

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ М. В. ЛОМОНОСОВА
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ

На правах рукописи

ДЫШКАНТ НАТАЛЬЯ ФЕДОРОВНА

**ЭФФЕКТИВНЫЕ АЛГОРИТМЫ СРАВНЕНИЯ
ПОВЕРХНОСТЕЙ, ЗАДАНЫХ ОБЛАКАМИ ТОЧЕК**

01.01.09 – дискретная математика и математическая кибернетика

Диссертация на соискание учёной степени
кандидата физико-математических наук

Научный руководитель
доктор технических наук,
профессор Л. М. Местецкий

Москва – 2011

Оглавление

Введение	5
Глава 1. Модели поверхностей и методы их сравнения	14
1.1. Задача сравнения поверхностей	14
1.1.1. Представление объекта облаком точек	14
1.1.2. Основные определения	16
1.1.3. Общая постановка задачи сравнения поверхностей	19
1.2. Способы задания поверхностей	20
1.2.1. Сетки регулярной структуры	20
1.2.2. Сетки нерегулярной структуры	21
1.3. Обзор методов сравнения поверхностей	22
1.4. Основные выводы	28
Глава 2. Меры сравнения поверхностей и алгоритмы их вычисления	30
2.1. Постановка задачи сравнения поверхностей	30
2.1.1. Основные определения	30
2.1.2. Математическая постановка задачи	32
2.1.3. Меры сравнения поверхностей	33
2.2. Алгоритм сравнения поверхностей объектов	39
2.2.1. Структура алгоритма	39
2.2.2. Построение триангуляций Делоне	40
2.2.3. Локализация триангуляций друг в друге	41
2.2.4. Интерполяция поверхностей	60
2.2.5. Объединённая триангуляция Делоне	63
2.2.6. Сравнение поверхностей на треугольной области	64

2.2.7. Оценка вычислительной сложности алгоритма	69
2.2.8. Программная реализация	71
2.2.9. Результаты экспериментов	76
2.3. Алгоритм выделения интерфейсных граней	78
2.3.1. Поиск начального стартера	85
2.3.2. Прослеживание цепочки интерфейсных граней	86
2.3.3. Поиск стартеров	88
2.3.4. Оценка вычислительной сложности алгоритма	89
2.3.5. Эксперименты	94
2.4. Подгонка поверхностей	95
2.4.1. Оптимизационная постановка задачи	96
2.4.2. Минимизация функционала различия поверхностей	96
2.5. Основные выводы	97

Глава 3. Некоторые задачи анализа поверхностей трёхмерных моделей человеческих лиц	100
3.1. Задача оценки асимметрии лица по 3d модели	101
3.1.1. Описание базы моделей лиц	104
3.1.2. Постановка задачи	105
3.1.3. Метод оценки асимметрии модели	106
3.1.4. Локальная асимметрия модели	109
3.1.5. Эксперименты	110
3.2. Построение совместной пространственной модели лица и зубов для задач ортодонтии	114
3.3. Задача оценки динамики движения нижней челюсти по трёх- мерной видеопоследовательности	115
3.3.1. Постановка задачи	116

3.3.2. Метод сегментации модели на статические и динамиче- ские области	117
3.3.3. Подгонка статических частей моделей	120
3.3.4. Эксперименты	121
3.4. Основные выводы	124
Заключение	126
Литература	127

Введение

Диссертационная работа посвящена исследованию и разработке математических методов сравнения поверхностей объектов, заданных облаками точек в трёхмерном пространстве. Вводятся меры для сравнения поверхностей, предлагаются методы, позволяющие эффективно вычислять такие меры.

Актуальность темы. Трёхмерные цифровые модели поверхностей объектов в настоящее время находят широкое применение в самых разных областях: в медицине, компьютерной графике, архитектуре, дизайне. На стыке компьютерного зрения и других областей (например, геоинформатики, медицины) возникают задачи, ориентированные на анализ и обработку моделей поверхностей, полученных трёхмерным сканированием объектов реального мира. При решении прикладных задач анализа, обработки и классификации моделей поверхностей, восстановления общей поверхности по съёмке её отдельных фрагментов необходимо сравнивать поверхности между собой. Вследствие быстрого развития технологий трёхмерного сканирования объектов возникают всё новые задачи и возможности использования этих технологий в различных приложениях и областях. В связи с этим требуется разработка эффективных методов для решения задач анализа и классификации полученных поверхностей, включающих построение мер сходства поверхностей и создание эффективных алгоритмов, способных работать в реальном масштабе времени.

Обычно для цифрового представления сложных негладких поверхностей применяется метод поточечного описания, когда поверхность задаётся дискретным облаком точек. Такое описание поверхностей можно получить, используя трёхмерный сканер, дигитайзер, топографическую съёмку местности, а также при помощи различного программного обеспечения и медицинского оборудования. Каждый снимок поверхности объекта, полученный при

помощи сканирования, является дискретной моделью однозначной поверхности (так называемой 2.5d поверхностью), так как он содержит информацию только о тех точках объекта, которые видны с позиции съёмки. Такие поверхности можно рассматривать как функции высот, определённые в картинной плоскости (ортогональной к направлению линий визирования) — в узлах некоторой сетки.

Задача сравнения поверхностей в общем виде состоит в том, чтобы для двух или более поверхностей оценить сходство этих поверхностей, либо их фрагментов. Сами поверхности представлены облаками точек, полученными в результате разных актов сканирования, с помощью разнотипных сканеров и т. п. При этом интерес представляет оценка сходства поверхностей при таком их взаимном расположении, когда они в максимальной степени близки друг к другу. Решение задачи предполагает, во-первых, оценку меры близости двух поверхностей в фиксированном заданном положении, а во-вторых, поиск такого их положения, при котором эта мера близости будет наибольшей. Процесс поиска наилучшего совпадения поверхностей в диссертации называется подгонкой.

Известно много работ, посвящённых решению этой задачи. Рассматриваемые в них подходы можно отнести к двум типам. Первый тип подходов состоит в вычислении меры близости поверхностей, представленных пространственными облаками точек, на основе попарных расстояний между точками. Для двух облаков из n_1 и n_2 точек при m итерациях подгонки вычислительная сложность такого подхода составляет $O(m n_1 \log n_2)$. При реальных размерностях задачи, когда число точек составляет $10^3 - 10^5$, такие алгоритмы не могут использоваться в реальном времени работы систем машинного зрения.

Второй тип подходов использует тот факт, что сравниваются однозначные поверхности. Это позволяет свести задачу к сравнению функций двух

переменных, заданных на дискретных множествах точек в картинной плоскости. Основная сложность этого подхода состоит в том, что функции заданы на разных, причём нерегулярных дискретных множествах, а для сравнения желательно было бы сравнивать значения функций в одних и тех же точках. Обычно эта сложность преодолевается путём вычисления значений функций в узлах регулярной квадратной решётки. Такой пересчёт осуществляется на основе интерполяции функций по их значениям в точках заданных дискретных нерегулярных сеток. Для сохранения точности описания функций размер ячейки регулярной квадратной решётки нужно выбирать достаточно малым, соизмеримым с минимальным расстоянием между точками в исходных нерегулярных сетках. Это приводит к тому, что количество узлов в регулярной сетке становится существенно большим, чем в исходных сетках. Особенно это заметно в тех случаях, когда сравниваются поверхности, заданные на сетках разной плотности, что часто имеет место на практике при использовании сканеров разного типа. К тому же пересчёт необходимо производить на каждой итерации подгонки. Всё это приводит также к неприемлемо большим затратам времени для решения задачи.

Таким образом, существующие подходы к решению рассматриваемой задачи имеют очень высокую алгоритмическую сложность, что препятствует их использованию во многих приложениях. Это обстоятельство определяет актуальность темы диссертации.

Цель диссертационной работы — исследование и разработка новых алгоритмов сравнения и анализа дискретных моделей однозначных поверхностей, обладающих высокой вычислительной эффективностью и не требующих пересчёта нерегулярных сеток в общую регулярную.

Предлагаемый подход к решению задачи основывается на следующих принципах:

- сравниваемые поверхности рассматриваются как кусочно-линейные функ-

ции на триангуляциях Делоне, построенных на проекциях облаков точек на картинную плоскость;

- для сравнения функций вычисляются их значения в узлах сеток друг друга на основе линейной интерполяции;
- мера близости поверхностей определяется на основе сравнения значений функций по объединённой сетке, составленной из исходных сеток сравниваемых функций.

Для интерполяции функций и последующего их сравнения в случае нерегулярных сеток необходимо решить задачу локализации узлов сеток друг в друге. Эта задача относится к классу задач геометрического поиска. Существующие методы решения этой задачи имеют достаточно высокую вычислительную сложность.

Научная задача состоит в разработке эффективных вычислительных алгоритмов, позволяющих реализовать предложенный подход в реальном времени работы систем машинного зрения. Задача состоит в том, чтобы обеспечить однократное вычисление меры близости двух поверхностей за время $O(n_1 + n_2)$, а при подгонке с m итерациями — за время $O(m(n_1 + n_2))$.

Для обоснования **реализуемости** предлагаемого решения и **достоверности полученных результатов** в диссертации рассматривается приложение разработанных алгоритмов к решению задач анализа трёхмерных моделей человеческих лиц:

- оценка асимметрии 3d модели человеческого лица;
- построение совместной пространственной модели лица и челюстей для задач ортодонтии;
- сегментация 3d модели лица на статические и динамические области по трёхмерному видеоряду.

Методы исследования. В работе использованы методы теории графов,

минимизации функций, вычислительной геометрии, теории сложности алгоритмов и вычислений. Работа носит теоретико-экспериментальный характер. Проведены эксперименты на модельных данных и дискретных моделях поверхностей реальных объектов, полученных методами трёхмерного сканирования. Также исследованы приложения предлагаемого подхода к задачам анализа моделей лиц.

На защиту выносятся следующие новые научные результаты:

1. Меры сравнения однозначных поверхностей, заданных на разных нерегулярных множествах узлов, основанные на интерполяции и подгонке кусочно-линейных функций на триангуляциях Делоне, и методы вычисления этих мер. В основе решения лежат разработанные в диссертации оригинальные методы локализации триангуляций Делоне друг в друге.
2. Эффективный в среднем $O(n_1 + n_2)$ алгоритм локализации узлов двух триангуляций Делоне друг в друге, основанный на построении и обходе минимальных остовных деревьев триангуляций.
3. Эффективный в худшем случае $O(n_1 + n_2)$ алгоритм прослеживания цепочек интерфейсных граней и локализации в них узлов сеток при объединении перекрывающихся триангуляций Делоне.
4. Метод оценки асимметрии 3d модели человеческого лица на основе сравнения исходной и отражённой моделей поверхности лица и поиска оптимальной плоскости симметрии.
5. Метод сегментации модели поверхности лица на статические и динамические области по трёхмерной видеопоследовательности.

Научная значимость работы состоит в разработке методов вычисления мер для сравнения поверхностей объектов, а также эффективных алгоритмов решения задачи геометрического поиска при локализации одной триан-

гуляции Делоне в другой. Предложен подход, позволяющий производить операции над функциями, заданными на разных нерегулярных сетках. Изложенная в работе методика даёт эффективный математический аппарат для конструирования общих и прикладных мер для сравнения поверхностей объектов.

Практическая значимость состоит в разработке эффективных алгоритмов сравнения поверхностей, позволяющих существенно расширить круг решаемых задач, в частности, в системах машинного зрения, требующих работы в режиме реального времени. Результаты работы могут найти применение в медицине, геоинформатике, биометрической идентификации.

Апробация работы. Результаты работы докладывались и обсуждались на следующих научных конференциях и семинарах:

- всероссийская конференция «Математические методы распознавания образов» ММРО–13 (Зеленогорск, 2007 год) [7];
- XV международная научная конференция студентов, аспирантов и молодых учёных «Ломоносов–2008» (Москва, 2008 год) [8];
- 7-я международная конференция «Интеллектуализация обработки информации» ИОИ’08 (Алушта, Украина, 2008 год) [9];
- 18я международная конференция по компьютерной графике и машинному зрению «ГрафиКон’08», (Москва, 2008 год) [14];
- 4я международная конференция «Машинное зрение: теория и приложения» VISAPP–2009 (Лиссабон, Португалия, 2009 год) [50];
- 2-ой международный семинар «Извлечение знаний из изображений. Теория и приложения» IMTA–2009 (Лиссабон, Португалия, 2009 год) [49];
- XVI международная научная конференция студентов, аспирантов и молодых учёных «Ломоносов–2009» (Москва, 2009 год) [12];

- 19я международная конференция по компьютерной графике и машинному зрению «ГрафиКон'09» (Москва, 2009 год) [3];
- всероссийская конференция «Математические методы распознавания образов» ММРО-14 (Суздаль, 2009 год) [4];
- научные семинары по совместному российско-индийскому проекту «Пространственное моделирование человеческих лиц для анализа и классификации в реальном времени» (МГУ, Москва, сентябрь 2009 года; Мангалорский университет, Мангалор, Индия, декабрь 2009 года; МГУ, Москва, октябрь 2010 года; Мангалорский университет, Мангалор, Индия, январь 2011 года);
- 8-я международная конференция «Интеллектуализация обработки информации» ИОИ'10 (Пафос, Республика Кипр, 2010 год) [15];
- научный спецсеминар «Дискретно-непрерывные преобразования изображений в задачах распознавания» под руководством д. т. н., профессора Л. М. Местецкого, (факультет ВМК МГУ, Москва, 2010 год);
- 2-я научно-техническая конференция «Техническое зрение в системах управления» TVCS 2011 (ИКИ РАН, Москва, 2011 год) [13].
- 16-я международная конференция Международной ассоциации по распознаванию образов (IAPR) «Дискретная геометрия для компьютерной обработки изображений» DGCI-2011 (Нанси, Франция, 2011 год) [51].

Описания отдельных результатов работы включены в отчёты по проектам РФФИ №№08-01-00670-а, 08-07-00305-а, 09-07-92652-ИНД_а, 10-07-00609-а.

Личный вклад. Все результаты, выносимые на защиту, получены автором самостоятельно. Постановка задачи была выполнена совместно с научным руководителем. В совместных публикациях в трудах конференций [3, 4] автору

принадлежат разработанные методы сегментации 3d модели лица на статические и динамические области. В совместно опубликованных работах [7, 9, 10, 14, 50] автору принадлежат модели и методы решения задач.

Публикации. Материалы диссертации опубликованы в 15 работах [3, 4, 7–15, 48–51], из них 2 работы [48, 51], включённые в Перечень ведущих рецензируемых научных журналов и изданий, 1 статья в журнале [10], 5 статей в сборниках трудов международных научных конференций и семинаров [3, 14, 15, 49, 50], 2 статьи в сборниках трудов всероссийских научных конференций [4, 7] и 5 тезисов докладов [8, 9, 11–13].

Структура и объём диссертации. Работа состоит из оглавления, введения, трёх глав, заключения и списка литературы. Содержание работы изложено на 139 страницах. Список литературы включает 97 наименований. Текст работы иллюстрируется 58 рисунками и 9 таблицами.

В первой главе содержится общая постановка задачи сравнения поверхностей; рассматриваются способы задания дискретных моделей поверхностей; приводится обзор и анализ литературных источников по теме сравнения поверхностей объектов, заданных облаками точек; формулируются основные задачи исследования.

Во второй главе вводятся основные определения и обозначения; формулируется математическая постановка задачи сравнения поверхностей; вводятся новые меры для сравнения таких поверхностей; описываются новые разработанные алгоритмы для вычисления предложенных мер; приводятся и доказываются оценки вычислительной сложности работы алгоритмов; описываются детали разработанного комплекса программ, реализующего предложенные алгоритмы; приводятся результаты вычислительных экспериментов для различных этапов предложенных алгоритмов.

В третьей главе рассматриваются приложения разработанных методов к задачам анализа трёхмерных портретов человеческих лиц: задаче оценки

асимметрии лица по 3d модели, задаче построения совместной пространственной модели лица и зубов для приложений в ортодонтии, задаче сегментации 3d модели лица на статические и динамические области по трёхмерному видеоряду. Приводятся результаты вычислительных экспериментов, проведённых на собранных базах моделей.

В заключении подводятся итоги работы.

Глава 1

Модели поверхностей и методы их сравнения

1.1. Задача сравнения поверхностей

Данные, полученные от трёхмерного сканера после сканирования поверхности объекта¹, обычно представляют собой дискретное множество точек с трёхмерными координатами — *дискретную модель* поверхности. Для такого множества не заданы в явном виде понятия связности, топологии или непрерывной поверхности. На основе полученного множества точек можно перейти к триангуляционной кусочно-линейной модели поверхности [37], которая уже будет являться *непрерывной моделью*.

1.1.1. Представление объекта облаком точек

Обозначим через \mathbb{E}^d d -мерное евклидово пространство.

Определение 1.1. *Облаком точек называется любой набор точек в пространстве E^d или набор проекций этих точек в подпространстве более низкой размерности.*

Если данные об объекте представлены наборами точек, лежащих на его поверхности, будем говорить, что *поверхность объекта задана облаком точек*.

¹ Существует объективная ошибка точности описания поверхности дискретным множеством точек, которой в работе приходится пренебречь.

Будем называть поверхность S *однозначной*, если существует такая ось a , что любая прямая, параллельная этой оси, пересекает поверхность не более чем в одной точке. Таким образом, однозначная поверхность не имеет самопересечений (см. рис. 1.1). Также такие поверхности называют 2,5-мерными (2,5d поверхности). Их можно рассматривать как функции высот, определённые на множествах точек в плоскости изображения, которая перпендикулярна оси a .

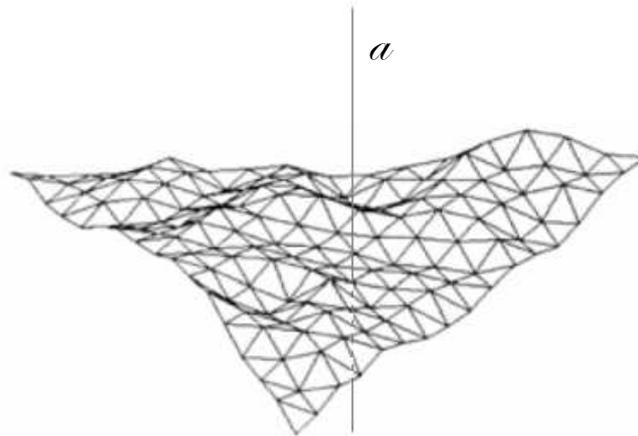


Рис. 1.1. Пример однозначной поверхности.

При съёмке объекта одной камерой можно получить только однозначную поверхность. Две камеры позволяют получить по точкам перекрытия, которые видны обеим камерам, восстановленную поверхность. Таким образом, восстановленная модель объекта состоит из подмножеств точек однозначных (относительно осей этих камер) поверхностей. Многие современные сканеры имеют встроенное программное обеспечение, позволяющее составлять целостную трёхмерную модель объекта, используя отдельные снимки его поверхностей, снятые с разных сторон. Это может быть осуществлено с помощью начального позиционирования сканирующей системы и получения на его основе информации о положении отдельных частей объекта в общей глобальной системе координат, нанесения опорных меток на объект [90], по-

иска общих областей перекрытия для фрагментов [1, 36, 41, 97]. Но изначально получаемые при сканировании поверхности являются однозначными, каждый снимок содержит информацию только о тех точках поверхности объекта, которые видны с позиции наблюдения.

Если построить проекцию исходного облака точек, задающего поверхность, на плоскость, перпендикулярную оси визирования, то можно получить *плоское облако точек*, для каждой из которых будет известна *высота* — расстояние от этой плоскости до соответствующей точки исходного облака.

1.1.2. Основные определения

Понятие триангуляции тесно связано с теорией графов. Далее приведены основные необходимые определения [2, 28, 33].

Определение 1.2. Множество $V = \{v_1, v_2, \dots\}$ и набор E неупорядоченных пар объектов (v_i, v_j) из V называется *графом* $G = (V, E)$. Элементы множества V называются *вершинами* (узлами) графа, а пары из E — его *рёбрами*.

Если v_i, v_j — вершины графа, $e = (v_i, v_j)$ — соединяющее их ребро, то каждая из вершин v_i, v_j и ребро e *инцидентны*. Две вершины, инцидентные одному и тому же ребру, называются *смежными* вершинами; два ребра, инцидентные одной и той же вершине, называются *смежными* рёбрами.

Определение 1.3. Граф $G_1 = (V_1, E_1)$ называется *подграфом* графа $G = (V, E)$, если $V_1 \subseteq V$ и $E_1 \subseteq E$. Обозначение: $G_1 \subseteq G$.

Определение 1.4. Последовательность рёбер графа $G = (V, E)$

$$E_{v_i v_j} = \{(v_{i_1}, v_{i_2}), (v_{i_2}, v_{i_3}), \dots, (v_{i_{k-1}}, v_{i_k})\},$$

где $v_{i_1} = v_i, v_{i_k} = v_j$, называется *путём*, соединяющим вершины v_i и v_j .

Определение 1.5. Граф $G = (V, E)$ называется *связным*, если для любых двух различных вершин v_i и v_j из V существует путь, соединяющий эти вершины.

Определение 1.6. Путь $E_{v_i v_j}$, не проходящий дважды через одно ребро, называется *циклом*, если $v_i = v_j$.

Определение 1.7. Связный граф, не содержащий циклов, называется *деревом*.

Определение 1.8. Граф $G = (V, E)$ называется *полным*, если для любых двух вершин v_i, v_j из V в E есть ребро (v_i, v_j) , соединяющее эти вершины.

Определение 1.9. Пусть задан граф $G = (V, E)$. Пусть каждой вершине v_i из V сопоставлена точка a_i в некотором евклидовом пространстве, причём $a_i \neq a_j$ при $i \neq j$. Пусть каждому ребру $e = (v_i, v_j)$ из E сопоставлена непрерывная кривая L , соединяющая точки a_i и a_j и не проходящая через другие точки a_k . Тогда если все кривые, сопоставленные рёбрам графа, не имеют общих точек, кроме концевых, то это множество точек и кривых называется *геометрической реализацией графа G* .

Определение 1.10. Граф называется *планарным*, если существует его геометрическая реализация на плоскости².

Планарный граф, уложенный на плоскости, называется *плоским*. Плоская укладка планарного графа $G = (V, E)$ — это отображение каждой вершины из V в точку на плоскости, а каждого ребра из E в простую линию, соединяющую пару образов концевых вершин этого ребра так, что образы рёбер пересекаются только в своих концевых точках [28]. Известно, что любой планарный граф всегда можно уложить на плоскости так, что все рёбра

² Без пересечения рёбер.

