

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Ижевский государственный технический
университет имени М.Т. Калашникова»

На правах рукописи

Кучуганов Александр Валерьевич

МЕТОДОЛОГИЯ СЕМАНТИЧЕСКОГО АНАЛИЗА И ПОИСКА
ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

Специальность 05.13.01 – Системный анализ, управление и обработка
информации (в промышленности) (технические науки)

Диссертация на соискание ученой степени
доктора технических наук

Научный консультант:
д.т.н., проф. В.А. Алексеев

Ижевск – 2018

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
1. ИССЛЕДОВАНИЕ И АНАЛИЗ СУЩЕСТВУЮЩИХ ПОДХОДОВ, МЕТОДОВ И ТЕХНОЛОГИЙ ОБРАБОТКИ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ	17
1.1. Особенности представления и использования различных видов графической информации.....	17
1.2. Краткий обзор существующих методов обработки графической информации в автоматизированных системах	21
1.2.1. Анализ методов выделения контуров и векторизации изображений	23
1.2.2. Анализ методов скелетизации изображений.....	26
1.2.3. Анализ методов цветовой сегментации изображений	32
1.2.4. Анализ методов сопоставления графов	37
1.3. Анализ существующих подходов и технологий поиска графической информации в автоматизированных системах	40
1.4. Сравнительные характеристики подходов, методов и технологий обработки графической информации.....	50
1.5. Выводы, цель и задачи исследования.....	53
2. КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ СЕМАНТИКИ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ.....	56
2.1. Задачи анализа изображений.....	56
2.2. Обоснование методологии содержательного описания и поиска графической информации.....	60
2.3. Особенности применения семиотического подхода к проблематике моделирования, обработки и анализа графической информации	65
2.4. Выводы по главе	73
3. СИНТАКСИЧЕСКИЙ АНАЛИЗ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ	75
3.1. Разработка методов локальной обработки.....	75
3.2. Разработка методов фрагментарной обработки	82

3.3. Общий алгоритм анализа и синтеза морфологических элементов изображения	88
3.4. Разработка методов аппроксимации графической информации	94
3.5. Синтаксическая модель изображения	104
3.6. Выводы по главе	112
4. РАСПОЗНАВАНИЕ ОБРАЗОВ, ЛОГИЧЕСКИЙ АНАЛИЗ И ВЕРБАЛИЗАЦИЯ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ	115
4.1. Многослойный атрибутивный граф изображения	115
4.2. Метод сопоставления нечётких пространственно нагруженных графов ...	118
4.3. Поиск и сопоставление графической информации, представленной нечёткими пространственно нагруженными графами.....	122
4.4. Логический анализ и вербализация графической информации.....	127
4.4.1. Дескрипционная логика с расширением на данные, представленные в виде атрибутивных графов.....	130
4.4.2. Функциональные отношения (атрибуты), описывающие геометрические характеристики формы цветковых сегментов и их скелетонов	136
4.4.3. Алгоритм логического вывода в ДЛ <i>ALC(GI)</i>	140
4.4.4. Примеры предметных терминологий графической информации	145
4.5. Контроль корректности вербальных представлений графической информации.....	155
4.6. Классификация. Группирование деталей в задачах технической подготовки производства	163
4.7. Выводы по главе	169
5. РАЗРАБОТКА ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ АНАЛИЗА, СОДЕРЖАТЕЛЬНОГО ОПИСАНИЯ И ПОИСКА ГРАФИЧЕСКОЙ ИНФОРМАЦИИ.....	173
5.1. Программный комплекс анализа изображений.....	173
5.2. Векторизация архивных чертежей.....	175
5.3. Графический поиск чертежей в архивах конструкторско-технологических данных.....	181

5.4. Выводы по главе	197
ЗАКЛЮЧЕНИЕ	200
СПИСОК ЛИТЕРАТУРЫ.....	203
Приложение А. Анализ и дешифрирование аэрокосмоснимков.....	222
Приложение Б. Анализ папиллярных узоров пальцевых отпечатков	231
Приложение В. RECO - программная система для распознавания старославянских текстов	238
Приложение Г. Акты внедрения (использования) результатов диссертационной работы.....	244

ВВЕДЕНИЕ

Большую часть объема информации, обрабатываемой человеком в процессе исследовательской, проектно-конструкторской и управленческой деятельности, составляет графическая информация, представленная в виде фотоизображений, рисунков, эскизов, чертежей. Но и сейчас, несмотря на бурное развитие технических и программных средств вычислительной техники, именно анализ графической информации и описание ее на естественном или формализованном языке в большинстве предметных областей науки и техники осуществляется, как правило, в интерактивном (диалоговом) режиме между человеком и машиной.

Так, при обработке данных видеонаблюдений, в том числе аэрокосмических снимков, в большинстве случаев практического применения поиск и выделение (обводка) объектов дорабатываются вручную из-за не вполне удовлетворительной релевантности результатов, полученных с помощью имеющихся программных средств.

Числовая информация также во многих случаях представляется лицу, принимающему решение, в графическом виде, что повышает ее наглядность и, следовательно, качество решения.

В связи с этим, все больше внимания уделяется развитию методов и средств автоматизации анализа графической информации: определение формы, структуры и отношений изображенных объектов, сопоставления, словесного описания, контентного поиска изображений в хранилищах данных и сети Интернет, создания онтологий знаний об изображениях для информационных технологий в различных предметных областях (биология, медицина и т.д.).

Системы поддержки принятия решений (СППР) являются неотъемлемой частью большинства современных автоматизированных систем. Для анализа информации в СППР используются средства:

- Data Mining – набор инструментальных средств, позволяющих извлекать

знания из структурированных данных в виде закономерностей, тенденций, шаблонов, правил, взаимосвязей;

- технология Text Mining – набор инструментальных средств, позволяющих анализировать текстовую информацию;

- технология Image Mining – набор инструментальных средств для обработки изображений, выделения объектов, определения их характеристик, взаимосвязей, закономерностей.

На сегодняшний день наиболее распространенными можно считать четыре подхода к анализу изображений:

- Лингвистический подход (К.Фу), в рамках которого осуществляется синтаксическое распознавание объектов строго predetermined структуры (буквы, символы, объекты простой формы), что существенно ограничивает область применения. Например, слитный рукописный текст распознается с большим количеством ошибок (А.Л. Горелик, А.Н. Горошкин, И.А. Зеленцов, Р.Б. Поцепаев, В.А. Скрипкин, Джин Чен, Даниэль Лопрести).

- Искусственные нейронные сети. Их недостатками считается необходимость длительного обучения каждому классу объектов, непредсказуемость обучения, непрозрачность принимаемого решения и ресурсоёмкость (Я. Бенджио, Л. Ботоу, Ю. ЛеЧин, С. Осовский, С. Хайкин, П. Хафнер).

- Дескрипционные алгебры изображений – описывают изображение в терминах дескрипторов – результатов вычисления различных интегрирующих и дифференцирующих функций, т.е. распознают объекты как целостные образы, но не осуществляют структурный анализ изображенных объектов (И.Б. Гуревич, Ю.И. Журавлев, И.В. Корябкина, Ю.О. Трусова, В.В. Яшина).

- Технологии CBIR (Content-Based Image Retrieval) осуществляют контентный поиск изображений по образцу на основе интегральных характеристик, гистограмм ориентации, локальных дескрипторов и их комбинаций. Выявляются контуры и их расположение. Чтобы машина вывода знала, какие изображения считать похожими, применяют обучение с учителем.

Недостатком существующих решений в рамках этого подхода следует считать ограниченность набора признаков формы. Значения признаков субъективны, поскольку принадлежность совокупностей локальных особенностей тому или иному предмету - их семантику - назначает человек, т.е. возникает разрыв между синтаксисом и семантикой изображений (Heba Aboulmagd, G.A. Bilodeau, R. Bergevin, R. Balasubramaniam, Neamat El-Gayar, R. Krishnapuram, S. Medasani, Hoda Onsi, S.-H. Jung, Y.-S. Choi).

Таким образом, несмотря на существенные достижения в области обработки изображений и распознавания образов, проблема "понимания" компьютерными системами и извлечения знаний из графической информации остается актуальной по сей день.

По-видимому, проблемы комплексного анализа графической информации от набора разноцветных пикселей до логического вывода понятийного описания структуры изображенных объектов, их деталей и отношений, содержательного поиска, выявления сходства и отличий, единообразного формализованного описания и классификации кроются в недостаточности существующего теоретического аппарата в части семантического анализа, поиска и классификации графической информации. Дальнейшее развитие научных основ семантического и прагматического анализа графической информации: извлечения формализованных представлений, сопоставления моделей, моделирования рассуждений на графической информации, позволило бы преодолеть семантический разрыв между сложными синтаксическими структурами, которые характерны для графической информации, и автоматическим ее понятийным описанием.

Предлагаемое решение проблемы семантического и структурного анализа, поиска и сопоставления графической информации относится к категории контентного поиска (CBIR), опирается на семиотический подход к организации знаковых коммуникативных систем и направлено на повышение качества анализа, степени детализации описаний изображенных объектов, релевантности их поиска и сопоставления с целью повышения эффективности современных СППР.

Использует биоинспирированные алгоритмы выделения морфологических признаков и синтаксических структур, математический аппарат теории графов и дескрипционной логики.

Степень научной разработанности проблемы. Обработке и анализу изображений посвящено множество работ отечественных и зарубежных ученых: Бонгарда М.М., Васина Ю.Г., Гуревича И.Б., Гонсалеса Р., Журавлева Ю.И., Лебедева Д.С., Марра Д., Мучника И.Б., Нарасимхана Р., Павлидиса Т., Прэтта У., Роджерса Д., Розенфельда А., Славина О.А., Стокхэма Т., Файна В.С., Форсайт Д., Фу К., Харалика Р.М., Цуккермана И.И., Шермана Г, Ярославского Л.П. и др.

В становление и развитие теоретического аппарата инженерии знаний внесли существенный вклад работы В.Н. Вагина, Г.Д. Волковой, Т.А Гавриловой, А.П. Еремеева, Л. Заде, В.М. Курейчика, Дж. Люггера, Н.Н. Непейводы, П. Норвига, Г.С. Осипова, Д.А. Поспелова, С. Рассела, А.В. Смирнова, П.И. Соснина, В.Б. Тарасова, В.Ф. Хорошевского и др.

Объектом исследования является процесс семантического анализа и поиска графической информации в системах поддержки принятия решений.

Предметом исследования являются модели, методы и алгоритмы семантического анализа и поиска графической информации двух разновидностей: изображения линейных объектов (рисунки, чертежи, рукописи и т.п.) и снимки природных и искусственных объектов.

Цель работы – повышение релевантности результатов семантического анализа и поиска графической информации в системах поддержки принятия решений путем разработки методологии семантического анализа и поиска графической информации на основе семиотического подхода к организации знаковых коммуникативных систем.

Для достижения поставленной цели требуется решение следующих **задач**:

- 1) разработка концепции семантического анализа графической информации;

2) разработка методов анализа и формализованного описания структуры графической информации и синтеза синтаксических моделей, позволяющих упростить задачи семантического анализа;

3) разработка методов сопоставления, выявления сходства и отличий синтаксических моделей графической информации для задач поиска аналогов и анализа отличительных особенностей;

4) разработка алгоритмов поиска графической информации по заданному образцу;

5) разработка онтологической базы знаний изображений для семантического и прагматического анализа графической информации;

6) разработка алгоритмов поиска графических объектов, удовлетворяющих заданным определениям;

7) экспериментальное исследование разработанных моделей, методов и алгоритмов.

Научная новизна диссертационного исследования заключается в следующем:

1. Впервые предложена концептуальная модель процесса семантического анализа графической информации, основанная на когнитивной модели "треугольник Фреге" восприятия информации человеком и отличающаяся развернутым представлением графических информационных структур с целью совершенствования алгоритмов анализа и преобразования представлений.

2. Разработана новая математическая модель изображения в виде многослойного атрибутивного графа, вершины и ребра которого содержат количественные и качественные значения атрибутов, характеризующих объекты изображения, отличающаяся тем, что с целью получения более информативного описания изображений, включает границы областей, скелетоны, контуры и наборы их атрибутов формы и пространственной ориентации. Позволяет осуществлять автоматический синтез синтаксических моделей графической информации в виде многослойных атрибутивных графов, освобождает экспертов

от задания синтаксиса изображений, обеспечивает решение задач семантического и прагматического анализа графической информации.

3. Разработан новый метод сопоставления атрибутивных графов изображений, отличающийся тем, что с целью повышения надежности распознавания и автоматического выявления сходства и отличий между объектами для каждой вершины сопоставляемых графов формируются лучевые графы, в которых дуги упорядочены по значениям атрибутов пространственной ориентации.

4. Впервые разработана онтологическая база знаний изображений на основе дескрипционной логики *ALC*, отличающаяся расширением на область данных, представленных в виде атрибутивных графов, позволяющая осуществлять анализ и поиск пространственно распределенных данных.

5. Разработан новый метод выделения скелетона графических объектов, позволяющий снизить вычислительную сложность за счет разбиения фигуры на выпуклые многоугольники в местах, где граница имеет отрицательный перегиб, и выделения осевых линий полученных выпуклых фигур с последующим объединением их в связные цепочки.

6. Разработана новая методология автоматического семантического анализа и поиска графической информации, основанная на семиотическом подходе к организации знаковых коммуникативных систем, включающая три уровня новых моделей и методов анализа пространственных отношений: синтаксический (морфология, грамматики конструкций), семантический (распознавание и логический вывод), прагматический (вербализация), развивающая средства автоматического анализа и извлечения знаний из графической информации.

Теоретическая значимость. В ходе диссертационного исследования разработана методология семантического анализа и поиска графической информации, содержащая взаимосвязанный комплекс моделей и методов, методики их реализации при компьютерном моделировании, проведены вычислительные эксперименты.

Практическая значимость. На основе разработанной методологии созданы информационные технологии и программный комплекс, осуществляющий содержательное описание чертежей и аэрокосмоснимков, поиск аналогов, выявление сходства и отличий, автоматическую кластеризацию по геометрическим характеристикам.

Методы исследования. Теоретические исследования выполнены с использованием математической логики, нечеткой логики, теории множеств, теории графов, теории распознавания образов, вычислительной геометрии, кластерного анализа.

Основные положения, выносимые на защиту:

1. Концептуальная модель процесса семантического анализа и поиска графической информации, основанная на когнитивной модели "треугольник Фреге" восприятия информации человеком и содержащая развернутые представления графических информационных структур, позволяет конкретизировать задачи анализа и преобразования представлений *(соответствует п. 2 паспорта специальности «Формализация и постановка задач системного анализа, оптимизации, управления, принятия решений и обработки информации»)*.

2. Предложенная модель изображения в виде многослойного атрибутивного графа позволяет осуществлять автоматический синтез синтаксических моделей графической информации, освобождая экспертов от задания синтаксиса изображений, обеспечивает решение задач семантического и прагматического анализа графической информации *(соответствует п. 3 паспорта специальности «Разработка критериев и моделей описания и оценки эффективности решения задач системного анализа, оптимизации, управления, принятия решений и обработки информации»)*.

3. Новый метод сопоставления и анализа атрибутивных графов пространственных отношений позволяет автоматически выявлять сходство и отличия между объектами графической информации в процессе поиска аналогов *(соответствует п.4 паспорта специальности «Разработка методов и*

алгоритмов решения задач системного анализа, оптимизации, управления, принятия решений и обработки информации»).

4. Впервые разработанная онтологическая база знаний изображений на основе дескрипционной логики *ALC*, расширенной на область данных, представленных в виде атрибутивных графов, позволяет осуществлять анализ пространственных отношений между структурными элементами объектов, сокращает область интерпретации понятий и количество интерпретирующих функций за счет перехода от пиксельного представления данных к атрибутивным графам (*соответствует п. 5 паспорта специальности «Разработка специального математического и программного обеспечения систем анализа, оптимизации, управления, принятия решений и обработки информации» и п.10 «Методы и алгоритмы интеллектуальной поддержки при принятии управленческих решений в технических, экономических, биологических, медицинских и социальных системах»*).

5. Новый метод аппроксимации графической информации путем выделения скелетона графических объектов с помощью разбиения фигуры на выпуклые многоугольники в местах, где граница имеет отрицательный перегиб, позволяет свести вычислительную сложность алгоритма к квадратичной зависимости от количества отрезков аппроксимирующих границы, в отличие от квадратичной зависимости от числа пикселей объекта в методах утоньшения линий или от числа пикселей границы в методах, основанных на диаграммах Вороного (*соответствует п. 4 паспорта специальности «Разработка методов и алгоритмов решения задач системного анализа, оптимизации, управления, принятия решений и обработки информации» и п. 12 «Визуализация, трансформация и анализ информации на основе компьютерных методов обработки информации»*).

6. Разработанная новая методология автоматического семантического анализа и поиска графической информации, основывающаяся на семиотическом подходе к организации знаковых коммуникативных систем, включающая три уровня моделей и методов анализа пространственных отношений: синтаксический

(морфология, грамматики конструкций), семантический (распознавание и логический вывод), прагматический (вербализация), позволяет преодолеть семантический разрыв между сложными синтаксическими структурами, которые характерны для графической информации, и автоматическим ее понятийным описанием, что способствует дальнейшему развитию технологий извлечения знаний из графической информации (*соответствует п.12 «Визуализация, трансформация и анализ информации на основе компьютерных методов обработки информации»*).

Достоверность и обоснованность полученных в работе результатов и выводов подтверждается:

1) корректным применением математического аппарата, а также сопоставлением результатов теоретических исследований и экспериментальных данных с опубликованными результатами других исследователей;

2) результатами применения разработанных моделей, методов и алгоритмов в автоматизированных системах поиска чертежей и эскизов по образцу, дешифрирования аэрокосмических снимков;

3) положительными результатами проведенных экспериментальных исследований и опытом практической эксплуатации разработанного программного комплекса, что подтверждается актами внедрения.

Реализация результатов работы. Создан программный комплекс для компьютерного моделирования методов семантического анализа, поиска и сопоставления графической информации.

Тема диссертационной работы поддержана:

- грантом Российского Фонда Фундаментальных Исследований (проект № 11-07-00632-а) «Создание программной системы анализа фотоизображений на основе когнитивного подхода»;

- грантом Российского Фонда Фундаментальных Исследований (проект № 15-07-08077 А) «Разработка, исследование и программная реализация системы распознавания старопечатных кириллических текстов с помощью грамматического словаря древнерусского языка»;

- Госзаданием МОиН РФ «Методология и система поиска, семантического реферирования и извлечения знаний из русско- и англоязычных текстовых и графических документов (в различных информационных ресурсах)».

Разработанные программные модули внедрены (использованы):

в работе ООО "Нордика Стерлинг", при реализации проектов по техническому перевооружению предприятий;

в работе ОАО «Ижевский электромеханический завод «Купол» для поиска графической информации в хранилищах данных;

в отделе патентно-информационных исследований и интеллектуальной собственности ИжГТУ – для автоматизированной системы патентно-лицензионного поиска (Государственный контракт № 01.647.12.3009 в рамках ФЦП «Развитие инфраструктуры nanoиндустрии в Российской Федерации на 2008-2010 годы»);

в рамках госбюджетной темы № 4043 Госзаказ МОиН по теме: "Разработка и экспериментальное исследование системы аэрокосмического и геоинформационного мониторинга" для визуализации результатов геоэкологических исследований северных экосистем;

в учебном процессе на кафедрах «Вычислительная техника» и «Автоматизированные системы обработки информации и управления» Ижевского государственного технического университета имени М.Т. Калашникова при выполнении лабораторных работ по дисциплинам "Инженерная и компьютерная графика", "Растровая и векторная графика", "Геоинформационные системы и технологии", курсовых и дипломных работ.

Апробация работы. Основные положения диссертационной работы докладывались: на Международной научно-технической конференции «Информационные технологии в инновационных проектах» (Ижевск, 2001); на 11-й Международной конференции по компьютерной графике и машинному зрению Графикон - 2001 (Нижний Новгород, 2001); на Международной научно-технической конференции IEEE AIS'05 (Дивноморское, 2005); на Международной научно-технической конференции IEEE AIS'06 (Дивноморское, 2006); на

Международной научно-технической конференции IEEE AIS'07 (Дивноморское, 2007); на Международной научно-технической конференции IEEE AIS'08 (Дивноморское, 2008); на 9-ой Международной конференций «Pattern Recognition and Image Analysis: New Information Technologies» (PRIA-9-2008) (Нижний Новгород, 2008); на международном конгрессе по интеллектуальным системам и информационным технологиям «AIS-IT'10» (Дивноморское, 2010); на междунар. науч. конф. Информационные технологии и письменное наследие (Уфа, 28-31 октября 2010 г.); Второй международной конференции «Трехмерная визуализация научной, технической и социальной реальности. Технологии высокополигонального моделирования» (Ижевск, 2010); Международном конгрессе по интеллектуальным системам и информационным технологиям «IS&IT'11» (Дивноморское, 2011); 8th Open German-Russian Workshop «PATTERN RECOGNITION and IMAGE UNDERSTANDING» (Nizhny Novgorod, Russia, 21-26 November 2011); Пятой международной конференции по когнитивной науке, 18 – 24 июня (Калининград, 2012); Международном конгрессе по интеллектуальным системам и информационным технологиям «IS&IT'13» (Дивноморское, 2013), Международной научно-практической конференции Теоретические и прикладные вопросы образования и науки (Тамбов, 2014), Международной конференции "Информационные технологии в науке, образовании, телекоммуникации и бизнесе IT + S&E`14" (АР Крым, Ялта-Гурзуф, 2014 г.), Международном конгрессе по интеллектуальным системам и информационным технологиям «IS&IT'14» (Дивноморское, 2014), Международной конференции "Информационные технологии в науке, образовании, телекоммуникации и бизнесе IT + S&E`15" (АР Крым, Ялта-Гурзуф, 2015 г.), Международном конгрессе по интеллектуальным системам и информационным технологиям «IS&IT'15» (Дивноморское, 2015), на международном форуме “Instrumentation Engineering, Electronics and Telecommunications – 2015” в рамках международной конференции “Приборостроение в XXI веке. Интеграция науки, образования и производства” (Ижевск, 2015), Международном конгрессе по интеллектуальным системам и информационным технологиям «IS&IT'16» (Дивноморское, 2016),

Международной научной конференции "Информационные технологии и письменное наследие (EI'Manuscript – 2016)", Седьмой международной конференции "Системный анализ и информационные технологии" САИТ - 2017 (13 - 18 июня 2017 г., г. Светлогорск, Россия), Международном конгрессе по интеллектуальным системам и информационным технологиям «IS&IT'17» (Дивноморское, 2017), конференция «Искусственный интеллект: проблемы и пути их решения – 2018» (Москва, 2018).

Личный вклад автора. Научные результаты диссертационной работы получены лично автором. Экспериментальные данные получены с помощью программного комплекса, который был разработан по инициативе, под руководством и при непосредственном участии автора, на основе его теоретических положений и методик.

Публикации. По материалам диссертации опубликовано 67 печатных работы, в том числе 30 статей в изданиях из перечня, рекомендованного ВАК Минобрнауки РФ (из них 5 статей в БД Scopus, 3 статьи в БД Web of Science). А также 2 монографии, одна из них коллективная (Изд-во: Физматлит).

Получено 8 свидетельств РФ о государственной регистрации программ для ЭВМ.

1. ИССЛЕДОВАНИЕ И АНАЛИЗ СУЩЕСТВУЮЩИХ ПОДХОДОВ, МЕТОДОВ И ТЕХНОЛОГИЙ ОБРАБОТКИ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

В главе проанализированы существующие способы формирования описания графической информации, представленной в виде изображений, чертежей, рисунков, фотографий и методы их поиска в хранилищах данных. Кратко описаны существующие методы выделения геометрических и цветовых особенностей на изображениях: контуров, цветовых сегментов, скелетонов. Проанализированы их возможности для решения задач семантического анализа изображений.

1.1. Особенности представления и использования различных видов графической информации

Графическая информация, представляется в виде изображений, рисунков, эскизов, чертежей, фотографий. Высокая информативность изображений порождает многообразие прикладных задач, решаемых с их помощью.

Компьютерная обработка оцифрованных изображений – одна из наиболее развитых областей применения вычислительной техники. Она используется для решения задач во многих сферах научной и профессиональной деятельности: биологии, медицине, геодезии, картографии, в различных опытно-конструкторских и научно-исследовательских разработках. Цифровой обработке изображений посвящено множество работ отечественных и зарубежных ученых, издаются специализированные научно-прикладные журналы. Следует отметить,

что соответствующее программное обеспечение не всегда успевает за вновь возникающими потребностями пользователей, в частности, в таких задачах, как Internet-поиск, создание, анализ и хранение электронных документов и др.

Большинство методов анализа и распознавания ориентированы на обработку двухградационных, линейчатых и других изображений, заранее оговоренных классов объектов. До сих пор даже такие емкие прикладные области как картография или чертежные архивы требуют значительной доли ручного труда.

Анализ и оценивание информации, представленной в виде изображений, нужны для решения прикладных задач идентификации, диагностики, прогнозирования и принятия решений. Информация, полученная из изображений, имеет принципиально важное значение для различных областей человеческой деятельности: промышленности (материаловедение, неразрушающий контроль, техническая диагностика), здравоохранения (медицинская диагностика и прогноз развития заболевания), обороны (обнаружение несанкционированных запусков, дистанционное зондирование, интеллектуальное оружие), экологии (мониторинг), предотвращения и прогноза природных катастроф и чрезвычайных ситуаций (извержений вулканов, лесных пожаров, наводнений, цунами, техногенных катастроф).

Графическая информация это:

- чертежи, схемы, рисунки, диаграммы;
- фотоизображения (портрет, натюрморт, пейзаж, жанровая сцена);
- аэрокосмоснимки;
- микроскопия (материалы, смеси);
- медицинские снимки: томография, рентгеноскопия, ультразвуковые исследования.

Информационный поиск играет важную роль в процессе проектной и исследовательской деятельности для извлечения и анализа полезной информации из текстов и изображений, особенно, в условиях информационного «взрыва», связанного с развитием сети Интернет. Так в соответствии с типовыми

требованиями, проектная документация должна содержать сведения о результатах проведенных патентных исследований и использованных в проекте изобретениях.

Задачи, в которых семантический анализ графической информации имеет большое значение:

- адресный поиск текстов и изображений;
- семантический поиск текстов;
- реферирование текстов;
- поиск изображений по содержанию;
- приём заявок на российский патент на полезную модель, изобретение, промышленный образец; на регистрацию товарных знаков, торговых марок, знаков обслуживания и наименования мест происхождения товаров;
- международная регистрация товарных знаков по Мадридской системе или путем подачи прямой заявки – в Евросоюз;
- регистрация программ для ЭВМ и баз данных;
- обеспечение режима правовой охраны секретов производства;
- сопровождение лицензионных договоров;
- регистрация авторских прав;
- оценка интеллектуальной собственности;
- перевод патентной и научно-технической документации.

Параллельно с задачами автоматизации получения и анализа информации из изображений остается актуальной и задача автоматизации разработки самих систем анализа графической информации.

Проблемами систем анализа графической информации являются, во-первых, недостаточная степень автоматизации, а во-вторых - отсутствие возможности получения не только заложенных разработчиками, но и более сложных характеристик, в частности описывающих форму объектов или динамику изменения (тренды).

Первыми начали развиваться системы распознавания (чтение текстов, обнаружение заданных объектов в данных видеонаблюдений).

Наиболее известными и, можно сказать, типичными являются системы OCR (оптического распознавания символов).

Распознанная страница может достигать наилучшей точности распознавания 99,9% для чистых изображений, составленных из обычных шрифтов, используя при этом метод проверки по словарю.

Недостатком систем распознавания зрительных образов считается их узкая специализация - "заточенность" на конкретный класс объектов. Так например, система ABBYY FineReader читает печатные тексты с надежностью до 98%, но, фактически, не справляется с рукописным текстом.

Поиск изображений по содержанию (CBIR - Content-Based Image Retrieval). - раздел компьютерного зрения, занимающийся поиском изображений по содержательным критериям. Он особенно требователен к автоматическому анализу и описанию содержания изображений, например, цвету представленных на нём объектов, их форме, текстуре, композиции сцены. При отсутствии возможности проанализировать сцену при поиске рассматриваются метаданные: ключевые слова, метки.

Сегодня CALS технологии (Continuous Acquisition and Lifecycle Support) являются, по сути, единственной глобальной стратегией информационного сопровождения бизнес-процессов, выполняемых в ходе жизненного цикла продукта. И одной из самых «тяжелых» составляющих являются электронные архивы конструкторско-технологической документации. Конструкторские бюро заводов работают, в основном, над модификациями изделий, запущенных в производство. Колоссальные знания «пылятся» на бумажных носителях в пассивном состоянии. Лишь небольшая их часть сканирована в различные растровые форматы. Еще меньшая часть переведена в векторную форму. Для российских предприятий проблема переноса чертежей с бумажных носителей в электронную форму особенно болезненна, поскольку требует высоких трудозатрат.

1.2. Краткий обзор существующих методов обработки графической информации в автоматизированных системах

На сегодняшний день наиболее распространенными можно считать следующие подходы к анализу изображений:

1. Структурный (лингвистический) подход, в рамках которого осуществляется распознавание объектов строго predetermined структуры (буквы, символы, объекты простой формы), что существенно ограничивает область применения. Например, слитный рукописный текст распознается с большим количеством ошибок (А.Л. Горелик, А.Н. Горошкин, И.А. Зеленцов, Р.Б. Поцепаев, В.А. Скрипкин, Джин Чен, Даниэль Лопрести).

2. Искусственные нейронные сети (ИНС). Их недостатками считается необходимость длительного обучения каждому классу объектов, непредсказуемость обучения, непрозрачность принимаемого решения и ресурсоёмкость (Я. Бенджио, Л. Ботоу, Ю. ЛеЧин, С. Осовский, С. Хайкин, П. Хафнер). С целью повышения эффективности ИНС О.А. Славиним предложен ряд адаптивных моделей и методов для задачи распознавания отсканированных текстовых документов [84].

3. Дескрипционные алгебры изображений – описывают изображение в терминах дескрипторов – результатов вычисления соответствующих функций, либо распознают объекты как целостные образы, но не осуществляют структурный анализ изображенных объектов (И.Б. Гуревич, Ю.И. Журавлев, И.В. Корябкина, В.В. Яшина).

4. Семантическая классификация изображений (CBIR - Content-Based Image Retrieval). Методы CBIR осуществляют контентный поиск изображений на основе семантических признаков. В процессе поиска выполняется сопоставление графов. Недостатком подхода следует считать ограниченность набора признаков формы. Значения признаков субъективны, поскольку задаются пользователем, и, как следствие, результаты поиска зачастую слабо релевантны поисковому запросу

(Heba Aboulmagd, G.A. Bilodeau, R. Bergevin, R. Balasubramaniam, Neamat El-Gayar, R. Krishnapuram, S. Medasani, Hoda Onsi, S.-H. Jung, Y.-S. Choi).

5. В работах Луцива В.Р. [60, 61] представлен объектно-независимый подход к структурному анализу изображений и модель структурной организации изображений окружающего мира на разных иерархических уровнях от микроуровня текстур до макроуровня объектов и подобъектов. Суть этой модели заключается в объектно-независимой структурной декомпозиции, построении структурных описаний изображений и объектно-независимом методе сопоставления изображений реальных сцен с помощью их иерархических структурных описаний. Описание изображения составляется из алфавита - производных элементов, которые выделяются на изображении, преобразованном в Фурье-спектр, путём поиска максимума функции корреляции изображения с весовой функцией детектора. В результате элементами первого уровня описания становятся отрезки контуров (прямые и дуги), а второго уровня "компактные" структурные элементы и их границы. Достоинством подхода следует считать независимость от области применения. К недостаткам можно отнести трудоёмкость создания набора детекторов для различных компактных подструктур.

6. Эвристический подход широко применяется для разных задач. Недостатком его считается зависимость наборов экспертных правил от предметной области, ошибки при возникновении неучтённых ситуаций и, зачастую, противоречия в правилах при большом их количестве.

В данном случае критерием для выбора подхода является применимость к задачам семантического анализа графической информации, а именно, возможность описания изображений объектов с иерархической структурой. Для чего, необходимо сочетание нескольких подходов.

1.2.1. Анализ методов выделения контуров и векторизации изображений

Методы обработки изображений, в основном, предназначены для повышения их визуального качества. Частотные способы обработки хорошо развиты, но поскольку они требуют значительных вычислительных ресурсов, в практике применяются реже, чем пространственные: разностные, сглаживающие, контрастирующие, гистограммные и т.п.

По математическому принципу все методы обработки растров в современных компьютерных системах можно подразделить на два крупных класса: методы в пространственной и частотной областях [97, 9, 77, 78]. Пространственные методы относятся к плоскости и анализируют расположение пикселей изображения. Методы обработки в частотной области (частотные методы) анализируют сигнал, формируемый изображением и модифицируют его, например, с помощью преобразования Фурье.

Рассмотрим пространственные методы, так как они напрямую связаны с темой выделения объектов на изображении.

Пространственные методы можно условно подразделить на группы методов поэлементной обработки изображения и методов обработки по маске.

С помощью методов первой группы решаются такие задачи как аффинные преобразования, градационные, например, преобразование в негатив, логарифмические, степенные преобразования, кусочно-линейные и др., а также задачи, в которых используется анализ гистограмм (гистограммные методы). Отличительной чертой этих методов является обработка каждого пиксела изображения независимо от окружающих.

Ко второй группе относятся задачи всевозможной фильтрации изображений, среди которых особое место занимают задачи выделения краев (контуров) объектов. В этих методах результирующее значение каждого пиксела зависит от его ближайшего окружения [97, 85, 17].

Выделение контуров. Для выделения перепадов яркости и границ однородных областей применяют специализированные пространственные фильтры, иногда называемые локальными операторами, например, оператор Собеля [165]. Рассмотрим его подробнее в качестве классического примера выделения контуров.

Оператор состоит из двух частей: X-оператор и Y-оператор (рисунок 1.1), которые представляют собой две соответствующие матрицы 3×3 чисел [97, 69]:

- 1	0	1
- 2	0	2
- 1	0	1

а)

- 1	- 2	- 1
0	0	0
1	2	1

б)

Рисунок 1.1 - Операторы Собеля:

а) X-оператор Собеля;

б) Y-оператор Собеля

Каждая матрица является экземпляром множества S с рабочей точкой, соответствующей центральному пикселу. X-оператор Собеля, дает величину горизонтальной составляющей градиента яркости в центральной точке этой матрицы, а Y-оператор – величину вертикальной составляющей градиента. Градиент функции яркости представляет собой результат пространственного дифференцирования, а именно, направление наибоыстрейшего роста функции в данной точке. В непрерывных системах модуль (длина вектора) градиента выражается следующей формулой:

$$|\nabla f(x, y)| = \sqrt{\left(\frac{\partial f(x, y)}{\partial x}\right)^2 + \left(\frac{\partial f(x, y)}{\partial y}\right)^2}, \quad (1.1)$$

где: $\nabla f(x, y)$ – градиент функции яркости $f(x, y)$ в точке (x, y) ;

x, y – пространственные координаты плоскости xOy .

В отношении дискретных систем X- и Y- операторы Собеля являются своеобразной дискретной аппроксимацией входящих в правую часть выражения

(1.1) частных производных по ∂x и ∂y соответственно. Обработка каждого пиксела изображения происходит сначала одним оператором, затем другим.

Операторы градиента. Первая производная изображения вычисляется с помощью дискретного приближения двумерного градиента [9]. По определению, градиент изображения $f(x, y)$ в точке (x, y) — это вектор

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}.$$

Модуль вектора или градиент равен значению максимальной скорости изменения функции.

$$|\nabla f| = \sqrt{G_x^2 + G_y^2}.$$

Направление вектора градиента. Угол $\alpha(x, y)$ между направлением вектора ∇f в точке (x, y) и осью x равен:

$$\alpha(x, y) = \operatorname{arctg} \left(\frac{G_y}{G_x} \right).$$

Направление контура в точке (x, y) перпендикулярно направлению вектора градиента в этой точке [9].

Перечисленные методы выделения первичных признаков достаточно просты, не требовательны к вычислительным ресурсам. Применяются во многих системах обработки изображений. В качестве критериев выбора для задач семантического анализа графической информации следует обратить внимание на надёжность работы в условиях среднего, низкого и переменного качества изображений, а также управляемость и адаптируемость в процессе анализа, когда требуются возвраты для уточнения информации в проблемных местах.

1.2.2. Анализ методов скелетизации изображений

Считается, что скелетоны хорошо представляют содержание изображения и при этом меньше по объему данных, чем контуры.

Алгоритмы утончения линии. Процесс утончения – это процесс преобразования изображения, при котором объекты, представляющие собой связные области, подвергаются "эрозии". Результатом такого воздействия в новом объекте, который является подмножеством старого, все пиксели равноудалены от границ исходного объекта. Процесс утончения называют скелетизацией.

Рассмотрим два алгоритма утончения линий: алгоритм Зонга–Суня и параллельно–симметричный алгоритм утончения линии [174, 19].

Алгоритм Зонга–Суня. Контурной точкой считается пиксел, в числе соседей которого хотя бы один имеет нулевое значение.

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

Рисунок 1.2 - Обозначение элементов окрестности алгоритма утончения линии

1 Шаг – контурная точка $P1$ помечается для удаления, если выполнены следующие условия:

- (а) $2 \leq N(p_1) \leq 6$
- (б) $T(p_1) = 1$
- (в) $p_2 \times p_4 \times p_6 = 0$
- (г) $p_4 \times p_6 \times p_8 = 0$

где $N(p_1)$ - число ненулевых соседей элемента p_1 , т.е.

$$N(p_1) = p_2 + p_3 + \dots + p_8 + p_9$$

$T(p_1)$ - число переходов 0-1 в упорядоченной последовательности

$$p_2, p_3, \dots, p_8, p_9, p_2$$

Например, на рисунке 1.3 $N(p_1) = 4$, $T(p_1) = 3$.

0	0	1
---	---	---

1	p_1	0
1	0	1

Рисунок 1.3 - Иллюстрация условий (а) и (б)

2 Шаг – условия (а) и (б) остаются прежними, а условия (в) и (г) заменяются на (д) и (е):

$$(д) \quad p_2 \times p_4 \times p_8 = 0$$

$$(е) \quad p_2 \times p_6 \times p_8 = 0$$

Параллельно–симметричный алгоритм утончения (ПСАУ) [16, 107].

Вводятся следующие понятия:

- левая итерация - однократное сканирование изображения слева-направо, сверху-вниз;

- правая итерация - однократное сканирование изображения справа-налево, сверху-вниз.

Исходное изображение сканируется попеременно с помощью левой и правой итераций. При сканировании используется только входная информация. Сканирование производится до тех пор, пока на всем изображении при двух подряд итерациях не будет изменений.

Рассмотрим сканирование первым способом подробнее. Каждый пиксел выделяется из изображения со своими 15 соседями, как это показано на рисунке 1.4.

a1	a2	a3	a4
a5	P	a6	a7
a8	a9	a10	a11
a12	a13	a14	a15

Рисунок 1.4 - Выделяемые пиксели

Далее, выделенные пиксели сравниваются с матрицами утончения (рисунок 1.5).

0	0	0	x	0	1	x	x	x	1	x	x	x	1	0	x
1	1	1	x	0	1	1	x	1	1	1	x	1	1	0	x
x	1	x	x	0	1	x	x	0	0	0	x	x	1	0	x

x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
(a)				(b)				(c)				(d)			

x	0	0	x	0	0	x	x	x	1	x	x	x	1	x	x
1	1	0	x	0	1	1	x	0	1	1	x	1	1	0	x
x	1	x	x	x	1	x	x	0	0	x	x	x	0	0	x
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
(e)				(f)				(g)				(h)			

Рисунок 1.5 - Матрицы утончения (x - незначащий пиксел)

Если при сравнении нашлась соответствующая матрица, то пиксел P должен быть обнулён. Чтобы сохранить связность, для удаленного пиксела выполняется сравнение с матрицами восстановления (рисунок 1.6).

x	x	x	x	x	0	x	x
0	1	1	0	x	1	0	x
x	x	x	x	x	1	x	x
x	x	x	x	x	0	x	x
(i)				(j)			

Рисунок 1.6 - Матрицы восстановления

В случае, если одна из матриц оказалась соответствующей, то удаленный пиксел восстанавливается, в противном случае он удаляется окончательно. Описанный процесс выполняется над каждым пикселом изображения в порядке слева-направо и сверху-вниз и называется итерацией.

Аналогичный процесс выполняется над изображением в противоположном направлении. В этом случае изображение сканируется справа-налево, сверху-вниз, и соседи рассматриваемого пиксела выделяются в порядке, указанном на рисунке 1.7.

a4	a3	a2	a1
a7	a6	P	a5
a11	a10	a9	a8
a15	a14	a13	a12

Рисунок 1.7 - Выделяемые пиксели "правой" итерации.

Соответственно изменяются матрицы утончения и матрицы восстановления. Итерация в каждом направлении выполняется один раз, поэтому в данном алгоритме они называются левым и правым проходами.

Результатом проходов в обоих направлениях являются два изображения $L(N \times M)$ и $R(N \times M)$, полученные соответственно при правом и левом проходах. Для получения конечного результата выполняется операция логического "И" над изображениями $L(N \times M)$ и $R(N \times M)$. Формально можно записать следующим образом.

$$L(N \times M) = \{l(i,j) \mid i = 1..N, j = 1..M\}$$

$$R(N \times M) = \{r(i,j) \mid i = 1..N, j = 1..M\}.$$

Конечный результат - изображение $S(N \times M)$:

$S(N \times M) = \{s(i, j) \mid i = 1..N, j = 1..M, s(i, j) = l(i, j) \text{ AND } r(i, j)\}$, т.е. $s(i, j) = 1$, если $l(i, j) = 1$ и $r(i, j) = 1$, и $s(i, j) = 0$ во всех остальных случаях. Остаются только те пиксели исходного изображения, которые "выжили" как при левом проходе, так и при правом.

На каждой итерации выполняется "срез" слоя изображения толщиной в один пиксел на границе с фоном. Поочередное использование левой и правой итераций позволяет сохранить симметричность изображения (что и явилось одним из достоинств этого метода) и получить в результате срединные линии исходного изображения. Толщина скелета составит один пиксел, что следует из алгоритма.

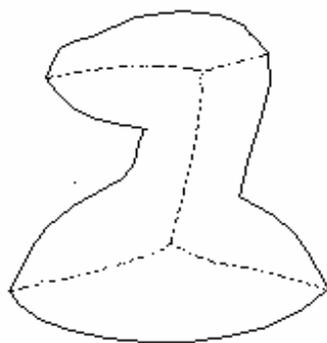


Рисунок 1.8

Диаграммы Вороного [137, 146]. Диаграммы Вороного определяют множество точек, равноудаленных от двух и более элементов множества граничных точек плоской фигуры и представляют собой так называемые «срединные» оси, или оси симметрии, или скелет фигуры. На нем располагаются центры окружностей, которые касаются двух или более граничных точек и не содержат и

не касаются любых других точек. На рисунке 1.8 показан простой, замкнутый полигон со своими срединными осями.

«Диаграммы Вороного для замкнутых форм имеют широкое практическое применение, начиная со станков с числовым программным управлением до генерации сетей связи. В литературе встречается много алгоритмов, которые вычисляют диаграммы для полигональных областей и для областей, ограниченных простыми кривыми типа дуг окружностей. Алгоритмов, применимых для искривленных областей меньше и они довольно сложны. Они требуют той или иной степени дискретности (аппроксимации) области или решения определенной системы нелинейных уравнений» [137].

Пусть граничная кривая $C(s)$ области D аппроксимируется дугами. В описании диаграмм Вороного определяющими элементами $\{e_i\}$ являются точки $C(s)$.

Пусть $C(s)$ построена так, что D всегда оказывается слева, когда мы движемся вдоль кривой. Если граничная кривая дифференцируема за исключением конечного числа точек разрыва, то в этих точках существуют левая и правая производные.

Точки на $C(s)$ с разрывом касательной называются углами: углами отражения являются углы больше 180 градусов, а выпуклыми углами - углы меньше 180 градусов.

Нормаль и кривизна в точке s обозначается как $n(s)$ и $k(s)$ соответственно. Кривизна имеет разные знаки, тогда как нормаль всегда направлена влево от кривой (внутри D). Кривизна имеет положительное значение, когда кривая выпуклая (искривленное направление) и отрицательное значение, когда она отражающая.

Алгоритм: Основная операция вычисляет биссектрису двух участков кривой $C(s)$. Окружность с центром в точке b , лежащей на биссектрисе, касается двух точек $C(s)$, обозначенных $T(b)$, по одной на каждом участке. При этом точка b удовлетворяет следующим условиям:

- 1) b равноудалена от двух точек $T(b)$;

- 2) b лежит на нормалях к $C(s)$ в $T(b)$;
- 3) b находится слева от обеих точек;
- 4) величина, обратная положительному расстоянию от b до точки s , принадлежащей $T(b)$, больше $k(s)$.

Трассировка использует два размера шага. Один - функция от параметра, наращиваемого вдоль граничной кривой, используется в местах, где кривая имеет непрерывную касательную. Другой размер - функция углового приращения угла нормали, используется, когда встречается угол. Значение размера шага может быть разным для каждого шага в применении к локальной форме кривой.

Процедура Step вычисляет одну точку на биссектрисе. В Stepе параметризация $C(s)$ считается периодической, так что $C(s + L) = C(s)$, где L - полная длина дуги $C(s)$.

Чтобы вычертить биссектрису, процедура Step применяется повторно с предыдущими результатами в качестве входных данных, пока либо она не найдет удовлетворительную точку, либо $s_l = s_r$. Во время трассировки поддерживается соответствие информации между точкой биссектрисы и касательными точками. В результате каждой трассировки получается одномерная кривая биссектрисы. Если процесс начинается с одной концевой точки "спина" и движется к другой, отслеженная кривая биссектрисы включает "спин" и может продолжаться дальше [137].

Преобразование изображения к векторному виду. Процедура векторизации преобразует уточненное изображение в кусочно-линейное представление, отбрасывая излишнюю информацию [14].

В ходе настоящего исследования выявлено, что на изображениях низкого качества, например архивные документы (чертежи, схемы, карты) качество обработки неудовлетворительно, поскольку уже процедура бинаризации дает не релевантные результаты на зашумленных участках изображений. Кроме того, методы уточнения линий, применяемые для получения скелетонов, хорошо работают на линейных объектах. Для площадных объектов применяют методы, основанные на диаграммах Вороного, которые требуют значительных

вычислительных ресурсов. На изображениях, которые содержат и тот и другой вид объектов, проблемой становится кооперация методов. Для решения этой проблемы требуется разработка таких методов построения скелетонов, которые бы одинаково эффективно работали в тех и других условиях.

1.2.3. Анализ методов цветовой сегментации изображений

Выделение объектов изображения по цвету – наиболее очевидная, если изображение цветное, операция с целью получения его содержательного описания. Близкими задаче выделения цветных зон изображения можно считать задачи сегментации и квантования.

Методы квантования цвета служат для уменьшения количества цветов цветового пространства.

Простейший способ – постеризация. Строится статичная таблица цветов (Static color look table - SLUT) путем разбиения цветового куба на заданное количество слоев вдоль каждой оси и построения таблицы соответствия цветов. Строится цветовая гистограмма [110, 175], где по оси абсцисс располагаются значения цвета – адреса в упрощенном RGB кубе, а по оси ординат - количество пикселей изображения, имеющих соответствующий цвет.

Более точный способ – вычисление центра масс пикселей для каждого не нулевого элемента гистограммы.

Алгоритм срединного значения (Median cut) итеративно разделяет цветовое пространство между минимальным и максимальным значениями координат цвета на параллелепипеды меньшего размера, пока не будет получено k цветов.

Алгоритм квантования с помощью октадеревьев (Octrees) заключается в построении древовидной структуры, содержащей k – цветов.

Методы сегментации изображений разделяют на интерактивные и автоматические.

Автоматическая сегментация сводится к задаче кластеризации пикселей в трехмерном цветовом пространстве. Чтобы учесть пространственные отношения, в состав признаков вводят координаты точек, либо после кластеризации точек изображения проводят процедуру выделения связанных компонент.

Методы, выполняющие разбиение изображения на цветовые области с помощью кластеризации, используют различные цветовые пространства: RGB, Lab, HSI (Hue, Saturation, Intensity), Luv, HDI (Hue, Distance, Intensity).

Метод поиска моды гистограмм (histogram mode seeking) [98] предполагает, что однородные объекты на изображении образуют кластеры в пространстве измерений, в качестве которого рассматриваются гистограммы. В рекурсивном гистограммном методе Оландера [156] строятся гистограммы для всех компонент цвета и выделяется «лучший» максимум в этом наборе, окруженный минимумами с обеих сторон. Процесс поиска мод проходит рекурсивно.

Сложность заключается в том, чтобы цвета из разных диапазонов хорошо различались человеком. Для этого в алгоритм включаются различные эвристики, которые, помимо усложнения алгоритма, трудно перенастраиваются на различные классы изображений.

Метод среднего сдвига (Mean Shift). Впервые процедура среднего сдвига (Mean Shift) была предложена в 1975 г. [120]. На основе градиента функции рассчитывается вектор среднего сдвига. Он всегда указывает в направлении максимального повышения плотности [111]. Пространство признаков рассматривается как эмпирическая функция плотности вероятности значения параметра. Области с большой плотностью соответствуют локальным максимумам функции. После того, как локальный максимум определен, выделяется связанный с ним кластер.

Отличительные особенности пространства признаков для реального изображения – многомодальность и произвольная форма кластеров. То, что процедура среднего сдвига перемещается к моде склона, на котором она была инициализирована, позволяет успешно производить анализ. Чтобы определить все

значимые моды, алгоритм нужно запустить несколько раз с точками инициализации, случайным образом распределенными по всему пространству признаков (рисунок 1.9).

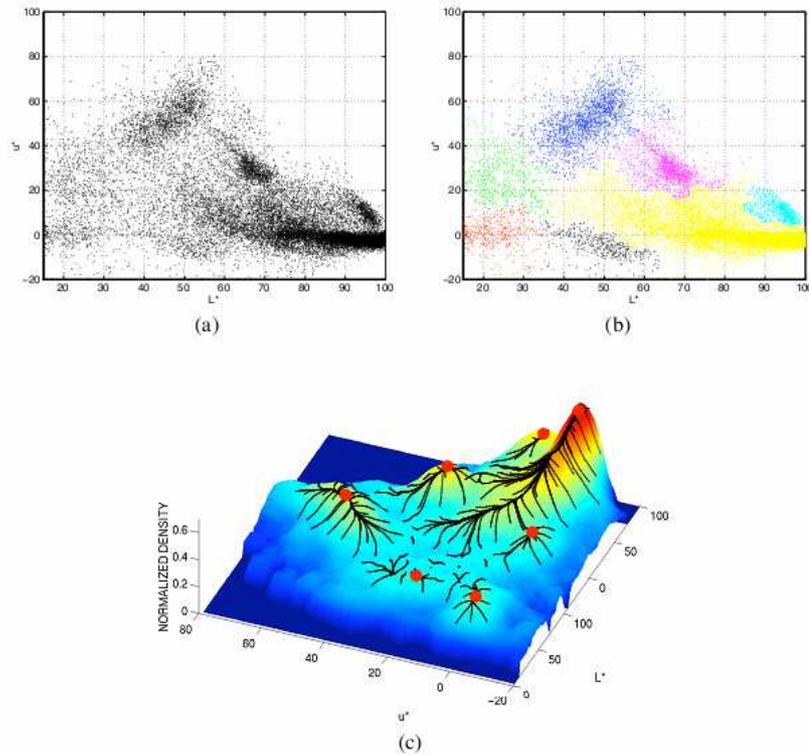


Рисунок 1.9 - Пример анализа двумерного пространства признаков:

- а – двумерный набор данных из 110 400 точек, представляющих первые два компонента цветового пространства $L^*u^*v^*$;
- б – разбиение, полученное после запуска 159 процедур среднего сдвига с различными точками инициализации;
- с – траектории процедур среднего сдвига. Полученные в результате моды отмечены красными точками

Кластеризация методом k -средних (iterative k -means clustering) – один из наиболее распространенных и простых методов кластеризации [98]. Он разбивает множество элементов векторного пространства на заданное число кластеров k . Алгоритм стремится минимизировать среднеквадратичное отклонение на множестве точек каждого кластера

$$D = \sum_{i=1}^k \sum_{x_j \in S_i} (x_j - m_i)^2,$$

где k – число кластеров, S_i – полученные кластеры, m_i – их математические ожидания [98].

Алгоритм:

1. Выбрать центры кластеров случайным образом.
2. Вычислить для каждого пиксела расстояние до каждого центра и приписать к ближайшему.
3. Пересчитать центры кластеров путем усреднения вектора цвета в каждом кластере.
4. Повторить n - раз, пока центры кластеров не перестанут смещаться.

Это многопроходный (итерационный) алгоритм. Недостаток кроется в его достоинстве - сколько задали цветов, столько и получили. Остаются без ответа вопросы: Сколько цветов реально имеется на изображении? Какие цвета имеют изображенные объекты?

Сегментация путем выращивания областей – это группировка пикселей или подобластей по заданным критериям [9].

Сегментация по морфологическим водоразделам [9]. Изображение представляется как трехмерная поверхность, заданная двумя пространственными координатами и уровнем яркости в качестве высоты поверхности (рельефа). Анализируются точки трех видов: а) точки локального минимума; б) точки, находящиеся на склоне у локального минимума; и в) точки, находящиеся на гребне, т.е. с которых "вода с равной вероятностью скатывается более чем в один такой минимум". Множества точек, удовлетворяющих третьему условию образуют линии гребней - водоразделов.

Метод основан на поиске локальных минимумов интенсивности и поиске "холмов" (гребней). Недостатки: метод работает с интенсивностями, то есть изображение из цветного преобразуется в полутоновое; алгоритм очень чувствителен к теням и бликам на изображении. Чтобы устранить чрезмерную сегментацию требуется настраивать пороги, тем самым возникает требование к адаптивности порогов, изменению их в зависимости от контента.

Графовые методы. Строится взвешенный граф $G = \langle V, E \rangle$, где узлы - векторы пикселей в пространстве измерений. Количество узлов равно количеству пикселей изображения. Ребра - веса w_{ij} , характеризующие сходство узлов i и j . Суть алгоритма заключается в нахождении такого разбиения вершин графа на

непересекающиеся подмножества V_1, V_2, \dots, V_k , которое образует высокое сходство цветовых параметров внутри каждого множества и низкое между элементами из разных множеств. Далее внутри кластеров выделяются связные области [172].

Степень сходства множеств A и B рассчитывается как общий вес удаленных ребер между ними – вес разреза.

Рассмотрим некоторые графовые методы подробнее [4].

Метод нормализованного разреза (Normalized cut) предложен J. Shi, J. Malik (1997) [163]. Вводится нормализованный функционал качества разреза так, чтобы одновременно максимизировать различие точек между классами и минимизировать различия точек внутри класса. Оптимизация нормализованного функционала сводится к задаче поиска собственных значений матрицы парных расстояний между всеми точками изображения. Для сегментации изображения на две части достаточно найти второе по величине собственное значение такой матрицы.

Сложность алгоритма поиска собственных значений разреженной матрицы линейна по числу точек изображения. Однако при каждой сегментации на две части метод требует хранения матрицы размером $n \times n$, где n – число точек изображения, и потому в исходном виде неприменим к большим изображениям.

Для данного метода предложены модификации, позволяющие сократить сложность алгоритма и требования по памяти за счет аппроксимации матрицы расстояний. Такой подход дает выигрыш в скорости работы в 10-20 раз по сравнению с исходным методом.

Поиск локальных максимумов на трехмерной гистограмме с помощью оператора Собеля. В работе [164] осуществляется квантование векторного пространства многомерной гистограммы, строится список векторов (подпространств) по типу словаря, исключая нулевые векторы, и применяется гистограммный алгоритм Нарендры [154], который не требует задания числа кластеров или количества итераций, а время классификации линейно зависит от числа векторов. При этом строится множество локальных графов, в каждом графе

вектор соединен с соседними векторами. Ребра графов содержат градиенты плотностей (длин векторов). Алгоритм применялся для обработки многоспектральных спутниковых изображений.

Таким образом, для осуществления цветовой сегментации изображений чаще всего применяют кластеризацию пикселей по цвету:

- рекурсивный гистограммный метод Оландера;
- кластеризацию методом k - средних с итерационным уточнением центров кластеров.

В целом, методы цветовой сегментации характеризуются высокой вычислительной сложностью, несмотря на использование методов динамического программирования, "жадного" поиска и др.

1.2.4. Анализ методов сопоставления графов

Общий случай задачи сопоставления графов проработан в проблеме понимания и оценки ситуаций. В работе Adam Stotz и др. *Incremental Graph Matching for Situation Awareness* [168] ситуация представляется в виде ориентированного атрибутивного графа отношений (DARG – Directed Attributed Relational Graph), которые представляют собой простые графы, содержащие множества атрибутов вершин и ребер. DARG-граф G обозначается как $G = \{V, E, AV, AE\}$, где V – множество вершин, E – множество ребер, AV – конечное множество атрибутов вершин, AE – конечное множество атрибутов ребер.

Сопоставление графов – это процедура поиска вхождений образа, или графа шаблона в обычно больший по размеру предъявленный граф. Поиск вхождений графа шаблона $G1 = \{V1, E1, AV1, AE1\}$ в граф данных $G2 = \{V2, E2, AV2, AE2\}$ может быть представлен как поиск подграфа $G1$ в графе $G2$. Говорят, что $G1$ является подграфом $G2$ тогда и только тогда, когда $V1 \subseteq V2, E1 \subseteq E2, AV1 \subseteq AV2, AE1 \subseteq AE2$. Такой вид сопоставления называется изоморфизмом подграфов

[102]. Изоморфизм позволяет находить топологически эквивалентные вхождения образов в графы, в отличие от гомеоморфизма, который находит семантически эквивалентные вхождения, которые при этом могут не быть топологически эквивалентными [167].

Методы сопоставления графов разделяются на две категории: (1) методы точного сопоставления графов, которые требуют строгого соответствия между их элементами, и (2) методы неточного сопоставления, позволяющие сравнивать графы даже в случае семантических и топологических различий между ними.

Точное сопоставление определяется как сопоставление узлов двух графов с сохранением ребер. Это означает, что два смежных узла одного графа также должны быть смежны во втором графе. Наиболее строгая форма точного сопоставления графов – изоморфизм графов. В этом случае имеет место взаимно однозначное соответствие между узлами и ребрами обоих графов. Несколько менее строгая форма точного сопоставления – изоморфизм подграфов [102]. В этом случае подграф одного из графов проверяется на изоморфизм со вторым графом. Если условие сюръекции не выполняется, имеет место мономорфизм, при котором соответствуют друг другу узлы графов, но соответствие по ребрам не обязательно. В еще менее строгой форме сопоставления – гомоморфизме – отбрасывается условие, что узлы первого графа должны быть сопоставлены отдельным узлам второго графа. Все указанные выше проблемы являются NP-полными, за исключением случая атрибутного графа, в котором узлы гарантированно обладают отдельными атрибутами, и точное сопоставление графов выполняется за полиномиальное время.

При неточном сопоставлении графов [121] некоторые строгие условия, налагаемые при точном сопоставлении, ослабляются с целью адаптации к помехам в системе. В этом случае два узла, не удовлетворяющие условию сохранения ребер, могут быть сопоставлены, но с определенной стоимостью. Также в случае атрибутного графа два узла могут быть сопоставлены, даже если $AV_1 \neq AV_2$. Алгоритм неточного сопоставления графов работает с целевой функцией минимизации стоимости сопоставления. Если точное решение

существует, оно всегда будет найдено с помощью алгоритма оптимизации для неточного сопоставления.

Задача сопоставления представляется как пошаговый поиск пути в пространстве состояний, в результате которого строится дерево переходов. Алгоритм аналогичен лучевому поиску. На каждом уровне как и лучевой поиск, алгоритм на основе усеченных деревьев поиска ограничен в пространстве, будучи функцией от x , где x – фиксированное значение «ширины луча». На каждом уровне дерева поиска эвристическая процедура образует новый уровень в глубину, создавая разветвление только из первых x наиболее перспективных узлов, где x - фиксированное значение «ширины» луча [116].

Следует отметить, что данный подход не учитывает топологию элементов ситуации, что очень важно в задаче понимания и поиска изображений.

Интерес также представляют методы и алгоритмы визуализации графов [64] (рисунок 1.10).

