

Иркутский государственный технический университет

На правах рукописи  
УДК xxx.xxxx

КАТАШЕВЦЕВ МИХАИЛ ДМИТРИЕВИЧ

Математические модели контурных изображений и алгоритмическая  
сложность их анализа

Специальность 05.13.18 —  
«Математическое моделирование, численные методы и комплексы программ»

Диссертация на соискание учёной степени  
кандидата физико-математических наук

Научный руководитель:  
д. ф-м. н., профессор  
Мартьянов В.И.

Иркутск – 2014

# Содержание

|   |    |
|---|----|
| Введение . . . . .  | 3  |
| 1 Обзор . . . . .   | 6  |
| 1.1 Распознавание текста . . . . .  | 7  |
| 1.2 Персептрон как метод распознавания образов . . . . .  | 8  |
| 1.3 Поисковые системы . . . . .   | 9  |
| 2 Распознавание изображений . . . . .   | 10 |
| 2.1 Базовые понятия . . . . .   | 10 |
| 2.2 Модель контурного изображения . . . . .   | 12 |
| 2.3 Преобразование растрового изображения . . . . .   | 13 |
| 2.3.1 Волновая скелетизация . . . . .   | 13 |
| 2.3.2 Построение модели $\mathcal{M}$ для графа $G$ . . . . .   | 15 |
| 2.4 Постановка задачи распознавания . . . . .   | 17 |
| 2.5 Интерпретация . . . . .   | 18 |
| 3 Применение . . . . .  | 22 |
| 3.1 Распознавание символов . . . . .  | 22 |
| 3.2 Оценка устойчивости битумных эмульсий . . . . .   | 22 |
| 3.2.1 Введение . . . . .  | 22 |
| 3.2.2 Анализ эмульсий . . . . .   | 23 |
| 3.2.3 Оценка качества анализа . . . . .   | 24 |
| 3.2.4 Определение среднего размера и дисперсии частиц битумной эмуль-<br>сии на модифицированном битуме . . . . . | 25 |
| 3.2.5 Результаты . . . . .  | 26 |
| 3.3 Автоматизация составления ПОДД . . . . .  | 27 |
| Заключение . . . . .  | 28 |
| Список рисунков . . . . .   | 29 |
| Список таблиц . . . . .   | 30 |
| Литература . . . . .  | 31 |
| А Структура БД . . . . .  | 32 |

# Введение

Важнейшим направлением развития современных информационных технологий (и в целом, прогресса, так как невозможно представить современную науку и технику без использования компьютеров) является создание математических методов быстрого решения информационных задач. В идеале, эти методы должны обеспечивать скорость решения ряда важных информационных задач вне зависимости от объема данных. И действительно, есть ряд важных информационных задач, где это возможно.

Наглядным (и очень важным) примером этого являются реляционные базы данных (БД), где вычислимость запросов определенных типов не зависит от объема данных, а только линейно от сложности проекта самой БД. Практическим подтверждением этого для обывателя является скорость работы банковских систем, использующих сетевые реляционные БД, с их мировыми сетями терминалов и банкоматов, где можно проводить операции с вкладами и денежными средствами в любой точке мира за считанные секунды.

Отметим, что весьма близко к этому классу примыкают задачи поиска данных по ключевым словам в Интернет - пространстве (полно текстовые поисковые системы «Google», «Яндекс» и др.), где скорость поиска не замедляется из-за экспоненциального роста информации в глобальной сети.

Другим важным классом информационных задач являются вопросы распознавания образцов (образов), где при организации данных близкой к таблицам реляционных БД также могут быть получены результаты независимости скорости распознавания образцов от их количества, вернее, верхней границы сложности распознавания одного образца с добавкой только количества образцов.

Данное направление исследований весьма актуально для современной математики, науки в целом и техники, как в теоретическом, так и практическом плане, где работают крупные транснациональные корпорации, реализуются технологии «Big Table», «Big Data» и предполагается получение прорывных результатов в робототехнике, молекулярной биологии, системах искусственного интеллекта и других важных областях для прогресса современного общества.

Целью данной работы является построение математических моделей контурных изображений, которые, хотя и имеют более сложную организацию данных, чем таблицы реляционных БД, но позволяют получить почти аналогичные результаты по алгоритмической сложности анализа контурных изображений, включая проверку изоморфных вложений образцов в анализируемое изображение. Для достижения поставленной цели необходимо было решить следующие задачи:

1. Разработать преобразование раstra контурного изображения в нагруженный граф специального вида.
2. Разработать преобразование нагруженных графов специального вида в математические модели, представленные многоосновными алгебраическими системами, где контурные изображения сведены к ориентированным дугам, связям дуг и их численным характеристикам в градусном измерении и относительных размеров длины дуг.

3. Исследовать алгоритмическую сложность анализа контурных изображений, включая проверку изоморфных вложений образцов в анализируемое изображение, где контурные изображения сведены к ориентированным дугам, связям дуг и их численным характеристикам в градусном измерении и относительных размеров длины дуг.
4. Разработать масштабные ряды контурных изображений и процедуры сжатия на основе относительных размеров дуг.
5. Исследовать возможность использования изоморфного вложения сжатого образца в сжатое изображение для уменьшения алгоритмической сложности построения изоморфного вложения исходного образца в исходное изображение.
6. Исследовать возможность использования полученных математических методов, математических моделей представления данных, алгоритмов и комплексов программ для решения прикладных задач:
  - (а) распознавания символов
  - (b) оценки устойчивости битумных эмульсий

#### Основные положения, выносимые на защиту

1. Сходящийся алгоритм волновой скелетизации, обеспечивающий преобразование растра контурного изображения в нагруженный граф специального вида (Утверждение 2.3.1.).
2. Оценка нижней границы алгоритмической сложности анализа контурных изображений, включая проверку изоморфных вложений образцов в анализируемое изображение, где контурные изображения сведены к ориентированным дугам, связям дуг и их численным характеристикам в градусном измерении и относительным размерам длины дуг (Теорема 2.5.1.).
3. Необходимое и достаточное условие продолжения изоморфного вложения сжатого образца в сжатое изображение до изоморфного вложения исходного образца в исходное изображение (Теорема 1.).
4. Верхняя граница алгоритмической сложности построения продолжения изоморфного вложения сжатого образца в сжатое изображение до изоморфного вложения исходного образца в исходное изображение (Теорема 2.).
5. Уменьшение верхней границы алгоритмической сложности построения изоморфного вложения исходного образца в исходное изображение при использовании изоморфного вложения сжатого образца в сжатое изображение (Теорема 3.).
6. Комплексы программ решения прикладных задач:
  - (а) распознавания символов
  - (b) оценки устойчивости битумных эмульсий

## Научная новизна

1. Впервые построены математические модели контурных изображений, представленных ориентированными дугами, связям дуг и их численными характеристиками в градусном измерении, а также относительными размерами длин дуг.
2. Впервые показано, что математические модели контурных изображений, хотя и имеют более сложную организацию данных, чем таблицы реляционных БД, но позволяют получить почти аналогичные результаты по алгоритмической сложности анализа контурных изображений, включая проверку изоморфных вложений образцов в анализируемое изображение.
3. Было выполнено оригинальное исследование использования масштабных рядов контурных изображений и процедуры сжатия на основе относительных размеров дуг, которые можно использовать для повышения эффективности анализа исходных изображений.

## Научная и практическая значимость

определяется во-первых, актуальностью направления исследований, которые обеспечивают применение информационных технологий буквально во всех сферах современной деятельности общества;

Во-вторых, применением разработанных программных комплексов для решения прикладных задач, что подтверждено официальными справками о применении результатов диссертации. Степень достоверности полученных результатов обеспечивается строгими математическими формулировками определений, а также строгими математическими доказательствами полученных утверждений, лемм и теорем.

Результаты находятся в соответствии с результатами, полученными другими авторами: А.И.Мальцевым, Ю.Л.Ершовым, С.В.Яблонским, А.И.Кокориным, А.В.Манциводой, Коддом, Д.Кнутом, В.И.Мартьяновым, Д.В.Пахомовым, В.В.Архиповым и др.

Объем и структура работы. Диссертация состоит из введения, трех глав, заключения и двух приложений. Полный объем диссертации составляет XXX страница с XX рисунками и XX таблицами. Список литературы содержит XXX наименований.

# Глава 1

## Обзор

Теория распознавания образа — раздел информатики и смежных дисциплин, развивающий основы и методы классификации и идентификации предметов, явлений, процессов, сигналов, ситуаций и т. п. объектов, которые характеризуются конечным набором некоторых свойств и признаков. Такие задачи решаются довольно часто, например, при переходе или проезде улицы по сигналам светофора. Распознавание цвета загоревшейся лампы светофора и знание правил дорожного движения позволяет принять правильное решение о том, можно или нельзя переходить улицу.

Необходимость в таком распознавании возникает в самых разных областях — от военного дела и систем безопасности до оцифровки аналоговых сигналов.

Проблема распознавания образа приобрела выдающееся значение в условиях информационных перегрузок, когда человек не справляется с линейно-последовательным пониманием поступающих к нему сообщений и в результате его голова переключается на режим одновременности восприятия и мышления, которому такое распознавание свойственно.