будут отображены в прямолинейные отрезки [54]. Следуя [28], будем называть графы, уложенные таким образом, плоскими *прямолинейными* графами. Прямолинейная укладка рёбер планарного графа определяет разбиение плоскости, называемое *планарным подразбиением*.

Определение 1.11. Область D , принадлежащая пространству \mathbb{E}^d , называется *выпуклой*, если для любой пары точек q_1 и q_2 из D прямолинейный отрезок $\overline{q_1q_2}$ целиком принадлежит D .

Определение 1.12. *Выпуклой оболочкой* $\text{Conv}(S)$ множества точек S , принадлежащих пространству \mathbb{E}^d , называется наименьшая выпуклая область в \mathbb{E}^d , которая содержит S .

Определение 1.13. Планарное подразбиение называется *триангуляцией*, если все его конечные грани являются треугольниками.

Определение 1.14. *Триангуляцией* $T(S)$ конечного множества точек S называется плоский граф, имеющий наибольшее возможное число рёбер (триангуляция $T(S)$ получена путём соединения точек из S непересекающимися прямолинейными отрезками так, что любая грань, лежащая внутри выпуклой оболочки $\text{Conv}(S)$, является треугольником).

В дальнейшем под термином *грань* будем понимать только конечную треугольную грань триангуляции.

Определение 1.15. Триангуляция $T(S)$ удовлетворяет *условию Делоне*, если внутри окружности, описанной вокруг любого построенного треугольника, нет ни одной точки из S . Такая триангуляция называется *триангуляцией Делоне* $DT(S)$ ³.

³ DT — Delaunay Triangulation.

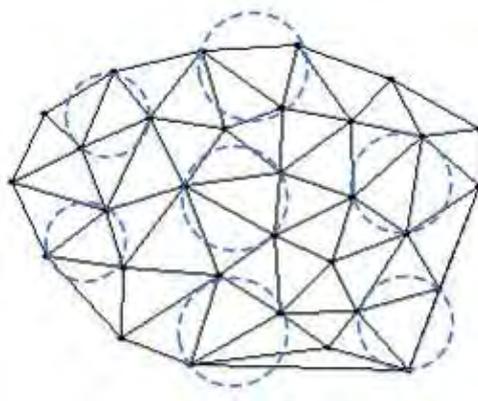


Рис. 1.2. Триангуляция Делоне для множества точек на плоскости.

Определение 1.16. *Двумерной (плоской) сеткой g мощности N будем называть набор взаимосвязанных геометрических элементов (узлы, рёбра, грани), узлы которого представляют собой конечное множество из N точек пространства \mathbb{E}^2 :*

$$g = \{g^i\}_{i=1}^N = \{(x^i, y^i) \in \mathbb{E}^2 \mid i = 1, \dots, N\}. \quad (1.1)$$

Множество точек на плоскости и построенную на нём триангуляцию будем называть *триангуляционной плоской сеткой*.

1.1.3. Общая постановка задачи сравнения поверхностей

Для однозначной поверхности при переходе от исходного облака точек к триангуляционной модели можно строить триангуляцию Делоне на плоском облаке точек. Тогда поверхность будет задана высотами в узлах триангуляционной плоской сетки и может быть рассмотрена как множество соответствующих пространственных треугольников, составляющих кусочно-линейное представление поверхности (см. рис. 1.3).

Общая постановка рассматриваемой задачи заключается в следующем: даны две однозначные поверхности, заданные облаками точек, требуется сравнить (вычислить некоторую меру сходства/различия) между этими поверхностями. Качество алгоритма сравнения обычно зависит от соотношения между

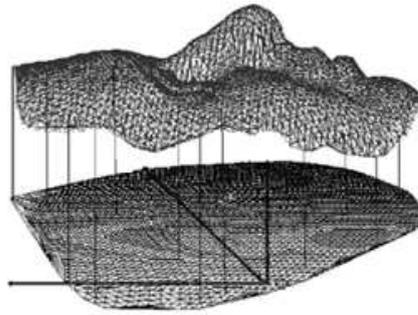


Рис. 1.3. Однозначная поверхность как функция высот, заданная в узлах плоской сетки.

точностью полученной оценки сходства и его вычислительной сложностью.

1.2. Способы задания поверхностей

Для моделирования однозначных поверхностей используются два основных способа представления: задание поверхности на регулярной (структурированной) и нерегулярной (хаотической) сетках (см. рис. 1.4). Оба способа имеют свои преимущества и недостатки [35, 89].

1.2.1. Сетки регулярной структуры

Будем называть двумерную сетку *регулярной*, если её грани представляют собой равные между собой фигуры. Узлы регулярной сетки имеют определённую упорядоченную структуру, каждый узел имеет координаты $(i \cdot dx, j \cdot dy)$, где $dx, dy \in \mathbb{R}$ представляют шаг сетки.

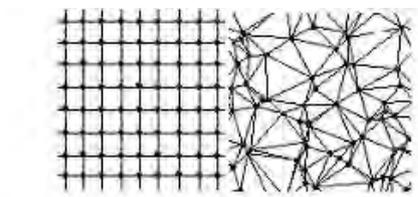


Рис. 1.4. Сетки с регулярной (слева) и нерегулярной структурой (справа).

Регулярные сетки просты в использовании, для их хранения не требуется большого объёма памяти, так как координаты её узлов можно вычислять,

а не хранить в явном виде. Однако при использовании регулярных сеток возникает проблема выбора оптимального размера ячейки (границы) для достижения приемлемой в той или иной задаче точности аппроксимации поверхности. Шаг сетки приходится выбирать в соответствии с элементами поверхности с наибольшим градиентом, то есть где требуется мелкий шаг. В этом случае такой же мелкий шаг устанавливается и для областей с небольшим градиентом, где для обеспечения хорошей точности было бы достаточно более крупного шага. Мелкий шаг в таких областях приводит к увеличению объёма вычислений. Таким образом, проблема выбора размера ячейки регулярной сетки приводит к значительной избыточности описания поверхности объекта, которая влечёт вычислительную и ёмкостную неэффективность.

Использование регулярных сеток для сравнения поверхностей предполагает, что в каждом узле общей сетки известны значения высот для обеих поверхностей, при этом нет необходимости использовать методы интерполяции для пересчёта высот каждой из поверхностей в промежуточных точках. Однако при поиске наилучшего совпадения поверхностей (т. е. при решении задачи подгонки) регулярные сетки требуют такого пересчёта, так как объединение исходной регулярной сетки и её трансформации, полученной сдвигами и поворотами в системе координат, не всегда будет регулярной сеткой.

1.2.2. Сетки нерегулярной структуры

Сетку, не являющуюся регулярной, будем называть *нерегулярной*. В нерегулярной сетке точки могут располагаться произвольно, то есть регулярная упорядоченная структура отсутствует. Обычно при моделировании поверхностей с помощью нерегулярных сеток используют триангуляционные сетки Делоне.

Поверхности объектов, полученные методами трёхмерного сканирова-

ния⁴, заданы на нерегулярных сетках.

Значимым преимуществом нерегулярных сеток является то, что при их использовании не возникает проблемы избыточности описания поверхности, как при использовании регулярных сеток. Для того, чтобы получить значения высот в узлах общей сетки при сравнении поверхностей, необходимо использовать методы интерполяции для пересчёта высот каждой из поверхностей в промежуточных точках — узлах противоположной сетки. Поэтому использование нерегулярных сеток требует введения и разработки более сложных мер для сравнения поверхностей, заданных на разных сетках, и соответствующих обрабатывающих алгоритмов.

1.3. Обзор методов сравнения поверхностей

Исследования в области сравнения трёхмерных поверхностей ведутся давно, и существует несколько подходов к решению этой задачи.

Многие известные методы предлагают использовать пересчёт исходных нерегулярных сеток в общую регулярную [17, 34, 64, 91] (см. рис. 1.5), после которого могут быть применены подходы к сравнению и подгонке поверхностей на регулярных множествах точек [27, 43].

При таком пересчёте возникает проблема выбора оптимального шага регулярной сетки, которая приводит к значительному объёму вычислений для достижения приемлемой точности аппроксимации поверхности и к неэффективности. В работе предлагается метод сравнения поверхностей с сохранением исходной нерегулярности сеток.

Существуют методы сравнения поверхностей на основе сравнения при-

⁴ В работе под *сканированием* подразумевается получение координат точек поверхности объекта с помощью 3d сканера. Существуют также сканеры компьютерной и магнитно-резонансной томографии, трёхмерное УЗИ, которые производят послойное сканирование, строящее воксельные модели объектов. Такие модели в работе не рассматриваются.

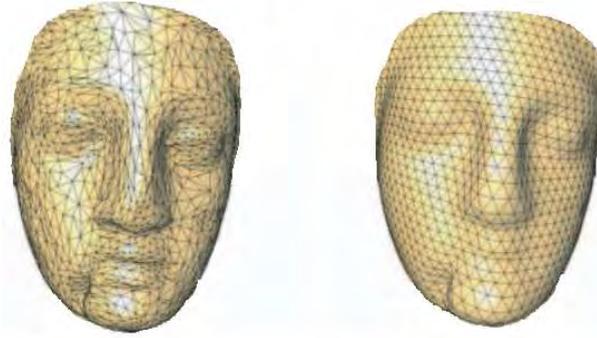


Рис. 1.5. Пересчёт нерегулярной сетки в регулярную.

знаковых описаний: так, в [55] задача сравнения объектов сведена к задаче сравнения графов, вершины которых представляют собой информацию об отдельных фрагментах поверхностей объектов, а рёбра отображают информацию о связности этих фрагментов.

Использование карт кривизны (Curvature Maps) основывается на построении изолиний в окрестностях определённых точек поверхности с последующим сравнением поверхностей по этим картам [59].

В большинстве существующих методов для сравнения двух поверхностей объектов предполагается, что для каждой точки одной поверхности есть соответствующая ей точка второй поверхности [36, 97].

Одним из базовых алгоритмов совмещения поверхностей является *итеративный алгоритм ближайших точек* (iterative closest point — ICP), предложенный в [36, 41, 97]. Алгоритм использует итеративную процедуру минимизации среднего расстояния между двумя облаками точек. При этом требуется начальная оценка грубого преобразования одного облака в другое, которая постепенно уточняется в процессе минимизации. Для двух заданных трёхмерных облаков точек S_1 и S_2 алгоритм находит наилучшее преобразование (трансформацию), переводящее S_1 в S_2 .

Алгоритм ближайших точек может быть использован для совмещения снимков одного и того же объекта, полученных с разных ракурсов, но имеющих общие участки — области перекрытия. При этом предполагается, что

в областях перекрытия находятся пары ближайших точек, расстояние между которыми меньше заданного порога. Если в список таких пар попадают «ошибочные» пары, не принадлежащие области перекрытия, это негативно сказывается на корректности работы алгоритма.

Пусть $\{(s_i^1, s_i^2)\}_{i=1}^N$ — множество пар ближайших точек для S_1 и S_2 . В процессе работы алгоритма минимизируется среднее расстояние между облаками точек S_1 и S_2 :

$$E = \frac{1}{N} \sum_{i=1}^N d(s_i^1, s_i^2) \rightarrow \min, \quad (1.2)$$

где $d(\cdot, \cdot)$ — евклидово расстояние между двумя точками.

Для вычисления расстояния между точкой $s^1 \in S_1$ и облаком точек S_2 в первоначальном варианте алгоритма [36] используется метрика «точка-точка» (point-to-point distance):

$$\rho(s^1, S_2) = \min_{s^2 \in S_2} d(s^2, s^1). \quad (1.3)$$

Метрика (1.3) использует дискретные модели поверхностей.

В [41] предлагается метрика «точка-плоскость» (point-to-plane distance), использование которой предполагает, что в каждой точке S_1 и S_2 известны нормали к поверхностям, заданным облаками точек S_1 и S_2 соответственно. Такая информация может быть задана изначально, либо вычислена через усреднение нормалей по инцидентным треугольникам при использовании триангуляционной кусочно-линейной аппроксимации поверхности. В этом случае будет минимизироваться сумма квадратов расстояний от s_i^1 до плоскости P_i^2 , перпендикулярной S_2 в точке s_i^2 по всем парам ближайших точек (s_i^1, s_i^2) :

$$E = \sum_{i=1}^N H^2(s_i^1, P_i^2) \rightarrow \min, \quad (1.4)$$

где $H(s, P)$ — евклидово расстояние от точки s до плоскости P .

Общая итеративная схема алгоритма включает следующие этапы:

1. Поиск пар ближайших точек (s_i^1, s_i^2) , $i = \overline{1, N}$ для текущего положения S_1 и S_2 .
2. Поиск преобразования (параметров сдвигов и поворотов) облака точек S_1 , уменьшающего ошибку E (1.2) или (1.4) с помощью метода наименьших квадратов. Если изменение ошибки E меньше определённого порогового значения, то работа алгоритма заканчивается.
3. Применение найденного на предыдущем этапе преобразования к облаку точек S_1 . Переход на этап 1.

Этапы 1–3 повторяются до тех пор, пока уменьшение ошибки превышает некоторое пороговое значение. Решением задачи является конечное положение облака точек S_1 .

Влияние «ошибочных» пар ближайших точек при использовании метрики «точка-плоскость» меньше, чем при использовании (1.3). Количество итераций алгоритма при использовании метрики «точка-точка» больше, так как в ней используются дискретные модели поверхностей. Недостатком метрики «точка-плоскость» является то, что она сильно зависит от начального взаимного положения S_1 и S_2 [60].

Основным достоинством алгоритма ближайших точек является его простая реализация, к недостаткам можно отнести сильную зависимость от начального приближения объектов, вычислительную сложность, связанную с поиском всех пар ближайших точек $\{(s_i^1, s_i^2)\}_{i=1}^N$.

Пусть N_1, N_2 — количество точек в исходных облаках точек S_1, S_2 соответственно, тогда при простой реализации оценка трудоёмкости такого поиска будет $O(N_1 N_2)$, т. е. квадратичной при $N_1 \approx N_2$. С помощью использования более сложных структур данных — например, k-d дерева [58] — поиск может быть осуществлён за время $O(N_1 \log N_2)$. Таким образом, общее коли-

чество требуемых операций для поиска пар ближайших точек при m итерациях составляет $O(mN_1 \log N_2)$.

Большое количество работ посвящено различным улучшениям алгоритма ближайших точек, заключающимся в:

- модификации методов выбора области перекрытия и пар ближайших точек (например, с помощью введения жёстких ограничений на класс движений, переводящих одну точку в другую [75], с использованием теории случайных величин [44], k-d деревьев [58, 97], генетических алгоритмов [40]);
- назначении весов парам ближайших точек [61];
- модификации формулы для расстояния между точками [56, 95];
- модификация формулы для минимизируемой ошибки [52];
- модификации метода минимизации ошибки [95].

Недостатком улучшений алгоритма ближайших точек является их сложность, настройка на конкретные экспериментальные данные, уменьшающая устойчивость алгоритмов.

В [88] рассматривалась задача сравнения поверхностей одного и того же объекта, заданных на разных множествах точек, при этом расстояние от точки на одной поверхности до другой поверхности вычислялась вдоль нормалей к сплайнам второй поверхности. Мера сравнения, основанная на вычислении разности расстояний вдоль нормалей к поверхностям, интересна тем, что не требует приведения сравниваемых функций к общей сетке. Однако сложность такого подхода квадратичная.

Такие меры, как, например, среднее расстояние между высотами, которые могут быть напрямую вычислены при задании обеих поверхностей в узлах общей сетки, не допускают обобщения на случай их задания в узлах разных сеток без этапа приведения сеток к регулярным.

В [43] мера для сравнения двух поверхностей S_1 и S_2 , заданных в узлах сеток g_1 и g_2 соответственно, вводилась как максимальное расстояние между

парами ближайших точек разных сеток, т. е.

$$\rho(S_1, S_2) = \max_{g_1[i] \in g_1} \min_{g_2[j] \in g_2} d(g_1[i], g_2[j]),$$

где $d(\cdot, \cdot)$ — евклидово расстояние между двумя точками. Такая мера может быть обобщена на случай разных нерегулярных сеток, но её вычисление будет иметь квадратичную сложность и, кроме того, её физический смысл не очевиден и приемлем в малом количестве приложений.

В [96] предложен алгоритм сопоставления поверхностей, основанный на минимизации меры близости между ними. При этом сравниваемые поверхности S_1 и S_2 рассматриваются как объекты линейного пространства, а мера близости $\rho(S_1, S_2)$ представляет собой норму $\|S_1 - S_2\|$ в этом пространстве. Пусть поверхность X представлена триангуляционной кусочно-линейной моделью из N треугольников, c_i — центроид i -ого треугольника, n_i — вектор нормали к i -ому треугольнику, длина которого равна площади этого треугольника. Норма поверхности X вводится как

$$\|X\| = \sum_{i=1}^N \sum_{j=1}^N (n_i, n_j) e^{-|c_i - c_j|^2 / \sigma^2}. \quad (1.5)$$

Пусть исходные поверхности S_1, S_2 заданы триангуляционными кусочно-линейными моделями из N_1, N_2 треугольников соответственно; $c_i^1(c_i^2)$ — центроид i -ого треугольника поверхностей $S_1(S_2)$, $n_i^1(n_i^2)$ — вектор нормали к i -ому треугольнику поверхности $S_1(S_2)$, длина которого равна площади этого треугольника. Мера близости, предлагаемая в [96], вводится следующим образом:

$$\begin{aligned} \rho(S_1, S_2) &= \sum_{i=1}^{N_1} \sum_{j=1}^{N_1} (n_i^1, n_j^1) e^{-|c_i^1 - c_j^1|^2 / \sigma^2} + \sum_{i=1}^{N_2} \sum_{j=1}^{N_2} (n_i^2, n_j^2) e^{-|c_i^2 - c_j^2|^2 / \sigma^2} - \\ &- 2 \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} (n_i^1, n_j^2) e^{-|c_i^1 - c_j^2|^2 / \sigma^2} = \|S_1 - S_2\|. \end{aligned} \quad (1.6)$$

Далее значение меры (1.6) минимизируется методом наискорейшего спуска. Недостатком такого подхода является квадратичная сложность вычисления меры близости.

1.4. Основные выводы

1. Проведённый обзор существующих методов моделирования однозначных поверхностей показал, что существует два основных способа представления таких поверхностей: задание в узлах регулярной и нерегулярной сеток; оба способа имеют свои преимущества и недостатки. Способ, использующий нерегулярные сетки допускает возможность адаптации к требуемой точности описания поверхности, не вносит в исходные данные избыточность, приводящую к большому перерасходу вычислительных ресурсов.
2. Обычно при сравнении двух поверхностей предполагается, что для каждой точки одной поверхности есть соответствующая ей точка второй поверхности. Такое предположение подразумевает задание поверхностей на одном и том же дискретном множестве точек. Исходные поверхности объектов, полученных трёхмерным сканированием, имеют нерегулярную структуру. При пересчёте исходных нерегулярных данных в регулярную сетку возникает проблема выбора оптимального размера для ячейки регулярной сетки, которая приводит к неэффективности такого подхода.
3. Многие существующие меры для сравнения поверхностей могут быть вычислены напрямую при условии задания обеих поверхностей в узлах некоторой общей сетки. Однако такие меры не допускают обобщения для случая задания поверхностей в узлах разных сеток без этапа при-

ведения сеток к регулярным. Существуют меры, которые легко могут быть обобщены на случай разных исходных сеток, но их вычисление в этом случае имеет квадратичную сложность.

4. Известные методы для сопоставления поверхностей можно разделить на два класса:
 - а. Подгонка на основании вычисления расстояний между точками в трёхмерном пространстве, которая имеет большую вычислительную сложность.
 - б. Пересчёт исходных данных в двумерные регулярные сетки, при котором возникает избыточность, приводящая также к существенному повышению вычислительной сложности.
5. Задача разработки новых алгоритмов сравнения поверхностей, представленных функциями высот на разных нерегулярных сетках, сохраняющих исходную нерегулярность сеток и обладающих высокой вычислительной эффективностью, является актуальной.

Глава 2

Меры сравнения поверхностей и алгоритмы их вычисления

Поверхность, полученная методами трёхмерного сканирования, описывается в виде облака точек, которое можно рассматривать как однозначную функцию двух переменных, заданную на некотором дискретном множестве.

2.1. Постановка задачи сравнения поверхностей

Для трёхмерной однозначной поверхности, заданной облаком S из N точек, существует система координат $Oxyz$ такая, что облако точек $S = \{(x^i, y^i, z^i)\}_{i=1}^N$ в ней можно рассматривать как функцию $z = f(x, y)$, заданную на дискретном множестве $\{(x^i, y^i)\}_{i=1}^N$. На этом множестве на плоскости строится триангуляция. *Триангуляционная модель поверхности* состоит из пространственных треугольников, определяемых значениями z^i в узлах этой триангуляции (x^i, y^i) . Такая кусочно-линейная модель аппроксимирует исходную поверхность.

2.1.1. Основные определения

Определение 2.1. *Окружность называется пустой относительно множества точек S , если она не содержит внутри себя точек из S .*

Описанная окружность любой грани триангуляции Делоне $DT(S)$ является пустой относительно S .

Окружность, проходящая через точку, называется *инцидентной* этой точке.

Максимальной пустой окружностью, инцидентной точке, будем называть пустую окружность, инцидентную этой точке и не содержащуюся в другой пустой окружности.

Определение 2.2. Пара точек из множества S , имеющих пустую относительно S инцидентную окружность, образует ребро Делоне. Грань триангуляции, для вершин которой существует общая инцидентная пустая окружность, называется *гранью Делоне*.

Непосредственно из определения триангуляции Делоне 1.15 и определения 2.2 следует, что: 1) Триангуляция $T(S)$, все грани которой являются гранями Делоне, является триангуляцией Делоне $DT(S)$; 2) Триангуляция $T(S)$, все рёбра которой являются рёбрами Делоне, является триангуляцией Делоне $DT(S)$.

Определение 2.3. Два треугольника триангуляции будем называть *смежными*, если они имеют общую сторону.

Определение 2.4. Пучком узла триангуляции будем называть множество всех рёбер триангуляции, инцидентных данному узлу.

Определение 2.5. *Остовным* (или *покрывающим*) деревом связного графа $G = (V, E)$ называется его связный подграф, являющийся деревом и имеющий то же множество вершин V .