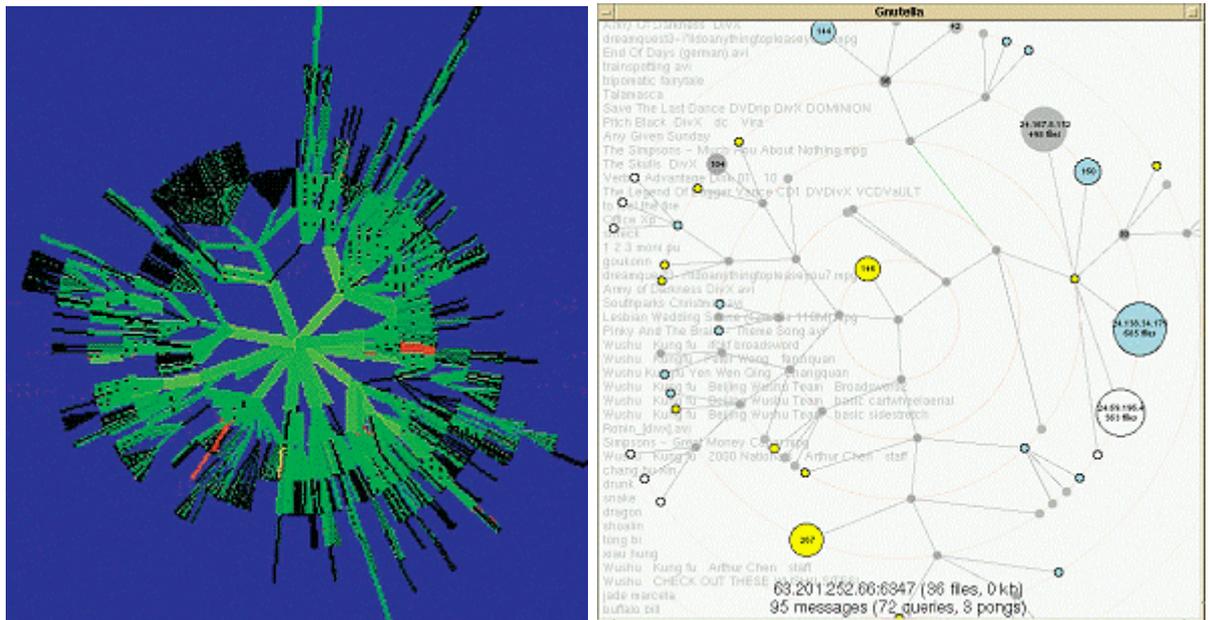


Рисунок 1.10 - Радиальные и круговые изображения деревьев

Радиальный алгоритм является весьма популярным инструментом визуализации не только иерархических структур, но и графов, в особенности, динамических. Этим он обязан, прежде всего, своей линейной сложности. Одна из первых широко известных реализаций радиального алгоритма была осуществлена системе NicheWorks, разработанной Bell Laboratories в 1998 году для анализа

телефонных звонков и выявления шаблонов для обнаружения телефонных мошенничеств. Каждое ребро графа в этой системе соответствовало одному телефонному вызову, и ширина ребра возрастала по мере увеличения количества звонков между абонентами. Произвольная вершина могла быть выбрана в качестве корня, после чего строилось остовное дерево путем поиска в ширину. Изображаемые деревья имели сотни тысяч вершин, а сама система позволила найти реальных мошенников.

Несложно заметить, что основной целью перестроения в радиальный граф является визуализация, когда вершины располагаются на окружностях определенного радиуса от центральной. В некоторых задачах применяется лучевой поиск по полученному графу.

В качестве описания изображений в задачах поиска в последнее время стали применяться атрибутивные графы отношений (ARG), нечеткие атрибутивные графы (FARG), дескрипционные логики. Однако задачи семантического анализа изображений объектов, выявления сходства и отличий при этом не ставились. Соответственно, описания менее структурированы и содержат ограниченный набор атрибутов формы.

1.3. Анализ существующих подходов и технологий поиска графической информации в автоматизированных системах

Современные OCR-системы начинают распознавание с анализа структуры. Как правило, при этом выделяют несколько иерархически организованных логических уровней. Объект наивысшего уровня только один – собственно страница, на следующей ступени иерархии располагаются таблица и текстовый блок, затем ячейка таблицы, абзац или картинка, за ними следует строка, потом слово или картинка внутри строки и, наконец, нижний уровень – символ.

На первом этапе системы OCR разбивают страницы на блоки текста, основываясь на особенностях правого и левого выравнивания и наличия

нескольких колонок. Затем распознанные блоки разбиваются на строки. На втором этапе строки разбиваются на непрерывные области изображения, которые, как правило, соответствуют отдельным буквам. Алгоритм распознавания делает предположения относительно соответствия этих областей символам, а затем делается выбор каждого символа, в результате чего страница восстанавливается в символах текста, причем, как правило, в соответствующем формате. Недостатком систем распознавания зрительных образов считается их узкая специализация - "заточенность" на конкретный класс объектов.

В течении ряда лет, в Вычислительном центре РАН (г. Москва) под руководством И.Б. Гуревича на основе алгебраической теории алгоритмов Ю.И. Журавлева и дескриптивного подхода И.Б. Гуревича разрабатывается алгоритмическо-программная система для автоматизации получения информации из изображений и ее анализа в виде набора инструментальных средств для автоматизации разработки новых алгоритмов обработки, анализа, распознавания и понимания изображений [79].

В настоящее время под руководством И.Б. Гуревича разработана экспериментальная версия тезауруса и онтологии средств анализа изображений (ОАИ), базирующаяся на подходе и результатах научной школы академика Ю.И. Журавлева. ОАИ включает в себя два основных компонента: онтологию задач и онтологию методов решения этих задач. ОАИ регламентирует тезаурус и основные понятия в области теории, математических методов и алгоритмов обработки, распознавания образов и анализа изображений, содержит более 900 словарных статей и оформлена в соответствии с существующими российскими и международными стандартами. ОАИ предназначена для разработчиков программного обеспечения, СППР, а также как справочная система. И то, и другое безусловно необходимо для дальнейшего развития систем анализа изображений.

Дескрипционные алгебры изображений описывают изображение в терминах дескрипторов – результатов вычисления соответствующих функций, либо

распознают объекты как целостные образы, но не осуществляют семантический анализ изображенных объектов (И.Б. Гуревич, Ю.И. Журавлев, И.В. Корябкина, В.В. Яшина). Для описания изображений обычно используют глобальные и локальные дескрипторы.

С помощью глобальных дескрипторов описываются цветовые характеристики изображения, получаемые гистограммными методами, и текстурные, описывающие структуру объектов на изображении.

Локальные дескрипторы описывают выбранные некоторым способом фрагменты изображения [3]. Наиболее распространены дескрипторы Scale Invariant Feature Transform [151], Speeded Up Robust Features [105], Rotation Invariant Fast Features [141], инвариантные к преобразованиям масштаба, поворота, шуму и освещению.

Подтверждением того, что анализ графической информации подчиняется общим принципам системного анализа, можно считать исследование Симанкова В.С., Луценко Е.В., Лаптева В.Н. [83], где предложена оригинальная технология синтеза адаптивных автоматизированных систем управления сложными системами, основанная на методах распознавания образов и принятия решений. Описана инструментальная программная оболочка "Эйдос", в которой реализована предложенная технология.

Поиск изображений по содержанию (CBIR - Content-Based Image Retrieval). Проблема содержательного поиска изображений и видеоинформации решается также в Институте прикладной математики имени М.В. Келдыша РАН. Разрабатываемые здесь подходы применяют различные средства, в т.ч., использующие текстовые описания и характеристики визуального содержания. Содержание изображения характеризуется визуальными примитивами [71, 133, 134, 135] такими, как: цветовая гамма, характеристики текстуры, параметры форм для контуров выделенных объектов. Авторы Байгарова и др. [69] обращают внимание на тот факт, что методы распознавания образов и понимания сцены в

настоящее время применяются в узких предметных областях из-за отсутствия эффективных универсальных алгоритмов.

Другой проблемой СВIR систем (вытекающей из первой) является ограниченность набора характеристик, используемых для поиска объектов, что приводит к недостаточной релевантности результатов поиска и избыточной выдаче, которую лицу принимающему решение, приходится тщательно пересматривать.

Испанские ученые [176] выделяют на изображении цветовые области, формируют их описание (форма, цвет) и пространственные характеристики (ориентация, топология/положение), а потом это описание переводят в дескрипционные логики с целью получения формального и явного смысла и классификации с помощью дескрипционных логик рассуждений. Для описания концептов в дескрипционной логике в качестве атомарных они берут эти характеристики областей изображения, т.е. алгоритмы вычисления характеристик областей есть интерпретация соответствующих атомарных концептов и ролей. В качестве примера анализируют ситуации, возникающие перед роботом, ориентирующимся в пространстве коридоров и комнат. Недостатком работы следует считать ограниченный набор характеристик изображения.

Ученые [127] более доказательно прописывают предложенную ими описательную логику, названную $ALC(F)$. Они описывают структуру объектов на изображении с помощью логики ALC , для которой ввели специальную, пространственную область в виде набора функций, вычисляющих пространственные отношения между совокупностями пикселей. Доказывают свойства предложенного языка пространственных рассуждений $ALC(F)$: допустимость, разрешимость, выполнимость и т.п. Авторы основываются на своей работе [128], где они описали онтологию пространственных отношений, используя математическую морфологию [161]. Основываясь на учебниках по анатомии, они создали набор функций, интерпретируемых на медицинских изображениях.

Таким образом, общий принцип применения дескрипционных логик для описания изображений заключается в создании специализированных наборов атомарных концептов и ролей. Несмотря на трудность функциональной интерпретации понятий в области графической информации с помощью классической или нечеткой логики предикатов, в дальнейшем, при создании предметных онтологий с помощью дескрипционных логик затраченные усилия многократно окупаются.

Графический поиск чертежей. В последние два десятилетия активизировалась исследовательская деятельность, направленная на создание методов и систем поиска чертежей путем сопоставления контуров деталей для определения общего структурного сходства, устойчивого к местным изгибам и растяжениям [122]. Для этого выделяется множество особых точек с их локальной геометрией и группы элементов между ними. Строятся таблицы поиска.

С целью получения особых точек осуществляется итерационное сглаживание контура на основе метода дискретной эволюции контура (discrete contour evolution) [147]. Локальная геометрия особой точки включает угол поворота (\pm показывает выпуклость/вогнутость) [103] и контекст фигуры (базисная ось, проходящая через особую точку и центральную точку фигуры) [106]. Эти две характеристики инвариантны к переносу, повороту, масштабу.

Сопоставление контуров включает:

- нахождение всех пар особых точек ($f_k, f_{k'}$), схожих в их локальной геометрии (по этим парам точек определяется также глобальный поворот одного контура относительно другого);

- установление вариантов соответствия структуры контуров (особых точек), начиная из каждой пары ($f_k, f_{k'}$), с помощью алгоритма ослабленной динамической трансформации шкалы времени (relaxed Dynamic Time Warping);

- сравнение геометрии контуров (групп геометрических элементов) на каждом варианте соответствия структуры контуров путем эластичного сопоставления (Elastic Matching);

- определение оптимального варианта соответствия структуры контуров.

Авторы указывают, что:

- предложенная технология сопоставления работает хорошо, когда два контура имеют большие порции структурного подобия, даже если имеются локальные геометрические деформации и локальные структурные отличия;

- производительность процесса установления соответствия зависит от результатов этапа выделения особых точек: когда фигура имеет большое количество особых точек, требуется относительно большое вычислительное время для осуществления сопоставления, при этом структурное представление становится очень сильно связанным с геометрией, поскольку доминируют мелкие детали.

В работе [119] представлена методика поиска векторных чертежей по запросу в виде эскиза. Информация чертежа раскладывается на топологию (пространственное расположение полигонов, имеющих на чертеже) и геометрию (форма этих полигонов).

Топологическая информация организуется в виде топологического графа, в котором узлы представляют полигоны чертежа, а ребра – отношения между ними. Граф преобразуется в многомерные дескрипторы, чтобы решать задачу путем вычисления расстояний между векторами дескрипторов. При этом используется три топологических отношения между полигонами: 1) вложенность; 2) смежность; 3) расстояние между полигонами.

Преобразование топологического графа в топологический дескриптор осуществляется на основе вычисления спектра графа [112]: построение матрицы смежности графа, вычисление ее собственных значений, упорядочивание абсолютных значений по убыванию.

Геометрическая информация о полигоне представляется в виде геометрического дескриптора и включает: площадь и периметр выпуклой

оболочки, площадь, периметр, высоту и ширину объемлющего прямоугольника, площадь и периметр наибольшего четырехугольника и треугольника, периметр исходного полигона.

Подход применялся в задачах поиска двумерных технических чертежей пресс-форм, поиска клипартов, а также (с некоторой модификацией) для неявного поиска 3D объектов с целью их предложения пользователю в процессе 3D моделирования.

Эксперименты с прототипом системы, проведенные автором методики на базах данных, содержащих 100 и 968 чертежей, показали эффективные результаты за короткое время поиска (менее чем за 1 секунду).

К недостаткам следует отнести: отсутствие возможности визуализации совпавших элементов; необходимость построения отдельного дескриптора каждого участка каждого чертежа для возможности поиска чертежей с заданным фрагментом.

В работе [150] предлагается метод поиска технических чертежей на основе сопоставления атрибутивных графов. Узлы графов соответствуют значимым примитивам (отрезкам и кривым), а ребра описывают пространственные отношения между ними. При построении графа чертежа применяется модификация метода тесселяции Делоне, с помощью которого пространственные отношения между примитивами представляются естественно и содержательно. Указывается, что традиционный метод Делоне приводит к потере важных отношений и возникновению ложных связей. Предложенная авторами модификация метода Делоне, названная стратегией выборки, заключается в следующем:

1. Равномерный выбор нескольких точек на длинном примитиве.
2. Использование выбранных точек в качестве входных данных для тесселяции Делоне.
3. Упрощение графа: слияние узлов и ребер графа, полученных из одного примитива.

Статистика экспериментов следующая: на компьютере P4-1.3G система просматривает 1 чертеж в среднем за 1.1 секунды, при этом средняя точность поиска составляет около 56%, а релевантность отзыва – 80%.

В работе [136] предложен метод поиска растровых чертежей, не требующий перевода их в векторный вид. Метод заключается в представлении чертежей векторами характеристик блоков пикселей и вычислении их подобия линейно-взвешенным методом косинусного подобия. Используются характеристики двух видов: признак плотности (степень плотности черных пикселей, т.е. пропорция черных пикселей среди всех пикселей блока); инвариантные моменты [123], улучшенные авторами с целью их независимости от переноса, поворота, масштаба (в дискретном случае).

Технология не поддерживает поиск чертежей с заданным фрагментом, требует перевода векторных изображений в растровый формат.

В работе [131] рассматриваются два альтернативных метода поиска технических чертежей в базе данных патентов: путем измерения подобия графов и путем сравнения гистограмм. Оба метода предполагают преобразование множества линий чертежа в направленный граф 6-ти ближайших (по евклидовому расстоянию центров линий) соседей, ребра которого нагружены инвариантными атрибутами: относительный угол и относительное расположение двух линий. Атрибуты ребер графа используются также при построении двумерных гистограмм. Сравнение графов осуществляется с использованием варианта расстояния Хаусдорфа [129]. Сравнение гистограмм осуществляется путем измерения корреляции Бхаттачарья [130].

Авторы делают следующие выводы: показатели отзыва во всех случаях выше с использованием сравнения графов; за лучшую производительность алгоритмов, основанных на графах, приходится расплачиваться временем поиска, напрямую зависящим от размера и сложности сравниваемых изображений.

Векторизация сканированных чертежей. Актуальность задачи векторизации чертежей:

- Конструкторские бюро заводов работают, в основном, над модификацией изделий, запущенных в производство.
- Колоссальные знания «пылятся» на бумажных носителях в пассивном состоянии.
- Лишь небольшая часть документации сканирована в различные растровые форматы. Еще меньшая часть переведена в векторную форму.
- Перенос чертежей с бумажных носителей на электронные требует высоких трудозатрат.

Из схемы движения информации чертежа в задаче создания электронного конструкторско-технологического архива (рисунок 1.11) видно, что к модулю векторизации здесь предъявляются очень высокие требования по качеству: минимизация количества отрезков прямых и дуг, минимум настроек в векторизаторе, которые необходимо сделать пользователю для получения приемлемого результата на множестве конкретных чертежей. Все это в условиях, когда качество исходных бумажных документов может быть низким, чертежи представлены на ксерокопиях, кальках, синьках или выполнены в карандаше.



Рисунок 1.11 - Схема обработки чертежной информации при создании электронного архива.

На данный момент разработано достаточно много подходов к этой проблеме, однако то качество автоматической векторизации, которое требуется для решения реальных производственных задач, не достигнуто.

К особенностям реальных чертежных изображений следует отнести: нечетко прорисованные линии разной толщины и качества вычерчивания; штриховые, штрихпунктирные, мелкие объекты (цифры, буквы, знаки); некачественная бумага (калька, синька); пятна и потертости. Характеристики

наиболее распространенных векторизаторов Scan2CAD, GTXRaster, Spotlight, WinTopo (Таблица 1.1.):

- базируются на бинарной сегментации чертежного изображения;
- недостаточно успешно обрабатывают места сближения линий, области перекрестков и острых углов;
- слабо развит высокоуровневый анализ чертежного изображения (чтение чертежа).

Таблица 1.1.

Сопоставление функциональности векторизаторов

	Необх-ть бинаризации полутоновых /цветных изображений	Необх-ть ручной доработки	Оценка качества векторизации	Аппроксимация дугами	Аппроксимация окружностями	Время, затрачиваемое на доработку	Удобство пользования интерфейса	Цена 1 места (на 2013 г.)
Scan2CAD v8	Да	Не обязательно	Удовлетворительно	Да	Да	от 5 минут	Простой	250 \$
GTXRaster	Да	Не обязательно	Удовлетворительно	Да	Да	от 5 минут	Сложный	Определяется индивидуально
Spotlight Pro 10.0	Да	Обязательно	Средняя	Нет	Нет	от 20 минут	Сложный	3125 \$
WinTopo v3.5	Да	Обязательно	Низкая	Нет	Нет	от 30 минут	Простой	340 \$

Анализ и дешифрирование аэрокосмоснимков. Дешифрирование – это метод изучения и картографирования местности по ее изображению, который заключается в поиске, обнаружении и распознавании объектов местности на изображениях, установлении по изображению их качественных и количественных характеристик и условном обозначении дешифрируемых объектов на карте или плане [7]. Интерпретация изображений производится на основе известных зависимостей между свойствами дешифрируемых объектов – дешифровочных признаков. Различают прямые дешифровочные признаки, непосредственно воспринимаемые глазом (размер, форма, тень, тон, цвет, структура), и косвенные

признаки, характеризующие объект опосредованно через свойства других объектов.

Классы объектов: площадные, линейные, точечные.

Классы местности: одноцветные, текстурные.

Дешифровочные признаки:

- прямые дешифровочные признаки;
- косвенные дешифровочные признаки.

Средства автоматизированного дешифрирования (ERDAS Imagine, модуль ArcGIS Spatial Analyst, Комплекс автоматизированного дешифрирования и векторизации, eCognition и др.) представлены инструментами классификации пикселей или групп пикселей растрового изображения. Отнесение каждого пикселя к определенному классу выполняется на основе сходства значений интенсивности, статистических текстурных характеристик и др. в различных спектральных каналах. При классификации (с учителем либо без учителя) применяются метод опорных векторов, деревья решений, кластеризация с помощью алгоритма ISODATA и др. Обязательно выполняется ручное редактирование и доработка полученных классов. Затем изображение векторизуется и производится редактирование полученных контуров.

1.4. Сравнительные характеристики подходов, методов и технологий обработки графической информации

В таблице 1.2. приведены основные характеристики подходов к распознаванию и анализу графической информации.

Таблица 1.2.

Анализ подходов
к распознаванию и анализу графической информации

Подход	Область применения	Достоинства	Недостатки
1. Структурно-лингвистический	Распознавание объектов заданной структуры (буквы, символы и т.п.)	Высокая надёжность распознавания	Узкая специализация методов

2. Искусственные нейронные сети	Распознавание, классификация	Широкий спектр задач с не дискретным входным сигналом, эффективность, хорошее качество	Необходимость длительного обучения, непредсказуемость обучения, непрозрачность принимаемого решения, ресурсоёмкость
3. Deskрипционные алгебры изображений	Комплексирование специализированных (задачно-ориентированных) методик и технологий	Универсальность, накопление знаний о методах, тестирующих наборов данных	При необходимости создания новых функций для новых прикладных задач, требуется привлекать математиков и программистов
4. Контекстный поиск с помощью ДЛ	Поисковые системы	Когнитивность	Ограниченность и субъективизм набора характеристик
5. Объектно-независимый подход к анализу изображений	Изображения объектов макро и микро уровня	Универсальность	Трудоёмкость создания набора детекторов для различных компактных подструктур и интерпретации понятий
6. Эвристический подход	Экспертные системы анализа графической информации	Универсальность	Зависимость наборов экспертных правил от предметной области, ошибки при возникновении неучтённых ситуаций и трудности контроля непротиворечивости

Анализ методов обработки и анализа графической информации показан в таблице 1.3.

Таблица 1.3.

Анализ методов обработки и анализа графической информации

Методы	Назначение	Достоинства	Недостатки
Операторы Собеля и аналогичные	Выделение контуров, утончение линий	Простота, быстрое действие	Двухцветные изображения
Диаграммы Вороного	Выделение скелетона	Надежность, качество.	Большие вычислительные затраты
Аппроксимация кривых	Сокращение объёма избыточной информации для сопоставления и поиска	Компактность описания	Для А. прямыми и дугами: Большое количество отрезков, плохое сглаживание. Для А. сплайнами: Не точность описания исходной информации, Большие вычислительные затраты. Сложность

			последующего анализа.
Методы «k-средних», графовые, гистограммные, «выравнивание» областей, сегментация по морфологическим водоразделам	Цветовая сегментация	Выделение компактных областей объектов по цвету.	Недостаточная релевантность выделяемых объектов, сложность и плохая управляемость процессом
Порождающие грамматики	Распознавание букв, знаков, фигур	Высокая надёжность распознавания	Для каждой задачи требуется разрабатывать свои грамматики; высокие требования к качеству изображений; сложность семантического анализа
Атрибутивные графы отношений (ARG)	Представление изображений	Удобство анализа	Ограниченность и субъективность признаков, предопределённость и ограниченность функций анализа
Нечёткие графы отношений (FARG)	Представление изображений	Удобство анализа, наглядность	Ограниченность и субъективность признаков, предопределённость и ограниченность функций анализа
Сопоставление атрибутивных графов	Поиск, выявление сходства и отличий	Хорошая релевантность	Большие вычислительные затраты (задача <i>NP</i> сложности)
Дескрипционные логики	Анализ и описание содержания изображений	Формализация семантики изображений, контроль корректности терминологий, лог. вывод при поиске экземпляров	Зависимость от полноты методов выделения атомарных концептов и отношений - первичных признаков объектов

Таблица 1.4.

Анализ технологий
и инструментальных средств обработки и анализа изображений

Технологии	Назначение	Достоинства	Недостатки
OCR - оптическое распознавание символов	Распознавание букв, знаков, фигур	Быстродействие, надёжность	Узкая специализация: для каждого символа своя грамматика
ДАИ - дескрипционные алгебры изображений	Комплексирование систем обработки изображений и распознавания образов	Гибкость, описание информации в виде результатов вычисления функций дескрипторов	Распознавание объектов как целостных образов без структурных отношений
CBIR - поиск объектов по содержанию	Поисковые системы	Когнитивность (естественность)	Ограниченность и субъективность набора

		описаний	характеристик, недостаточная степень детализации объектов
Векторизация	Создание электронных архивов	Автоматизация процесса перевода из растрового в векторный формат	Падение надёжности при низком качестве носителей, излишнее количество геометрических примитивов (отрезков), значительный процент ручной доработки
Графический поиск чертежей по образцу	Сокращение сроков проектирования новых изделий, автоматизация патентных исследований	Заимствование конструкторских и технологических наработок	Недостаточная релевантность, отсутствие возможности диалога для уточнения запроса, сложность выявления совпадающих элементов
Дешифрирование аэрокосмоснимков	Геоинформационные системы	Описание объектов на макроуровне	Ограниченность и субъективность набора характеристик, большая доля ручного труда
Инструментальные средства - открытые библиотеки графических функций и процедур	Создание задачно-ориентированных технологий	Готовый набор инструментов для обработки и выделения первичных информативных признаков	Все недостатки существующих методов, отсутствие адаптации к конкретным предметным областям

1.5. Выводы, цель и задачи исследования

Несмотря на бурное развитие технических и программных средств вычислительной техники, именно анализ графической информации или ее описание на ограниченном естественном языке в большинстве предметных областей науки и техники осуществляется с участием человека, в лучшем случае, в интерактивном (диалоговом) режиме. Поэтому вопросы дальнейшей автоматизации обработки и интеллектуального анализа изображений остаются весьма актуальными.

Выявлено, что на изображениях низкого качества, например, на архивных документах (чертежи, схемы, карты) качество обработки неудовлетворительно. С другой стороны, методы утончения линий, применяемые для получения скелетонов, хорошо работают на линейных объектах. Для площадных объектов применяют методы, основанные на диаграммах Вороного, которые требуют значительных вычислительных ресурсов. На изображениях, которые содержат и тот и другой вид объектов, проблемой становится кооперация методов. Для решения этой проблемы требуется разработка таких методов построения скелетонов, которые бы одинаково эффективно работали в тех и других условиях.

Всё более и более востребованной становится задача поиска изображений в хранилищах данных и сети Интернет. На предприятиях проблема поиска чертежей и трехмерных моделей машиностроительных деталей главным образом решается путем анализа кодов классификации и текстовых описаний, создаваемых человеком. Уже существуют коммерческие и бесплатные системы поиска, использующие методы распознавания образов, но эти средства по релевантности еще далеки от желаемого, поскольку в качестве поискового образа берется совокупность контурных точек или дескрипторов, что позволяет находить только изображения близких по форме объектов.

Задачи анализа и извлечения семантики из изображений до сих пор ставились лишь в рамках проблемы содержательного поиска изображений. В остальных задачах в явном виде они пока не ставились. Возможно потому, что требуется более детализированное, иерархическое описание структур изображенных объектов – природных или искусственных в зависимости от прикладной области.

Как правило, существующие методы ориентированы на поиск или распознавание изображений (отнесение к какому либо классу). Но для описания этого недостаточно. Требуется выделить на изображении объекты разных типов (в ГИС это называется "тематические слои"), анализировать любой из объектов и его составляющие, вычислять требуемые параметры объектов, элементов и отношений.

Общим недостатком подходов и методов анализа и поиска графической информации является методологический разрыв между морфологией (набором пикселей) и семантикой структуры объекта, что делает затруднительным семантический анализ изображенных объектов и динамики их изменения. Особенно это проявляется, когда яркостные переходы плавные или качество съёмки неудовлетворительное.

Существуют как бы отдельно методы повышения качества изображений - для визуального анализа человеком и методы извлечения информативных элементов - для распознавания образов и поиска. Нужны методы анализа, выделения и синтеза первичных информативных элементов, специально разработанные для содержательного описания. Тогда повысится и эффективность систем контекстного поиска и описания графической информации.

Исходя из вышесказанного, цель работы – разработка методологии семантического анализа и поиска графической информации для повышения релевантности результатов семантического анализа, поиска и классификации графической информации в САПР и ГИС путем разработки новых моделей, методов и алгоритмов на основе семиотического подхода к организации знаковых коммуникативных систем.

Для достижения поставленной цели требуется решение следующих **задач**:

- разработка концептуальной модели и методологии семантического анализа и поиска графической информации;

- разработка методов автоматического синтеза синтаксических моделей графической информации;

- разработка новых методов поиска, анализа и сопоставления графической информации, выявления сходства и отличий для анализа конструкторских и технологических проектных решений.

При этом быстродействие того или иного метода в данном исследовании не ставится во «главу угла». В идеале, хотелось бы иметь такие методы, которые обрабатывали изображения не хуже, чем это делает человек.

2. КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ СЕМАНТИКИ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

В главе представлена разработанная в ходе данного исследования концептуальная модель процесса обработки, семантического анализа и поиска графической информации, её методологические и теоретические основания, структура методологии и требования к формализованным представлениям.

2.1. Задачи анализа изображений

Значимость проблемы обработки и анализа графической информации заключается в том, что большую часть объема информации, обрабатываемой человеком в процессе исследовательской, проектно-конструкторской и управленческой деятельности, составляет графическая информация, представленная в виде фотоизображений, рисунков, эскизов, чертежей. Но и сейчас, несмотря на развитие технических и программных средств вычислительной техники, именно анализ графической информации и ее описание на естественном или формализованном языке в большинстве предметных областей науки и техники осуществляется с участием человека, в лучшем случае, в интерактивном (диалоговом) режиме.

Причиной этого является слабая структурированность изображений и далеко не всегда удовлетворительное качество исходной информации. В то же время, человек успешно справляется с большинством задач анализа и распознавания. Естественно, с соответствующими затратами ручного труда и большой долей субъективности результатов.

Традиционно задачи обработки изображений направлены на решение трех проблем: 1) передача и повышение качества; 2) распознавание образов; 3) анализ изображений.

Задачи распознавания образов (укрупнено) – это вычисление меры близости между точками в пространстве параметров (байесовский подход), либо структурно-лингвистический подход – сравнение с эталонами, заданными в виде грамматик для составляющих элементов. Однако, до сих пор остается проблематичным, например, распознавание слитного рукописного текста, хотя здесь всего три основных элемента: палочка, крючок, петля. Но очень большое разнообразие форм элементов, требующее соответствующего количества эталонов.

Анализ изображений, по сути, отличается от распознавания образов как логическое мышление от образного. Анализ изображений это проблема моделирования рассуждений на графической информации, для чего требуется извлечение формального описания из изображения, привлечение методов логического вывода и сопоставления описаний, т.е. развитие методов анализа структур и, тем самым, Image Mining – понимания изображений компьютером.

Очевидно, алгоритмы анализа изображений зависят от качества изображений и от природы объектов, на них изображенных. Приведенная ниже классификация не претендует на полноту, ее назначение – показать, какие основные факторы влияют на сложность алгоритмов анализа.

Виды объектов на изображениях:

По происхождению: природные; искусственные.

По цвету: цветные; монохромные.

По размерности: плоские - на плоскости носителя изображения или на плоскости в пространстве; объемные (с плоскими гранями; ограниченные поверхностями не выше 2-го порядка типа цилиндр, конус, шар, тор, плоскость); скульптурные, содержащие сплайновые поверхности.

По форме: линейные; площадные; смешанные; текстурные (стохастические или фрактальные).

Качество изображения (помехи):

Естественные: возраст носителя; качество носителя или источника цифрового изображения; освещенность; тени; блики на поверхности объекта.

Искусственные (наносные), более позднего происхождения чем объект: пятна; штрихи; потертости; разрывы носителя.

Исходная информация: цвета, яркости.

В числе наиболее распространенных задач, которые приходится решать на информации, представленной в виде изображений, можно перечислить следующие:

1. Повышение качества изображения с целью улучшения их зрительного восприятия.
2. Распознавание объектов, представленных на изображениях, с целью их идентификации.
3. Анализ объектов по их изображениям с целью измерения и исследования параметров объектов и выявления закономерностей:
 - 3.1. Статистический анализ.
 - 3.2. Анализ зависимостей между параметрами, в том числе на серии изображений.
4. Аппроксимация (упрощение) изображений с целью сокращения объема данных.
5. Вербализация – текстовое описание изображений с целью выявления семантики и содержательного поиска графической информации.

Таким образом, целью системы анализа изображений является определение параметров объектов вышеперечисленных разновидностей на основе параметров пикселей изображения.

Методы цифровой обработки изображений в настоящее время хорошо проработаны (см. например, Гонсалес, Вудс [9]) и продолжают развиваться. Проблемы анализа и содержательного описания нуждаются в более глубокой проработке в плане Image Mining – извлечения знаний из изображений и понимания изображений компьютером.

Рассмотрим требования к математическому обеспечению задач обработки и анализа графической информации. Основные требования к методам локального анализа – возможность оперативной смены разрешающей способности и чувствительности в зависимости от контекста [80].

Необходимость разработки новых методов для синтаксического анализа обусловлена зашумленностью, затенением объектов на реальных снимках или недостаточным разрешением сканирования/съемки. Ранее подобные задачи решались вручную, например, в ГИС эти операции выполняются пользователем с помощью специальных инструментов.

Для задач семантического анализа изображений объектов особенно важно выделить как можно надежнее первичные информативные элементы, которые будут служить исходной информацией для последующих этапов. Кроме того, необходимо обеспечить адаптируемость инструментария к контексту, поскольку на разных участках изображения, например, яркостные перепады различны.

Назначение этапа аппроксимации графической информации - упрощение последующих методов семантического анализа и сопоставления путём замены сложных объектов простыми: цветовых областей, границ, скелетонов.

Требования к методам данного этапа: рекурсивность и итерационное приближение к необходимой и достаточной точности (в зависимости от задачи, например, в аппроксимации это может быть отклонение дуги от реальной линии).

Требования к синтаксической модели изображения:

1. Структурированность в отношениях "часть - целое";
2. Содержательность - отображение всего, что удалось извлечь из изображения;
3. Возможность возврата к предыдущим этапам анализа в тех областях, где может потребоваться уточнение информации по критерию непротиворечивости семантики.

Требование к методам сопоставления, оценки сходства и отличий - удовлетворительное время выполнения и релевантность результатов ожидаемым.

Семантический анализ изображений выдвигает повышенные требования к

описанию структуры, формы объектов и методам их анализа. Что, в свою очередь, требует адаптации либо усовершенствования существующих методов сопоставления атрибутивных графов, разработки дескрипционных логик для описания изображений в терминах предметной области задачи.

Результатами анализа графической информации являются семантическая интерпретация синтаксических структур и прагматические измерения параметров, отношений, трендов [42].

2.2. Обоснование методологии содержательного описания и поиска графической информации

В современных условиях развития средств вычислительной техники, программного обеспечения и бурного роста потоков информации, в том числе графической, возникает потребность в её содержательном анализе и поиске. Актуальность анализа графической информации проистекает, с одной стороны, из её большого объема и весомости в информации об объекте, с другой – узкой (предметной) специализации большинства методов распознавания изображений.

Очевидно, содержательное описание изображений объектов не только требует всестороннего анализа объектов, но и глубокого, зачастую, многопроходного и разнопланового анализа, включая синтаксис, семантику и прагматику поступающей информации.

Целью содержательного описания графической информации является как можно более точное и полное извлечение практически полезной информации о параметрах объектов, имеющих на изображениях.

Основаниями предложенной методологии служат:

- Семиотика - раздел теории познания, исследующий реальный мир на трёх уровнях: физическом (прагматика), лингвистическом (синтаксис), понятийном (семантика);

- Когнитивистика - научное направление, рассматривающее теорию познания как единый комплекс когнитивной психологии, нейрофизиологии, когнитивной лингвистики, теории искусственного интеллекта, компьютерного моделирования.

Анализ современных достижений когнитивистики, методов анализа и извлечения знаний из слабоструктурированных данных позволил сформулировать общие принципы организации обработки графической информации, придерживаться которых особенно нужно при комплексировании систем с целью обеспечения структурированности компонентов системы, упрощения взаимодействия между ними и достижения необходимой степени достоверности и качества результатов:

1. Многоуровневая обработка – распределение функциональности по уровням процессов, каждый из которых специализируется на решении своего круга родственных задач.

2. Иерархия конструктивная и гетерархия по управлению (в зависимости от ситуации какой-то функциональный элемент становится главным) – обеспечение взаимосвязи между уровнями анализа для уточнения информации, поступающей с младших уровней.

3. Рекурсивный анализ информации – при выборе методов отдавать предпочтение рекурсивным с целью унификации алгоритмов управления процессом решения задач.

4. Оперативная смена разрешающей способности – динамическая смена степени интеграции при избыточности информации и степени детализации при недостаточности информации для получения компонентов описания анализируемых объектов.

5. Переход от количественных признаков к качественным (лингвистическим) – для удобства логического вывода и аналитика/пользователя.

6. Возможность организации параллельной обработки – предпочтение методам, удобным для распараллеливания в связи с большими объемами информации.

Перечисленные принципы в какой-то степени реализуют бионический подход, т.е. имитируют процессы анализа информации биологическими системами.

В анализе изображений разрыв по треугольнику Фреге (рисунок 2.1) наблюдается между представлением и понятием, то есть пониманием изображения.

Причиной (или, наоборот, следствием) является отсутствие взаимодействия между методами распознавания образов и логическим анализом. Методы распознавания образов применяются в задачах поиска, а логический анализ и рассуждения на графической информации начали использоваться относительно недавно, не более 10 лет, в том числе для поиска, в частности, в сети Интернет.

Очевидно, в практических задачах одного распознавания недостаточно. Возьмем проблему чтения слитного рукописного текста. Когда ребёнок учится писать, он составляет буквы из палочек, крючков, петелек. Потом эта работа доводится до автоматизма и он начинает распознавать. Но когда нам приходится читать чей-то неаккуратный почерк, опять подключаются рассуждения.

В данном исследовании этот разрыв устраняется за счет усиления методов извлечения первичных (низкоуровневых) информативных признаков, формирования представления в виде многослойного атрибутивного графа отношений и анализа путем сопоставления графов и дескрипционной логики, расширенной на данные, представленные атрибутивными графами отношений. Благодаря этому разрыв как бы сужается одновременно снизу и сверху.

Проблемами существующих систем анализа графической информации являются, во-первых, недостаточная степень автоматизации, а во-вторых, отсутствие возможности получения не только заложенных (запрограммированных), но и более сложных характеристик, в частности, описывающих форму объектов или динамику изменения (тренды), которые пользователь может запрашивать в интерактивном режиме.

Недостатком систем распознавания зрительных образов считается их узкая специализация - "заточенность" на конкретный класс объектов. Так, до сих пор нет удовлетворительного решения задачи распознавания слитного рукописного текста, хотя здесь всего три основных элемента: палочка, крючок, петля. Но очень большое разнообразие форм элементов, требующее соответствующего количества эталонов.

Проблемой систем поиска изображений по содержанию является ограниченность набора характеристик, используемых для поиска объектов, что приводит к недостаточной релевантности результатов и избыточной выдаче, которую приходится тщательно пересматривать. Следует отметить, что задачи содержательного описания изображений объектов, выявления сходства и отличий при этом не ставились. Соответственно, результаты менее структурированы и содержат ограниченный набор атрибутов.

Методы выделения первичных признаков достаточно просты, хоть и требовательны к вычислительным ресурсам. Применяются во многих системах обработки изображений. В качестве критериев выбора методов для задач анализа графической информации следует обратить внимание на надёжность работы в условиях среднего, низкого и переменного качества изображений, а также управляемость и адаптируемость в процессе анализа, требующего возврата для уточнения информации в проблемных местах.

Выявлено, что на изображениях низкого качества, например, архивные документы (чертежи, схемы, карты) результаты обработки не всегда удовлетворительны. Методы утончения линий, применяемые для получения скелетонов, хорошо работают на линейных объектах, а для площадных объектов применяют методы, основанные на диаграммах Вороного, которые требуют значительных вычислительных ресурсов. На изображениях, которые содержат и тот и другой вид объектов, проблемой становится кооперация методов. Для решения этой проблемы требуется разработка таких методов построения скелетонов, которые бы одинаково эффективно работали в тех и других условиях.

Методы, основанные на локальных дескрипторах не выделяют сходство и отличия объектов на разных изображениях. Затруднителен анализ пространственных отношений. Описание изображения в виде множества (до 500) дескрипторов не удобно для понимания человеком.

Методы цветовой сегментации характеризуются высокой вычислительной сложностью, несмотря на использование методов динамического программирования, "жадного" поиска и др.

Анализ теории и практики в области обработки графической информации (гл. 1) позволил выявить следующие проблемы.

К векторизации предъявляются очень высокие требования по качеству: минимизация количества отрезков прямых и дуг, минимум настроек, которые необходимо сделать пользователю для получения приемлемого результата на определенном изображении. Все это в условиях, когда качество исходных бумажных документов может быть низким, чертежи представлены на ксерокопиях, кальках, синьках или выполнены в карандаше. Особенности реальных чертежных изображений: нечетко прорисованные линии разной толщины и качества вычерчивания; штриховые, штрихпунктирные, мелкие объекты (цифры, буквы, знаки); некачественная бумага (калька, синька); пятна и потертости; места сближения линий, области перекрестков и острых углов.

При проектировании новых изделий немаловажным является применение накопленных знаний: существующих компонентов, конструкторских решений, технологических процессов. Известно, что проектировщики проводят от 20 до 30% своего времени, осуществляя поиск проектной информации и используя ее для завершения своей работы [159]. Однако найти требуемый документ среди тысяч непросто, учитывая графический характер технической документации, которую образуют чертежи, 3D-модели, сканированные документы, содержащие как текст, так и графику.

В настоящий момент проблема поиска чертежей решается главным образом на основе текстово-числовых атрибутов, кодов классификации и текстовых

описаний, создаваемых человеком. При этом, помимо высоких затрат ручного труда при создании и сопровождении архива, неполноты описания геометрии деталей, трудно формулировать запросы, в том числе из-за влияния субъективизма, когда люди описывают одну и ту же деталь по-разному, например, исходя из разных задач в разных отделах (конструкторском, технологическом). Стандартные способы поиска чертежей в некоторых ситуациях не приводят к удовлетворительным результатам.

К недостаткам систем поиска чертежей по образцу следует отнести невысокую надежность, если качество сканированных архивных чертежей низкое, а также отсутствие визуализации совпавших фрагментов чертежей (прозрачности принимаемых решений). Последнее связано с недостатками алгоритмов.

Автоматическое описание и классификация деталей по геометрическим, техническим, технологическим, производственным параметрам могла бы повысить степень унификации технологических процессов (ТП) и оснастки, повысить обоснованность и качество нормирования труда, оптимизировать планы производства. Для этого, в первую очередь, необходимо решить задачи Image Mining – автоматического извлечения признаков формы и относительного расположения структурных элементов из чертежей, 3D геометрических моделей, технологических эскизов.

2.3. Особенности применения семиотического подхода к проблематике моделирования, обработки и анализа графической информации

Известный в семиотике логический треугольник Фреге моделирует отношения между объектом реального мира и его представлением, пониманием (понятием) и обозначением (именем) в ментальном мире (рисунок 2.1) [74].

Предлагаемая методология семантического анализа и поиска графической информации основывается на семиотическом подходе к организации знаковых коммуникативных систем и включает три уровня: синтаксический

(морфологические элементы - первичные, низкоуровневые элементы объектов, грамматики конструкций), семантический (логический анализ и распознавание - понятийный уровень), прагматический (вербализация).

Применяя известный треугольник Фреге к задаче анализа изображений, получаем: денотат - видимое изображение; представление - контуры, цветовые области, границы, скелетоны; понятие - образ, характеристика свойств и отношений; имя - словесное обозначение. Использование когнитивного подхода позволяет построить многослойный граф изображения (контуры, цветовые области, границы, скелетоны), методы сопоставления графов и дескрипционная логика формируют



Рисунок 2.1 - Треугольник Фреге

онтологию понятий в терминах предметной области, предоставляют возможность проводить логический анализ и получать вербальное (словесное) описание. Таким образом, в ходе процесса анализа многослойного атрибутивного графа изображения формируется связь между вершинами "понятие" и "представление" семантического треугольника, что позволяет преодолеть семантический разрыв между атомарными (низкоуровневыми) элементами изображений и смыслом сложных структур, составленных из них, т.е. перейти от синтаксической модели представления изображения к пониманию.

Представленная на рисунке 2.2 концептуальная модель семантики изображения моделирует отношения между изображением реального объекта и его компьютерным представлением, пониманием и описанием (обозначением). Она отображает процессы перехода от традиционного представления изображения в виде двумерной матрицы пикселей к представлению в вычислительной среде: формирование многослойного атрибутивного графа - извлечение понятий и отношений между ними - описание изображённых

объектов. Прямоугольниками обозначены сущности, ромбами - отношения, овалами - атрибуты.

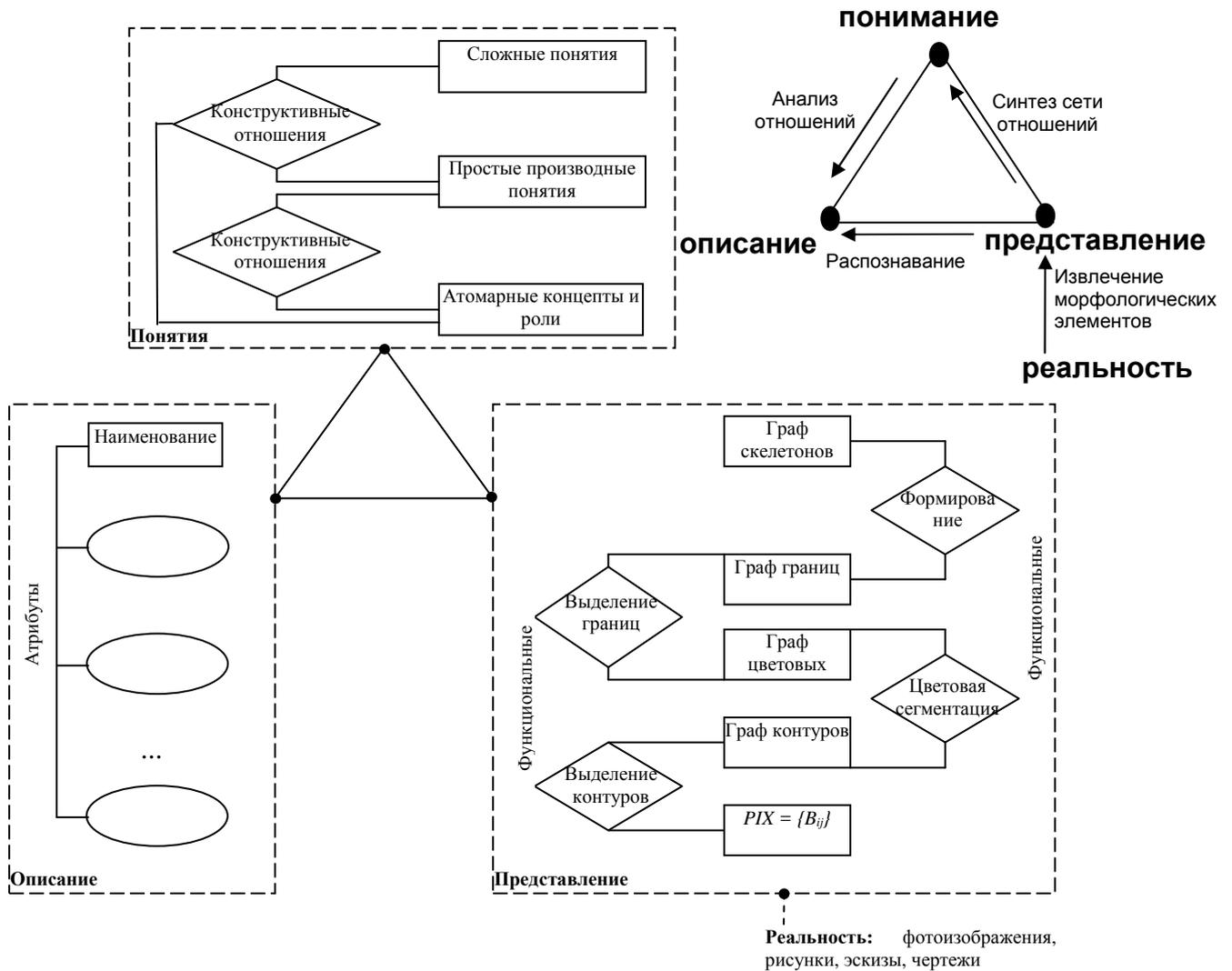


Рисунок 2.2 - Концептуальная модель процесса семантического анализа графической информации

Процесс понимания (осознания) графической информации предлагается рассматривать как процесс синтеза сети отношений между свойствами элементов и связей представленной графической информации и имеющейся в базе знаний графической и прагматической информации.

Развитие семиотического подхода осуществлялось в направлении семантического анализа и синтеза формализованных понятийных описаний изображений объектов со сложной структурой с помощью методов сопоставления графов и дескрипционной логики. В качестве аналогии была принята физиология

зрительного восприятия: синтаксический уровень - глаза, латеральное колленчатое тело, первичная зрительная кора; образное мышление - правое полушарие; логический анализ - левое.

Назначение предлагаемой методологии семантического анализа и поиска графической информации состоит в получении последовательности формализованных моделей синтаксического, семантического и прагматического уровней с помощью адекватных методов анализа и правил, задаваемых дескрипционной логикой изображений объектов в предметной области задачи.

В целом, семиотический подход к анализу графической информации рассматривается в данной работе как направление решения проблем автоматизации интеллектуального труда [6].

Под *семантикой изображения* условимся понимать тройку:

$$IS = \langle G, D, V \rangle, \text{ где:}$$

G – атрибутивный граф морфологических элементов и пространственных связей между ними – невербальный уровень понимания изображения;

D – формализованное словесное описание изображенных объектов, пространственных и сопоставительных отношений между ними – вербальный уровень;

V – граф-схема, структурная схема – визуальное (схематичное) представление информации изображения для контроля корректности восприятия системой, визуального общения между системой и человеком. Например, в системах автоматического анализа текстов широко применяются семантические сети – информационные графические модели, имеющие вид ориентированного графа, вершины которого соответствуют объектам, а дуги – отношениям между ними.

Формально модель системы семантического анализа графической информации можно представить как кортеж:

$$SA = \langle G(B), MLAG, \Phi, \Gamma(T), S, C', R' \rangle, \text{ где:}$$

$G(B)$ – исходная матрица пикселей изображения.

$MLAG = G(v0, GSEGM, \{GEDGE\}, \{GSKEL\}, GCONT, \{R\})$ – синтаксическая

модель изображения в виде многослойного атрибутивного графа:

$v0$ – корневая вершина с общими атрибутами изображения;

$GSEGM$ – граф цветных сегментов;

$\{GEDGE\}$ – множество графов границ сегментов;

$\{GSKEL\}$ – множество графов скелетов сегментов;

$GCONT$ – граф контуров изображения;

$\{R\}$ – множество отношений между слоями графа.

$\Phi = \Phi1 \cup \Phi2 \cup \Phi3 \cup \Phi4$ – множество функций и правил морфологического, синтаксического, семантического, прагматического анализа соответственно.

$\Gamma(T)$ – граф (дерево) терминологии, отображающий таксономию формирования сложных понятий из простых.

S – стратегия, определяющая порядок последующего анализа в зависимости от состояния процесса, включая возвраты для уточнения информации.

C', R' – множество понятий и отношений терминологии, описывающих объекты изображения, пространственные и сопоставительные отношения между ними.

Числовые и других типов фактические данные атрибутивного графа рассматриваются здесь как прагматика графической информации - результат анализа изображений в соответствии с задачами предметной области.

Предлагаемая грамматика относится к категории контекстно-свободных грамматик. Терминальными символами служат цветовые области, границы, контуры, скелетоны, полученные на этапе морфологического анализа. Синтаксические конструкции из них, представленные в виде *многослойного атрибутивного графа*, анализируются путем интерпритации понятий, заданных с помощью дескрипционной логики с расширением на пространственные данные, представленные в виде атрибутивных графов. Это позволяет уйти от контекстно-зависимых грамматик при описании объектов со сложной структурой благодаря тому, что терминология строго ограничивается графом $\Gamma(T)$ типа дерево.

Таким образом, сокращение семантического разрыва в анализе графической информации идет с двух сторон:

снизу - вверх: со стороны формальных грамматик семиотического анализа многослойного атрибутивного графа с помощью дескрипционной логики - к многоуровневой (количество уравней теоретически не ограничено) структуре понятий и отношений;

и на оборот: со стороны предметной онтологии через дескрипционную логику конструирования сложных понятий - к извлечению их семантики и прагматики с помощью рассуждателя на графах, вплоть до 3D геометрических моделей.

Отметим, что в работах по поиску изображений, опирающихся на дескрипционные логики, пока, в основном, рассматривается только один уровень - объекты и отношения соседства. В работах, основанных на структурно-лингвистическом подходе к анализу графической информации, обрабатывают только отрезки, особые точки и предопределенные последовательности (порождающие грамматики) из них.

В предлагаемом подходе процесс анализа графической информации осуществляется на трех уровнях, оперативно взаимодействующих между собой (рисунок 2.3).

В данной схеме для уточнения и корректировки информации введены обратные связи, а также экспертные правила взаимодействия уровней.

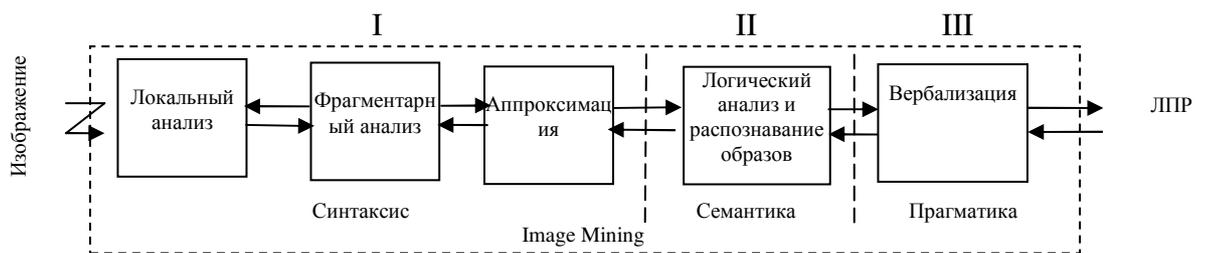


Рисунок 2.3 - Схема процесса семантического анализа и поиска графической информации

С одной стороны это этапы итерационного процесса осмысления информации для достижения необходимой степени конкретизации описаний, с другой - уровни контекстного охвата содержания и, соответственно, качественного состояния системы.

Локальный анализ составляют методы выделения локальных особенностей

на изображении. Это первая группа методов синтаксического анализа. Здесь формируются простые морфологические элементы описания изображения: контурные точки, разветвления и концы линий.

Для выделения локальных особенностей применяется метод выделения контуров на изображениях путем интегрально-лучевого определения направлений на соседние контурные пиксели [66, 68].

Основные требования к методам этой группы – возможность оперативной смены разрешающей способности и чувствительности в зависимости от контекста. Также здесь вычисляются интегральные и статистические характеристики локальной области: средняя яркость, распределение яркостей

Фрагментарный анализ - второй этап синтаксического анализа. Осуществляет выделение структурных элементов (деталей) изображения объекта в условиях зашумленности, затенения или недостаточной разрешающей способности. Здесь выполняются методы: оптимизация положения особых точек; уточнение особых точек (углов и разветвлений) путем экстраполяции лучей, исходящих из особых точек; выделение последовательностей точек; утончения и других операций. С их помощью конкретизируются первичные морфологические элементы объектов (или помех – отличить на данном этапе, без семантического анализа невозможно). При этом активизируются функции первого этапа, возвращаясь непосредственно к анализируемой области изображения. В частности, данный этап может затребовать смену разрешающей способности или чувствительности локального анализатора, позволяя находить слабо выраженные, размытые по площади особенности. Чем выше степень интеграции изображения, тем ниже разрешающая способность. С другой стороны, чем выше разрешающая способность, тем больше линий, изломов и других шумов, мешающих «видеть» целостную картину.

На этом уровне осуществляется:

- оптимизация и уточнение положения особых точек;
- трассировка линий контуров;
- адаптация с целью повышения надежности отслеживания контурных

линий, путем динамической смены степени интеграции и пороговых значений на анализируемом участке.

Необходимость разработки новых методов для этапа фрагментарного синтаксического анализа обусловлена зашумленностью, затенением объектов на реальных снимках или недостаточным разрешением сканирования/съемки. Ранее подобные задачи решались вручную, например, в ГИС эти операции выполняются пользователем с помощью специальных инструментов.

Аппроксимация и кодирование информации завершают формирование синтаксической структуры изображения и выполняет: кластеризацию цветовой палитры и цветовую сегментацию, скелетизацию, аппроксимацию линий отрезками дуг и прямых.

Также на данном этапе выполняются операции по сжатию описания изображения: аппроксимация контуров отрезками прямых и дуг. В ходе аппроксимации снова, но уже на основе цепочек уточняются особые точки и рекурсивно подбираются оптимальные точки перегиба – концы аппроксимирующих отрезков.

Требования к методам данного этапа: рекурсивность и итерационное приближение к необходимой и достаточной точности (в зависимости от задачи, например, в аппроксимации это может быть отклонение дуги от реальной линии).

Уровни распознавания и вербализации. Проблема семантического анализа изображений выдвигает повышенные требования к описанию структуры, формы объектов и методам их анализа. Для чего разработаны метод сопоставления атрибутивных графов и дескрипционная логика (ДЛ) для анализа семантики изображений в терминах предметной области изображений.

Распознавание методом сопоставления нечетких пространственно нагруженных графов заключается в выделении таких связных подграфов, которые имеют наибольшую степень сходства графа запроса с имеющимися в базе.

Результатами этих уровней анализа являются семантическая интерпретация синтаксических структур и прагматические измерения параметров, отношений, трендов [42, 158].

Вербализация изображения заключается в логическом анализе графа и выявлении таких конструкций, которые удовлетворяют терминологическому разделу базы знаний, созданному на основе дескрипционной логики.

Таким образом, можно выделить следующие этапы и задачи (рисунок 2.4.):



Рисунок 2.4 - Этапы и выполняемые задачи

2.4. Выводы по главе

Таким образом, на основе современных достижений когнитивистики, методов анализа и извлечения знаний из изображений разрабатывается методология, содержащая комплекс взаимосвязанных моделей и методов, а также методики на ее основе, позволяющие расширить набор инструментальных средств анализа изображений в автоматизированных системах и повысить их надежность - релевантность результатов человеческому восприятию изображений.

Необходимо подобрать и собрать в единую методологию такую совокупность методов, которые могли бы успешно взаимодействовать между собой, «помогая» друг другу в достижении наибольшей надежности при

обработке широкого круга классов изображений. При этом внимание уделяется прежде всего качеству, а не быстродействию того или иного метода.

Методология рассматривается как один из инструментариев технологии Image Mining – извлечения знаний из изображений. Сформулированы требования к методам, обеспечивающим выполнение этапов.

Проанализированы задачи, решаемые на информации, представленной в виде изображений, и выделены основные категории задач, в соответствии с которыми специализируются методы и алгоритмы обработки изображений: выделение и синтез первичных информативных элементов изображений; аппроксимация синтаксических конструкций; распознавание зрительных образов объектов; анализ параметров объектов по изображениям; вербализация изображений. Представлена разработанная в ходе данного исследования концептуальная модель процесса обработки, семантического анализа и поиска графической информации на основе когнитивной модели "треугольник Фреге" восприятия информации человеком, отличающаяся развернутым представлением графических информационных структур с целью разработки алгоритмов анализа и преобразования представлений. Приведены методологические и теоретические основания, структура методологии и требования к формализованным представлениям.

3. СИНТАКСИЧЕСКИЙ АНАЛИЗ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

3.1. Разработка методов локальной обработки

С помощью методов данной группы реализуется первый этап синтаксического анализа графической информации (рисунок 2.3).

К морфологическим элементам изображения отнесём линии, составленные из контурных точек, и особые точки на этих линиях: конец ветви, разветвление, угол. Задачу их выделения обычно называют векторизацией, т.е. переводом изображения из растрового формата в векторный.

Основными требованиями к векторизации, исходя из последующих задач, являются:

- 1) надежность в условиях плавных перепадов яркости;
- 2) качество в местах сближения линий;
- 3) минимальное количество отрезков прямых и дуг, при максимальном соответствии объектам;
- 4) минимум ручной "доводки";
- 5) минимальное количество настроек векторизатора для получения приемлемого результата.

Задача извлечения простых элементов (часто называемых морфологическими) включает следующие шаги, выполняемые в зависимости от ситуации на участке изображения, т.е. в зависимости от контекста:

- поиск особых точек (конец ветви, разветвление, угол);
- поиск объектов на изображениях, состоящих из линий, не содержащих особые точки;
- поиск объектов на фотоизображениях пространственных сцен, не содержащих особые точки.

Результатом анализа и выделения простых элементов графической информации служат: множество $\{OTZ\}$ особых точек, $OTZ_i = \langle x, y, a_1, \dots, a_w \rangle$,

где x, y - координаты точки;

a_1, \dots, a_w - список направлений ветвей из особой точки.

множество $\{L\}$ цепочек, состоящее из соседних единичных векторов, $L_i = \langle k_i, x_1, y_1, \alpha_1, \dots, x_{k_i}, y_{k_i}, \alpha_{k_i} \rangle$,

где x, y - координаты точки;

a - направление на следующую (соседнюю) точку;

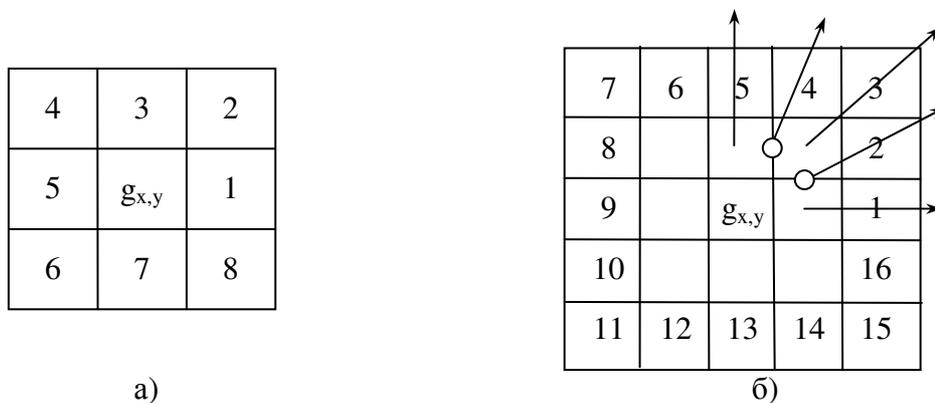
k - количество элементов цепочки.

Для выделения локальных особенностей на изображении разработаны методы:

- 1) метод анализа лучей, исходящих из центрального пиксела окрестности;
- 2) метод вычисления вектора контурной линии;
- 3) метод выделения особых точек;
- 4) метод выделения граничных точек площадного объекта.