Неслучайно, таким образом, проблема распознавания образа оказалась в поле междисциплинарных исследований - в том числе в связи с работой по созданию искусственного интеллекта, а создание технических систем распознавания образа привлекает к себе всё большее внимание.

Распознавание образов — это отнесение исходных данных к определенному классу с помощью выделения существенных признаков, характеризующих эти данные, из общей массы несущественных данных.

Классическая постановка задачи распознавания образов: близка к постановке задачи классификации и предполагает наличие некоторого базового набора заранее известных классов (набор может расширяться), и образа (не обязательно визуального) которые надо классифицировать.

Наиболее часто в задачах распознавания визуальных образов рассматриваются монохромные изображения, что дает возможность рассматривать изображение как функцию на плоскости.

Множество же всех возможных функций  $f(x, y)$  на плоскости  $T$  — есть модель множества всех изображений  $X$ . Вводя понятие сходства между образами можно поставить задачу распознавания. Конкретный вид такой постановки сильно зависит от последующих этапов при распознавании в соответствии с тем или иным подходом.

В данной работе распознавание рассматривается в контексте задачи классификации. Можно выделить следующие классы алгоритмов для решения задачи классификации объектов.

## 1.1 Распознавание текста

Оптическое распознавание текста является исследуемой проблемой в областях распознавания образов, искусственного интеллекта и компьютерного зрения.

Системы оптического распознавания текста требуют калибровки для работы с конкретным шрифтом; в ранних версиях для программирования было необходимо изображение каждого символа, программа одновременно могла работать только с одним шрифтом. В настоящее время больше всего распространены так называемые «интеллектуальные» системы, с высокой степенью точности распознающие большинство шрифтов. Некоторые системы оптического распознавания текста способны восстанавливать исходное форматирование текста, включая изображения, колонки и другие нетекстовые компоненты.

Точное распознавание латинских символов в печатном тексте в настоящее время возможно только если доступны чёткие изображения, такие как сканированные печатные документы. Точность при такой постановке задачи превышает 99%, абсолютная точность может быть достигнута только путем последующего редактирования человеком. Проблемы распознавания рукописного «печатного» и стандартного рукописного текста, а также печатных текстов других форматов (особенно с очень большим числом символов) в настоящее время являются предметом активных исследований.

Точность работы методов может быть измерена несколькими способами и поэтому может сильно варьироваться. К примеру, если встречается специализированное слово, не используемое для соответствующего программного обеспечения, при поиске несуществующих слов, ошибка может увеличиться.

Распознавание символов он-лайн иногда путают с оптическим распознаванием символов. Последний — это офф-лайн метод, работающий со статической формой представления текста, в то время как он-лайн распознавание символов учитывает движения во время письма. Например, в он-лайн распознавании, использующем PenPoint OS или планшетный ПК, можно определить, с какой стороны пишется строка: справа налево или слева направо.

Он-лайн системы для распознавания рукописного текста «на лету» в последнее время стали широко известны в качестве коммерческих продуктов. Алгоритмы таких устройств используют тот факт, что порядок, скорость и направление отдельных участков линий ввода известны. Кроме того, пользователь научится использовать только конкретные формы письма. Эти методы не могут быть использованы в программном обеспечении, которое использует сканированные бумажные документы, поэтому проблема распознавания рукописного «печатного» текста по-прежнему остается открытой. На изображениях с рукописным «печатным» текстом без артефактов может быть достигнута точность в 80% — 90%, но с такой точностью изображение будет преобразовано с десятками ошибок на странице. Такая технология может быть полезна лишь в очень ограниченном числе приложений.

Ещё одной широко исследуемой проблемой является распознавание рукописного текста. На данный момент достигнутая точность даже ниже, чем для рукописного «печатного» текста. Более высокие показатели могут быть достигнуты только с использованием контекстной и грамматической информации. Например, в процессе распознавания искать целые слова в словаре легче, чем пытаться проанализировать отдельные символы из текста. Знание грамматики языка может также помочь определить, является ли слово глаголом или существительным. Формы отдельных рукописных символов иногда могут не содержать достаточно информации, чтобы точно (более 98%) распознать весь рукописный текст.

Для решения более сложных проблем в сфере распознавания используются как правило интеллектуальные системы распознавания, такие как искусственные нейронные сети.

## 1.2 Персептрон

Ф. Розенблатт вводя понятие о модели мозга, задача которой состоит в том, чтобы показать, как в некоторой физической системе, структура и функциональные свойства которой известны, могут возникать психологические явления — описал простейшие эксперименты по различению. Данные эксперименты целиком относятся к методам распознавания образов, но отличаются тем, что алгоритм решения не детерминированный.

Простейший эксперимент, на основе которого можно получить психологически значимую информацию о некоторой системе, сводится к тому, что модели предъявляются два различных стимула и требуется, чтобы она реагировала на них различным образом. Целью такого эксперимента может быть исследование возможности их спонтанного различения системой при отсутствии вмешательства со стороны экспериментатора, или, наоборот, изучение принудительного различения, при котором экспериментатор стремится обучить систему проводить требуемую классификацию.

В опыте с обучением персептрону обычно предъявляется некоторая последовательность образов, в которую входят представители каждого из классов, подлежащих различению. В соответствии с некоторым правилом модификации памяти правильный выбор реакции подкрепляется. Затем персептрону предъявляется контрольный стимул и определяется вероятность получения правильной реакции для стимулов данного класса. В зависимости от того, совпадает или не совпадает выбранный контрольный стимул с одним из образов, которые использовались в обучающей последовательности, получают различные результаты:

Если контрольный стимул не совпадает ни с одним из обучающих стимулов, то эксперимент связан не только с чистым различением, но включает в себя и элементы обобщения. Если контрольный стимул возбуждает некоторый набор сенсорных элементов, совершенно отличных от тех элементов, которые активизировались при воздействии ранее предъявленных стимулов того же класса, то эксперимент является исследованием чистого обобщения.

Персептроны не обладают способностью к чистому обобщению, но они вполне удовлетворительно функционируют в экспериментах по различению, особенно если контрольный стимул достаточно близко совпадает с одним из образов, относительно которых персептрон уже накопил определенный опыт.

Элементарный персептрон состоит из элементов 3-х типов: S-элементов, A-элементов и одного R-элемента. S-элементы — это слой сенсоров, или рецепторов. В физическом воплощении они соответствуют, например, светочувствительным клеткам сетчатки глаза или фоторезисторам матрицы камеры. Каждый рецептор может находиться в одном из двух состояний — покоя или возбуждения, и только в последнем случае он передаёт единичный сигнал в следующий слой, ассоциативным элементам.

A-элементы называются ассоциативными, потому что каждому такому элементу, как правило, соответствует целый набор (ассоциация) S-элементов. A-элемент активизируется, как только количество сигналов от S-элементов на его входе превысило некоторую величину  $\theta$ . Таким образом, если набор соответствующих S-элементов располагается на сенсорном поле в форме буквы «Д», A-элемент активизируется, если достаточное количество рецепторов сообщило о появлении «белого пятна света» в их окрестности, то есть A-элемент будет как бы ассоциирован с наличием/отсутствием буквы «Д» в некоторой области.

Сигналы от возбужденных A-элементов, в свою очередь, передаются в сумматор R, причём сигнал от  $i$ -го ассоциативного элемента передаётся с коэффициентом  $w_i$ . Этот коэффициент называется весом A — R связи.



Так же как и  $A$ -элементы,  $R$ -элемент подсчитывает сумму значений входных сигналов, помноженных на веса (линейную форму).  $R$ -элемент, а вместе с ним и элементарный перцептрон, выдаёт «1», если линейная форма превышает порог  $\theta$ , иначе на выходе будет «-1». Математически, функцию, реализуемую  $R$ -элементом, можно записать так:

$$f(x) = \text{sign}\left(\sum_{i=1}^n w_i x_i - \theta\right)$$

Обучение элементарного перцептрона состоит в изменении весовых коэффициентов  $w_i$  связей  $A - R$ . Веса связей  $S - A$  (которые могут принимать значения  $\{-1; \theta; +1\}$ ) и значения порогов  $A$ -элементов выбираются случайным образом в самом начале и затем не изменяются. (Описание алгоритма см. ниже.)

После обучения перцептрон готов работать в режиме распознавания или обобщения. В этом режиме перцептрону предъявляются ранее неизвестные ему объекты, и перцептрон должен установить, к какому классу они принадлежат. Работа перцептрона состоит в следующем: при предъявлении объекта, возбудившиеся  $A$ -элементы передают сигнал  $R$ -элементу, равный сумме соответствующих коэффициентов  $w_i$ . Если эта сумма положительна, то принимается решение, что данный объект принадлежит к первому классу, а если она отрицательна — то ко второму.

## 1.3 Алгоритмы обучения

Важным свойством любой нейронной сети является способность к обучению. Процесс обучения является процедурой настройки весов и порогов с целью уменьшения разности между желаемыми (целевыми) и получаемыми векторами на выходе. В своей книге Розенблатт пытался классифицировать различные алгоритмы обучения перцептрона, называя их системами подкрепления.