Определение 2.6. Граф $G = (V, E)$ называется *взвешенным*, если каждому ребру e_i из E поставлено в соответствие некоторое число w_i , называемое *весом* ребра.

Определение 2.7. *Весом* W_G взвешенного связного графа $G = (V, E)$ называется суммарный вес всех его рёбер: $W_G = \sum_{i=1}^{|E|} w_i$.

Определение 2.8. *Минимальным остовным деревом взвешенного связного графа $G = (V, E)$ называется его остовное дерево, имеющее минимальный вес.*

Определение 2.9. *Евклидовым минимальным остовным деревом (МОД) конечного множества точек S на плоскости $MST(S)$ ¹ называется минимальное остовное дерево взвешенного полного графа с множеством вершин S , каждому из рёбер которого приписан вес, равный (евклидовой) длине этого ребра.*

В дальнейшем в работе под минимальным остовным деревом, или минимальным остовом, будем понимать евклидово минимальное остовное дерево.

Определение 2.10. *Плотностью двумерной сетки g мощности N будем называть величину p_g :*

$$p_g = S_{Conv(g)}/N, \quad (2.1)$$

где $S_{Conv(g)}$ — площадь выпуклой оболочки множества узлов g .

2.1.2. Математическая постановка задачи

Рассмотрим следующую задачу. Пусть две однозначные поверхности S_1 , S_2 заданы облаками точек $\{(x_1^i, y_1^i, z_1^i)\}_{i=1}^{N_1}$, $\{(x_2^i, y_2^i, z_2^i)\}_{i=1}^{N_2}$ соответственно в системе координат $Oxyz$ в \mathbb{E}^3 , так, что S_1 , S_2 однозначно проецируются на плоскость Oxy . Рассмотрим пространство \mathbb{E}^2 и его систему координат Oxy . Пусть g_1 и g_2 — нерегулярные двумерные сетки (1.1), узлы которых есть проекции исходных облаков точек S_1 и S_2 на плоскость Oxy :

$$g_1 = \{(x_1^i, y_1^i)\}_{i=1}^{N_1}, \quad g_2 = \{(x_2^i, y_2^i)\}_{i=1}^{N_2}. \quad (2.2)$$

¹ MST — Minimum Spanning Tree.

Рассмотрим функции f_1 и f_2 , заданные в узлах сеток g_1 и g_2 соответственно так, что:

$$f_1^i = f_1(x_1^i, y_1^i) = z_1^i, \quad f_2^i = f_2(x_2^i, y_2^i) = z_2^i. \quad (2.3)$$

Таким образом, функции заданы наборами значений $\{f_1^i\}_{i=1}^{N_1}$, $\{f_2^i\}_{i=1}^{N_2}$.

Требуется ввести меру для сравнения поверхностей S_1 , S_2 , удовлетворяющую аксиомам полуметрики, и разработать вычислительно эффективный алгоритм, позволяющий её вычислять.

Задачу сравнения поверхностей S_1 , S_2 можно свести к задаче сравнения функций f_1 , f_2 .

2.1.3. Меры сравнения поверхностей

Пусть G — множество нерегулярных двумерных сеток, содержащихся внутри некоторого *объемлющего* прямоугольника R , а F — множество однозначных функций, заданных на сетках из множества G .

Введём меры для сравнения функций на множестве F .

Будем говорить, что функция \hat{f} *аппроксимирует* f на множестве узлов сетки $g \in G$, если $\hat{f} \equiv f$ в узлах g .

Пусть $f_1, f_2 \in F$; функции \hat{f}_1 и \hat{f}_2 непрерывны в \mathbb{E}^2 и аппроксимируют f_1 и f_2 на множествах узлов сеток g_1 и g_2 соответственно. Будем называть сетку $g = g_1 \cup g_2$ *общей*, или *объединённой, сеткой*. Будем считать, что исходные сетки состоят из непересекающихся множеств узлов: $g_1 \cap g_2 = \emptyset$. Тогда количество узлов в сетке g равно $N = N_1 + N_2$.

В рамках предлагаемого подхода рассмотрим меру различия между поверхностями, представляющую собой среднее расстояние между двумя функциями по всем точкам объединённой сетки — *среднее осевое расстояние*:

$$\rho_M(f_1, f_2) = \sum_{(x,y) \in g} |\hat{f}_1(x, y) - \hat{f}_2(x, y)| / N. \quad (2.4)$$

В реальных данных образуются шумовые эффекты, проявляющиеся в появлении выбросов. Для подавления таких шумов можно использовать следующую меру.

Пусть задано число $\alpha \in [0, 1]$, в качестве значения меры $\rho_M^\alpha(f_1, f_2)$ будем использовать максимальное из $\lfloor \alpha N \rfloor$ минимальных осевых расстояний (см. рис. 2.1).

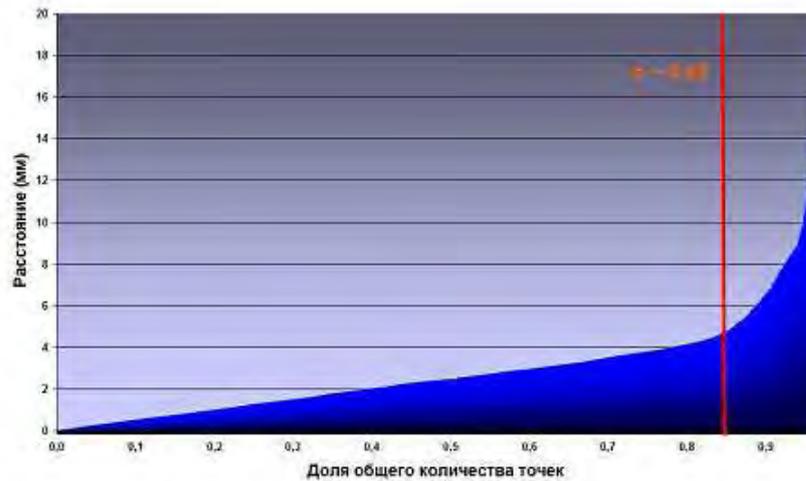


Рис. 2.1. Пример упорядоченных по возрастанию осевых расстояний для двух моделей человеческих лиц при $\alpha = 0.85$.

Введём величину $\rho_M^\alpha(f_1, f_2)$ следующим образом:

$$\rho_M^\alpha = \arg \min \{ \rho \mid K(\rho) \geq \alpha N \}, \quad (2.5)$$

где $K(\rho) = |\{i : |f_1^i - f_2^i| \leq \rho\}|$ — количество точек, в которых значения функций отличаются не более чем на величину ρ .

Будем называть меру $\rho_M^\alpha(f_1, f_2)$ *отсечённым осевым расстоянием*.

2.1.3.1. Мера различия поверхностей на объединённой триангуляции Делоне (ОТД)

Пусть $T_1 = DT(g_1)$, $T_2 = DT(g_2)$ и $T = DT(g)$ — триангуляции Делоне, построенные на сетках g_1 , g_2 и объединённой сетке g соответственно. Триан-

гуляцию T будем называть *общей*, или *объединённой* триангуляцией Делоне (ОТД).

Введём обозначение взвешенного объёма разности между поверхностями на треугольной области, ограниченной узлами A, B, C :

$$V_\mu(A, B, C, f_1, f_2) = \iint_{\Delta ABC} |\hat{f}_1(x, y) - \hat{f}_2(x, y)| \mu(x, y) dx dy, \quad (2.6)$$

где $\mu(x, y) \geq 0$ — функция, определяющая вес (значимость) различия сравниваемых поверхностей в точке (x, y) . Будем считать, что $\mu(x, y)$ определена и конечна во всех точках $Conv(g)$, равна нулю вне $Conv(g)$ и интегрируема.

V_μ при $\mu \equiv 1$ является метрикой L_1 для функций \hat{f}_1 и \hat{f}_2 на ΔABC .

Обозначим через $S_{Conv(g)}$ площадь выпуклой оболочки множества g , которая будет равна сумме площадей всех треугольников триангуляции множества g :

$$S_{Conv(g)} = \sum_{\Delta ABC \in T} S_{\Delta ABC}.$$

Введём меру для f_1 и f_2 как

$$\rho_{V_\mu}(f_1, f_2) = \sum_{\Delta ABC \in T} V_\mu(A, B, C, f_1, f_2) / S_{Conv(g)}. \quad (2.7)$$

Суммирование ведётся по всем треугольникам объединённой триангуляции T .

Определённая таким образом мера ρ_{V_μ} задаёт функцию расстояния между двумя триангуляционными моделями поверхностей. Назовём меру ρ_{V_μ} *мерой различия поверхностей на ОТД*.

Покажем, что введённая мера является полуметрикой.

Теорема 2.1. *Для введённой меры $\rho_{V_\mu}(f_1, f_2)$ выполнены все аксиомы метрики, за исключением, быть может, неравенства треугольника.*

Доказательство. Выполнение аксиом тождества и симметричности следует из определения введённой меры.

Неравенство треугольника для рассматриваемой меры

$$\rho_{V_\mu}(f_1, f_3) \leq \rho_{V_\mu}(f_1, f_2) + \rho_{V_\mu}(f_2, f_3) \quad (2.8)$$

может быть не выполнено из-за того, что для каждой пары поверхностей из трёх, заданных функциями f_1, f_2, f_3 в узлах сеток g_1, g_2, g_3 соответственно, в левой и правой частях неравенства (2.8) будут стоять суммы по треугольникам трёх разных триангуляций: $T_{12} = DT(g_1 \cup g_2), T_{13} = DT(g_1 \cup g_3), T_{23} = DT(g_2 \cup g_3)$. Теорема доказана. ■

Замечание 2.1. Для трёх поверхностей, заданных на одной и той же сетке, неравенство треугольника будет всегда выполнено, так как три суммы в (2.8) будут вычисляться по одному и тому же множеству треугольников. В этом случае введённая мера $\rho_{V_\mu}(f_1, f_2)$ будет метрикой.

Введённая мера может быть адаптирована для каждого конкретного приложения с помощью введения функции $\mu(x, y)$.

Если в качестве функции $\mu(x, y)$ взять тождественную единицу, сходство всех фрагментов поверхностей будет учитываться с одинаковым весом. Обозначим через V значение (2.6) при $\mu(x, y) \equiv 1$ на $Conv(g)$, а через ρ_V — соответствующую (2.7) меру сравнения:

$$\rho_V(f_1, f_2) = \sum_{\Delta ABC \in T} V(A, B, C, f_1, f_2) / S_{Conv(g)}. \quad (2.9)$$

Один и тот же вес для всех фрагментов может быть оправдан в том случае, когда обе сетки имеют одинаковую плотность (2.1) и их узлы расположены равномерно.

2.1.3.2. Мера различия поверхностей на интерфейсных гранях ОТД

Объективная природа данных такова, что сетки, на которых заданы сравниваемые функции не всегда являются равномерными, и в объединённой сетке, по которой происходит сравнение, присутствуют существенные по площади области, состоящие из узлов только одной из исходных сеток.

Недостатком меры ρ_V является то, что она с одинаковым весом учитывает разницу поверхностей и там, где точки двух сеток «перемешаны» между собой, и там, где одна из сеток гуще, чем другая. На практике этот недостаток проявляется при решении задачи сравнения поверхностей объектов, полученных разными сканерами, которые могут отличаться по плотности точек в облаке на несколько порядков. В случае сильной разреженности одной из сеток возможны большие ошибки линейной интерполяции поверхностей, связанные с тем, что внутри одной плоской грани первой сетки расположены несколько плотно лежащих узлов второй сетки (рис. 2.2).

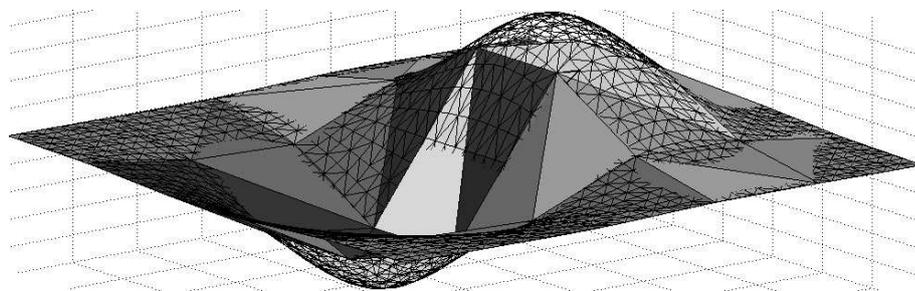


Рис. 2.2. Модели, аппроксимирующие одну и ту же поверхность $z = xe^{-x^2-y^2}$, заданные на сетках с разными плотностями.

В результате, в случае, когда исходные сетки сильно отличаются по плотности, в мере (2.7) влияние ошибок линейной интерполяции будет очень велико.

Это приводит к идее отбора для сравнения только тех пар узловых точек разных сеток, которые расположены близко друг к другу. При этом критерий

близости будет определяться соседством узлов в ОТД. Предлагается модифицировать ρ_V так, чтобы она учитывала только репрезентативные данные — те области, где сосредоточены точки обеих сеток, т. е. те треугольники ОТД T , которые не входят ни в одну из триангуляций T_1 и T_2 исходных сеток. Введём обозначение $S_{Int(g)}$ для суммарной площади таких треугольников:

$$S_{Int(g)} = \sum_{\substack{\Delta ABC \in T: \\ \Delta ABC \notin T_1, \\ \Delta ABC \notin T_2}} S_{\Delta ABC}.$$

Тогда новая мера будет вычисляться по формуле:

$$\rho_{\partial V}(f_1, f_2) = \sum_{\substack{\Delta ABC \in T: \\ \Delta ABC \notin T_1, \\ \Delta ABC \notin T_2}} V(A, B, C, f_1, f_2) / S_{Int(g)}. \quad (2.10)$$

Определение 2.11. Рёбра и грани (треугольники) общей триангуляции T , построенной на множестве узлов сетки $g = g_1 \cup g_2$, соединяющие узлы из разных исходных сеток g_1 и g_2 , будем называть *интерфейсными рёбрами и гранями*.

Суммирование в (2.10) происходит только по интерфейсным граням.

Утверждение 2.2. Мера (2.10) является частным случаем меры (2.7):

$\rho_{\partial V}(f_1, f_2) \equiv \rho_{V_\mu}(f_1, f_2)$ при весовой функции $\mu = \mu^*$:

$$\mu^*(x, y) = \begin{cases} 1, & (x, y) \in \Delta ABC : \Delta ABC \in T, \\ & \Delta ABC \notin T_1 \text{ и } \Delta ABC \notin T_2; \\ 0, & \text{иначе.} \end{cases} \quad (2.11)$$

Доказательство. Непосредственно из определения введённой меры $\rho_{V_\mu}(f_1, f_2)$ вытекает, что использование в ней весовой функции (2.11) приводит к формуле для меры $\rho_{\partial V}(f_1, f_2)$, что доказывает утверждение. ■

Для вычисления мер $\rho_M, \rho_M^\alpha, \rho_{V_\mu}$ необходимы значения обеих функций f_1, f_2 в каждой точке общей сетки g . Так как изначально в каждом узле сетки g известно значение только одной из функций, вычисление мер напрямую невозможно. Далее предлагается метод, позволяющий с высокой вычислительной эффективностью восполнить недостающие значения функций с помощью локализации триангуляций Делоне друг в друге.

2.2. Алгоритм сравнения поверхностей объектов

Предлагаемый метод решения задачи состоит в следующем: на исходных плоских облаках точек строятся триангуляции Делоне, далее значения каждой из функций интерполируются в узлах противоположной сетки, строится общая триангуляция для двух сеток. После этого в каждой точке общей сетки известны значения *двух* функций, и можно производить операции на отдельных ячейках (треугольниках) общей триангуляции, анализируя взаимное расположение пространственных треугольников, заданных функциями, относительно друг друга.

В основе предлагаемого алгоритма лежит идея интерполяции значений каждой из функций в узлах другой сетки через построение триангуляций Делоне и локализацию их друг в друге.

2.2.1. Структура алгоритма

Рассмотрим структуру алгоритма сравнения поверхностей, вычисляющего значение меры между поверхностями, представленными функциями f_1, f_2 в узлах сеток g_1, g_2 соответственно.

Предлагаемый алгоритм решения поставленной задачи включает следующие этапы:

1. Построение триангуляций Делоне T_1, T_2 на множествах узлов сеток g_1, g_2 соответственно;
2. Локализация узлов каждой из триангуляций в треугольниках другой триангуляции: узлов сетки g_1 в триангуляции T_2 , узлов сетки g_2 в триангуляции T_1 ;
3. Интерполяция значений функции f_1 в узлах сетки g_2 , интерполяция значений функции f_2 в узлах сетки g_1 на основе результатов локализации;
4. Построение общей триангуляции T (на сетке $g_1 \cup g_2$);
5. Сравнение функций (вычисление мер) на отдельных гранях общей триангуляции.

Замечание 2.2. *Вычисление мер $\rho_M(f_1, f_2)$ (2.4) и $\rho_M^\alpha(f_1, f_2)$ (2.5) можно производить уже после этапа 3, без построения общей триангуляции T .*

Замечание 2.3. *В разделе 2.3 будет показано, что вычисление значения меры $\rho_{\partial V}(f_1, f_2)$ (2.10) можно производить непосредственно на этапе построения общей триангуляции.*

Рассмотрим каждый из этапов алгоритма подробнее.

2.2.2. Построение триангуляций Делоне

Задача построения триангуляции Делоне по заданному набору точек S заключается в построении триангуляции множества точек S , в которой все треугольники удовлетворяют условию Делоне.

Впервые задача была поставлена в работе Б.Н.Делоне [6]. Трудоёмкость её решения составляет $O(N \log N)$. Существуют алгоритмы, позволяющие

достичь этой оценки как в среднем, так и в худшем случае. Широкий обзор алгоритмов построения триангуляции Делоне представлен в [29, 30].

Одним из алгоритмов, имеющих сложность $O(N \log N)$ в среднем и в худшем случае, где N — количество точек в исходном множестве, является алгоритм, предложенный Ли и Шехтером [73]. В его основе лежит парадигма рекурсивной декомпозиции («разделяй и властвуй», [2]): разделение исходного множества на два примерно равных подмножества, рекурсивное построение триангуляций этих множеств и слияние двух разделённых триангуляций (рис. 2.3).

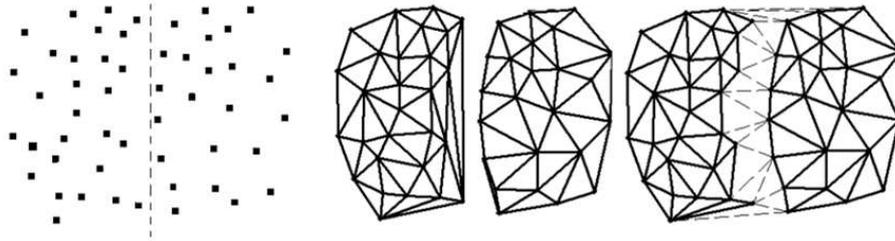


Рис. 2.3. Рекурсивное построение триангуляции Делоне слиянием.

Важным этапом построения триангуляции Делоне является проверка условия Делоне. Одним из способов проверки является проверка *углового критерия*, то есть проверка суммы противолежащих углов [29]: для треугольника триангуляции $\Delta((x_1, y_1), (x_2, y_2), (x_3, y_3))$ условие Делоне выполнено тогда и только тогда, когда для любого другого узла триангуляции (x, y) выполнено неравенство $\alpha + \beta \leq \pi$ (см. рис. 2.4).

2.2.3. Локализация триангуляций друг в друге

После этапа построения триангуляций следует этап взаимной локализации узлов исходных сеток в триангуляциях Делоне. В данном разделе предложен алгоритм, позволяющий выполнять такую локализацию за линейное по общему количеству узлов в двух сетках время. Алгоритм основан на прослеживании рёбер минимального остовного дерева, построенного на узлах сетки,

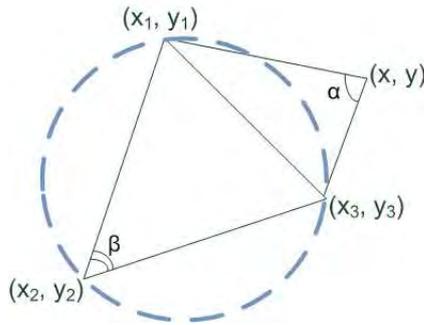


Рис. 2.4. Угловой критерий: проверка условия Делоне через сумму противоположных углов.

которую необходимо локализовать.

Далее рассматриваются задачи:

- построения минимального остовного дерева на основе триангуляции Делоне;

- локализации отдельного узла в триангуляции Делоне;

- локализации узлов триангуляционной сетки в триангуляции Делоне.

Последняя задача является массовым запросом для задачи локализации отдельного узла. Показывается, что оптимальное время решения задачи локализации сетки с помощью неструктурированного массового запроса локализации отдельных узлов может быть решена за время $O(N \log N)$. Предлагается использовать структурированность множества узлов для массового запроса (т. е. триангуляцию Делоне) для получения эффективного $O(N)$ решения задачи.

2.2.3.1. Построение минимальных остовных деревьев

Известные алгоритмы поиска МОД произвольного графа основываются на следующей лемме [85]:

Лемма 2.3. Пусть $G = (V, E)$ — взвешенный граф, $\{V_1, V_2\}$ — разбиение множества V . В G имеется МОД, содержащее кратчайшее из рёбер, соединяющее вершину из V_1 и вершину из V_2 .

Пусть S — множество точек на плоскости, $DT(S)$ — триангуляция Делоне, построенная на множестве S . На основе леммы 2.3 доказывается [28], что

Лемма 2.4. Пусть $\Delta(p)$ — множество точек, смежных с $p \in S$ в $DT(S)$. Тогда при любом разбиении $\{S_1, S_2\}$ множества S , если \overline{qp} — кратчайший отрезок, соединяющий точки из S_1 и S_2 , то $q \in \Delta(p)$.

Имеет место следующая теорема [86]:

Теорема 2.5. МОД множества S является подграфом триангуляции Делоне этого множества: $MST(S) \subseteq DT(S)$.

В силу теоремы 2.5 при построении МОД конечного множества точек, можно находить МОД не полного, а планарного графа. Таким образом, евклидовым минимальным остовным деревом триангуляции Делоне $DT(S)$ (см. рис. 2.5) является её связный подграф с множеством вершин S , имеющий наименьшую суммарную длину рёбер.