Модель исходного растрового изображения представлена двумерной матрицей $G(B)$ пикселей с яркостью B_{ij} , $i = 1..N, j = 1..M$. Требуется выделить из G подмножество K точек контуров объектов на изображении, $K \subset G$.

Для определения принадлежности точки контурам объектов необходимо определить её локальные характеристики. Для этого рассматриваются окрестности пиксела 3×3 и 5×5 - подмножества R^1, R^2 пикселей, лежащих на концах лучей (рисунок 3.1):



а)

б)

Рисунок 3.1 - Виды анализируемых окрестностей:
а) множество R^1 ; б) множество R^2 и анализируемые лучи

$$R^1 = \{r^1_i\} \text{ где } r^1_i = \{g(x+IX1[i], y+IY1[i]), \rho(r^1_i, g_{x,y})=1\}; \quad (3.1)$$

$$R^2 = \{r^2_j\} \text{ где } r^2_j = \{g(x+IX2[j], y+IY2[j]), \rho(r^2_j, g_{x,y})=2\}, \quad (3.2)$$

где: $i=1..8; j=1..16;$

x, y - координаты центрального пиксела окрестности;

ρ – функция расстояния от конца луча до центра;

$$IX1 = (1, 1, 0, -1, -1, -1, 0, 1);$$

$$IY1 = (0, -1, -1, -1, 0, 1, 1, 1);$$

$$IX2 = (+2, +2, +2, +1, 0, -1, -2, -2, -2, -2, -2, -2, -1, 0, +1, +2, +2);$$

$$IY2 = (0, -1, -2, -2, -2, -2, -2, -1, 0, +1, +2, +2, +2, +2, +2, +2, +1).$$

Метод анализа лучей, исходящих из центрального пиксела окрестности.

Чтобы в дальнейшем произвести идентификацию точки, необходимо проанализировать перепады яркостей в окрестностях $\rho = 2$ относительно выбранной центральной точки.

Введем вспомогательное множество лучей TB , где каждый луч (половина диагонали) - это последовательность пикселов внутри рассматриваемой окрестности, через которые проходит линия, исходящая из центрального пиксела $g_{x,y}$.

С учетом формулы (3.2) множество TB можно представить в следующем виде:

$$TB = \{tb_j\};$$

$$tb_j = \begin{cases} \beta(r_j^2), & \text{если } (\beta(r_{j-1}^2) - \beta(r_j^2) \geq P) \\ & \vee (\beta(r_{j-1}^2) - \beta(r_j^2) < P) \wedge (tb_{j-1} \neq Const) \wedge (\beta(r_j^2) \leq \beta(r_{j-1}^2)) \\ & \vee (\beta(r_{j+1}^2) - \beta(r_j^2) \geq P) \\ \text{иначе } Const \end{cases}, \quad (3.3)$$

где: $\beta(r_j^2)$ – яркость j -той точки в окрестности радиуса $\rho = 2$;

j – принимает значение 1..16;

P – исходное пороговое значение перепада яркости.

Формула (3.3) интерпретируется следующим образом. Если разность яркостей предыдущего и текущего пикселей окрестности больше либо равна P , то произошло изменение яркостей рассматриваемых точек в сторону уменьшения, то есть переход на менее яркий (далее назовем его "темный") луч. Яркость текущего пикселя заносится в множество TB с индексом номера направления луча (рисунок 3.2). Если разность яркостей меньше порогового значения, но текущий пиксел темнее или равен по яркости предыдущему темному лучу, то пикселы находятся в яркостной "впадине" и рассматриваемый пиксел также будет занесен в множество TB . Когда текущий луч находится в яркостной впадине, он сравнивается с последующим и, если разность превышает или равна пороговому значению, он также заносится в множество TB .

Таким образом, в множество TB записываются лучи, которые попали в темный (с точностью до P) сектор лучей, исходящих из центрального пикселя окрестности. Остальные лучи принимают значение $Const$, равное белому цвету (например 255).

Несколько соседних темных лучей составляют "темный" сектор, например, сектор, образованный лучами r_3, r_4 (рисунок 3.2).

Здесь и далее, поскольку все лучи исходят из одного центра и яркость конечной точки становится интегрированной яркостью всего луча, будем для краткости обозначать их индексом конечной точки.

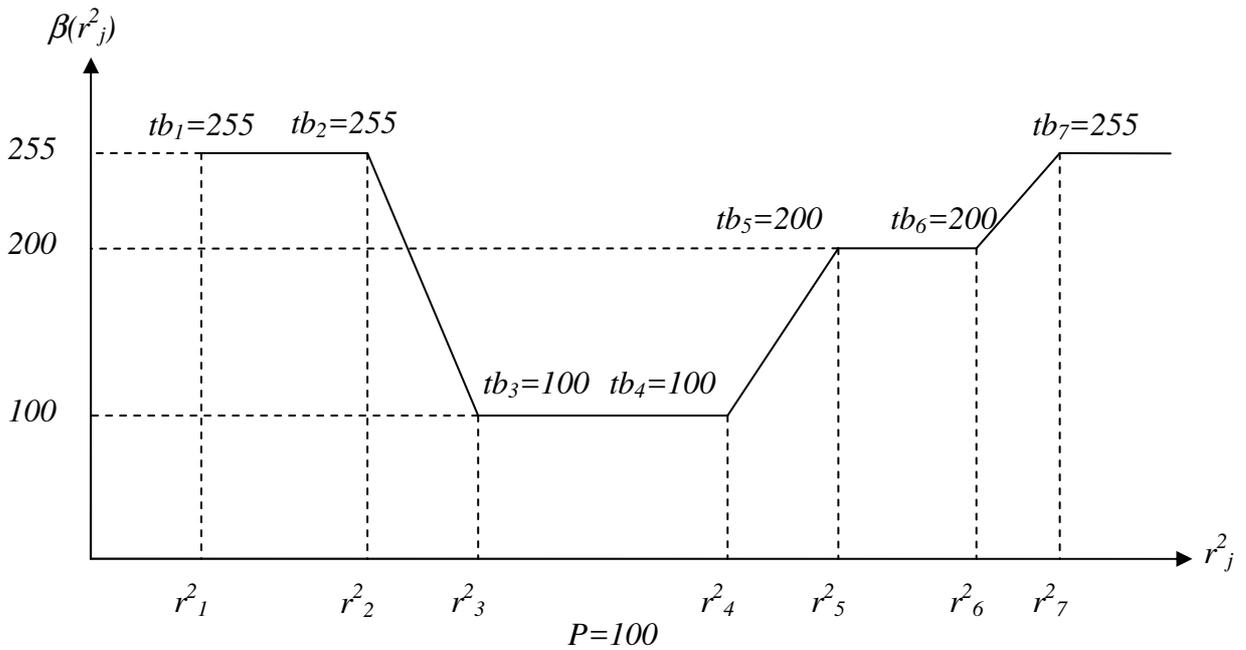


Рисунок 3.2 - Определение принадлежности точки множеству темных лучей

Метод вычисления вектора контурной линии в заданной точке. Для удобства последующей трассировки каждый темный сектор заменяется направлением, указывающим на следующий пиксел контурной линии (в отличие от градиентного метода) путём вычисления средневзвешенного арифметического значения яркостей лучей, принадлежащих темному сектору [87]. При этом считается, что чем темнее луч, тем выше его вес, вычисляемый как $255 - tb_j$, т.е. инвертированная величина по отношению к яркости. Направление на средневзвешенный центр считается искомым направлением контурной линии на следующий пиксел.

С учетом того, что из одной и той же точки растра может исходить несколько линий, множество их направлений $A = \{a_i\}$, где $a(i)$ – угол i -того направления, определяется как средневзвешенное арифметическое яркостей лучей соответствующего тёмного сектора:

$$a_i = \begin{cases} \frac{\sum_{j=k}^l ((255 - tb_j) \cdot \alpha(j))}{\sum_{j=k}^l (255 - tb_j)}, & \text{пока } tb_j \neq Const \\ Const, & \text{в противном случае} \end{cases}, \quad (3.4)$$

где: i – изменяется от 1 до 8, т.к. на окрестности R^2 не может быть более 8 независимых направлений одновременно;

k, l - номера начального и конечного лучей i - того темного сектора;

$\alpha(j) = (j - 1)\pi/8$ – угол луча $j \in TB$,

то есть количество ветвей $W(g_{x,y})$, исходящих из точки $g_{x,y}$ равняется количеству элементов множества A . Число $Const$ означает, что направление не определено.

Метод выделения особых точек. Под множеством особых точек $OT = \{ot_i\}$ подразумеваются точки типа разветвление, конец ветви или острый угол (рисунок 3.3) [75]:

$OT = \{ot_i\}$, где $ot_i = g_{xy}$ если $(W(g_{xy}) \neq 2) \vee (W(g_{xy}) = 2) \wedge (a_1(g_{xy}) - a_2(g_{xy}) \leq \pi/2)$

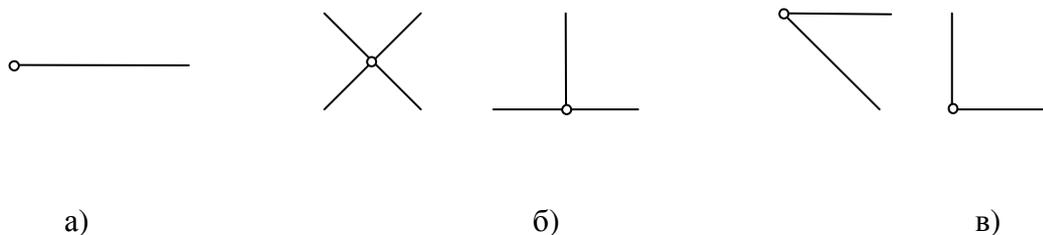


Рисунок 3.3 - Особые (узловые) точки изображения:
 а) конец ветви; б) разветвление; в) угол

Метод выделения граничных точек площадного объекта. При стандартном режиме работы метода вычисления вектора контурной линии, находясь на границе площадного объекта, по формуле (3.4) получаем одно направление на яркостную середину, т.е. внутрь площадного объекта, вместо необходимых двух направлений по "краям" (рисунок 3.4).

Поскольку направления по "краям" есть направления лучей на границах темного сектора, то скорректируем множество A так, что

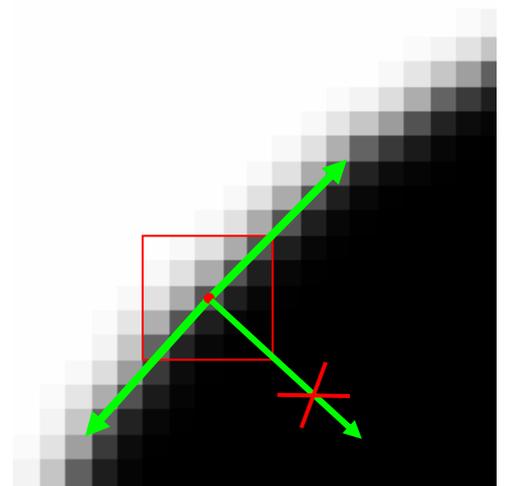


Рисунок 3.4 - Анализ границ площадного объекта

$$a(W) = \alpha(i) \text{ и } a(W+1) = \alpha(j),$$

если $(tb(i) \neq Const) \wedge (tb(i-1) = Const) \wedge (tb(j) \neq Const) \wedge (tb(j+1) = Const) \wedge (i-j > 5)$, (3.5)

где: $i, j = 1..16$ и соответствуют направлениям (рисунок 3.1);

$\alpha(i), \alpha(j)$ – углы начала и конца темного сектора окрестности $(g_{x,y})$;

W – количество ветвей из пиксела $(g_{x,y})$, значение изменяется от 1 до 8.

За счёт этого последнее направление в исходном множестве $a(W)$ заменится на первое направление по "краю", второе направление по "краю" запишется в новый элемент $a(W+1)$, что позволяет в процессе выделения контуров объектов на изображении беспрепятственно переходить с линейных объектов на границы площадных и обратно.

Таким образом, алгоритм локального анализа для формирования множеств особых точек заключается в следующем.

Дано: матрица $G(B)$ пикселей с яркостью B_{ij} , $i = 1..N, j = 1..M$.

Требуется: сформировать множество особых точек $OT = \langle x, y, b_{xy}, a_1, \dots, a_W \rangle$, где b_{xy} - средняя яркость окрестности $\rho = 1$.

Алгоритм:

$i := 0; j := 0;$

Цикл $x = 1..N$ по строкам матрицы $G(B)$ пикселей

Цикл $y = 1..N$ по столбцам матрицы $G(B)$ пикселей

{Формирование множества TB лучей окрестности;

Формирование множества A направлений линий;

$W(g_{xy}) := \text{КолНаправлений};$

Если $(W \neq 0) \text{ AND } (W \neq 2)$

То $\{j := j+1; \text{Запись в список } OT\}$

}

3.2. Разработка методов фрагментарной обработки

К методам **фрагментарного анализа** отнесем методы, анализирующие *совокупности* соседних окрестностей:

- метод рекурсивной оптимизации особых точек;
- метод экстраполяции особых точек;
- метод выделения цепочек контурных линий.

Под артефактами (особыми точками) понимаются следующие типы точек:

- концевая точка – из точки исходит одна ветвь;
- угловая точка – из точки исходят две ветви, угол между которыми составляет не более $\pi/2$;
- разветвление – из точки исходят три и более ветвей;

В процессе автономного анализа изображения, который осуществляется оператором-анализатором локальных областей, как правило, в местах, где линии сближаются или образуют острые углы, выделяется связанное подмножество особых точек (ОТ), которое образует область **артефактов** (рисунок 3.5а). Возникает задача выбора из множества особых точек одной точки.

Метод рекурсивной оптимизации особых точек. Метод заключается в рекурсивной оптимизации особых точек объекта в артефактной области путем поиска локального экстремума яркости окрестности *ОТ* (рисунок 3.5).

Дано: множество $OT = \langle x, y, b_{xy}, a_1, \dots, a_w \rangle$ особых точек.

Требуется: множество $OT2 = \langle x, y, a_1, \dots, a_w \rangle$ оптимизированных особых точек.

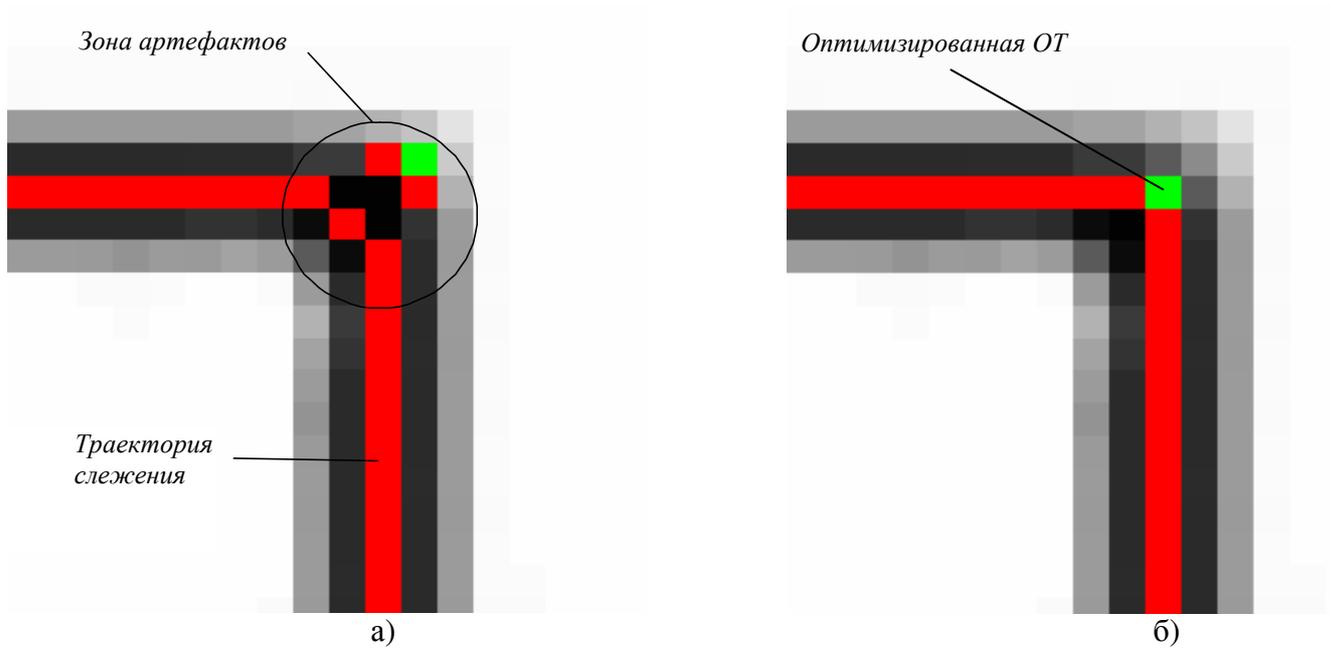


Рисунок 3.5 - Оптимизация особой точки

Алгоритм: В цикле по множеству OT из каждой особой точки осуществляется последовательное перемещение по окрестности $\rho = 1$ (по формуле 3.1). Если средняя яркость в окрестности соседней OT (того же типа) меньше исходной (темнее), то из неё рекурсивно вызывается эта же функция:

$$\exists ot_{min} = MIN_{\rho}^1(ot_i),$$

где ot_{min} – точка с минимальной яркостью в связном подмножестве OT ;

ot_i – i -тая точка из множества OT , $i = 1..k$, где k - количество точек;

MIN_{ρ}^1 – функция выбора точки с минимальной яркостью окрестности в области артефактов.

Таким образом, из связного подмножества особых точек выбирается точка с наименьшей средней яркостью (наиболее темная) (рисунок 3.5б).

Метод экстраполяции особых точек – это расчет положения особой точки путем экстраполяции ветвей, исходящих из OT , и вычисления гипотетической точки их пересечения.

Процесс осуществляется только для особых точек типа "разветвление" и "угол". Для этого из ранее отобранной OT выполняется шаг $h = 2$ (или более) в каждом из направлений ветвей. В полученных точках вычисляются направления "назад" $\alpha'_i = \alpha_i + \pi$, где α_i - направление продолжения ветви. Искомая уточненная OT есть точка пересечения всех направлений "назад" (рисунок 3.6).

Дано: множество $OT2 = \langle x, y, a_1, \dots, a_w \rangle$ оптимизированных особых точек.

Требуется: множество $OT3 = \langle x, y, a_1, \dots, a_w \rangle$ уточнённых особых точек.

Алгоритм:

1. Расчет точки пересечения произведем по углам и координатам точек на исходящих ветвях. Для этого вычислим координаты точки на первой ветви:

$$X_1 = X_{OT} + h \cdot \cos(a_1);$$

$$Y_1 = Y_{OT} - h \cdot \sin(a_1),$$

где $h = 2$ – размер шага в пикселах;

a_1 – угол первой ветви;

X_{OT}, Y_{OT} – координаты особой точки.

Затем вычислим точки на всех других исходящих ветвях:

$$X_i = X_{OT} + h \cdot \cos(a_i);$$

$$Y_i = Y_{OT} - h \cdot \sin(a_i),$$

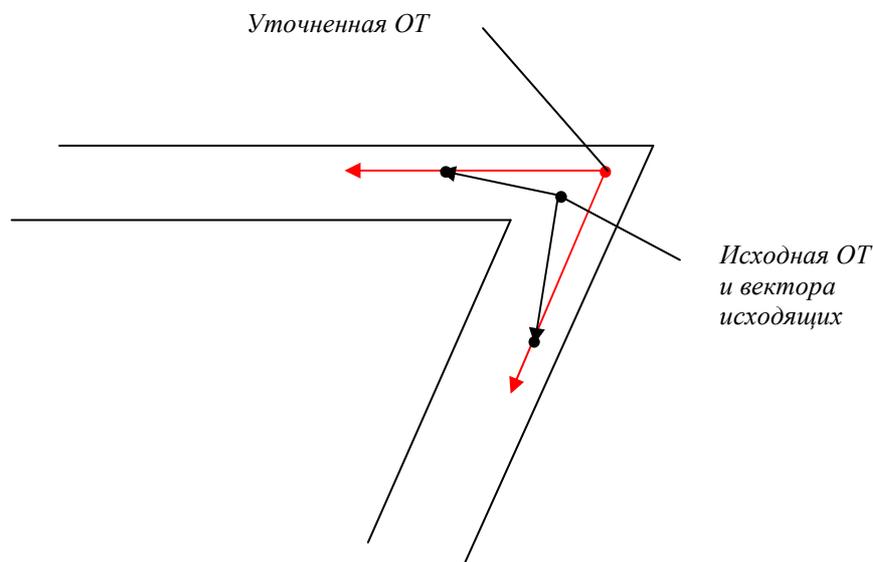


Рисунок 3.6 - Уточнение особой точки по пересечению исходящих ветвей

где i – изменяется от 2 до W ;

W – количество ветвей из точки X_{OT}, Y_{OT} .

2. Имея угол прямой и точку на ней, последовательно вычислим точки пересечения первой ветви с каждой последующей ветвью. При выводе рабочего уравнения прямой за основу взято уравнение:

$$y = kx + b,$$

где $k = \operatorname{tg} \varphi$,

т.е.

$$y = x \frac{\sin \varphi}{\cos \varphi} + b,$$

$$y \cos \varphi = x \sin \varphi + b \cos \varphi;$$

$$x \sin \varphi = -y \cos \varphi + b \cos \varphi.$$

Или в каноническом виде:

$$Ax + By + C = 0,$$

$$A = \sin \varphi; B = \cos \varphi; C = -x_1 \sin \varphi + y_1 \cos \varphi = -(A x_1 + B y_1),$$

где x_1, y_1 – координаты любой точки, принадлежащей описываемой прямой.

3. Искомыми координатами пересечения двух прямых является решение системы двух линейных уравнений с двумя неизвестными:

$$\begin{cases} A_1 + B_1 + C_1 = 0 \\ A_2 + B_2 + C_2 = 0 \end{cases} \text{ или } \begin{cases} x = \frac{B_1 C_2 - B_2 C_1}{A_1 B_2 - A_2 B_1} = \frac{B_1 C_2 - B_2 C_1}{Z} \\ y = \frac{C_1 A_2 - C_2 A_1}{A_1 B_2 - A_2 B_1} = \frac{C_1 A_2 - C_2 A_1}{Z} \end{cases} \quad (3.6)$$

Когда значение знаменателя $Z = A_1 B_2 - A_2 B_1$ в выражении стремится к нулю, значение координат точки стремится к бесконечности:

$$\lim_{|Z| \rightarrow 0} x \rightarrow \pm \infty \quad \lim_{|Z| \rightarrow 0} y \rightarrow \pm \infty,$$

т.е. прямые параллельны. В программе если Z принимает значение меньше 0.01, то прямые считаются параллельными.

4. Усредним полученные точки пересечения ветвей, исходящих из разветвления:

$$X_{sum} = \sum X P + X_{sum};$$

$$Y_{sum} = \sum Y P + Y_{sum};$$

$$X_{отр} = X_{sum} / W;$$

$$Y_{отр} = Y_{sum} / W.$$

5. Таким образом, получили усредненную точку ot_p пересечения исходящих ветвей из точки ot_{min} и, если полученная точка удовлетворяет условию:

$$ot_p \in OT \wedge W_{отр} \geq W_{отmin} \wedge \beta_{cp}(ot_p) \leq \beta_{cp}(ot_{min}),$$

где β_{cp} – средняя яркость в окрестности 5x5 рассматриваемой точки.

Метод выделения цепочек путём анализа вектора линии в локальной окрестности контурной точки и захвата особых точек. Упорядоченные последовательности точек, составляющих контурные линии объектов на изображении, назовем "цепочками". Началом и концом цепочек являются Особые Точки:

$$g_{x,y} \in K, \text{ если } (W(g_{x,y}) = 2) \vee (g_{x,y} \in OT3),$$

где K - множество контурных точек.

Последовательно перебирая особые точки и все направления из каждой ОТ, выделяются все цепочки (множество линий L) на изображении. При этом каждая следующая точка цепочки, в которой осуществляется анализ, вычисляется по формулам:

$$x_s = x_n + h \cos(a_i);$$

$$y_s = y_n - h \sin(a_i),$$

где x_n, y_n – текущая точка (на первом шаге - это ОТ);

x_s, y_s – вычисляемая точка в направлении a_i ;

a_i – i -тое направление на следующий пиксел.

Изначально a_i равняется i -тому направлению из особой точки. На последующих шагах - направлению "не назад" из вычисленной (новой) точки цепочки, т.е. соответствует:

$$a_i = \begin{cases} a_1 & \text{если } (a_1 - a_i) < (a_2 - a_i) \\ a_2 & \text{если } (a_2 - a_i) < (a_1 - a_i) \end{cases} .$$

При входе в область артефактов производится скачок по прямой в ранее оптимизированную особую точку, имеющую направление противоположное направлению слежения цепочки, т.е. "Захват" ОТ.

Алгоритм выделения цепочек (последовательностей) контурных пикселей:

Дано: множество $OT3 = \langle x, y, a_1, \dots, a_w \rangle$ особых точек,

где x, y - координаты точки;

a_1, \dots, a_w - список направлений ветвей из особой точки.

Требуется: множество $\{L\}$ цепочек, L_i - список соседних единичных векторов $L_i = \langle k_i, x_i, y_i, \alpha_1, \dots, \alpha_{k_i}, y_{k_i}, \alpha_{k_i} \rangle$,

где x, y - координаты точки;

a - направление на следующую (соседнюю) точку;

k - количество элементов цепочки.

Алгоритм:

Цикл j по множеству $OT3$ особых точек

Цикл i по направлениям из OT_j

$\{k := 1; xp := x; yp := y; ap := a_{ij};$

Запись k -той $\langle xp, yp, ap \rangle$ в цепочку

Цикл пока не найдена особая точка

$\{xs := xp + h * \cos(ap); ys := yp - h * \sin(ap);$

Формирование множества TB лучей окрестности (xs, ys) ;

Формирование множества A направления линий;

$W := \text{КолНаправлений};$

Если $(W \neq 0)$ AND $(W > 2)$

То Если в $OT3$ существует $OT(xs, ys)$ имеющая встречное направление $(ap + \pi$ или $ap - \pi)$

То $\{k := k + 1; \text{Запись } k\text{-той точки } \langle x_{ob}, y_{ob}, ap \rangle \text{ в цепочку};$

Выход из цикла}

Если $(a_1 - ap) < (a_2 - ap)$

То $ap := a_1;$

Иначе $ap := a_2;$

$k := k + 1; \text{Запись } k\text{-той точки } \langle xs, ys, ap \rangle;$

}

}

3.3. Общий алгоритм анализа и синтеза морфологических элементов изображения

Алгоритм анализа и синтеза морфологических элементов изображения включает следующие шаги, выполняемые в зависимости от ситуации на участке изображения (в зависимости от контекста):

- отслеживание линий и захват особых точек;
- переход на слежение по краю площадного объекта;
- адаптация к линиям с затухающей контрастностью;
- смена разрешающей способности в процессе анализа.

1. Вначале выделяются особые точки путём построчного сканирования исходного растрового изображения локальным анализатором по методу анализа лучей, исходящих из центрального пиксела окрестности $R = 2$, и методу выделения особых точек.

2. Если особые точки существуют, то в артефактных областях - зонах компактного размещения особых точек - запускается метод рекурсивной оптимизации особых точек и метод экстраполяции особых точек.

3. Для выделения объектов без особых точек осуществляется поиск "гладких" линий (с плавными контурами без резких изгибов). После того, как все линии, кроме гладких, уже выделены, снова выполняется построчное сканирование кадра:

$$g_{x,y} \in OTI, \text{ если } (KW_{x,y} \neq 0) \wedge (g_{x,y} \notin K),$$

где OTI – множество точек гладких контуров (объектов), K - множество ранее найденных контурных точек, т.е. если точка имеет исходящие ветви и не была выделена, то точка принадлежит гладкому объекту (рисунок 3.7).

Края трёхмерных объектов на изображениях могут быть размыты. Тогда при сканировании кадра первая найденная точка гладкого объекта вероятнее всего будет находиться в размытой области. Поскольку на изображениях, составленных из линий, задача - как можно более точно выделить объекты, а не

размытую часть контуров, необходимо скорректировать положение найденной точки перед началом слежения:

$$\exists ot_{min} = MIN^l_k(ot_i), \quad (3.7)$$

где ot_{min} – точка с минимальной яркостью из множества OTI ;

ot_i – i -тая точка из множества OTI ;

i – изменяется от 1 до k , где k – количество точек множества OTI ;

MIN^l_k – функция выбора точки с минимальной яркостью из множества OTI ,

т.е. если в окрестности $R=1$ точки ot_i есть более темная точка, отвечающая условию (3.5), то необходимо сместиться в нее и повторять анализ и смещение до тех пор, пока есть более темная точка в окрестности.

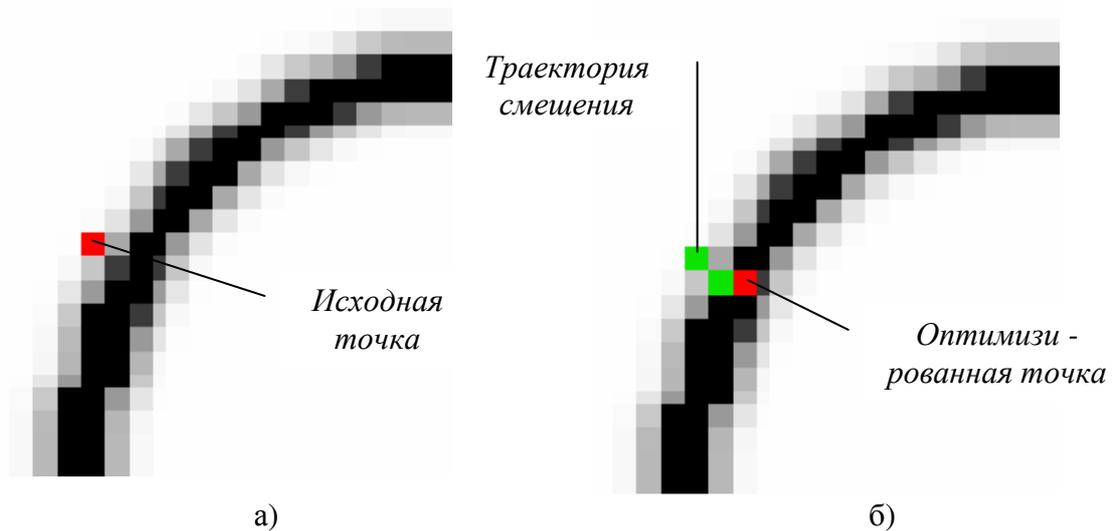


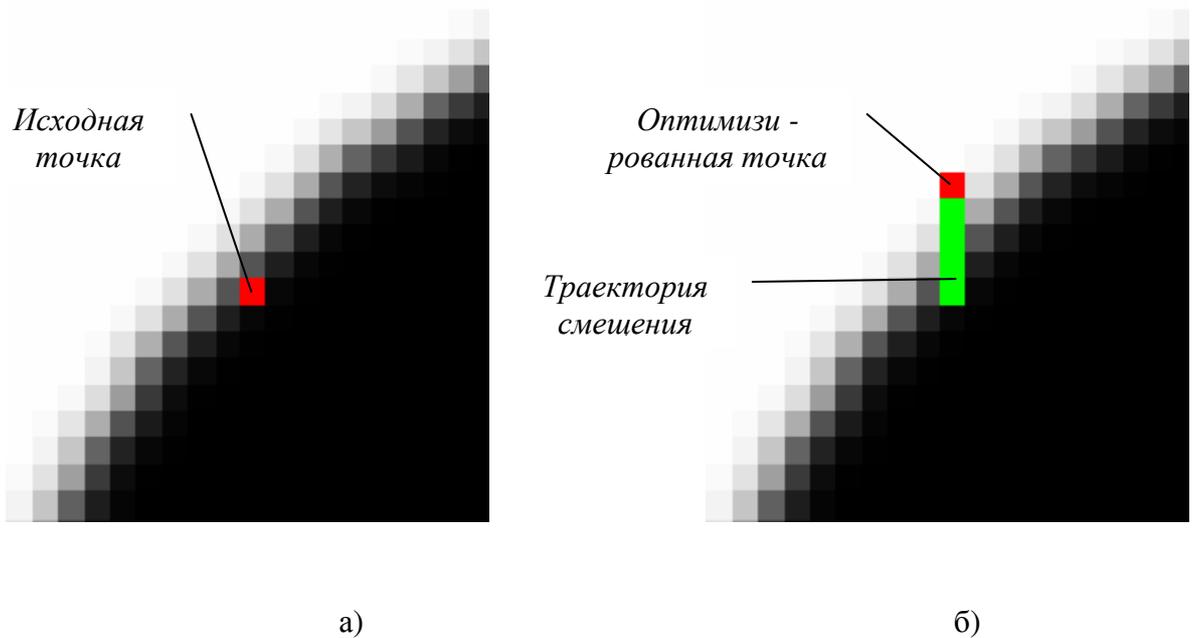
Рисунок 3.7 - Оптимизация точки начала отслеживания "гладких" линий

Таким образом, начиная выделение контуров с наиболее темной точки, выделяется самая яркая (значимая для этого типа изображений) часть объекта (рисунок 3.7б).

4. Выделение объектов без особых точек на фотографических изображениях трёхмерных объектов. В отличие от изображений предыдущего типа, здесь необходимо наиболее полно выделить объект. Для этого после сканирования изображения и нахождения точки, принадлежащей гладкому объекту, смещаем ее не в темную область, а в более светлую (рисунок 3.8):

$$\exists ot_{i_{max}} = MAX^l_k(oti_i), \quad (3.8)$$

где OTI_{max} – точка с максимальной яркостью из множества OTI ;
 oti_i – i -тая точка из множества OTI ;
 i – изменяется от 1 до k , где k – количество точек множества OTI ;
 MAX^l_k – функция выбора точки с максимальной яркостью из множества OTI ,



а) б)
 Рисунок 3.8 - Оптимизация точки начала слежения на изображениях трехмерных объектов

т.е. если в окрестности $R=1$ точки oti_i есть более светлая точка, отвечающая условию (3.8), то необходимо сместиться в нее и повторить анализ и смещение до тех пор, пока есть более светлая точка (и не фон) в окрестности (рисунок 3.8б). Начиная слежение с полученной точки, более качественно выделяется размытый объект.

5. После выделения и уточнения особых точек запускается метод выделения цепочек и «Захват OT » – скачкообразный переход в оптимизированную особую точку, как только линия входит в область артефактов.

5.1. В тех областях изображения, где линейные и площадные объекты примыкают друг к другу, выполняется переход на слежение по краю. На реальных изображениях линейные объекты часто сочетаются с площадными (рисунок 3.9а).

При выделении цепочек на линейном объекте направлением на следующий пиксел считается средневзвешенный центр в "темном секторе" (п. 3.1), что приводит к углублению внутрь на площадном объекте. Но автоматическое получение из оператора анализа локальных областей признака "граница площадного объекта" (п. 3.1) позволяет благополучно переходить с линейного объекта на площадной и обратно (рисунок 3.9б), т.е. автоматически заменять направление вглубь объекта на два направления движения по "краям".

На рисунке 3.9а исходное изображение, а на рисунке 3.9б исходное изображение с наложенными на него цепочками пикселов, полученными с помощью автоматического переключения на режим "слежение по краю".

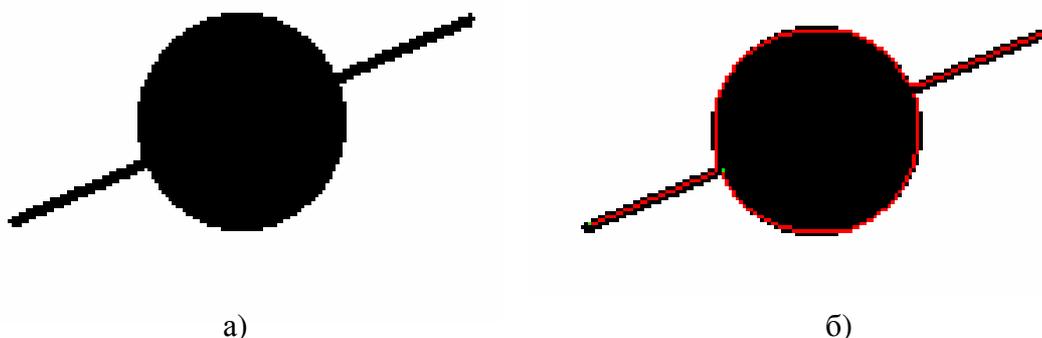


Рисунок 3.9 - Автоматическое переключение на режим "слежение по краю":
а) исходное изображение; б) изображение и цепочки пикселов, полученные путём перехода на "слежение по краю"

5.2. В ходе отслеживания контуров осуществляется адаптивная настройка пороговых величин по характеру окрестности. С целью более качественного выделения линий, если в процессе трассировки контура количество исходящих ветвей из контурного пиксела становится меньше либо равно единице, то яркостный порог динамически уменьшается [67].

Адаптация порога слежения за исчезающими линиями позволяет отслеживать на изображении даже очень тонкие (ярко слабо) линии, если удалось "зацепиться" за такую линию на каком-то участке при исходном яркостном пороге (рисунок 3.10г). Иначе искомые линии становятся короче или превращаются в точки (шумы) (рисунок 3.10б). В то же время, если минимальный порог установить изначально, то на обработанном изображении появится много

ложных, шумовых линий (рисунок 3.10в). Особенно это относится к объектам, имеющим трехмерную природу.



Рисунок 3.10 - Отслеживание контурных линий: а) исходное изображение; б) высокий порог на перепад яркости; в) низкий порог; г) адаптивный порог

6. Если на изображении одновременно присутствуют крупные и мелкие объекты, перед его обработкой создается несколько изображений таких, что каждое последующее *сжимается вдвое*. Оператор анализа локальных областей при получении значения ширины линии больше либо равно четырем пикселям автоматически переходит на изображение с уменьшенной разрешающей способностью и наоборот. Данный прием позволяет видеть линии вне зависимости от их толщины, получать среднюю линию объекта, корректно обрабатывать (исключать или прослеживать) шумовые разрывы линий размером 1-3 пикселя и помехи на изображении.

Формула преобразования **усредняющее сжатие**:

$$X'_{ij} = \frac{1}{9} \sum_{i-1}^{i+1} \sum_{j-1}^{j+1} X_{ij} ,$$

где окрестность 3x3 изображения заменяется одним пикселем, получающим среднее значение цвета.

Цель усредняющего сжатия – преобразовать изображение так, чтобы оно занимало меньшую площадь в поле зрения и при этом избежать искажений, но сохранить основные характеризующие свойства.

В отличие от простого прореживания пикселей, качество изображения, практически, не ухудшается. Наблюдатель как бы удаляется от объекта, чтобы увидеть его целиком для лучшего понимания.

Если теперь рассматривать только заменяющие пиксели, то получится исходное изображение с более низкой разрешающей способностью. Данную операцию можно производить неоднократно. Двигаясь по средней линии линейного объекта на исходном изображении, и встретив разрыв или расширение линии, анализатор переходит на уровень с более низкой разрешающей способностью. Если же линия закончилась на уровне с низкой разрешающей способностью, то переход производится на уровень с большей разрешающей способностью, чтобы попытаться найти продолжение тонких линий.

Метод усредняющего сжатия является вспомогательным и позволяет повысить качество решения нескольких задач: выделение средней линии линейных объектов без предварительной процедуры утончения, выделение линий, имеющих шумовые разрывы изображения, удаление мелких шумов и выделения текстур путем усредняющего сжатия и цветовой сегментации изображения.

Таким образом, для повышения устойчивости процесса трассировки предлагается применять выявленные в процессе моделирования и научных экспериментов дополнительные *функции поиска продолжения*:

1. *«Захват ОТ»* – скачкообразный переход в оптимизированную особую точку, как только линия входит в область артефактов.

2. *Переход на слежение по краю* – автоматическая замена направления вглубь объекта на два направления движения по «краям» в местах, где линейный объект переходит в площадной.

3. *Повышение чувствительности локального анализа* – если в процессе трассировки контура количество исходящих ветвей из точки локального анализа становится меньше либо равно единице, то яростный порог автоматически

уменьшается. Адаптация порога позволяет отслеживать на изображении даже очень тонкие (яркостно слабые) линии, если удалось «зацепиться» за такую линию при исходном пороге. Если же изначально установить минимальный порог, то появляется много мелких отрезков.

4. *Смена разрешающей способности* осуществляется в тех случаях, когда на изображении одновременно присутствуют крупные и мелкие объекты. Для этого создается несколько изображений таких, что каждое последующее сжимается вдвое. Процесс анализа локальных областей при сужении линии автоматически переходит на изображение с увеличенной разрешающей способностью и наоборот.

3.4. Разработка методов аппроксимации графической информации

В результате локального и фрагментарного этапов синтаксического анализа модель изображения содержит:

$M2 = \langle G(B), OT3, L \rangle$, где

$G(B)$ - матрица $N \times M$ цветовых параметров $\langle R, G, B \rangle$ пикселей;

$OT3$ - множество особых точек, содержащих координаты и направления ветвей, исходящих из них;

L - множество цепочек, принадлежащих линиям объектов.

Задача аппроксимации графической информации включает следующие шаги:

- цветовая сегментация изображения;
- выделение границ цветовых областей;
- аппроксимация контуров и границ отрезками прямых и дугами окружностей;
- аппроксимация фигур выпуклыми многоугольниками и синтез их скелетонов;
- формирование синтаксической модели изображения.

Сегментация цветных изображений. Цветовая сегментация осуществляется для того, чтобы выделять различные объекты на цветном изображении. От качества сегментации зависит и надежность их распознавания, в частности, при наличии теней от других объектов или при различной ориентации объектов относительно луча зрения и источника света.

Аппроксимация цветовой палитры, осуществляемая в результате кластеризации, может быть использована также для сжатия изображений в тех случаях, когда необходимо сократить трафик передачи видеопотока.

Алгоритм цветовой сегментации изображения. Предлагаемый алгоритм цветовой сегментации изображения заключается в построении трёхмерной гистограммы количества пикселей в цветовом пространстве HSV. Здесь Hue - цветовой тон, варьируется в пределах $0—360^\circ$; Saturation — насыщенность, изменяется в пределах $0—100$ или $0—1$; Value (значение цвета) или Brightness — яркость, задаётся в пределах $0—100$ или $0—1$. Отличается выявлением "центров" кластеров путем поиска локальных максимумов с помощью сканирования пространства трёхмерным оператором анализа окрестности.

На рисунке 3.11 показан пример изображения, гистограмма его цветовой палитры и результат закраски полученной палитрой. На рисунке 3.11б: вверху – гистограмма (столбцы закрашены средним цветом), внизу – соответствующие значения составляющих компонентов цвета - S и V.

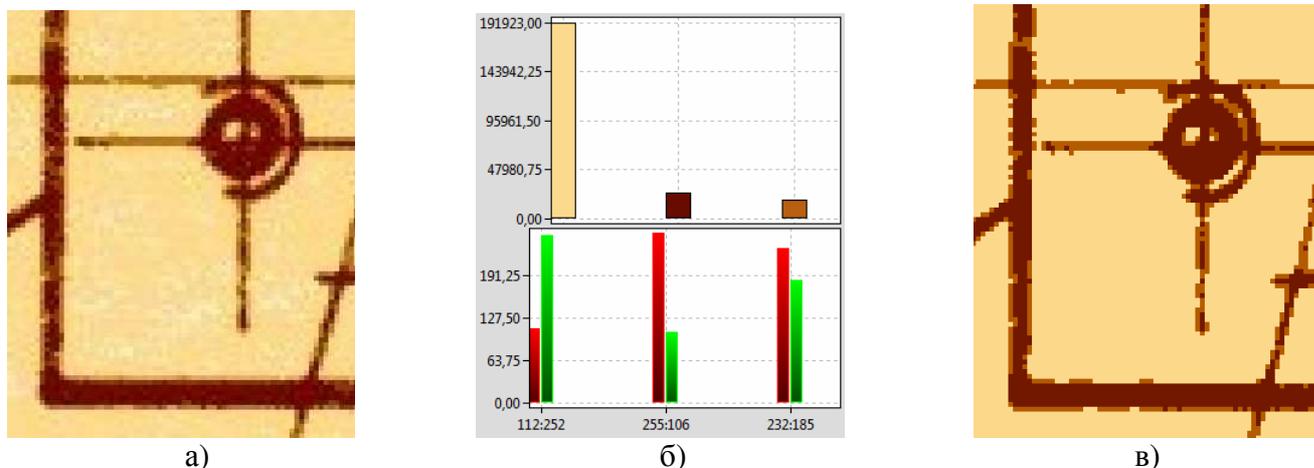


Рисунок 3.11 - Цветовая сегментация: а) фрагмент исходного изображения; б) созданные цветовые кластеры (для каждого кластера вверху отобран его цвет и количество отнесенных к нему пикселей исходного изображения, а внизу – его S- и V-компоненты); в) полученное тернарное изображение

Оператор поиска локальных концентраций объектов в трехмерном пространстве цветовых характеристик. Пространство HSV возможных значений с равномерным шагом по каждой из осей разбито на множество $W = \{W_{ijk}\}$, $i = 1, \dots, M1$, $j = 1, \dots, N1$, $k = 1, \dots, K1$ вокселей (подпространств) и вычислен вес w_{ijk} каждого из подпространств - количество пикселей изображения, у которых значения признаков попали в соответствующее подпространство.

Требуется найти множество E подпространств, в которых значение веса является локальным максимумом.

В отличие от существующих методов цветовой сегментации, данный алгоритм осуществляет поиск центров кластеров цветовой палитры изображения за 1 просмотр цветового пространства.

Алгоритм:

1. Квантование цветового пространства на заданное количество вокселей (подпространств). Вычисление весов и среднего цвета вокселей:

$$W_{ijk} = \langle w_{ijk}, g'_{ijk} \rangle,$$

где w_{ijk} - количество пикселей с координатами i, j, k ; $g'_{ijk} = \langle H_{cp}, S_{cp}, V_{cp} \rangle$ - цвет вокселя - усредненные значения параметров цветности пикселей, отнесенных к данному вокселю.

2. Выборка в качестве кластерообразующих локальных максимумов, т.е. таких вокселей, в которые попадает максимальное количество пикселей изображения. Если два таких вокселя расположены ближе заданного минимального расстояния между центрами кластеров, то подпространство с меньшим весом исключается из списка кластерообразующих.

Для каждого подпространства w_{ijk} сформируем вспомогательное множество W' соседних подпространств:

$$W'_{ijk} = \{w_{i-1jk}, w_{i+1jk}, w_{ij-1k}, w_{ij+1k}, w_{ijk-1}, w_{ijk+1}\}.$$

Тогда подпространство, в котором функция веса принимает значение локального максимума, определяется с помощью простого правила:

$$\bigvee_{l=1}^6 (w_{ijk} > w'_l) \Rightarrow w_{ijk} \in E$$

Таким образом, чтобы найти множество E локальных максимумов нужно просканировать таким оператором всё множество подпространств.

3. Чтобы осуществить кластеризацию на заданное количество цветов палитры, в которую вошли бы все воксели цветового пространства, а значит и все пиксели изображения, необходимо воксели, принадлежащие водоразделам, присоединить к ближайшему кластеру.

К кластерообразующим вокселям присоединяются соседние по трем направлениям воксели с меньшим весом до

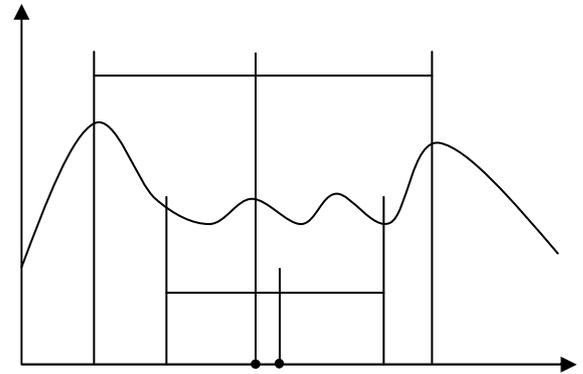


Рисунок 3.12 - Распределение водоразделов

тех пор, пока их веса не перестанут уменьшаться. Процесс анализа соседних вокселей с помощью локального оператора анализа окрестности производится рекурсивно. Проверяются разности весов (количеств пикселей) между центральным вокселем локального оператора и каждым из соседних не присоединённых вокселей. Если какая-либо разность меньше заданной пороговой величины, считается, что текущий центральный воксел локального оператора лежит на границе кластера в данном направлении.

Если между двумя "тяжелыми" кластерами имеется "легкий" (рисунок 3.12), который не вошел в заданную палитру, то он должен быть поглощен, т.е. разделен между ними.

Это можно сделать двумя способами:

- 1) по расстоянию от пика кластера, тогда горизонтальный участок притянется к большему по объему кластеру;
- 2) по расстоянию от края кластера, тогда горизонтальный участок водораздела будет равномерно делиться между соседними кластерами.

В первом случае каждый из оставшихся вокселей приписывается к ближайшему по расстоянию от вершины кластеру.

Во втором случае для каждого из оставшихся вокселей отыскивается ближайший по расстоянию воксел из числа записанных в кластеры на первом этапе и формируется дополнительный список МК, чтобы не исказились истинные границы кластеров. При этом вычислительная сложность алгоритма возрастает.

В обоих случаях средние цвета кластеров, полученные на первом этапе, не корректируются.

4. Если количество цветов палитры ограничено пользователем, то "лишние" кластеры с меньшими весами автоматически присоединяются к более весомым кластерам.

5. Для сжатия изображения может быть произведена цветовая аппроксимация изображения путем рекурсивной закрашки областей средневзвешенными цветами кластеров [76]. С этой целью все воксели содержат адресные ссылки на соответствующие им пиксели.

Переход на следующий, более высокий уровень анализа осуществляется для большей детализации объектов изображения путем смены соответствующих порогов чувствительности или разрешающей способности (степени сжатия). Например, при анализе растительности на космических снимках цвет меняется незначительно, а насыщенность и интенсивность плавают почти по всему диапазону. Для этого процесс кластеризации можно повторять в каждой из полученных областей изображения, что дает возможность пользователю более детально анализировать интересующие его области.

Например, если на изображении выделены водные объекты, то можно провести повторно процесс кластеризации только для водных объектов, с целью более детального изучения их структуры (границ, глубин, мелей).

Выделение границ цветовых областей осуществляется автоматически в ходе рекурсивного рассмотрения. Пиксел принадлежит границе цветовой области в тех случаях, когда хотя бы один из соседних пикселей принадлежит другому кластеру цветовой палитры:

$$\forall x = 1..N, y = 1..M, (\forall i = 1..8, \exists(g'(x, y) \neq g'(x + IXI(i), y + IYI(i))) \Rightarrow g'(x, y) \in OE,$$

где IXI, IYI - приращение координат в пределах окрестности 3×3 (см. 3.1);
 $g'(x, y)$ - цвет пиксела, полученный в результате цветовой сегментации;

OE - множество пикселей, принадлежащих границам цветовых сегментов.

Аппроксимация контурных и граничных линий отрезками прямых и дугами окружностей. С целью сокращения объема информации и, главное, для удобства анализа последовательности контурных и граничных точек, они аппроксимируются отрезками прямых и дуг. Точность аппроксимации определяется максимальным расстоянием точек отрезков от соответствующих точек реального контура. Расстояние задается в пикселах исходного изображения. По сути, это единственная настройка данного этапа.

Предварительно цепочка пикселей преобразуется в последовательность прямолинейных отрезков длиной 2 - 4 пиксела: $O_j = \langle x_s, y_s, x_e, y_e \rangle$, в которой координаты x_e, y_e конца отрезка совпадают с координатами x_s, y_s начала следующего отрезка.

Алгоритм принимает на вход цепочку $O = \langle O_1, \dots, O_n \rangle$ отрезков, имеющих примерно равную длину, и возвращает цепочку дуг и отрезков $E = \langle E_1, \dots, E_m \rangle$, обладающую следующими свойствами:

- $m \leq n$;
- элементы E_i удалены от исходной цепочки не более чем на заданный порог $d \in \mathcal{R}^+$;
- концы элементов E_i лежат на исходной цепочке, т.е. $\forall i \exists j: E_i.EndPoint = O_j.EndPoint$;
- углы в E не сильно отличаются от соответствующих углов в O , т.е.

$$|\angle(T(E_i), T(E_{i+1})) - \angle(O_j, O_{j+1})| \leq da,$$

где $T(E_i)$ - касательный вектор к элементу E_i в его концевой точке.

$l \leftarrow 1, r \leftarrow n$

ВыделитьДугу(l, r) // Рекурсивная процедура.

// l и r – границы рассматриваемой

// подцепочки.

```

{
   $v \leftarrow 0, w \leftarrow 0$ 
  Определить  $s \in [l, r-1]$ , такое что  $|\angle(O_s, O_{s+1})| \rightarrow \max$ 
  for  $i := s$  downto  $l$  do
    for  $j := s+1$  to  $r$  do
       $E \leftarrow$  ВычислитьПараметрыДугиИлиОтрезка( $i, j$ )
      if  $E$  удовлетворяет по отклонению и углам стыка then
         $v \leftarrow i, w \leftarrow j$ 
  if  $v \neq 0$  then
     $E \leftarrow$  ВычислитьПараметрыДугиИлиОтрезка( $v, w$ )
    Добавить  $E$  в результат
  else
    Добавить  $O_s$  в результат
  if  $s > 1$  then
    ВыделитьДугу( $l, s$ )
  if  $s+1 < r$  then
    ВыделитьДугу( $s+1, r$ )
}

```

Процесс аппроксимации имеет рекурсивный характер: изначально на всей цепочке определяется место экстремума (максимальный угол), из этого места происходит построение оптимальной аппроксимирующей дуги/отрезка, а затем процесс рекурсивно повторяется для подцепочек слева и справа от созданной дуги.

Пример. На исходном изображении (рисунок 3.13а, 284×356) выделено 6 цепочек, описывающих объекты общей длиной 2005 пикселей. Далее эти цепочки были обработаны с допустимой величиной отклонения (ошибки) аппроксимации 1 пиксел (рисунок 3.13б: белые – исходные цепочки, черные линии – результат аппроксимации). Получено 37 дуг и 5 прямых. На рисунке 3.13в максимальная

допустимая величина ошибки аппроксимации установлена 2 пикселя, получено 27 дуг и 1 отрезок.

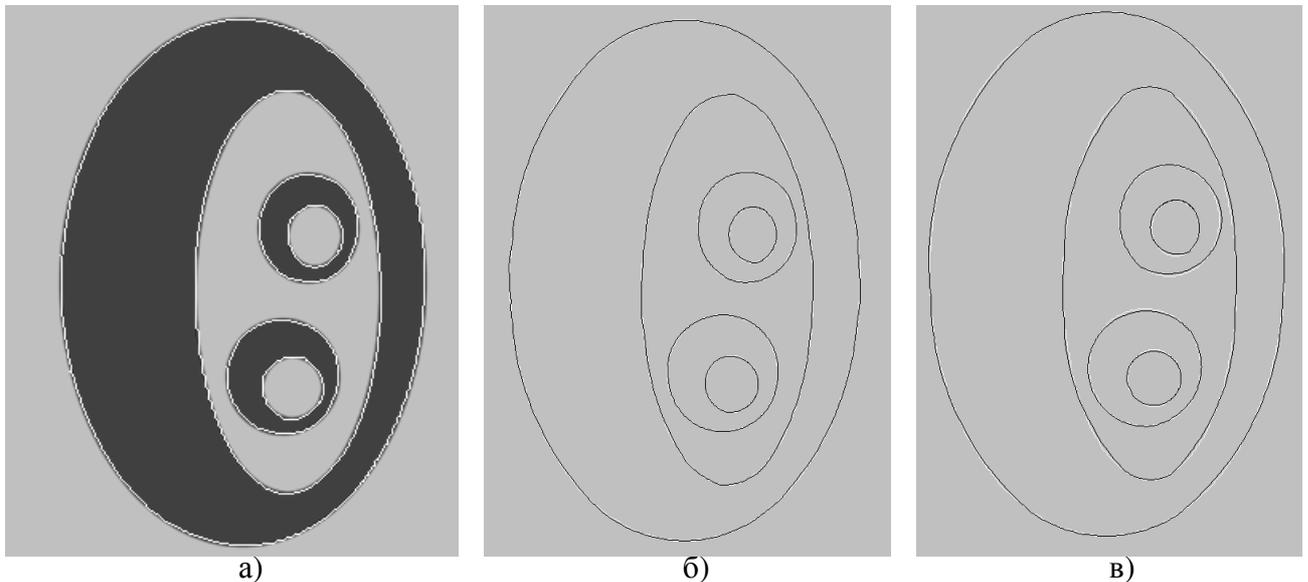


Рисунок 3.13 - Настройка на величину допустимой ошибки аппроксимации:
 а) исходное изображение с выделенными последовательностями пикселей;
 б) максимальная величина ошибки аппроксимации 1 пиксель;
 в) максимальная величина ошибки аппроксимации 2 пикселя

Аппроксимация плоских невыпуклых фигур.

Определение 1. Отрицательный перегиб границы – это такая вершина многоугольника, где при положительном обходе фигуры приращение угла вектора имеет отрицательное значение, т.е. происходит поворот вектора отрезка от площади фигуры (расширение площади, рисунок 3.14). Вектор отрезка направлен от начала отрезка к концу.

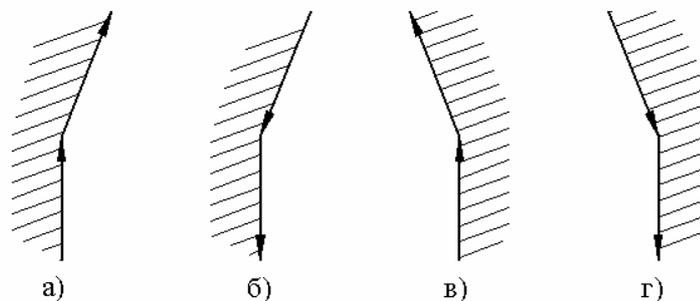


Рисунок 3.14 - Отрицательные перегибы границы

Положительный обход многоугольника – это последовательный обход его границы в таком порядке, что сумма приращений угла вектора равняется $+2\pi$, т.е. вектор совершает полный поворот против часовой стрелки.

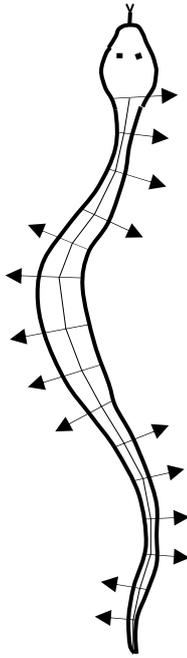


Рисунок 3.15 - Разбиение фигуры на выпуклые участки и построение скелетона

Метод выделения скелетона. С целью снижения вычислительной сложности и повышения качества выделения скелетона на широкополосных объектах разработан метод, заключающийся в рекурсивном отсечении сегментов фигуры прямолинейными отрезками в местах, где граница имеет отрицательный перегиб, так, что оба конца каждого отрезка лежат на границах исходной фигуры, и объединении в цепочки осевых линий полученных выпуклых фигур. При этом длины секущих (рисунок 3.15) несут информацию о ширине подобласти [45].

Дано: замкнутая цепочка $O = \langle x_s, y_s, x_e, y_e \rangle$ - граница области; $da < 0$ - порог на величину отрицательного перегиба.

Требуется: множество элементов скелетона $ES = \langle l, x_s, y_s, x_e, y_e \rangle$, где l - длина сечения фигуры в точке x_s, y_s элемента скелетона.

Алгоритм:

Цикл i по элементам границы

{Вычисление приращения $delta$ угла вектора

Если $delta < da$

То {Вычисление вектора биссектрисы, направленного внутрь области;

Вычисление точки пересечения с элементом, лежащим на противоположной стороне;

Формирование границы секции;

Определение главной оси инерции секции;

Вычисление координат концов $\langle x_s, y_s, x_e, y_e \rangle$ оси секции;

Запись $es = \langle l, x_s, y_s, x_e, y_e \rangle$

}

}

Таким образом, рассмотренный здесь алгоритм заключается в разбиении фигуры на выпуклые (с заданным приближением) секции путем построения секущих лучей, что у человека, вероятно, происходит где-то на подсознательном уровне, интуитивно.

Вычислительная сложность алгоритма скелетизации с помощью секущих лучей не превышает $O(K^2)/2$, где K – количество отрезков границы объекта, т.е. на эффективность алгоритма существенно влияет качество аппроксимации границ отрезками прямых и дугами окружностей.

Аппроксимация цветowych областей и синтез скелетонов выполняется рекурсивно и зависит от желаемой степени детализации объекта. Вначале секущие строятся в местах, где имеется существенный отрицательный перегиб границы, например, более 30 градусов, затем он снижается до величины, заданной пользователем. Релевантность разбиения фигуры на секции зависит также от качества аппроксимации границ отрезками (рисунок 3.16).

Аппроксимация скелетонов отрезками прямых и дугами окружностей осуществляется аналогично аппроксимации контуров и границ.

В тех случаях, когда фигура изначально выпуклая, в качестве скелетона служит прямая, проходящая через геометрический центр тяжести под углом, равным преобладающему углу наклона отрезков, которые были до аппроксимации границ. Преобладающий угол отрезков вычисляется с помощью простой кластеризации значений углов.

На рисунке 5.3. показан пример построения нечеткого скелетона чертежа в условиях среднего качества изображения.