Система подкрепления - это любой набор правил, на основании которых можно изменять с течением времени матрицу взаимодействия (или состояние памяти) перцептрона.

### 1.3.1 Обучение с учителем

Классический метод обучения перцептрона — это метод коррекции ошибки. Он представляет собой такой вид обучения с учителем, при котором вес связи не изменяется до тех пор, пока текущая реакция перцептрона остается правильной. При появлении неправильной реакции вес изменяется на единицу, а знак (+/-) определяется противоположным от знака ошибки.

Допустим, мы хотим обучить перцептрон разделять два класса объектов так, чтобы при предъявлении объектов первого класса выход перцептрона был положителен (+1), а при предъявлении объектов второго класса — отрицательным (1). Для этого выполним следующий алгоритм:[5]

Случайным образом выбираем пороги для  $A$ -элементов и устанавливаем связи  $S - A$  (далее они изменяться не будут). Начальные коэффициенты  $w_i$  полагаем равными нулю. Предъявляем обучающую выборку: объекты (например, круги либо квадраты) с указанием класса, к которым они принадлежат.

Показываем перцептрону объект первого класса. При этом некоторые  $A$ -элементы возбуждаются. Коэффициенты  $w_i$ , соответствующие этим возбуждённым элементам, увеличиваем на 1.

Предъявляем объект второго класса и коэффициенты  $w_i$  тех  $A$ -элементов, которые возбуждаются при этом показе, уменьшаем на 1.

Обе части шага 3 выполним для всей обучающей выборки. В результате обучения сформируются значения весов связей  $w_i$ .

Теорема сходимости перцептрона, описанная и доказанная Ф. Розенблаттом (с участием Блока, Джозефа, Кестена и других исследователей, работавших вместе с ним), показывает, что элементарный перцептрон, обучаемый по такому алгоритму, независимо от начального состояния весовых коэффициентов и последовательности появления стимулов всегда приведет к достижению решения за конечный промежуток времени.

### 1.3.2 Обучение без учителя

Кроме классического метода обучения перцептрона Розенблатт также ввёл понятие об обучении без учителя, предложив следующий способ обучения:

Альфа-система подкрепления - это система подкрепления, при которой веса всех активных связей  $c_{ij}$ , которые ведут к элементу  $u_j$ , изменяются на одинаковую величину  $r$ , а веса неактивных связей за это время не изменяются.

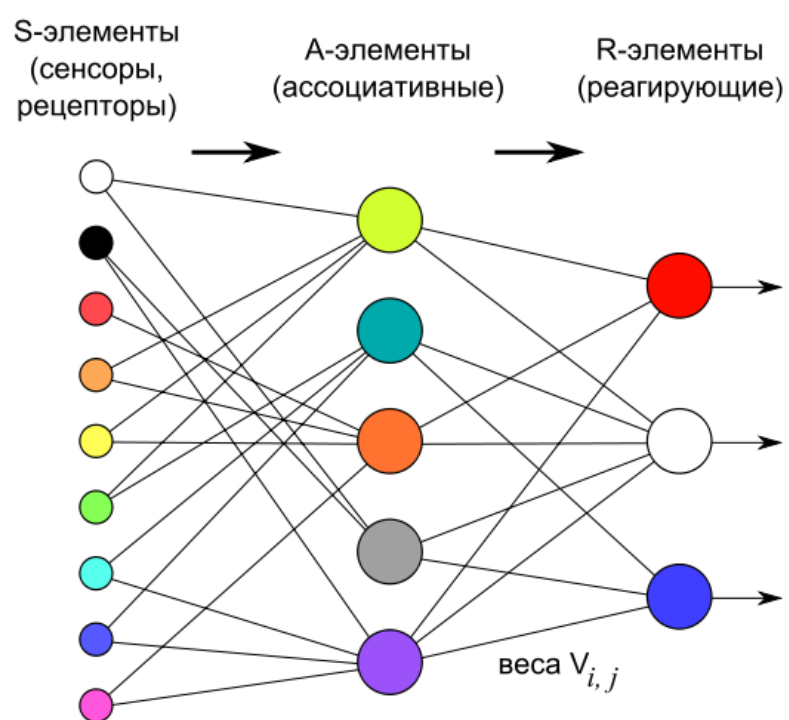
Затем, с разработкой понятия многослойного перцептрона, альфа-система была модифицирована и её стали называть дельта-правило. Модификация была проведена с целью сделать функцию обучения дифференцируемой (например, сигмоидной), что в свою очередь нужно для применения метода градиентного спуска, благодаря которому возможно обучение более одного слоя.

### 1.3.3 Метод обратного распространения ошибки

Для обучения многослойных сетей рядом учёных, в том числе Д. Румельхартом, был предложен градиентный алгоритм обучения с учителем, проводящий сигнал ошибки, вычисленный выходами перцептрона, к его входам, слой за слоем. Сейчас это самый популярный метод обучения многослойных перцептронов. Его преимущество в том, что он может обучить все слои нейронной сети, и его легко просчитать локально. Однако этот метод является очень долгим, к тому же, для его применения нужно, чтобы передаточная функция нейронов была дифференцируемой. При этом в перцептронах пришлось отказаться от бинарного сигнала, и пользоваться на входе непрерывными значениями.

## 1.4 Поисковые системы

1.



## Глава 2

# Распознавание изображений

### 2.1 Базовые понятия

Определение 2.1.1. Растровое изображение есть функция

$$I_{rgb}(x, y) : N \times N \rightarrow ([0, 255], [0, 255], [0, 255])$$

Таким образом каждой точке  $(x, y)$  мы сопоставляем тройку  $(r, g, b)$ . Первый элемент тройки соответствует красной компоненте цвета в растровом изображении, второй – зеленой и третий – синей. Далее, для краткости, будем использовать следующую запись:

$$\begin{aligned} I_r(x, y) &= I_{rgb}(x, y)_r - \text{красная компонента} \\ I_g(x, y) &= I_{rgb}(x, y)_g - \text{зеленая компонента} \\ I_b(x, y) &= I_{rgb}(x, y)_b - \text{синяя компонента} \end{aligned}$$

Определение 2.1.2. Определим растровое изображение заданного в оттенках серого как функцию

$$I_{grey}(x, y) : N \times N \rightarrow [0, 255]$$

Определение 2.1.3. Введем оператор «обесцвечивания»  $D$ , который обеспечивает переход от цветного изображения  $I_{rgb}$  к изображению  $I_{grey}$ , заданному в оттенках серого:

$$D(I_{rgb}) = I_{grey}$$

Существует несколько основных способов обесцвечивания изображения:

1. Красный канал  $D_{red}(I_{rgb}) = I_r$
2. Зеленый канал  $D_{green}(I_{rgb}) = I_g$
3. Синий канал  $D_{blue}(I_{rgb}) = I_b$
4. Среднее значение (average):  $D_{avg}(I_{rgb}) = \frac{I_r + I_g + I_b}{3}$
5. Лума (luma), учитывает особенности восприятия цвета человеком:

$$D_{luma}(I_{rgb}) = I_r \cdot 0.3 + I_g \cdot 0.59 + I_b \cdot 0.11$$

Значения коэффициентов, иногда, могут отличаться от приведенных выше, но их сумма всегда равна 1

6. Минимум  $D_{min}(I_{rgb}) = \min(I_r, I_g, I_b)$

7. Максимум  $D_{min}(I_{rgb}) = \max(I_r, I_g, I_b)$

8. Обесцвечивание (desaturtaion):  $D_{desaturation}(I_{rgb}) = \frac{D_{min}(I_{rgb}) + D_{max}(I_{rgb})}{2}$

Замечание 2.1. Наилучший результат для средне-статистического изображения (с гистограммой близкой к нормальной) получается при использовании 5-го и последнего способов. Под наилучшим результатом понимается сохранение яркости (компоненты value в модели HSV) цветов исходного изображения.

Определение 2.1.4. Растровое монохромное изображение есть функция

$$I_m(x, y) : N \times N \rightarrow \{0, 1\}$$

Переход от изображения заданного в оттенках серого к монохромному изображению осуществляется через операцию отсечения. Операция отсечения реализуется через оператор отсечения  $T$ , для некоторого фиксированного  $t \in [0, 255]$

Определение 2.1.5. Оператор отсечения  $T$  есть:

$$T(t, I_{grey}) = \begin{cases} 1 & , I_{grey} < t \\ 0 & , \text{иначе} \end{cases}$$

Таким образом монохромное изображение есть  $I_m = T(t, I_{grey})$

Замечание 2.2. Значение  $t$  определяется опытным путем и зависит от исходного изображения. Для рукописного текста написанного черной ручкой на офисной бумаге берутся значения близкие к 100 (чем меньше значение, тем темнее изображение).

Замечание 2.3. В данной работе не рассматриваются методы адаптивного отсечения, в силу их медленной производительности излишней в данном контексте точности.

Определение 2.1.6. Точки  $(x, y)$  для которых верно  $I(x, y) = 1$  будем называть заполненными.

Определение 2.1.7. Точки  $(x, y)$  для которых верно  $I(x, y) = 0$  будем называть пустыми.