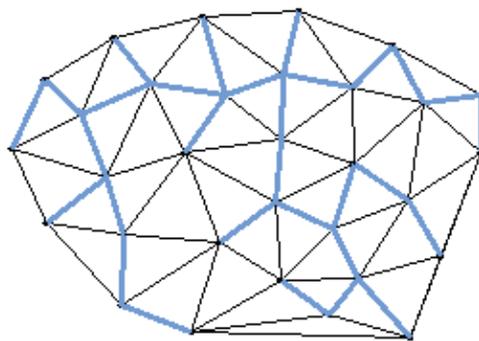


Рис. 2.5. Минимальное остовное дерево триангуляции Делоне.

Теоретическая оценка трудоёмкости задачи построения минимального остовного дерева на N точках равна $\Theta(N \log N)$ [28]. Известно, что на основе триангуляции Делоне минимальный остов можно построить за время $O(N)$ [28, 69]:

Теорема 2.6. *Евклидово минимальное остовное дерево множества S из N точек на плоскости может быть построено, исходя из триангуляции Делоне множества S , за оптимальное время $\Theta(N)$.*

Алгоритм Черитона и Тарьяна [42], подробное описание которого можно найти в [28], позволяет построить $MST(S)$ для множества точек S за *линейное время* с помощью $DT(S)$. Рассмотрим основную стратегию алгоритма.

На каждом шаге обрабатывается очередь деревьев (лес) F , которая первоначально представляет собой совокупность всех вершин N . На каждом шаге из очереди выбирается некоторое дерево T и производится поиск такой пары вершин (U', V') , что U' принадлежит выбранному дереву T , а V' — другому дереву леса F , такому, что расстояние между этими вершинами минимальное из возможных, то есть

$$d(U', V') = \min_{(U, V)} \{ d(U, V) \mid U \in T, V \in \{N\} - T \},$$

далее находится дерево T' , содержащее V' , деревья T и T' объединяются в T'' , которое добавляется в F , а дерево T' удаляется из F .

Линейное время работы достигается за счёт *операции очистки*, предложенной Черитоном и Тарьяном, которая сжимает исходный граф H (граф Делоне) в некоторый другой граф H^* так, что каждое дерево из F превращается в одну «супервершину» — вершину графа H^* (см. рис. 2.6). Все внутренние рёбра деревьев из F не рассматриваются, остаются только внешние рёбра. Таким образом, применение описанной операции позволяет графу H содержать на каждом шаге работы алгоритма только *необходимую информацию*.

Рассмотрим вопрос о выборе структуры данных для реализации очереди деревьев. Следует учитывать, что алгоритм подразумевает удаление дерева T' из очереди. Если реализовать её в виде списка, сама операция удаления

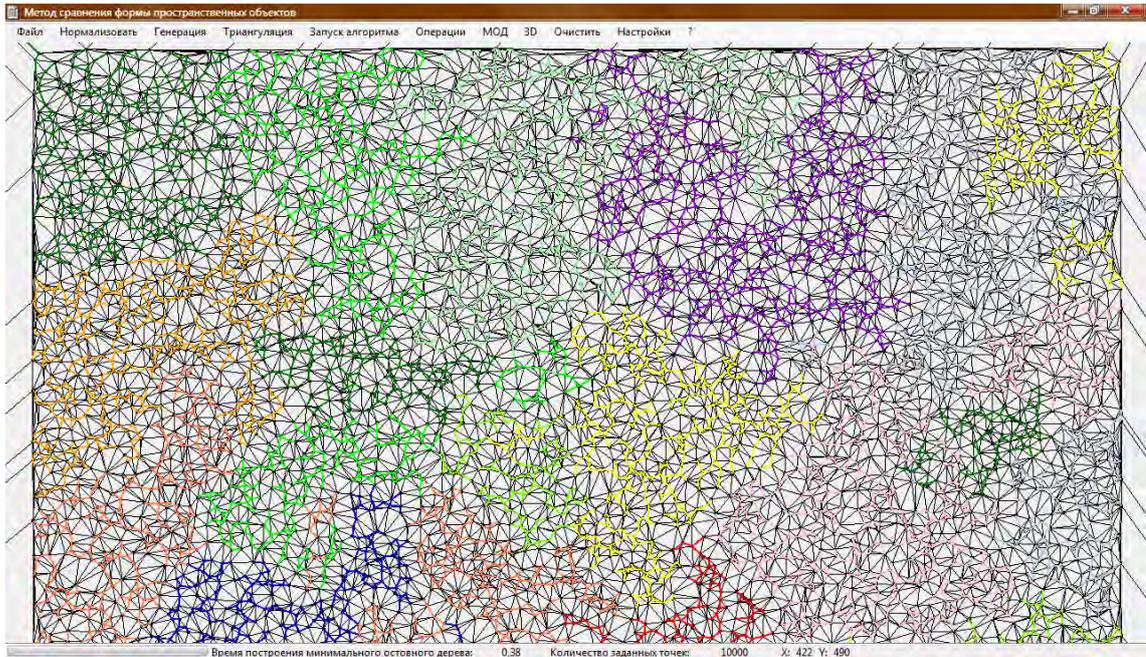


Рис. 2.6. Построение МОД. Разными цветами отмечены поддеревья, сжимаемые в процессе работы алгоритма в одну вершину.

дерева будет иметь сложность $O(1)$, но линейного поиска дерева перед его удалением не избежать. Если реализовать очередь в виде циклического массива поиск элемента будет занимать время $O(1)$, но на операцию удаления будет затрачено в худшем случае линейное время.

Наиболее быстрые известные алгоритмы для задач построения МОД существенно используют *фибоначчиевы кучи* (Fibonacci heap) [18]. Эту структуру данных ввели Фредман и Тарьян в [57], [92]. Использование фибоначчиевых куч для реализации очереди с приоритетами имеет преимущество в том, что операции, в которых не требуется удаление, выполняются за время $O(1)$. Кроме того, фибоначчиева куча позволяет за время $O(1)$ выполнять операцию слияния двух куч. Подробнее о применении этой структуры данных см. подраздел 2.2.8.

2.2.3.2. Задача геометрического поиска: локализация узла в триангуляции

Рассмотрим *задачу локализации узла в триангуляции Делоне*. Пусть заданы узел Q и триангуляция T . Требуется определить треугольник триангуляции T , содержащий узел Q . Если узел Q совпадает с одним из узлов триангуляции, то можно указать любой из инцидентных данному узлу треугольников триангуляции. Если узел Q принадлежит одному из рёбер триангуляции, то можно указать любой из инцидентных данному ребру треугольников. Если узел заходит за границы триангуляции, то есть не принадлежит выпуклой оболочке множества узлов триангуляции, то можно определять ближайший к узлу треугольник.

Обзор существующих методов. Известные методы решения задачи геометрического поиска делятся на точные (использующие точные геометрические вычисления) и неточные, в которых на планарное разбиение (триангуляцию) накладывается равномерная прямоугольная сетка, и запрашиваемая точка локализуется с точностью до размера ячейки сетки.

Существуют методы (алгоритм Киркпатрика, [69]) решения задачи, работающие за время $O(\log N)$ в худшем случае и использующие структуры данных размера $O(N)$, где N — количество узлов в триангуляции. Есть методы, работающие медленнее в худшем случае, но быстрее в среднем. В [67] показано, что не существует подходов, которые являлись бы оптимальными одновременно по всем параметрам: сложности предварительной обработки, использованию памяти и скорости локализации.

Точные методы решения задачи геометрического поиска:

1. Наивный алгоритм (перебор всех граней исходного графа).
2. Алгоритм Киркпатрика [69].

На этапе предобработки алгоритм создаёт иерархию разбиений, все грани которых являются треугольниками (иерархию триангуляций). В любой триангуляции можно за линейное время выделить независимое множество вершин наименьшей степени. Размер этого множества определяется по числу вершин в триангуляции. Для перехода на следующий уровень иерархии, вершины этого множества удаляются из триангуляции, и строятся триангуляции многоугольников, образовавшиеся в результате удаления вершин. Процесс повторяется логарифмическое число раз, после чего образуется «грубая» триангуляция с константным числом вершин. Во время выполнения запроса локализации точки в треугольнике, сначала происходит поиск треугольника в самой «грубой» триангуляции. Про этот треугольник известно, при построении триангуляции какого многоугольника он образовался. Удалённая из этого многоугольника вершина восстанавливается, далее происходит локализация заданной точки в инцидентных ей треугольниках. Время на выполнение одного запроса — $O(\log N)$.

3. Алгоритм последовательных переходов вдоль прямой («Walk along a line» strategy) [45].

Основная стратегия алгоритма — построение путей из попарно смежных треугольников к искомому треугольнику. Прослеживание по треугольникам может осуществляться следующими способами:

- прямое прослеживание (переходы по всем треугольникам, которые пересекают отрезок, соединяющий начальную точку с исходной);
- ортогональное прослеживание (переходы по треугольникам пути, параллельном одной из осей координат);
- прослеживание по обозримому пути (переход от одного треугольника к соседнему осуществляется через первое ребро треугольника, если оно отделяет начальную точку от исходной; в противном случае проверяется второе ребро, и т. д.);

— стохастическое прослеживание (аналогично предыдущему способу, только вместо первого ребра выбирается случайное ребро).

В отличие от алгоритма Киркпатрика, этот алгоритм не требует построения и поддержки дополнительных структур данных. Если вершины триангуляции выбирались случайно из равномерного распределения, среднее время работы $O(\sqrt{N})$, время работы в худшем случае — $O(N)$.

4. Алгоритм прыжков и переходов (Jump and Walk) [47, 81]. Работает только для триангуляций Делоне.

Исходная триангуляция — триангуляция Делоне для N вершин. Случайным образом из них выбираются k вершин. При локализации точки Q из k вершин выбирается ближайшая к точке Q точка M . Далее применяется алгоритм последовательных переходов вдоль отрезка, соединяющего M и Q . Время работы в среднем — $O(k + \sqrt{N/k})$, которое при k равном $O(N/3)$ является оптимальным — $O(N/3)$.

Позже в [46] было предложено улучшение данного алгоритма — алгоритм бинарного поиска и переходов (Binsearch and Walk). k выбирается равным $N/4$, для поиска используется бинарное дерево поиска. С помощью дерева находится ближайший узел, от которой применяется алгоритм последовательных переходов. Там же предложен алгоритм $2d$ поиска и переходов (2d Search and Walk), который использует сбалансированные k - d -деревья (k - d -trees). Для триангуляции Делоне на N вершинах, независимо и равномерно распределённых на плоскости, среднее время локализации $O(\log N)$.

5. Алгоритм опознавательных точек (Landmarks strategy) [67].

На этапе предобработки алгоритма выбираются опознавательные точки (точки-ориентиры), для которых решается задача локализации в триангуляции. Далее точки-ориентиры помещаются в иерархическую структуру данных, обеспечивающую быстрый поиск ближайшего соседа. Когда требуется локализовать новую точку Q , сначала находится ближайшая точка-ориен-

тир Q_i , применяется стратегия последовательных переходов вдоль отрезка, соединяющего Q_i и Q . Теоретическая оценка сложности одного запроса локализации узла для алгоритма - $O(\log N)$.

Таким образом, наиболее быстрые алгоритмы решения задачи локализации узла имеют сложность $O(\log N)$ и требуют предобработки. В задаче локализации сетки из N узлов в триангуляции возникает массовый запрос на решение задачи локализации узла. Неструктурированный массовый запрос из N узлов может быть обработан за время $O(N \log N)$. Далее предлагается использовать структуру локализуемой сетки (триангуляцию Делоне) для получения более эффективного решения.

Для изложения предложенного решения рассмотрим более подробно стратегию локализации узла в триангуляции Делоне, основанную на упоминавшемся выше алгоритме последовательных переходов вдоль прямой при прямом прослеживании [45].

Метод локализации узла. Стратегия метода состоит в выборе некоторой начальной точки M , локализация которой в триангуляции известна, и постепенном переходе от M к Q вдоль прямой (MQ) . На каждом шаге осуществляется переход на смежный треугольник. Сложным случаем является ситуация, когда отрезок $[MQ]$ проходит через какую-либо вершину триангуляции.

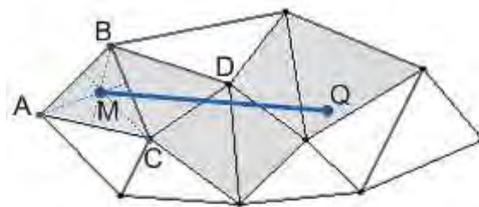


Рис. 2.7. Локализация точки в триангуляции.

В процессе локализации узла строится путь из треугольников триангу-

ляции, каждый из которых (кроме начального) является смежным с предыдущим. На рис. 2.7 и 2.8 выделены треугольники, принадлежащие путям локализации.

Метод локализации узла в триангуляции состоит из следующих шагов:

1. Выбрать начальный треугольник — $\triangle ABC$. Присвоить ему статус текущего. Найти координаты точки $M = (M_x, M_y)$ пересечения медиан (центроида) этого треугольника: $M_x = (A_x + B_x + C_x) / 3$, $M_y = (A_y + B_y + C_y) / 3$.

2. Определить, какую из сторон текущего треугольника пересекает отрезок $[MQ]$. Если данный пункт выполняется первый раз, то мы проверяем все три стороны текущего треугольника на предмет пересечения с отрезком $[MQ]$, в противном случае достаточно проверять только две стороны и запоминать сторону, через которую перешли на текущий треугольник.

а. Пусть отрезок $[MQ]$ не пересекает ни одну из сторон или пересекает её в точке Q . Тогда текущий треугольник является искомым и работа алгоритма заканчивается.

б. Пусть отрезок $[MQ]$ пересекает одну из вершин текущего треугольника. Тогда переходим на треугольник, смежный с текущим по любой из сторон, содержащей данную вершину.

Пример. На рис. 2.8 отрезок $[MQ]$ проходит через вершину C . При переходе на следующий треугольник можно перейти на любой из треугольников $\triangle BCD$ или $\triangle ACE$. Если мы переходим на треугольник $\triangle BCD$ ($\triangle ACE$), то на следующем этапе проверяются на предмет пересечения с $[MQ]$ отрезки $[BD]$ и $[CD]$ ($[EC]$ и $[AE]$), и переход к следующему треугольнику — $\triangle CDF$ ($\triangle CEG$) — будет однозначным.

Треугольник, на который мы перешли, становится текущим. Переходим в начало пункта 2.

в. Пусть отрезок $[MQ]$ пересекает одну из сторон треугольника во внутренней точке. Тогда переходим к треугольнику, смежному с текущим по дан-

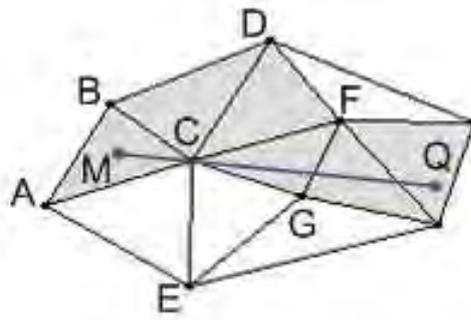


Рис. 2.8. MQ проходит через вершину триангуляции.

ной стороне.

Пример. На рис. 2.7 отрезок $[MQ]$ пересекает сторону BC начального треугольника. На следующем шаге мы переходим к треугольнику $\triangle BCD$.

Треугольник, на который мы перешли, становится текущим. Переходим в начало пункта 2.

Запишем алгоритм в формальном виде.

Пусть триангуляция Делоне представлена в виде структуры данных «Узлы с соседями» [31], в которой *пучок* каждого узла триангуляции представлен в виде двунаправленного циклического списка соседних узлов, упорядоченных по углу.

Пусть методы `Найти_левого_соседа(Node X, Node V)` и `Найти_правого_соседа(Node X, Node V)` возвращают соответственно левого и правого соседа узла X относительно узла V , в массиве $X.Triangle$ хранятся узлы треугольника, в котором локализован узел X . Тогда алгоритм локализации узла можно записать в следующем виде:

Алгоритм. Локализация узла в триангуляции.

Вход:

T — триангуляция Делоне с узлами $g[0], \dots, g[N - 1]$;

$\Delta(g[i]) = \{\Delta(g[i])[0], \dots, \Delta(g[i])[i_k - 1]\}$ — упорядоченный по углу список узлов, смежных с $g[i]$ в T ;

Q — узел для локализации.

Выход:

узлы A, B, C такие, что $\triangle ABC \in T$ и Q находится внутри или на границе $\triangle ABC$.

- 1: $\tilde{A} := g[0]$; $\tilde{B} := \Delta(g[0])[0]$; $\tilde{C} := \Delta(g[0])[1]$;
- 2: $M_x := (\tilde{A}_x + \tilde{B}_x + \tilde{C}_x)/3$; $M_y := (\tilde{A}_y + \tilde{B}_y + \tilde{C}_y)/3$;
- 3: $\triangle ABC :=$ Локализовать_узел($M, \tilde{A}, \tilde{B}, \tilde{C}, T, \text{ИСТИНА}$);
- 4: вернуть A, B, C ;

ПРОЦЕДУРА Локализовать_узел($Node M, Node A, Node B, Node C, Node Q, bool flag$)

- 1: **если** ($flag = \text{ИСТИНА}$) и ($[MQ]$ и $[AB]$ пересекаются) **то**
- 2: $R :=$ Найти_правого_соседа(A, B);
- 3: **если** ($R = C$) **то**
- 4: $L :=$ Найти_левого_соседа(A, B);
- 5: Локализовать_узел($M, A, B, L, Q, T, \text{ЛОЖЬ}$);
- 6: **иначе**
- 7: Локализовать_узел($M, A, B, R, Q, T, \text{ЛОЖЬ}$);
- 8: **иначе если** ($[MQ]$ и $[AC]$ пересекаются) **то**
- 9: $L :=$ Найти_левого_соседа(A, C);
- 10: **если** ($L = B$) **то**
- 11: $R :=$ Найти_правого_соседа(A, C);
- 12: Локализовать_узел($M, A, C, R, Q, T, \text{ЛОЖЬ}$);
- 13: **иначе**
- 14: Локализовать_узел($M, A, C, L, Q, T, \text{ЛОЖЬ}$);
- 15: **иначе если** ($[MQ]$ и $[BC]$ пересекаются) **то**
- 16: $R :=$ Найти_правого_соседа(B, C);
- 17: **если** ($R = A$) **то**

```

18:     L := Найти_левого_соседа(B, C);
19:     Локализовать_узел(M, B, C, L, Q, T, ЛОЖЬ);
20:     иначе
21:     Локализовать_узел(M, B, C, R, Q, T, ЛОЖЬ);
22:     иначе
23:     вернуть  $\triangle ABC$ ;

```

Таким образом, процедура Локализовать_узел($M, A, B, C, Q, T, flag$) локализует точку Q в триангуляции T , двигаясь вдоль отрезка $[MQ]$, причём точка M уже локализована в треугольнике $\triangle ABC$ триангуляции T . Индикатор $flag$ показывает, первый раз выполняется локализация узла или нет.

Трудоёмкость локализации одной точки определяется количеством расположенных вдоль отрезка $[MQ]$ треугольников и составляет $O(\sqrt{N})$ в среднем (при равномерном распределении узлов) [87] и $O(N)$ в худшем случае (когда $[MQ]$ пересекает все треугольники триангуляции).

2.2.3.3. Локализация сетки в триангуляции

Задачей локализации двумерной сетки g в триангуляции T называется задача локализации каждой точки сетки g в данной триангуляции.

Предлагается алгоритм локализации сетки, использующий МОД, вершинами которого являются узлы данной сетки (рис. 2.9). Локализация узлов организовывается таким образом, чтобы пути локализации проходили вдоль рёбер МОД.

Предлагаемый алгоритм локализации сетки можно записать в следующем виде:

Алгоритм. Локализация узлов сетки в триангуляции.

Вход:

g_1 — сетка с узлами $g_1[0], \dots, g_1[N_1 - 1]$ для локализации;

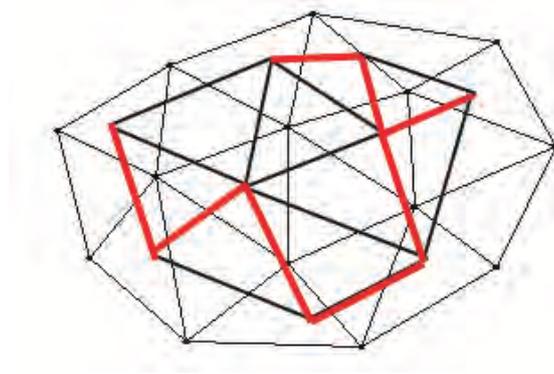


Рис. 2.9. Локализация сетки в триангуляции с помощью МОД узлов сетки.

T_2 — триангуляция Делоне множества узлов $g_2 = \{g_2[0], \dots, g_2[N_2 - 1]\}$;
 MST_1 — МОД сетки g_1 , состоящее из списка рёбер d с концевыми узлами $d.orig, d.dest$.

Выход:

заполненный массив *Triangle* для всех узлов сетки g_1 .

-
- 1: $A := g_2[0]; B := \Delta(g_2[0])[0]; C := \Delta(g_2[0])[1];$
 - 2: $M_x := (A_x + B_x + C_x)/3; M_y := (A_y + B_y + C_y)/3;$
 - 3: $M.Triangle := \Delta ABC;$
 - 4: Локализовать_узел_сетки($M, g_1[0], T_2, MST_1$);
 - 5: **вернуть** g_1 ;
-

ПРОЦЕДУРА Локализовать_узел_сетки(*Node* $M, Node$ $Q,$
Triangulation $T, Tree$ MST)

- 1: $Q.Triangle =$ Локализовать_узел($M, M.Triangle[0], M.Triangle[1],$
 $M.Triangle[2], Q, T, ИСТИНА$);
 - 2: **если** (число рёбер инцидентных Q и принадлежащих $MST = 1$) **то**
 - 3: **ВЫХОД**;
 - 4: **для всех** (рёбер $d \in MST$ таких, что $d.orig = Q$, а $d.dest$ ещё не локализована)
 - 5: Локализовать_узел_сетки($Q, d.dest, T, MST$);
-

Процедура Локализовать_узел_сетки(M, Q, T, MST) рекурсивная. В качестве первого параметра ей передаётся центроид любого треугольника триангуляции T ; в качестве второго параметра — узел $g[0]$; третий параметр — это триангуляция T , в которой нужно локализовать сетку; четвёртый параметр — минимальное остовное дерево сетки g . Алгоритм вызывает процедуру для первого узла сетки $g[0]$. Далее с помощью процедуры Локализовать_узел выполняется локализация первого узла, затем рекурсивно вызывается для всех узлов, соседних с данным узлом в МОД. Таким образом, при каждом (кроме самого первого) вызове процедуры Локализовать_узел_сетки(M, Q, T, MST) её параметры будут такими, что MQ — ребро МОД сетки g , и пути локализации будут проходить вдоль рёбер минимального остова.