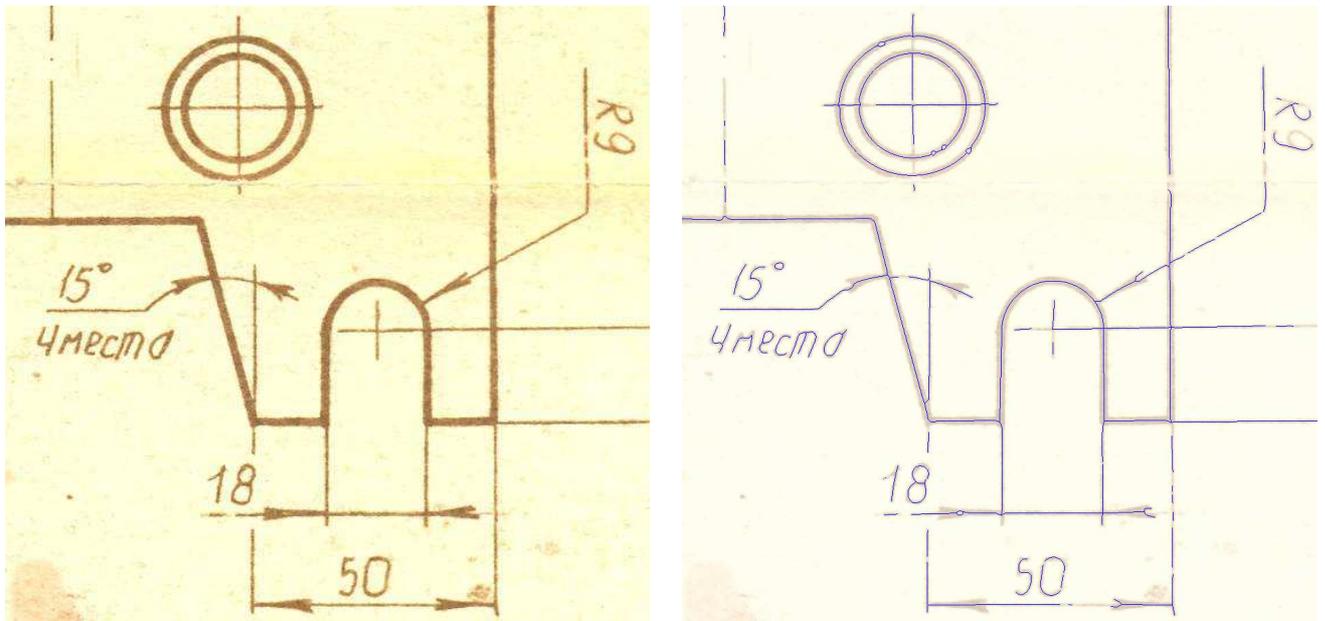


Рисунок 3.16 - Примеры работы алгоритма построения скелетона путем рекурсивной сегментации

3.5. Синтаксическая модель изображения

В результате выполнения предыдущих этапов обработки изображения получается синтаксическая модель изображения в виде набора множеств (в форме Бэкуса-Науэра):

$\langle \text{Изображение} \rangle ::= \{S\}$, где S – цветовой сегмент

$\langle \text{Цветовой сегмент} \rangle ::= Color, T, Perimeter, Square, Edge, [Skeleton]$

где: $Color$ – цвет (HSV), T – точка привязки (x, y), $Perimeter$ – периметр (суммарная длина отрезков в масштабе изображения), $Square$ – площадь (в пикселах), $Edge$ – граница (замкнутая цепочка из прямых и дуг), $Skeleton$ – скелет замкнутой области.

$\langle \text{Граница цветového сегмента} \rangle ::= \{E\}$ – множество элементов границы.

$\langle \text{Элемент границы} \rangle ::= form, xs, ys, xe, ye, xm, ym,$

где: $form \in \{Line, Arc\}$;

xs, ys – координаты точки начала элемента;

xe, ye – координаты точки конца элемента;

xm, ym – координаты точки середины дуги;

$\langle \text{Скелет цветového сегмента} \rangle ::= \{ES\}$ – множество элементов скелета.

$\langle \text{Элемент скелета} \rangle ::= l, form, xs, ys, xe, ye, xm, ym,$

где: l – длина сечения фигуры в точке начала элемента (в масштабе изображения).

Синтаксическая модель аппроксимированного векторизованного изображения представляется в виде многослойного атрибутивного графа (3.6) и сохраняется на языке XML.

Язык XML (eXtensible Markup Language) служит для текстового описания объектов и их атрибутов, а также для обмена данными между информационными системами [169].

Общая структура XML-описания изображения:

$\langle \text{AutoPhoto} \rangle$

$\langle \text{Image} \rangle$

$\langle \text{Edges} \rangle \dots \langle / \text{Edges} \rangle$

$\langle \text{Skeleton} \rangle \dots \langle / \text{Skeleton} \rangle$

$\langle \text{Contours} \rangle \dots \langle / \text{Contours} \rangle$

{ произвольное число узлов $\langle \text{Edges} \rangle$, $\langle \text{Skeleton} \rangle$, $\langle \text{Contours} \rangle$ }

$\langle / \text{Image} \rangle$

$\langle / \text{AutoPhoto} \rangle$

Узел $\langle \text{Image} \rangle$ содержит сведения об исходном растровом изображении.

Атрибуты узла $\langle \text{Image} \rangle$:

file - имя исходного растрового файла с расширением;

path - абсолютный путь к исходному растровому файлу;

sizeX, sizeY - размеры исходного растрового файла;

label - имя узла в дереве задач.

Узел $\langle \text{Edges} \rangle$ формируется при выполнении задачи отслеживания границ цветových сегментов и содержит сведения о границах выделенных цветových областей. Атрибуты узла $\langle \text{Edges} \rangle$:

colorR, colorG, colorB - RGB-компоненты цвета границы области;

label - имя узла в дереве задач.

Узел <Edges> имеет следующую структуру:

<Edges>

<Cell> ... </Cell>

<Cell> ... </Cell>

{ произвольное число узлов <Cell> типа «Граница цветовой области» }

<Edges> ... </Edges>

{ произвольное число вложенных списков границ областей }

</Edges>

Узел <Skeleton> формируется при выполнении задачи выделения скелетов.

Атрибуты узла <Skeleton>:

colorR, colorG, colorB - RGB-компоненты цвета скелетона;

label - имя узла в дереве задач.

Узел <Skeleton> имеет следующую структуру:

<Skeleton>

<Cell> ... </Cell>

<Cell> ... </Cell>

{ произвольное число узлов <Cell> типа «Цепочка скелетона» }

</Skeleton>

Узел <Contours> формируется при выполнении задачи выделения контуров. Атрибуты узла <Contours>:

colorR, colorG, colorB - RGB-компоненты цвета контура;

label - имя узла в дереве задач.

Узел <Contours> имеет следующую структуру:

<Contours>

<Cell> ... </Cell>

<Cell> ... </Cell>

{ произвольное число узлов <Cell> типа «Цепочка контура» }

</Contours>

Узел <Cell> содержит данные о цепочках геометрических примитивов, образующих границы цветовых областей, скелетоны и контуры, выделенные на изображении. XML-описание изображения может содержать четыре типа узлов <Cell>: 1) граница цветовой области; 2) граница полости; 3) цепочка скелета; 4) цепочка контура. Первые два типа узлов формируются по результатам выделения границ сегментов и соответствуют «внешней» и «внутренним» цепочкам примитивов, определяющим границы цветовой области. Узлы <Cell> типа «Цепочка скелета» содержат геометрические примитивы скелетной линии объекта. Узлы <Cell> типа «Цепочка контура» соответствуют контурам.

Атрибуты узла <Cell> типа «Граница цветовой области»:

fillColorR, fillColorG, fillColorB - RGB-компоненты цвета заливки области;

area - площадь цветовой области в пикселах без учета площади полостей;

length - длина цепочки примитивов;

id - идентификатор границы, уникальный в пределах родительского узла <Edges>, используется для учета вложенности границ цветовых областей;

parentID - идентификатор «родительской» границы. Если *parentID* = -1, данная область является «самой внешней» (таких областей может быть несколько), в противном случае содержит идентификатор внутренней цепочки (относящейся к другой цветовой области), для которой данная область является вложенной.

Узел <Cell> типа «Граница цветовой области» имеет следующую структуру:

<Cell>

<Node></Node>

<Node></Node>

{ ненулевое число узлов <Node> со значением атрибута *type* = 1 или 2 }

<Cell> ... </Cell>

<Cell> ... </Cell>

{ произвольное число узлов <Cell> типа «Граница полости», отсутствует, если цветовая область не содержит полостей }

<Skeleton> ... </Skeleton>

{ произвольное число узлов <Skeleton>, присутствует необязательно }

</Cell>

Атрибуты узла <Cell> типа «Граница полости»:

area - площадь полости в пикселах;

length - длина цепочки примитивов;

id – идентификатор цепочки, уникальный в пределах родительского узла <Edges>, используется для учета вложенности границ цветовых областей;

parentID - идентификатор «родительской» цепочки. Для внутренних цепочек (полостей) *parentID* содержит идентификатор *id* соответствующей внешней цепочки цветовой области.

Узел Cell типа «Граница полости» имеет следующую структуру:

<Cell>

<Node></Node>

<Node></Node>

{ ненулевое число узлов <Node> со значением атрибута *type* = 1 или 2 }

</Cell>

Узел <Cell> типа «Цепочка скелетона». Узлы этого типа являются потомками узла <Skeleton> и формируются по результатам выполнения задачи Skeleton. Атрибуты узла <Cell> типа «Цепочка скелетона»:

xB, *yB*, *xE*, *yE* - атрибуты добавляются в том случае, когда цепочка начинается с перекрестка (*xB*, *yB*) и/или заканчивается на разветвлении (перекрестке) с координатами (*xE*, *yE*). Все четыре атрибута отсутствуют, если скелет состоит из единственной цепочки и не содержит разветвлений;

length - длина цепочки примитивов.

Узел <Cell> типа «Цепочка скелетона» имеет следующую структуру:

<Cell>

<Node></Node>

<Node></Node>

{ ненулевое число узлов <Node> со значением атрибута *type* = 1 или 2 }

<Node type = "0"></Node> { завершающий узел }

</Cell>

Узел <Cell> типа «Цепочка контура». Узлы этого типа являются потомками узла <Contours>. Атрибуты узла <Cell> типа «Цепочка контура»:

length - длина цепочки примитивов.

Узел Cell типа «Цепочка контура» всегда имеет следующую структуру:

<Cell>

<Node></Node>

<Node></Node>

{ ненулевое число узлов <Node> со значением атрибута *type* = 1 или 2 }

<Node type = "0"></Node> { завершающий узел }

</Cell>

Узлы <Node> являются потомками узлов <Cell> и содержат координаты конечных точек геометрических примитивов в составе цепочки. Атрибуты узла Node:

type – тип геометрического примитива, *type* = 0 соответствует конечной точке незамкнутой цепочки геометрических примитивов, *type* = 1 соответствует отрезку прямой с началом в точке (*x*, *y*) и концом в точке, заданной следующим по порядку узлом <Node>, *type* = 2 соответствует дуге окружности с началом в точке (*x*, *y*) и концом в точке, заданной следующим узлом <Node>;

cross – длина в пикселах поперечного сечения области в точке (только для узлов скелетона);

x, *y* - координаты точки;

xC, *yC* - координаты центра дуги (только при *type* = 2);

dir – направление обхода дуги, $dir = 1$ соответствует направлению дуги против часовой стрелки, $dir = -1$ соответствует направлению дуги по часовой стрелке.

Узлы <Node> не содержат потомков и представляют собой конечные узлы дерева XML-описания изображения.

Преобразование количественных значений атрибутов в качественные (словесные) осуществляется путем формирования лингвистического представления (лингвистической модели) атрибута.

Ниже приведены примеры лингвистических моделей атрибутов Величина, Удлиненность, Направление.

Лингвистическое представление $LV(x)$ атрибута x – это лингвистическая переменная, комплексная модель атрибута, формализующая представление эксперта об атрибуте. Используется математическое определение из теории нечеткой логики: $LV(x) = \{x, T(x), X, G, M\}$. Поясним на примере атрибута *направление* (рисунок 4.9):

- x : «*направление*»;
- X : $[0^\circ, 360^\circ]$ – все возможные значения $QN(\text{направление})$;
- $T(x)$: значения «*Север*», «*Юг*», «*Запад*», «*Восток*»;
- G : «*Северо-*», «*Юго-*», «*Западо-*», «*Востоко-*» - такие добавки позволяют образовывать новые значения: «*Северо-Запад*», «*Восток-Юго-Восток*» и пр.;
- M : математическая функция или правило, определяющее принадлежность для каждого значения из множества $T(x)$, например, $M(\text{Север}) = \{0^\circ/1, \dots, 90^\circ/0, \dots, 270^\circ/0, \dots, 360^\circ/1\}$ (рисунок 3.17).

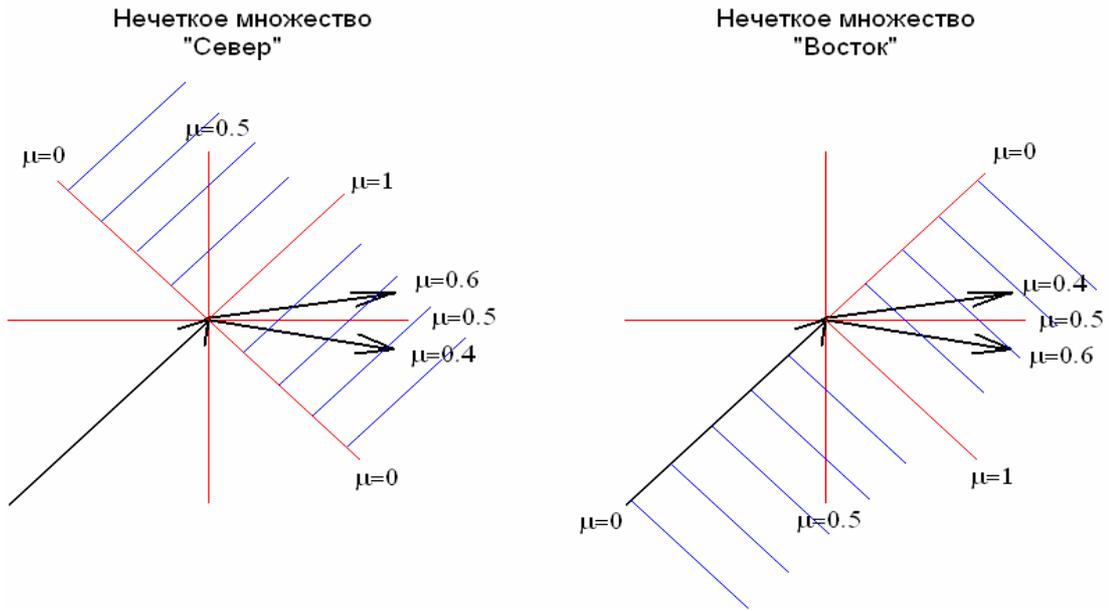


Рисунок 3.17 - Нечеткие значения в лингвистической модели атрибута *направление*

Лингвистическое представление позволяет на близком к естественному языке, в режиме диалога, задавать правила вывода и правила вычисления оценки сходства. Пример тривиального правила представлен на рисунке 3.18. Аппарат нечеткой логики позволяет интерпретировать подобные правила и получать в результате нечеткие значения, которые могут быть преобразованы в четкие числа. Следует отметить, что процедура преобразования нечеткого значения в четкое число является ресурсоемкой операцией.

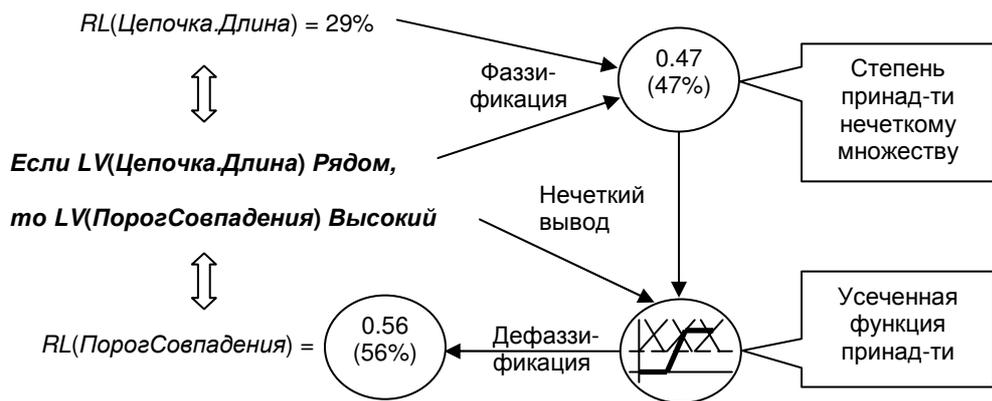


Рисунок 3.18 - Пример правила нечеткой логики и его выполнения

Таким образом, один и тот же атрибут имеет несколько форм представления, например, *длина цепочки*:

- 1) количественное значение в пикселах или миллиметрах;

- 2) относительное значение в виде процента от самой длинной цепочки или размера эталона;
- 3) комплексная форма в виде лингвистической переменной.

3.6. Выводы по главе

1. Для локального и фрагментарного этапов синтаксического анализа графической информации разработаны методы:

- анализа лучей, исходящих из центрального пиксела окрестности, для определения светлых и тёмных секторов;
- вычисления вектора контурной линии в заданной точке как средневзвешенного арифметического лучей, принадлежащих тёмному сектору окрестности;
- выделения особых точек: Конец Ветви, Разветвление, Угол на основе количества темных секторов в окрестности и углов между ними;
- выделения граничных точек площадных объектов в случаях, когда тёмный сектор в окрестности один и его центральный угол больше некоторой константы.

Перечисленные методы анализа локальной окрестности непосредственно связаны между собой и ориентированы на обработку таких изображений, где имеются плавные перепады яркостей и объекты различной ширины.

2. Разработаны методы фрагментарного этапа, анализирующего совокупности соседних окрестностей:

- метод оптимизации положения особых точек путем рекурсивного анализа области артефактов;
- метод уточнения положения особых точек, заключающийся в вычислении гипотетической точки пересечения участков исходящих ветвей;

- метод трассировки линий контуров, включающий скачкообразное перемещение в предварительно выделенную особую точку области артефактов изображения с целью сохранения геометрии объектов в этой области.

Методы контекстного анализа характеристик изображения приближены к алгоритмам биологических систем, что позволяет повысить качество выделения структурных элементов двумерных и трехмерных объектов на изображениях низкого качества.

3. Разработаны методы аппроксимации графической информации с целью сокращения избыточной информации:

- метод рекурсивной аппроксимации контуров, границ цветовых областей и скелетонов отрезками прямых и дугами окружностей с помощью рекурсивной процедуры слияния соседних отрезков и дуг;

- алгоритм цветовой сегментации изображения, заключающийся в построении трёхмерной гистограммы количества пикселей в цветовом пространстве HSV, отличающийся поиском локальных максимумов путем сканирования пространства трёхмерным оператором анализа окрестности. Выделение вершин кластеров цветовой палитры изображения осуществляется за 1 просмотр;

- метод аппроксимации плоских невыпуклых фигур путем выделения их скелетона, заключающийся в рекурсивном отсечении сегментов фигуры прямолинейными отрезками в местах, где граница имеет отрицательный перегиб, так, что оба конца каждого отрезка лежат на границах исходной фигуры, и объединении в цепочки осевых линий полученных выпуклых фигур. При этом длины секущих несут информацию о ширине подобласти.

4. Разработана синтаксическая модель аппроксимированного векторизованного изображения, содержащая множества цветовых областей, границ, скелетонов, контуров, отрезков и отношений "предок-потомок" между ними.

Таким образом, аппроксимация графической информации предназначена для замены объектов более простыми и удобными представлениями в задачах семантического анализа. Методика содержит шаги:

- цветовая аппроксимация изображения путем кластеризации цветовой палитры;
- выделение границ цветовых областей;
- аппроксимация контуров, границ цветовых областей дугами окружностей и отрезками прямых путём рекурсивного слияния соседних отрезков;
- синтез скелетов с помощью метода разбиения невыпуклых фигур на выпуклые многоугольники и сборки скелетона из осевых линий полученных многоугольников.

Методика позволяет регулировать точность аппроксимации (степень упрощения) с помощью параметров: количество цветов палитры, максимальная величина отклонения прямых и дуг от соответствующих объектов изображения, степень детализации областей, заменяемых скелетом. Также при вычислении нечётких параметров изображения с помощью лингвистических переменных, принимающих качественные (словесные) значения, выполняется фаззификация числовых характеристик в соответствующем метрическом пространстве.

4. РАСПОЗНАВАНИЕ ОБРАЗОВ, ЛОГИЧЕСКИЙ АНАЛИЗ И ВЕРБАЛИЗАЦИЯ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

В главе представлена синтаксическая модель изображения в виде многослойного атрибутивного графа, рассматриваются методы логического анализа, распознавания образов и вербализации: сопоставления моделей изображений, оценки сходства, дескрипционная логика семантического анализа и описания изображенных объектов, методика контроля корректности описаний, алгоритм синтеза классификаторов объектов.

4.1. Многослойный атрибутивный граф изображения

Для удобства последующего анализа, сопоставления и описания изображений формируется *модель изображения в виде многослойного атрибутивного графа*, содержащего несколько информационных слоев, связанных пространственными отношениями, где каждый слой состоит из вершин и ребер со своим набором атрибутов, с целью семантического анализа изображенных объектов и поиска аналогов:

$$MLAG = G(v^0, G^{SEGM}, \{G^{EDGE}\}, \{G^{SKEL}\}, G^{CONT}, \{R\}), \quad (4.1)$$

где: v^0 – корневая вершина с общими атрибутами изображения;

G^{SEGM} – граф цветowych сегментов;

$\{G^{EDGE}\}$ – множество графов границ сегментов;

$\{G^{SKEL}\}$ – множество графов скелетонov сегментов;

G^{CONT} – граф контуров изображения;

$\{R\}$ – множество отношений между слоями графа.

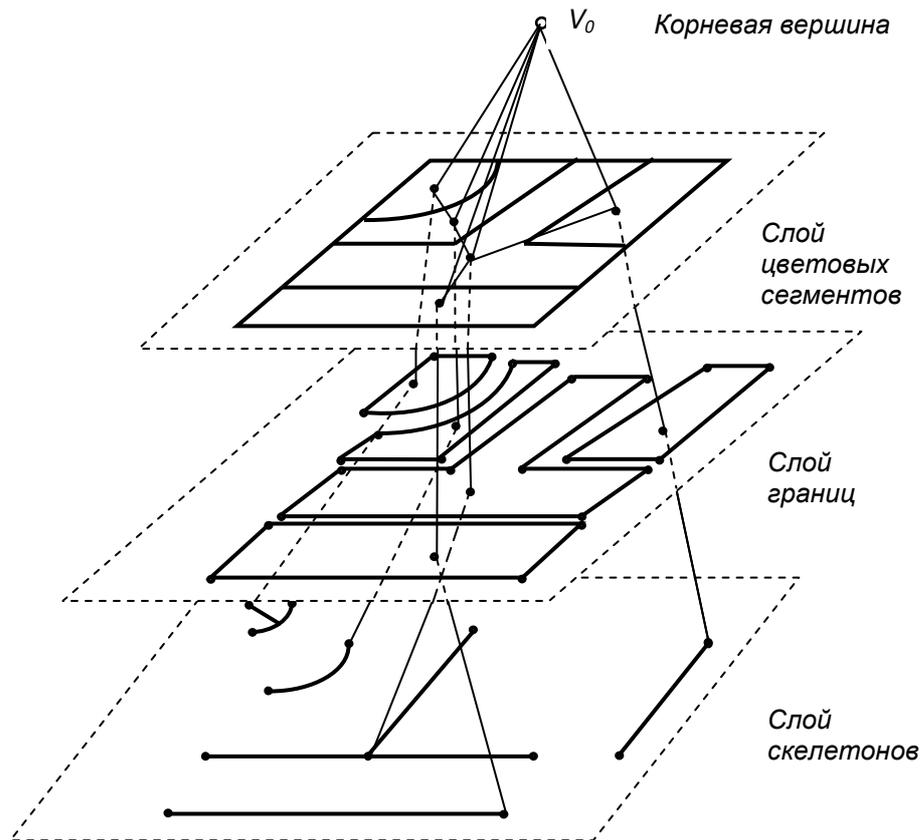


Рисунок 4.1 – Пример многослойного атрибутивного графа изображения

По аналогии с тематическими слоями в ГИС, многослойный атрибутивный граф *MLAG* (Multi Layer Attributive Graph) облегчает анализ информации (рисунок 4.1).

В общем случае, граф слоя (тематический атрибутивный граф) – это нечеткий пространственно нагруженный граф (НПНГ), вершины и ребра которого содержат количественные и качественные (нечеткие) значения геометрических параметров, далее нечеткий пространственно нагруженный граф:

$$G = (V, A, E, R), \quad (4.2)$$

где $v \in V$ – множество типовых опорных узлов (ТОУ): Конец Ветви, Разветвление, Угол, параметры которых $a \in A$ – это множество лингвистических переменных, принимающих значение из соответствующих нечетких множеств;

$e \in E$ – множество ребер (неориентированных дуг), отображающих пространственные отношения между ТОУ с помощью параметров $r \in R$ – множество лингвистических переменных, принимающих значения из нечетких множеств.

Множество атрибутов вершин:

$$A = AT \cup AN \cup AL,$$

где: AT – атрибуты, принимающие текстовые значения;

AN – числовые атрибуты;

AL – нечеткие (лингвистические) атрибуты.

Множество атрибутов рёбер:

$$R = RT \cup RN \cup RL,$$

где: RT – атрибуты, принимающие текстовые значения;

RN – числовые атрибуты;

RL – нечеткие (лингвистические) атрибуты.

К нечетким пространственным атрибутам вершин относятся, например, нечеткие координаты, к нечетким пространственным атрибутам ребер относятся, например, нечеткая длина и нечеткий угол в системе координат изображения.

Множество AL лингвистических переменных [18], описывающих вершины, определяется кортежами:

$$\langle a, X, Ca \rangle,$$

где a – наименование переменной;

X – область определения переменной – набор возможных значений x ;

$Ca = \{ \langle Ma(x)/x \rangle \}$ – нечеткое множество, описывающее степень принадлежности $Ma(x)$ некоторого числа лингвистическому значению x .

Множество $RL = \{ \langle r, Y, Cr \rangle \}$ лингвистических переменных, описывающих ребра графа изображения объекта, аналогично множеству AL для вершин.

Разработанная в ходе диссертационного исследования модель изображения в виде многослойного атрибутивного графа отличается от известных тем, что с целью получения более информативного описания изображений, включает границы областей, скелетоны, контуры и наборы их атрибутов формы и пространственной ориентации.

Методы аппроксимации графической информации, в частности, описанные в предыдущей главе, позволяют автоматически создавать синтаксическую модель

изображения в виде многослойного атрибутивного графа с заданной степенью детализации.

4.2. Метод сопоставления нечётких пространственно нагруженных графов

Лучевой граф как инструмент сопоставления нечетких пространственно нагруженных графов изображений. Введем понятие лучевого графа, предназначенного для анализа окружения заданной вершины на заданную глубину, определяющую длины цепочек, исходящих из этой вершины (рисунок 4.2).

Лучевой граф (ЛГ) с центром в точке v_0 для графа $G(I)$ - это граф

$$GB(v_0) = (VB, AB, EB, RB, RLA), v_0 \in V, VB \subseteq V,$$

$$AB \subseteq A, EB \subseteq E, RB \subseteq R,$$

где RLA – множество относительных нечетких углов дуг, вычисляемых:

- первый угол - против часовой стрелки от вектора входящей дуги, остальные – относительно ближайшей справа дуги (рисунок 4.3);

- если нет входящей дуги, то первый угол вычисляется относительно оси абсцисс.

Благодаря этому, независимо от того, как сориентирован объект на изображении, ЛГ обнаруживает соответствие с фрагментом эталонного объекта, причем значение угла, взятое из исходного НПНГ, позволяет определять различие в пространственной ориентации с точностью до заданной нечеткости.

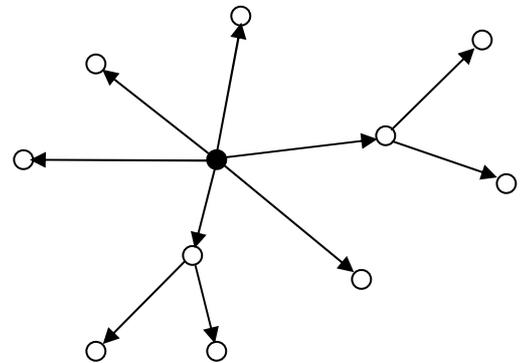


Рисунок 4.2 - Лучевой граф

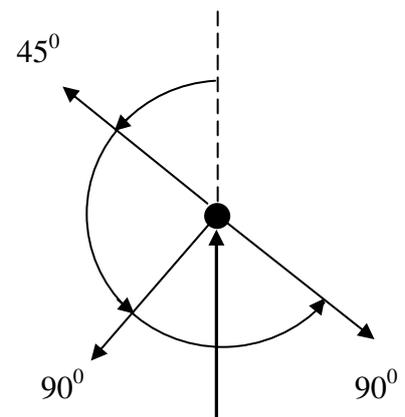


Рисунок 4.3 - Вычисление относительных углов в лучевом графе

В зависимости от поставленной задачи, можно изменять проекции множеств атрибутов $A \rightarrow AB$ и $R \rightarrow RB$ и, тем самым, находить совпадение графов по интересующим атрибутам.

Лучевой граф является одной из форм представления графа [18, 100].

Итак, лучевой граф (ЛГ) - это упорядоченное (нагруженное) дерево. Его характерной особенностью является то, что обязательной нагрузкой (кроме прочей) вершин и дуг ЛГ являются геометрические величины - координаты, углы, расстояния, радиусы кривизны и др., которые характеризуют геометрию и относительное расположение объектов изображения. Важнейшей пространственной характеристикой дуг ЛГ являются координаты дочерней вершины ЛГ в системе полярных координат родительской вершины.

Например, если взять атлас автомобильных дорог, то лучевой граф с центром в населенном пункте, где находится, автомобилист наглядно представляет все возможные маршруты относительно его текущего местонахождения.

Математический смысл понятия ЛГ состоит в том, что это не просто нагруженный граф, а *геометрический объект* - инвариант некоторой (в зависимости от вида геометрической нагрузки вершин и ребер) группы преобразований исходного изображения.

ЛГ с геометрической нагрузкой, рассматриваемой в данной работе, является инвариантным относительно группы аффинных преобразований, поскольку строится «из точки рассмотрения», т.е. из любой вершины графа изображения. Это означает, в частности, что при сдвигах, поворотах, масштабировании, симметриях исходного изображения соответствующие (построенные из «одной» вершины) ЛГ исходного и анализируемого изображений признаются «равными» (подобными до заданной нечеткости).

При «глубине» $d = 1$ лучевой граф превращается в *звездный граф* $G(a)$, определяемый вершиной a , который является подграфом графа $G(V)$ с множеством вершин V , если $a \in V$ [72]. Звездный граф состоит из

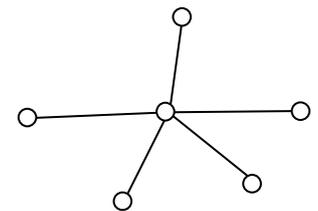


Рисунок 4.4 - Звездный граф

всех ребер $G(V)$, инцидентных вершине a (включая возможные петли), т.е. имеющих a концевой вершиной. Граф $G(a)$ можно назвать графом соседства (рисунок 4.4). Звездный граф можно также определить как **двудольный граф** $G(V_1, V_2)$, в котором множество V_1 состоит из одной вершины $a \in V$ и множество $V_2 \subset V$ является подмножеством множества V .

Лучевой граф как инструмент распознавания пространственных (не только типа «дерево») графов используется для поиска связных подграфов по заданному образцу, выявления сходства и различий.

Метод сопоставления анализируемого графа GA с графом GE эталона с помощью лучевых графов заключается в сопоставлении атрибутов всех вершин и ребер эталонного графа с одноимёнными атрибутами всех вершин и ребер анализируемого графа и отличается тем, что с целью повышения надежности сходство вершин оценивается путем сравнения их лучевых графов, в которых дуги упорядочены по атрибутам пространственной ориентации.

Алгоритм:

1. Установить исходные значения меток распознанных узлов эталона:

$$\forall(i \in [1, N])(found_i = 0).$$

2. Установить указатели узлов анализируемого графа для сравнения на первом шаге:

$$\forall(j \in [1, M])(pointer_j = 1);$$

$\forall(j \in [1, M])(conform_j = 0)$ – обнуление индексов соответствующих узлов эталона.

3. Цикл i по нераспознанным узлам эталона:

3.1. Установить шаг сравнения $Step = 1$.

3.2. Сформировать *лучевой граф EGB* из текущего узла эталона.

3.3. Установить $Flag = 'False'$ – признак совпадения эталонного узла.

3.4. Цикл j по узлам анализируемого графа, активизированным для сравнения на очередном шаге:

3.4.1. Сформировать *лучевой граф AGB* из узла анализируемого графа.

3.4.2. Если $Compare(EGB, AGB, Difference)$, то

3.4.2.1. Записать факт совпадения в слой соответствий анализируемого графа: $conform_j = Step$, номер шага $Step$ - это номер вершины эталона с учетом упорядоченности цепочек.

3.4.2.2. $Flag = 'True'$ – есть хотя бы одно совпадение эталонного узла.

3.5. Цикл k по узлам анализируемого графа, совпавшим с данным эталонным узлом, для разметки узлов, которые можно сравнивать на следующем шаге:

3.5.1. Цикл l по ветвям совпавшего узла в анализируемом графе:

3.5.1.1. Если тип видимого узла равен "конец ветви",

то $conform_j = Step$ пометить его как совпавший,

иначе установить указатель $pointer_j = Step+1$, где j – номер узла анализируемого графа, который будет сравниваться на следующем шаге.

3.6. Если $Flag = 'True'$ – произошло хотя бы одно совпадение, то:

3.6.1. $(found_i = 1) \wedge (WorkStep = Step) \wedge (Inc(Step))$ – увеличить номер шага сравнения, сохранив предыдущее значение в $WorkStep$ для возврата к сравнению по другой ветви в эталонном графе.

3.6.2. Цикл по ветвям эталонного узла:

3.6.2.1. Если следующий эталонный узел еще не распознан, то если тип этого узла равен "конец ветви",

то в слое $Found$ пометить его как совпавший,

иначе $Step = WorkStep$ – восстановление номера шага при переходе в цикле к очередной ветви.

4. Вычислить оценку сходства эталона:

$$ETQuality.Q = \sum_{i=1}^n found_i$$

5. Выход.

Отметим, что для случаев, когда графы искомым объектов имеют относительно простую структуру (10 – 20 узлов), эталоны изначально строятся в виде лучевых графов и тогда алгоритм сравнения упрощается.

Кроме того, функция сопоставления двух лучевых графов (п. 3.4.2 алгоритма) серьезно упрощается: если установлен единый порядок обхода ветвей (например, против часовой стрелки), то достаточно сравнить символьные коды этих графов.

В результате работы алгоритма вершины анализируемого графа помечаются индексами вершин эталонного графа, что очень удобно для визуализации процесса. При этом короткие цепочки, возникающие при «ложных» совпадениях, поглощаются (покрываются) более длинными. Например, если какой-то фрагмент эталонного графа совпадает одновременно с несколькими в анализируемом графе, то в слой соответствий *conform* заносятся номера узлов эталона, образовавших более длинные цепочки непрерывных совпадений в анализируемом графе и они замещают номера коротких цепочек.

В существующих методах в процессе сопоставления атрибутивных графов вершины сравниваются по своим атрибутам. В ходе сопоставления строится дерево поиска. При неудаче осуществляется «откат» к предыдущей совпавшей вершине. В случае сложных графов, таких как чертежи, это заметно ухудшает производительность. В данном алгоритме сравниваются вершины вместе с их соседями. Дерево поиска строится также, но откатов существенно меньше.

4.3. Поиск и сопоставление графической информации, представленной нечёткими пространственно нагруженными графами

Рассмотрим задачу сопоставления графов на примере поиска чертежей в базе конструкторско-технологических данных. Процесс формирования запроса на поиск изображения по заданному образцу представим в виде BPMN диаграммы (рисунок 4.5).

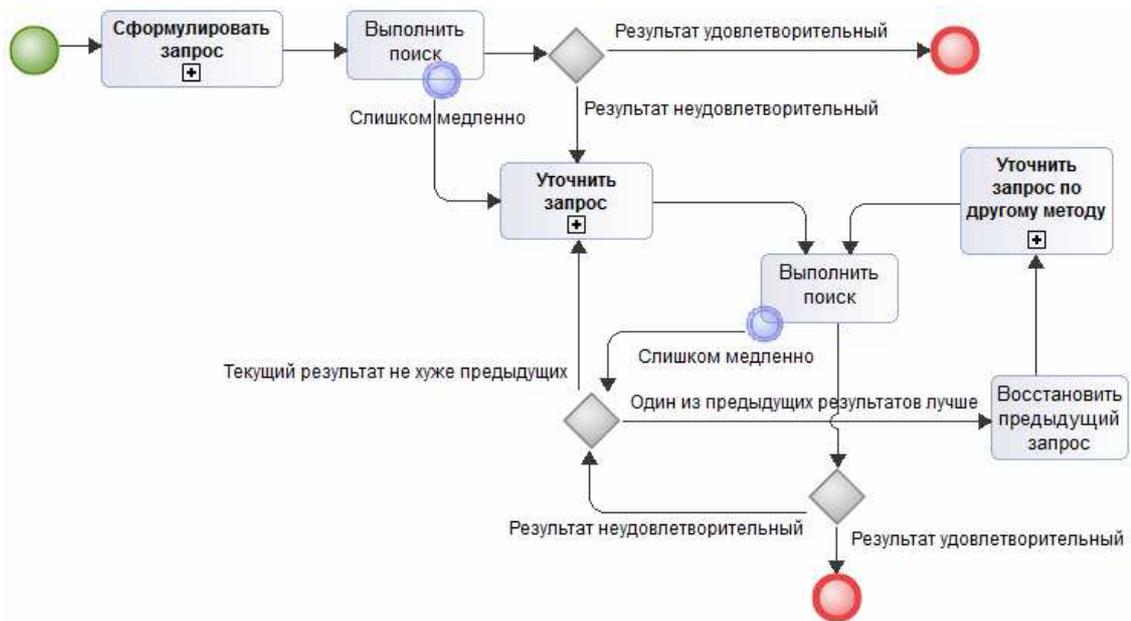


Рисунок 4.5 – Общая стратегия пользователя

Из рисунка видно, что в ходе поискового сеанса часто запрос приходится уточнять.

К результатам поиска можно задать общие требования, такие как:

- а) минимальная доля чертежа-запроса/искомого чертежа, которую должен составлять совпавший фрагмент;
- б) минимальное суммарное подобие совпавших фрагментов;
- в) выявление одного или всех вхождений чертежа-запроса в искомый чертеж;
- г) пороги совпадения атрибутов графических элементов (размер, ориентация и т.п.).

Чтобы порог минимального размера совпавшего фрагмента запроса лучше ощущался пользователем, визуализируется соответствующая доля составляющих, прилегающих к базовому узлу.

Также можно сократить пространство поиска правилами отбрасывания по признакам, определяемым поисковой системой в процессе индексации чертежей:

- а) текстово-числовые атрибуты;
- б) габариты;
- в) симметричность;
- г) наличие/отсутствие разрезов и выносок;
- д) количество проекций;

- е) словесно выраженная приблизительная форма проекции: круглая, полукруглая, Т-образная, квадратная, прямоугольная и т.п.;
- ж) наличие/отсутствие изолированных элементов внутри проекционного вида;
- з) наличие длинной линии обрыва;
- и) наличие таблиц;
- к) тип чертежа по классификации поисковой системы: множество плотно размещенных объектов и фрагментов, присутствие одинаковых или очень похожих объектов и т.п.

Благодаря содержательному представлению чертежей в поисковом образе чертежа-запроса можно задать специальные требования к результатам поиска.

Можно задавать требования к атрибутам отдельных составляющих в форме степени подобия/отклонения от атрибута запроса или списка/диапазона допустимых значений. В компьютерной семантике [145] это означает, что задействованные в сравнении атрибуты (гребенка поисковых признаков) и их веса определяют уровень абстрагирования. Например, искать сходство только по значениям углов прямой, тупой, острый или только по длинам.

Когда в качестве запроса используется эскиз, то обычно ожидается, что на чертежах-результатах будет присутствовать все, что изображено в этом эскизе. Однако дело может обстоять иначе в случае использования существующего чертежа в качестве запроса: наличие одних элементов в искомым чертежах желательно, но не обязательно, другие элементы следует совсем игнорировать. По этой причине предусмотрена возможность указания обязательных и исключаемых из рассмотрения составляющих. Также можно помечать элементы, которых на искомым чертежах быть не должно.

Для сравнения нечетких графов эталона и анализируемого графа используется метод, описанный в п. 4.2. При этом глубина лучевых графов эталона и анализируемого графа может ограничиваться заранее заданной величиной.

Оценки сходства. Пусть заданы критерии сходства графов G_1 и G_2 в виде:
- набор A анализируемых атрибутов вершин и ребер;

- набор T актуальных особых точек (напр., разветвления и концы ветвей или типовые конструктивные элементы чертежа).

Для ускорения поиска в базе данных функция графической индексации формирует поисковые образы изображений в виде нечетких графов:

1. Удаление атрибутов не участвующих в поиске.
2. Сокращение вершин и преобразование цепочек, проходящих через эти вершины, в простые ребра.
3. Аппроксимация и сглаживание ребер.
4. Фаззификация заданных атрибутов.
5. Формирование лучевого графа.

Цель формирования нечеткого графа путем фаззификации значений атрибутов – ускорение поиска без снижения релевантности результатов.

Как известно, в ходе фаззификации некоторого числового атрибута он заменяется лингвистической переменной, принимающей нечеткие (качественные) значения из заданного ряда. Благодаря этому, для того чтобы сказать, что вершина v_{1i} графа G_1 , эквивалентна вершине v_{2j} графа G_2 , достаточно проверить равенство всех их атрибутов. Так же для ребер. При этом лучевой граф предопределяет порядок сопоставления вершин. А именно:

1. Сравнение корневой вершины v_{10} со всеми $v_{2j} \in V_2$.
2. Сравнение лучей G_1 с лучами, исходящими из совпавшей вершины v'_{2j} .

Тогда, сходство полученных графов G_1 и G_2 выражается оценкой:

$$F(G_1, G_2, A, T) = L'_1/L_1 * 100\% \quad (4.3.)$$

где: L_1 – количество вершин и ребер в графе G_1 ;

L'_1 - количество вершин и ребер, совпавших с вершинами и ребрами графа G_2 .

Если в сопоставляемых графах пути (цепочки) между заданными особыми точками из набора T могут различаться по количеству ребер, то целесообразно вначале просуммировать значения их атрибутов вдоль таких путей, а уже потом выполнять фаззификацию для получения нечетких значений.

Оценка сходства графов G_1 и G_2 по количеству совпавших особых (peculiar) вершин и путей:

$$FP(G_1, G_2, A, T) = P_1'/P_1 * 100\% \quad (4.4)$$

где: P_1 – количество вершин из набора T и путей между ними в графе G_1 ;

P_1' – количество вершин и путей, совпавших с графом G_2 .

Следующая оценка сходства дает более релевантные результаты, когда при поиске аналогов требуется учитывать комплексную характеристику путей.

Вычислим взвешенную сумму значений атрибутов из набора A всех ребер вдоль цепочки (v_i, v_j) , соединяющей вершины v_i, v_j :

$$W(v_i, v_j) = \sum_{k=i}^{k=j} \sum_{l=1}^m a_{kl} * w_l$$

где a_{kl} – значение l -того атрибута k -того ребра;

w_l – вес атрибута заданный экспертом или пользователем.

Введем меру близости D цепочек (v_{1i}, v_{1j}) и (v_{2m}, v_{2n}) в графах G_1 и G_2 соответственно как разность взвешенных сумм их атрибутов:

$$D = |W(v_{1i}, v_{1j}) - W(v_{2m}, v_{2n})|$$

Совпадение цепочек:

$$C = \begin{cases} 1, & \text{если } D \leq \delta \\ 0, & \text{иначе} \end{cases}$$

Оценка сходства графов по весу совпавших путей:

$$FW(G_1, G_2, A, T) = \sum W_1' / \sum W_1 * 100\%$$

где: $\sum W_1$ – общий вес всех путей в графе G_1 ;

$\sum W_1'$ – вес путей из G_1 , совпавших с графом G_2 .

Пользователь имеет возможность выбирать критерии оценки сходства и устанавливать порог на величину этой оценки для того, чтобы система считала найденное изображение удовлетворительным и включала его в результирующую выдачу.

Таким образом, при заданной глубине лучевого графа алгоритм распознавания на каждом шаге «видит» одновременное совпадение текущей вершины, всех исходящих из нее ребер и вершин, инцидентных этим ребрам

Очевидно, чем больше глубина лучевых графов, тем точнее оценка сходства и выше надежность получения адекватного ответа на запрос о сходстве графов. Но выше объём вычислений.

4.4. Логический анализ и вербализация графической информации

Дескрипционные или описательные логики относятся к языкам представления знаний, позволяющим описывать понятия предметной области в формализованном виде [114]. С помощью логических операций, правил и аксиом проверяется корректность определений и иерархий построения сложных понятий из простых.

Дескрипционные логики (ДЛ) служат для:

- формального определения понятий (классов);
- определения отношений между понятиями;
- организации иерархий (подкласс – суперкласс);
- определения свойств и атрибутов понятий и ограничений на их значения.

Дескрипционные логики нашли широкое применение в различных аналитических и поисковых системах. Достоинство дескрипционных логик заключается в том, что с их помощью эксперт описывает термины и определения понятий предметной области, а машина вывода (reasoner) системы осуществляет контроль корректности определений и поиск соответствующих экземпляров в разделе данных.

Дескрипционные логики работают с понятиями и отношениями между ними (ролями):

- синтаксис задает правила конструирования сложных понятий из простых;
- семантика - это логическая интерпретация сложных структур на базе фактов.

В данной работе для семантического анализа и формализованного описания графической информации предлагается использовать дескрипционную логику

ALC, расширив ее на область данных, представленных в виде атрибутивных графов, назовем её *ALC(GI)*. Назначение этой логики – сокращение области интерпретации для задач с графовой структурой данных. Кроме того, для задач анализа изображений в состав атомарных концептов и ролей вводятся предикаты на основе геометрических функций.

Таким образом, дескрипционная логика *ALC(GI)* - это расширение дескрипционной логики *ALC* для работы с данными, представленными в виде графов (элементов и связей между ними).

Логика *ALC* (Attributive Language with Complement) [114] является одной из базовых ДЛ, на основе которой строятся многие другие ДЛ для работы с базами знаний и онтологиями в различных предметных областях. Собственно *ALC* – это не одна логика, а язык и правила построения логик, с помощью которых создано целое семейство логик, где каждая отличается набором множеств атомарных концептов и ролей.

Синтаксис дескрипционной логики *ALC* задает правила в виде допустимых конструкций составных концептов.

Чтобы сформулировать синтаксис какой-либо ДЛ, необходимо задать непустые (и обычно конечные) множества символов, так называемых атомарных концептов и атомарных ролей, из которых будут строиться выражения языка данной логики. ДЛ характеризуется набором конструкторов и индуктивным правилом, с помощью которого составные концепты данной логики строятся из атомарных концептов и атомарных ролей, используя эти конструкторы. Типичными конструкторами для построения составных концептов являются:

- пересечение (или конъюнкция) концептов, обозначается как $C \sqcap D$;
- объединение (или дизъюнкция) концептов, обозначается как $C \sqcup D$;
- дополнение (или отрицание) концепта, обозначается как $\neg C$;
- ограничение на значения роли (или ограничение квантором всеобщности), обозначается как $\forall R.C$;

- экзистенциальное ограничение (или ограничение квантором существования), обозначается как $\exists R.C$;

- численные ограничения на значения роли, например: $\leq n R$, $\geq n R.C$, задающие количество R -последователей, удовлетворяющих отношению R , и другие.

Существуют дескрипционные логики, в которых имеются также составные роли, строящиеся из простых ролей с помощью операций: инверсии, пересечения, объединения, дополнения, композиции ролей, транзитивного замыкания и других [104].

Множество концептов логики ALC задается следующим индуктивным правилом:

- всякий атомарный концепт является концептом;
- выражения \top и \perp являются концептами (*истина* и *ложь*);
- если C есть концепт, то его дополнение $\neg C$ является концептом;
- если C и D есть концепты, то их пересечение $C \sqcap D$ и объединение $C \sqcup D$ являются концептами;
- если C есть концепт, а R есть роль, то выражения $\forall R.C$ и $\exists R.C$ являются концептами, задающими общее ограничение на значения.

Семантика логики ALC – это интерпретация, которая ставит в соответствие концепту все удовлетворяющие ему экземпляры (индивиды). Интерпретирующая функция распространяется на составные концепты логики по следующим правилам:

- \top интерпретируется как весь домен: $\top^I = \Delta^I$, где Δ - область интерпретации (множество экземпляров), I – интерпретирующая функция;

- \perp интерпретируется как пустое множество: $\perp^I = \emptyset$;

- дополнение концепта интерпретируется как дополнение множества:

$$(\neg C)^I = \Delta^I \setminus C^I;$$

- пересечение концептов интерпретируется как пересечение множеств:

$$(C \sqcap D)^I = C^I \cap D^I;$$

- объединение концептов интерпретируется как объединение множеств:

$$(C \sqcup D)^I = C^I \cup D^I;$$

- выражение $\forall R.C$ интерпретируется как множество тех индивидов, у которых все R -последователи принадлежат интерпретации концепта C .

Формально:

$$(\forall R.C)^I = \{x \in \Delta^I \mid \forall y \in \Delta^I ((x, y) \in R^I \Rightarrow y \in C^I)\}$$

- выражение $\exists R.C$ интерпретируется как множество тех индивидов, у которых имеется R -последователь, принадлежащий интерпретации концепта C .

Формально:

$$(\exists R.C)^I = \{x \in \Delta^I \mid \exists y \in \Delta^I ((x, y) \in R^I \wedge y \in C^I)\}$$

База знаний состоит из:

- набор определений - терминологических аксиом $TBox$;
- набор утверждений об индивидах $ABox$ - аксиом вложенности и эквивалентности.

Аксиомой вложенности концептов называют выражение вида $C \sqsubseteq D$, а аксиомой эквивалентности концептов $C \equiv D$.

Аналогичным образом записываются аксиомы вложенности ролей $R \sqsubseteq S$ и эквивалентности $R \equiv S$.

4.4.1. Дескрипционная логика с расширением на данные, представленные в виде атрибутивных графов

Для задач семантического анализа изображений и формального описания графической информации, в состав атомарных концептов и ролей вводятся предикаты на основе геометрических функций. Предполагается, что задачи, в настоящее время решаемые на графах, можно использовать в качестве атомарных концептов и ролей при моделировании рассуждений.

Расширение логики языка $ALC(GI)$ на область данных в виде атрибутивных графов позволяет упростить описание запросов на поиск и сократить область интерпретации для задач с графовой структурой данных. Например, вместо перебора множества пикселей [176, 127, 160] или множества отрезков будут проверяться только цепочки, соединяющие пары вершин.

Каждому графу соответствует домен – область интерпретации ДЛ-описаний. Домен «Изображение» является расширением домена «Граф» в части атрибутов вершин и ребер. В число производных (атомарных) понятий предлагаемой логики $ALC(GI)$ входят понятия, интерпретируемые на графах, и понятия вычисляемые на данных изображения:

Атомарные концепты на графах:

C1) *Вершина*(x) – x является вершиной графа, его интерпретация – множество вершин рассматриваемого графа;

C2) *Дуга*(x) – x является дугой графа, интерпретация – множество дуг рассматриваемого графа;

C3) *ПодГраф*(x) – x является подграфом рассматриваемого графа, интерпретация – множество всех подграфов (без изолированных вершин) рассматриваемого графа;

C4) *СвязныйПодГраф*(x) – x является связным подграфом рассматриваемого графа. Поскольку изолированных вершин нет, концепт *ПодГраф* в стандартной интерпретации представляет множество всех подмножеств концепта *Дуга*.

Атомарные роли и обратные отношения, интерпретируемые на графах:

R1) *НачалоДуги*(x :*Дуга*, y :*Вершина*) – вершина y является началом дуги x , – функциональное отношение (атрибут дуги).

Обратная (определяемая) роль

R2) *ИсходящаяДуга*(x :*Вершина*, y :*Дуга*) – дуга y исходит из вершины x , т.е. x является началом дуги y .

R3) *КонецДуги*(x :*Дуга*, y :*Вершина*) – вершина y является концом дуги x , – функциональное отношение (атрибут дуги).

Обратная (определяемая) роль

R4) *ВходящаяДуга*(x :Вершина, y :Дуга) – дуга y входит в вершину x т.е. x является концом дуги y .

Некоторые простейшие производные понятия на графах:

Инцидентно(x :Вершина, y :Дуга) \equiv *ИсходящаяДуга* \sqcup *ВходящаяДуга* – дуга y инцидентна вершине x .

0_Вершина (x :Вершина) \equiv *Вершина* $\sqcap \neg \exists$ *Инцидентно.T* – x является вершиной степени 0 (изолированной).

n_Вершина (x :Вершина) \equiv *0_Вершина* $\sqcup \leq n$ *Инцидентно.T* – x является вершиной степени $\leq n$, $n \in \mathbb{N}$.

ПромежуточнаяВершина (x : Вершина) \equiv *2_Вершина* – является вершиной степени 2;

Цикл(x : ПодГраф) \equiv *СвязныйПодГраф* $\sqcap \neg \exists$ *1_Вершина* - не существует вершин степени один.

Дерево(x : Граф) \equiv *СвязныйПодГраф* $\sqcap \neg \exists$ *Цикл* - граф без циклов.

НачальнаяВершина(x : Граф, y : Вершина) \equiv *НачалоДуги* $\sqcap \neg$ *КонецДуги* – вершина y является начальной (не имеет входящих дуг) в подграфе x .

КонечнаяВершина(x : Граф, y : Вершина) \equiv \neg *НачалоДуги* \sqcap *КонецДуги* – вершина y является конечной (не имеет исходящих дуг) в подграфе x .

Цепочка(x :Граф) \equiv *Дерево* $\sqcap = 1.*НачальнаяДуга.T* $\sqcap = 1.*КонечнаяДуга.T* – подграф x является цепочкой (деревом без ветвлений).$$

Атомарные концепты и роли, интерпретируемые на изображениях:

Особенностью предлагаемого подхода к анализу изображений является то, что с помощью дескрипционной логики описываются не характеристики отдельных пикселей изображения, а **сегменты** изображения – связные множества пикселей с «близкими» характеристиками, их **границы, скелетоны и контуры**.

Выделение сегментов, построение их границ и скелетонов производится на этапе предварительной обработки изображения [30]. Элементы (отрезки) границ, скелетонов, контуров содержат атрибуты:

$$\langle form, xs, ys, xe, ye, xm, ym \rangle,$$

где: $form \in \{Line, Arc\}$ – прямая или дуга окружности;

xs, ys – координаты точки начала элемента;

xe, ye – координаты точки конца элемента;

xm, ym – координаты точки середины дуги.

Сегменты, границы, скелетоны, контуры образуют граф изображения (4.1), вершины и дуги которого нагружены геометрической и другой информацией, необходимой для анализа исходного изображения без обращения к пикселям.

IC5) *Сегмент* (x) – x является сегментом изображения – атомарный концепт, интерпретация – множество цветовых сегментов изображения, сформированных по принципу близости параметров пикселей в цветовом пространстве.

IR1) *XВершины* (x : *Вершина*, y : *Число*), – число y является координатой X вершины x – функциональное отношение (атрибут).

IR2) *YВершины* (x : *Вершина*, y : *Число*), – число y является координатой Y вершины x – функциональное отношение (атрибут).

IR3) *MXДуги* (x : *Вершина*, y : *Число*), – число y является координатой X середины дуги x – функциональное отношение (атрибут).

IR4) *MYДуги* (x : *Вершина*, y : *Число*), – число y является координатой Y середины дуги x – функциональное отношение (атрибут).

IR5) *ГраницаСегмента* (x : *Сегмент*, y : *МножествоЦиклов*), – подграф y является границей сегмента x – функциональное отношение (атрибут).
Интерпретация: окружность или замкнутая последовательность отрезков прямых и дуг, ограничивающий сегмент;

IR6) *СкелетонСегмента* (x : *Сегмент*, y : *Дерево*), – подграф y является скелетоном сегмента x – функциональное отношение (атрибут).

IR7) *ЧастьСегмента*(x : Сегмент, y : Сегмент), – сегмент y является частью (подмножеством множества пикселей) сегмента x – функциональное отношение (атрибут).

IR8) *ВнутриСегмента*(x :Сегмент, y :Сегмент) – сегмент y является частью (подмножеством множества пикселей) сегмента x и границы сегментов x , y не пересекаются, т.е. все точки границы y лежат внутри многоугольника, образованного границей x .

IR9) *ПлощадьСегмента*(x : Сегмент, y : Число), y - количество пикселей сегмента.

IR10) *ПериметрСегмента*(x : Сегмент, y : Число), y – длина границы сегмента x .

IR11) *ЦветСегмента*(x : Сегмент, y : Число), y - средний цвет пикселей сегмента x .

IR12) *Угол*(x : 2_Вершина, y : Число), – функция, возвращающая значение разности между углами наклона геометрических хорд входящей дуги и выходящей дуги.

IR13) *XСегмента*(x : Сегмент, y : Число), – координата X точки привязки сегмента x . Интерпретация – среднее значение координат x пикселей, принадлежащих сегменту X :

$$x = \left(\sum_{i=1}^n x_i \right) / n, \text{ где } n - \text{ количество пикселей сегмента.}$$

IR14) *YСегмента*(x : Сегмент, y : Число), – координата Y точки привязки сегмента x . Интерпретация - среднее значение координат y пикселей, принадлежащих сегменту x :

$$y = \left(\sum_{i=1}^n y_i \right) / n, \text{ где } n - \text{ количество пикселей сегмента.}$$

IR15) *ДлинаСегмента*(x : Сегмент, y :Число), – интерпретация: длина в пикселях описывающего прямоугольника вдоль оси инерции фигуры.

IR16) *ШиринаСегмента*(x : Сегмент, y :Число), – интерпретация: ширина в пикселях описывающего прямоугольника вдоль оси инерции фигуры.

IR17) *КоличествоПолостей*(x : *Сегмент*, y : *Число*), – интерпретация: сегмент x содержит y сегментов, для которых выполняется условие:

$$(z \in \text{Сегмент}S) \wedge \text{ВнутриСгмента}(x, z),$$

где S - множество сегментов изображения;

Внутри - функциональное отношение IR8.

IR18) *КоличествоРазветвлений*(x : *Скелетон*, y : *Число*), – интерпретация: y - количество вершин степени ≥ 3 в связном подграфе x , т.е. удовлетворяющих понятию ≥ 3 *Вершина*.

Для удобства описания объектов, имеющих на изображении, введем следующие **аксиомы эквивалентности**:

КонецВетви $\equiv 1$ *Вершина*;

Разветвление $\equiv \geq 3$ *Вершина*;

ОстрыйУгол $\equiv 2$ *Вершина* $\cap \exists$ *имеетУгол*.($Угол > \pi/4$) \vee ($Угол < -\pi/4$);

ОсобаяТочка \equiv *КонецВетви* \sqcup *Разветвление* \sqcup *ОстрыйУгол*;

Узел \equiv *Node* $\equiv \geq 2$ *Вершина*;

Петля \equiv *Loop* \equiv *Цикл*;

Chain \equiv *СвязныйПодГраф* $\cap \exists (=2. \text{ОсобаяТочка})$ – цепочка, начинающаяся и заканчивающаяся особыми точками, т.е. связный подграф, который содержит ровно две особые точки.

Как видим, в отличие от понятия *Цепочка*, которое ранее было определено на графе, *Chain* – цепочка на изображении может быть ограничена еще и особой точкой типа «острый угол». Это понадобится в дальнейшем для более надежного разделения объектов на изображении.

Таким образом, каждому изображению соответствует домен – область интерпретации ДЛ. Интерпретация атомарных концептов и ролей ДЛ определяется изображением, т.е. методами и алгоритмами преобразования изображения в атрибутивный граф.

4.4.2. Функциональные отношения (атрибуты), описывающие геометрические характеристики формы цветковых сегментов и их скелетонов

Ниже приведены атомарные функции анализа форм объектов, наиболее актуальные для анализа их изображений.

FR1) *ФормаСегмента*(x : *Сегмент*, y : *НечёткоеЗначение*) - вычисляет нечёткое значение *{Площадной, Широкополосный, Линейный}* на основе отношения площади к длине периметра.

FR2) *ChordLength*(x : *Chain*, y : *НечёткоеЗначение*) – вычисляет нечёткое значение *{Short, Medium, Long}* длины хорды, соединяющей концы цепочки скелетона.

FR3) *SegmentNumber*(x : *Chain*, y : *Number*) – число сегментов, образованных цепочкой скелетона слева и справа от хорды цепочки.

FR4) *Area*(x : *Chain*, y : *НечёткоеЗначение*) – вычисляет нечёткое значение *{Small, Medium, Large}* площади сегмента цепочки.