Множество заполненных точек образует множество связных областей

$$\{K_1, K_2 \dots K_n\},$$

таких что:

1.  $\forall (x_1, y_1) \forall (x_2, y_2) (|x_1 - x_2| > 1 \wedge |y_1 - y_2| > 1)$ , где  $(x_1, y_1) \in K_i$ ,  $(x_2, y_2) \in K_j$  и  $i \neq j$
2.  $\forall (x_1, y_1) \exists (x_2, y_2) (|x_1 - x_2| \leq 1 \wedge |y_1 - y_2| \leq 1)$ , где  $(x_1, y_1), (x_2, y_2) \in K_i$  и  $(x_1, y_1) \neq (x_2, y_2)$
3.  $|K_i| \geq 2$

Определение 2.1.8. Всякую связную область  $K_i$  будем называть контуром

Определение 2.1.9. Растровое изображение  $I$  содержащее по крайней мере один контур будем называть растровым контурным изображением

Замечание 2.4. Не исключая общности, далее будут рассматриваться только растровые изображения содержащие один контур, а под растровым контурным изображением будет пониматься растровое изображение содержащие только один контур.

## 2.2 Модель контурного изображения

В качестве математической модели представления растрового контурного изображения будем использовать четырех-основную алгебраическую систему вида.

Определение 2.2.1. Контурное изображение (далее, изображение) есть система вида

$$\mathfrak{M} = \langle A, R, V, M; \text{Sector}, \text{Angle}, \text{Metric}, \text{Relation} \rangle \quad (2.1)$$

где

$A$  – множество всевозможных дуг,

$R$  – множество связей дуг,

$V \subset Z$  – множество допустимых углов (например от 0 до 360 градусов),

$M \subset Z$  – множество относительных мер,

$\text{Sector} : A \rightarrow V$  – задает градусную меру дуги,

$\text{Metric} : A \rightarrow M$  – функция сопоставляющая каждой дуге ее относительную величину,

$\text{Angle} : R \rightarrow V$  – задает угол соединения двух дуг

$\text{Relation} : R \rightarrow A \times A$  – сопоставляет каждой связи дуги, те дуги, которые она соединяет.

Замечание 2.5. Важно отметить, что в этом представлении все множества являются конечными, и, если множества  $A$  и  $R$  являются фиктивными (чисто техническими элементами) данной модели, и определяются через функции, то множество  $V = \{v_0, v_1, \dots, v_n\}$  – есть конечное множество чисел, с максимальным  $v_{max}$  и минимальным  $v_{min}$  элементами, разбитое шагом  $\delta$  на  $n + 1$  элементов, где

$$\begin{aligned} n &= \frac{v_{max} - v_{min}}{\delta} \\ v_0 &= v_{min} \\ v_i - v_{i-1} &= \delta, \forall i = 1, \dots, n \\ v_n &= v_{max} \end{aligned}$$

Замечание 2.6. Как и множество  $V$ , множество  $M$  конечно. Однако выбор верхней границы для  $M$  не столь очевиден, так как не исключена возможность того, что разница в размерах между двумя дугами может быть весьма существенна (например в несколько миллионов раз). Однако, в рамках нашей области применения (распознавание символов), когда в качестве «эталонного наблюдателя» выступает человеческий глаз, разница что в 1000, что в 1000000 раз почти неразличима, и поэтому ею вполне можно пренебречь, выбрав в качестве максимального значения например 100000 процентов, а в качестве шага одну десятую процента. Таким образом всякая дуга может быть как больше так и меньше любой дуги не более чем в 1000 раз.

Для наших целей важно всегда работать только с конечными множествами, что достигается рассмотрением конечных множеств  $A$ ,  $R$ , а также предположением о наличии минимального шага возрастания количественных характеристик дуг и связей дуг.

Таким образом контурное изображение будет представляет собой систему дуг и связей дуг. Где всякая дуга определяется через ее градусную меру и через относительную (в данном контуре) длину дуги. А всякая связь определяется через угол связи и пару дуг которые она связывает.

## 2.3 Преобразование растрового изображения

Переход от растрового контурного изображения к изображению состоит из двух этапов. Первый этап — волновая скелетизация. С помощью скелетизации на основе растрового изображения строится граф (скелет), который визуальнo адекватно соответствует исходному изображению.

Пусть  $I$  — растровое контурное изображение и  $K$  — есть его контур.

Определение 2.3.1. Точку  $q(x_1, y_1)$  будем называть соседом точки  $p(x, y)$  если  $|x - x_1| \leq 1$  и  $|y - y_1| \leq 1$  и  $p \neq q$ . Введем отношение соседства  $N(p, q)$ , которое истинно если  $p$  сосед  $q$ .

Очевидно что точка  $p$  не может иметь более 8 соседей. Обозначим через  $N_K^p$  множество всех соседей точки  $p(x, y)$  лежащих контуре  $K$ :

$$N_K^p = \{q | q \in K \wedge N(p, q)\}$$

Согласно определению контура (2.1.8) очевидно, что  $K$  не имеет изолированных точек т.е.

$$\forall p \exists q : N(p, q)$$

$$p, q \in K$$

Замечание 2.7. Скелетом  $I$  будем называть граф  $G(V, E)$ , «интуитивно адекватно отражающий» исходное изображение.

Определение 2.3.2. Волной  $w$  будем называть конечное множество точек  $\{p_j\}$ .

Определение 2.3.3. Множество волн  $\{w_1, w_2, \dots, w_n\}$  будем называть подволнами волны  $w$  если:

$$\bigcup_{i=1, n} w_i = w$$

$$w_i \cap w_j = \emptyset, i, j = \overline{1, n}, i \neq j;$$

и для любых двух точек  $p \in w_i, g \in w_j$ , где  $i \neq j$  верно  $\neg N(p, g)$ .

Введем функцию вычисляющую центр масс точек волны

$$g(w) = \frac{\sum_{p \in w} p}{|w|}$$

### 2.3.1 Волновая скелетизация

Опишем алгоритм построения скелета растрового контурного изображения.

Зададим начальные условия. В качестве начальной волны подойдет любая точка из  $F$ .

Имеем следующую начальную конфигурацию:

$w_0^0 = \{p\}, p \in K$  — начальная волна,

$W_0 = \{w_0^0\}$  — множество волн,

$F_0 = K$  — состояние заполненной области,

$G_0(V_0, E_0), V_0 = \{p\}, E_0 = \emptyset$  — начальное состояние скелета.

Определим  $n$ -ый шаг итерации следующим образом. Для всякой  $i$ -ой волны  $w_i^{k_i-1}$  из  $W_{n-1}$  ( $k_i$  — соответствует  $k_i$ -ой итерации  $w_i$ ):

$$w_i^{k_i} = \bigcup_{p \in w_i^{k_i-1}} N_{F_{n-1}}^p \setminus \bigcup_{j < i} w_j^{k_j-1} \quad (2.2)$$

Если  $u_1, \dots, u_m$  есть подволны волны  $w_i^{k_i}$ , тогда

$$W_n^i = \{w_{l+1}, \dots, w_{l+m}\}$$

где

$$w_{l+j} = u_j, j = \overline{1, m}$$

$$l = |W_{n-1}| + \sum_{j < i} |W_n^j|.$$

Ребра в графе образуют вектора, связывающие центры масс полученных подволн с центром массы  $w_i^{k_{i-1}}$

$$E_n^i = \{(v_i^{k_{i-1}}, v_{l+j}^{k_i})\}, j = \overline{1, m}$$

$$V_n^i = \{v_{l+j}^{k_i}\}, j = \overline{1, m}$$

$$v_i^j = g(w_i^j)$$

Если же волна  $w_i^{k_i}$  не имеет разрывов и  $w_i^{k_i} \neq \emptyset$ , то

$$W_n^i = \{w_i^{k_i}\}$$

$$E_n^i = \{(v_i^{k_{i-1}}, v_i^{k_i})\}$$

$$V_n^i = \{v_i^{k_i}\}$$

Если  $w_i^{k_i} = \emptyset$

$$W_n^i = \emptyset, E_n^i = \emptyset, V_n^i = \emptyset$$

Таким образом, при  $s_n = |W_{n-1}|$ :

$$W_n = \{W_n^i\}, i = \overline{1, s_n}$$

$$V_n = V_{n-1} \bigcup_i V_n^i, i = \overline{1, s_n}$$

$$E_n = E_{n-1} \bigcup_i E_n^i, i = \overline{1, s_n}$$

$$F_n = F_{n-1} \setminus P_{n-1},$$

$$P_{n-1} = \{p : p \in w, w \in W_{n-1}\}$$

Если  $|F_n| = 0$  то алгоритм прекращает цикл итераций, а граф

$$G = (V_n, E_n)$$

является скелетом исходного изображения  $f$ .

Утверждение 2.3.1. Алгоритм волновая скелетизация, остановится на любом контуре мощности  $m$  удовлетворяющей условиям алгоритма.

Доказательство. Пусть  $m = 1$ , тогда согласно алгоритму

$$|F_1| = |F_0 \setminus \{p : p \in w, w \in W_0\}| = |\{p\} \setminus \{p : p \in w_0\}| = |\{p\} \setminus \{p\}| = \emptyset$$

следовательно алгоритм прекращает свою работу а граф  $G = (V_1, E_1) = (\{p\}, \emptyset)$  является скелетом изображения.