Так как по определению МОД не содержит в себе циклов и проходит через все узлы сетки g , алгоритм будет работать корректно: он не зациклится и произведёт локализацию всех N узлов сетки g .

2.2.3.4. Вычислительная сложность этапа локализации

Оценим вычислительную сложность предложенного алгоритма локализации узлов сетки в триангуляции Делоне на основе минимального остова узлов данной сетки в среднем и в худшем случае.

Пусть координаты узлов исходных сеток g_1 и g_2 распределены равномерно и независимо по X и Y в прямоугольнике R . Внутри R выделим прямоугольник R' со сторонами a и b так, чтобы внутри R' не было «вытянутых» треугольников триангуляции T_2 , которые могут располагаться вдоль выпуклой оболочки множества узлов g_2 (см. рис. 2.10). Пусть внутри прямоугольника R' содержится N'_2 узлов сетки g_2 . В таких условиях и при достаточно больших N_2 можно считать, что $N'_2 \approx N_2$.

В работе [20] было получено значение средней евклидовой длины \bar{l} ребра

триангуляции Делоне T_2 для описанных условий:

$$\bar{l} = \frac{\pi\sqrt{5}}{6} \sqrt{\frac{ab}{N_2'}}. \quad (2.12)$$

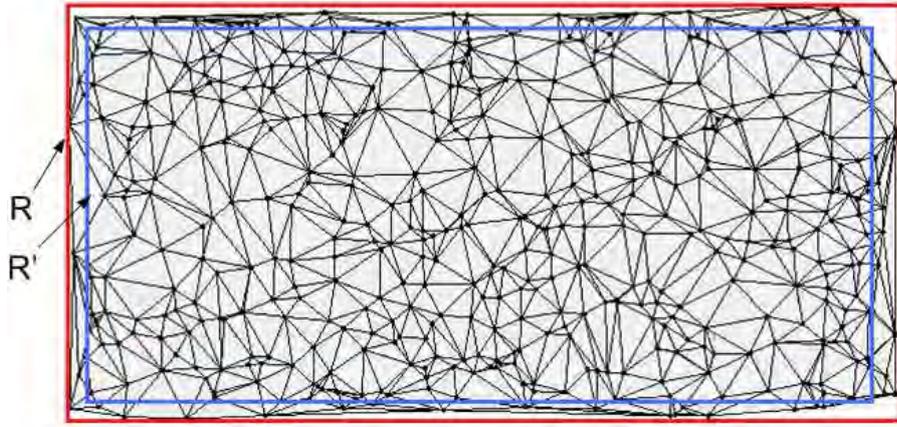


Рис. 2.10. Выбор прямоугольника R' .

Пусть минимальный остов $MST_1 = MST(g_1)$ пересекает s рёбер триангуляции T_2 , имеющих длины l_1, \dots, l_s . Количество s пересекаемых рёбер не превосходит общего количества рёбер триангуляции, линейного по числу её узлов, поэтому $s = O(N_2)$. Однако МОД может пересекать какие-то рёбра несколько раз. Обозначим через Q количество пересечений минимального остова MST_1 и s пересекаемых им рёбер. Для определения вычислительной сложности алгоритма локализации требуется оценить Q .

В [38] доказана теорема, оценивающая среднее количество пересечений МОД и прямолинейного отрезка заданной длины:

Теорема 2.7. Пусть X — множество из n точек X_1, \dots, X_n , независимо распределённых с плотностью f на выпуклом компакте C на плоскости. Пусть для некоторых констант α, β на C выполнено: $0 < \alpha \leq f(x) \leq \beta < \infty$. Пусть L — прямолинейный отрезок длины l , содержащийся в C . Тогда существуют такие константы d и n_0 , зависящие от α, β и C , что

$$\bar{Q} \leq dl\sqrt{n}, \quad n \geq n_0, \quad (2.13)$$

где \bar{Q} — среднее количество пересечений минимального остовного дерева множества X и отрезка L .

Для рассматриваемой задачи выполнены условия теоремы 2.7, поэтому для среднего количества \bar{Q}_i пересечений минимального остова MST_1 и ребра триангуляции T_2 длины l_i справедлива оценка: $\bar{Q}_i = dl_i\sqrt{N_1}$. Складывая оценку (2.13) для s пересекаемых рёбер, получим среднее количество пересечений MST_1 с рёбрами T_2 :

$$\bar{Q} = \sum_{i=1}^s \bar{Q}_i \leq d \sum_{i=1}^s l_i \sqrt{N_1}.$$

Используя значение для средней длины ребра (2.12), получим:

$$\bar{Q} \leq d \frac{\pi\sqrt{5}}{6} s \sqrt{\frac{ab}{N_2}} \sqrt{N_1} = d \frac{\pi\sqrt{5ab}}{6} s \sqrt{\frac{N_1}{N_2}}.$$

При $N_1/N_2 \leq c^2 = const$ получим, что

$$\bar{Q} \leq cd \frac{\pi\sqrt{5ab}}{6} s.$$

При примерно равной мощности сеток $N_1 \approx N_2$, можно полагать $c = 1$.

Учитывая то, что $s = O(N_2)$, такая же оценка будет справедлива для \bar{Q} : $\bar{Q} = O(N_2)$.

Таким образом, доказана следующая лемма:

Лемма 2.8. Пусть узлы сеток g_1, g_2 с мощностями N_1, N_2 соответственно распределены равномерно в прямоугольнике и $N_1/N_2 \leq c = const$. Тогда среднее количество пересечений МОД множества узлов g_1 с рёбрами триангуляции Делоне, построенной на узлах g_2 , линейно по N_2 .

Лемма 2.8 позволяет оценить трудоёмкость алгоритма локализации в среднем:

Теорема 2.9. Алгоритм локализации множества узлов сетки g_1 мощности N_1 в триангуляции Делоне, построенной на множестве узлов g_2 мощности N_2 , на основе МОД g_1 имеет линейную по $\max(N_1, N_2)$ вычислительную сложность в среднем при равномерном распределении узлов обеих сеток в прямоугольнике и условии ограничения отношения мощностей сеток $N_1/N_2 \leq c = \text{const}$.

Доказательство. Локализация первого узла сетки g_1 производится методом последовательных переходов вдоль прямой. Количество треугольников в пути этой локализации не превосходит общее количество треугольников в триангуляции T_2 , т.е. $O(N_2)$.

Далее алгоритм осуществляет проход вдоль рёбер минимального остова MST_1 (всего $N_1 - 1$ рёбер) по всем пересекаемым им треугольникам триангуляции T_2 . При этом переход на следующий треугольник осуществляется за константное количество операций $O(1)$, позволяющих анализировать пересечения ребра минимального остова с рёбрами триангуляции. В соответствии с леммой 2.8, среднее количество таких пересечений в условиях теоремы есть $O(N_2)$.

Во время обхода MST_1 для каждого узла g_1 определяется треугольник триангуляции T_2 , внутри которого содержится данный узел (константное количество операций на узел).

Таким образом, вычислительная сложность алгоритма в среднем будет линейной по количеству узлов в сетке большей мощности $O(\max(N_1, N_2))$.

Теорема доказана. ■

В худшем случае время работы предложенного алгоритма будет квадратичным. Рассмотрим модельный контрпример для худшего случая:

Пример. Худший случай для алгоритма локализации сетки в триангуляции Делоне с использованием МОД.

Пусть точки первой сетки, состоящей из N_1 точек, расположены на двух

вертикальных отрезках длины H с фиксированным шагом (см. рис. 2.11). На каждом из отрезков примерно одинаковое количество точек, расстояние между отрезками равно L , $L > H$. Таким образом, выпуклая оболочка множества точек первой сетки, есть некоторый прямоугольник R со сторонами L и H .

Точки второй сетки, состоящей из N_2 точек, расположены на двух горизонтальных отрезках, расстояние между которыми равно H , содержащихся в прямоугольнике R , с фиксированным шагом так, что расстояние между двумя соседними точками на отрезках превосходит расстояние между двумя ближайшими точками из разных отрезков. Тогда МОД второй сетки будет состоять из отрезков, соединяющих точки, лежащие на разных горизонтальных отрезках.

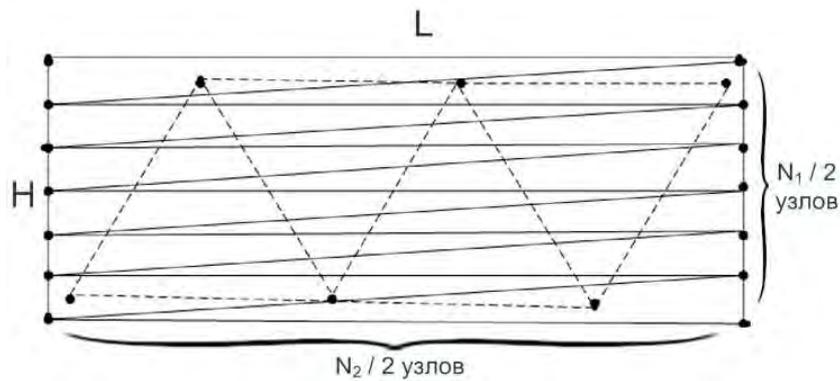


Рис. 2.11. Модельный пример для задачи локализации сетки в триангуляции.

Оценим количество операций при решении задачи локализации второй сетки в триангуляции Делоне первой сетки. Каждое из $(N_2 - 1)$ рёбер второго МОД пересекает $O(N_1)$ треугольников первой триангуляции. Поэтому общее число операций при локализации, равное $O(N_1 N_2)$, будет квадратичным при $N_1 \approx N_2$.

Найдём соотношение между L и H , при котором реализуется рассматриваемый пример. Пусть расстояние между двумя соседними во второй триангуляции точками, лежащими на разных горизонталях, равно a , а между двумя соседними точками, лежащими на одной горизонтали — b ; по усло-

вию примера $b > a$ (см. рис. 2.12). Тогда в $\triangle ABC$ угол $\angle BAC < 60^\circ$ и высота h , опущенная на AC из B $h < a\frac{\sqrt{3}}{2}$. Будем считать, что $h \approx H$, например, $H = h(1 + \alpha)$ для некоторого малого α . Тогда

$$H < (1 + \alpha)a\frac{\sqrt{3}}{2} \Rightarrow a > \frac{2}{3}\sqrt{3}H\frac{1}{1 + \alpha};$$

$$\frac{2L}{N_2} \approx b > a > \frac{2}{3}\sqrt{3}H\frac{1}{1 + \alpha} \Rightarrow L > \frac{\sqrt{3}}{3}HN_2\frac{1}{1 + \alpha}.$$

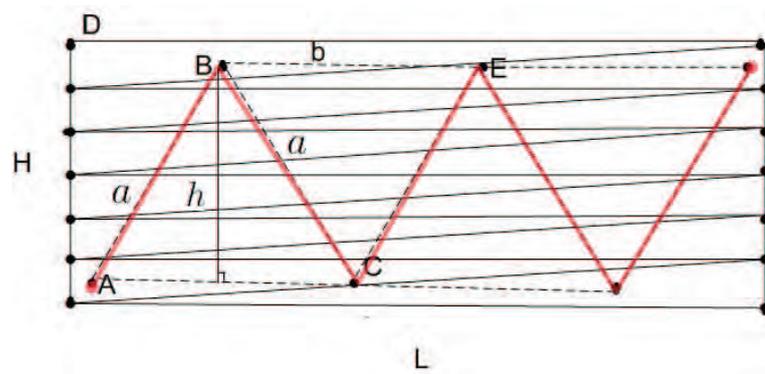


Рис. 2.12. Модельный пример для задачи локализации сетки в триангуляции.

Таким образом, в рассматриваемом примере при $L > \frac{\sqrt{3}}{3}HN_2\frac{1}{1+\alpha}$ время работы алгоритма будет квадратичным.

С помощью предложенного алгоритма локализации сетка g_1 локализуется в триангуляции сетки g_2 , а сетка g_2 — в триангуляции сетки g_1 . Таким образом, каждая точка сетки локализована в некотором треугольнике триангуляции другой сетки, и можно рассмотреть задачу интерполяции функции, заданной на одной сетке, в точках другой сетки.

2.2.4. Интерполяция поверхностей

В рамках предлагаемого подхода используется линейная интерполяция поверхностей. Следует отметить, что для некоторых приложений получаемое кусочно-линейное представление поверхности может оказаться слишком грубым. В этом случае можно использовать интерполирующую функцию более

высокой степени. Например, квадратичную [23] или кубическую [19] функции. Однако такое решение будет трудоёмким.

Пусть в треугольнике $\Delta (V_1(x_1, y_1), V_2(x_2, y_2), V_3(x_3, y_3))$ локализован узел первой сетки $V_0(x_0, y_0)$, а в узлах этого треугольника задана функция f : $f(x_1, y_1) = f_1$, $f(x_2, y_2) = f_2$, $f(x_3, y_3) = f_3$ (см. рис. 2.13).

Требуется проинтерполировать значение функции f в узле V_0 .

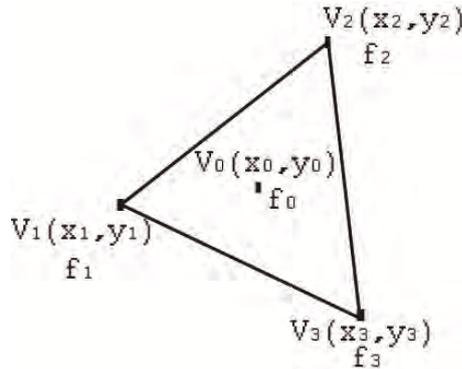


Рис. 2.13. Линейная интерполяция функций.

Будем использовать линейную интерполяцию и барицентрические координаты $\lambda_1, \lambda_2, \lambda_3 \in [0, 1]$: $f_0 = \lambda_1 f_1 + \lambda_2 f_2 + \lambda_3 f_3$, где

$$\begin{cases} V_0 = \lambda_1 V_1 + \lambda_2 V_2 + \lambda_3 V_3 \\ 1 = \lambda_1 + \lambda_2 + \lambda_3 \end{cases} \Leftrightarrow \begin{cases} x_0 = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 \\ y_0 = \lambda_1 y_1 + \lambda_2 y_2 + \lambda_3 y_3 \\ 1 = \lambda_1 + \lambda_2 + \lambda_3 \end{cases} .$$

Определитель матрицы

$$A = \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix}$$

данной системы $\Delta = \det(A)$ не равен нулю, так как вершины треугольника не лежат на одной прямой. Обозначим $\lambda = (\lambda_1 \lambda_2 \lambda_3)^T$, $\mathbf{b} = (x_0 \ y_0 \ 1)^T$.

Найдём λ из уравнения $A\lambda = \mathbf{b}$ с помощью формул Крамера:

$$\lambda_i = \frac{\Delta_i}{\Delta}, \quad i \in \{1, 2, 3\},$$

где

$$\Delta_1 = x_0(y_2 - y_3) - x_2(y_0 - y_3) + x_3(y_0 - y_2);$$

$$\Delta_2 = x_1(y_0 - y_3) - x_0(y_1 - y_3) + x_3(y_1 - y_0);$$

$$\Delta_3 = x_1(y_2 - y_0) - x_2(y_1 - y_0) + x_0(y_1 - y_2);$$

$$\Delta = x_1(y_2 - y_3) - x_2(y_1 - y_3) + x_3(y_1 - y_2).$$

Результаты работы данного метода на модельных данных показаны на рис. 2.14 – 2.17. Красным цветом изображена триангулированная поверхность, заданная функцией f_1 на сетке g_1 , зелёным цветом — триангулированная поверхность, заданная функцией f_2 на сетке g_2 , синим цветом — поверхность, полученная с помощью интерполяции функции f_2 на сетке g_1 . Как видно из рисунков, поверхности, заданные функцией f_2 на первой и второй сетках, задают одну и ту же поверхность.

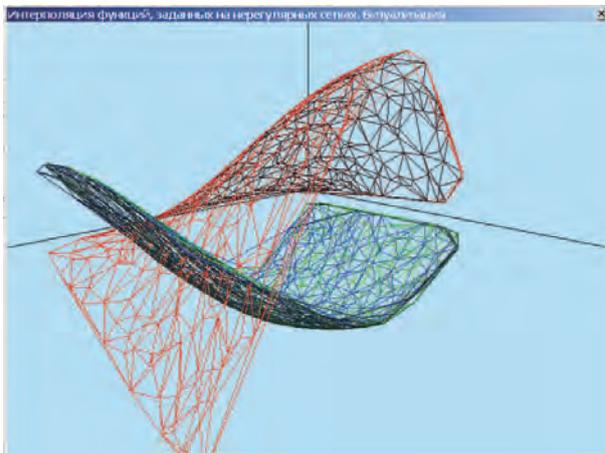


Рис. 2.14. Линейная интерполяция функций, $N_1 = N_2 = 1000$.

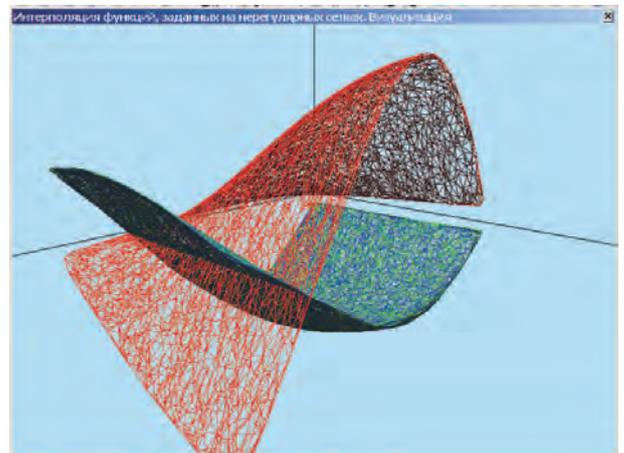


Рис. 2.15. Линейная интерполяция функций, $N_1 = N_2 = 10000$.

Поверхности, изображённые на рисунках, заданы функциями вида

$$f(x, y) = \varphi(x + a)\psi(y + b),$$

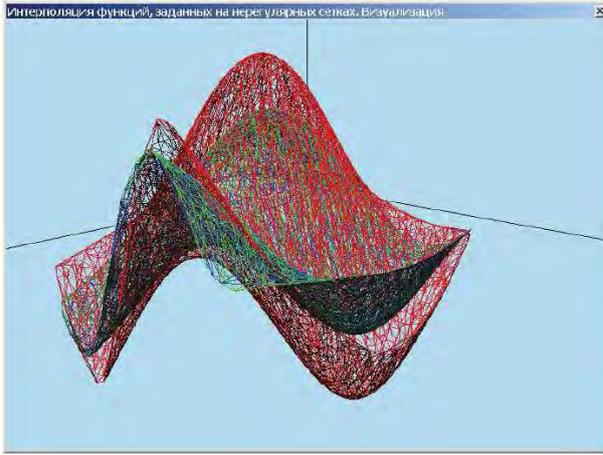


Рис. 2.16. Линейная интерполяция функций, $N_1 = N_2 = 10\,000$.

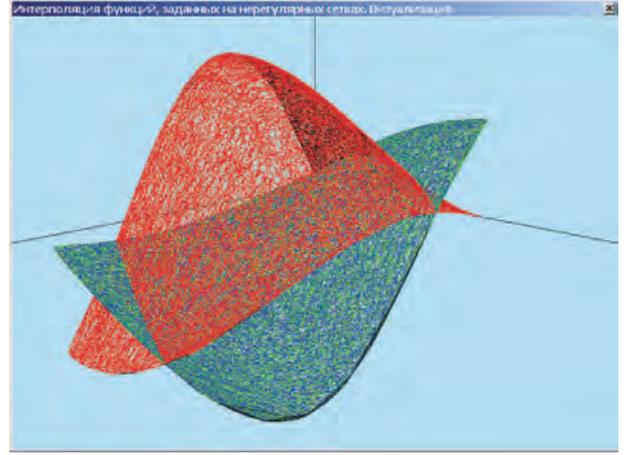


Рис. 2.17. Линейная интерполяция функций, $N_1 = N_2 = 15\,000$.

где $\varphi(x) = \sin\left(\frac{x}{X}k_1\pi\right)$, $\psi(y) = \sin\left(\frac{y}{Y}k_2\pi\right)$, a , b , k_1 , k_2 — константы, X , Y — разброс точек исходного множества по осям координат Ox и Oy соответственно.

С помощью описанного метода значения функции f_1 интерполируются во всех точках двумерной сетки g_2 , а значения функции f_2 — во всех точках двумерной сетки g_1 .

2.2.5. Объединённая триангуляция Делоне

Теперь в каждой точке обеих сеток известны значения двух функций: одно из этих значений было задано, а другое получено в результате интерполяции.

Определение 2.12. Триангуляции T_1 и T_2 называются *перекрывающимися (неразделёнными)*, если пересечение выпуклых оболочек множеств их вершин не пусто, то есть $Conv(g_1) \cap Conv(g_2) \neq \emptyset$.

Построим на узлах обеих сеток триангуляцию Делоне — объединённую триангуляцию. Алгоритм слияния двух неразделённых триангуляций Делоне,

предложенный в [25], обеспечивает слияние за линейное по общему количеству узлов время.

В каждом узле общей триангуляции заданы две функции. Уже на данном этапе мы можем производить некоторые операции над ними — например, операции сложения и вычитания (для этого в каждом узле общей сетки нужно сложить/вычесть значения двух функций).

Отметим, что на данном этапе можно вычислить значения мер $\rho_M(f_1, f_2)$ (2.4) и $\rho_M^\alpha(f_1, f_2)$ (2.5) между поверхностями.

Вычисление меры различия на ОТД $\rho_{V_\mu}(f_1, f_2)$ (2.7) производится на отдельных ячейках (треугольниках) общей триангуляции. Для каждого треугольника считается объём разности между двумя функциями, затем все результаты суммируются. В следующем подразделе описывается метод вычисления этого объёма.

2.2.6. Сравнение поверхностей на треугольной области

Рассмотрим треугольник $\Delta A_0 B_0 C_0$ из ОТД T .