FR5) *LeftSegmentArea*(x : *Chain*, y : *НечёткоеЗначение*) – вычисляет нечёткое значение *{Small, Medium, Large}* площади сегмента цепочки слева от хорды, считая хорду направленной от начала цепочки к концу.

FR6) *RightSegmentArea*(x : *Chain*, y : *НечёткоеЗначение*) – вычисляет нечёткое значение *{Small, Medium, Large}* площади сегмента цепочки справа от хорды.

FR7) *EachSegmentArea*(x : *Chain*, y : *НечёткоеЗначение*) – вычисляет нечёткое значение *{Small, Medium, Large}* площади всех сегментов цепочки.

FR8) *MaxAreaHeight*(x : *Chain*, y : *НечёткоеЗначение*) – вычисляет нечёткое значение *{Small, Medium, Height}* максимальной высоты всех сегментов цепочки.

FR9) *TiltAngle*(x : *Chain*, y : *НечёткоеЗначение*) – вычисляет нечёткое значение *{Vertically, LeftTilt, RightTilt, UpTilt, Horizontally, DownTilt}* угла наклона хорды цепочки.

Хорда цепочки – это направленный отрезок, соединяющий ее концы.

FR10) *Удлиненность*(x : Сегмент, y : НечёткоеЗначение) - интерпретация: y - нечеткое значение лингвистической переменной *Elongation*, принимающей значения {Округлый, Овальный, Вытянутый, Линейный} вычисляется как лингвистическая переменная на основе отношения *ШиринаСегмента* (IR16) к *ДлинаСегмента* (IR15) при отсутствии скелетона объекта, иначе - отношение длины скелетона к максимальному сечению:

$$R = \begin{cases} \min(Lx, Ly) / \max(Lx, Ly), & \text{если } \{ES\} = \emptyset, \\ \sum_{i=1}^n \sqrt{(xs_i - xe_i)^2 + (ys_i - ye_i)^2} / \max(l_i) & \text{иначе} \end{cases}$$

FR11) *ИзвилистостьГраницы*(x : Сегмент, y : Число) выражается целым числом – количеством отрицательных перегибов границы (Определение 1, п.3.4) сегмента x .

FR12) *ПрямолинейностьГраницы*(x : Сегмент, y : Число) интерпретация: относительное значение суммарной длины прямолинейных отрезков в процентах от длины периметра (IR10):

$$LL := 0; \forall i \in [1..n](form_i = Line) \Rightarrow LL := LL + \sqrt{(xs_i - xe_i)^2 + (ys_i - ye_i)^2};$$

$$Straight = \frac{LL}{Perimeter} * 100\%$$

Эта величина также может быть преобразована в лингвистическую переменную, принимающую нечеткие значения «правильная», «неправильная», «смешанная».

Роли, описывающие пространственные отношения между понятиями:

Начало цепочки – тот ее конец, который лежит левее (выше, если хорда вертикальна) другого.

Right(x : Node, y : Chain) – цепочка y примыкает к узлу x правым концом.

RightUp(x : Node, y : Chain) – цепочка y примыкает к узлу x правым верхним концом.

Up(x : Node, y : Chain) – цепочка y примыкает к узлу x верхним концом.

LeftUp(x : Node, y : Chain) – цепочка y примыкает к узлу x левым верхним концом.

Left(x : Node, y : Chain) – цепочка y примыкает к узлу x левым концом.

$LeftDown(x: Node, y: Chain)$ – цепочка y примыкает к узлу x левым нижним концом.

$Down(x: Node, y: Chain)$ – цепочка y примыкает к узлу x нижним концом.

$RightDown(x: Node, y: Chain)$ – цепочка y примыкает к узлу x правым нижним концом.

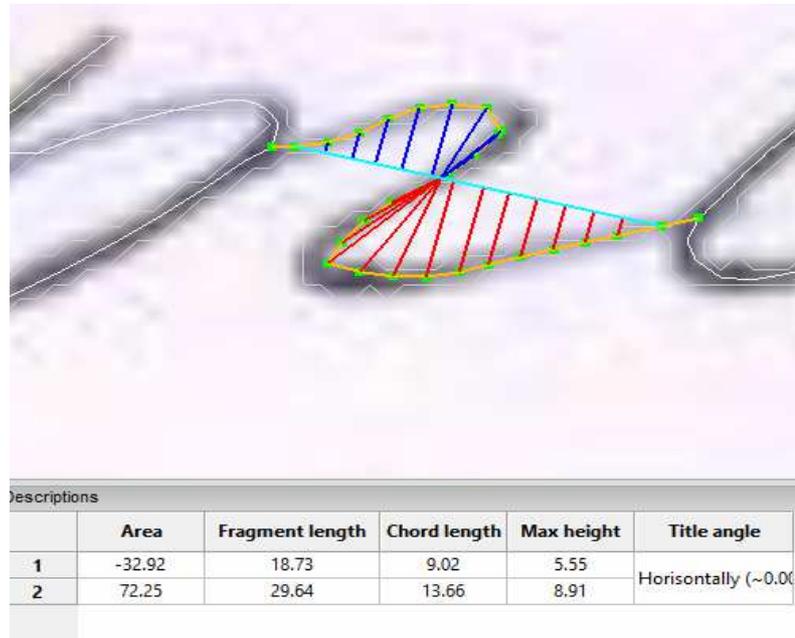


Рисунок 4.6 – Вычисление атрибутов цепочки скелетона

На рисунке 4.6 показан принцип работы геометрических функций, вычисляющих атрибуты формы. Концы цепочки между особыми точками скелетона (разветвления, концы ветвей, острые углы) соединяются хордой. Подсчитывается ее длина, угол наклона, а также площадь и высота для каждого сегмента фигуры, образованной хордой и цепочкой скелетона.

Поиск объектов на представленном в виде многослойного атрибутивного графа ($MLAG$ – Multi Layer Attributive Graph) векторизованном изображении заключается в упорядоченной проверке истинности данных высказываний с помощью функций интерпретации концептов и ролей (отношений). Затем, по логически описанной структуре объектов предметной области осуществляется автоматический отбор экземпляров из реальных данных путём логического вывода с помощью специализированного рассуждателя (reasoner).

Дескрипционные логики (ДЛ), такие как ALC , по существу являются фрагментом логики предикатов первого порядка (FOL), ориентированным для

работы с иерархическими структурами данных. ДЛ работают с понятиями и отношениями между ними (ролями), но в языке ДЛ не используются переменные. Это сближает их с обычной пропозициональной логикой и существенно упрощает понимание формул. Но самое важное – хотя семантика основных конструкций языка ДЛ и определяется через логику первого порядка, отношение *логического следования* в большинстве из них остается разрешимым, как и в пропозициональной логике.

Следующий пример поясняет взаимосвязь ДЛ и FOL.

Пример перевода некоторых описаний концептов и ролей с языка ДЛ на язык FOL:

1) если A – атомарный концепт (унарный предикатный символ), R – атомарная роль (бинарный предикатный символ), то их переводы $[A]$, $[R]$ – это формулы FOL $A(x)$, $R(x,y)$ соответственно;

2) если $[A](x)$, $[B](x)$ – переводы описаний концептов A и B , $[R](x,y)$ – перевод описания роли R , то переводы описаний (составных) концептов $\neg A$, $A \sqcup B$, $\exists R.A$ – это формулы FOL $\neg[A]$, $[A] \vee [B]$, $\exists y([R](x, y) \wedge [A](y))$ соответственно;

3) если $[P](x, y)$, $[Q](x, y)$ – переводы описаний ролей P и Q , то переводы описаний (составных) ролей $\neg P$, $P \sqcup Q$, $P \circ Q$, P^- – это формулы FOL $\neg[P]$, $[P] \vee [Q]$, $\exists z([P](x, z) \wedge [Q](z, y))$, $[P](y, x)$ соответственно.

Таким образом, расширение логики языка $ALC(GI)$ на область данных в виде атрибутивных графов позволило сократить область интерпретации для задач с графовой структурой данных. Проверяются только цепочки, соединяющие пары вершин. В то же время, за счет более развитой модели изображения в виде предложенного многослойного атрибутивного графа расширяются возможности структурного анализа изображений с помощью дескрипционных логик.

Предложенная дескрипционная логика рассуждений $ALC(GI)$ содержит два вида атомарных концепта:

1) осуществляющие поиск универсальных понятий и отношений на

графовых структурах;

2) порождаемые спецификой предметной области и обеспечивающие логические рассуждения о семантике и свойствах объектов, которые представлены графами изображений.

Чтобы создать онтологию для пространственно распределенных данных какой-либо предметной области, нужно дополнить множество понятий на графах понятиями, определяющими предметные свойства объектов. И наоборот, если имеется предметная онтология и требуется развить ее для анализа пространственных отношений между объектами и их структурными элементами, нужно дополнить множество атомарных понятий этой онтологии множеством понятий на графах.

Предполагается, что задачи, в настоящее время решаемые на графах, можно использовать в качестве атомарных концептов и ролей при моделировании рассуждений. Не нужно будет постоянно расширять наборы правил и ограничений и утяжелять интерпретацию ДЛ описаний. Существующие решения в теории графов и ее приложениях, а также в графовых базах данных можно применять в более широком перечне предметных областей, например, при анализе аэрокосмических снимков (техногенные ситуации, природные от арктических до лесных объекты) и других областях.

4.4.3. Алгоритм логического вывода в ДЛ *ALC(GI)*

В дескрипционной логике рассуждатель (reasoner) проверяет истинность аксиом и выполнимость концептов. Алгоритм рассуждателя отыскивает в разделе данных АВох базы знаний объекты, удовлетворяющие заданным понятиям, т.е. осуществляет проверку выполнимости определений понятий: концепт выполним, если он не пуст.

Эти два вида состояний являются листьями дерева поиска подходящих объектов. Узлы дерева отображают промежуточные состояния – результаты

($AVox A'$), полученные после применения одного логического правила. Если получено непротиворечивое множество фактов, то ветвь продолжается, иначе – возврат к предыдущему узлу дерева состояний.

Соответственно, на каждом шаге получается либо одно, либо два новых множества фактов $AVox A'$.

Порядок применения правил произвольный, но с помощью *стратегий* можно задавать их приоритетность с целью ускорения поиска. Так же можно задавать *ограничения*, прерывающие заведомо неперспективные пути.

В отличие от ALC , логика $ALC(GI)$ ищет «связанные» факты, которые выражают такие отношения, как: причинно-следственные, темпоральные, пространственные, составляющие прецеденты (ситуации), представляемые графами. Чтобы в атрибутивном графе найти заданный связный подграф, отображающий некоторое составное понятие, процесс ДЛ-поиска включает дополнительно этапы перебора фактов из $AVox A'$ для поиска в исходном $AVox$ связанных с ними фактов. Результаты этого (вторичного) поиска записываются в $AVox A''$.

В синтаксис $ALC(GI)$ добавлено правило:

Если $k = 1, 2, \dots, K$ целое положительное число и C – концепт, то $k:C$ – концепт такой, что $k:C \subseteq C$.

Если префиксы $k \neq l$, то $k:C \cap l:C = \emptyset$.

Теперь, когда требуется один и тот же концепт C ввести в определение концепта D несколько раз и в различных сочетаниях с ролями, то каждое вхождение концепта C , не пересекающееся по экземплярам с предыдущим вхождением, будем сопровождать новым префиксом.

Соответственно, если в обозначениях подконцептов концепта C префиксы совпадают, значит должен быть найден один и тот же экземпляр концепта C .

Отметим, что префиксы имеют локальную область действия – в пределах одного составного понятия.

Список правил логики $ALC(GI)$ приведен в таблице 4.1. В отличие от алгоритма проверки выполнимости понятий, заданных в ALC , здесь добавлено

правило $\exists R.C'$ установления связанных фактов (функция $FPrefix$ сравнения префиксов концептов), которое исключает «лишние» факты из $ABox A$. Так, в примере 4.2 (см. ниже) ДЛ-описания буквы “ m ” вначале отыскиваются все заданные цепочки и узлы в графе изображения вместе с их атрибутами, а затем для них ищутся факты, заданные связанными отношениями. Эти факты записываются в $ABox A'$.

В процессе поиска (доказательства выполнимости) некоторого (составного) концепта формируется $ABox A$ и строится размеченный граф $GL = (G, L)$. Изначально он содержит одну (корневую) вершину x , к которой приписан искомый концепт. Далее в ходе проверки правил при $x: C \sqcap D$ правило создает $ABox A'$, где добавлены $\{x: C, x: D\}$, т.е. $ABox A \cup \{x: C, x: D\}$. При $x: \exists R.C$ создается новая точка y , в которой будет $ABox A' := ABox A \cup \{xRy, y: C\}$. В графе GL появится ребро, связывающее точку x с точкой y . За счет таких ребер в графе образуются цепочки, связывающие некоторый концепт со всеми его атрибутами. Элементарные факты вида xRy или $y: A$ считаются элементами области интерпретации атомарных концептов и ролей.

Правила установления «связанных» фактов $\exists R.C'$:

(1) Если найдено более одного отношения, в которых связанный концепт C' имеет один и тот же префикс, то в множествах их элементарных фактов оставить только их общие экземпляры: $C_1' := C_1' \cap C_2'$ и $C_2' := C_1' \cap C_2'$. Правило срабатывает, когда один и тот же элемент встречается в различных отношениях с другими элементами.

(2) Если найдены отношения, где концепт C' имеет разные префиксы, то для этих отношений нужно брать разность множеств, т.е. те элементы, которые принадлежат одному множеству, но не принадлежат другому: $C_1' := C_1' \setminus C_2'$ и $C_2' := C_2' \setminus C_1'$. Правило срабатывает, когда в определении понятия упоминаются «такие же, но другие элементы».

Результатом доказательства выполнимости составного концепта будет полный и непротиворечивый $ABoxA$, содержащий его под- и подподконцепты, заканчивая атомарными, а также граф GL , с помощью которого множество индивидов, удовлетворяющих искомому концепту, извлекается из $ABox$ – раздела данных базы знаний.

Таблица 4.1

Список правил логики $ALC(GI)$

Правило	Условия применения	Действие
\perp	IF (x – активная точка) AND ($x:(C \perp L \in ABoxA)$ AND ($x:((C \notin ABoxA) \text{ OR } D \notin ABoxA))$)	THEN $ABoxA' := ABoxA \cup \{x:C, x:D\}$
\sqcup	IF (x – активная точка) AND ($x:(C \sqcup L \in ABoxA)$ AND ($x:((C \notin ABoxA) \text{ OR } (D \notin ABoxA))$)	THEN $ABoxA' := ABoxA \cup \{x:C\}$ $ABoxA'' := ABoxA \cup \{x:D\}$
\exists	IF (x – активная точка) AND ($x:\exists R.C \in ABoxA$) AND ($\neg \exists y, (xRy \in ABoxA) \wedge (y:C \in ABoxA)$)	THEN (Создать точку(y); $ABoxA' := ABoxA \cup \{xRy, y:C\}$)
\forall	IF (x – активная точка) AND ($x:\forall R.C \in ABoxA$) AND ($\exists y, (xRy \in ABoxA) \wedge (y:C \notin ABoxA)$)	THEN (Создать точку(y); $ABoxA' := ABoxA \cup \{y:C\}$)
$\sqsubseteq E$	IF (x – активная точка) AND ($x:E \notin ABoxA$)	THEN $ABoxA' := ABoxA \cup \{x:E\}$
$\exists R.C'$	IF (x – активная точка) AND ($x:\exists R.C_i' \in ABoxA$) AND ($x:\exists R.C_j' \in ABoxA$) AND ($i \neq j$) AND ($FPrefix(C_i', C_j') = True$)	THEN ($x:C_i' := \{C_i' \setminus C_j'\}$, $x:C_j' := \{C_i' \cap C_j'\}$)
	IF (x – активная точка) AND ($x:\exists R.C_i' \in ABoxA$) AND ($x:\exists R.C_j' \in ABoxA$) AND ($i \neq j$) AND ($FPrefix(C_i', C_j') = False$)	THEN ($x:C_i' := \{C_i' \setminus C_j'\}$, $x:C_j' := \{C_j' \setminus C_i'\}$)

Алгоритм.

1. Создать начальную точку x дерева поиска и записать в протокол состояний $AVoxA := \{x: C_0\}$, где C_0 – проверяемый концепт, а также факты – все подконцепты, содержащиеся в C_0 .
2. Из текущих условий (состояния $AVoxA$) выводятся новые факты. Для этого на каждом шаге к текущему состоянию базы найденных фактов применяется какое-либо правило ДЛ (таблица 4.1) и создается один или два новых состояния $AVox A'$ и (для правила « L ») $AVox A''$, которые далее будут проверяться независимо.
3. Если в новом состоянии базы фактов появились противоречивые факты, т.е. для некоторого индивида x атомарного концепта A имеется $x: A$ и $x: \neg A$, то переход к п. 6 (данное направление поиска дало отрицательный результат).
4. Если какое-либо правило выполнилось, то возврат к п.2, иначе – к следующему пункту.
5. Установление связанных фактов (правила « $\exists R.C'$ »).
6. Останов.

Таким образом, в отличие от базовой дескрипционной логики ALC , рассуждатель $ALC(GI)$ ищет «связанные» факты, выражающие такие отношения, как: причинно-следственные, темпоральные, пространственные, которые составляют прецеденты, представляемые графами.

Возможными интерпретациями атомарных концептов здесь являются не произвольные множества, а множество вершин и множество дуг заданного графа. При проверке выполнимости формул, использующих концепты *Подграф* и *Связный Подграф*, проверяется наличие искомым подграфов в предъявленном графе, ищутся цепочки или циклы нужного вида. На изображениях логика $ALC(GI)$ сокращает область интерпретации понятий за счет перехода от пиксельного представления данных к атрибутивным графам. Например, вместо

перебора множества пикселей изображения или множества отрезков будут проверяться только цепочки, соединяющие пары вершин.

В сложных случаях, например, при поиске чертежей по заданному образцу или других объектов со сложной структурой, вступает в силу метод сопоставления атрибутивных графов с помощью *лучевых графов* (см. п. 4.2), автоматически порождающий свои поддеревья поиска.

Немаловажным достоинством рассуждателя *ALC(GI)* является также возможность автоматически генерировать ДЛ-определение для выделенного объекта, основываясь на его атрибутивном графе, что существенно упрощает работу пользователей по созданию терминологического раздела базы знаний.

4.4.4. Примеры предметных терминологий графической информации

Пример 4.1. **Дескрипционная логика *ALC(GI)* и онтология деталей-сборочных единиц.** Назначение: Организация базы знаний и данных онтологического типа для системы поддержки принятия конструкторско-технологических решений (**КТР**).

Пусть имеются атомарные концепты: *Point* – точка; *Element* – отрезок.

и атомарные роли:

$R1 \equiv \text{иметьСоседаПоНачалу}(a, b);$

$R2 \equiv \text{иметьСоседаПоКонцу}(a, b).$

$R3(a, b) \equiv \text{бытьПаралельными}(a, b);$

$R4(a, b) \equiv \text{бытьПерпендикулярными}(a, b);$

$R5(a, b) \equiv \text{равныПоДлине}(a, b);$

Пусть заданы также производные понятия:

ПоследовательностьСоседнихЭлементов \equiv *Цепочка* \equiv *Sequence*;

Контур \equiv $Sequence(e_1, e_n) \sqcap \exists R2(e_n, e_1)$ – замкнутая последовательность из n

элементов;

Четырёхугольник $\equiv Sequence(a,b,c,d) \sqcap \exists R2(d, a)$. Четырёхугольник – это замкнутая последовательность из четырех прямолинейных отрезков;

Прямоугольник $\equiv Четырёхугольник \sqcap \exists R3(a, c) \sqcap \exists R3(b, d) \sqcap \exists R4(a, b) \sqcap \exists R4(b, c) \sqcap \exists R4(c, d)$. Прямоугольник – это четырёхугольник, у которого противоположные стороны параллельны, а смежные стороны перпендикулярны;

Квадрат $\equiv Четырёхугольник \sqcap \exists R5(a,b,c,d) \sqcap \exists R4(a, b) \sqcap \exists R4(b, c) \sqcap \exists R4(c, d)$. Квадрат – это четырёхугольник, у которого все стороны равны и смежные стороны перпендикулярны;

Параллелограмм $\equiv Четырёхугольник \sqcap \exists R3(a,c) \sqcap \exists R3(b, d)$. Параллелограмм – это четырёхугольник, у которого противоположные стороны параллельны.

Функциональные отношения (атрибуты), характеризующие форму контуров:

Угол(x, y : вершина, a : НечеткоеЗначение) – функция, возвращающая нечёткое значение {Острый, Прямой, Тупой} на основе разности между углами векторов соседних элементов.

Форма(x : Сегмент, y : НечёткоеЗначение) - вычисляет нечёткое значение {Треугольник, Четырёхугольник, Прямоугольник, Пятиугольник} на основе количества вершин и отношений между элементами границы.

Хорда (x : Цепочка, y : НечёткоеЗначение) – вычисляет нечёткое значение {Короткая, Средняя, Длинная} длины хорды, соединяющей концы цепочки.

Ранее были введены понятия: *Разветвление*, *XВершины*, *YВершины*,

Обычно, системы геометрического моделирования предоставляют пользователю множество способов формообразования при создании трехмерных геометрических моделей. Ниже рассмотрим наиболее популярные.

Операции формообразования. Для описания трехмерных объектов, получаемых кинематическим способом из плоских, введем операции формообразования: Вращение, Лифтинг, Трансформация, которые порождают 3D примитивы, часто называемые кинематическими объектами, т.к. образуются в процессе перемещения кривой в пространстве.

Вращение(Цепочка) – унарная операция, порождающая 3D объект типа «вал» путем вращения цепочки из элементов типа дуга окружности, отрезок прямой вокруг оси, проходящей через точки начала и конца цепочки:

$Вал \equiv 3DPrimitive \sqcap \exists = 1. Вращение(Цепочка)$

$\sqcap \rightarrow \exists$ Лифтинг. $T \sqcap \rightarrow \exists$ Трансформация. T

Цепочка может включать в качестве подцепочек ранее определенные цепочки, например, для образования кольцевых канавок.

Лифтинг(Контур) – унарная операция, образующая 3D объект типа «призма» (называемая также призмоцилиндрическим телом), при поступательном перемещении контура вдоль вертикальной оси на заданную высоту:

$Призма \equiv 3DPrimitive \sqcap \exists = 1. Лифтинг(Контур)$

$\sqcap \rightarrow \exists$ Вращение. $T \sqcap \rightarrow \exists$ Трансформация. T

Трансформация(a, b, c: Контур) – формообразующая операция, определяющаяся бинарным или тернарным (в общем случае, n –арным) отношением. В процессе перемещения, в частности, вдоль оси Z первый контур масштабируется другими контурами.

Отношения пространственного положения. Далее введем бинарные отношения, описывающие некоторые возможные способы задания взаимного расположения трехмерных геометрических примитивов.

Поскольку составные 3D объекты отображаются в пространственные атрибутивные графы, здесь также приходится применять префиксы (п. 4.4.3) для обозначения тех подобъектов, которые встречаются в описании сборки более одного раза.

Примечание. Соединение между одними и теми же двумя геометрическими примитивами может быть задано одновременно несколькими способами, чтобы однозначно установить их пространственное положение относительно друг друга.

СовпадениеПоверхностей($A, B: 3DPrimitive, a, b: Element$) – отношение указывает, что трехмерные примитивы A, B соединяются между собой так, что поверхность, образованная элементом a первого объекта, и поверхность, образованная элементом b второго, совпадают. При этом границы поверхностей могут отличаться.

СовпадениеРебер($A, B: 3DPrimitive, a, b: Вершина$) – отношение указывает, что трехмерные примитивы A, B соединяются между собой так, что ребро, образованное вершиной a первого объекта, и ребро, образованное вершиной b второго, совпадают.

Этим способом, например, формируются с помощью операции **вычитания** (\setminus) фаски и скругления или галтели – с помощью операции **сложения** (\cup).

СовпадениеОсей($A, B: Вал$) – отношение указывает, что трехмерные примитивы A, B , образованные операцией вращения располагаются на одной оси. При этом объект B может быть как прибавляемым, так и вычитаемым.

Расстояние($A, B: 3DPrimitive, a, b: Вершина, y: Число$) – расстояние между точками a, b примитивов A, B соответственно.

Положение 3D объекта в пространстве задается составным атрибутом, содержащим стандартные параметры вектора: $Vector3DObject(x, y, z: Число, a, b, c, d, e, f: Угол)$.

Пример. Имеется эскиз детали «Упор» (рисунок 4.7).

Элементы 3, 9, 10, 11, 12, 13 (рис. 4.10 а) за счет операции «лифтинг» образуют трехмерный геометрический примитив типа «призма»:

$Ребро \equiv 3DPrimitive \sqcap \exists Лифтинг(Треугольник) \sqcap \exists имеетВысота.=h1$

Так же контур, составленный из элементов 1 – 8, образует 3D примитив:

$Уголок \equiv 3DPrimitive \sqcap \exists Лифтинг(Контур) \sqcap \exists имеетВысота.=h2$

Цепочка элементов 10, 14, 6 путем вращения образует цилиндр, который при вычитании даст нам отверстие:

$Вал \equiv 3DPrimitive \sqcap \exists =Вращение(Цепочка)$

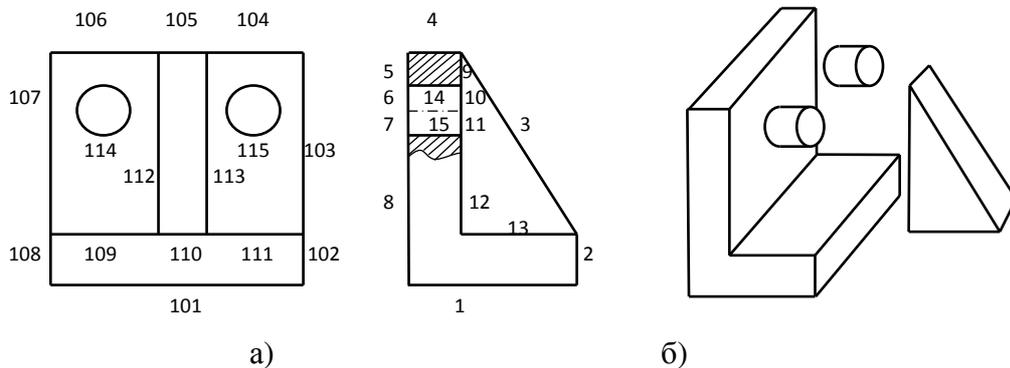


Рисунок 4.7 – Описание 3D объекта: а) эскиз детали "Упор"; б) 3D примитивы

Здесь $h1, h2$ – переменные, задающие толщину ребра и длину уголка, связанные с размерными параметрами соответствующих элементов чертежа.

Тогда деталь «упор» может быть получена следующим образом (в общем случае существует несколько способов):

$Упор \equiv 3DGM \sqcap \exists A: Уголок \sqcup \exists B: Ребро \setminus \exists C: Вал \setminus \exists D: Вал$

$\sqcap \exists СовпадениеПоверхностей(A, B, 12, 12)$

$\sqcap \exists СовпадениеПоверхностей(A, B, 13, 13)$

$\sqcap \exists Расстояние(A, B, 13, 13, h3)$

$\neg \exists$ Совпадение Поверхностей(A, C, 6, 6)

$\neg \exists$ Расстояние(A, C, 107, 114, h4) $\neg \exists$ Расстояние(A, C, 109, 114, h5)

$\neg \exists$ Совпадение Поверхностей(A, D, 6, 6)

$\neg \exists$ Расстояние(A, D, 103, 115, h4) $\neg \exists$ Расстояние(A, D, 111, 115, h5)

Очевидно, создание ДЛ описаний трехмерных геометрических моделей – работа достаточно трудоемкая. Но можно создать примеры, на которых машина обучится автоматически строить описания предъявленных моделей, т.к. ДЛ-описание однозначно определяет алгоритм преобразования «чертеж – 3D ГМ».

Преобразование «3D – 2D – логическое описание 3D» может быть осуществлено двумя способами:

- 1) генерация проекционных видов чертежа и синтез описания;
- 2) поиск 3D примитивов в 3D геометрической модели объекта и описание их связей.

Таким образом, формализованное машиночитаемое описание деталей сборочных единиц (ДСЕ) с помощью дескрипционной логики служит для однозначного определения семантики геометрических моделей с целью поддержки принятия конструкторских и технологических решений.

Пример 4.2. Фрагмент терминологии, описывающей изображения рукописных букв на чертеже. В качестве примера использования логики $ALC(GI)$ для описания и анализа изображений рассмотрим задачу распознавания рукописного текста. Основной проблемой в этой задаче является получение общего и, в то же время, точного описания букв, легко проверяемого неким общим алгоритмом. Покажем, что язык $ALC(GI)$ достаточен (может использоваться) и удобен для этих целей.

Исходное изображение – отсканированная страница рукописного текста.

С помощью векторизатора (см. например, [157]) автоматически строится модель изображения – ABox домена изображения в виде атрибутивного графа. В

данном примере интересует только слой G^{SKEL} изображения – граф скелетона (см. выше).

Примеры терминологических аксиом, описывающих конструктивные элементы букв с использованием ранее определенных понятий на графах изображений:

Вертикальная палочка – цепочка изображения, имеющая малое отклонение от хорды, соединяющей ее концы, и вертикальное положение:

$$VerticalStick \equiv Chain \sqcap (\exists hasMaxAreaHeight. = Small) \sqcap (\exists hasTiltAngle. = Vertically)$$

Короткая вертикальная палочка – цепочка изображения, имеющая малое отклонение от хорды, соединяющей ее концы, и вертикальное положение:

$$ShortVerticalStick \equiv Chain \sqcap (\exists hasMaxAreaHeight. = Small) \sqcap (\exists hasTiltAngle. = Vertically) \sqcap (\exists hasChordLength. = Short)$$

Горизонтальная палочка – цепочка, имеющая малое отклонение от хорды и горизонтальное положение:

$$HorizontalStick \equiv Chain \sqcap (\exists hasMaxAreaHeight. = Small) \sqcap (\exists hasTiltAngle. = Horizontally)$$

Петля малая – цепочка, имеющая один сегмент (слева или справа от хорды), площадь которого малая:

$$SmallLoop \equiv Chain \sqcap (\exists Петля \sqcap (\exists hasSegmentNumber. = 1) \sqcap (\exists hasArea. = Small))$$

Левая дуга – цепочка, имеющая хорду средней длины и один сегмент слева (по ходу) от хорды, площадь которого средняя:

$$LeftArc \equiv Chain \sqcap (\exists hasChordLength. = Medium) \sqcap (\exists hasSegmentNumber. = 1) \sqcap (\exists hasLeftSegmentArea. = Medium)$$

Извилина – цепочка, имеющая два сегмента (один слева, другой справа от хорды) и площадь каждого сегмента средняя:

$$Gyrus \equiv Chain \sqcap (\exists hasSegmentNumber. = 2) \sqcap (\exists hasEachSegmentArea. = Medium)$$

Терминологические аксиомы, описывающие буквы.

Для описания букв, составленных из элементов, подобных описанным выше, часто требуется один и тот же концепт C элемента вводить в определение концепта D несколько раз и в различных сочетаниях с ролями. Тогда каждое вхождение концепта C , не пересекающееся по экземплярам с предыдущим вхождением, должно сопровождаться новым префиксом. Соответственно, если в обозначениях подконцептов концепта C префиксы совпадают, значит должен быть найден один и тот же экземпляр концепта C .

В общем случае концепт $k:C$ содержит более одного экземпляра. Например, в тексте одинаковые буквы могут встречаться многократно. Тогда концепт $k:C$ будет выделять, например, все вторые палочки в буквах “m”.

Напомним, что префиксы имеют локальную область действия – в пределах одного составного понятия.

Буква «C» – конструкция, состоящая из правой дуги и петли, которые входят в узел: дуга – верхней частью, а петля – левой:

$$\text{LetterC} \equiv \text{ConnectedSubgraph} \sqcap \exists a: \text{RightArc} \sqcap \exists b: \text{SmallLoop}$$

$$\sqcap \exists (A: \text{Node} \sqcap \exists \text{hasTop}(A,a) \sqcap \exists \text{hasLeft}(A,b))$$

Буква строчная «T» – конструкция (рисунок 4.8), состоящая из короткой палочки, имеющей идентификатор 1, двух вертикальных палочек 2, 4, одной левой дуги 3 и одной извилины 5, образующих два узла. В первый входит палочка 1 нижним концом, палочка 2 верхним концом и левая дуга 3 – левым концом. Во втором узле правый конец той же дуги (3) соединен с верхним концом другой вертикальной палочки (4) и с левым концом извилины 5:

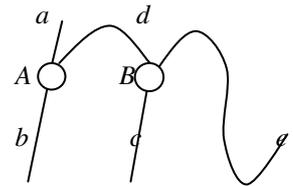


Рисунок 4.8 - Рукописная буква

$$\text{LetterSmallT} \equiv \text{ConnectedSubGraph}$$

$$\sqcap (\text{Node} \sqcap \exists \text{Bottom}(1: \text{ShortVertStick}) \sqcap \exists \text{Top}(2: \text{VerticalStick}) \sqcap \exists \text{Left}(3: \text{LeftArc}))$$

$$\sqcap (Node \sqcap \exists Right(3:LeftArc) \sqcap \exists Top(4:VerticalStick) \sqcap \exists Left(5:Gyrus))$$

Рассмотрим *интерпретацию* приведенных терминологических аксиом на графах изображений. Концепт в ДЛ аналогичен высказыванию в логике предикатов, чаще называемой логикой первого порядка (ЛП1). Интерпретация (т.е. семантика) некоторого концепта (понятия) логики *ALC* – это множество элементов, найденных в области данных с помощью интерпретирующей функции, которые (элементы) удовлетворяют данному концепту.

Интерпретирующая функция записывается с помощью синтаксиса ЛП1. Для удобства понимания будем искать интерпретацию всех производных концептов и всех ролей, а потом их пересечение или объединение как задано в определении понятия.

Интерпретация концепта $\exists hasMaxAreaHeight.= Small$ на области данных графа:

$$(\exists hasMaxAreaHeight.= Small)^G = \{a \in \Delta^G \mid \text{существует } aA \in \Delta^G \text{ такой,}$$

$$\text{что } \langle a, aA \rangle \in R^G \wedge MaxAreaHeight(a) = Small\}$$

– это множество, которое образуется с помощью двух правил: элемент a принадлежит области данных графа и существует элемент (атрибут) aA , принадлежащий области данных графа, такой, что функциональное отношение *MaxAreaHeight* между ними имеет значение *Small*.

Аналогично интерпретации:

$$(\exists hasTiltAngle.= Vertically)^G =$$

$$\{a \in \Delta^G \mid \text{существует } aA \in \Delta^G \text{ такой, что } \langle a, aA \rangle \in R^G \wedge TiltAngle(a) =$$

$$Vertically\};$$

$$(\exists hasChordLength.= Short)^G =$$

$$\{a \in \Delta^G \mid \text{существует } aA \in \Delta^G \text{ такой, что } \langle a, aA \rangle \in R^G \wedge ChordLength(a) =$$

$$Short\};$$

$$(\exists hasSegmentNumber.= 1)^G =$$

$$\{a \in \Delta^G \mid \text{существует } aA \in \Delta^G \text{ такой, что } \langle a, aA \rangle \in R^G \wedge SegmentNumber(a) = 1\};$$

$$(\exists hasLeftSegmentArea. = Medium)^G =$$

$$\{a \in \Delta^G \mid \text{существует } aA \in \Delta^G \text{ такой, что } \langle a, aA \rangle \in R^G \wedge LeftSegmentArea(a) = Medium\};$$

$$(\exists hasEachSegmentArea. = Medium)^G =$$

$$\{a \in \Delta^G \mid \text{существует } aA \in \Delta^G \text{ такой, что } \langle a, aA \rangle \in R^G \wedge EachSegmentArea(a) = Medium\}.$$

Интерпретации концептов *VerticalStick*, *ShortVerticalStick*, *LeftArc*, *Gyrus*, составляющих концепт *LetterSmallT*, будут получены путем пересечения соответствующих множеств цепочек, которые выделяются на предыдущем этапе интерпретации. За счет этого выделим на графе изображения конструктивные элементы (КЭ) букв.

Следующий этап интерпретации концепта буквы удобнее рассматривать на множестве именованных (классифицированных) КЭ, обозначим его *CE*. Для этого нужно выделить в графе соединения между КЭ, соответствующие заданным в определении буквы, т.е. найти узлы, содержащие заданные соединения.

Например, интерпретация концепта $\exists hasConnectionType(a, b). = Down_Up$:

$$(\exists hasConnectionType(a, b). = Down_Up)^{CE} = \{a \in CE \mid \text{существует } b \in CE \text{ такой, что } \langle a, b \rangle \in R^G \wedge ConnectionType(a, b) = Down_Up \}.$$

Таким образом, алгоритм выявления букв в предлагаемом подходе – это стандартный алгоритм проверки общезначимости формул ДЛ путем индукции по построению концептов.

Функции интерпретации концептов и ролей, определенных на графах как таковых, т.е. абстрактных понятий теории графов, встроены в модуль логического вывода, реализующий логику *ALC(GI)*.

4.5. Контроль корректности вербальных представлений графической информации

После того как с помощью дескрипционной логики заданы термины и определения (терминологический раздел базы знаний), блок логического вывода осуществляет контроль корректности определений, проверку выполнимости, непересекаемости и собственно выборку удовлетворяющих экземпляров из раздела данных.

Каждая программа, которая «распознает» множества точек как, например, «отрезки» или «дуги» – по сути является некоторым формальным определением этих понятий. Но, чем сложнее образы (эталон), тем запутанней «логика рассуждений» программы, тем больше всяческих исключений и эвристик, тем сложнее находить пробелы и ошибки в этих автоматических «рассуждениях». А еще и надо их как то исправлять...

Именно поэтому, в современных подходах, используется многоуровневое построение интеллектуальных систем (ИС). При этом каждый из уровней, по сути - это уровень формализации понятий и рассуждений. Уже почти общепризнанной является необходимость наличия в базе знаний ИС одного или нескольких уровней логической классификации понятий (онтологии), которые позволяют [28] структурировать выводы ИС и управлять ими. Обычно онтология создается экспертом предметной области.

Предлагается методика контроля корректности определений графической информации, задаваемых экспертом предметной области, которая заключается в контроле выполнимости производного концепта, непересекаемости концептов, корректности терминологии в целом, оценке и регулировании релевантности результатов интерпретации. Здесь известные методы адаптированы к новой задаче контроля корректности описания графической информации.

1. Контроль выполнимости концепта С.

В логике *ALC* как и в любой другой ДЛ: концепт *C* выполним, если существует такая его интерпретация *X*, что $C^x \neq \emptyset$.

Причинами невыполнимости концепта в логике *ALC(GI)* могут быть:

- Недоопределенность – из-за отсутствия какого-то термина среди атомарных концептов или среди ранее определенных составных. Например: "Залив – водное пространство, вдающееся в сушу". Словарь русского языка. Составитель: С. И. Ожегов. 1953.

- В базе данных нет ни одного экземпляра, удовлетворяющего заданной интерпретации.

- Определение противоречит атомарным или ранее заданным производным концептам.

Пример: "перпендикуляр, опущенный из вершины треугольника, разбивает его на два прямоугольных треугольника, сумма площадей которых равна площади исходного треугольника" – противоречит аксиомам геометрии в случае тупоугольного треугольника.

2. Контроль непересекаемости концепта *C* с другими концептами. В логике *ALC* концепты *C* и *D* называются непересекающимися, если в любой интерпретации *X* имеем $C^x \cap D^x = \emptyset$.

Пример 4.3.

Пусть имеются атомарные концепты: *Point*; *Element*.

и атомарные роли:

$R1 \equiv \text{иметьСоседаПоНачалу}(a, b);$

$R2 \equiv \text{иметьСоседаПоКонцу}(a, b).$

$R3(a, b) \equiv \text{бытьПаралельными}(a, b);$

$R4(a, b) \equiv \text{бытьПерпендикулярными}(a, b);$

$R5(a, b) \equiv \text{равныПоДлине}(a, b);$

Пусть заданы также производные понятия:

$\text{ПоследовательностьСоседнихЭлементов} \equiv \text{Sequence};$

$\text{Четырехугольник} \equiv \text{Sequence}(a, b, c, d) \sqcap \exists R2(d, a)$ - Четырехугольник – это

замкнутая последовательность из четырех прямолинейных отрезков;

$\text{Прямоугольник} \equiv \text{Четырёхугольник} \cap \exists R3(a, c) \cap \exists R3(b, d) \cap \exists R4(a, b) \cap \exists R4(b, c) \cap \exists R4(c, d)$ - Прямоугольник – это четырехугольник, у которого противоположные стороны параллельны, а смежные стороны перпендикулярны;

Введем следующее понятие:

$\text{Квадрат} \equiv \text{Четырёхугольник} \cap \exists R5(a,b,c,d) \cap \exists R4(a, b) \cap \exists R4(b, c) \cap \exists R4(c, d)$ - Квадрат – это четырехугольник, у которого все стороны равны и смежные стороны перпендикулярны.

Утверждение 1. Концепты *Прямоугольник* и *Квадрат* пересекаются.

Требуется показать, что при заданных выше интерпретациях $\text{Прямоугольник}^x \cap \text{Квадрат}^x \neq \emptyset$. Доказательство этого утверждения легко осуществляется с помощью аксиом геометрии (в четырехугольнике со всеми равными сторонами, противоположные стороны параллельны).

Поскольку нет других фигур, которые являясь квадратами не были бы прямоугольниками, следовательно справедлива аксиома:

$\text{Квадрат} \sqsubseteq \text{Прямоугольник}$.

С точки зрения эксперта, доказательство кажется тривиальным, но для ИС (например: оптимальный раскрой, укладка и другие решатели геометрических задач) алгоритмы конструктивного доказательства должны быть заложены в блок логического вывода.

3. Контроль корректности терминологии для описания графической информации [114]:

Пусть T – система определений – терминология базы знаний.

Утверждение 2. Терминология T корректна, если все ее концепты выполнимы и никакой концепт A не содержит производные концепты $B \equiv A$ или $D \supset A$ (D включает A).

Пример 4.4. В терминологию, описанную в примере 4.3, введем понятия:

$\text{Параллелограмм} \equiv \text{Четырёхугольник} \cap \exists R3(a,c) \cap \exists R3(b, d)$ -

параллелограмм – это четырехугольник, у которого противоположные стороны параллельны;

Ромб \equiv *Параллелограмм* $\sqcap \exists R5(a, b, c, d)$ - Ромб – это параллелограмм, у которого все стороны равны.

Из этих аксиом следует, что *Параллелограмм* является потомком *Четырёхугольника*, а *Ромб* – потомком *Параллелограмма*. Соответственно, в графе $\Gamma(T)$ из этих вершин будут построены исходящие дуги.

Введем следующее понятие:

ЧетырёхугольникСРавнымиСторонами \equiv *Четырёхугольник* $\sqcap \exists$ *равныПоДлине*(a, b, c, d).

В логике *ALC* доказано, что терминология T корректна, если орграф зависимостей концептов $\Gamma(T)$ не содержит циклов и петель.

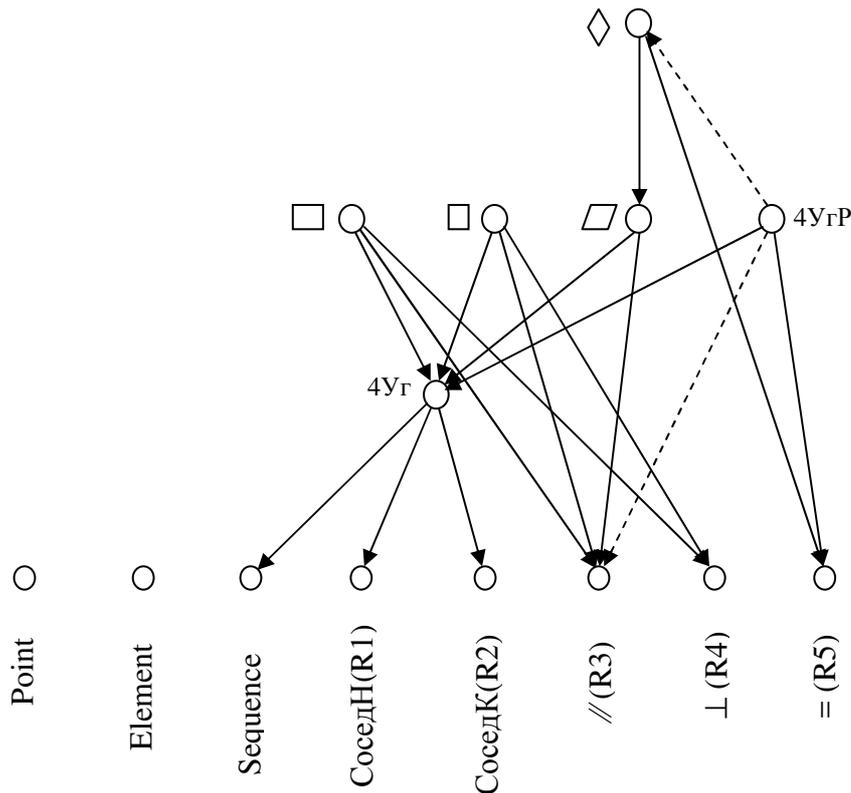


Рисунок 4.9 – Граф $\Gamma(T)$ терминологии к примерам 4.5, 4.6

Из рисунка 4.9 видим, что понятия *ЧетырёхугольникСРавнымиСторонами* и *Ромб* включают отношение $R5$ равенства сторон. Кроме того, *ЧетырёхугольникСРавнымиСторонами* и *Ромб* (последний через *Параллелограмм*)

выходят на отношение R3 параллельности сторон, причем *ЧетырехугольникСРавнымиСторонами* – за счет аксиом геометрии.

Следовательно, *интерпретация ЧетырехугольникСРавнымиСторонами*^x = *Ромб*^x, т.е. *ЧетырехугольникСРавнымиСторонами* \equiv *Ромб* (понятия эквивалентны) и мы должны в графе $\Gamma(T)$ соединить их дугой, в результате чего в графе зависимости понятий появится цикл, означающий, что четырехугольник с равными сторонами – это ромб.

Таким образом, понятия "ромб" и "четырёхугольник с равными сторонами" являются синонимами, так как в дереве определений образуется цикл, т.е. машина сможет сделать вывод о том, что появилась некорректность.

Рассмотрим ещё один пример системы определений на графической информации. В терминологии базы знаний для распознавания рукописных текстов (пример 4.2) экспертом определены понятия конструктивных элементов и букв, составляемых из них.

Базой данных *АВох* для рассуждателя (reasoner) служит растровое изображение и атрибутивный граф скелетона текста. Это совокупность двух множеств:

- множество вершин (узлов) $\{V\}$, $V_i = \langle x_i, y_i \rangle$;
- множество ребер – цепочек, состоящих из прямолинейных и дуговых отрезков (1), которые соединяют между собой вершины.

Функциональные свойства цепочек и пространственные отношения между цепочками вычисляются на *АВох* с помощью предикатов, основывающихся на геометрических интерпретирующих функциях (FR1 – FR10), которые анализируют форму и относительное положение элементов конструкций.

При распознавании букв задача рассуждателя заключается в том, чтобы найти в графе изображения такие совокупности цепочек, которые обладают заданными функциональными свойствами и находятся между собой в отношениях, указанных определением буквы, например, *Буква «С»*.

В примере 4.2 терминология *ТВох* базы знаний содержит определения типа

аксиом эквивалентности, поэтому для корректного взаимодействия с базой данных АВох и эффективной реализации запросов, необходимо показать *корректность* (ацикличность) терминологии.

Построим ориентированный граф $\Gamma(T)$ зависимостей концептов. Вершинами этого графа будут концепты терминологии T и из вершины A в вершину B идет дуга, если есть концепт C , эквивалентный концепту A , или концепт B входит в концепт C . При этом концепты $A \equiv C$, если в интерпретации $A^x = C^x$. В графе $\Gamma(T)$ базовые концепты не имеют исходящих дуг. Производные концепты имеют одну исходящую дугу.

Ациклический орграф $\Gamma(T)$ зависимостей понятий, определяющих рукописную букву, показан на рисунке 4.10. Как видим, граф терминологии не содержит циклов (получили дерево), т.е. терминология корректна.

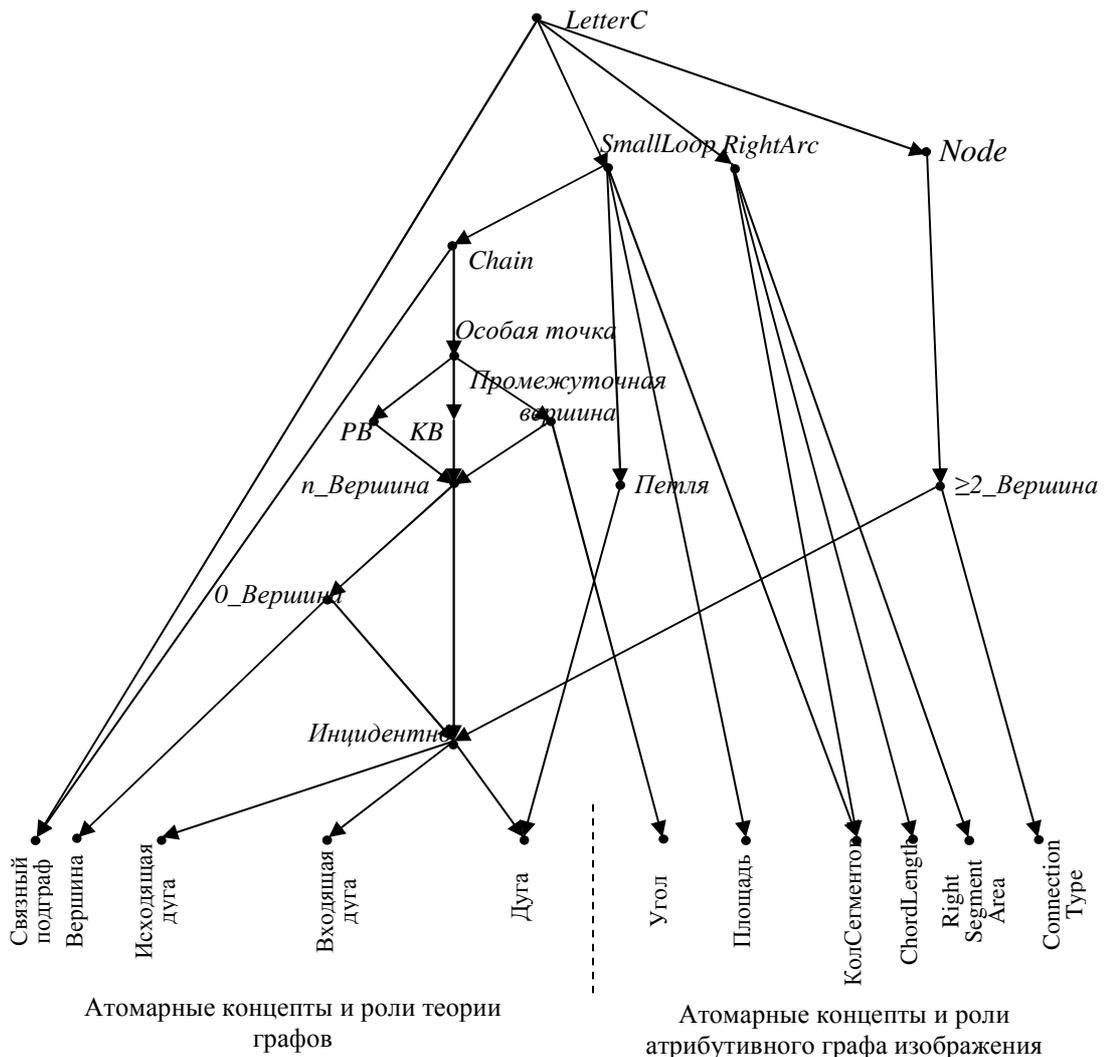


Рисунок 4.10 - Граф системы понятий "Рукописная буква С"

4. Оценка релевантности компьютерной интерпретации определений, заданных на графической информации.

Для оценки релевантности информационного поиска в базах документов [81] обычно используют показатели точности и полноты выдачи.

Точность определяется как отношение числа релевантных документов, найденных информационно-поисковой системой, к общему числу найденных документов:

$$\text{Precision} = |(D_{\text{rel}} \cap D_{\text{retr}})| / |D_{\text{retr}}|$$

Полнота - это отношение числа найденных релевантных документов, к общему числу релевантных документов в базе:

$$\text{Recall} = |(D_{\text{rel}} \cap D_{\text{retr}})| / |D_{\text{rel}}|,$$

где D_{rel} – множество релевантных документов;

D_{retr} – множество документов, найденных системой.

С целью уменьшения влияния субъективизма релевантность поиска предлагается оценивать следующим образом. *Релевантным* будем считать объект такой, что оценка сходства C атрибутивных графов, состоящих из узлов (элементов объекта) и связей между ними, превышает заданную пороговую величину p :

$$C = V_s / V_d \geq p, \text{ где}$$

V_d - количество вершин в заданном графе;

V_s - количество вершин, соответствующих описанию-запросу.

Формула оценки сходства позволяет более строго (формализовано) определить, что отнести к релевантным (похожим) объектам.

Тогда **полнота** выдачи характеризуется отношением количества релевантных объектов, найденных и системой, и экспертами, к общему количеству объектов, найденных экспертами:

$$\Pi = |(ROS(C) \cap \bigcup_{i=1}^N OE_i)| / |\bigcup_{i=1}^N OE_i|, \quad (4.5)$$

где: $ROS(C)$ - множество релевантных объектов, найденных системой по оценке сходства C , $|ROS(C)|$ - количество элементов множества;

$|\bigcup_{i=1}^N OE_i|$ - количество элементов объединения множеств объектов,

найденных группой из N экспертов.

Точность выдачи определяется дополнением объединённого множества объектов, найденных экспертами, на множестве объектов, найденных системой, как:

$$T = \frac{|\bigcup_{i=1}^N OE_i \cap ROS(C)|}{|ROS(C)|}, \quad (4.6)$$

то есть чем меньше ошибочно найденных системой объектов, тем выше точность выдачи.

Приведенные здесь формулы показывают зависимость оценок точности и полноты выдачи от оценки сходства C , которая "регулирует" выдачу.

5. Регулирование релевантности производится путем корректировки диапазонов и форм функций фаззификации лингвистических переменных, описывающих признаки объектов на изображении (рисунок 4.11). Пользователю предоставляется

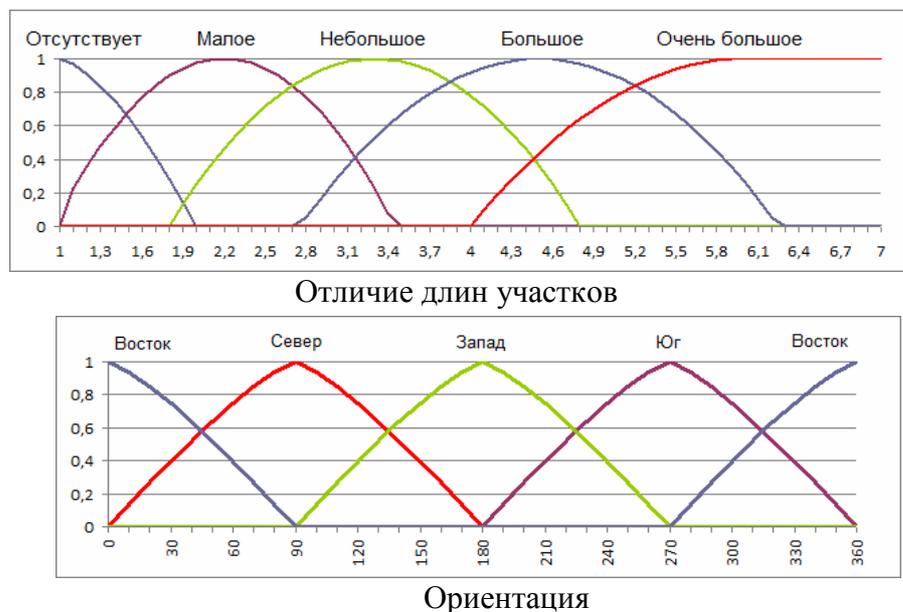


Рисунок 4.11 – Примеры функции фаззификации отношений формы возможность изменять ширину и углы наклона фигур, определяющих значения лингвистической переменной.

4.6. Классификация. Группирование деталей в задачах технической подготовки производства

Групповые технологии являются основой унификации технологических решений, повышая эффективность машиностроительного производства [65].

В работе [138] предлагается объединять принципы бережливого производства и групповой технологии изготовления. Определяется схожесть изготовления деталей исходя из используемого оборудования. Длительность изготовления полученных групп должна быть не больше такта изготовления изделия. Также в [153] описан метод, позволяющий сократить производственный цикл изготовления изделия за счет реорганизации производственных потоков с помощью группировки деталей. В статье [162] предложена концептуальная модель, которая позволяет связать принципы бережливого производства и групповой технологии.

Другой принцип группирования деталей – по близости календарно-плановых сроков исполнения заданий. В статье [86] рассмотрена комплектно-цикловая система оперативного календарного планирования, где в плане-графике каждого цеха указываются сроки запуска-выпуска комплектов деталей, сгруппированных по одинаковым или близким календарно-плановым нормативам, величинам опережений, технологическим маршрутам и периодичности запуска-выпуска.

Для внедрения групповых технологий важно классифицировать детали и выполнять их выборку в разрезе конструктивных и технологических признаков. Базисом такой выборки является объединение деталей по конструкторско-технологическим признакам. В работе [10] описываются способы группировки деталей.

При классификации деталей с целью их группировки предлагается учитывать ряд признаков, включая конструктивные признаки деталей согласно

классификатору ЕСКД, массогабаритные параметры и особенности технологии изготовления деталей.

Наиболее распространён метод группировки деталей с помощью построения комплексной детали, которая включает в себя наибольшее число конструкторско-технологическим признаков группы. При этом остальные детали из группы могут содержать только часть признаков.

Полученный классификатор деталей позволяет определить группы деталей, которые следует изготавливать совместно, тем самым уменьшая время переналадки оборудования, и соответственно это позволяет увеличить его загрузку. Групповая технология, в отличие от типовой, характеризуется по общности оборудования и оснастки, необходимых для выполнения определенной операции или полного цикла изготовления детали, а не общности технологического процесса.

Ниже предлагается подход к обеспечению группирования деталей на ранних стадиях проектирования новых изделий или модификации существующих путем логического синтеза классификаторов деталей и конструктивных элементов с помощью гранулярной онтологии и интерпретирующих функций, анализирующих геометрические формы.

Онтология представляет иерархии (таксономии) множеств понятий предметной области и отношений между ними. Базы знаний и данных, основывающиеся на онтологиях, являются важнейшим компонентом информационного обеспечения процессов проектирования изделий и управления производством.

Термин «грануляция информации» ввел Л.Заде [173]. Под гранулой понимается группа схожих, не различимых системой объектов. Многомерные пространственные гранулы объединяют схожие объекты в N-мерном пространстве их параметров.

В отличие от иерархических онтологий, аппарат гранулярных онтологий и нечетких гранулярных онтологий (границы гранул размыты) эффективен в случаях, когда необходимо группировать объекты различной физической

природы [88] и исследовать отношения между объектами, образующими кластеры в многомерном пространстве.

В задачах автоматизации технической подготовки производства гранулярные онтологии удобны для описания отношений между детали-сборочными единицами, маршрутными и операционными техпроцессами, оборудованием, технологической оснасткой, режущим и мерительным инструментом и т.п.

Детали могут быть крупногабаритные и мелкие, корпусные и тела вращения, из алюминия или пластмассы и т.д. Причем классы зачастую пересекаются между собой: деталь типа "тело вращения" содержит шпоночный паз, требующий фрезерных операций, а корпусная деталь может содержать поверхности, которые формируются с помощью расточных операций. Кроме того, на трудоемкость изготовления влияет наличие угловых и радиальных размеров, требуемая точность, шероховатость, твердость.

Предлагаемый алгоритм осуществляет грануляцию информации по формам деталей и конструктивно-технологических элементов с помощью аппарата гранулярных онтологий и функций интерпретации семантики графической информации. Получив классификатор, специалисты предприятия могут продолжить декомпозицию гранул по дополнительным интересующим характеристикам и связать их с базой технологических решений, что позволит повысить эффективность САПР в целом.

Гранулярную онтологию некоторого множества деталей формально определим как

$$CO = \langle C, R, \Omega, \Phi, \Gamma \rangle, \text{ где:}$$

C – базовое множество объектов - деталей и КТЭ, обрабатываемых на предприятии, а также их атрибутов;

R – множество отношений на C ;

Ω – множество операций на C и R ;

Φ – множество функций фазсификации значений концептов-свойств;

Γ – множество функций интерпретации геометрических (пространственных) и иных понятий на множествах C и R .

Множество пространственных гранул формируется из базового множества C :

$$C = (C_1, C_2, \dots, C_n),$$

где $C = C_1 \cup C_2 \cup \dots \cup C_n$ и

$C_i \cap C_j = \emptyset, \forall i, j = 1, \dots, n, i \neq j$ – в случае четкой гранулярной онтологии.

Очевидно, при изменении функций фаззификации изменяются и кластеры.

Алгоритм синтеза классификаторов деталей и конструктивных элементов для синтеза групповых технологических решений на основе семантического анализа и описания графической информации заключается в следующем:

1. Подготовка исходных данных. Подбор репрезентативной выборки объектов, заполнение *массива измерений* (таблица 4.2), где M – количество объектов в «обучающей выборке», N – количество параметров.