Пусть  $m > 2$  и существует такое  $l$  что для всякого  $k < l, |F_k| < |F_{k-1}|$  и  $|F_l| = |F_{l-1}|$ , тогда  $P_{n-1} = \emptyset$ , отсюда следует, что  $\{p : p \in w, w \in W_{n-1}\} = \emptyset$ , а это возможно только в двух случаях:



1. если  $W_{n-1}$  пусто, тогда, в силу и в силу отсутствия изолированных точек,  $F_{n-2} = \emptyset$ , получаем противоречие с условием остановки.
2. если  $\forall w \in W : |w| = 0$ , тогда, опять же в силу и в силу отсутствия изолированных точек, получаем что  $F_{n-2} = \emptyset$ , снова получаем противоречие с условием остановки.

Следовательно такого  $l$  не существует, и алгоритм сходится для всякой непустой заполненной области без изолированных точек.  $\square$

### 2.3.2 Построение модели $\mathfrak{M}$ для графа $G$

Прежде чем дать формальное описание алгоритма, рассмотрим как он работает неформально:

1. Для каждого простого пути выполняется:
  - (a) Разбиение пути по точкам смены направления обхода
  - (b) Для каждого разбиения выполняются
    - i. Разбиение спиралей. Чтобы определить закручен ли путь по спирали, надо проверить пересекает ли хорда путь. Если пересечение есть, то необходимо разбить путь точками пересечения
    - ii. Для каждого разбиения выполняется:
      - А. Разбиение по точкам перегиба. Точками перегиба считаются образующие две дуги отклонившиеся от угла идеального соединения. Угол  $\gamma$  идеального соединения двух дуг градусной меры  $\alpha$  и  $\beta$ :

$$\gamma = 2\pi - (\alpha + \beta)$$

2. Результатом п.1 является множество подпутей, каждый из которых переводится в дугу. Градусная мера дуги вычисляется с использованием формулы Гюйгенса. Определив наиболее удаленную точку пути от хорды стягивающей путь, и вычислив её положение относительно хорды направленной от начала к концу пути, мы определяем направление обхода. Если точка слева, то обход ведется по часовой стрелке, если точка справа — обход ведется против часовой стрелки, если же точка лежит на прямой, то верны оба утверждения.
3. Расчет связей дуг. Связь между двумя дугами существует, если пути образующие дуги имели общие вершины. Угол соединения между дугами рассчитывается, как угол между стягивающими их хордами

Замечание 2.8. Стоит отметить что разбиение на дуги, как правило, выполняется на интерполированном графе в котором часть узлов удаленно в силу их избыточности.

Замечание 2.9. Хотя разбиение и может быть использовано как есть, на практике полезнее добавлять возможность вариации параметров, например, допускать возможность некоторого отклонения от угла идеального соединения или для точек смены направления расширять область проверки на смену направления обхода.

Пусть  $\{P_1, P_2, \dots, P_m\}$  — множество простых цепей графа  $G = (V, E)$  таких что:

1.  $P_i = v_1^i, v_2^i, \dots, v_{n_i}^i$
2.  $\forall v \in P_i \forall u \in P_j (v \neq u), \text{ где } i \neq j$
3.  $d(v_1^i) \neq 2 \text{ и } d(v_{n_i}^i) \neq 2$
4.  $\forall j \notin \{1, n_i\} \left[ d(v_j^i) = 2 \right]$
5.  $\bigcup_i P_i = V$

Определение 2.3.4. Пусть  $r(p, v)$  – есть расстояние от точки  $p$  до вектора  $v$  со знаком

Определение 2.3.5. Будем говорить что в узле  $v_i$  меняется направление обхода простого

$$\{v_1, \dots, v_{i-1}, v_i, v_{i+1}, \dots, v_n\}$$

в окрестности  $\varepsilon \in \{2, 3, \dots\}$ , если

$$\text{Sign}\left(\sum_{j=i-\varepsilon} r(v_j, h)\right) \neq \text{Sign}\left(\sum_{j=i+\varepsilon} r(v_j, h)\right)$$

где  $h = (v_{i-\varepsilon}, v_{i+\varepsilon})$

Если соотношение выполняется для  $\varepsilon = 2$  будем просто говорить, что в точке  $v_i$  меняется направление обхода.

Пусть множество точек  $\{v_{j_1}^i, \dots, v_{j_{k_i}}^i\}$  множество точек смены направления обхода цепи  $P_i$ . Определим множество индексов задающих разбиение пути  $P_i = v_1, \dots, v_n$  по направлению обхода

$$I_{P_i}^{dev} = \{j_1^i, \dots, j_{k_i}^i\} \cup \{1, n\}$$

$$j_k^i < j_{k+1}^i$$

Тогда следующие семейство множеств узлов определяют разбиение пути  $P_i$  на подпути:

$$S_{P_i}^{dev} = \bigcup_{l \in I_{P_i}^{dev}} \{\{v_t \mid t \in \{j \mid i_l \leq j \leq i_{l+1}\}\}\}$$

Каждому множеству узлов можно однозначно сопоставить подпуть  $P_i$

Определение 2.3.6. Определим множество индексов узлов задающие разбиение подпути  $P = v_1, \dots, v_n$  закрученного в спираль следующим образом

$$I_P^{spir} = \left\{ \min_{|r(h, v_i)|} (i, i+1) \mid \text{Sign}(r(h, v_i)) \neq \text{Sign}(r(h, v_{i+1})) \wedge i = 1, n-1 \right\}$$

$$h = (v_1, v_n)$$

Определение 2.3.7. Определим множество индексов узлов задающие разбиение подпути  $P = v_1, \dots, v_n$  по резкости угла перегиба.

Определение 2.3.8. Разбиением пути  $P$  на характеристические подпути будем считать разбиение определенное следующим образом:

$$S_{P_i} = \bigcup_{j \in I_P^{spir}, P \in S_{P_i}^{dev}} \{\{v_t \mid t \in \{j \mid i_l \leq j \leq i_{l+1}\}\}\}$$

Определение 2.3.9. Пусть  $P = v_1, \dots, v_n$  простой путь тогда:

1.  $f_{begin}(P) = v_1$  – начало пути
2.  $f_{end}(P) = v_n$  – конец пути
3.  $h_P = v_n - v_1$  – вектор хорды стягивающей путь

Определение 2.3.10. Пусть задано разбиение графа  $G = (V, E)$  на характеристические подпути  $S = \{P_1, \dots, P_m\}$ , где  $P_i = v_1^i, \dots, v_{n_i}^i$ , будем говорить что алгебраическая система

$$\mathfrak{M} = \langle A, R, V, M; Sector, Angle, Metric, Relation \rangle$$

соответствует графу  $G$  если выполняются следующие условия:

1.  $Sector(a_i) = 2l + \frac{1}{3}(2l - L)$ , где  $l = |v_1^i - v_m^i|$ ,  $i_m = \frac{n_i}{2}$ , а  $L = |h_{P_i}|$
2.  $Metric(a_i) = \frac{a_i}{a_1}$
3.  $Relation(r_{ij}) = (a_i, a_j)$ , где  $i \neq j$  и неопределенно в противном случае

$$4. Angle(r_{ij}) = \left\{ \begin{array}{ll} (-h_{P_i}, -h_{P_j}) & f_{begin}(P_i) = f_{end}(P_j) \\ (-h_{P_i}, h_{P_j}) & f_{begin}(P_i) = f_{begin}(P_j) \\ (h_{P_i}, h_{P_j}) & f_{end}(P_i) = f_{begin}(P_j) \\ (h_{P_i}, -h_{P_j}) & f_{end}(P_i) = f_{end}(P_j) \\ \text{неопределенно,} & \text{если } r_{ij} - \text{неопределенно} \end{array} \right\}$$

## 2.4 Постановка задачи распознавания

Общую схему решения комбинаторных задач высокой сложности логико-эвристическими методами можно трактовать как отображение начальной (инициальной) многоосновной алгебраической системы.

$$\mathfrak{M}_{ini} = \langle A_1, \dots, A_s; f_1, \dots, f_n; p_1, \dots, p_k \rangle \quad (2.3)$$

где  $A_i$  – основные множества,  $f_i$  – операции (функции) на основных множествах,  $p_i$  – предикаты (отношения) на основных множествах, в конечную (финальную)  $\mathfrak{M}_{fin}$ , удовлетворяющую ограничениям  $\mathfrak{R}_1, \mathfrak{R}_2, \dots, \mathfrak{R}_m$ .

Уточним формализацию описания плоских контурных изображений для данного варианта проверки выполнимости ограничений.

В качестве составляющих элементов образцов и анализируемого изображения выступают, очевидно, дуги и связи дуг.

Пусть многоосновные а.с.

$$\begin{aligned} \mathfrak{R}_1 &= \langle A_1, R_1, V, M; Sector, Angle, Metric, Relation \rangle \\ \mathfrak{R}_2 &= \langle A_2, R_2, V, M; Sector, Angle, Metric, Relation \rangle \\ &\dots \\ \mathfrak{R}_m &= \langle A_m, R_m, V, M; Sector, Angle, Metric, Relation \rangle \end{aligned} \quad (2.4)$$

задают искомые образцы в анализируемом изображении 2.1.