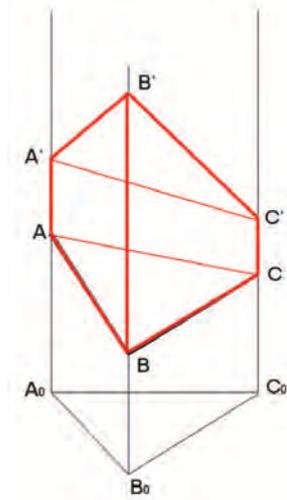


Рис. 2.18. Треугольник общей триангуляции.

Пусть ΔABC ($\Delta A'B'C'$) — треугольник, каждая вершина которого имеет такие же координаты по осям Ox и Oy , что и соответствующая вершина

треугольника $\Delta A_0 B_0 C_0$, а координата по оси Oz равна значению функции f_1 (f_2) в этой вершине (см. рис. 2.18).

Пусть $a = A'_z - A_z$, $b = B'_z - B_z$, $c = C'_z - C_z$.

Для решения задачи необходимо сравнить функции f_1 и f_2 на треугольнике $\Delta A_0 B_0 C_0$, то есть посчитать объём разности между ними. Для этого нужно рассмотреть все возможные случаи взаимного расположения пространственных треугольников ΔABC и $\Delta A'B'C'$.

Базовой фигурой, объём которой требуется вычислять в большинстве случаев, является треугольная пирамида:

Подзадача вычисления объёма треугольной пирамиды по координатам её вершин. Объём пирамиды $P_0 P_1 P_2 P_3$ может быть вычислен по формуле:

$$V_{P_0 P_1 P_2 P_3} = \frac{1}{3} H_{(P_1 P_2 P_3)}^{P_0} S_{\Delta P_1 P_2 P_3},$$

где $H_{(P_1 P_2 P_3)}^{P_0}$ — расстояние от точки P_0 до плоскости $(P_1 P_2 P_3)$, $S_{\Delta P_1 P_2 P_3}$ — площадь треугольника, заданного координатами своих вершин².

Найдём формулу расстояния $H_{(P_1 P_2 P_3)}^{P_0}$ от точки $P(x_0, y_0, z_0)$ до плоскости, заданной тремя точками $P_1(x_1, y_1, z_1)$, $P_2(x_2, y_2, z_2)$, $P_3(x_3, y_3, z_3)$.

Найдём уравнение исходной плоскости в виде $x + By + Cz + D = 0$:

$$\begin{cases} x_1 + By_1 + Cz_1 + D = 0 \\ x_2 + By_2 + Cz_2 + D = 0 \\ x_3 + By_3 + Cz_3 + D = 0 \end{cases} \Leftrightarrow \tilde{A}\mathbf{x} = \mathbf{b},$$

где

$$\tilde{A} = \begin{pmatrix} y_1 & z_1 & 1 \\ y_2 & z_2 & 1 \\ y_3 & z_3 & 1 \end{pmatrix}, \mathbf{x} = \begin{pmatrix} B \\ C \\ D \end{pmatrix}, \mathbf{b} = \begin{pmatrix} -x_1 \\ -x_2 \\ -x_3 \end{pmatrix}.$$

² Может быть вычислена по формуле Герона.

Пусть $\Delta = \det(\tilde{A})$, а Δ_i — определитель матрицы, полученной из \tilde{A} заменой i -ого столбца на вектор-столбец \mathbf{b} . Тогда

$$\Delta = y_1(z_2 - z_3) - y_2(z_1 - z_3) + y_3(z_1 - z_2);$$

$$\Delta_1 = -x_1(z_2 - z_3) + x_2(z_1 - z_3) - x_3(z_1 - z_2);$$

$$\Delta_2 = y_1(-x_2 + x_3) - y_2(-x_1 + x_3) + y_3(-x_1 + x_2);$$

$$\Delta_3 = y_1(-z_2x_3 + x_2z_3) - y_2(-z_1x_3 + x_1z_3) + y_3(-z_1x_2 + x_1z_2).$$

Если $\Delta = 0$, то $z_1 = z_2 = z_3$ ³. В этом случае искомое расстояние равно $|z_0 - z_1|$.

Пусть теперь $\Delta \neq 0$. Воспользуемся формулами Крамера: $\mathbf{B} = \Delta_1/\Delta$, $\mathbf{C} = \Delta_2/\Delta$, $\mathbf{D} = \Delta_3/\Delta$. Запишем уравнение плоскости:

$$x + \frac{\Delta_1}{\Delta}y + \frac{\Delta_2}{\Delta}z + \frac{\Delta_3}{\Delta} = 0; \quad \Delta x + \Delta_1 y + \Delta_2 z + \Delta_3 = 0.$$

Искомое расстояние будет вычисляться по формуле

$$H_{(P_1P_2P_3)}^{P_0} = \frac{\Delta x_0 + \Delta_1 y_0 + \Delta_2 z_0 + \Delta_3}{\sqrt{\Delta^2 + \Delta_1^2 + \Delta_2^2}}. \quad (2.14)$$

В пересечении треугольников получаются разные фигуры в зависимости от знаков чисел a, b, c (всего существует 27 вариантов, так как каждое из трёх чисел может быть больше, меньше или равно нулю). Каждый вариант можно отнести к одному из трёх случаев. Рассмотрим эти случаи подробнее.

I-ый случай. $a \geq 0, b \geq 0, c \geq 0$, либо $a \leq 0, b \leq 0, c \leq 0$.

Искомый объём равен объёму усечённой призмы:

$$V = S_{\Delta A_0 B_0 C_0} \frac{|a| + |b| + |c|}{3}. \quad (2.15)$$

На рис. 2.19 отображены случаи, которые являются частными для рассматриваемого. Формула (2.15) подходит для расчёта объёма и в этих случаях.

³ y_1, y_2, y_3 не могут совпадать, так как это ординаты вершин треугольника триангуляции; первые два столбца матрицы также линейно независимы.

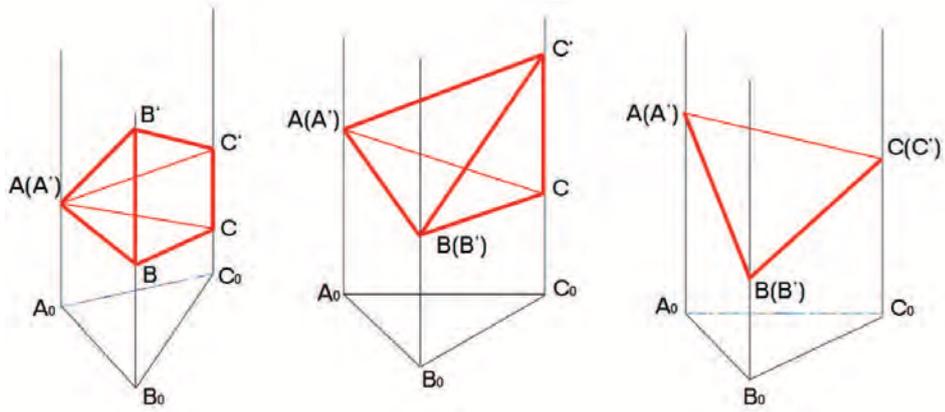


Рис. 2.19. Частные случаи для I-ого: $a = 0, b > 0, c > 0$ (слева); $a = 0, b = 0, c > 0$ (в центре); $a = 0, b = 0, c = 0$ (справа).

II-ой случай. Два числа из $\{a, b, c\}$ имеют один знак, а третье — другой.

Не ограничивая общности рассуждений, будем считать, что $a > 0, c > 0, b < 0$. Пусть MN — отрезок, по которому пересекаются $\triangle ABC$ и $\triangle A'B'C'$ (см. рис. 2.20).

В рассматриваемом случае искомый объём будет равен сумме объёмов пирамиды и пятигранника: $V = V_{BMNB'} + V_{AA'MCC'N}$.

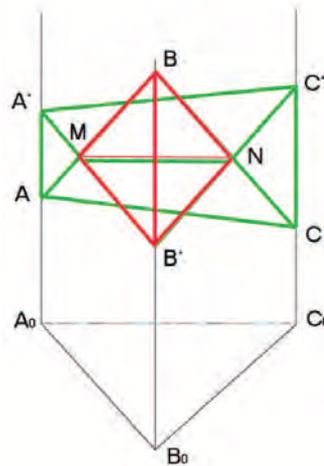


Рис. 2.20. $a > 0, b < 0, c > 0$.

Координаты точек M и N находятся по формулам:

$$M_t = \frac{A_t|a| + B_t|b|}{|a| + |b|}, \quad N_t = \frac{B_t|b| + C_t|c|}{|b| + |c|}, \quad t \in \{x, y, z\}.$$

Объём пятигранной фигуры $AA'MCC'N$ равен сумме объёмов трёх треугольных пирамид (см. рис. 2.21): $V_{AA'MCC'N} = V_{MAA'C} + V_{MAC'C} + V_{MNCC'}$.

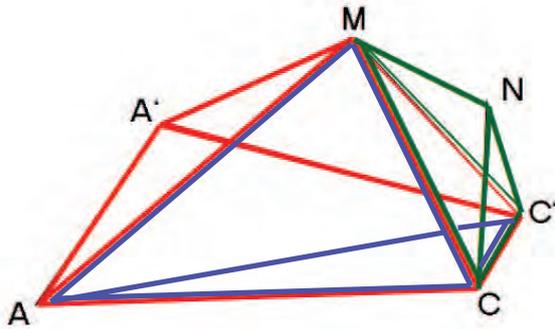


Рис. 2.21. Пятигранник $AA'MCC'N$.

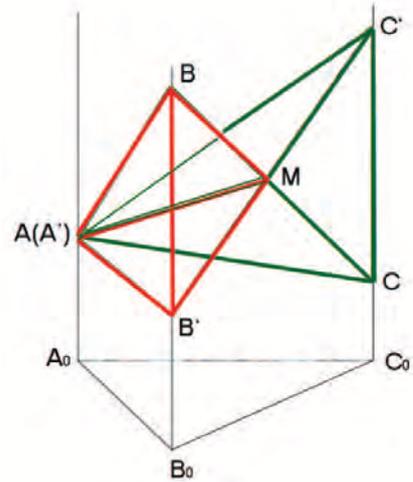


Рис. 2.22. $a = 0, b < 0, c > 0$.

Таким образом, искомый объём разности для рассматриваемого случая можно вычислить по формуле:

$$V = \frac{1}{3} \left(H_{(MNB')}^B S_{\Delta MNB'} + H_{(AA'C)}^M S_{\Delta AA'C} + H_{(AC'C)}^M S_{\Delta AC'C} + H_{(NCC')}^M S_{\Delta NCC'} \right). \quad (2.16)$$

III-ий случай. Одно из чисел $\{a, b, c\}$ равно 0, а два другие имеют разные знаки.

В этом случае искомый объём равен сумме объёмов двух треугольных пирамид $BAMB'$ и $MAC'C$ (см. рис. 2.22) и вычисляется по формуле:

$$V = \frac{1}{3} \left(H_{(AMB')}^B S_{\Delta AMB'} + H_{(AC'C)}^M S_{\Delta AC'C} \right). \quad (2.17)$$

Таким образом, объём разности между поверхностями на каждом треугольнике ОТД T вычисляется по одной из формул (2.15)–(2.17), где H вычисляется по формуле (2.14) из рассмотренной подзадачи.

2.2.7. Оценка вычислительной сложности алгоритма

Оценим трудоёмкость предложенного алгоритма сравнения поверхностей. Будем обозначать через \mathbf{A}^ρ алгоритм, вычисляющий значение меры ρ между поверхностями. Оценим сложность каждого из этапов алгоритма \mathbf{A}^ρ :

1. Построение триангуляций Делоне T_1 и T_2 на сетках g_1 и g_2 соответственно $\sim O(N_1 \log N_1) + O(N_2 \log N_2)$;
2. Построение МОД для триангуляций $T_1, T_2 \sim O(N_1) + O(N_2)$ (алгоритм Черитона и Тарьяна);
3. Локализация узлов сеток g_1, g_2 в триангуляциях T_2, T_1 соответственно на основе обхода МОД $\sim O(N)$ в среднем при равномерном распределении узлов сеток g_1, g_2 и условии $N_1/N_2 \leq c = \text{const}$ (теорема 2.9);
4. Интерполяция значений функций f_1, f_2 в узлах сеток g_2, g_1 соответственно на основе локализации $\sim O(N)$;
5. Вычисление меры $\rho \sim O(N)$.

Таким образом, при условиях равномерности распределения узлов исходных сеток и ограниченности отношения их мощностей (что приемлемо в большинстве приложений), каждый из этапов, за исключением построения триангуляций, выполняется за линейное по числу узлов в сетках время. Сложность алгоритма построения триангуляции Делоне составляет $O(N \log N)$ и является определяющей для предложенного подхода.

Алгоритм \mathbf{A}^ρ . Вычисление меры различия ρ между поверхностями.

Вход:

g_1, g_2 — двумерные сетки, заданные множествами узлов (2.2);

f_1, f_2 — поверхности, заданные значениями высот (2.3) в узлах сеток.

Выход:

значение меры ρ для f_1 и f_2 .

- 1: $T_1 :=$ Построить_триангуляцию_Делоне(g_1);
 - 2: $T_2 :=$ Построить_триангуляцию_Делоне(g_2);
 - 3: $MST_1 :=$ Построить_МОД(T_1);
 - 4: $MST_2 :=$ Построить_МОД(T_2);
 - 5: $g_1 :=$ Локализовать_сетку_в_триангуляции(g_1, T_2, MST_1);
 - 6: $g_2 :=$ Локализовать_сетку_в_триангуляции(g_2, T_1, MST_2);
 - 7: $\hat{f}_1 :=$ Интерполировать_функцию_в_сетке(f_1, g_2);
 - 8: $\hat{f}_2 :=$ Интерполировать_функцию_в_сетке(f_2, g_1);
 - 9: $\rho :=$ Вычислить_меру(\hat{f}_1, \hat{f}_2);
 - 10: **вернуть** ρ ;
-

ПРОЦЕДУРА Вычислить_меру(\hat{f}_1, \hat{f}_2) {для меры ρ_V (2.9)}

- 1: $T :=$ Объединить_триангуляции(T_1, T_2);
 - 2: $S_{Conv(g)} := 0$;
 - 3: $\rho := 0$;
 - 4: **для всех** $\triangle ABC \in T$
 - 5: $\rho := \rho +$ Вычислить_объём_разности($f_1, f_2, \triangle ABC$);
 - 6: $S_{Conv(g)} := S_{Conv(g)} + S_{\triangle ABC}$;
 - 7: **вернуть** $\rho/S_{Conv(g)}$;
-

ПРОЦЕДУРА Вычислить_меру(\hat{f}_1, \hat{f}_2) {для меры ρ_M (2.4)}

- 1: $\rho := 0$;
- 2: **для всех** $i = 1, \dots, N_1$
- 3: $\rho := \rho + |\hat{f}_1(x_1^i, y_1^i) - \hat{f}_2(x_1^i, y_1^i)|$;
- 4: **для всех** $i = 1, \dots, N_2$
- 5: $\rho := \rho + |\hat{f}_1(x_2^i, y_2^i) - \hat{f}_2(x_2^i, y_2^i)|$;
- 6: $\rho := \rho/(N_1 + N_2)$;
- 7: **вернуть** ρ ;

ПРОЦЕДУРА Вычислить $_меру(\hat{f}_1, \hat{f}_2, \alpha)$ {для меры ρ_M^α (2.5)}

- 1: для всех $i = 1, \dots, N_1$
 - 2: $Dist[i] := |\hat{f}_1(x_1^i, y_1^i) - \hat{f}_2(x_1^i, y_1^i)|;$
 - 3: для всех $i = 1, \dots, N_2$
 - 4: $Dist[i + N_1] := |\hat{f}_1(x_2^i, y_2^i) - \hat{f}_2(x_2^i, y_2^i)|;$
 - 5: $k := \lfloor (1 - \alpha)(N_1 + N_2) \rfloor;$
 - 6: $\rho :=$ Найти $_порядковую_статистику(Dist, k)$ ⁴;
 - 7: вернуть ρ ;
-

2.2.8. Программная реализация

В рамках исследований была разработана детальная программная реализация предложенного подхода к сравнению поверхностей объектов. В данном подразделе рассмотрены некоторые вопросы выбора структуры данных для разработанной программной реализации.

2.2.8.1. Выбор структур данных

Вопрос о выборе структуры данных для триангуляции и минимального остовного дерева очень важен, так как он оказывает существенное влияние на трудоёмкость алгоритмов, использующих данную структуру.

Для хранения триангуляции Делоне использована структура данных «Узлы с соседями». Для реализации очереди деревьев в алгоритме построения минимального остовного дерева использованы фибоначчиевы кучи. Далее приводится описание полей и методов класса минимального остовного дерева (MST) и некоторых других, которые необходимы для корректной работы алгоритма:

⁴ k -ой порядковой статистикой массива называется k -ый элемент этого массива, упорядоченного по неубыванию. Существует алгоритм её поиска, работающий за $O(N)$ в худшем случае [18].

```

class Queue          // класс очередь
{
    int size;        // текущий размер
    // массив элементов, реализованный в виде фибоначчиевой кучи
    FibonacciHeap FHeap;
    void Insert(Tree T); // добавить дерево в конец очереди
    void Remove(Tree T); // удалить дерево из очереди
    Tree Get();         // взять дерево из начала очереди
    // вернуть дерево из начала очереди, не удаляя его оттуда
    Tree Peek();
    void Clear();      // очистить дерево
    Queue();           // конструктор
}

class Edge           // ребро
{
    Vertex orig, dest; // инцидентные вершины
    double length;     // длина
    bool in_MST;       // флаг принадлежности МОД
    Edge();            // конструктор
    Edge(Vertex v1, Vertex v2); // конструктор
}

class SuperVertex   // супервершина
{
    Edge[] edges;      // массив инцидентных рёбер
    Tree T; // дерево в очереди, которому супервершина принадлежит
    Vertex[] vertices; // массив объединяемых вершин
    SuperVertex(Vertex vert, Tree T); // конструктор
}

```

```

}

class Tree                // дерево
{
    int stage;            // номер этапа
    int unique;           // уникальный номер дерева
    // массив вершин, принадлежащих дереву
    SuperVertex[] Points;
    // объединить данное дерево с деревом T
    void Combine(Tree T);
    // добавить к дереву ребро ab (вершина a принадлежит дереву)
    void AddEdge(Vertex a, Vertex b);
    // вернуть минимальное ребро, соединяющее дерево с деревом T
    Edge GetMinEdge(Tree T);
    // превратить дерево в супервершину
    SuperVertex TransformIntoSuperVertex();
    Tree(SuperVertex v); // конструктор
}

class MST                 // МОД
{
    Queue Forest;        // лес деревьев
    // построить МОД триангуляции
    bool BuildMST(Triangulation Triang);
    bool Clean();        // произвести операцию очистки
    MST();               // конструктор
}

```

2.2.8.2. Фибоначчиевы кучи

Для реализации очереди деревьев выбрана структура данных *фибоначчиева куча* (Fibonacci heap), введённая Фредманом и Тарьяном в 1984 году.

В данном подразделе кратко описаны строение фибоначчиевой кучи и поддерживаемые ей операции с оценкой их сложности. Более подробное описание фибоначчиевых куч и доказательство вычислительной сложности операций представлено в [18].

Пример фибоначчиевой кучи приведён на рис. 2.23.

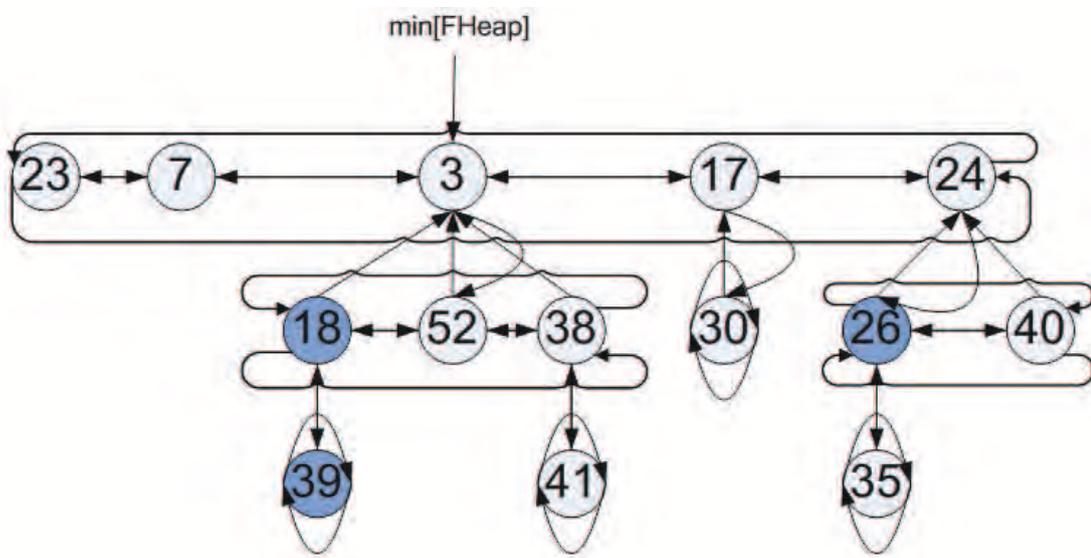


Рис. 2.23. Строение фибоначчиевой кучи. Стрелки показывают значения полей p , $child$, $left$ и $right$. Три отмеченные вершины (для которых $mark = \text{TRUE}$) выделены тёмным.

Каждый элемент кучи имеет поле key (ключ), в котором хранится некоторое число, уникальное для каждого дерева из леса F . Фибоначчиева куча состоит из нескольких деревьев. В каждом из деревьев, входящих в кучу, выполнено следующее свойство: ключ каждой вершины не превосходит ключей её детей. Каждая вершина x содержит указатель $p[x]$ на своего родителя и указатель $child[x]$ на какого-нибудь из своих детей. Дети каждой вершины образуют двусторонний циклический список — список детей этой вершины.

Каждая вершина y этого списка имеет поля $left[y]$ и $right[y]$, указывающие на её соседей в списке. Такие списки удобны, так как из них можно удалить любую вершину за время $O(1)$, и два таких списка можно объединить в один за время $O(1)$.

Кроме указанной информации, каждая вершина имеет поле $degree[x]$ (степень), где хранится число её детей, и поле $mark[x]$ с булевым значением, которое истинно, если данная вершина потеряла ребёнка после того, как сама сделалась чьим-то ребёнком.