Таблица 4.2

Структура массива измерений

Объекты	Параметры			
	1	2	...	N
1				
2				
.				
.				
.				
M				

2. Фаззификация значений атрибутов, описывающих объекты чертежей. В результате фаззификации диапазон значений каждого параметра разбивается на конечное (небольшое) количество «нечетких» значений. Предварительная (перед собственно кластеризацией) фаззификация параметров существенно сокращает *объем вычислений*.

2.1. Подбор функций, интерпретирующих числовые и геометрические понятия онтологии на данных изображений.

2.2. Задание функций нечеткой принадлежности (как правило, трапецеидального или треугольного типов).

3. Синтаксическое распознавание типовых конструктивных элементов.

4. Построение N -мерной гистограммы. Гиперпространство содержит не более, чем H^N локусов (ячеек, подпространств), где H – максимальное количество различных *нечетких* значений параметра (например, по каждому из параметров может быть получено одно из $H = 5$ возможных нечетких значений). Для этого в цикле по массиву измерений сформировать массив локусов (таблица 4.3), где для каждого *непустого* локуса указать количество объектов в нем, т.е. объектов, имеющих соответствующие нечеткие значения всех N параметров, в массиве измерений отметить, к какому локусу относится объект.

Таблица 4.3

Массив локусов

Локус	Координаты (нечеткие) локуса				№ кластера	Вес (количество объектов)
	1	2	...	N		
1					1	p_1
.					.	.
.					.	.
.					.	.
L					K	p_L

5. Поиск локальных максимумов в N -мерном пространстве измерений. *Локальный максимум* – максимальное связанное (по отношению соседства в N -мерном пространстве) множество непустых локусов одного веса, такое, что все соседние, не входящие в максимум, имеют меньший вес. Для этого с помощью *локального анализатора окрестностей локусов* просканировать пространство и найти такие локусы, в которых по каждому из N направлений выполняется условие:

$$\forall l = 1..L, i \in \{i_1, \dots, i_N\} (p_{F1(i_1, \dots, h(i)-1, \dots, i_N, l)} < p_l \geq p_{F1(i_1, \dots, h(i)+1, \dots, i_N, l)}) \Rightarrow L_l \in \{C\}, \quad (4.7)$$

где: $l = 1..L$ – индекс локуса в одномерном массиве;

$F1(i1, \dots, iN, l)$ – функция, вычисляющая положение локуса в одномерном массиве;

$i1, \dots, iN$ – координаты локуса в N -мерном пространстве измерений;

$h_i = 0..H$ – положение локуса вдоль i -той оси координат, $i \in \{i1, \dots, iN\}$;

p, p_{F1} – вес локуса (количество объектов, по своим значениям попавших в соответствующее подпространство);

$\{C\}$ – множество локальных максимумов.

Найденные локусы в гиперпространстве параметров служат «центрами» кластеров – скоплений объектов с близкими характеристиками.

6. Создать список кластеров

$$c_k = \langle p_k, x_1, \dots, x_N \rangle, k = 1..K,$$

где: x_1, \dots, x_N – координаты вершины кластера, удовлетворяющего условию (4.7) в N -мерном пространстве.

7. Найти границы кластеров. Для этого из вершины кластера c_k , имеющего координаты x_1, \dots, x_N , рекурсивно запускается тот же локальный анализатор. Если вес $p_{F1(x_1, \dots, h(i) \pm 1, \dots, x_N, l)}$ соседнего по какому-либо i -тому направлению локуса меньше или равен весу локуса, принадлежащего кластеру, то этот соседний локус относится к тому же кластеру:

$$\begin{aligned} \forall L_{F1(x_1, \dots, h(i) \pm 1, \dots, x_N, l)}, i \in \{x_1, \dots, x_N\} p_{F1(x_1, \dots, h(i) \pm 1, \dots, x_N, l)} \leq p_{F1(x_1, \dots, x_N, l)} \\ \Rightarrow L_{F1(x_1, \dots, h(i) \pm 1, \dots, x_N, l)} \in c_k; \end{aligned}$$

где: N – количество параметров;

$h_i = 0..H$ – положение локуса вдоль i -той оси координат, $i \in \{x_1, \dots, x_N\}$;

H – количество возможных значений i -го параметра;

l – индекс локуса в одномерном массиве локусов;

$L_{F1(x_1, \dots, h(i) \pm 1, \dots, x_N, l)}$ – локус соседний с анализируемым;

c_k – кластер с номером k .

Очевидно, таким же образом процесс кластеризации можно осуществлять в каждом из полученных кластеров – фаззицифицировать и кластеризовать, по вышеописанному алгоритму не всю «обучающую» выборку, а уже полученный

кластер, что дает возможность более детально анализировать интересные области пространства измерений и детализировать классификацию объектов.

Функция, определяющая принадлежность *нового* объекта тому или иному кластеру – это мера относительной близости, вычисляемая как расстояние в N -мерном метрическом пространстве между объектом и каждым из «центров» кластеров.

Вычислительная сложность процедуры кластеризации (пп. 4 – 6 алгоритма) не превосходит значения функции O :

$$O(M^2 * N * \log H),$$

где M – количество объектов в «обучающей выборке», N – количество параметров, H – максимальное количество различных нечетких значений параметра, т.е. алгоритм заканчивает работу за $(M^2 * N * \log H)$ шагов. Следовательно, его сложность не является экспоненциальной (не экспоненциально зависит от количества объектов и измерений).

4.7. Выводы по главе

В главе изложены разработанные методы семантического анализа и вербализации графической информации:

1. Разработана модель изображения, обеспечивающая анализ, сопоставление и описание изображений, в виде многослойного атрибутивного графа, содержащего несколько информационных слоев, связанных пространственными отношениями, где каждый слой состоит из вершин и ребер со своим набором атрибутов, с целью семантического анализа изображенных объектов и поиска аналогов. Модель изображения в виде многослойного атрибутивного графа отличается от известных тем, что с целью получения более информативного описания изображений, включает границы областей, скелетоны, контуры и наборы их атрибутов формы и пространственной ориентации.

2. Введено понятие лучевого графа как инструмента сопоставления нечётких пространственно нагруженных графов изображений.

3. Разработан метод сопоставления нечетких пространственно нагруженных графов с помощью лучевого графа, учитывающего пространственную ориентацию ребер и анализирующего текущие ситуации процесса сопоставления на заданную глубину.

4. Разработан метод анализа и описания изображений объектов с иерархической структурой с помощью дескрипционной логики $ALC(GI)$, расширенной на область данных, представленных многослойными атрибутивными графами, с целью упрощения формулировки запросов на поиск и повышения уровня непроизводных (атомарных) понятий. В частности, на изображениях логика $ALC(GI)$ сокращает область интерпретации понятий за счет перехода от пиксельного представления данных к атрибутивным графам. Например, вместо перебора множества пикселей изображения или множества отрезков будут проверяться только цепочки, соединяющие пары вершин. Собственно описание структуры изображённых объектов формируется путём логического вывода, осуществляемого специализированным рассуждателем (reasoner).

Алгоритм рассуждателя $ALC(GI)$ отличается правилами поиска «связанных» фактов на атрибутивных графах, поскольку возможными интерпретациями атомарных концептов являются не произвольные множества, а множество вершин и множество дуг заданного графа. При проверке выполнимости формул, использующих концепты *Подграф* и *Связный Подграф*, проверяется наличие искомым подграфов в предъявленном графе, ищутся цепочки или циклы нужного вида. В сложных случаях, например, при поиске чертежей по заданному образцу или других объектов со сложной структурой, вступает в силу метод сопоставления атрибутивных графов с помощью *лучевых графов*, автоматически порождающий свои поддеревья поиска.

Немаловажным достоинством рассуждателя $ALC(GI)$ является также возможность автоматически генерировать ДЛ-определение для выделенного

объекта, основываясь на его атрибутивном (под)графе, что существенно упрощает работу пользователей по созданию терминологического раздела базы знаний.

5. Предложена методика контроля корректности определений графической информации, задаваемых экспертом предметной области, которая заключается в контроле выполнимости производного концепта, непересекаемости концептов, корректности терминологии в целом, оценке и регулировании релевантности результатов интерпретации.

6. Классификаторы деталей достаточно широко применяются в конструкторском и технологическом проектировании с целью унификации конструкций и технологической оснастки, регламентированы единой системой конструкторской документации ЕСКД. Но применение их для разработки групповых технологий сдерживается проблемами автоматизации процесса группирования (кластеризации) деталей по их чертежам или 3D геометрическим моделям. Оперативная классификация деталей по геометрическим, техническим, технологическим, производственным параметрам, с помощью разработанного алгоритма и подсистемы синтеза классификаторов деталей, может повысить степень унификации технологических процессов (ТП) и оснастки, повысить обоснованность и качество нормирования труда, оптимизировать планы производства. Для этого, в первую очередь, необходимо решить задачи Image Mining – автоматического извлечения признаков формы и относительного расположения структурных элементов из чертежей, 3D геометрических моделей, технологических эскизов.

Таким образом, за счет семиотического подхода разработанный теоретический аппарат анализа изображений выстраивается в методологию: методы синтаксического анализа – методы семантического анализа – методы извлечения прагматики (отношений, измерений, трендов).

Стратегия поиска, основанная на дереве понятий онтологии, может повысить также эффективность анализа динамических сцен (структур), содержащих неподвижные и движущиеся объекты – это предстоит исследовать. Предполагается также дополнить синтаксис концептов предложенной ДЛ

теоретико-множественными операциями над областями изображений.

5. РАЗРАБОТКА ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ АНАЛИЗА, СОДЕРЖАТЕЛЬНОГО ОПИСАНИЯ И ПОИСКА ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

В главе описана методика и информационные технологии семантического анализа, сопоставления и поиска графической информации, основанные на разработанной автором методологии. Показаны технологии применения методологии в задачах графического поиска чертежей по наброску или эскизу, кластеризации множеств деталей по геометрическим характеристикам. Приведены структурная схема программного комплекса анализа графической информации и результаты экспериментов.

5.1. Программный комплекс анализа изображений

Методика практического применения разработанной методологии заключается в следующем:

1. Описание ограничений и условий применения – общей характеристики изображений объектов, фоновых (сопутствующих) объектов и помех, путем инициализации рекомендуемых значений пороговых величин и экспертных правил.
2. Выделение первичных признаков.
3. Описание понятий и отношений прикладной области на языке дескрипционной логики изображений.
4. Описание плана обработки информации.

4.1. Цветовая сегментация и выделение морфологических элементов изображения.

4.2. Аппроксимация контурных и граничных линий.

4.3. Выделение и аппроксимация скелетонов.

4.4. Формирование синтаксической модели изображения.

5. Распознавание образов путем сопоставления моделей.

6. Семантический анализ с помощью дескрипционной логики $ALC(GI)$.

7. Формирование описания изображённых объектов, их деталей и отношений.

8. Конкретизация методики до технологии и руководства пользователя.

На рисунке 5.1 показана структура программного комплекса анализа графической информации.

Состав:

1. Интерфейс пополнения знаний – предназначен для задания экспертом терминов и определений предметной области графической информации.

2. База знаний – геометрические функции, расчетные процедуры, справочники, термины, определения, правила предметной области изображений.

3. Векторизатор - выделяет первичные информативные признаки.

4. Интерпретатор – предназначен для анализа первичных признаков, контроля корректности заданных определений и интерпретации определений на данных изображения.

5. Блок управления диалогом – реализует стратегии диалога и поиска.

6. База данных – рисунки, эскизы, чертежи, фотоснимки, результаты анализа, описания и сопоставления информации.

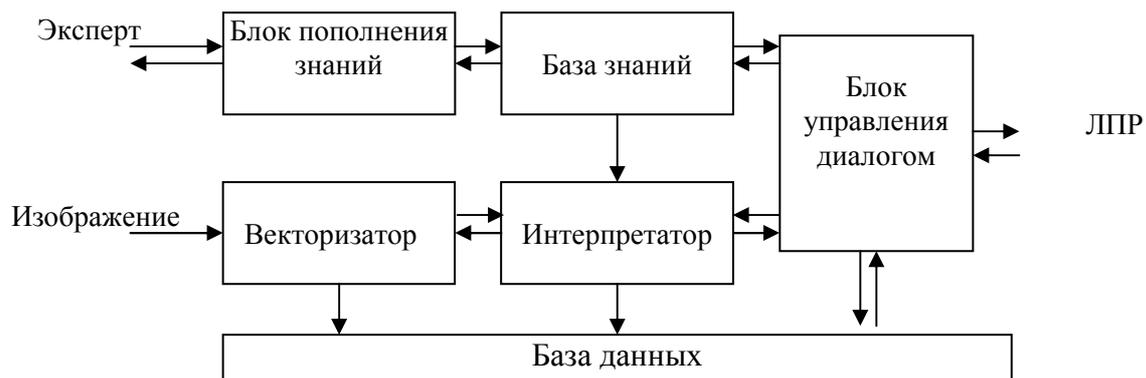


Рисунок 5.1 – Структурная схема программного комплекса семантического анализа изображений

Привязка к предметной области и задачам осуществляется экспертом с помощью блока пополнения знаний без перепрограммирования модулей комплекса.

5.2. Векторизация архивных чертежей

Исходя из обзора (глава 1, п. 1.3) можно выделить следующие требования к векторизатору:

- обработка нечетко прорисованных линий разной толщины и качества вычерчивания;
- распознавание штриховых, штрихпунктирных линий, мелких объектов (цифры, буквы, знаки) и т.п.;
- работа в условиях некачественной бумаги (калька, синька), наличия пятен и потертостей;
- минимизация количества создаваемых отрезков и дуг;
- минимум настроек, которые необходимо сделать пользователю для получения приемлемого результата на конкретном наборе чертежей.

Основные этапы технологии:

1. Тернарная сегментация чертежного изображения.
2. Выделение границ и синтез скелетона.
3. Аппроксимация.
 - 3.1. Аппроксимация прямыми.
 - 3.2. Аппроксимация В - сплайнами.
 - 3.3. Аппроксимация прямыми и дугами.
4. Построение нечеткого графа скелетона.

На рисунках 5.2 – 5.5 приведены примеры работы автоматического векторизатора «AutoPhoto» [56], основанного на разработанной методологии.

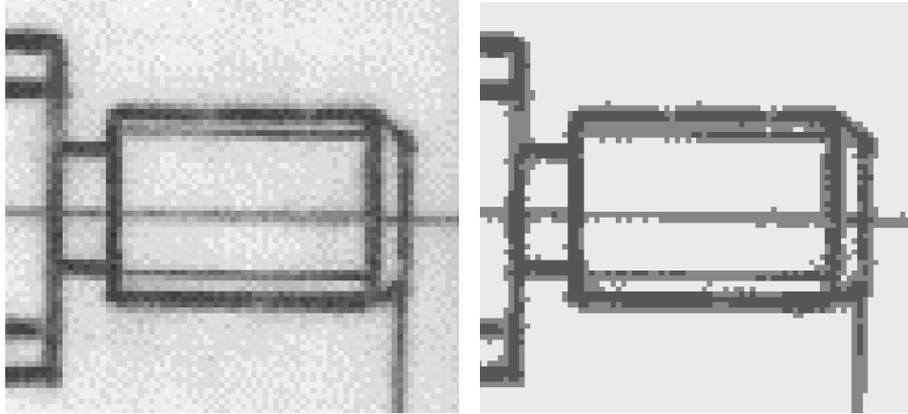


Рисунок 5.2 - Переход к нечеткому тернарному (трехградационному) изображению

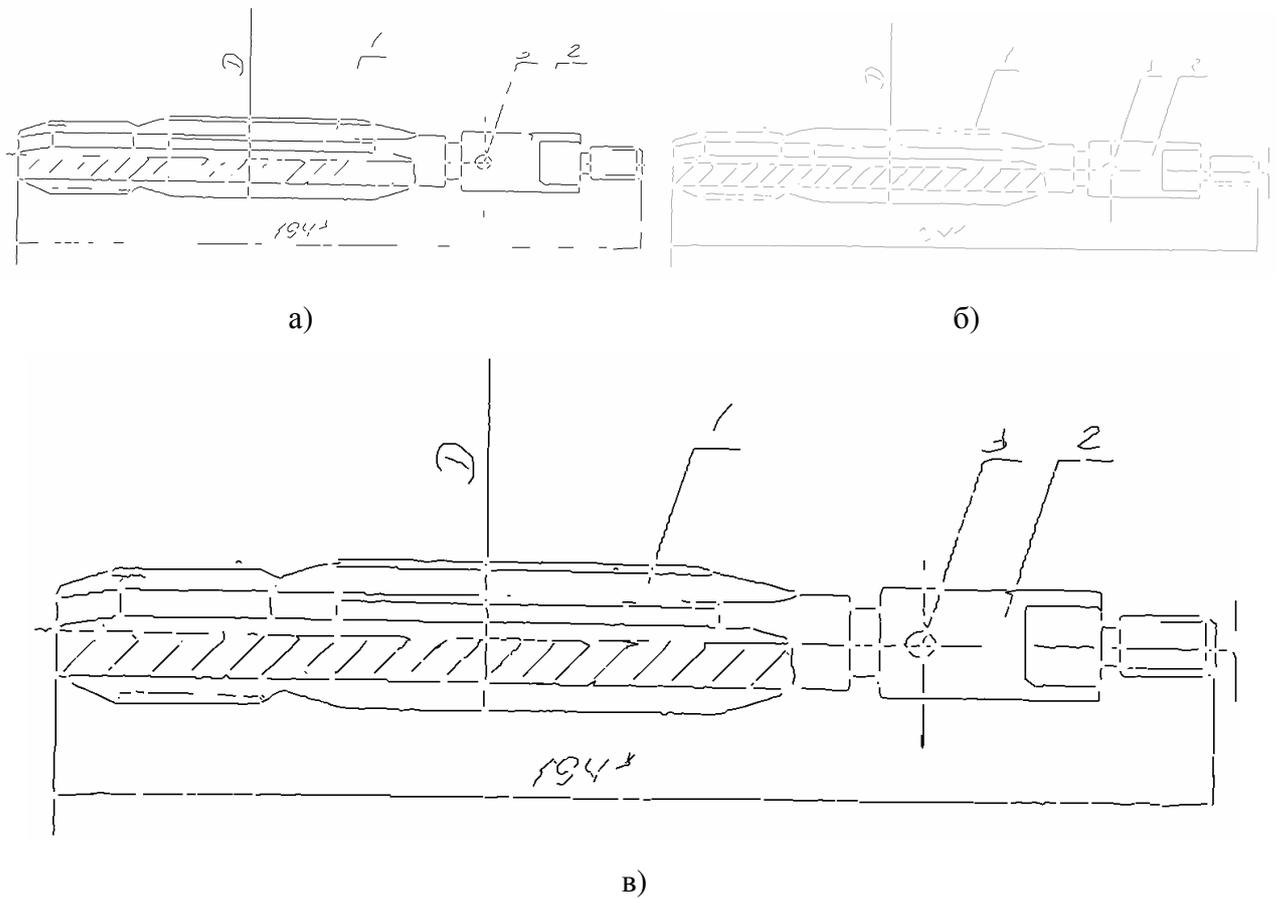


Рисунок 5.3 - Формирование скелетона: а) скелетон черного; б) скелетон серо-черного; в) новый скелетон

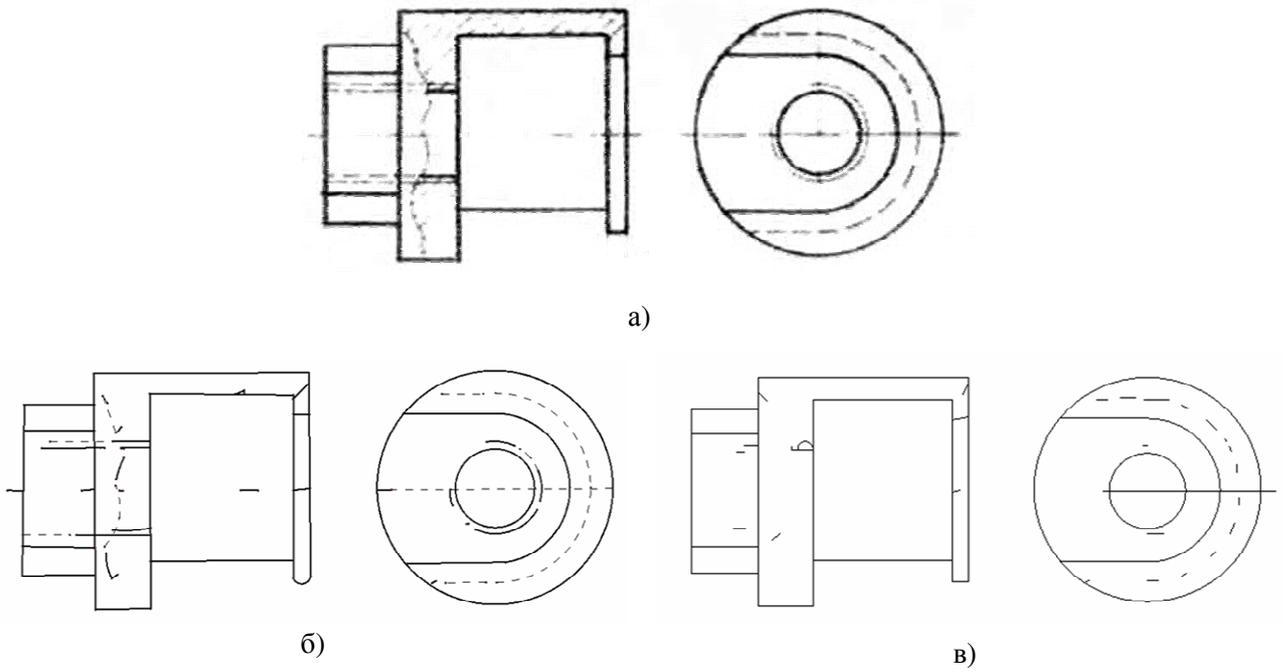


Рисунок 5.4 - Пример работы: а) исходный растровый чертеж; б) результат AutoPhoto; в) результат системы GTXRaster (England)

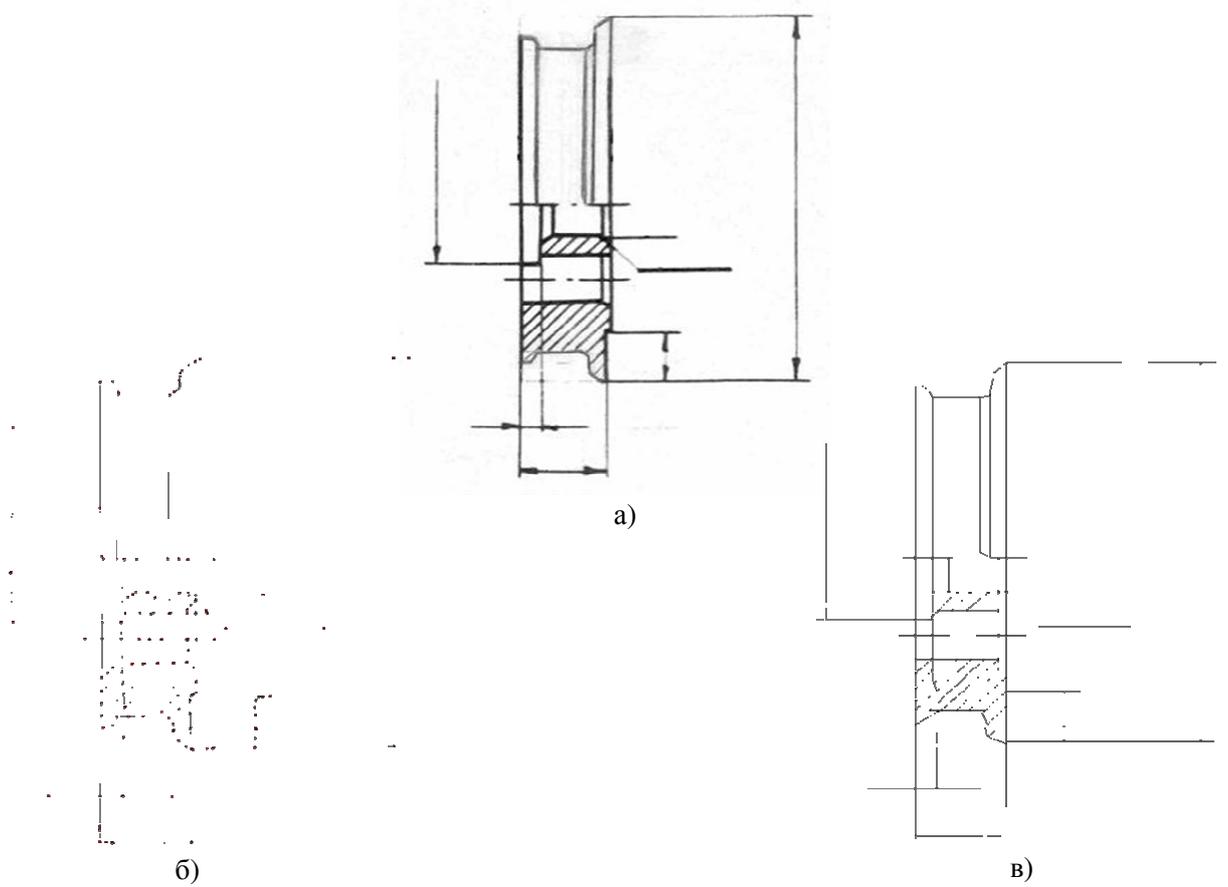


Рисунок 5.5 - Пример работы: а) исходный растровый чертеж; б) результат AutoPhoto; в) результат системы GTXRaster (England)

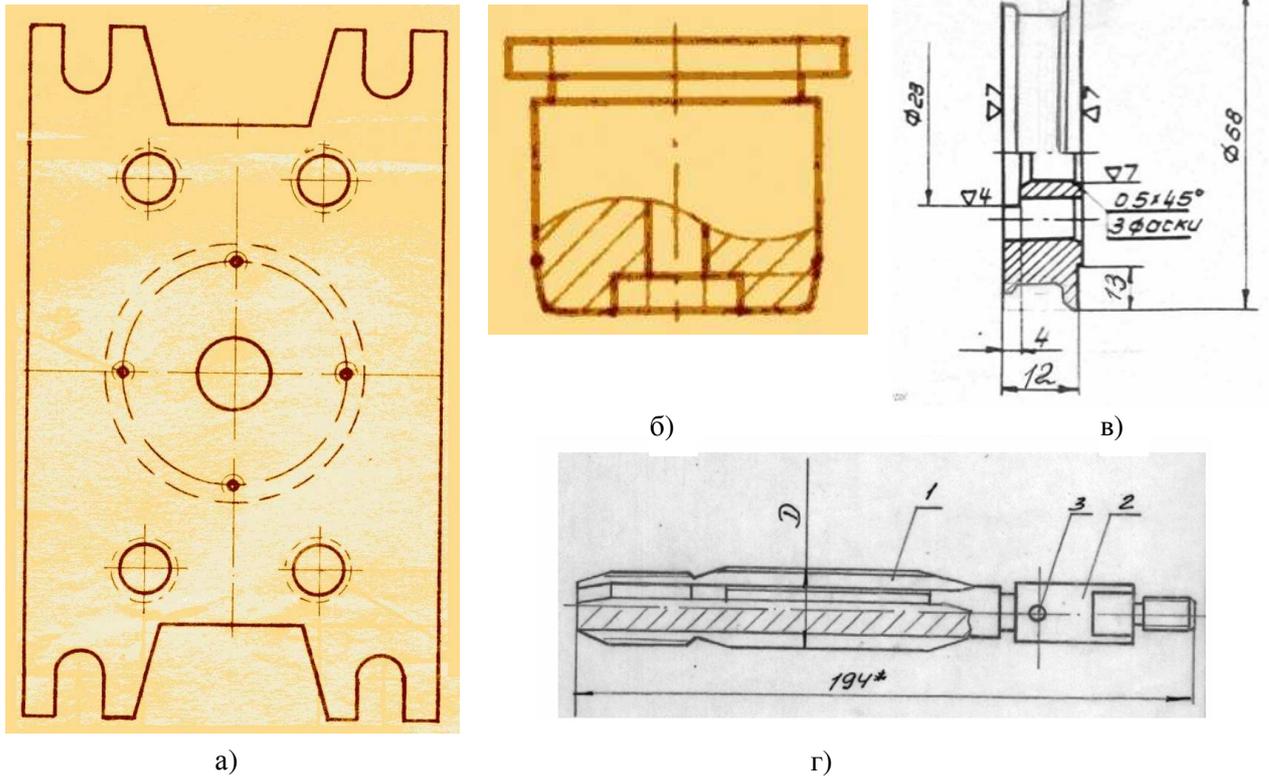


Рисунок 5.6 - Отсканированные растровые изображения чертежей из архива технической документации

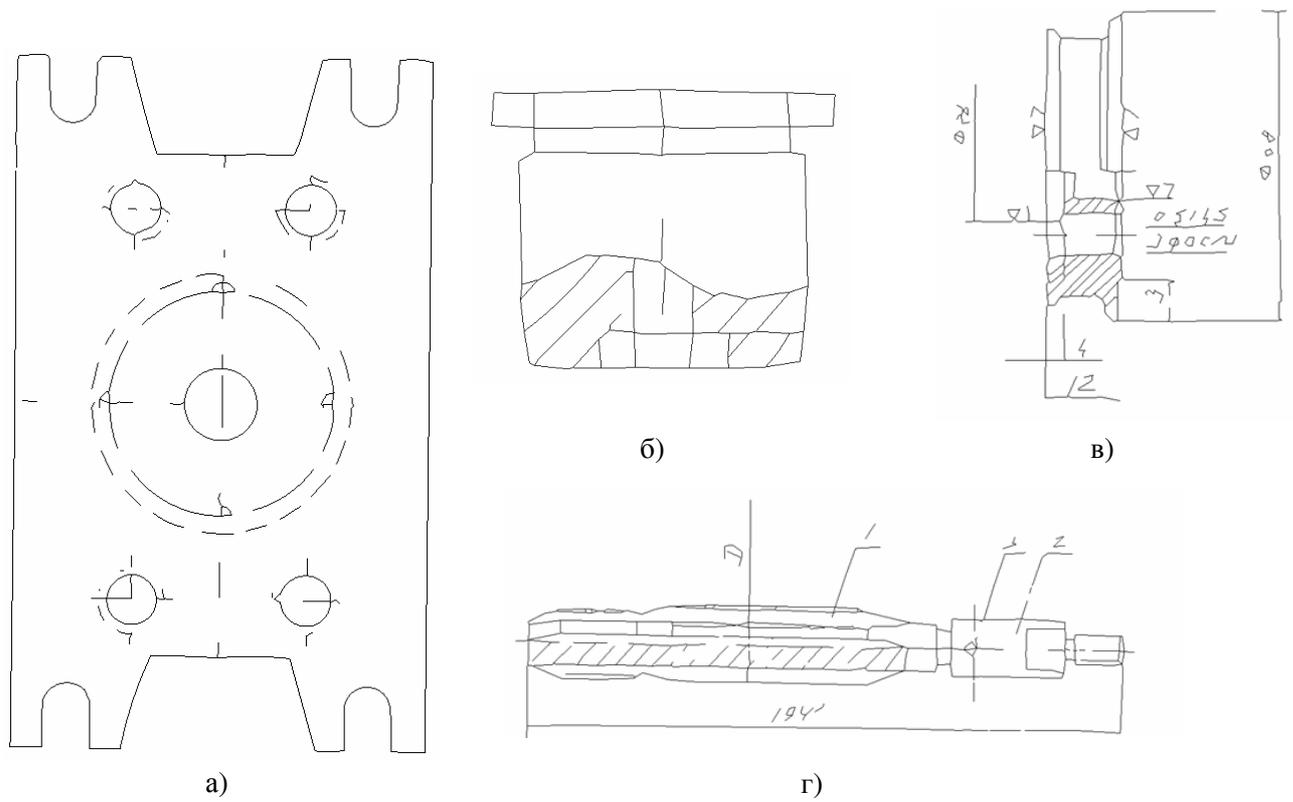


Рисунок 5.7 - Результаты автоматической векторизации растровых изображений чертежей

В таблицах 5.1, 5.2 показаны результаты сопоставления автоматических векторизаторов по количеству создаваемых примитивов в ходе векторизации чертежей, приведенных на рисунке 5.6.

Таблица 5.1.

Сравнение векторизаторов

Рисунок 5.6а	Отрезков	Дуг	Окруж-ей	Рисунок 5.6б	Отрезков	Дуг
AutoPhoto	114	68		AutoPhoto	81	6
Scan2CAD	148	10	5	Scan2CAD	33	
GTXRaster	175	15	5	GTXRaster	52	1
Spotlight Pro	402			Spotlight Pro	92	
WinTopo v3.5	1156			WinTopo v3.5	387	

Таблица 5.2.

Сравнение векторизаторов

Рисунок 5.6в	Отрезков	Дуг	Рисунок 5.6г	Отрезков	Дуг	Окруж-ей
AutoPhoto	353		AutoPhoto	291	12	
Scan2CAD	155	5	Scan2CAD	163	2	
GTXRaster	178	6	GTXRaster	117	19	2
Spotlight Pro	310		Spotlight Pro	321		
WinTopo v3.5	1136		WinTopo v3.5	885		

По результатам тестирования векторизаторов различных фирм можно сделать следующие выводы:

1. В зависимости от желаемого конечного результата тестируемые векторизаторы можно разделить на три группы:
 - а. Повышенное качество векторизации, низкая производительность (AutoPhoto).
 - б. Удовлетворительное качество векторизации, высокая скорость (Spotlight 6.0 Pro).
 - с. Высокая производительность, но низкое качество векторизации (Vextractor, RasterVect, Vectory 6.0).

Таблица 5.3.

Сопоставление функциональности векторизаторов

	Необх-ть бинаризации и полутоновых/цветных изображений	Необх-ть ручной доработки	Оценка качества векторизации	Аппроксимация дугами	Аппроксимация окружностями	Время, затрачиваемое на доработку	Удобство пользов.интерфейса	Цена 1 места (на 2013 г.)
Scan2CAD v8	Да	Не обязательно	Удовлетворительно	Да	Да	от 5 минут	Простой	250 \$
GTXRaster	Да	Не обязательно	Удовлетворительно	Да	Да	от 5 минут	Сложный	Определяется индивидуально
AutoPhoto	Нет	Не обязательно	Высокая	Да	В разработке	от 10 минут	Простой	–
Spotlight Pro 10.0	Да	Обязательно	Средняя	Нет	Нет	от 20 минут	Сложный	3125 \$
WinTopo v3.5	Да	Обязательно	Низкая	Нет	Нет	от 30 минут	Простой	340 \$

2. В конечных результатах всех апробированных аналогах требуется существенная ручная доработка, что приводит к повышению временных затрат.

Результаты сравнительного анализа качества векторизации чертежей, сканированных с бумажных носителей, обработанные известными программами и с помощью разработанной программной подсистемы AutoPhoto, показывают, что подсистема AutoPhoto дает снижение количества отрезков векторизованного чертежа, повышает надежность их выделения на некачественных участках, что сокращает долю ручного труда на последующих этапах занесения архивных чертежей в банк конструкторско-технологических данных.

Следует отметить фактор удобства и доступности для пользователей разных квалификаций – для получения удовлетворительных результатов в AutoPhoto достаточно задать одно значение порога на трассировку и одно значение для аппроксимации отрезками и дугами.

5.3. Графический поиск чертежей в архивах конструкторско-технологических данных

При проектировании новых изделий немаловажным является применение накопленного опыта: существующих компонентов, конструкторских решений, технологических процессов. Известно, что проектировщики проводят около 25% своего времени, осуществляя поиск проектной информации и используя ее в своей работе [159]. Однако найти требуемый документ среди тысяч непросто, учитывая графический характер технической документации, которую образуют чертежи, 3D-модели, сканированные документы, содержащие как текст, так и графику.

В настоящий момент проблема поиска чертежей решается главным образом на основе текстово-числовых атрибутов, кодов классификации и текстовых описаний, создаваемых человеком. При этом, помимо высоких затрат ручного труда при создании и сопровождении архива, неполноты описания геометрии деталей, трудно формулировать запросы, в том числе из-за влияния субъективизма, когда люди описывают одну и ту же деталь по-разному. Исходя из разных задач стоящих перед различными подразделениями (напр.: конструкторскими и технологическими). Стандартные способы поиска чертежей по наименованию, шифру в таких ситуациях не дают удовлетворительных результатов.

В то же время опыт применения систем, ведущих поиск чертежей по задаваемому пользователем изображению-образцу, показывает их превосходство перед традиционными способами поиска [108, 132, 101]. Однако ни одна из существующих систем графического поиска чертежей не обладает достаточным уровнем прозрачности и управляемости, которым должна соответствовать любая поисковая система [113]. Возможной причиной этого является то, что

представление чертежа-запроса, используемое в этих подходах, слабо связано с семантикой содержания.

В конструкторско-технологической деятельности возникают поисковые задачи различного рода. На практике требуется рассматривать конструктивный элемент (канавка, фаска и т.п.) детали как решающий отличающий признак в одних ситуациях и игнорировать в других случаях, изменять поисковое предписание без необходимости перечерчивания чертежа-запроса, устанавливать специальные требования к отдельным составляющим и т.д.

Описываемая система графического поиска чертежей опирается на методы автоматической векторизации чертежей, формирования и сопоставления графических образов, включающих опорные точки, контуры и типовые конструктивные элементы, что существенно повышает качество поиска и, тем самым, позволяет сократить сроки проектирования новых изделий [41, 24, 31, 90, 92, 23, 139].

Методика графического поиска:

1. Векторизация растрового изображения чертежа или наброска: тернарная (трехградационная) сегментация, выделение скелетонов, тройная аппроксимация (вначале аппроксимация прямолинейными отрезками длиной 2-3 пиксела, затем сплайн аппроксимация и наконец аппроксимация в прямые и дуги).

2. Формирование синтаксической модели в виде многослойного атрибутивного графа изображения.

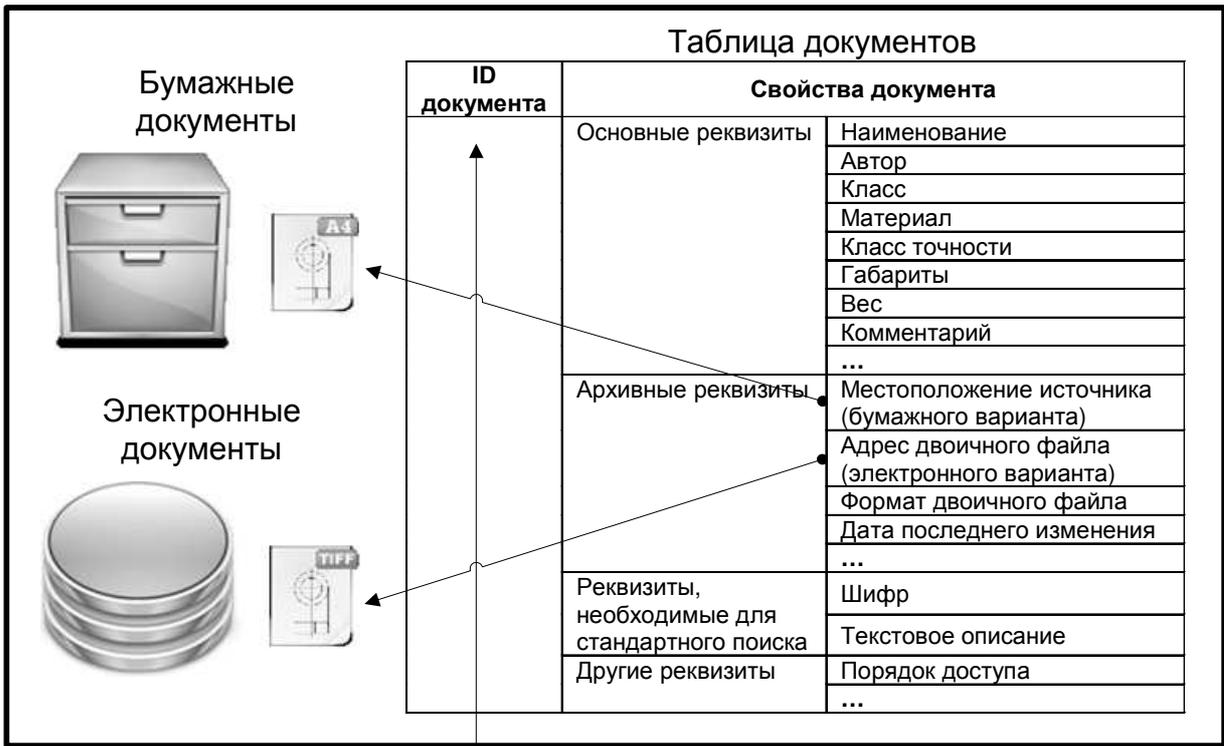
3. Поиск в архиве информации о деталях, аналогичных предъявленной или содержащих фрагмент, подобный заданному. Поиск осуществляется путем сопоставления графических образов чертежей, эскизов, 3D моделей.

4. Управление базой типовых составляющих. При построении графических образов чертежей используются: типовые опорные точки, типовые контуры, типовые конструктивные элементы, унифицированные детали и узлы.

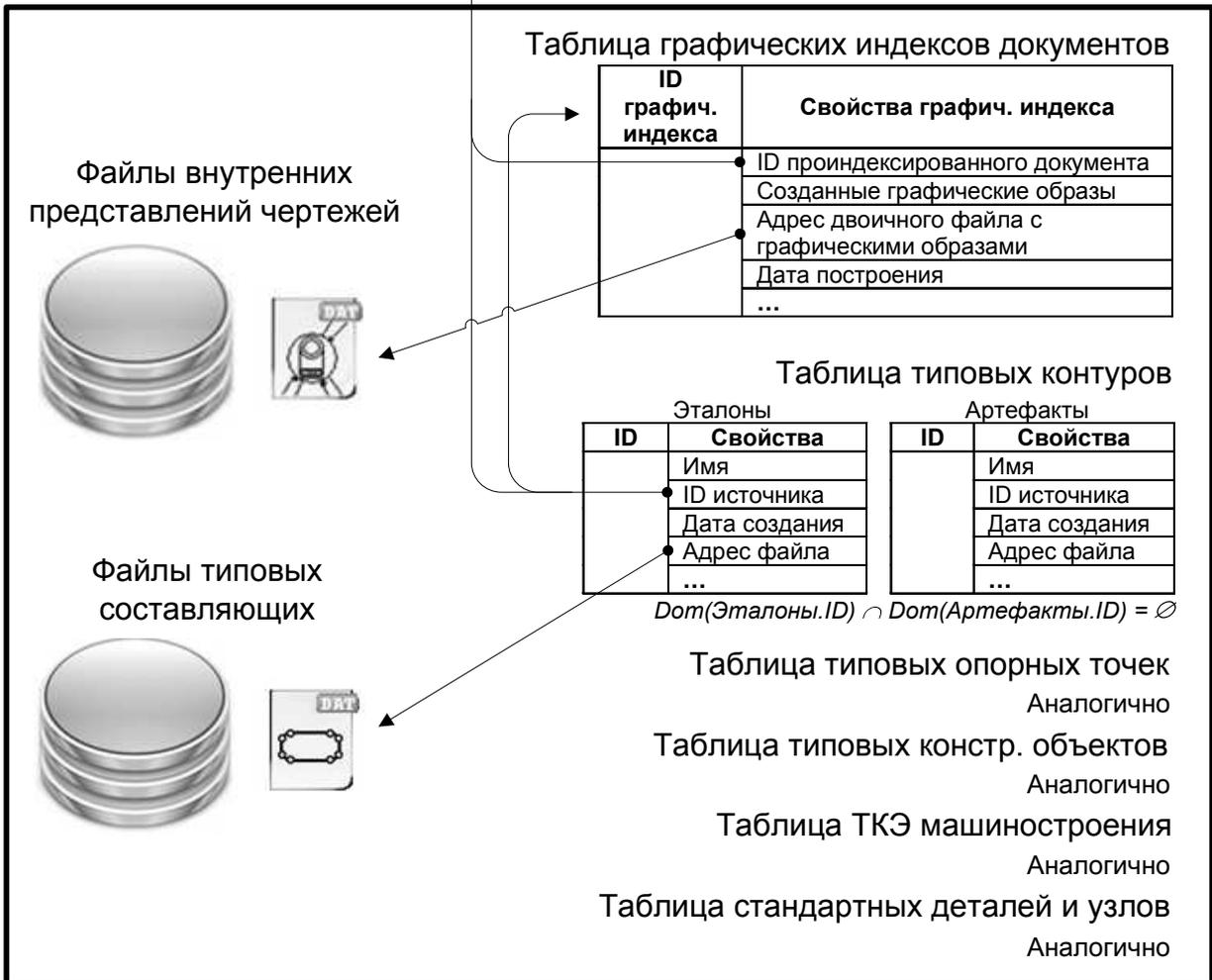
5. Управление индивидуальными стратегиями графического поиска чертежей.

Поиск в архиве информации о деталях, аналогичных предъявленной или содержащих похожий фрагмент, осуществляется путем сопоставления графических образов чертежей, эскизов, 3D моделей с помощью метода "лучевой граф" (п. 4.2).

Поисковая база данных является промежуточным звеном между системой анализа изображений и базой данных (архивом) предприятия (рисунок 5.8). Основу её составляют графические индексы архивных документов. Графические индексы хранятся в отдельных файлах. В общей таблице ведутся аннотации к индексам, включающие адрес файла и идентификатор проиндексированного архивного документа.



Архив (БД предприятия)



Поисковая БД (БД системы графического поиска чертежей)

Рисунок 5.8 - Поисковая база данных

Графический поиск чертежей в электронном архиве построен на базе разработанной методологии и включает следующие основные этапы:

1. Заполнение паспорта. Каждый чертеж при записи в электронный архив, как обычно, снабжается паспортом. В число текстово-числовых атрибутов паспорта входят источник документа, автор документа, наименование изделия, класс изделия, шифр изделия, материал изделия, класс точности, габариты, вес и другие характеристики, комментарий, адрес графического файла. Эти атрибуты используются для стандартного атрибутивного поиска.

2. Графическая индексация. Этап включает следующие шаги:

2.1. Векторизация.

2.2. Выделение вспомогательных элементов чертежа.

2.3. Сегментация чертежа (разбиение на составляющие).

2.4. Распознавание составляющих (поиск типовых конфигураций).

2.5. Формирование графа образа чертежа. Данная обработка применяется как для чертежа-запроса, так и для чертежей в архиве.

3. Поиск аналогов в архиве. Этап включает следующие шаги:

3.1. Выбор класса деталей по текстово-числовым атрибутам (паспорту).

3.2. Сопоставление графических образов чертежа-запроса и архивного чертежа.

3.3. Оценка сходства.

3.4. Если оценка сходства выше заданной, то визуализация аналога и его конструктивных элементов, которые совпали с конструктивными элементами чертежа-запроса.

Типовые составляющие описаний хранятся в базе данных системы, могут пополняться в процессе ее эксплуатации и служат для ускорения поиска, поскольку сокращается количество графических сопоставлений.

Тезаурус системы графического поиска чертежей включает следующие категории понятий.

1. Общие характеристики деталей, например: тело вращения, плоская деталь, штамповка, корпусная, трубчатая, крестовина и т.п.

2. Вспомогательные характеристики: ось, разрез, сечение, количество видов и т.д.

3. Контур: прямоугольник, параллелограмм, трапеция, квадрат, ромб, треугольник, звезда, круг, ...

4. Стандартные детали и узлы (крепежные изделия и др.).

5. Типовые конструктивные элементы (ТКЭ) машиностроения (фаски, канавки, отверстия и др.). Некоторые ТКЭ представлены на рисунке 5.9.

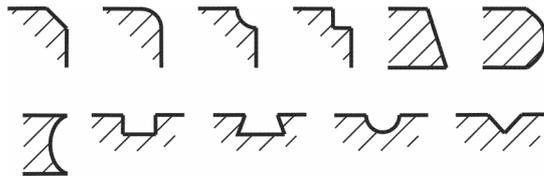


Рисунок 5.9 - Примеры типовых конструктивных элементов чертежа

6. Термины, характеризующие положение и связи элементов, составляющих графический образ: слева, справа, сверху, внизу, в углу, внутри и т.п.

Словари типовых составляющих можно пополнять автоматически, по мере появления новой конфигурации или артефакта (рисунок 5.10).

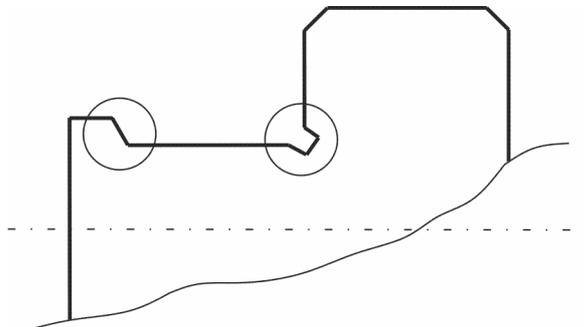


Рисунок 5.10 - Выделение неизвестных ТКЭ

Можно провести аналогию с механизмом восприятия текста (рисунок 5.11).

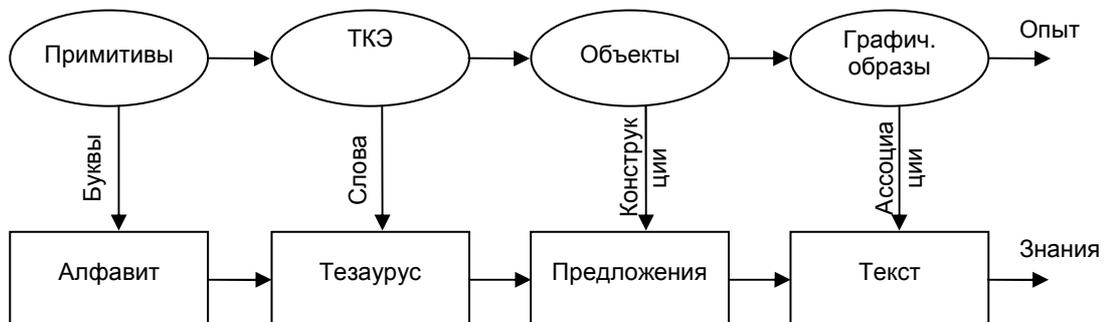


Рисунок 5.11 - Когнитивная модель узнавания и описания рисунков

Рёбра графа контуров отображают отношения вложенности и соседства между контурами (рисунок 5.12). Например: пронумерованные дуги связывают соответствующие элементы следующими отношениями: 1, 2 – внутри; 3 – справа внизу; 4 – справа вверху; 5 – слева вверху; 6 – слева внизу; 7 – в центре; 8 – внизу; 9 – справа; 10 – вверху; 11 – слева.

В граф входят типовые конструктивные элементы: фаски, скругления, канавки, галтели, пазы и т.п.

Стандартные детали и ТКЭ используются двумя способами:

1) предоставляются пользователю в виде каталога для удобного задания уточняющих запросов;

2) встраиваются в графический образ чертежа для сокращения его объема.

Это позволяет системе успешно работать в условиях «зашумленности» чертежа конструктивными элементами: фасками, проточками, галтелями и т.п., за счет того, что:

1) на этапе предобработки чертежей выполняется распознавание и, при необходимости, осуществляется автоматическое удаление ТКЭ с помощью базы графических образов типовых КЭ с целью получения обобщенного представления о детали (рисунок 5.13);

2) на этапе поиска осуществляется рекурсивный предварительный анализ сопоставляемых графических образов на заданную глубину, тем самым достигается выравнивание степеней детализации чертежных объектов.

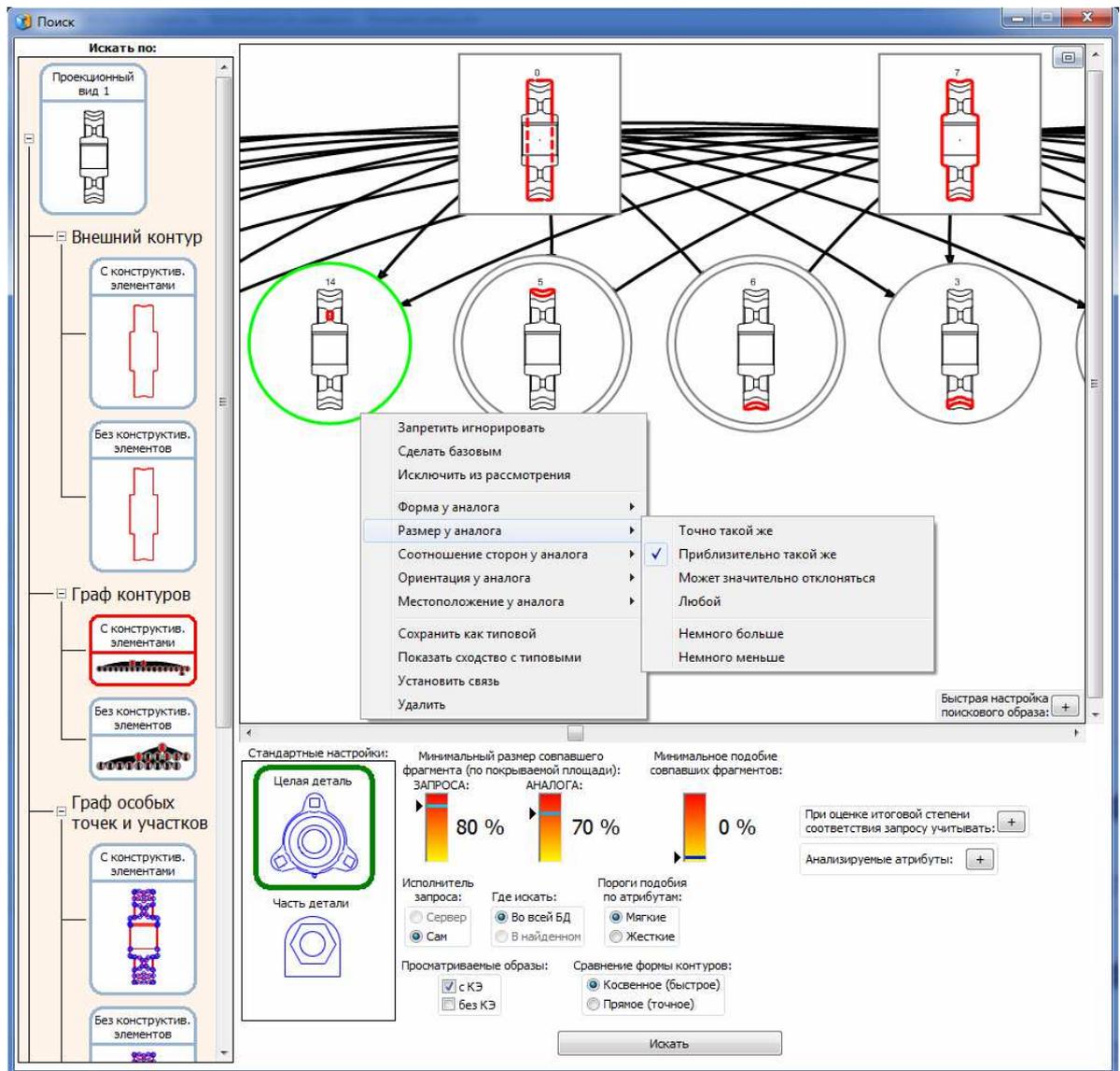


Рисунок 5.14 – Диалоговое окно поиска

Описание эксперимента. Тестовая база данных содержала 2497 чертежей. Часть чертежей была взята с сайтов промышленных компаний (Mitutoyo, Tracerparts, Watts, Ames, Blucher и др.), где они находились в свободном доступе. В качестве запросов использовались как чертежи из тестовой базы, так и упрощенные наброски, созданные в системе КОМПАС-3D.

Результаты поиска система упорядочивает по убыванию оценки релевантности. Релевантность или близкое к ней понятие адекватности может быть оценена экспертным путем, что всегда оставляет некоторые сомнения, или вычислена с помощью формального алгоритма. Поэтому в данном случае за основу были взяты формулы (4.5), (4.6) полноты и точности выдачи, основанные на методе оценки сходства (4.3), (4.4) нечетких нагруженных графов.

Документ считался найденным, если он присутствовал на первой странице выдачи, среди первых 60 лучших результатов.

Таблица 5.4

Сравнение систем графического поиска чертежей

	«CADFind», Love, Barton (Англия, 2004)	Liu, Baba, Masumoto (Китай, Япония, 2004)	Huet, Kern, Guarascio, Merialdo (Франция, 2001)	Tiwari, Bansal (Индия, 2004)	Среднее по максималь ным значениям	Разработан ная подсистем а "GrSearch".	Разработан ная подсистема "GrSearch". Средние значения.	Разработа нная подсистем а "GrSearch". . Разность.
Данные о качестве поиска	Полнота – более 90% после 40 первых результатов .	Полнота – более 80% среди первых 50 результат ов. Точность – 56% среди первых 22 результат ов.	Полнота гистограммн ого метода А – 16.67%. Полнота гистограммн ого метода В – 36.46%. Полнота графового метода А – 47.25%. Полнота графового метода В – 52.16%.	Средняя Полнота – 72,8 %. Точность варьируетс я между 10% и 35%.	Полнота - 73,74 %. Точность - 45,5 %.	Поиск по внешнему контуру Полнота - 95%. Точность - 82%. Поиск по графу контуров Полнота - 100 %. Точность - 89%.	Полнота - 97,5%. Точность - 85,5%.	Полнота - 23,76 % Точность - 40 %.
Быстрод ействие	5 сек.	111.5 сек.	Графовый метод ищет гораздо дольше гистограммн ого.	-		3 сек.		

В задаче графического поиска чертежей наиболее важен именно показатель полноты выдачи ввиду того, что главная цель поиска – использование существующих конструкторско-технологических решений при проектировании новых изделий. В среднем релевантность результата составила: полнота 97,5 %; точность - 85,5 %, что характеризует степень автоматизации поиска как высокую. По сравнению с существующими системами полнота повышена на 23,76 %, а точность на 40 % (таблица 5.4).

Эксперименты показали эффективность применения нечетких множеств, лингвистических переменных, лучевого графа при сопоставлении графов пространственных отношений и описании изображений в задаче графического поиска чертежей (рисунок 5.15), что также позволило расширить функции системы по сравнению с распространенными системами CADFind (Англия), IMShare (Белоруссия) за счет: поиска по фрагменту чертежа или наброску;

визуализации совпавших конструктивных элементов (таблица 5.5). Также в систему включен разработанный автором модуль автоматической векторизации который может работать на полутоновых изображениях, создает меньшее количество геометрических примитивов чем такие распространенные векторизаторы, как Scan2CAD, GTXRaster, Spotlight, WinTopo.

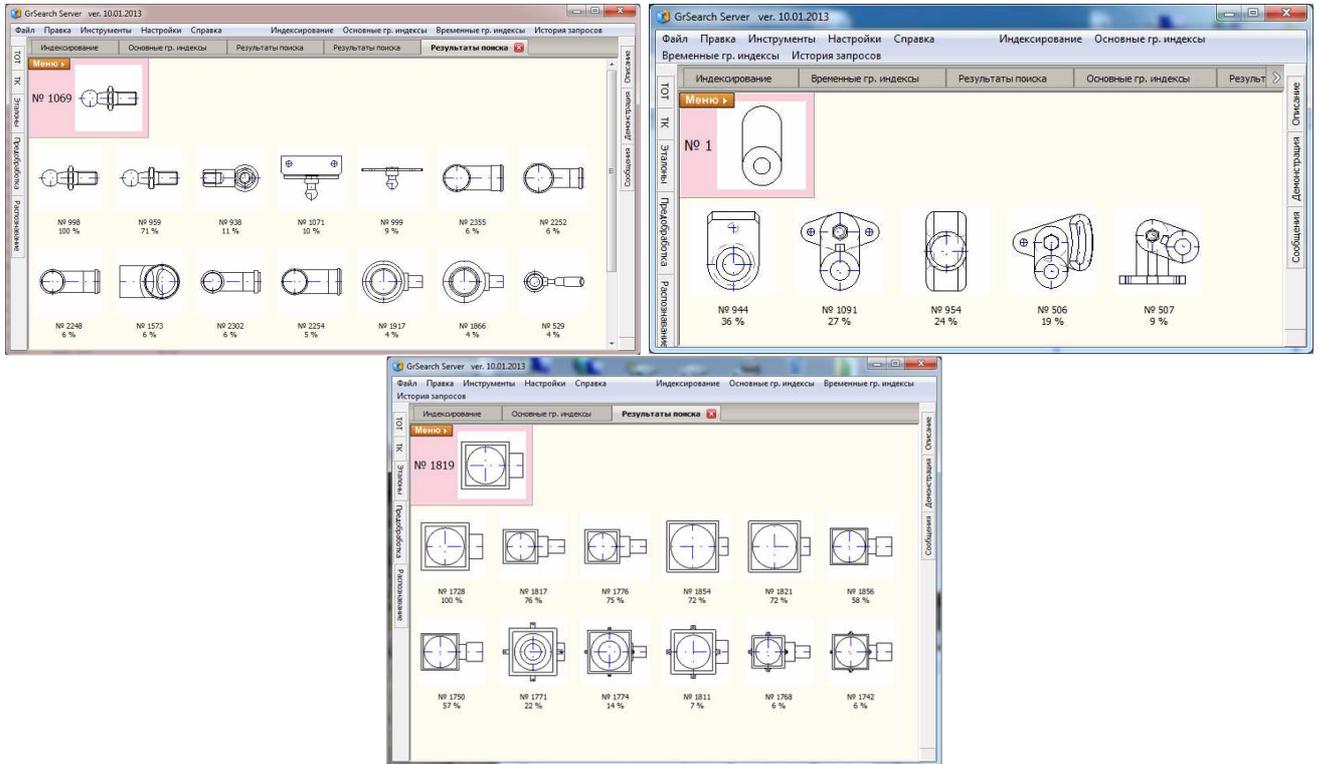


Рисунок 5.15 - Поиск чертежей

Таблица 5.5

Сравнение функциональности

Характеристики	CADFind (Англия)	IMShare (Израиль)	"GrSearch"
Поиск чертежей	+	-	+
Поиск 3D моделей	+	+	+
Разнообразие уровней восприятия	-	-	+
Синтез 3D образов по чертежу	-	-	+
Запрос как фрагмент	-	-	+
Ручная настройка образа запроса	-	-	+
Визуализация совпавших конструктивных элементов	-	-	+
Сканированные	-	-	+

чертежи			
Настройка на предметную область	-	-	+

Предложенный подход позволяет вынести часть работы из этапа поиска в этап графической индексации, за счет чего повышается качество и скорость поиска.