Анализ изображения 2.1 состоит в поиске всех изоморфных вложений  $\mu_{i,j}$  многоосновных а.с.  $\mathfrak{R}_1, \mathfrak{R}_2, \dots, \mathfrak{R}_m$  в многоосновную а.с.  $\mathfrak{M}$  2.1, т.е. изоморфное вложение  $\mu_{i,j} : \mathfrak{R}_i \rightarrow \mathfrak{M}$  состоит из инъективных отображений

$$\mu_{i,j} : A_i \rightarrow A$$

$$\mu_{i,j} : R_i \rightarrow R$$

таких, что:

а) если  $\mu_{i,j}(a') = a$ , где  $a' \in A_i, a \in A$  и

$$Sector(a') = [c'_{min}, c'_{max}],$$

$$Sector(a) = [c_{min}, c_{max}],$$

$$\text{то } [c_{min}, c_{max}] \in [c'_{min}, c'_{max}];$$

б) если  $\mu_{i,j}(a') = a$ , где  $a' \in A_i, a \in A$  и

$$Metric(a') = [c'_{min}, c'_{max}],$$

$$Metric(a) = [c_{min}, c_{max}],$$

$$\text{то } [c_{min}, c_{max}] \in [c'_{min}, c'_{max}];$$

в) если  $\mu_{i,j}(r') = r$ , где  $r' \in R_i, r \in R$  и

$$Angle(r') = [c'_{min}, c'_{max}],$$

$$Angle(r) = [c_{min}, c_{max}],$$

$$\text{то } [c_{min}, c_{max}] \in [c'_{min}, c'_{max}];$$

г) если  $Relation(r, a_1, a_2)$ , где  $r \in R_i, a_1, a_2 \in A_i$ , то

$$Relation(\mu_{i,j}(r), \mu_{i,j}(a_1), \mu_{i,j}(a_2))$$

## 2.5 Интерпретация

Определим универсум для изображений, имеющих не более  $n$  дуг, и не более  $k$  связей дуг (это ограничение ни сколько не повлияет на результат, но немного упростит нам форму записи).

Пусть  $A_1, A_2, \dots, A_m$  – множества дуг всех характеристик (образцов), т.е.

$$\begin{aligned} A_1 &= \{a_{1,1}, a_{1,2}, \dots, a_{1,n}\}; \\ A_2 &= \{a_{2,1}, a_{2,2}, \dots, a_{2,n}\}; \\ &\dots \\ A_m &= \{a_{m,1}, a_{m,2}, \dots, a_{m,n}\}; \end{aligned} \tag{2.5}$$

Далее пусть  $R_1, R_2, \dots, R_{m-1}$  – множества связей дуг всех характеристик, т.е.

$$\begin{aligned} R_1 &= \{r_{1,1}, r_{1,2}, \dots, r_{1,k}\}; \\ R_2 &= \{r_{2,1}, r_{2,2}, \dots, r_{2,k}\}; \\ &\dots \\ R_{m-1} &= \{r_{m-1,1}, r_{m-1,2}, \dots, r_{m-1,k}\}; \end{aligned} \tag{2.6}$$

Таким образом наше дерево соответствует следующему декартову произведению:

$$G_1 \times M_1 \times V_1 \times G_2 \times M_2 \times \dots \times V_{n-1} \times G_1 \times M_1 \tag{2.7}$$

где

$$\begin{aligned} G_i &= \{Angle(a_{i,1}), Angle(a_{i,2}), \dots, Angle(a_{i,n})\} \\ M_i &= \{Metric(a_{i,1}), Metric(a_{i,2}), \dots, Metric(a_{i,n})\} \\ V_i &= \{Sector(r_{i,1}), Sector(r_{i,2}), \dots, Sector(r_{i,k})\} \end{aligned} \tag{2.8}$$

Очевидно, что 2.7 содержит все изображения, имеющие не более  $n$  дуг, и  $k$  связей дуг. Каждый уровень дерева представляет собой набор узлов, значения которых принадлежат одному из множеств:  $V_{arc}$ ,  $M$  либо  $V_{rel}$ , где  $V_{arc} \subseteq V$  – множество допустимых значений секторов дуг,  $V_{rel} \subseteq V$  – множество допустимых углов соединения связей дуг. У всякого дерева изображений значения узлов первого уровня принадлежат  $V_{arc}$ , значения узлов последнего уровня принадлежат  $M$ .

Однако работать с таким деревом достаточно сложно, намного удобнее редуцировать третий уровень и использовать «двух-слойную» (состоящую из двух основных множеств) модель дерева. Так как множества значения углов и относительных величин ограничено, мы можем упорядочить множества дуг следующим образом.

Пусть  $A = \{a_1, \dots, a_k, \dots, a_n\}$  – множество всевозможных дуг

$$Sector(a_k) = g_i, 0 < i < |V|, V = \{g_0, \dots, g_{|V|-1}\}$$

$$Metric(a_k) = m_j, 0 < j < |M|, M = \{m_0, \dots, m_{|M|-1}\}$$

тогда  $k = i * |M| + j$ .

Таким образом мы сможем оперировать двумя основными множествами: множеством дуг  $A$  и множеством связей  $R$ .

$$|A| = |V_{rel}| * |M|$$

$$|R| = |V_{arc}|$$

$$V_{rel}, V_{arc} \subseteq V$$

Пусть дерево изображений имеет следующий вид:

Замечание 2.10. на практике универсум (схема 2) не строиться а строится только его часть, состоящая из дуг и связей дуг многоосновных а.с.  $\mathfrak{R}_1, \mathfrak{R}_2, \dots, \mathfrak{R}_m$ .

Лемма 2.5.1. Верхняя граница сложности построения интерпретации  $\xi$  для связного изображения 2.1 не превышает  $O(w + t)$ , где  $w$  – количество дуг,  $t$  – связей дуг.

Теорема 2.5.1. Пусть каждая из многоосновных а.с.  $\mathfrak{R}_1, \mathfrak{R}_2, \dots, \mathfrak{R}_m$  имеет не более  $n$  дуг и представляет связное изображение. Тогда анализ связного изображения 2.1 имеет верхнюю границу сложности не превышающую  $O((w + t) * w + m)$ , где  $w$  – количество дуг ( $t$  – количество связей дуг) изображения 2.1.

Доказательство. Построим интерпретации всех многоосновных а.с.  $\mathfrak{R}_1, \mathfrak{R}_2, \dots, \mathfrak{R}_m$  на универсуме схема 2 для всех изображений, имеющих не более  $n$  дуг, и  $k$  вариантов дуг и связей дуг (сложность этой процедуры, конечно, не входит в оценку доказываемой теоремы).

Далее, пометим все вершины схемы 2 номерами многоосновных а.с., чьи элементы соответствуют этим вершинам. Каждой многоосновной а.с.  $\mathfrak{R}_i$  сопоставим пару чисел  $(a_i, b_i)$ , где  $a_i$  – количество помеченных вершин 2, соответствующих дугам,  $b_i$  – количество помеченных вершин схемы 2, соответствующих связям дуг (конечно, помеченных номером  $i$ ).

Построим совокупность интерпретаций  $\xi_1, \xi_2, \dots, \xi_w$  на схеме 2 (с помеченными вершинами), которые отличаются выбором первой дуги для основания индукции. А именно, интерпретация  $\xi_1$  начинается традиционно с дуги  $ar_1$ , интерпретация  $\xi_2$  начинается с дуги  $ar_2$ , и так далее. Последняя интерпретация  $\xi_w$  начинается, соответственно, с дуги  $ar_w$ .

Введем для каждой интерпретации  $\xi_i$  множество пар

$$(a_{i_1}, b_{i_1}), (a_{i_2}, b_{i_2}), \dots, (a_{i_m}, b_{i_m}) \quad (2.9)$$

где  $a_{i_j}(b_{i_j})$  - количество помеченных  $j$  вершин схемы 2, соответствующих дугам (соответственно, связям дуг), полученных для интерпретации  $\xi_i$ . Если пара  $(a_{i_j}, b_{i_j})$  равна паре  $(a_j, b_j)$ , то, таким образом, найдено изоморфное вложение  $j$ -го изображения (образца) в анализируемое изображение 2.1.

В силу леммы 2.5.1, построение каждого отображения  $\xi_i$  требует не более  $w+t$  шагов и, таким образом, верхняя граница сложности поиска всех изоморфных вложений не более  $O(((w+t)*w)+m)$ , где «добавка»  $O(m)$  возникает из-за необходимости сравнивать пары 2.9 с парами  $(a_j, b_j)$ .  $\square$

Рисунок 2.1: Схема разбиения графа «особыми» точками

Рисунок 2.2: Дерево изображений

# Глава 3

## Применение

### 3.1 Распознавание символов

Наиболее естественным применением разработанного подхода является, очевидно, распознавание символов (букв, логотипов и т.п.).

Рассмотрим метод на примере анализа буквы «а»

...

Отдельно стоит упомянуть задачу поиска партизана в лесу

...

