode.ReadOnly, image.PixelFormat);
            unsafe
            {
                int byteCounts = 3;
                for (int i = 0; i < imageHeight; i++)
                {
                    byte* OriginalRowPtr = (byte*)bitmapData.Scan0 +
                        i * bitmapData.Stride;
                    for (int j = 0; j < imageWidth; j++)
                    {
                        int ColorPosition = j * byteCounts;
                        imageData[i][j] = OriginalRowPtr[ColorPosition]; // blue color
                    }
                }
            }
            image.UnlockBits(bitmapData);
        }
    }
}
```

```

    return imageData;
}

public static byte[][][] GetGrayscaleImagesData(string[] filenames,
    ref int imagesNumber, ref int imageHeight, ref int imageWidth)
{
    // метод GetGrayscaleImageESData преобразует набор входных
    изображений в трёхмерный массив байтов в градациях серого

    // Загрузить первый растровый рисунок и его параметры

    Bitmap firstBitmap = new Bitmap(filenames[0]);
    imagesNumber = filenames.Length;
    imageHeight = firstBitmap.Height;
    imageWidth = firstBitmap.Width;

    byte[][][] imageData = new byte[imagesNumber][][];
    for (int i = 0; i < imagesNumber; i++)
    {
        imageData[i] = new byte[imageHeight][];
        for (int j = 0; j < imageHeight; j++)
            imageData[i][j] = new byte[imageWidth];
    }
    imageData[0] = GetGrayscaleImageData(firstBitmap);

    // Загрузить остальные изображения

    for (int i = 1; i < filenames.Length; i++)
        imageData[i] = GetGrayscaleImageData(new Bitmap(filenames[i]));

    return imageData;
}

public static Bitmap GetBitmapFrom32Matrix(byte[][] matrix, int height, int
width)
{
    // преобразует матрицу байтов в изображение(обратный метод для метода
    GetGrayscaleImageData)

    //resultBitmap - объявляем переменную для хранения результирующего
    изображения в окне слева

    Bitmap resultBitmap = new Bitmap(width, height,
    PixelFormat.Format24bppRgb);

```

```

unsafe
{ // преобразование

    byte BytesCount = 3;
    BitmapData UpdatingData = resultBitmap.LockBits(new Rectangle(
        new Point(0, 0), resultBitmap.Size), ImageLockMode.WriteOnly,
resultBitmap.PixelFormat);

    for (int i = 0; i < height; i++)
    {
        byte* BitmapRowPtr = (byte*)UpdatingData.Scan0 + i *
UpdatingData.Stride;
        for (int j = 0; j < width; j++)
        {
            int ColorPosition = j * BytesCount;
            BitmapRowPtr[ColorPosition] = matrix[i][j];
            BitmapRowPtr[ColorPosition + 1] = matrix[i][j];
            BitmapRowPtr[ColorPosition + 2] = matrix[i][j];
        }
    }

    resultBitmap.UnlockBits(UpdatingData);
}
//возврат значения
return resultBitmap;
}

public static Bitmap GetBitmapFromRegions(int[][] regionsMatrix, int height,
int width, int bodyRegionIndex, ColorFactory cFactory)
{
    //метод GetBitmapFromRegions получает растровое изображение из регионов,
    где resultBitmap - переменная для хранения растрового изображения

    Bitmap resultBitmap = new Bitmap(width, height,
PixelFormat.Format24bppRgb);

    unsafe
    {
        byte BytesCount = 3;
        BitmapData UpdatingData = resultBitmap.LockBits(new Rectangle(
            new Point(0, 0), resultBitmap.Size), ImageLockMode.WriteOnly,
resultBitmap.PixelFormat);

        int segmentIndex;

```

```

        for (int i = 0; i < height; i++)
        {
            byte* BitmapRowPtr = (byte*)UpdatingData.Scan0 + i *
UpdatingData.Stride;
            for (int j = 0; j < width; j++)
            {
                int ColorPosition = j * BytesCount;

                segmentIndex = regionsMatrix[i][j];

                if (segmentIndex == -1) // Области, отсекаемые как результат
фильтрации
                {
                    BitmapRowPtr[ColorPosition] = 0;
                    BitmapRowPtr[ColorPosition + 1] = 0;
                    BitmapRowPtr[ColorPosition + 2] = 200;
                    continue;
                }

                if (segmentIndex == bodyRegionIndex)
                {
                    BitmapRowPtr[ColorPosition] = 255;
                    BitmapRowPtr[ColorPosition + 1] = 255;
                    BitmapRowPtr[ColorPosition + 2] = 255;
                }
                else
                {
                    BitmapRowPtr[ColorPosition] = 0;
                    BitmapRowPtr[ColorPosition + 1] = 0;
                    BitmapRowPtr[ColorPosition + 2] = 0;
                }
            }
        }

        resultBitmap.UnlockBits(UpdatingData);
    }

    return resultBitmap;
}
}
}

```