Реализация описанной системы графического поиска возможна двумя способами:

- 1) система работает как надстройка к существующей на предприятии системе;
- 2) система представляет собой отдельное приложение [48].

Таким образом, разработанная когнитивная среда поиска чертежей по содержанию позволяет конкретизировать запросы, реализовывать стратегии, которые наиболее эффективны для текущей поисковой задачи, за счёт настраиваемых графических поисковых образов, различных режимов сравнения и визуализации совпавших элементов. Проведенные эксперименты показывают, что предложенный подход дает хорошие результаты в плане релевантности и удобен в плане управляемости. Точность поиска при уточнении запроса в ходе диалогового взаимодействия с проектировщиком (возможность, отсутствующая в других системах) увеличивается на 10–15%. На практике, это значительно повышает степень повторного использования проектной документации.

Группирование деталей в задачах технической подготовки производства. Классификаторы деталей достаточно широко применяются в конструкторском и технологическом проектировании с целью унификации конструкций и технологической оснастки, регламентированы единой системой конструкторской документации ЕСКД. Но применение их для разработки групповых технологий сдерживается проблемами автоматизации процесса группирования (кластеризации) деталей по их чертежам или 3D геометрическим моделям.

Автоматическое описание и классификация деталей по геометрическим, техническим, технологическим, производственным параметрам могла бы повысить степень унификации технологических процессов (ТП) и оснастки, повысить обоснованность и качество нормирования труда, оптимизировать планы производства. Для этого, в первую очередь, необходимо решить задачи Image Mining – автоматического извлечения признаков формы и относительного расположения структурных элементов из чертежей, 3D геометрических моделей, технологических эскизов.

К признакам формы детали были отнесены такие как: плоскость открытая, плоскость закрытая, ступень, уступ, карман, окно, колодец, канавка, паз и т.д. Другие параметры, влияющие на трудоемкость изготовления: размеры линейные, угловые, радиальные, классы точности, твердости, шероховатости и т.п.

В ходе экспериментов по синтезу классификаторов осуществлялась автоматическая кластеризация 2488 шт. деталей разных категорий (корпусные, тела вращения, плоские) по их чертежам.

В первом эксперименте анализировались такие параметры деталей как количество отверстий, фасок, галтелей, пазов, выступов, окон, колодцев, карманов и т.д. (рисунок 5.16). Распознавание КТЭ на чертежах осуществляла разработанная подсистема графического поиска чертежей GrSearch [25]. Общее количество анализируемых параметров – 12. Каждый параметр мог принимать 4 нечетких значения (очень мало, мало, средне, много) плюс одно значение, равное 0. На этапе определения локальных максимумов было получено 113 кластерообразующих точек. В результате рекурсивного поиска границ кластеров было получено 56 групп деталей. Общее время работы алгоритма с учетом загрузки информации из базы данных составило 50 секунд на персональном компьютере (ПК) стандартной конфигурации.

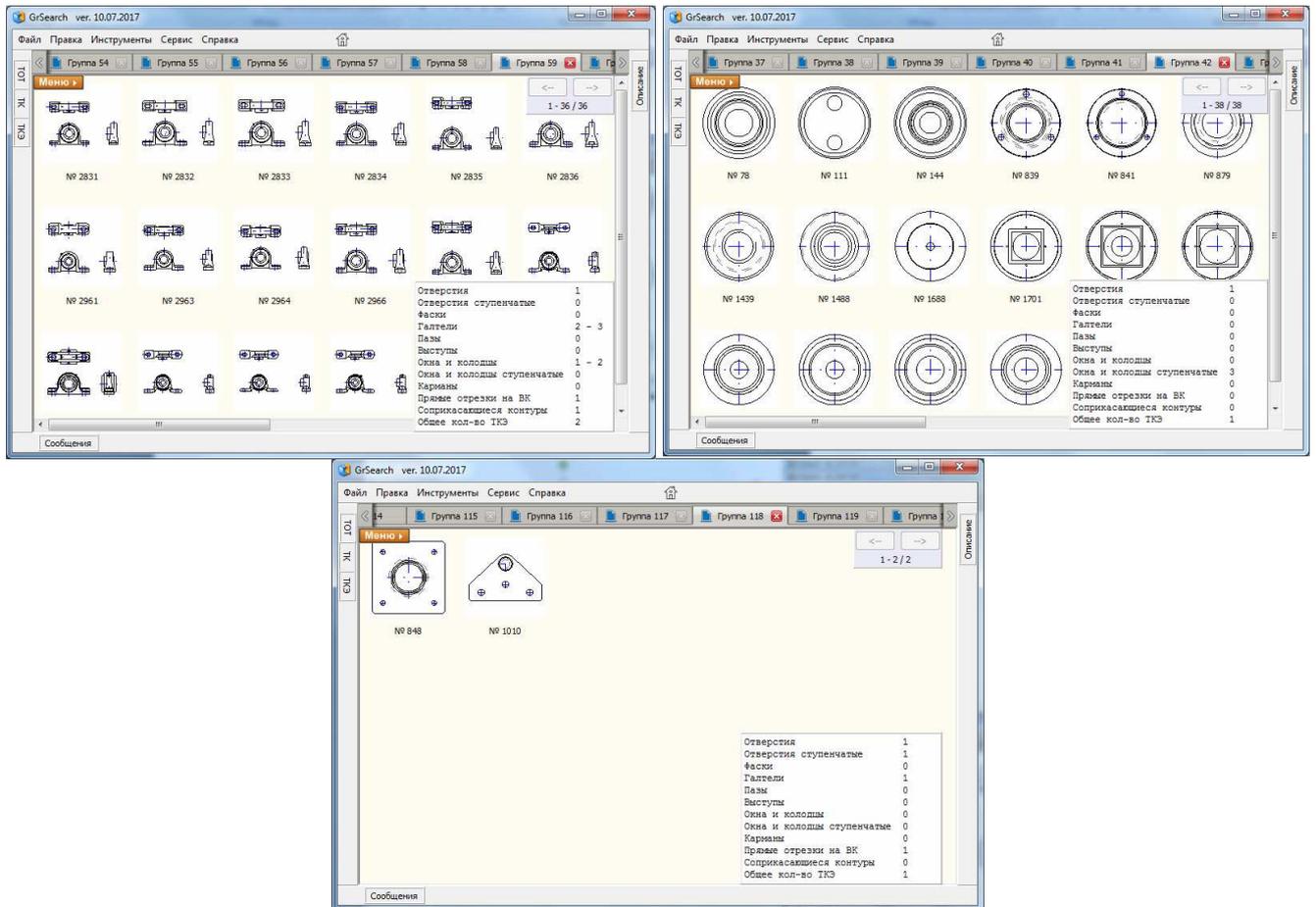
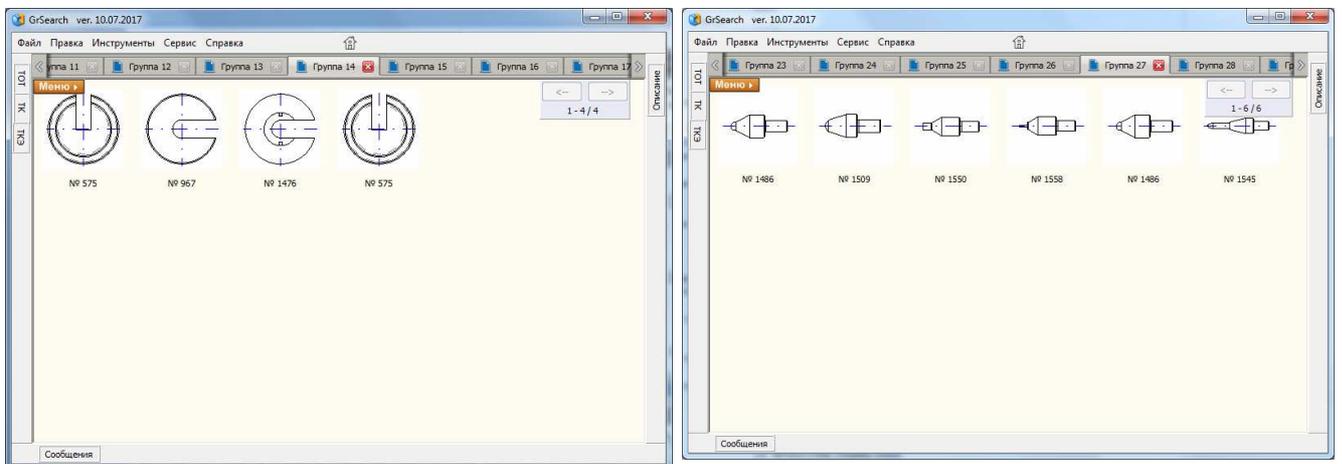


Рисунок 5.16 - Автоматическая классификация деталей по количеству КТЭ

Во втором эксперименте детали группировались по сходству внешних контуров (рисунок 5.17). В качестве меры близости объектов была взята оценка сходства внешних контуров деталей, вычисленная разработанной подсистемой "GrSearch" и выраженная в диапазоне [0-100]. Сопоставление внешних контуров деталей друг с другом (~3 млн. пар) заняло 2.5 минуты на стандартном ПК. В результате работы алгоритма было получено 45 групп деталей.



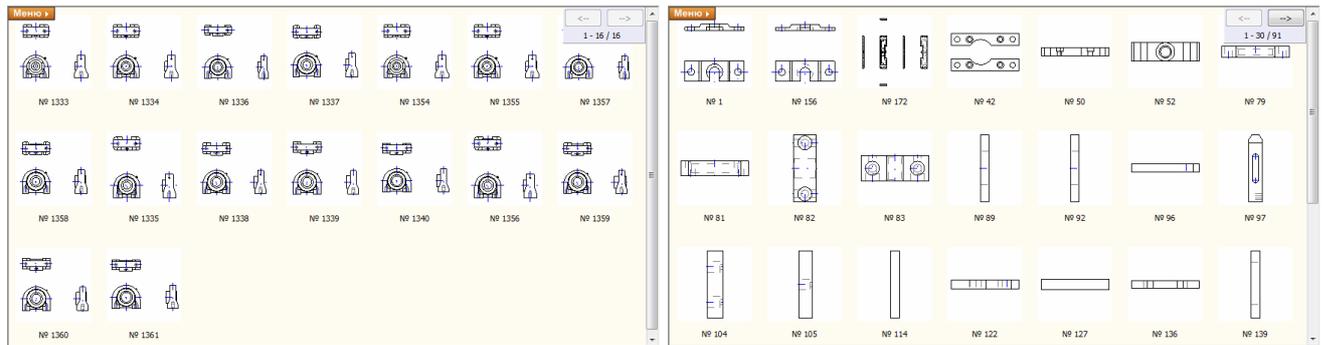


Рисунок 5.17 - Автоматическая классификация деталей по сходству внешних контуров

Таким образом, предложенный алгоритм синтеза классификаторов деталей и конструктивных элементов для синтеза групповых технологических решений отличается тем, что множество признаков объектов включает атрибуты с нечеткими значениями, описывающими геометрические формы, и строится N-мерная гистограмма нечетких значений атрибутов, в которой отыскиваются локальные максимумы, принимаемые за центры кластеров.

Динамическая (исходя из текущей ситуации и опыта, накопленного предприятием) классификация деталей по геометрическим, техническим, технологическим, производственным параметрам позволяет повысить эффективность задач:

- поиска аналогов по геометрическим и другим параметрам;
- выявления сходства и отличий;
- унификации технологических процессов (ТП) и оснастки;
- масштабирования и параметризации групповых техпроцессов;
- прецедентного (CBR – Case Based Reasoning) подхода к проектированию ТП;
- повышения степени обоснованности нормирования труда;
- оптимизации планов производства.

5.4. Выводы по главе

Задачам извлечения семантики и поиска изображений в настоящее время уделяется самое серьезное внимание. Многими исследователями получены достаточно весомые результаты. Однако, остается проблемой получение более полного детализированного описания структур объектов, их анализа, выявления отношений.

В главе описано использование системы анализа изображений, базирующейся на методологии, принципах и подходах изложенных в главах 2 - 4.

Подсистема векторизации, предназначенная для выделения первичных информативных признаков, дает снижение количества отрезков (без потери качества) по сравнению с известными программами. Эффект достигнут за счет того, что заложенные алгоритмы имитируют работу зрительной системы приматов, как наиболее сбалансированной с точки зрения этапности обработки зрительного сигнала и функциональности. Подсистема векторизации отличается интегрально лучевым способом вычисления вектора контурной линии, оптимизацией положения особых точек, трассировкой линий включающей саккадическое перемещение. Качество выделения первичных информативных признаков обеспечивается также контекстной адаптацией пороговых величин и сменой степени интеграции информации.

Созданная под руководством и при непосредственном участии автора экспериментальная версия системы поиска изображений отличается от существующих наличием блока управления сценарием диалога, расширенными возможностями по заданию поискового запроса и его уточнения в зависимости от полученных результатов, а также автоматическим выделением совпавших элементов. Эти новые возможности обеспечены разработанными автором методом сопоставления графов путем формирования лучевых графов, в которых дуги упорядочены по атрибутам пространственной ориентации, для каждой вершины сопоставляемых графов.

Информационная потребность проектировщика может иметь множество тонкостей, не выразимых полностью текстовыми атрибутами или глобальными агрегированными характеристиками изображения. Разработанная когнитивная среда поиска инженерных чертежей по содержанию позволяет пользователю конкретизировать запросы, реализовывать стратегии, которые наиболее эффективны для текущей поисковой задачи за счёт настраиваемых графических поисковых образов, различных режимов сравнения и визуализации совпавших элементов. Проведенные эксперименты показывают, что подход позволяет повысить результаты в плане релевантности и удобен в плане управляемости. Возможность настраивать поисковые образы приводит к улучшению точности поиска на 10–15%. На практике, это значительно увеличивает степень повторного использования проектной информации.

Система позволяет сократить затраты времени и трудоемкость поиска аналогов, выбора прототипов, необходимых конструкторам и технологам при подготовке к производству новых изделий или модернизации выпускаемых, в том числе с целью технико-экономического обоснования принимаемых решений, например, путем использования уже готовых техпроцессов и групповых технологий.

В среднем релевантность результатов поиска составила: полнота 97,5%; точность - 85,5%, что характеризует степень автоматизации поиска как высокую. По сравнению с существующими системами полнота повышена на 23,76%, а точность на 40%. Экспертами (действующими проектировщиками) результаты поиска оценены на 9,0 из 10 баллов.

Разработан алгоритм логического синтеза классификаторов деталей и конструктивных элементов, отличающийся тем, что множество признаков объектов дополнительно включает атрибуты с нечеткими значениями, описывающими геометрические формы, и строится N-мерная гистограмма значений атрибутов, в которой отыскиваются локальные максимумы, принимаемые за вершины кластеров. Автоматически создаваемые

классификаторы деталей и конструктивных элементов позволяют создавать СППР по анализу степени унификации конструкторских решений.

На примере данных задач прослеживается качественный переход от локальных, морфологических признаков на растровом изображении к характеристикам объектов на языке специалиста в предметной области. Эксперименты показали удобство, простоту и прозрачность задания новых правил для изучения закономерностей в изображениях такого рода.

Дальнейшее повышение эффективности системы анализа изображений, связано с распараллеливанием процессов с целью повышения быстродействия.

ЗАКЛЮЧЕНИЕ

В ходе исследования существующих моделей, методов и систем автоматизации процессов анализа и поиска графической информации обоснована целесообразность дальнейшего развития научных основ семантического анализа графической информации для выявления взаимосвязей между первичными морфологическими особенностями (признаками), получаемыми из изображений, и их семантикой с целью повышения релевантности результатов автоматического анализа и извлечения знаний из сложных синтаксических структур, которые характерны для графической информации, широко представленной в СППР.

Основные результаты работы:

1. Впервые предложена концептуальная модель процесса семантического анализа графической информации, основанная на когнитивной модели "треугольник Фреге" восприятия информации человеком и отличающаяся развернутым представлением графических информационных структур с целью совершенствования алгоритмов анализа и преобразования представлений.

2. Разработана новая математическая модель изображения в виде многослойного атрибутивного графа, вершины и ребра которого содержат количественные и качественные значения атрибутов, характеризующих объекты изображения, отличающаяся тем, что с целью получения более информативного описания изображений, включает границы областей, скелетоны, контуры и наборы их атрибутов формы и пространственной ориентации. Позволяет осуществлять автоматический синтез синтаксических моделей графической информации в виде многослойных атрибутивных графов, освобождает экспертов от задания синтаксиса изображений, обеспечивает решение задач семантического и прагматического анализа графической информации.

3. Разработан новый метод сопоставления атрибутивных графов изображений, отличающийся тем, что с целью повышения надежности распознавания и автоматического выявления сходства и отличий между объектами для каждой вершины сопоставляемых графов формируются лучевые графы, в которых дуги упорядочены по значениям атрибутов пространственной ориентации.

4. Впервые разработана онтологическая база знаний изображений на основе дескрипционной логики *ALC*, отличающаяся расширением на область данных, представленных в виде атрибутивных графов, позволяющая осуществлять анализ и поиск пространственно распределенных данных.

5. Разработан новый метод выделения скелетона графических объектов, позволяющий снизить вычислительную сложность за счет разбиения фигуры на выпуклые многоугольники в местах, где граница имеет отрицательный перегиб, и выделения осевых линий полученных выпуклых фигур с последующим объединением их в связные цепочки.

6. Разработан алгоритм логического синтеза классификаторов деталей и конструктивных элементов, отличающийся тем, что множество признаков объектов дополнительно включает атрибуты с нечеткими значениями, описывающими геометрические формы, и строится N-мерная гистограмма значений атрибутов, в которой отыскиваются локальные максимумы, принимаемые за вершины кластеров. Автоматически создаваемые классификаторы деталей и конструктивных элементов позволяют создавать СППР по анализу степени унификации конструкторских решений.

7. Разработана новая методология автоматического семантического анализа и поиска графической информации, основанная на семиотическом подходе к организации знаковых коммуникативных систем, включающая три уровня новых моделей и методов анализа пространственных отношений: синтаксический (морфология, грамматики конструкций), семантический (распознавание и логический вывод), прагматический (вербализация),

развивающая средства автоматического анализа и извлечения знаний из графической информации.

8. Эксперименты показали эффективность программного комплекса для применения разработанной методологии анализа, содержательного описания и сопоставления графической информации.

За счет разработанной методологии семантического анализа, поиска и классификации графической информации:

- в задачах поиска чертежей-аналогов релевантность результатов составила: 97,5% - полнота выдачи и 85,5% - точность. По сравнению с существующими системами полнота повышена на 23,76%, а точность на 40%. При уточнении запроса в ходе диалогового взаимодействия с проектировщиком (возможность, отсутствующая в других системах) точность увеличивается на 10 - 15%.

- автоматическое описание, поиск и классификация деталей по геометрическим, техническим, технологическим, производственным параметрам повышает степень унификации технологических процессов (ТП) и оснастки, обоснованность и качество нормирования труда, позволяет оптимизировать планы производства.

Дальнейшим развитием исследований может быть расширение применения разработанной методологии в интернет-технологиях.

СПИСОК ЛИТЕРАТУРЫ

1. Абламейко, С. В. Обработка оптических изображений клеточных структур в медицине / С. В. Абламейко, А. М. Недзьведь ; Нац. акад. наук Беларуси, Объед. ин-т проблем информатики. – Минск : [ОИПИ НАН Беларуси], 2005. – 155 с.
2. Большой психологический словарь / под ред. Б. Г. Мещерякова, акад. В. П. Зинченко. — М. : Прайм-ЕВРОЗНАК, 2003.
3. Васильева, Н. А. Методы поиска изображений по содержанию / Н. А. Васильева // Программирование. — 2009. — № 3. — С. 1-30.
4. Вежневек, А. Методы сегментации изображений: автоматическая сегментация [Электронный ресурс] / А. Вежневек, О. Баринаова // Компьютерная графика и мультимедиа : Сетевой журнал. – Режим доступа: <http://cgm.computergraphics.ru/content/view/147>. - Загл. с экрана. – Яз. рус.
5. Вежневек, В. П. Использование контурных моделей для выделения черт лица на фронтальном изображении / В. П. Вежневек // Математические методы распознавания образов (ММРО-10) : труды конф. - 2001. – С.179-181.
6. Волкова Г. Д. Методология автоматизации интеллектуального труда / Г. Д. Волкова. - М. : Янус-К, 2013. - 102 с.
7. ГКИНП-02-121-79. Руководство по дешифрированию аэроснимков при топографической съемке и обновлении планов масштабов 1:2000 и 1:5000. М.: ЦНИИГАиК, 1980. – 134 с. (Геодезические, картографические инструкции, нормы и правила).
8. Глезер, В. Д. Механизмы опознания зрительных образов / В. Д. Глезер ; Академия наук СССР, Институт физиологии имени И. П. Павлова. – М. ; Л. : Наука, 1966.
9. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. - М. : Техносфера, 2006. - 1072 с.

10. Горлов И.В., Полетаева Е.В., Калинин Н.А. Групповая технология как основа автоматизации широкономенклатурного производства. Вестник ТвГТУ. – 2016. – Вып. 29, № 1. – С. 59-65.

11. ГУРКМ НКВД СССР пособие по дактилоскопии / ГУРКМ НКВД СССР // Москва – 1935.

12. Двигательный аппарат глаз [Электронный ресурс] // Офтальмологическая клиника «Сфера» профессора Эскиной Э. Н : [сайт]. - Режим доступа: http://www.sfe.ru/v_book_dvig.php

13. Дескрипционная логика [Электронный ресурс] : материал из Википедии. – Режим доступа: https://ru.wikipedia.org/wiki/Дескрипционная_логика

14. Дуд, А. Р. Распознавание образов и анализ сцен : пер. с англ. / А. Р. Дуд, П. Харт. – М. : Мир, 1976 – 368 с.

15. Душков, Б. А. Энциклопедический словарь : Психология труда, управления, инженерная психология и эргономика / Б. А. Душков, А. В. Королев, Б. А. Смирнов. – 2005. – 848 с.

16. Еременко, Г. В. Модифицированный параллельно-симметричный алгоритм утончения [Электронный ресурс] / Еременко Г. В. // Программные продукты и системы : Международный журнал. – 1994. - № 4. – Режим доступа: <http://www.swsys.ru/index.php?page=article&id=1177>.

17. Журавель, И. М. Краткий курс теории обработки изображений [Электронный ресурс] / И. М. Журавель // Matlab& Toolboxes : [сайт]. – Режим доступа: <http://matlab.exponenta.ru/imageprocess/book2/64.php>.

18. Заде, Л. Понятие лингвистической переменной и его применение к принятию приближенных решений / Л. Заде. – М. : Мир, 1976. – 165 с.

19. Заугольнова, Е. В. Алгоритм уточнения предварительной сегментации изображений с нечеткими, слабоконтрастными границами двумерных объектов / Е. В. Заугольнова, Д. В. Юрин. // 16-я международная конференция по компьютерной графике и ее приложениям ГрафиКон'. - Новосибирск, 2006. – С. 487-490.

20. Зинин, А. М. Криминалистическая фотопортретная экспертиза : учебное пособие / А. М. Зинин, Л. З. Кирсанова; под ред. В. А. Снеткова, З. И. Кирсанова. – М. : ВНКД МВД СССР, 1991. – 88 с.

21. Исупов, Н. С. Использование теории графов в задаче распознавания рукописных текстов / Н. С. Исупов, А. В. Кучуганов // Вестник Ижевского государственного технического университета. – 2012. – № 4 (56). – С. 160-162.

22. Исупов, Н. С. Распознавание слитных рукописных текстов с использованием аппарата нечеткой логики / Н. С. Исупов, А. В. Кучуганов // Вестник Ижевского государственного технического университета. – 2012. – № 1 (53). – С. 104-107.

23. Касимов, Д. Р. Поиск чертежей в хранилищах данных с помощью графических образов / Д. Р. Касимов, А. В. Кучуганов // 'IS&IT'11 : тр. Конгр. по интеллект. системам и информ. технологиям : в 4 т. – М., 2011. – Т. 3. – С. 21-28.

24. Касимов, Д. Р. Система графического поиска чертежей / Д. Р. Касимов, А. В. Кучуганов, А. Е. Лопаткин // Интеллектуальные системы в производстве. – 2012. – № 1 (19). – С. 152-157.

25. Касимов, Д. Р., Кучуганов, А. В., Кучуганов, В. Н. GrSearch – Графический поиск. Свидетельство о государственной регистрации программы для ЭВМ № 2013618535, 2013.

26. Керницкий, Д. IMShare – выбери себе модель [Электронный ресурс] / Д. Керницкий // САПР и графика. - 2009. – № 9. - Режим доступа: <http://www.sapr.ru/article.aspx?id=20721&iid=944>

27. Кондаков, И. М Психологический словарь / И. М. Кондаков. – М., 2000.

28. Конев Б.Ю. «Онтология и представление знаний» курс лекций, Date Views 16.04.2013 <http://logic.pdmi.ras.ru/csclub/courses/ontology>, дата создания 25.09.2010.

29. Копытчук Н. Б. Разработка формализованного языка анализа рисков на основе дескрипционной логики / Н. Б. Копытчук, П. М. Тишин, К. В. Ботнар, М. В. Цюрупа // Електротехнічні та комп'ютерні системи. - 2011. - № 2. - С. 103-108.

30. Кучуганов, А. В. Биоинспирированные алгоритмы выделения информативных признаков изображений / А. В. Кучуганов // Известия Томского политехнического университета. - 2012. - Т. 321, № 5 : Управление, вычислительная техника и информатика. — С. 141-145.

31. Кучуганов, А. В. CALS и чертежные архивы [Электронный ресурс] / А. В. Кучуганов // Перспективные информационные технологии и интеллектуальные системы. – 2003. – № 4 (16). – С. 92-98. – Режим доступа : <http://pitis.tsure.ru>.

32. Кучуганов, А. В. RECO – программная система для распознавания старославянских текстов / А. В. Кучуганов, Д. Р. Касимов // Информационные технологии и письменное наследие : материалы Междунар. науч. конф. (Уфа, 28-31 окт. 2010 г.) / отв. ред. В. А. Баранов. – Уфа ; Ижевск, 2010. – С.144-148.

33. Кучуганов, А. В. Автоматизация обработки и семантическое кодирование цифровых изображений / А. В. Кучуганов, П. П. Осколков // Вестник компьютерных и информационных технологий. – 2013. – № 1. – С. 41-44.

34. Кучуганов, А. В. Автоматизация проектирования скульптурных портретов на основе фотоизображений / А. В. Кучуганов, А. Н. Соловьева // Трехмерная визуализация научной, технической и социальной реальности. Технологии высокополигонального моделирования : тр. 2 Междунар. конф. / Ижевск. ин-т компьютер. исслед. – Ижевск, 2010. – Т. 2, Секции 3, 4. – С. 62-66.

35. Кучуганов, А. В. Автоматизированный анализ психологического портрета личности по почерку / А. В. Кучуганов, Г. В. Лапинская // Наука Удмуртии. – Ижевск, 2006. – № 3. – С. 77-82.

36. Кучуганов, А. В. Алгоритмы вербализации изображений на примере словесного портрета человека / А. В. Кучуганов, А. Н. Соловьева // Известия Волгоградского государственного технического университета. Сер. «Актуальные проблемы управления вычислительной техники и информатики в технических системах». - 2012. – № 4 (91). – С. 60-64.

37. Кучуганов, А. В. Биоалгоритмы анализа изображений. Распознавание графов отношений / А. В. Кучуганов // Интеллектуальные системы (AIS'08) и

Интеллектуальные САПР (CAD-2008) : тр. Междунар. науч.-техн. конф. : в 4 т. – М., 2008. – Т. 1. – С. 369-373.

38. Кучуганов, А. В. Биоалгоритмы обработки изображений в геоинформационных системах / А. В. Кучуганов, П. П. Осколков // Интеллектуальные системы (AIS'07) и Интеллектуальные САПР (CAD-2007) : тр. Междунар. науч.-техн. конф. : в 4 т. – М., 2007. – Т. 1. – С. 50-56.

39. Кучуганов, А. В. Биоинспирированные и когнитивные алгоритмы вербализации изображений / А. В. Кучуганов // Информационные технологии в науке, образовании, телекоммуникации и бизнесе IT+SE'2012 : материалы XXXX Юбилейн. Междунар. конф., X Междунар. конф. молодых ученых (Ялта-Гурзуф, 24 мая-04 июня 2012 г.). – Крым, 2012. – С. 105-107.

40. Кучуганов, А. В. Биоинспирированные методы в задачах обработки, вербализации и поиска графической информации / А. В. Кучуганов // Приволжский научный журнал, 2013. – № 1. – С. 49-55.

41. Кучуганов, А. В. Графический поиск чертежей в хранилищах данных / А. В. Кучуганов, Д. Р. Касимов // Прикладная информатика. – 2012. – № 2 (38). – С. 84-92.

42. Кучуганов, А. В. Извлечение из видеоряда кинематической схемы и определение графиков движения человека / А. В. Кучуганов, Д. С. Глухов // Известия ЮФУ. Технические науки. Темат. вып. : Интеллектуальные САПР. – 2012. - № 7 (132). – С. 181-187.

43. Кучуганов, А. В. Использование контуров и границ цветовых областей в проектировании скульптурных портретов по фотографии / А. В. Кучуганов, А. Н. Соловьева // ГрафиКон'2009 : сб. тр. 19 Междунар. конф. по компьютер. графике и зрению (Москва, 5-9 окт. 2009 г.) / МГУ им. М. В. Ломоносова. – М., 2009. – С. 315-316.

44. Кучуганов, А. В. Когнитивное и компьютерное моделирование процессов анализа изображений / А. В. Кучуганов // Пятая международная конференция по когнитивной науке (Калининград, 18-24 июня 2012 г.) : тез. докл. – Калининград, 2012. – Т. 1. – С. 483-484.

45. Кучуганов, А. В. Когнитивный алгоритм построения геометрического остова невыпуклых фигур / А. В. Кучуганов // Приволжский научный журнал. – 2012. – № 3 (23). – С. 84-89.

46. Кучуганов, А. В. Описание содержания изображений на ограниченном естественном языке / А. В. Кучуганов, А. Н. Соловьева // Вестник Ижевского государственного технического университета. – 2012. – № 4 (56). – С. 153-157.

47. Кучуганов, А. В. Построение трехмерной модели с использованием системы выделения контуров на полутоновых изображениях / А. В. Кучуганов // Известия Таганрогского радиотехнического университета. Темат. вып. : Интеллектуальные САПР. – 2006. – № 8. – С. 182-186.

48. Кучуганов, А. В. Распознавание изображений и семантический анализ текстов в задачах патентно-лицензионного поиска / А. В. Кучуганов, М. Н. Мокроусов // Интеллектуальные системы в производстве. - 2010. – № 1. – С. 292-299.

49. Кучуганов, А. В. Распознавание нечетких зрительных образов с помощью лучевых графов / А. В. Кучуганов // 'AIS-IT'10 : тр. конгр. по интеллект. системам и информ. технологиям : в 4 т. – М., 2010. – Т. 1. – С. 258-264.

50. Кучуганов, А. В. Распознавание рукописных текстов / А. В. Кучуганов, Г. В. Лапинская // Современные информационные технологии и письменное наследие : от древних рукописей к электронным текстам : материалы Междунар. науч. конф. (Ижевск, 13-17 июля 2006 г.) / Ижевск. гос. техн. ун-т. – Ижевск, 2006. – С. 98-103.

51. Кучуганов, А. В. Распознавание старославянских текстов методами, основанными на биоалгоритмах анализа изображений / А. В. Кучуганов, П. П. Осколков // Современные информационные технологии и письменное наследие : от древних текстов к электронным библиотекам : тр. Междунар. науч. конф. (Казань, 26-30 авг. 2008 г.). – Казань, 2008. – С. 168-172.

52. Кучуганов, А. В. Распознавание характерных точек по фотографиям в экспертной системе построения словесного портрета / А. В. Кучуганов, А. Н. Соловьева // Известия Волгоградского государственного технического

университета. Сер. «Актуальные проблемы управления вычислительной техники и информатики в технических системах». – 2012. – № 10 (97). – С. 92-97.

53. Кучуганов, А. В. Рекурсии в задачах анализа изображений / А. В. Кучуганов // Pattern Recognition and Image Analysis : New Information Technologies (PRIA-9-2008) : тр. 9 Междунар. конф. : в 2 т. – Н. Новгород, 2008. – Т. 1. – С. 365-367.

54. Кучуганов, А. В. Экспертная система синтеза психологического портрета личности по почерку / А. В. Кучуганов, Г. В. Лапинская // Вестник Ижевского государственного технического университета. – 2008. – № 4. – С. 174-177.

55. Кучуганов, А. В. Экспертная система синтеза психологического портрета личности по почерку / А. В. Кучуганов, Г. В. Лапинская // Современные технологии в учебном процессе : сб. науч. работ науч.-практ. конф. / учеб. центр МВД по Удмурт. Респ. – Ижевск, 2007. – С. 13-17. - В авт. ред.

56. Кучуганов В.Н., Кучуганов А. В., Осколков П. П. Векторизатор полутоновых и цветных растровых изображений AutoPhoto. Свидетельство о государственной регистрации программы для ЭВМ № 2014612661.

57. Кучуганов В. Н., Кучуганов А. В., Соловьева А. В. Программа семантического кодирования изображений “TexSeg”. Свидетельство о государственной регистрации программы для ЭВМ № 2016612059, 2016.

58. Лабутина И.А. Дешифрирование аэрокосмических снимков. М.: АСПЕКТ ПРЕСС, 2004. – 184 с.

59. Левкович-Маслюк Л. Вейвлет-анализ и его приложения [Электронный ресурс] / Л. Левкович-Маслюк, А. Переберин // Алгоритмы. Методы. Исходники. – Режим доступа: http://algotlist.manual.ru/compress/image/leo_lev/lecture3/wav3_0.php. - Загл. с экрана.

60. Луцив В.Р. Объектно-независимый подход к структурному анализу изображений // Оптический журнал. 2008. Т. 75, №11. С. 26-34.

61. Луцив Вадим. Автоматический анализ изображений. Объектно-независимый структурный подход. – Саарбрюккен, Германия: Ламберт Академик Пабблишинг, 2011 г. – 300с.

62. Люк, Ли Глаз да глаз нужен : робот присваивает зрение стрекозы [Электронный ресурс] / Люк Ли // Membrana : Люди, идеи, технологии : [сайт]. – Режим доступа: <http://www.membrana.ru/articles/technic/2005/11/18/210000.html>, свободный.- Загл. с экрана. – Яз. рус.

63. Местецкий, Л. М. Непрерывное скелетное представление изображения с контролируемой точностью / Л. М. Местецкий, И. А. Рейер // ГРАФИКОН-2003 : труды 13 междунар. конф. – М., 2003. - С. 246-249.

64. Методы визуализации информации при помощи графов. Раздел 7 : Радиальные и круговые изображения деревьев. Тема 7.2 : Приложения, использующие радиальный алгоритм изображения деревьев [Электронный ресурс] / Отдел систем электронного обучения и мультимедиа ИВТЦ НГУ. – Режим доступа: http://193.124.209.204/default.aspx?db=book_apanovich&int=VIEW&el=1746&templ=I206.

65. Митрофанов С.П. Групповая технология машиностроительного производства. В 2-х т. Т.1. Организация группового производства. – 3-е изд., перераб. и доп. – Л.: Машиностроение, Ленингр. отд-ние, 1983. – 407 с.: ил.

66. Моченов, С. В. Адаптивные алгоритмы векторизации растровых изображений / С. В. Моченов, А. В. Кучуганов // Информационные технологии в инновационных проектах : тр. III Междунар. науч.-техн. конф. (Ижевск, 23-24 мая 2001 г.). – Ижевск, 2001. – Ч. 1. – С. 181-182.

67. Моченов, С. В. Метод адаптивной векторизации фотоизображений / С. В. Моченов, А. В. Кучуганов // Графикон – 2001 : сб. тр. 11-й Междунар. конф. по компьютер. графике и машин. зрению (Н. Новгород, 10-15 сент. 2001 г.) / Нижегород. гос. архитектур.-строит. ун-т. – Н. Новгород, 2001. – С. 227-230.

68. Моченов, С. В. Модель высокопроизводительной системы для выделения контуров полутоновых изображений / С. В. Моченов, А. В. Кучуганов // Интеллектуальные системы (AIS'05) и Интеллектуальные САПР (CAD-2005) :

материалы Междунар. науч.-техн. конф. 3-10 сент. 2005 г. – Дивноморское, 2005. – Т. 2. – С. 353-359.

69. Некоторые подходы к организации содержательного поиска изображений и видеоинформации / Н. С. Байгарова [и др.]. – М. : ИПМ им. М. В. Келдыша РАН, 2002.

70. Онтология и представление знаний [Электронный ресурс] // Computer Science Клуб [Сайт] [Дата публикации: 25.09.2010]. URL: <http://logic.pdmi.ras.ru/csclub/courses/ontology> (Дата обращения 03.03.2014)

71. Организация управления базами визуальных данных / Н.С. Байгарова [и др.]. - Препринт Института прикладной математики им. М. В. Келдыша РАН, 2000. - N 6.

72. Оре О. Графы и их применение. / О. Оре - М.: МИР, - 1965. - 174 с.

73. Основы современной дерматоглифики [Электронный ресурс] // elsys.ru [Сайт] [Дата публикации 30.01.2008]. URL: <http://elsys.ru/downloads/reports/DermatoglyphicsBases.pdf> (Дата обращения 04.06.2014)

74. Поспелов Д.А., Осипов Г.С. Прикладная семиотика // Новости искусственного интеллекта. М.: Изд-во РАИИ, 1999. № 1. С. 9-35.

75. Построение SIFT дескрипторов и задача сопоставления изображений [Электронный ресурс]. – Режим доступа: <http://habrahabr.ru/post/106302/>

76. Программный комплекс для моделирования и анализа распространения шума на урбанизированных территориях / В. А. Алексеев, М. А. Городилов, А. В. Кучуганов, А. А. Коробейников // Интеллектуальные системы в производстве. - 2012. – № 2 (20). – С. 140-143.

77. Прэтт, У. Цифровая обработка изображений : в 2 кн. / У. Прэтт. - М. : Мир, 1982. – кн. 1. - 312 с.

78. Прэтт, У. Цифровая обработка изображений : в 2 кн. / У. Прэтт. - М. : Мир, 1982. - кн. 2. - 480 с.

79. Разработка и реализация алгоритмическо-программной среды для автоматизации получения информации из изображений и ее анализа [Электронный ресурс] // Конкурс русских инноваций : [сайт]. – Режим доступа:

80. Рекурсивные биоалгоритмы анализа изображений / А. В. Кучуганов [и др.] // Интеллектуальные системы : коллект. моногр. – М., 2009. – Вып. 3. – С. 179-187.

81. Релевантность. Информационный поиск. [Электронный ресурс] : материал из Википедии. – Режим доступа: https://ru.wikipedia.org/wiki/Информационный_поиск

82. Робертс, Л. Машинное восприятие трехмерных объектов / Л. Робертс. - М. : Мир, 1973.

83. Симанков В.С., Луценко Е.В., Лаптев В.Н. Системный анализ в адаптивном управлении: Монография (научное издание). /Под науч. ред. В.С.Симанкова. – Краснодар: ИСТЭК КубГТУ, 2001. – 258с.

84. Славин, О.А. Адаптивное распознавание и его применение к системе ввода печатного текста: Автореф. дис. д-ра. техн. наук. - Москва: Институт системного анализа РАН, 2011. - 32 с.

85. Сойфер, В. А. Компьютерная обработка изображений. Ч. 2 : Методы и алгоритмы / Сойфер В. А. // Соросовский образовательный журнал. - 1996. – № 3. - С. 110-118.

86. Составление календарных планов-графиков цехов (межцеховое планирование). [Электронный ресурс]. Режим доступа: <http://www.carmultisystem.ru/sostavlenie-kalendarnykh-planov-grafikov-tsekhov-mezhtsekhovoe-planirovanie.html> (Дата обращения: 20.05.2017).

87. Среднее_арифметическое_взвешенное [Электронный ресурс] : материал из Википедии. – Режим доступа: <http://ru.wikipedia.org/wiki/>

88. Тарасов В.Б., Калуцкая А.П., Святкина М.Н. Гранулярные, нечеткие и лингвистические онтологии для обеспечения взаимопонимания между когнитивными агентами. Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems

(OSTIS-2012): материалы II Международной научно-технической конференции (Минск, 16-18 февраля 2012) / В. В. Голенков (отв. ред.) – Минск: БГУИР, 2012. – С. 267-278.

89. Топорков, А. А. Словесный портрет : практ. пособие / А. А. Топорков. – М. : Юристъ, 1999. – 112 с.

90. Фасхутдинов, А. Н. Графический поиск документов / А. Н. Фасхутдинов, А. В. Кучуганов // Информационные технологии в промышленности и образовании : сб. тр. науч.-техн. конф. – Ижевск, 2009. – С. 243-246.

91. Форсайт, Д. Компьютерное зрение. Современный подход / Д. Форсайт, Д. Понс. - М. : Вильямс, 2004.

92. Фу, К. С. Структурные методы в распознавании образов/ К. С. Фу. - М. : Мир, 1977. - 319 с.

93. Харин, В. В. Кинематический синтез трехмерной геометрической модели на основе связной сетки кривых / В. В. Харин // Высокие технологии 2004 : сб. трудов науч.-техн. форума с междунар. участием : в 4ч. – Ижевск, 2004. – Ч. 1. - С. 162-169.

94. Харина, А. С. Логический синтез модели интерьера по фотоизображениям / А. С. Харина, А. В. Кучуганов // Известия ЮФУ. Технические науки. Темат. вып. : Интеллектуальные САПР. - 2009. – № 4 (93) – С. 124-129.

95. Хрисанфова Е. Н. Перевозчиков И.В. Антропология / Хрисанфова Е. Н. Перевозчиков И.В. // Наука-2005-С. 245

96. Хьюбел, Д. Мозг. / Д. Хьюбел [и др.]. – М. : Изд-во. Мир, 1984. – 280 с.

97. Цифровая обработка изображений в информационных системах: учебное пособие / И. С. Грузман [и др.]. - Новосибирск : Изд-во НГТУ, 2002. - 352 с.

98. Шапиро, Л. Компьютерное зрение. / Л. Шапиро, Дж. Стокман – М. : БИНОМ. Лаборатория знаний, 2006. – 752 с.

99. Шеперд, Г. Этапы // Шеперд, Г. Нейро-биология. - М., 1987. – Т.2. - С. 419-450.
100. Яхьяева, Г. Э. Нечеткие множества и нейронные сети / Г. Э. Яхьяева. – 2-е изд., испр. - М. : БИНОМ. Лаборатория знаний ; Интернет-университет информационных технологий – ИНТУИТ.ру, 2008. – 320 с. – (Основы информационных технологий).
101. 3D Model Retrieval System [Электронный ресурс]. – Режим доступа: <http://3d.csie.ntu.edu.tw/>
102. A (sub) graph isomorphism algorithm for matching large graphs / L. P. Cordella [et al.] // IEEE Transactions on Pattern Analysis and Machine Intelligence. - 2004. - Vol. 26. - №. 10. - P. 1367.
103. Arkin, E. An efficiently computable metric for comparing polygonal shapes / E. Arkin [et al.] // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1991. - Vol. 13, Is. 3. – P. 209-216.
104. Baader, F., D. Calvanese, D. L. McGuinness, D. Nardi and P. F. Patel-Schneider, (eds), (2003). The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, Cambridge.
105. Bay, H. SURF : Speeded Up Robust Features / H. Bay, T. Tuytelaars, L. V. Gool // Computer Vision and Image Understanding. - 2008. – Vol. 110(3). – P. 346–359.
106. Belongie, S. J. Shape matching and object recognition using shape contexts / S. J. Belongie, J. Malik, J. Puzicha // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2002. - Vol. 24, Is. 4. – P. 509-522.
107. Bourbaxis, N. G. A parallel-symmetric thinning algorithm / N.G Bourbaxis // Partem Recognition. - 1989. - Vol. 22, No 4. - P. 387-369.
108. CADFind - 2D and 3D Graphical Design Retrieval System [Электронный ресурс] // [Сайт компании Applied Search Technology Ltd]. – Режим доступа: <http://www.sketchandsearch.com/>
109. CADFind – Design Retrieval The Comprehensive Solution [Электронный ресурс]. - - - - - Режим доступа:

http://www.sketchandsearch.com/downloads/CADFind_design_retrieval_the_comprehensive_solution.pdf

110. Color Quantization [Электронный ресурс] : Wikipedia. – Режим доступа: <http://education.siggraph.org/resources/cgsource/instructional-materials/slidesets/FrontPage/color-quantization>. – Загл. с экрана. – Яз. англ.

111. Comaniciu, D. Mean shift: a robust approach toward feature space analysis / Comaniciu D., Meer P. // IEEE Trans. on PAMI. – 2002. – Vol. 24, № 5. – P. 603–619.

112. Cvetković, D. Eigenspaces of Graphs / Dragos Cvetković, Peter Rowlinson, Slobodan Simić ю. - Cambridge University Press, 1997. – 276 p.

113. D. Tunkelang, Reconsidering Relevance and Embracing Interaction, Bull. Am. Soc. Inf. Sci. Technol. 36, 1 (2009) 20–23. DOI:10.1002/bult.2009.1720360107

114. Description Logic Complexity Navigator. Maintained by Evgeny Zolin. Date Views 16.04.2013 <http://www.cs.man.ac.uk/~ezolin/dl/> стр. 9.

115. Eckstain, R. Interactive Search Processes in Complex Work Situations A Retrieval Framework. – Bamberg: University of Bamberg Press, 2011. – 223 p.

116. Enhancements to High Level Data Fusion using Graph Matching and State Space Search / K. Sambhoos [et al.] // Information Fusion (under revision), 2009.

117. Face Processing : Advanced Modeling and Methods / edited by W. Zhao, R. Chellappa. – Elsevier: Academic Press, 2005. – 768 p.

118. Face Recognition by Humans : Nineteen Results All Computer Vision Researchers Should Know About / Sinha P. [et al.] // Proceedings of the IEEE. – 2006. - Vol. 94, №. 11. - P. 1948-1962.

119. Fonseca, M. J. Sketch-Based Retrieval of Vector Drawings / Manuel J. Fonseca, Alfredo Ferreira, Joaquim A. Jorge // Sketch-based Interfaces and Modeling. – 2011. - Vol. 12. – P. 181-204.

120. Fukunaga, K. The estimation of the gradient of a density function, with applications in pattern recognition / K. Fukunaga, L. D. Hostetler // IEEE Trans. Information Theory. – 1975. - Vol. 21. - P. 32-40.

121. Hlaoui, A. A new algorithm for inexact graph matching / A. Hlaoui, W. Shengrui // Proceedings of 16th International Conference on Pattern Recognition. – 2002. - Vol. 4. - P. 180–183.
122. Hou, S. Structure-oriented contour representation and matching for engineering shapes / Suyu Hou, Karthic Ramani // Computer-Aided Design. – 2008. - Vol. 40. – Is. 1. – P. 94-108. <http://www.inno-expert.ru/projects/?cat=3&id=217&page=1>
123. Hu, M. K. Visual Pattern Recognition by Moment Invariants / M. K. Hu // IEEE Transactions on Information Theory. – 1962. - Vol. 8, Is. 1. –P. 179-187.
124. Hubel, W. L. Visual transduction in vertebrate photoreceptors / W. L. Hubel, M. D. Bownds // Annual Review of Neuroscience. - 1979. – Vol. 2. - P. 17-34.
125. Hubel, D. H. Anatomical demonstration of orientation columns in macaque monkey / D. H. Hubel, T. N. Wiesel, M. P. Stryker // Journal of Comparative Neurology. - 1978. – Vol. 177, Is. 3. - P. 361-380.
126. Hubel, D.H. Sequence regularity and geometry of orientation columns in the monkey striate cortex / D. H., Hubel, T. N Wiesel // Journal of Comparative Neurology. - 1974. – Vol. 158. - P. 267-294.
127. Hudelot C., Atif J. Bloch I. / Alc(f): a new description logics for spatial reasoning in images // 1st International workshop on computer vision + ontologyapplied cross-disciplinary technologies. - 2014 - P. 1-15.
128. Hudelot, C., Atif, J., Bloch, I.: Integrating bipolar fuzzy mathematical morphology in description logics for spatial reasoning. In: European Conference on Artificial Intelligence ECAI 2010. pp. 497–502. Lisbon, Portugal (Aug 2010).
129. Huet, B. Fuzzy relational distance for large-scale object recognition / B. Huet, E. R. Hancock // Conference on Computer Vision and Pattern Recognition (CVPR '98, June 23-25, 1998). – P. 138–143.
130. Huet, B. Line pattern retrieval using relational histograms / B. Huet, E. R. Hancock // Pattern Analysis and Machine Intelligence. – 1999. - Vol. 21, Is. 12. – P. 1363–1370.

131. Huet, B. Relational Skeletons for Retrieval in Patent Drawings / B. Huet [et al.] // Proceedings of the International Conference on Image Processing. - 2001. - Vol. 3. – P. 737-740.

132. ImShape - система поиска 3D-моделей, основанная на сравнении геометрической формы [Электронный ресурс] / [Сайт компании Intermech]. – Режим доступа: <http://www.intermech.ru/imshape.htm>

133. Jain, R. Computer Vision and Visual Information Retrieval / R Jain, A Gupta, 1996 [Электронный ресурс]. – Режим доступа: <http://vision.ucsd.edu/papers/rosenfeld/>

134. Jain, R. Visual Information Retrieval / R. Jain, A. Gupta // Communications of the ACM. – 1997. - Vol. 40. - №. 5.

135. Jain, R., Workshop Report : NSF – ASPA Workshop on Visual Information Management Systems / R. Jain, A. P. Pentland, D. Petkovic [Электронный ресурс], 1995.- Режим доступа: <http://www.virage.com/vim/vimsreport95.html>

136. Jiao, L. An Engineering Drawings Retrieval Method based on Density Feature and improved Moment Invariants / LiLi Jiao, FaJun Huang, Zi Teng // Proceedings of the 2009 International Symposium on Information Processing (ISIP'09, Huangshan, P. R. China, August 21-23, 2009), 2009. – P. 352-355.

137. Jin, J. Chou VORONOI DIAGRAMS for PLANAR SHAPES / J. Jin // Computer Sciences Corporation ; Computer Graphics. -1995. - March.

138. Kant R., Pattanaik L.N., Pandey V. Framework for strategic implementation of cellular manufacturing in lean production environment. Journal of Manufacturing Technology Research. – 2015. – Vol. 6, No. 3-4. – P. 177-191.

139. Kasimov, D. R. Initial level of perception in the task of graphical search of technical drawings / D. R. Kasimov, A. V. Kuchuganov // PATTERN RECOGNITION and IMAGE UNDERSTANDING : 8 Open German-Russian Workshop : proceedings (Nizhny Novgorod, November, 21-26, 2011). – Nizhny Novgorod, 2011. – P. 103-106.

140. Kasimov D.R., Kuchuganov A.V., Kuchuganov V.N. Individual strategies in the tasks of graphical retrieval of technical drawings. Journal of Visual Languages and Computing. – 2015. – Vol. 28. – P. 134-146.

141. Ke, Y. PCA-SIFT : A More Distinctive Representation for Local Image Descriptors / Y. Ke, R. Sukthankar // Computer Vision and Pattern Recognition – 2004. – Vol. 2. - P. 506-513.

142. Kuchuganov, A. V. Automated Synthesis of Person Psychological Portrait by Handwriting / A.V. Kuchuganov, G. V. Lapinskaya // First Forum of Young Researches. In the framework of International Forum "Education Quality – 2008" : proceeding (Izhevsk, Russia, 23 April, 2008). – Izhevsk, 2008. – P. 33-34.

143. Kuchuganov, A. V. Recursions in Image Analysis Problems / A. V. Kuchuganov // Pattern Recognition and Image Analysis. – 2009. – Vol. 19, N. 3. – P. 501-507.

144. Kuchuganov, A. V., The description and analysis of graphical information in a description logic formal language. - World Applied Sciences Journal 24 (Information Technologies in Modern Industry, Education & Society): 105-110, 2013, ISSN 1818-4952

145. V.N. Kuchuganov, Verbalization of Reality and Virtuality. Associative semantic, Artif. Intell. Decis. Mak. 1 (2011) 55–66. (in Russian).

146. Kuchuganov, V. N. The closed contour skeleton construction by the golden section method / V. N. Kuchuganov // 11-th Intern. Conf. on Computer Graphics GraphiCon'2001, Nizhny Novgorod. - P. 114-119.

147. Latecki, L. J. Convexity rule for shape decomposition based on discrete contour evolution / L. J. Latecki, R. Lakämper // Computer Vision and Image Understanding. – 1999. - Vol. 73, Is. 3. – P. 441-454.

148. Li Q., Luo S., Shi Z. Semantics-based Art Image Retrieval Using Linguistic Variable // Fuzzy Systems and Knowledge Discovery 2007 : Proc. Intern. Conf. – 2007. – Vol. 2. – P. 406-410.

149. Li, S. Z. Handbook of face recognition / Li S. Z., A. K. Jain. – NY, Springer Science+Business Media, 2005. – 395 p.

150. Liu, R. Attributed Graph Matching Based Engineering Drawings Retrieval / Rujie Liu, Takayuki Baba, Daiki Masumoto // Lecture Notes in Computer Science. - 2004. - Vol. 3163. – P. 378-388.

151. Lowe, David G. Object recognition from local scale-invariant features / David G. Lowe // Proceedings of the International Conference on Computer Vision, 1999. – P. 1150–1157.

152. McMurry, D. Defense Parts Management. Program Update / Donna McMurry // Defense Standardization Program Journal. – 2008. – P. 3-11.

153. Metts G.A., Apigian C.H. Reducing Manufacturing Flow Times: Job Shop Simulation and Reconfiguration. World Review of Business Research. – 2011. – Vol. 1, No. 1. – P. 97-108.

154. Narendra P.M., Goldberg M. A non-parametric clustering scheme for LANDSAT // Pattern Recognition. 1977. V. 9. Is. 4. pp. 207-215.

155. Nefedov, A. Method of the pen trace reconstruction in the handwriting recognition process, with the training possibility / Aleksey Nefedov [Электронный ресурс] : GraphiCon 2007 Proceedings // GraphiCon. Компьютерная графика в России [сайт]. - Режим доступа: <http://www.graphicon.ru/ru/conference/2007/proceedings>

156. Ohlander, R. Picture Segmentation Using a Recursive Region Splitting Method / R. Ohlander, K. Price, D. R. Reddy // Computer Graphics and Image Processing. - 1978. – Vol. 8, Is. 3. – P. 313-333.

157. Ore, O. Theory of graphs / American mathematical society, 190 Hope street, Providence, Rhode Island. 1962.

158. Root, J Performance and Motion Capture /J. Root // The VES Handbook of Visual Effects : Industry Standard VFX Practices and Procedures, 2010 - P. 335-386.

Russel, S. and P. Norvig, 2002. Artificial Intelligence: A Modern Approach, Second Edition. Prentice Hall, pp: 482.

159. S. Allard, K.J. Levine, C. Tenopir, Design Engineers and Technical Professionals at Work: Observing Information Usage in the Workplace, J. Am. Soc. Inf. Sci. Technol. 60, 3 (2009) 443–454. DOI:10.1002/asi.v60:3

160. S. Dasiopoulou, I. Kompatsiaris, M.G. Strintzis. Applying Fuzzy DLs in the Extraction of Image Semantics. Journal on Data Semantics XIV. Lecture Notes in Computer Science. Vol. 5880. 2009. pp. 105-132.

161. Serra, J.: Image Analysis and Mathematical Morphology. Academic Press, New York (1982).
162. Shahin A., Nassibeh J. Group Technology (GT) and Lean Production: A Conceptual Model for Enhancing Productivity. International Business Research. – 2010. – Vol. 3, No. 4. – P. 105-118.
163. Shi, J. Normalized Cuts and Image Segmentation / J. Shi, J. Malik. - University of California at Berkeley, 1997.
164. Sidorova V.S. Histogram Hierarchical Algorithm and the Reduction of the Dimensionality of the Spectral Features Space // Journal of Siberian Federal University. Engineering & Technologies. 2017. V. 10. Is. 6. pp. 714-722.
165. Sobel, I. An isotropic image gradient operator / I. Sobel // Machine Vision for Three-Dimensional Scenes : Academic Press, 1990. - P. 376–379.
166. Solovyova, A. N. Using generic wireframe parametrization in sculptural portrait design / A. N. Solovyova, A. V. Kuchuganov // Education Quality - 2010 : Second Forum of Young Researches. In the framework of International Forum : Proceedings (Russia, Izhevsk, 22 April, 2010). – Izhevsk, 2010. – P. 163-170.
167. Stotz, A. Advancements in Computational Situation Assessment with Dynamic Graph-Based Procedures. : PhD Dissertation / Stotz A. - Dept of Industrial & Systems Engineering, University at Buffalo, February 2009.
168. Stotz, A. Incremental Graph Matching for Situation Awareness / A. Stotz, R. Nagi, M. Sudit // 12th International Conference on Information Fusion, 2009. – P. 452 – 459.
169. XML на сайте Консорциума Всемирной паутины (W3C) [Электронный ресурс] : 2016 – Режим доступа: <https://www.w3.org/XML/>
170. Wang W.-N., Yu Y.-L. Image Emotional Semantic Query Based on Color Semantic Description // Machine Learning and Cybernetics 2005 : Proc. Intern. Conf. – 2005. –Vol. 7. – P. 4571-4576.
171. Wen, Zh. 3D Face Processing. Modeling, Analysis and Synthesis / Wen Zh., Huang Th.S ; edited by M. Shah. – Boston : Kluwer Academic Publishers, 2004. – 164 p.

172. Wu, Z. An optimal graph theoretic approach to data clustering : Theory and its application to image segmentation / Z. Wu, R. Leahy // Pattern Analysis and Machine Intelligence, IEEE Transactions on. – 1993. – Vol. 15, №. 11. - P. 1101-1113.

173. Zadeh L.A. Toward a Theory of Fuzzy Information Granulation and Its Centrality in Human Reasoning and Fuzzy Logic. Fuzzy Sets and Systems. – 1997. – Vol. 90. – P. 111-127.

174. Zhang, T. Y. A fast parallel algorithm for thinning digital patterns / T. Y. Zhang, C. Y. Suen // Communications of the ACM. – 1984. - Vol. 27, №. 3. - P. 236-239.

175. Zhiang, Xiang Color Image Quantization by Agglomerative Clustering / Zhiang Xiang, Gregory Joy // IEEE Computer Graphics. – 1994. – May. - P. 44-48.

176. Zoe Falomir, Ernesto Jimenez-Ruiz, M. Teresa Escrig, Lledó Museros. Describing Images using Qualitative Models and Description Logics / Preprint submitted to SPATIAL COGNITION AND COMPUTATION / Taylor & Francis. - Volume 11, Issue 1, January 2011. P. 45-74 (<http://dx.doi.org/10.1080/13875868.2010.545611>)

ПРИЛОЖЕНИЕ А**(справочное)****Анализ и дешифрирование аэрокосмоснимков**

Дешифрирование – это метод изучения и картографирования местности по ее изображению, который заключается в поиске, обнаружении и распознавании объектов местности на изображениях, установлении по изображению их качественных и количественных характеристик и условном обозначении дешифрируемых объектов на карте или плане [7]. Интерпретация изображений производится на основе определенных зависимостей между свойствами дешифрируемых объектов – дешифровочных признаков. Различают прямые дешифровочные признаки, непосредственно воспринимаемые глазом (размер, форма, тень, тон, цвет, структура), и косвенные признаки, характеризующие объект опосредованно через свойства других объектов.

Классы объектов: площадные, линейные, точечные.

Классы местности: одноцветные, текстурные.

Дешифровочные признаки:

- прямые дешифровочные признаки;
- косвенные дешифровочные признаки.