### 3.2 Оценка устойчивости битумных эмульсий

#### 3.2.1 Введение

В настоящее время трудно назвать область строительства, где бы не применялись эмульсии. Они используются в дорожном и в гражданском строительстве в качестве связующих с различными наполнителями, а также в качестве гидроизоляционных и лакокрасочных материалов. При любых технологиях использования эмульсий мы сталкиваемся с одними и теми же проблемами, касающимися подбора состава, приготовления, определения физико-механических характеристик, стабильности, контроля распада эмульсий и получения продукции с необходимыми свойствами [1]. Далее мы будем рассматривать только прямые битумные и битумно-латексные эмульсии, которые являются наиболее крупнотоннажным продуктом: мировое использование составляет миллионы тонн в год

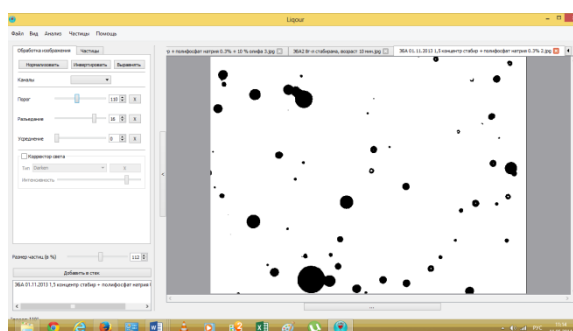
Традиционные методы оценки свойств битумных эмульсий включают: определение содержания вяжущего с эмульгатором, определение устойчивости эмульсии при перемешивании, определение остатка на сите, определение условной вязкости, определение устойчивости при хранении, определение адгезии эмульсий с поверхностью наполнителей, определение устойчивости при транспортировке и т.п. [2]. Наряду с традиционными методами изучения качества эмульсии, во многих приложениях желательно знать более тонкие характеристики, например: функцию распределения по размерам. Эта характеристика является одним из важнейших параметров и позволяет предсказывать большинство свойств эмульсии. Обычно размер частиц оценивают с помощью определения остатка на сите с заданным размером ячейки, но такой метод позволяет оценивать только верхний предел размеров частиц эмульсии. Полная картина распределения частиц по размеру может быть измерена с использованием таких технических приёмов как рассеяние света, микроскопия с анализом изображений, или же с помощью техники электроозонирования («техники Культера» - Coulter). Точный анализ размеров частиц битумной эмульсии может решить



многие проблемы, которые в настоящее время являются актуальными в сфере производства битумных эмульсий:

1. Влияние эмульгатора и его концентрации на размер битумных частиц эмульсии.
2. Влияние модифицирующих битум добавок на качество получаемой эмульсии.
3. Корректировка технологической схемы производства эмульсии.
4. Влияние размера битумных частиц на основные физические свойства эмульсии.

Оптическая микроскопия, как способ распределения частиц по размерам, является наиболее удобным и точным. Например, если в способе «рассеяние света» могут возникнуть проблемы с отражением света от черных поверхностей, какими являются частицы битума, то в способе микроскопии, при высоком контрасте черного цвета, напротив, можно отличить частицы от среды, в которой они находятся.



Очевидно изображения такого вида являются одним из примеров растровых контурных изображений, с несколькими контурами. В отличие от задачи распознавания символов, здесь нет необходимости анализировать скелет изображения. Куда более важную роль играет внешний контур. Разбивая контур на дуги мы можем классифицировать частицы по уровню распада:

1. одиночные
2. слипшиеся
3. распавшиеся

Большое количество распавшихся частиц является свидетельством того что смесь является неустойчивой, а следовательно некачественной.

### 3.2.2 Анализ эмульсий

В результате сотрудничества с кафедрой «Автомобильных дорог» НИ ИрГТУ была разработана программа для первичной оценки качества битумных эмульсий

На снимке программы (рис. 3.1), видно, что в проходящем свете частицы легко отличить друг от друга, но для получения такого изображения необходимо создать некоторое пространство между ними, в противном случае изображение получатся как на рис. 3.2. Поэтому перед микроскопическими исследованиями образцы эмульсии распределяют небольшим количеством (концентрация от 1:100 – 1:50) в специальном стабилизирующем растворе. Необходимость данной процедуры объясняется тем что, частицы битума могут слипаться под стеклышком и отсутствие сцепления между ними осуществляется за счет pH среды, в которой они находятся. Для анионных эмульсий это pH-щелочной, для катионных pH-кислотный.

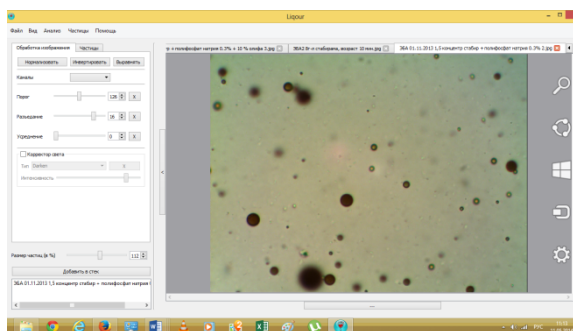


Рисунок 3.1: Скриншоты программы «анализ изображений»

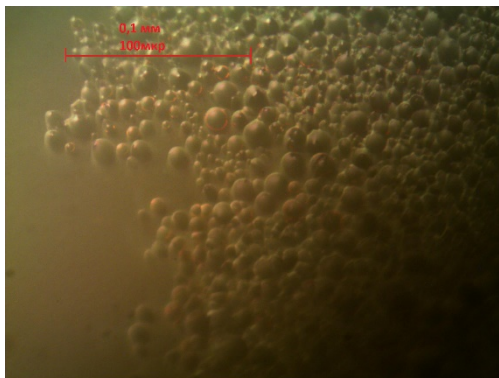


Рисунок 3.2: Концентрированная битумная анионная эмульсия в отраженном свете

Гибкость настроек программы позволяет редактировать изображение вручную, убирая «мнимые частицы», затемненные области и другие недочеты фотосъемки. После чего происходит автоматический подсчет частиц и построение графика функции распределения по размерам, рис. 5.

Для данной программы не требуется специализированного оборудования. Достаточно оптического микроскопа с возможностью подключения цифровой камеры и компьютер с ОС (Windows, Linux, OS).

### 3.2.3 Оценка качества анализа

На рисунке (рис. 3.3) приведены два графика: тот что сильнее смещен влево – соответствует исходным авто-сгенерированным данным, смещенный вправо соответствует распознанным данным. Формы графиков практически идентичны. Смещение, как правило, вызвано ошибками округления при распознавании объектов.

Таблица 3.1: Анализ распределения частиц на рис. 3.3

| График       | мин. размер частиц, мкм. | макс. размер частиц, мкм. | Среднее значение, мкм. | Ср.кв. отклонение, мкм. | Всего частиц |
|--------------|--------------------------|---------------------------|------------------------|-------------------------|--------------|
| Исходный     | 0.87                     | 2.87                      | 1.69                   | 0.42                    | 500          |
| Распознанный | 0.87                     | 2.69                      | 1.59                   | 0.41                    | 493          |

Вторым фактором (первый – ошибки округления) отрицательно влияющим на качество распознавания является наличие слипшихся частиц. На рис.3.4 приведены два гра-

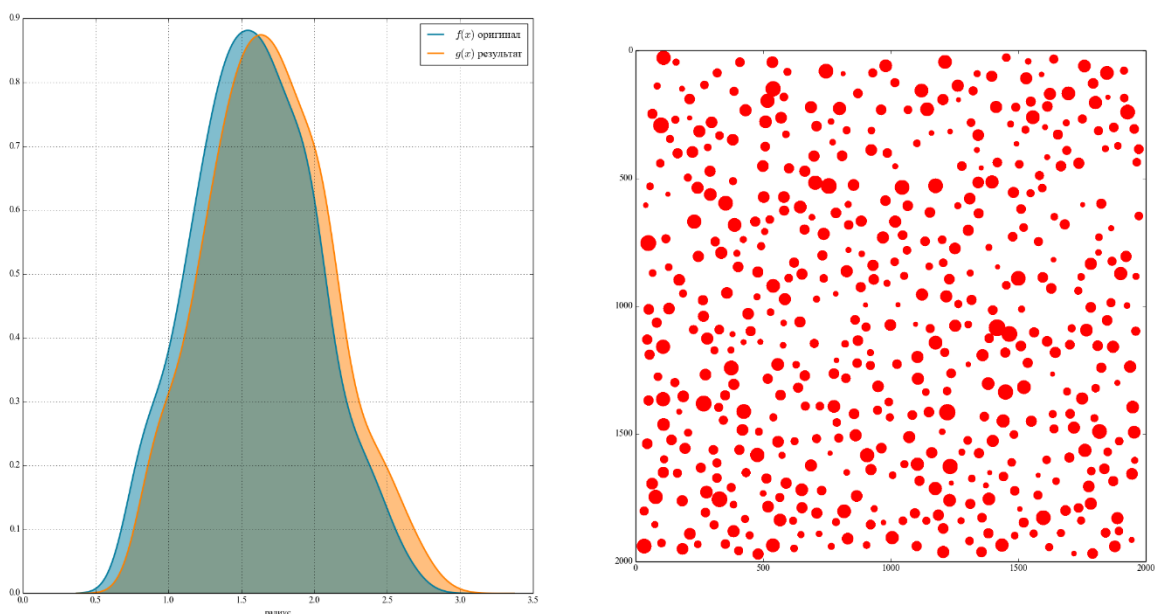


Рисунок 3.3: Результат работы программы на искусственно сгенерированных данных

фика исходного (тот что более сплюснен) и распознанного распределения частиц (тот что повыше). Сильное различие между графиками обусловлено тем что, при распознавании слипшиеся объекты не учитываются.