Корни деревьев, составляющих фибоначчиеву кучу, объединены в двунаправленный циклический список, называемый *корневым списком* кучи. Доступ к фибоначчиевой куче $FHeap$ осуществляется с помощью указателя $min[FHeap]$ на минимальную вершину — вершину корневого списка с минимальным ключом.

Фибоначчиевы кучи поддерживают следующие операции:

$Make_Heap()$ — создание новой кучи;

$Insert(FHeap, x)$ — добавление вершины x в кучу $FHeap$;

$Minimum(Fheap)$ — получение указателя на минимальную вершину;

$Extract_Min(FHeap)$ — изъятие элемента с минимальным ключом;

$Decrease_Key(Fheap, x, k)$ — уменьшение ключа вершины x и присвоение ему значения k (предполагается, что новое значение не превосходит старого);

$Delete(FHeap, x)$ — удаление вершины x из кучи $FHeap$;

Все перечисленные операции, кроме $Extract_Min$ и $Decrease_Key$, имеют вычислительную сложность $O(1)$. Операции $Extract_Min$ и $Decrease_Key$ выполняются за время $O(\log N)$. Операция $Extract_Min$ вызывает функцию $Consolidate(FHeap)$ — уплотнение корневого списка кучи. Уплотнение происходит за счёт того, что два дерева с корнями одинаковой степени соединяются в одно. Именно за счёт вызова $Consolidate$ достигается вычислительная сложность операций. Уплотнение происходит до тех пор, пока в корневом списке

останется не более одного дерева каждой степени. Таким образом, операция Consolidate не даёт корневому списку «разрастаться».

2.2.9. Результаты экспериментов

Для подтверждения корректности программной реализации и оценки вычислительной сложности предложенного алгоритма были проведены вычислительные эксперименты⁵ для задач большой размерности (10 000 – 100 000). Полученные результаты приведены в таблицах 2.2–2.6 и на графиках 2.25, 2.24, 2.26. Массив исходных точек для экспериментов был получен случайным выбором в прямоугольнике с равномерным распределением вероятности, количество точек в исходных сетках было одинаковым.

Количество точек (N)	Время, сек. (t_1) точки с целыми координатами	$N \log N$	$N \log N/t_1$
10 000	0,265	40 000	150 943,4
50 000	1,656	234 948,5	141 877,1
100 000	3,718	500 000	134 480
500 000	21,515	2 849 485	132 441,8
1 000 000	45,984	6 000 000	135 480,2

Таблица 2.1. Время построения триангуляции Делоне методом «разделяй и властвуй».

В таблицах 2.2, 2.1, 2.4 и 2.5 значения в последнем столбце приводятся для экспериментального подтверждения теоретической оценки сложности этапов алгоритма.

В таблице 2.6 представлен расход времени на каждый из этапов алгоритма при сравнении двух поверхностей, каждая из которых содержит 3 000

⁵ Все вычислительные эксперименты из подраздела 2.2.9 проводились на машине с процессором AMD Athlon 2600+ с оперативной памятью 512 Мб.

точек.

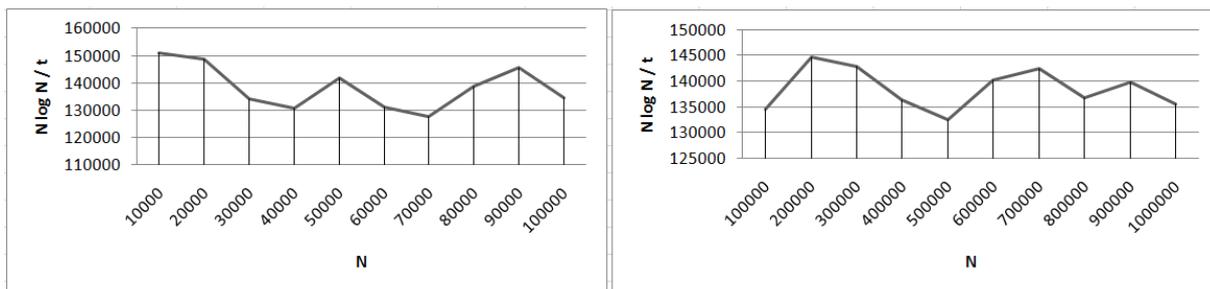


Рис. 2.24. График зависимости $N \log N / t$ от N , где N — общее количество точек в исходных сетках, t — время на построение триангуляции Делоне.

Количество точек (N)	Время, сек. (t) на построение МОД	N/t
10 000	0,453	22 075
50 000	2,156	23 191
100 000	4,468	22 381

Таблица 2.2. Время построения двух МОД триангуляций Делоне алгоритмом Черитона и Тарьяна с использованием фибоначчиевых куч.

Результаты экспериментов подтвердили, что

— сложность реализованного алгоритма построения триангуляции Делоне есть $O(N \log N)$ (см. таблицу 2.1, график 2.24);

— сложность алгоритма построения минимального остова триангуляции Делоне есть $O(N)$ (см. таблицу 2.2, график 2.25);

— сложность локализации узлов сеток в триангуляциях друг друга предложенным алгоритмом с помощью обхода МОД есть $O(N)$, что экспериментально подтверждает теорему 2.9 (см. таблицу 2.4, график 2.26);

— время интерполяции значений функций в узлах сеток на основе локализации есть $O(N)$ (см. таблицу 2.5).

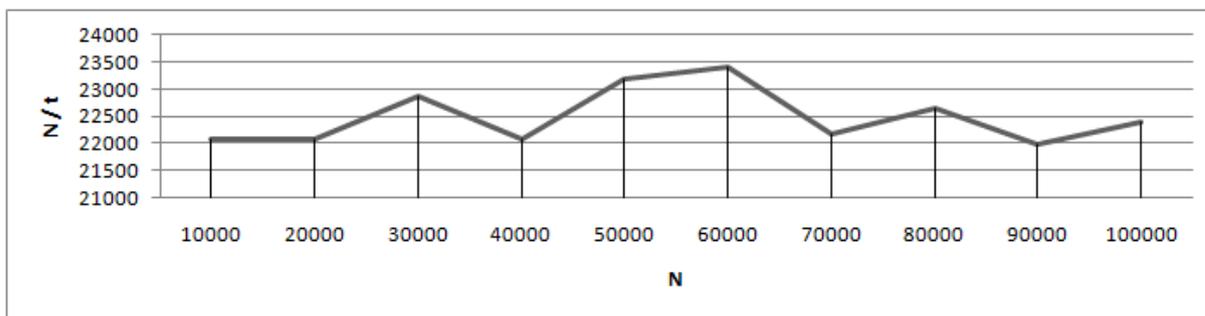


Рис. 2.25. График зависимости N/t от N , где N — количество исходных точек, t — время построения МОД исходного множества точек по его триангуляции Делоне.

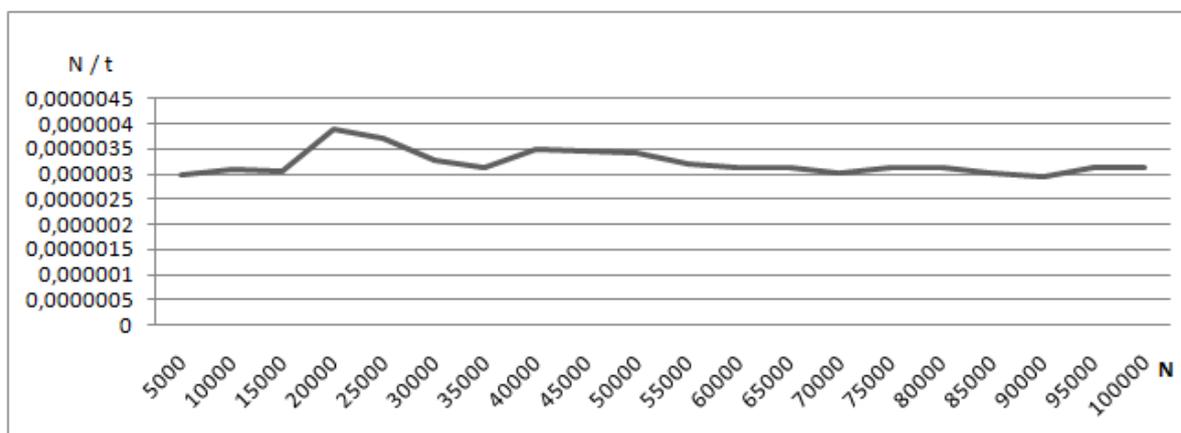


Рис. 2.26. График зависимости N/t от N , где $N \approx N_1 \approx N_2$ — количество точек в исходных сетках, t — время локализации узлов триангуляций друг в друге.

2.3. Алгоритм выделения интерфейсных граней

При объединении триангуляций Делоне T_1 и T_2 некоторые рёбра и треугольники перейдут без изменений в объединённую триангуляцию T , а некоторые разрушатся. Таким образом, в T будут присутствовать новые рёбра и треугольники, которые будут соединять узлы из разных сеток. Такие рёбра и треугольники (границы) мы назвали *интерфейсными* (см. определение 2.11), так как они соединяют узлы, соответствующие разным триангуляциям. На рис. 2.27 все интерфейсные границы, получившиеся при объединении триангуляций, закрашены.

Мера (2.10) вычисляется только по интерфейсным границам.

Количество точек (N)	Время, сек. (t_1) точки с целыми координатами	Время, сек. (t_2) точки с вещественными координатами
10 000	0,265	0,328
50 000	1,656	2,109
100 000	3,718	4,725
200 000	7,328	10,25
300 000	11,5	15,812
400 000	16,421	22,156
500 000	21,515	29,375

Таблица 2.3. Сравнение времени построения триангуляций на наборах точек с целыми и вещественными координатами.

Следует отметить, что в случае, когда точки обеих сеток распределены равномерно и с одинаковой плотностью (т.е. точки двух сеток хорошо «перемешаны» между собой), все или почти все грани общей триангуляции могут быть интерфейсными. Тогда меры сравнения (2.7) и (2.10) будут примерно равны между собой. Поэтому преимущество меры (2.10) проявляется в тех случаях, когда существуют значимые по площади области, в которых одна из функций представлена более детально, чем другая (см. рис. 2.28). В этих случаях точки сеток, на которых заданы функции, будут распределены неравномерно, либо равномерно, но с разной плотностью. На практике с таким случаем можно встретиться при решении задачи подгонки друг к другу поверхностей, полученных 3d сканерами различной точности.

Множество интерфейсных граней разбивается на несколько непересекающихся подмножеств, каждое из которых является *цепочкой* из смежных по рёбрам треугольников: либо *замкнутой* (циклической), в которой все интер-

Количество точек сетки G_1 (N_1)	Время, сек. (t) точки с 4 знаками после запятой	N_1/t
10 000	0,031	$3,1 \cdot 10^{-6}$
25 000	0,093	$3,72 \cdot 10^{-6}$
50 000	0,171	$3,42 \cdot 10^{-6}$
75 000	0,234	$3,12 \cdot 10^{-6}$
100 000	0,312	$3,12 \cdot 10^{-6}$

Таблица 2.4. Время локализации одной сетки в треугольниках триангуляции другой сетки.

Количество точек сеток G_1 и G_2 ($N_1 + N_2$)	Время, сек. (t)	$(N_1 + N_2)/t$
50 000	0,015	3 333 333,33
100 000	0,031	3 225 806,45
150 000	0,046	3 260 869,56
200 000	0,062	3 225 806,45

Таблица 2.5. Расход времени на линейную интерполяцию обеих поверхностей.

фейсные рёбра являются внутренними рёбрами общей триангуляции T , либо *разомкнутой*, в которой крайние треугольники имеют хотя бы одну граничную сторону, т. е. сторону выпуклой оболочки $Conv(g)$ (см. рис. 2.29).

Поэтому алгоритм выделения всех интерфейсных граней сводится к прослеживанию цепочек из таких граней. Таким образом, *задача прослеживания цепочек интерфейсных граней* состоит в построении списка всех таких цепочек по двум исходным триангуляциям.

Определим понятия связности элементов триангуляции.

Определение 2.13. Множество узлов и рёбер в триангуляции называется

Этап алгоритма	Время, сек. (t)
Построение триангуляций T_1 и T_2	0,124
Построение МОД MST_1 и MST_2	0,203
Локализация триангуляций друг в друге	0,015
Интерполяция функций	<0,001
Объединение триангуляций	0,093
Вычисление меры ρ_V	0,031
Общее время	0,467

Таблица 2.6. Расход времени для этапов алгоритма. $N_1 = N_2 = 3000$.

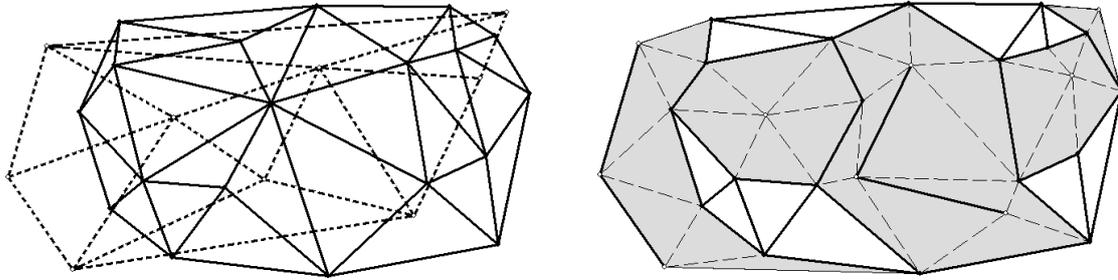


Рис. 2.27. Две триангулированные сетки (слева) и объединённая триангуляция Делоне с закрасненными интерфейсными гранями (справа).

связным, если для любой пары входящих в него узлов существует цепь из попарно инцидентных узлов и рёбер. Множество рёбер и граней называется *связным*, если для каждой пары входящих в него рёбер существует цепь из попарно инцидентных рёбер и граней.

Определение 2.14. Максимальные связные подмножества узлов и рёбер, перешедшие в T из исходной триангуляции T_1 (или T_2) без изменений, будем называть *лоскутами* триангуляции T_1 (или T_2).

Лоскут является связным подграфом триангуляции. При объединении двух триангуляций Делоне, все рёбра, не вошедшие в лоскуты, разрушаются. Интерфейсные рёбра и грани «сшивают» лоскуты из разных триангуляций.

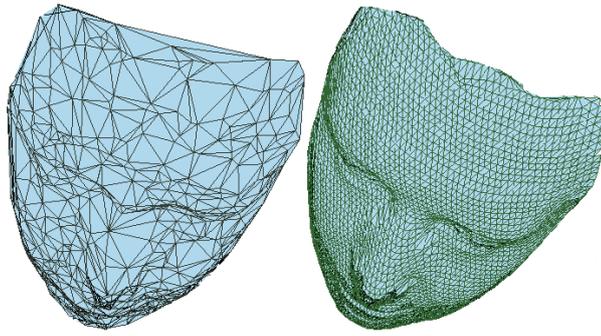


Рис. 2.28. Пример поверхностей человеческого лица разных уровней детализации.

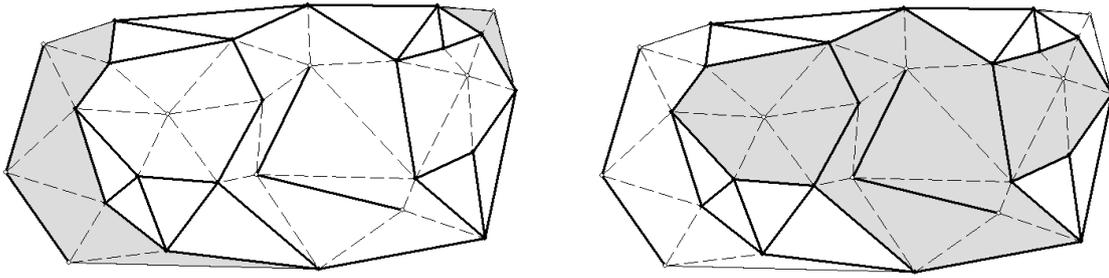


Рис. 2.29. Две разомкнутые (слева) и три замкнутые цепочки интерфейсных граней (справа).

Лоскуты могут иметь простую форму: состоять только из одного узла или из цепочки узлов и рёбер, не содержащей циклов (см. рис. 2.30). Будем называть такие лоскуты, множество рёбер которых не образует ни одной грани исходной триангуляции, *простыми лоскутами*.

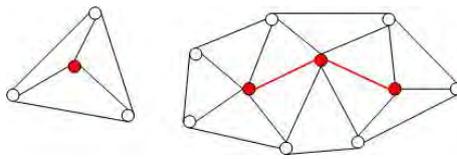


Рис. 2.30. Примеры простых лоскутов.

На рис. 2.31 представлен лоскут более сложной формы, который содержит несколько треугольных граней исходной триангуляции и не является простым.

Будем называть упорядоченное замкнутое множество попарно инцидентных узлов и рёбер лоскута, инцидентных интерфейсным треугольникам, *кон-*

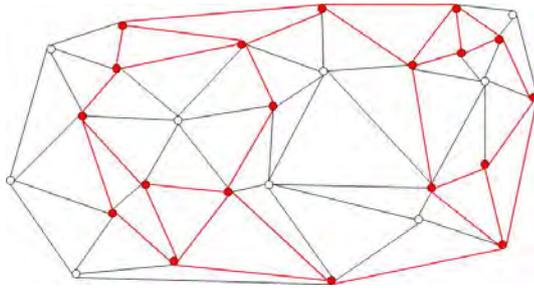


Рис. 2.31. Общая триангуляция Делоне, лоскут выделен цветом.

туром лоскута. Каждому ребру лоскута, принадлежащему его контуру, соответствует одна или две цепочки интерфейсных граней.

Локализовать лоскут C в общей триангуляции T означает построить множество L всех интерфейсных треугольников триангуляции T , инцидентных каждому контуру этого лоскута. Будем говорить, что любой треугольник из L *участвует в локализации* лоскута C , множество всех треугольников из L *локализует* лоскут C , а лоскут C *локализуется* с помощью множества интерфейсных треугольников L .

Рассмотрим *задачу локализации лоскутов в триангуляции* T . Даны исходные триангуляции T_1, T_2 , требуется локализовать все лоскуты обеих триангуляций.

Докажем справедливость следующего утверждения:

Утверждение 2.10. *Задача прослеживания цепочек интерфейсных граней линейно сводится к задаче локализации лоскутов в триангуляции.*

Доказательство. Пусть построен список всех цепочек интерфейсных граней. Докажем, что с помощью него можно за линейное время локализовать все лоскуты обеих триангуляций.

Заметим, что верно следующее:

— Если хотя бы одна интерфейсная грань из цепочки участвует в локализации лоскута, то все грани этой цепочки участвуют в локализации этого лоскута.

— Каждая замкнутая цепочка интерфейсных граней полностью локализует один лоскут.

— Любой простой лоскут однозначно локализуется с помощью одной цепочки интерфейсных граней: замкнутой, если лоскут не принадлежит выпуклой оболочке узлов общей триангуляции, и разомкнутой в противном случае.

Выбирая рёбра контура лоскута и соответствующие им цепочки интерфейсных граней, можно произвести локализацию всего лоскута.

При таком подходе каждый интерфейсый треугольник будет просмотрен ровно два раза, так как он соединяет одну вершину одной триангуляции и две вершины другой, а значит участвует в локализации двух лоскутов, что доказывает утверждение. ■

Определение 2.15. *Интерфейсное ребро общей триангуляции Делоне T , принадлежащее ещё не прослеженной цепочке граней, будем называть стартером.*

Для каждой цепочки существует стартер, который инициализирует процесс прослеживания цепочки интерфейсных граней.

Тогда алгоритм выделения интерфейсных граней состоит из следующих этапов:

1. Поиск начального стартера;
2. Прослеживание цепочки граней, соответствующей найденному стартеру;
3. Поиск очередного стартера. Если стартер найден, перейти к предыдущему пункту, иначе закончить.

Рассмотрим каждый из этапов подробнее.

2.3.1. Поиск начального стартера

Будем считать, что узлы сеток g_1 и g_2 упорядочены лексикографически по координатам (x, y) , тогда $g_1[0]$ и $g_2[0]$ — самые левые узлы сеток. Пусть $g_1[0] \prec g_2[0]$.

Для поиска *начального стартера* рассмотрим окружность, проходящую через узлы $g_1[0]$, $g_2[0]$ с центром на горизонтальном луче, выходящем из $g_2[0]$ влево. Если $g_1[0].x = g_2[0].x$, то окружность вырождается в прямую линию, по одну сторону от которой нет узлов триангуляции. Окружность называется *пустой* относительно сетки, если она не содержит внутри себя узлов этой сетки. По построению рассматриваемая окружность пуста относительно g_2 , однако внутри неё могут находиться узлы из g_1 . Проведём через каждый из таких узлов и $g_2[0]$ окружности с центрами на том же горизонтальном луче и выберем такой узел $g_1[k]$, который лежит на окружности минимального радиуса (см. рис. 2.32). Эта окружность будет пуста относительно сеток g_1 , g_2 , а значит и g . Поэтому для ребра, соединяющего $g_1[k]$ и $g_2[0]$, выполняется *условие Делоне* — условие существования пустой окружности, проходящей через концевые точки ребра, и оно является ребром общей триангуляции Делоне T . Ребро $g_1[k]g_2[0]$ будет начальным стартером, так как выполняются условия следующей теоремы.

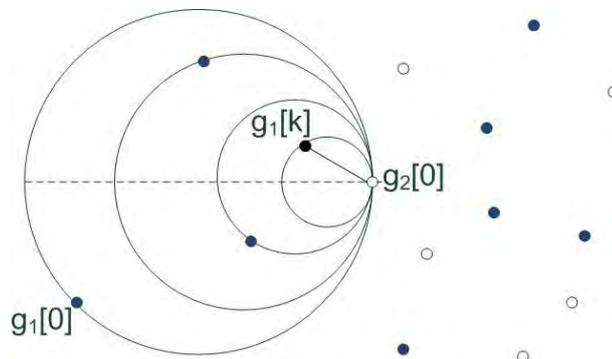


Рис. 2.32. Поиск начального стартера.

Теорема 2.11 (критерий существования стартера) Узел сетки g_1 (g_2)

имеет инцидентное интерфейсное ребро общей триангуляции Делоне, построенной на сетке $g_1 \cup g_2$, тогда и только тогда, когда существует инцидентная этому узлу окружность, пустая относительно остальных узлов из g_1 (g_2), но содержащая внутри себя или проходящая через хотя бы один узел противоположной сетки g_2 (g_1).