Некоторые терминологические аксиомы предметной области "Дешифрирование аэрокосмоснимков" в логике $ALC(GI)$. Пользуясь введенными выше понятиями и отношениями, можно конструировать формальные определения для поиска объектов на изображениях. Например:

Дорога - это цветовой сегмент, который имеет серый цвет, широкополосную форму для всех ветвей, линейность границ - "смешанная", извилистость "несильная", скелетон разветвлённый:

Дорога \equiv *Сегмент* $\wedge \neg \exists$ *ВнутриСегмента.Т* $\wedge \exists$ *ЦветСегмента = Серый* $\wedge \exists$ *ФормаСегмента = Широкополосный* $\wedge \exists$ *ЛинейностьГраницы = Смешанная* $\wedge \exists$ *ИзвилистостьГраницы = Несильная* $\wedge \exists$ *Knot = True*.

Дом, Здание - это цветовой сегмент малой площади, имеющий прямолинейные границы:

Дом \equiv *Здание* \equiv *Сегмент* $\wedge \exists$ *ФормаСегмента = Площадной* $\wedge \exists$ *ПлощадьСегмента = Малая* $\wedge \exists$ *ЛинейностьГраницы = Прямолинейная*.

Озеро - это цветовой сегмент тёмносинего цвета, имеющий форму площадного, извилистость границы несильная:

Озеро \equiv *Сегмент* $\wedge \exists$ *ЦветСегмента = Тёмносиний* $\wedge \exists$ *ФормаСегмента = Площадной* $\wedge \exists$ *ИзвилистостьГраницы = Несильная*.

Залив является частью озера, который начинается и заканчивается отрицательным перегибом границы, и отношение высоты образованного сегмента к длине хорды сегмента больше 1 (Залив также вода, залившаяся, вдавшаяся в землю, губа, заводь; он обширнее губы и более относится до моря, озера; а речной, мелкий залив: заводь. Словарь В. Даля):

Залив \equiv *Озеро* $\wedge (\exists$ *ЧастьСегмента* $\wedge (\exists$ *НачальнаяВершина* $\wedge \exists$ *Угол < 0*) $\wedge (\exists$ *КонечнаяВершина* $\wedge \exists$ *Угол < 0*) $\wedge \exists$ [*MaxAreaHeight*, *ChordLength*, *K1*].*P1*)

где атомарные конкретные атрибуты высота и хорда сегмента связаны предикатом *P1* с константой *K1*.

На рисунке А.1 показана типичная конфигурация, которая будет отнесена к озеру, если она имеет цвет синий. Как видим, принадлежность той или иной последовательности (части) понятию "Залив" (рисунок А.2) зависит от константы *K1*, которую визуально подбирает эксперт предметной области, оценивая релевантность.

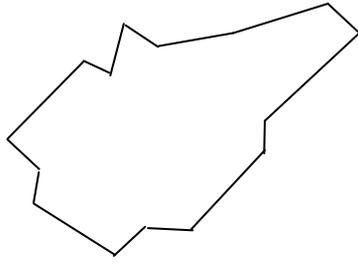


Рисунок А.1 – Конфигурация "Озеро"

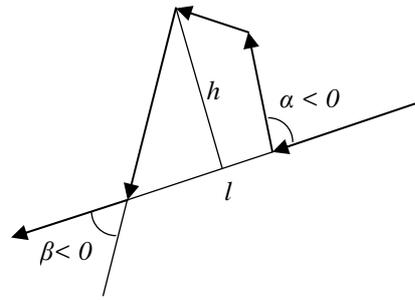


Рисунок А.2 – Определение понятия "Залив"

Корабль – это область изображения, которая имеет форму, близкую к прямоугольнику, ее граница содержит один *острый* угол и окружающий фон темный:

$$\text{Корабль} \equiv \text{Область} \cap (\exists \text{ФормаСегмента.} = \text{Прямоугольник}) \cap (\exists \text{Угол.} = \text{Острый}) \cap (\exists \text{Фон.} = \text{Темный}).$$

Самолет – это область изображения, которая имеет форму описывающего n -угольника, близкую к пятиугольнику, и ее граница содержит пять острых углов или имеет форму четырехугольника и ее граница содержит три острых угла:

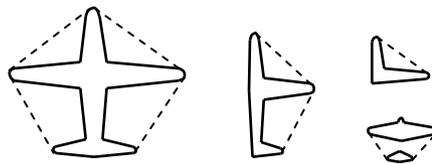
$$\text{Самолет} \equiv \text{Область} \cap ((\exists \text{ФормаСегмента.} = \text{Пятиугольник}) \cap (\exists = 5(\text{Угол.} = \text{Острый})) \cup (\exists \text{ФормаСегмента.} = \text{Четырехугольник}) \cap (\exists = 3(\text{Угол.} = \text{Острый}))).$$


Рисунок А.3 – Варианты описывающего многоугольника при низком качестве изображения

Автомобиль – это область изображения, которая имеет форму, близкую к прямоугольнику, площадь меньше площади самолета или корабля и окружающий фон темный:

$$\text{Авто} \equiv \text{Область} \cap (\exists \text{ФормаСегмента.} = \text{Прямоугольник}) \cap ((\exists \text{ПлощадьСегмента} < \text{Самолет. ПлощадьСегмента}) \cup (\exists \text{ПлощадьСегмента} < \text{Корабль. ПлощадьСегмента})) \cap (\exists \text{Фон.} = \text{Темный}).$$

Лодка – это область изображения, которая имеет форму, близкую к прямоугольнику, имеет малую площадь, ее граница содержит один или два острых угла и окружающий фон темный:

$Лодка \equiv Область \sqcap (\exists ФормаСегмента. = Прямоугольник) \sqcap (\exists ПлощадьСегмента. = Малая) \sqcap (\exists < 3(Угол. = Острый) \sqcap (\exists Фон. = Темный).$

Разработанная автором информационная технология для анализа и описания аэрокосмических снимков заключается в следующем:

1. Описание условий применения – общей характеристики изображений объектов, фоновых (сопутствующих) объектов и помех, путем инициализации рекомендуемых значений пороговых величин.

2. Описание понятий и отношений прикладной области на языке дескрипционной логики изображений $ALC(GI)$ – в специализированном экспертном модуле.

3. Описание плана обработки информации.

4. Конкретизация технологии до руководства пользователя.

Технология, реализующая методологию содержательного описания и поиска графической информации, даёт возможность интерпретировать снимки на нескольких уровнях абстракции. Это позволяет при дешифрировании представлять составные объекты через структуры объектов более низкого уровня, а также применять косвенные дешифровочные признаки, что недоступно при обычной классификации областей изображения. Исходными данными являются ранее векторизованные изображения [33]. Векторизация заключается в сегментации изображения с последующим выделением границ и скелетов полученных однородных областей.

Для анализа и описания простых объектов рассчитываются показатели, соответствующие прямым дешифровочным признакам: *форма* (вытянутость, наклон, разветвленность, извилистость границы, геометрическая правильность границы, плотность точек границы, выраженность полостей), *размер* (площадь абсолютная и относительно других областей), *тон, цвет* (оттенок, насыщенность, яркость), *контрастность*. Каждому из полученных показателей сопоставляется

качественная характеристика, представляемая лингвистической переменной [18]. Функции принадлежности для каждой качественной характеристики могут корректироваться экспертом.

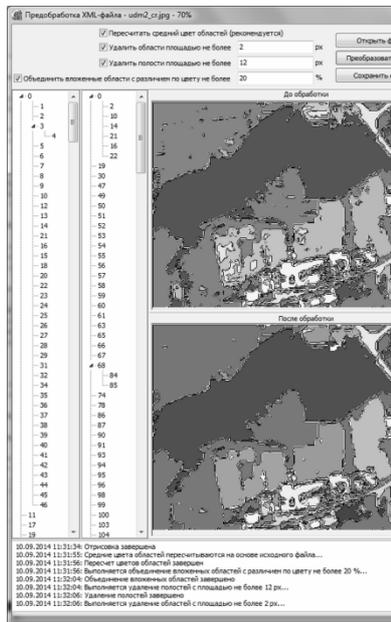
Анализ и описание значимых объектов производится путем выявления на изображении объектов, характеризующихся значимостью для предметной области. В качестве значимых объектов могут выступать текстурные или однородные области (площадные объекты: водоемы, сады и т.п.), цепочки геометрических примитивов (дороги, ограждения, просеки и т.п.) и точечные объекты: колодцы, постройки и т.п.

Автоматизация анализа осуществляется за счет разработанного на основе дескрипционной логики *ALC* формального языка (глава 4), в котором заложены: формальные определения понятий (классов); определения отношений; организация иерархий (подкласс – суперкласс); определения свойств и атрибутов понятий и ограничений на их значения. С помощью логических операций, правил и аксиом проверяется корректность определений и иерархий, а также производится построение сложных понятий из простых, автоматически вычисляются такие отношения, как: расстояние между объектами; направление от одного объекта к другому; закономерности: является ли один объект продолжением другого (слабовыраженная проселочная дорога или тропа, пересохшая речка на аэрокосмоснимке и т.п.), а также закономерности вида: кластер однотипных объектов; расположение сложных объектов (например: домов, окон, отверстий и т.д.) на прямой, на дуге, на узлах сетки.

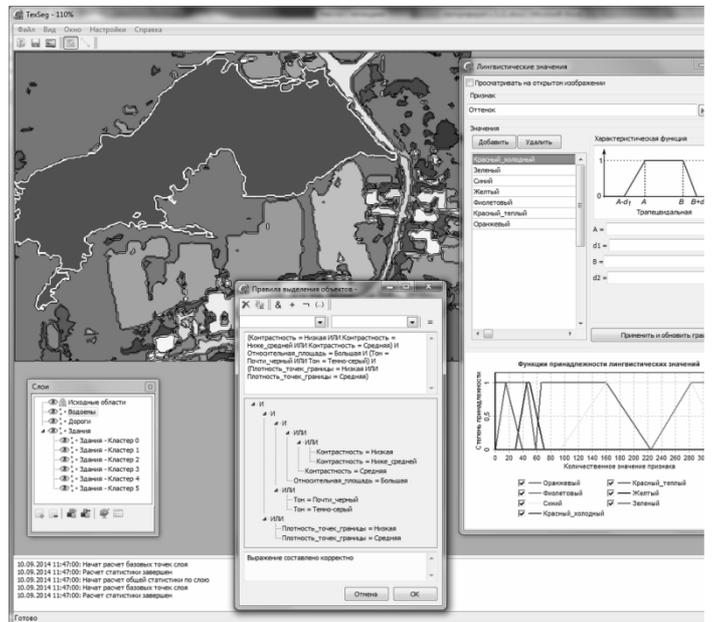
В дополнение к логике *ALC(GI)* реализуются правила, типичные для задачи дешифрирования. Например, *ЕСЛИ «Контрастность = Ниже средней» \wedge «Плотность точек = Высокая», ТО «Объект естественного происхождения»*. Косвенные дешифровочные признаки могут быть также представлены правилами. Например, область грунтовых вод обнаруживают на снимке по полосам растительности, что можно смоделировать правилом с левой частью *«Тип = Растительность» \wedge «Вытянутость = Сильная»*); двор располагается на свободном месте в центре группы близко стоящих однотипных зданий, что

соответствует правилу с левой частью «*Расположение = В центре*», где объект связан с объектом «*Группа однотипных зданий*». В предметных областях, для которых характерна априорно известная пространственная структура объекта, необходимость дополнительного формирования правил может быть исключена [46].

Таким образом, когнитивный подход и биоинспирированные методы позволили повысить релевантность полученных первичных дешифровочных признаков. Модель изображения становится более информативной по сравнению с известными за счет добавления более сложных отношений между структурными элементами объектов. При этом снижается влияние субъективизма, снимаются ограничения на выбор признаков (рисунок А.4, А.5).



а



б

Рисунок А.4 - Пример работы системы семантического кодирования:

а) окно результатов векторизации аэрокосмических снимков; б) пример выделения водоемов на изображении (открыты окно слоев, окно настройки правил выделения объектов, окно настройки лингвистических значений)

В дополнение к низкоуровневым характеристикам (цвет, текстура, форма) применяются количественные и качественные характеристики отдельных однородных областей, текстурных областей и пространственных отношений между ними, а также характеристики, значимые для предметной области (могут быть специфическими в различных задачах).

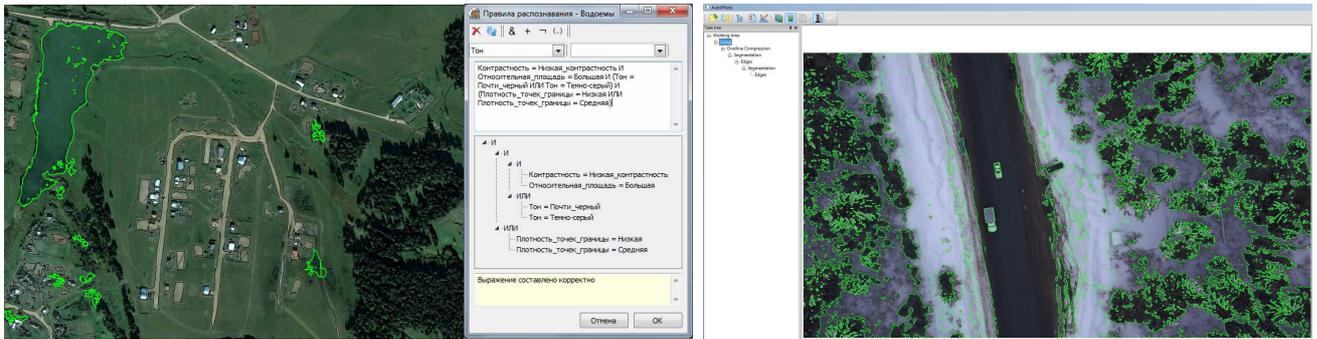


Рисунок А.5 - Выделение искусственных объектов на аэрокосмическом снимке
(Источник: Время съёмки: 2012 г., GoogleEarth; 2016 г., ООО "Финко")

На рисунке А.6 показан пример решения задачи координатной привязки снимков. Применен метод "лучевой граф", позволяющий производить привязку снимков разного масштаба, сделанных в разное время, выделять совпадения и отличия.

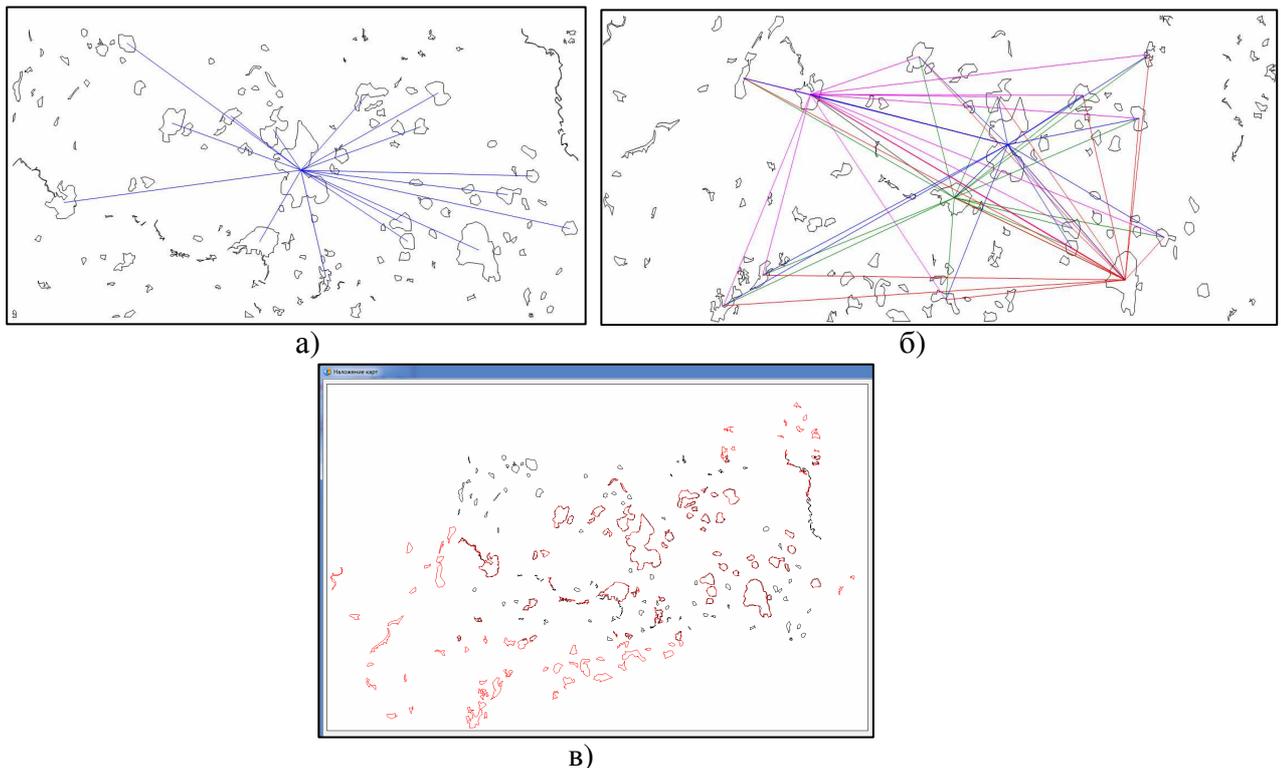


Рисунок А.6 – Координатная привязка: а) граф контуров снимка-запроса; б) граф контуров снимка из БД; в) автоматическое совмещение снимков

Оценка качества выделения значимых объектов на аэрокосмических снимках:

Таблица А.1

Сравнение систем дешифрирования

	eCognition (X.Sun, J.Zhang, Z.Liu, 2005)	eCognition (L.Huang, L.Ni, 2008)	eCognition (Md.D.Rahman, S.K.Saha, 2008)	eCognition (L.A.Arroyo, K.Johansen, S.Phinn, 2010)	Feature Analyst (N.D.Riggan, Jr., R.C.Weih, Jr., 2009)	Feature Analyst (M. Králová, 2013)	Extraction (M.C.Petrita, 2015, на основе продукционных)	ENVI Feature Extraction (M.C.Petrita, 2015, на основе эталонов)	TextSeg
Водоемы	Полнота 100% Точность 90,91%	Полнота 100% Точность 90%	Полнота 100% Точность 100%	Полнота 100% Точность 83,9%	Полнота 100% Точность 89,5%	Полнота 100% Точность 99%	Полнота 100% Точность 100%	Полнота 100% Точность 100%	Полнота 100% Точность 92,53%
Здания	Полнота 74,34 Точность 90%	Полнота 88,79% Точность 84,22%	Полнота 80,11% Точность 96,58%	Полнота 95,9% Точность 98,6%	Полнота 85,9% Точность 85,9%	Полнота - % Точность 96%	Полнота 99,99% Точность 100%	Полнота 92,63% Точность 89,47%	Полнота 87,08% Точность 82,06%
Дороги	Полнота 60% Точность 75%	Полнота 57,89% Точность 80,49%	Полнота - % Точность - %	Полнота - % Точность - %	Полнота 94,3% Точность 86,8%	Полнота 93% Точность 95%	Полнота 100% Точность 99,73%	Полнота 96,95% Точность 97,67%	Полнота 86,04% Точность 79,13%
Посадки и агрокультуры	Полнота - % Точность - %	Полнота - % Точность - %	Полнота 92,52% Точность 89,79%	Полнота 92,1% Точность 87,2%	Полнота - % Точность - %	Полнота 99% Точность 97%	Полнота - % Точность - %	Полнота - % Точность - %	Полнота 86,13% Точность 99,56%
Луга	Полнота - % Точность - %	Полнота 100% Точность 94,12%	Полнота 95,2% Точность 87,95%	Полнота 45,7% Точность 48,6%	Полнота 83,8% Точность 91,2%	Полнота 100% Точность 96%	Полнота 99,85% Точность 99,4%	Полнота 68,66% Точность 85,5%	Полнота 94,76% Точность 95,42
Леса	Полнота 100% Точность 68,18%	Полнота - % Точность - %	Полнота - % Точность - %	Полнота 65,3% Точность 98,3%	Полнота 76,23% Точность 78,6%	Полнота 96% Точность 99%	Полнота 94,01% Точность 100%	Полнота 82,98% Точность 96,21%	Полнота 93,21% Точность 97,12
Тени	Полнота 80% Точность 100%	Полнота - % Точность - %	Полнота - % Точность - %	Полнота - % Точность - %	Полнота - % Точность - %	Полнота - % Точность - %	Полнота 100% Точность 47,79%	Полнота 96,95% Точность 97,67%	Полнота 94,56% Точность 97,84%
Среднее значение	Полнота 82,67%	Полнота 86,67%	Полнота 91,96%	Полнота 79,8% Точность	Полнота 88,05%	Полнота 97,6% Точность	Полнота 98,98%	Полнота 89,7% Точность	Полнота 91,68%

	Точность 84,82%	Точность 87,21%	Точность 93,58%	83,32%	Точность 86,4%	точность 97%	Точность 91,15%	точность 94,42%	Точность 91,95%
--	--------------------	--------------------	--------------------	--------	-------------------	--------------	--------------------	--------------------	--------------------

Сопоставление результатов выделения и расчета признаков объектов в разработанной подсистеме дешифрирования TexSeg [57] и автоматизированных расчетов, выполненных в существующих зарубежных системах автоматизации дешифрирования, показало, что средняя по различным классам объектов полнота и точность сопоставимы, но за счет разработанной методологии система позволяет получать дополнительные характеристики формы, отыскивать и сопоставлять границы одноименных объектов с целью изучения динамики объектов и явлений по разновременным снимкам.

Задачи анализа и дешифрирования аэрокосмоснимков в существующих автоматизированных системах решаются на основе прямых дешифровочных признаков, извлекаемых из изображения (цветовые характеристики, границы областей), в то время как для качественной интерпретации объектов требуется комплексный анализ связной совокупности прямых и косвенных признаков, с последующими логическими заключениями.

Принципы и методы, примененные в разработанном комплексе анализа изображений, в частности, методы цветовой сегментации, скелетизации, многослойный атрибутивный граф изображения, метод сопоставления и формальный язык для автоматического анализа и описания изображений позволили повысить точность выделения значимых для предметной области объектов на аэрокосмических снимках в среднем до 91,95% и полноту до 91,68%.

ПРИЛОЖЕНИЕ Б

(справочное)

Анализ папиллярных узоров пальцевых отпечатков

Известны три основных типа папиллярного узора – дуговой, петлевой и завитковый, которые представлены на рисунке Б.1 [11].

Признаки объектов изображения, запрашиваемые у пользователя:

1. Величина *Пол человека* $\in \{male, female\}$

где: *male/female* – значение переменной *Пол Человека*, которые вводит пользователь.

2. Величина *Возраст* $\in \{1-14, 15-20, 20-45, от 45\}$

где: *1-14, 15-20, 20-45, от 45* – значение переменной *Возраст*, которые вводит пользователь.

Базовые отношения:

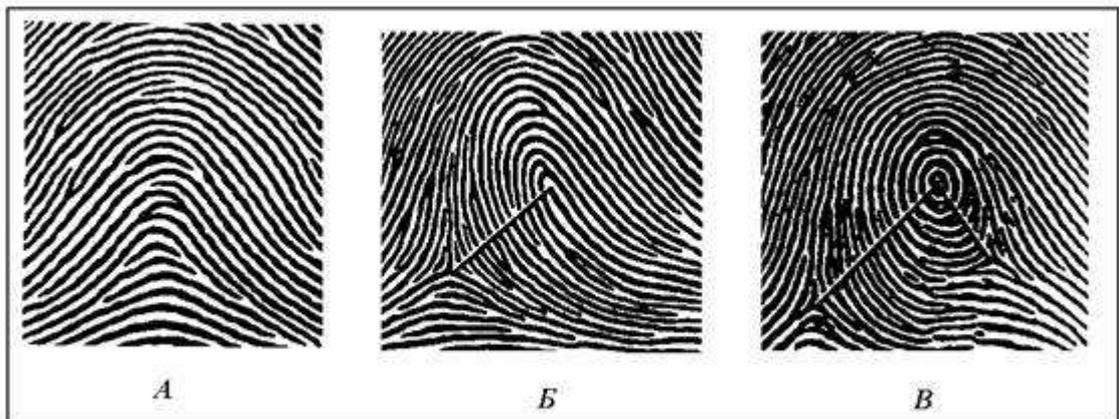


Рисунок Б.1 – Типы папиллярных узоров. А- дуговой узор, Б- петлевой узор, В- завитковый узор

R1: иметь *СоседаПоНачалу* (*a*, *b*) – связь начала элемента *a* началом или концом элемента *b*. Интерпретация:

$$\exists a \in ES, b \in ES, (a.xs = b.xs \wedge a.ys = y.xs) \vee (a.xs = b.xe \wedge a.ys = y.xe) \Rightarrow R1(a, b) = True$$

$R2$: *иметьСоседаПоКонцу* (a, b) – связь конца элемента a с началом или концом элемента b . Интерпритация:

$$\exists a \in ES, b \in ES, (a.xe = b.xs \wedge a.ye = y.xs) \vee (a.xe = b.xe \wedge a.ye = y.xe) \Rightarrow R1(a, b) = True$$

Производные понятия (терминологические аксиомы):

(1) *КонецВетви* – точка начала или конца элемента, не совпадающая с началом или концом какого либо другого элемента. Семантика:

$$LineEnd(a) \mid \neg R1(a, b) \vee \neg R2(a, b), a \in ES, b \in ES$$

(2) *Узел* – точка соединения трех элементов. Семантика:

$$Node(a, b, c) \mid R1(a, b) \wedge R1(a, c) \vee R2(a, b) \wedge R2(a, c), a \in ES, b \in ES, c \in ES$$

(3) *ОсобаяТочка*: $ParticularPoint \equiv LineEnd \sqcup Node$;

$$PP \equiv ParticularPoint$$

Далее введем рекурсивное определение Последовательности соседних элементов:

$$Sequence ::= a \mid a \sqcup b \ R2(a, b) \mid Sequence \sqcup c \ R2(Sequence, c)$$

Тогда, *Цепочка* – это последовательность, которая начинается и заканчивается особой точкой:

$$(4) Chain \sqsubseteq \forall Sequence (=2 R3 PP), где$$

$R3$: *hasPP*($Chain, PP$) – отношение *ИметьОсобуюТочку*,

т.е. такие последовательности, которые имеют ровно две особые точки.

Кривизна (curvature) цепочки:

$$CC = \frac{1}{n} \sum_{i=1}^n da_i$$

где: da_i – приращение угла наклона i -того и $i+1$ -го элементов;

n – количество элементов в цепочке.

Хорда цепочки:

$h ::= \langle xs_1, ys_1, xe_n, ye_n \rangle$, где N – количество элементов цепочки.

$R4$: *hasCC*($chain, CC$) – отношение *ИметьКривизну* в множестве CC .

Завиток – замкнутая последовательность элементов, в которой последний элемент соединяется с первым и отсутствуют особые точки:

$$Whorl \sqsubseteq \forall Sequence, R2(a_n, a_1), \neg R3(PP),$$

где N – количество элементов последовательности.

Петля – элемент скелетона, имеющий форму дуги и радиус, равный (близкий к) минимальному на всем множестве дуговых элементов:

$$Loop \sqsubseteq \forall ES, Form=Arc, r = MinR.$$

Расстояние *Space* между двумя цепочками (хорды которых имеют близкие углы наклона) – это свойство, которое принимает значение функции, вычисляющей среднее значение между парами ближайших точек этих цепочек.

Функция интерпретации:

$$Space = 0;$$

Цикл $i=1..N$ //по элементам Chain1

$$Distance = 1000;$$

Цикл $j=1..M$ //по элементам Chain2

$$l1 = Sqrt((xs_j - xs_i)^2 + (ys_j - ys_i)^2);$$

$$l2 = Sqrt((xe_j - xs_i)^2 + (ye_j - ys_i)^2);$$

$$l3 = Sqrt((xm_j - xs_i)^2 + (ym_j - ys_i)^2);$$

$$MinL = \min(l1, l2, l3);$$

$$Distance = \min(Distance, MinL)$$

$$Space = Space + Distance;$$

$$Space = Space/M \quad //при \quad M < N$$

Поток (Strim) – это семейство ближайших цепочек (хорды которых имеют близкие углы наклона) таких, расстояние *Space* между которыми в поперечнике минимально. Поиск линий, принадлежащих одному потоку, осуществляется рекурсивно.

Дельта или трирадиус – это место, где сходятся три различно направленные папиллярные линии [11]. С учетом определения особой точки (ОТ), определение дельты может выглядеть следующим образом.

Дельта – это ОТ, в которой углы между соединяющимися элементами близки к 120° :

$$ThreeRadius \sqsubseteq \forall Node(a, b, c), R5(a, b), R5(b, c), R5(a, c),$$

где отношение

R7: haveBluntAngle(a, b) – иметь тупой угол равно *True*, если функция *FA(a, b)*, вычисляющая нечеткое значение [*Острый, Прямой, Тупой*] угла между элементами *a, b*, принимает значение *Тупой*.

R8: иметьДельта – наличие или отсутствие дельт на папиллярном узоре;

R9: иметьГребневойСчет – имеется свойство, принимающее значение гребневого счета.

$$ПапиллярныйУзор \equiv иметьВозраст \sqcup иметьПол \sqcup иметьЗавиток \sqcup \\ иметьПетля \sqcup иметьДельта \sqcup иметьГребневойСчет$$

Примеры определения некоторых психологических характеристик: Сложный характер поведения (высокая мозговая активность), если узор содержит более 5 дельт [73]:

$$Сложный\ характер \sqsubseteq ПапиллярныйУзор(>5\ R6\ Дельта).$$

Коммуникабельный человек, хорошо работает в команде, если гребневой счет меньше 2:

$$Коммуникабельный \sqsubseteq ПапиллярныйУзор(<2\ R7\ Цепочка).$$

Если гребневой счет больше 2, то человек замкнутый:

$$Замкнутый \sqsubseteq ПапиллярныйУзор(>2\ R7\ Цепочка).$$

Таким образом, предлагаемый формальный язык на основе дескрипционной логики с расширением на графы для автоматического построения формализованных описаний объектов, содержащихся на изображении, характерен наличием множества интерпретирующих функций, анализирующих геометрическую форму и относительное расположение структурных элементов. Благодаря формальному языку изображение, по сути, становится источником данных, из которых получается не только статистическое, но и структурное описание. Подобным образом формальный язык может применяться и для

автоматизации определения различных значений признака "Форма лица", употребляемого при создании словесного портрета [89, 20, 39].

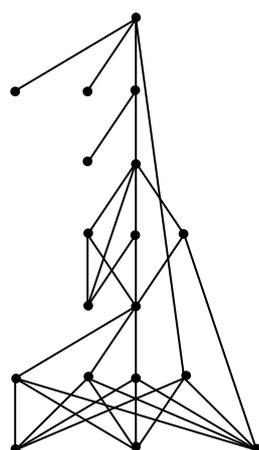
Пример алгоритма анализа изображений, основанный на системе определений онтологической базы знаний. Очевидно, процесс анализа изображений в некоторой предметной области строится на основе системы понятий, заданной экспертом. Чем сложнее структуры анализируемых объектов, тем сложнее алгоритм и ниже его быстродействие.

Алгоритм *анализа иерархических структур изображения*, заданных формальными описаниями элементов и отношений между ними, заключается в упорядоченной проверке истинности функций интерпретации, заданных терминологией T , начиная с младшего уровня и кончая верхним уровнем ациклического орграфа системы понятий.

Утверждение 1. Дерево понятий, заданное экспертом, определяет эвристическую стратегию, которая позволяет избавиться от необходимости полного перебора в процессе поиска за счет перехода от анализа отрезков (атомарных концептов) к анализу "парных" структур, а затем, на их основе и к более сложным структурам, построенным из них, и т.д.

Построим ориентированный граф $\Gamma(T)$ зависимостей концептов для системы определений из выше приведенного примера. Вершинами этого графа будут атомарные концепты терминологии T и из вершины A в вершину B идет дуга, если есть концепт C , эквивалентный концепту A , и концепт B входит в концепт C . При этом концепты $A \equiv C$, если в интерпретации $A^x = C^x$. В графе $\Gamma(T)$ из каждой вершины исходит только одна дуга. Базовые концепты не имеют входящих дуг. Производные концепты имеют одну исходящую дугу.

Ациклический орграф $\Gamma(T)$ зависимостей понятий показан на рисунке Б.2.



Папиллярный узор

Пол, Возраст, Гребневой счет

Расстояние, Поток

Дуга, Завиток, Петля

Кривизна, Цепочка

Пара, Конец, Узел, Дельта

Отрезок, Сосед по Началу, Сосед по Концу

Рисунок Б.2 - Граф системы понятий "Пальцевые отпечатки"

Соответствующий ему алгоритм выглядит следующим образом:

1. Выявление отношений соседства и пар связанных элементов (полный перебор).
2. Поиск объектов типа *Конец ветви, Узел, Дельта* (перебор пар).
3. Выделение цепочек и вычисление кривизны (перебор пар).
4. Выделение объектов типа *Дуга, Петля, Завиток* (перебор цепочек).
5. Выделение потоков (перебор однотипных цепочек: сортировка по типу, анализ совместимости цепочек).

Таким образом, алгоритм анализа заключается в упорядоченной проверке истинности функций интерпретации, заданных терминологией T .

По сравнению с NP полным перебором, на каждом из более высоких уровней орграфа $\Gamma(T)$ количество экземпляров, подвергающихся проверке, сокращается.

Утверждение 2. Для обеспечения эффективности анализа онтологическое дерево класса запросов должно быть:

1) *логически корректным* – конъюнкция формул-признаков на любом пути из корня к листу должна быть логически непротиворечивой, иначе этот лист или все поддерево – просто лишние.

2) *однозначным*: для любой вершины формулы-признаки дочерних вершин должны быть взаимоисключающими, чтобы процесс поиска ответа не содержал ветвлений.

Утверждение 3. При выполнении условия 2 утверждения 2 временная сложность вычисления «запроса» имеет порядок $O(n^k * \log_2 N)$, где n – количество элементов в модели, N – количество *ALC* определений в онтологии, k – (максимальное) количество кванторов в определениях классов.

Такое представление запроса также удобно для модификации. Например, заменяем в какой-нибудь вершине формулу на более частную или общую и получаем, соответственно, более частный или общий вариант понятия-ответа.

Таким образом, стратегия анализа изображений в разделе данных базы знаний определяется графом $\Gamma(T)$ терминологии. Алгоритм анализа есть доказательство истинности высказываний (предикатов) о семантике графической информации. В то же время на примерах 4.3., 4.4. из геометрии видно, что одну и ту же сущность можно определить разными путями. В результате длительность поиска будет отличаться.

ПРИЛОЖЕНИЕ В**(справочное)****РЕСО - программная система для распознавания старославянских текстов**

При создании электронных коллекций старославянских текстов (рукописи 15 – 18 веков) серьезной проблемой остаются задачи обработки и распознавания изображений. В условиях, когда оригиналы документов сильно изношены, цвет бумаги неоднороден, имеется множество потертостей, затемнений, пятен, более поздних пометок, предъявляются высокие требования к программам обработки изображений и распознавания.

Описываемая ниже программная система Ресо – экспериментальная версия, особенностями ее являются:

- 1) использование структурного (лингвистического) подхода к распознаванию букв;
- 2) развитый режим предобработки, редактирования и пополнения эталонов;
- 3) применение корреляционного сопоставления при возникновении коллизий в результате структурного метода.

Программа содержит стандартный набор инструментов для повышения качества изображения: сглаживания, фильтрации, контрастирования, многоступенчатой статистической бинаризации.

Этап обработки – выделение скелетона (рисунок В.1).

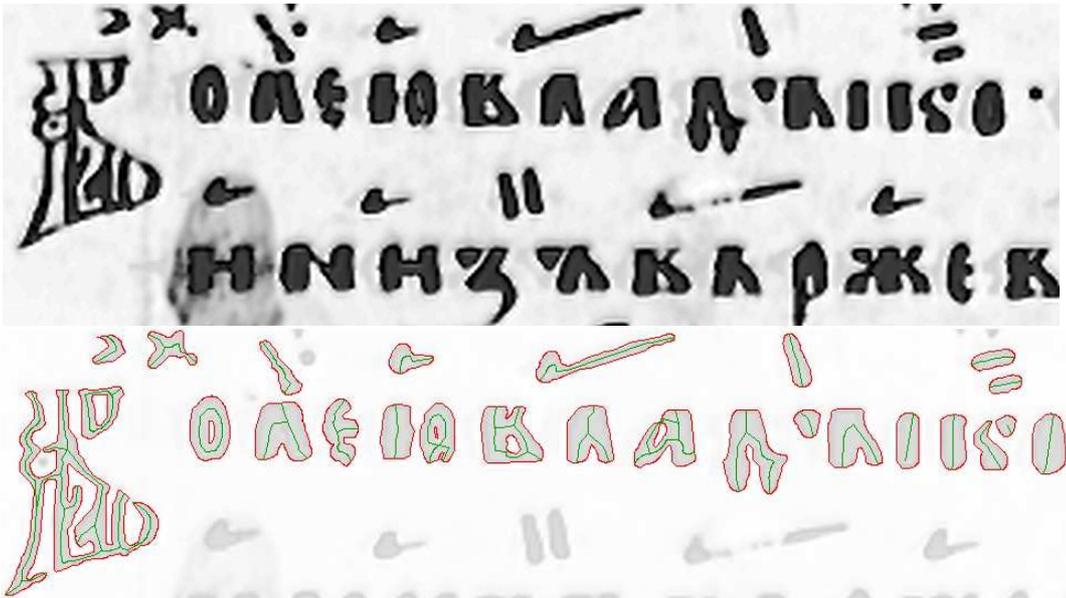


Рис. В.1. Двухступенчатая бинаризация и скелетон

Предобработка скелетного изображения. В предобработку помимо аппроксимирования входного скелетного изображения включены операции построения его графа и определения качественных параметров объектов изображения (рисунок В.2).

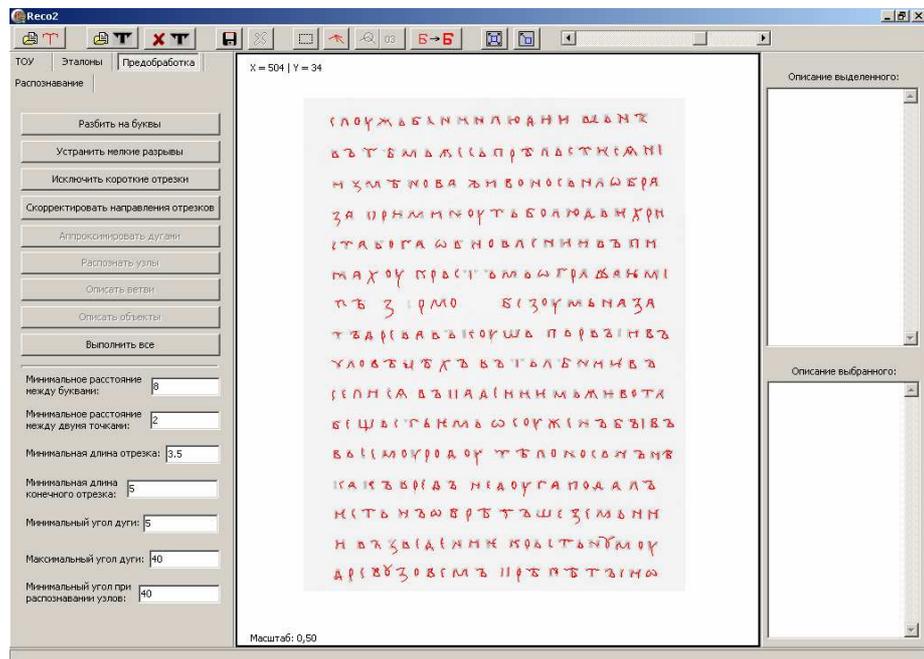


Рис. В.2. Режим предобработки

В системе Reso дополнительно был реализован в упрощенном виде метод корреляционного сопоставления растрового изображения буквы с растровыми изображениями потенциальных эталонов, выдаваемых структурным методом.

Алгоритм корреляционного сопоставления:

1) Масштабируем растровое изображение эталона под размер анализируемой буквы с сохранением исходных пропорций. Осуществлять поворот эталона при работе со старославянскими текстами не требовалось, поскольку буквы написаны без наклона.

2) Помещаем отмасштабированное растровое изображение эталона в центр анализируемой буквы.

3) Вычисляем коэффициент корреляции изображений относительно центральной точки буквы по формуле:

$$k(x, y) = \frac{\sum_s \sum_t [f(x + s, y + t) - \bar{f}][w(s, t) - \bar{w}]}{\sqrt{\sum_s \sum_t [f(x + s, y + t) - \bar{f}]^2 \sum_s \sum_t [w(s, t) - \bar{w}]^2}},$$

где $f(x+s, y+t)$ – функция, возвращающая яркость пикселя с координатами $(x+s, y+t)$, принадлежащего растровому изображению буквы;

$w(s, t)$ – функция, возвращающая яркость пикселя с координатами (s, t) , принадлежащего растровому изображению эталона;

\bar{f} – среднее значение элементов изображения f в области, пересекающейся с изображением эталона w ;

\bar{w} – среднее значение пикселей в эталоне w .

4) Итог оформляем с учетом разницы площади эталона и площади буквы (для того, чтобы учесть, например, что буква «у» вытянута по вертикали).

Примечание: значение коэффициента корреляции лежит в промежутке $[-1, 1]$, где -1 означает, что буква и эталон совпадают на 100%, но представлены в противоположных цветах, а 1 – что, буква и эталон совпадают на 100% в обычном смысле.

Получив значение корреляции по каждому потенциальному эталону, букве ставим в соответствие эталон с наибольшим значением корреляции.

Работа с эталонами. В программе имеется возможность просматривать, редактировать, добавлять и удалять эталоны.

Для добавления эталона необходимо выполнить предобработку скелетного изображения, выбрать букву и нажать на кнопку «Сохранить как эталон ->».

Появится окно ввода имени эталона (рисунок В.3). В данном окне необходимо выбрать шрифт, в котором будет выводиться имя. Имя вводится в соответствующем поле или выбирается из таблицы символов выбранного шрифта. После ввода имени необходимо нажать на кнопку «Ок». Программа добавит новый эталон в конец списка. Если под буквой находилась подложка, то к эталону приложится и его исходное изображение.

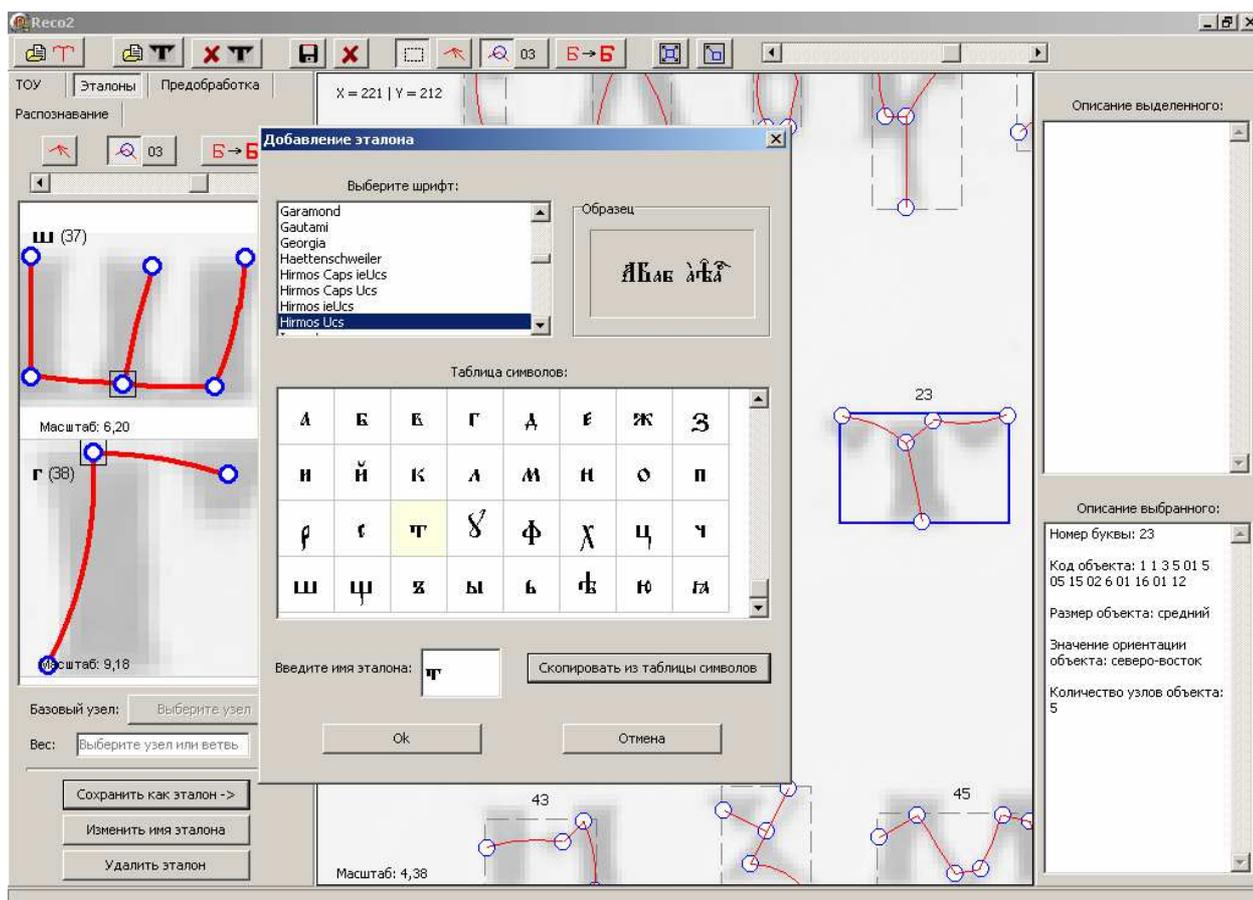


Рис. В.3. Добавление эталона

Имеется возможность назначать вес для узлов и ветвей – степень их важности, учитываемая при составлении статистики совпадения при распознавании. Чтобы назначить вес, необходимо выбрать узел или ветвь, ввести значение в поле «Вес» и снять выделение с элемента. Значения весов могут быть произвольными, важно лишь их взаимное различие. Эталоны отображаются с учетом веса узлов и ветвей: менее весомые элементы отображаются тоньше и светлее (рисунок В.4).

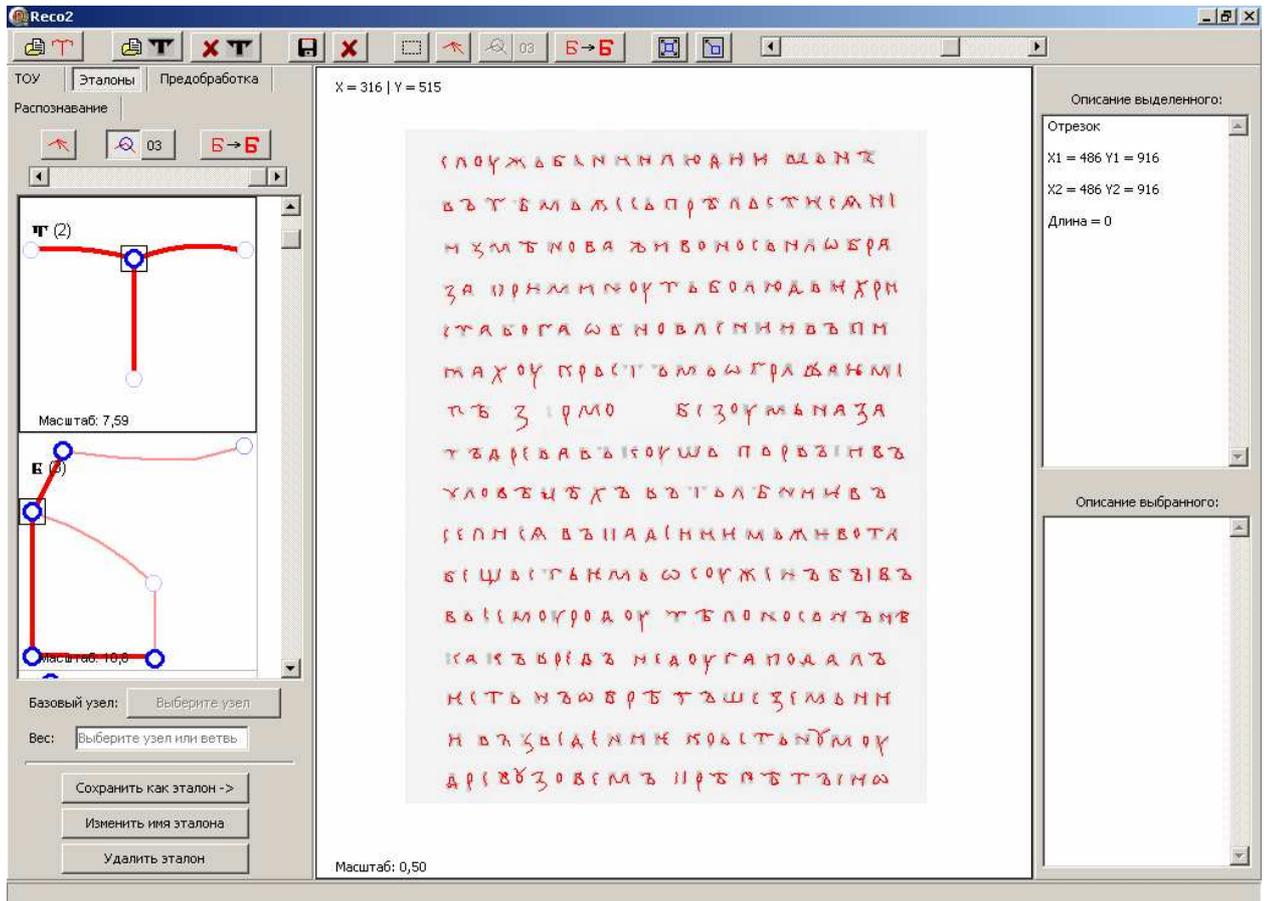
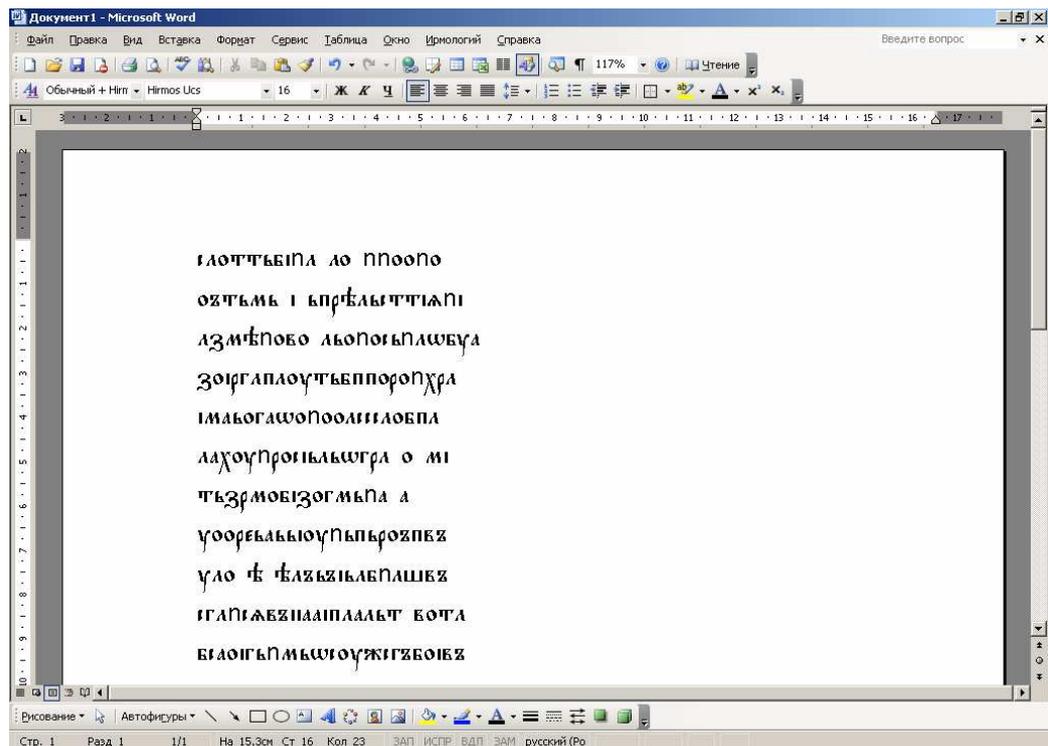


Рис. В.4. Визуализация эталонов с учетом веса узлов и ветвей

Распознавание. Результат распознавания передается в текстовый редактор Microsoft Word (рисунок В.2), (использованы шрифты с сайта: <http://www.irmologion.ru/>).



На данный момент программа Ресо дает 86% распознанных символов на рукописях хорошего качества и 64% на рукописях низкого качества.

Таким образом, можно сделать вывод о том, что в тех случаях, когда структурный метод распознавания выдает для какой-то буквы несколько удовлетворительных кандидатов, применением корреляционного метода для разрешения коллизий можно улучшить результат примерно на 10% ценой увеличения времени обработки. Кроме того, существенное повышение надежности дает адаптивная побуквенная бинаризация изображения [32, 51].

ПРИЛОЖЕНИЕ Г

(справочное)

Акты внедрения (использования) результатов диссертационной работы



АКТ
о внедрении (использовании) результатов
диссертационной работы
Кучуганова Александра Валерьевича

Комиссия в составе: председатель – Кутергин В.А., члены комиссии: Бекмансуров В.А., Коробейников А.С., Столов А.А. составила настоящий акт о том, что результаты диссертационной работы *Кучуганова А.В.* использованы в работе ООО «Нордика Стерлинг», при реализации проектов по техническому перевооружению предприятий.

в следующем виде:

- системы поиска графической информации в хранилищах данных;
- подсистемы автоматизированной векторизации графической информации;
- информационной технологии системы автоматизированной обработки графической информации;
- модуля анализа и интерпретации изображений, формирующего краткую характеристику (описание изображения) с целью организации семантического поиска по словесному портрету;
- модуля поиска изображений по словесному портрету предназначенного для интерпретации с помощью БЗ заданного пользователем словесного портрета и поиска графической информации, соответствующей интерпретированному словесному портрету.

Использование указанных результатов позволяет:

- осуществлять графический поиск, схем, планов и чертежей в архивах, опираясь на методы автоматической векторизации чертежей, формирования и распознавания графических образов:
 - повысить эффективность и качество поиска информации за счет высокоуровневого анализа графической информации;
 - сократить сроки проектирования новых изделий.

Председатель комиссии:

Члены комиссии:



(Signature) (Кутергин В.А.)

(Signature) (Бекмансуров В.А.)

(Signature) (Коробейников А.С.)

(Signature) (Столов А.А.)



АКТ

о внедрении (использовании) результатов
диссертационной работы
Кучуганова Александра Валерьевича

Комиссия в составе: председатель – Яркеев Геннадий Трофимович, заместитель
начальника управления 090,

члены комиссии:

Кубатко Данил Викторович, главный инженер ОКБ,

Бузанов Алексей Валентинович, заместитель главного технолога по подготовке
производства,

Локинский Николай Сергеевич, главный специалист по ИТ

составила настоящий акт о том, что результаты диссертационной работы *Кучуганова А.В.*
использованы на ОАО «Ижевский электромеханический завод «Купол»

в следующем виде:

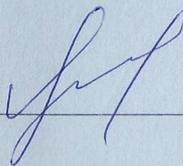
- системы поиска графической информации в хранилищах данных;
- подсистемы автоматической векторизации растровых чертежей и эскизов;
- информационной технологии системы автоматизированной обработки графической информации.

Использование указанных результатов позволяет:

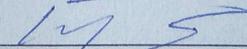
- осуществлять графический поиск схем, планов и чертежей в архивах, опираясь на методы автоматической векторизации чертежей, формирования и распознавания графических образов:

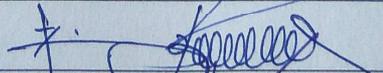
- повысить эффективность и качество поиска информации за счет высокоуровневого анализа графической информации.

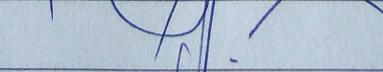
Председатель комиссии:

 (Г.Т. Яркеев)

Члены комиссии:

 (Д.В. Кубатко)

 (А.В. Бузанов)

 (Н.С. Локинский)

УТВЕРЖДАЮ

Проректор по научно-
организационной работе ГОУ
ВПО «Ижевский государственный
технический университет»

В.А. Алексеев

« _____ » _____ 2010 г.



АКТ

о внедрении (использовании) результатов
докторской диссертации
Кучуганова Александра Валерьевича

Комиссия в составе: председатель – начальник управления научно-исследовательских работ ИжГТУ, д.т.н., Коршунов А.И., члены комиссии: начальник отдела патентно-информационных исследований и интеллектуальной собственности ИжГТУ Васильева О.В., директор АНО «Региональный центр наноиндустрии Удмуртской Республики», д.х.н., профессор Плетнёв М.А., к.т.н., доцент Кузнецов А.П.

составила настоящий акт о том, что результаты диссертационной работы *Кучуганова А.В.* использованы для разработки автоматизированной системы патентно-лицензионного поиска в области нанотехнологий по Государственному контракту № 01.647.12.3009 в рамках ФЦП «Развитие инфраструктуры наноиндустрии в Российской Федерации на 2008-2010 годы» в следующем виде:

- модуль графического поиска производит поиск по содержащейся в текстах охранных документов графической информации путем распознавания изображений и чертежей;
- модуль анализа и интерпретации изображений формирует и записывает в БД краткую характеристику (описание изображения) с целью организации семантического поиска по словесному портрету;
- модуль поиска изображений по словесному портрету предназначен для интерпретации с помощью БЗ заданного пользователем словесного портрета и поиска охранных документов, которые содержат графическую информацию, соответствующую интерпретированному словесному портрету.

Использование указанных результатов позволяет:

- повысить эффективность и качество поиска информации путем анализа графической информации, содержащейся в текстах охранных документов.

Председатель комиссии:

_____ (А.И. Коршунов)

Члены комиссии:

_____ (О.В. Васильева)

_____ (М.А. Плетнев)

_____ (А.П. Кузнецов)

УТВЕРЖДАЮ

Первый проректор по стратегическому
направлению и научно-инновационной
работе ФГАОУ ВПО "Северо-
Восточный федеральный университет
им. М. К. Аммосова"

В.И. Васильев

« » 20 г.

АКТ

о внедрении (использовании) результатов
диссертационной работы

Кучуганова Александра Валерьевича

Комиссия в составе: председатель – Николаев Анатолий Николаевич д.б.н., декан биолого-географического факультета, члены комиссии: Слепцова Надежда Петровна – к.б.н., доцент зав. кафедрой географии, Саввинова Антонина Николаевна к.г.н., доцент кафедры географии, Данилов Юрий Георгиевич – к.г.н., доцент кафедры географии составила настоящий акт о том, что результаты диссертационной работы *Кучуганова А.В.* использованы в рамках госбюджетной темы №4043 Госзаказ МОиН на 2012 год по теме: «Разработка и экспериментальное исследование системы аэрокосмического и геотинформационного мониторинга для визуализации результатов геоэкологических исследований северных экосистем»

в следующем виде:

1. Методологии построения системы обработки аэрокосмических снимков.
2. Информационная технология подсистемы автоматизированной обработки аэрокосмических снимков.
3. Программа подсистемы автоматизированной обработки аэрокосмических снимков.

Использование указанных результатов позволяет:

- повысить эффективность, быстродействие и качество анализа графической информации на примере аэрокосмических снимков;
- повышение быстродействия, надежности и степени универсальности методов и инструментальных средств обработки, анализа и распознавания изображений в том числе в учебном процессе;
- автоматизировать обработку аэрокосмических снимков;
- повысить наглядность, удобство принятия решений, осуществляемых на основе цветовой сегментации. Цветовая сегментация позволяет увидеть на экране не только количественные, но и качественные значения, получить границы и площадь зоны с заданным уровнем. Качественные значения используются также в экспертном модуле системы поддержки принятия решений для нечеткого логического вывода и синтеза плана мероприятий.

Председатель комиссии:

 (Николаев А.Н.)

Члены комиссии:

 (Слепцова Н.П.)

 (Саввинова А.Н.)

 (Данилов Ю.Г.)

УТВЕРЖДАЮ

Проректор по учебной работе ФГБОУ
ВПО "Ижевский государственный тех-
нический университет имени М. Т. Ка-
лашников"

В.В. Хворенков
2013 г.

АКТ

о внедрении (использовании) результатов
диссертационной работы
Кучуганова Александра Валерьевича

Комиссия в составе: председатель – декан факультета ИВТ, д.т.н., д.э.н., д.г.-м.н. профессор Лялин В.Е., члены комиссии:
зав. кафедрой ВТ д.т.н., профессор Куликов В.А., к.т.н., доцент Егоров С.Ф., ст. преподаватель Коробейников А.А.

составила настоящий акт о том, что результаты диссертационной работы *Кучуганова А.В.* использованы *в учебном процессе на кафедрах «Вычислительная техника» и «Автоматизированные системы обработки информации и управления» Ижевского государственного технического университета имени М.Т. Калашникова при выполнении лабораторных работ по дисциплинам "Инженерная и компьютерная графика", "Растровая и векторная графика", "Геоинформационные системы и технологии" направления "Информатика и вычислительная техника"*

в следующем виде:

1. Программного модуля ArtPhoto цветовой сегментации и векторизации изображений.
2. Методических указаний к выполнению лабораторных работ по цифровой обработке изображений.

Использование указанных результатов позволяет:

- повысить быстродействие, качество обработки, анализа и распознавания изображений в учебном процессе;
- повысить степень автоматизации труда при создании геоинформационных систем на основе аэрокосмических снимков.

Председатель комиссии:

Члены комиссии:

(В.Е. Лялин)

(В.А. Куликов)

(С.Ф. Егоров)

(А.А. Коробейников)