Таблица 3.2: Распределение частиц на рис. 3.4

| График       | мин. размер<br>частиц, мкм. | макс. размер<br>частиц, мкм. | Среднее<br>значение,<br>мкм. | Ср.кв.<br>отклонение,<br>мкм. | Всего<br>частиц |
|--------------|-----------------------------|------------------------------|------------------------------|-------------------------------|-----------------|
| Исходный     | 0.87                        | 2.98                         | 1.71                         | 0.44                          | 100             |
| Распознанный | 0.87                        | 2.52                         | 1.6                          | 0.46                          | 57              |

Из таблицы 3.2 видно, что, не смотря на то, что почти половина частиц не была распознана, характеристики распределения были определены достаточно точно к исходным, и погрешность составила около 4-5%.

Получение высококачественной, долговечной битумной эмульсии зависит в основном от вязкости битума поступающего в диспергатор вместе с раствором ПАВ, а так же от скорости вращения и вида диспергирующих элементов. После выхода готовой эмульсии из диспергатора необходимо оценить её качество.

Внешние признаки распада эмульсии видны невооруженным глазом, но если речь идет о количественном сравнении двух визуально схожих эмульсий, то в таком случае применение программы для анализа размера частиц будет весьма кстати.

### 3.2.4 Определение среднего размера и дисперсии частиц битумной эмульсии на модифицированном битуме

Стабильность эмульсии в большой степени определяется размером частиц, который в свою очередь зависит от вязкости исходного битума. Во многих практических ситуациях

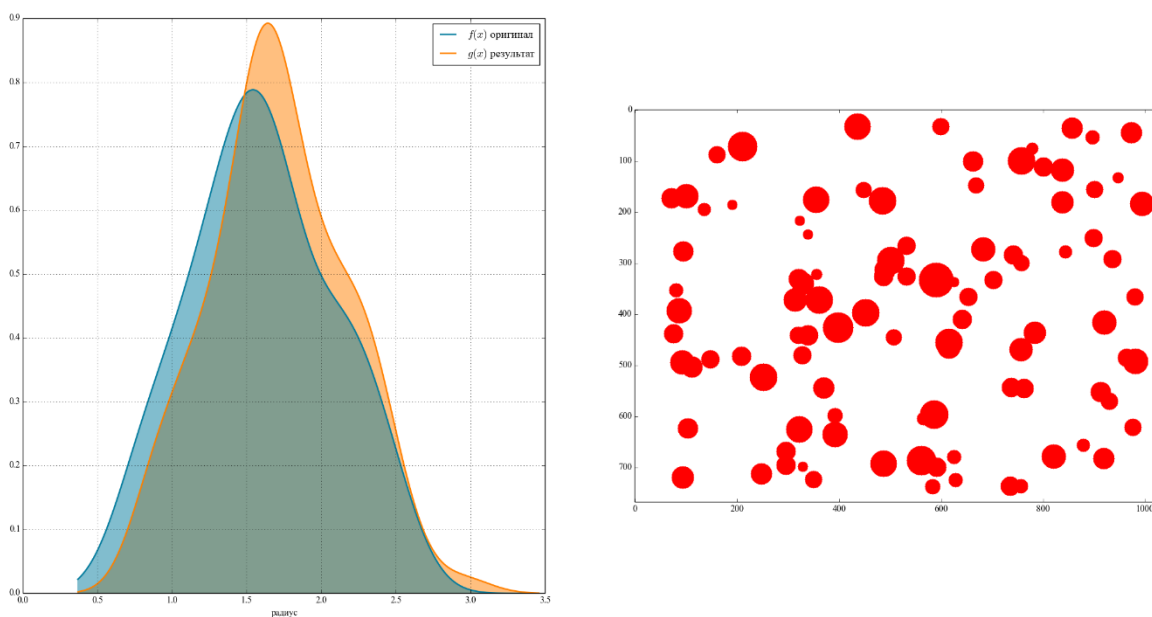


Рисунок 3.4: Проверка работы программы при наличии слипшихся частиц

необходимо получать мелкие (1-5 мкм) частицы эмульсии, это даёт очень хорошую стабильность при хранении и хорошее обволакивание заполнителей. Для получения таких эмульсий необходимо специализированное дорогостоящее оборудование.

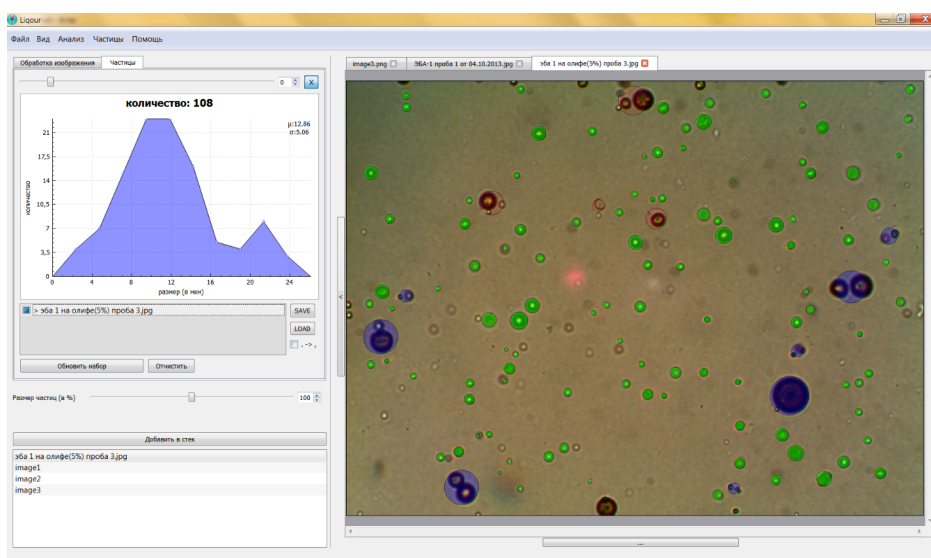


Рисунок 3.5: ЭБА-1 на битуме разжиженном олифой (5%)

Стандартные диспергаторы на которых дорожники производят битумные эмульсии позволяют получать средний размер частиц примерно 10 -20 мкм. Одним из возможных способов уменьшения размеров частиц эмульсии может являться понижение вязкости битума.

### 3.2.5 Результаты

Разработанный программный комплекс, показал что изложенные выше алгоритмы отлично справляются с задачей классификации объектов по контуру ...

Таблица 3.3: Распределение частиц на рис.3.5

| Среднее<br>значение,<br>мкм. | Ср.кв.<br>отклонение,<br>мкм. | Всего<br>частиц |
|------------------------------|-------------------------------|-----------------|
| 12.86                        | 5.06                          | 108             |

### 3.3 Автоматизация составления ПОДД

# Заключение

Основные результаты работы заключаются в следующем.

1. На основе анализа ...
2. Численные исследования показали, что ...
3. Математическое моделирование показало ...
4. Для выполнения поставленных задач был создан ...

И какая-нибудь заключающая фраза.

## Список рисунков

|     |   |    |
|-----|---|----|
| 2.1 | Схема разбиения графа «особыми» точками . . . . .                       | 21 |
| 2.2 | Дерево изображений . . . . .  | 21 |
| 3.1 | Скриншоты программы «анализ изображений» . . . . .                      | 24 |
| 3.2 | Концентрированная битумная анионная эмульсия в отраженном свете . . . . | 24 |
| 3.3 | Результат работы программы на искусственно сгенерированных данных . . . | 25 |
| 3.4 | Проверка работы программы при наличии слипшихся частиц . . . . .        | 26 |
| 3.5 | ЭБА-1 на битуме разжиженном олифой (5%) . . . . .                       | 26 |

## Список таблиц

|     |   |    |
|-----|---|----|
| 3.1 | Анализ распределения частиц на рис. 3.3 . . . . . | 24 |
| 3.2 | Распределение частиц на рис. 3.4 . . . . .        | 25 |
| 3.3 | Распределение частиц на рис.3.5 . . . . .         | 27 |



# Литература

1. И. Мартьянов В., М.Д. Каташевцев. Комбинаторные задачи высокой сложности и анализ плоских контурных изображений // Известия Иркутского Государственного Университета серия «Математика». 2013. С. 31–47.
2. М.Д. Каташевцев. Анализ плоских контурных изображений с метрикой // Известия Иркутского Государственного Университета серия «Математика». 2014. С. в печати.
3. М.Д. Каташевцев. Волновая скелетизация // Вестник Иркутского Государственного Технического Университета. 2013. С. 89–92.
4. Название буклета.
5. “This is english article” / Author1, Author2, Author3 et al. // Journal. 2012. Vol. 2. P. 200.

Приложение А

Структура БД