Доказательство. Необходимость. Если два узла из разных сеток g_1 и g_2 образуют интерфейсное ребро Делоне, то по определению они имеют пустую инцидентную окружность, удовлетворяющую условиям теоремы.

Достаточность. Пусть для некоторого узла $V \in g_1$ существует инцидентная ему окружность, содержащая внутри и на своей границе узлы $U_1, \dots, U_m \in g_2$, но пустая относительно остальных узлов из g_1 . Рассмотрим множество из m вложенных окружностей, инцидентных парам узлов V и U_i , $i = 1, \dots, m$ и имеющих в V общую с исходной окружностью касательную. Окружность из этого множества, имеющая минимальный радиус, инцидентна узлам из двух разных сеток и является пустой относительно $g_1 \cup g_2$. Значит она образует интерфейсное ребро Делоне и может быть выбрана в качестве стартера.

Теорема доказана. ■

Время поиска начального стартера складывается из времени на вычисление самых левых узлов сеток g_1 и g_2 и времени однократного просмотра узлов сетки g_2 при поиске окружности минимального радиуса. Таким образом, оно линейно по числу узлов в объединённой сетке.

2.3.2. Прослеживание цепочки интерфейсных граней

Алгоритм прослеживания сходен с алгоритмом построения швов при объединении неразделённых триангуляций Делоне [24, 25].

При прослеживании цепочки происходит последовательное выделение

интерфейсных рёбер. Интерфейсное ребро, выделенное на определённом шаге прослеживания, объявляется *текущим*. Первым текущим ребром становится найденный стартер.

Пусть AB — текущее ребро. При присоединении рёбер AB и BA к пучкам узлов A и B соответственно, рёбра этих пучков, для которых не выполнено условие Делоне в объединённой сетке, должны быть разрушены. Поэтому требуется процедура *коррекции пучков* интерфейсного ребра (см. рис. 2.33), основанная на проверке условия Делоне по угловому критерию, описанному в подразделе 2.2.2.

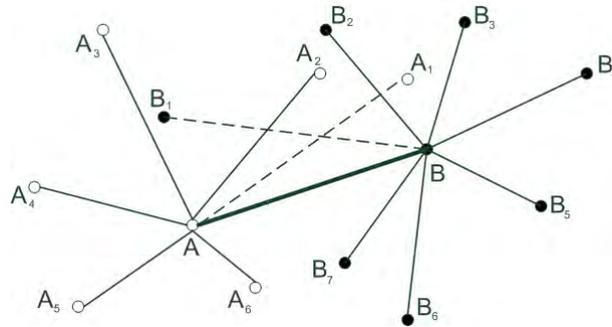


Рис. 2.33. Коррекция пучков интерфейсного ребра AB . Ребра AA_1 и BB_1 , показанные пунктиром, разрушаются.

Процесс прослеживания цепочки интерфейсных граней по найденному стартеру состоит в следующем:

- а) объявить стартер текущим ребром;
- б) произвести в триангуляциях T_1 и T_2 коррекцию рёбер, инцидентных концевым узлам текущего ребра — удаляем рёбра, для которых не выполнено условие Делоне;
- в) построить новое ребро, инцидентное текущему, соединяющее узлы разных сеток;

- г) удалить рёбра триангуляций T_1 и T_2 , пересекающие треугольную грань, образованную текущим и новым рёбрами;
- д) объявить построенное новое ребро текущим, перейти на шаг (б).

Если на шаге (в) новое ребро совпадает со стартером, то цепочка прослеживаемых граней замкнутая; а если новое ребро построить не удалось, то цепочка разомкнутая, и текущее ребро является концевым ребром прослеживаемой цепочки. В последнем случае следует продолжить процесс прослеживания цепочки от стартера в обратную сторону до тех пор, пока не будет найдено второе концевое ребро цепочки.

2.3.3. Поиск стартеров

Поиск остальных стартеров осуществляется с помощью МОД триангуляций Делоне T_1 и T_2 , так как справедливо следующее утверждение:

Утверждение 2.12. *Если в процессе прослеживания цепочки интерфейсных граней нарушается связность триангуляции Делоне T_1 (T_2), то среди разрушенных рёбер присутствует ребро МОД.*

Будем называть рёбра МОД триангуляций T_1 и T_2 , удалённые в процессе прослеживания, *мостами*, а окружность, диаметром которой является мост — *окружностью влияния моста*. Один из концевых узлов моста принадлежит уже прослеженной интерфейсной грани, а второй узел остаётся *свободным*. Справедлива следующая лемма [26]:

Лемма 2.13. *Для каждого свободного узла существует инцидентный ему стартер.*

Таким образом, при разрушении моста образуется свободный узел, для которого существует инцидентный стартер, инициализирующий процесс прослеживания цепочки интерфейсных граней.

Поиск второго узла искомого стартера можно производить, используя следующую лемму [26]:

Лемма 2.14. Пусть B — свободный узел, A — узел триангуляции Делоне T , пара AB образует стартер. Тогда узел B попадёт внутрь хотя бы одной из максимальных пустых окружностей в T , инцидентных узлу A .

Поиск стартера по свободному узлу A и мосту AB состоит из следующих шагов:

- а) поиск прослеженной интерфейсной грани, внутри описанной окружности которой находится свободный узел A (перебором вдоль AB);
- б) поиск интерфейсных граней, смежных с найденной гранью, внутри описанных окружностей которых находится свободный узел A ;
- в) построение множества $D = \{D_1, \dots, D_n\}$ узлов найденных граней, расположенных внутри окружности влияния моста AB ;
- г) для каждого узла $D_i \in D$ построить окружность, проходящую через узлы D_i и A с центром на AB , и выбрать узел D_{i^*} , соответствующий окружности минимального радиуса (аналогично поиску начального стартера). Ребро $D_{i^*}A$, соединяющее выбранный узел со свободным, является новым стартером.

2.3.4. Оценка вычислительной сложности алгоритма

По определению мост является ребром МОД. Вследствие леммы 2.4, окружность влияния моста пуста относительно концевых и серединных точек других мостов. С помощью этого свойства оценивается мера пересечения моста и окружности влияния другого моста. Такие оценки получены в [26]. Они основываются на следующих леммах 2.15–2.18:

Лемма 2.15. *Мост может вырезать из окружности влияния другого моста дугу размером не более 60° .*

Лемма 2.16. *Если два моста AA_1 и BB_1 триангуляции Делоне T_1 пересекают ребро PQ триангуляции Делоне T_2 и концевая точка P ребра попадает внутрь обеих окружностей влияния мостов, то разность углов $\angle APA_1$ и $\angle BPB_1$ не меньше 60° .*

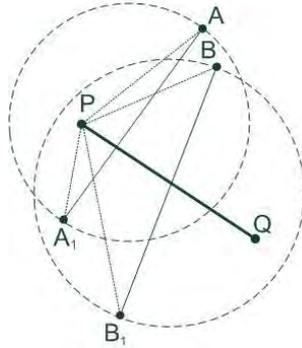


Рис. 2.34. К лемме 2.16.

Лемма 2.17. *Если ребро триангуляции Делоне T_1 пересекает несколько мостов триангуляции Делоне T_2 , то концевая точка ребра может попасть внутрь окружностей влияния не более чем двух мостов.*

Следовательно, можно оценить количество мостов из T_1 , которые может разрушить ребро T_2 .

Лемма 2.18. *При объединении двух триангуляций Делоне каждое ребро пересекается не более чем с четырьмя мостами, которые оно разрушает.*

Имеет место следующая теорема [26]:

Теорема 2.19 (оценка сложности поиска стартеров) *Вычислительная сложность поиска всех стартеров составляет $O(N)$, где N — общее число узлов в объединённой сетке.*

Теорема 2.19 позволила оценить сложность алгоритма поиска интерфейсных граней.

Теорема 2.20 (оценка сложности алгоритма) Алгоритм выделения всех интерфейсных граней имеет сложность $O(N)$, где N — общее число узлов в объединённой сетке.

Доказательство. Покажем, что каждый из трёх этапов алгоритма имеет линейную сложность.

Как было показано выше, сложность этапа поиска начального стартера линейна по N .

Этап прослеживания интерфейсных граней. Во время работы алгоритма происходит разрушение некоторых рёбер из T_1, T_2 и построение интерфейсных рёбер из T . Как упоминалось выше, при реализации структуры данных для триангуляции в задаче удобно использовать структуру «Узлы с соседями» [29].

Построение нового ребра включает в себя проверку углового критерия (см. подраздел 2.2.2) для двух рёбер, которая производится за $O(1)$, и проверку углового критерия для двух инцидентных с ними рёбер из T_1 и T_2 . В случае, когда ребро не удовлетворяет угловому критерию, оно разрушается, и проверка подвергается инцидентное ему ребро. При включении в T первого интерфейсного ребра, образованного стартером, в худшем случае потребуются полный перебор всех рёбер, инцидентных концевым узлам нового ребра ($O(N)$). При построении каждого следующего интерфейсного ребра, потребуется константное время $O(1)$, так как оно соседнее для текущего ребра, т. е. включается в фиксированную позицию. Общее количество разрушенных рёбер не превосходит количество рёбер в T_1 и T_2 ($O(N_1) + O(N_2)$), а общее количество построенных рёбер не превосходит количества рёбер в T ($O(N)$). Следовательно, общее время на разрушение и построение всех рёбер линейно по числу узлов в T .

Сложность этапа поиска стартеров есть $O(N)$ (теорема 2.19).

Теорема доказана. ■

Для вычисления меры (2.10) предлагается алгоритм $\mathbf{A}^{\rho\partial v}$ — модификация алгоритма \mathbf{A}^ρ , состоящая в том, что вместо этапа локализации сеток в триангуляции с использованием МОД производятся два этапа: поиска интерфейсных граней и локализации сеток на основе списка найденных интерфейсных граней.

Теорема 2.21. Локализация сетки в триангуляции на основе списка интерфейсных граней может быть осуществлена за время $O(N)$ в худшем случае.

Доказательство. Согласно утверждению 2.10, задачу прослеживания цепочек интерфейсных граней можно свести за линейное время к задаче локализации лоскутов в триангуляции. Поэтому для каждого лоскута из T_1 можно получить упорядочение узлов каждого из его контуров, например, по часовой стрелке: A_1, A_2, \dots, A_n (см. рис. 2.35).

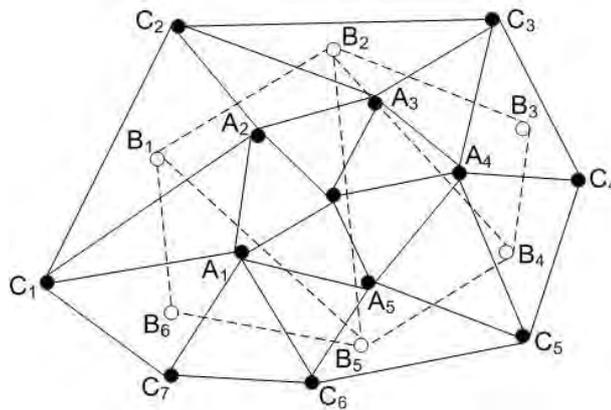


Рис. 2.35. Локализация сетки в триангуляции на основе списка интерфейсных граней.

Тогда стратегия локализации сетки на основе списка интерфейсных граней будет состоять в следующем. Произведём локализацию узла A_1 в треугольнике триангуляции T_2 (за время $O(N_2)$) и составим циклический список треугольников триангуляции T_2 (каждый следующий смежен по стороне с предыдущим), начинающийся с найденного треугольника. Этот список будет состоять из $O(N_2)$ треугольников.

Узел A_1 будет локализован в первом треугольнике $\Delta B_1 B_5 B_6$. При локализации каждого следующего узла контура достаточно просматривать лишь «хвост» списка, начиная с треугольника, в котором локализован предыдущий узел. Благодаря этому такая локализация будет осуществлена за $O(N_2)$. Аналогичным образом происходит локализация контура B_1, B_2, \dots, B_m в списке треугольников из T_1 .

Таким образом, описанная стратегия позволит произвести локализацию сетки в триангуляции за линейное время $O(N)$. Теорема доказана. ■

Следующая теорема позволяет оценить трудоёмкость алгоритма вычисления меры $\rho_{\partial V}$ (2.10).

Теорема 2.22. Алгоритм $\mathbf{A}^{\rho_{\partial V}}$ имеет вычислительную сложность $O(N \log N)$.

Доказательство. Алгоритм $\mathbf{A}^{\rho_{\partial V}}$ включает следующие этапы (для каждого этапа указана оценка сложности):

1. Построение триангуляций Делоне T_1 и T_2 на сетках g_1 и g_2 соответственно $\sim O(N_1 \log N_1) + O(N_2 \log N_2)$;
2. Построение МОД для триангуляций $T_1, T_2 \sim O(N_1) + O(N_2)$ (алгоритм Черитона и Тарьяна);
3. Поиск интерфейсных треугольников $\sim O(N)$ (теорема 2.20);
4. Локализация узлов сеток g_1, g_2 в триангуляциях T_2, T_1 соответственно $\sim O(N)$ (теорема 2.21);
5. Интерполяция значений функций f_1, f_2 в узлах сеток g_2, g_1 соответственно на основе локализации $\sim O(N)$;
6. Вычисление меры $\rho_{\partial V}(f_1, f_2)$ по интерфейсным треугольникам $\sim O(N)$.

Таким образом, все этапы, кроме первого, который имеет сложность $O(N \log N)$, выполняются за линейное по числу узлов в общей сетке время. Поэтому общая сложность алгоритма равна $O(N \log N)$.

Теорема доказана. ■

Следствие 2.22.1. *При построенных триангуляциях Делоне T_1, T_2 на этапе предобработки вычислительная сложность алгоритма \mathbf{A}^{pav} равна $O(N)$.*

2.3.5. Эксперименты

Вычислительные эксперименты проводились на моделях человеческих лиц, снятых трёхмерным сканером Broadway компании Artec Group⁶. Использовалась база из 48 моделей лиц 8 разных людей (для каждого человека по 6 разных моделей). В качестве сравниваемых моделей (поверхностей) использовались разные модели, принадлежащие одному и тому же человеку. Каждая модель содержала около 3 000 точек.

Пусть S_1 и S_2 — сравниваемые поверхности, а S'_2 — редуцированная (упрощённая) поверхность S_2 , полученная равномерным случайным прореживанием сетки второй поверхности. При этом в результате редуцирования удалялись 15% точек модели.

Сетки поверхностей S_1 и S_2 нерегулярные, но распределены равномерно с примерно одинаковой плотностью, поэтому оценку разницы между ними с помощью меры (2.7) при $\mu \equiv 1$ можно считать адекватной. Примем её значение за исходную оценку разницы между поверхностями. Сетки поверхностей S_1 и S'_2 равномерные, но имеют разную плотность.

В таблице 2.7 приведен пример значения мер для сравнения двух поверхностей из базы. Из таблицы видно, что мера (2.10) более адекватно оценивает разницу между S_1 и S'_2 , заданными на сетках разной плотности, чем

⁶ <http://www.artec-group.com>.

мера (2.7): значения обеих мер (2.7) и (2.10) больше исходной оценки, но, если значение меры (2.7) её значительно превышает, то мера (2.10) более приближена к исходной оценке. Аналогичный результат был получен для остальных моделей базы.

Мера сравнения	Значение меры (мм)	Комментарий
$\rho_V(S_1, S_2)$	7094	Исходная оценка для S_1 и S_2
$\rho_V(S_1, S'_2)$	17963	$\rho_V(S_1, S'_2) > \rho_V(S_1, S_2)$
$\rho_{\delta V}(S_1, S'_2)$	10884	$\rho_{\delta V}(S_1, S'_2) \approx \rho_V(S_1, S'_2)$

Таблица 2.7. Значение мер (2.7) и (2.10) для сравнения поверхностей.

2.4. Подгонка поверхностей

Вопрос о сложности одного вычисления меры для сравнения поверхностей важен, потому что в практических приложениях, как правило, требуется не одно вычисление меры, а подгонка из m итераций, во время которой требуется m вычислений меры. Тогда влияние от неэффективного вычисления меры будет возрастать на каждой итерации.

Задача подгонки, или пространственного совмещения, поверхностей состоит в приведении нескольких изображений объекта в единую глобальную систему координат. Такое приведение заключается в трансформации изображений с помощью преобразований сдвигов и поворотов вокруг осей координат. Если рассматривать малые значения параметров преобразования, при которых триангуляции Делоне исходных сеток остаются триангуляциями, то предлагаемый подход позволит вычислить значение меры между совмещёнными поверхностями за оптимальное время $O(mN)$, затратив $O(N \log N)$ на предобработку при построении триангуляций Делоне.

2.4.1. Оптимизационная постановка задачи

Обозначим через M движение в пространстве \mathbb{E}^3 , состоящее в последовательных поворотах на углы $\alpha_M, \beta_M, \gamma_M$ вокруг осей Ox, Oy, Oz соответственно и параллельного переноса на вектор $\Delta_M = (\Delta x_M, \Delta y_M, \Delta z_M)$.

Пусть однозначная поверхность S задана облаком точек с координатами $\{(x_i, y_i, f(x_i, y_i))\}_{i=1}^N$ в пространстве \mathbb{E}^3 . Будем рассматривать движения M , не выводящие S из класса поверхностей, однозначно проецируемых на плоскость, на которой расположено множество точек $\{(x_i, y_i)\}_{i=1}^N$. Обозначим через f^M образ f при движении M .

Пусть $\rho(\cdot, \cdot)$ — мера для сравнения поверхностей. Рассмотрим оптимизационную задачу для функционала $\rho(f_1, f_2^M)$:

$$\rho(f_1, f_2^M) \rightarrow \min_M. \quad (2.18)$$

Минимизация в задаче (2.18) производится в пространстве \mathbb{E}^6 по 6 параметрам преобразования $\alpha_M, \beta_M, \gamma_M, \Delta x_M, \Delta y_M, \Delta z_M$, которыми определяется движение M .

2.4.2. Минимизация функционала различия поверхностей

Для решения задачи (2.18) обосновано использовать метод Нелдера-Мида [83], известный также как симплекс-метод или метод деформируемого многогранника. Метод не накладывает ограничения гладкости на минимизируемый функционал, так как не использует производных и градиентов функции. В результате находится *локальный* минимум функции, поэтому метод зависит от начального приближения поверхностей.

Симплексом в пространстве \mathbb{E}^n называется выпуклая оболочка $(n+1)$ точек, называемых вершинами симплекса, не лежащих в одной гиперплоско-

сти \mathbb{E}^n .

Для решения задачи (2.18) методом Нелдера-Мида выбирается симплекс в пространстве \mathbb{E}^6 параметров преобразования. Идея метода заключается в последовательном деформировании симплекса и его перемещении вокруг экстремума функции: значения функции сравниваются в вершинах начального симплекса, с помощью итерационной процедуры симплекс перемещается в направлении оптимальной точки.

2.5. Основные выводы

1. Предложены меры для сравнения однозначных поверхностей, заданных значениями высот на разных нерегулярных сетках.
2. Предложен метод сравнения поверхностей, включающий
 - построение триангуляций Делоне для обеих сеток и аппроксимацию каждой поверхности кусочно-линейной функцией двух переменных;
 - вычисление значений каждой из этих функций в узловых точках другой сетки на основе локализации узлов сеток в триангуляциях друг друга;
 - вычисление меры различия на основе сравнения значений обеих функций в узлах обеих сеток.
3. В качестве меры близости двух поверхностей выбраны расстояния между кусочно-линейными функциями, аппроксимирующими поверхности по триангуляциям Делоне. Расстояния между функциями вычисляются на отдельных гранях общей триангуляции Делоне, за основу берется метрика L_1 .

Для поверхностей, заданных на сетках, имеющих приблизительно рав-

ные плотности, расстояние вычисляется по всей области определения этих функций.

4. Отдельная мера близости предложена для поверхностей, заданных на сетках с существенно различной плотностью. В этом случае расстояние между функциями предлагается вычислять только в тех частях области определения этих функций, где узловые точки их сеток лежат близко друг другу. Предложен метод для определения таких узловых точек. Метод состоит в вычислении интерфейсных граней объединённой триангуляции Делоне обеих сеток.
5. Задача локализации узлов сеток относится к классу задач геометрического поиска вычислительной геометрии. Известные оптимальные алгоритмы локализации сетки из n_1 узлов в триангуляции, имеющей n_2 узлов, имеют вычислительную сложность $O(n_1 \log n_2)$. Разработка более эффективных алгоритмов, имеющих сложность $O(n_1 + n_2)$ принципиально возможна за счёт использования дополнительной информации о том, что сетка из n_1 узлов представлена триангуляцией Делоне.
6. Предложен алгоритм локализации узлов одной триангуляции Делоне в гранях другой триангуляции Делоне на основе минимальных остовных деревьев. Показано, что в случае равномерности каждой из исходных сеток, вычислительная сложность алгоритма в среднем является линейной по числу узлов в двух сетках. Предположение о равномерном распределении узлов сеток является адекватным для большинства практических приложений.

Построен искусственный пример, показывающий, что в худшем случае теоретическая сложность предложенного метода является квадратичной.

7. Проведены вычислительные эксперименты на реальных данных и модельных задачах большой размерности, подтвердившие корректность и вычислительную эффективность разработанного подхода к сравнению поверхностей.
8. Предложен эффективный алгоритм перечисления интерфейсных граней объединённой триангуляции Делоне двух нерегулярных сеток, т. е. тех треугольников общей триангуляции, у которых вершины принадлежат разным исходным сеткам. Доказана линейная оценка сложности алгоритма в худшем случае.
9. Предложен алгоритм локализации узлов сетки в триангуляции Делоне на основе списка интерфейсных граней. Доказана теоретическая линейная сложность алгоритма в худшем случае.
10. Разработан метод подгонки двух поверхностей, позволяющий найти такое их положение, при котором они наиболее точно совмещаются друг с другом. Подгонка сводится к итерационному сравнению аппроксимирующих функций с использованием преобразования (сдвигов и поворотов) их сеток узлов. Это позволяет эффективно сравнивать поверхности, используя в качестве меры их близости минимальное расстояние между аппроксимирующими функциями, полученное в результате подгонки.

Глава 3

Некоторые задачи анализа поверхностей трёхмерных моделей человеческих лиц

Трёхмерный портрет лица можно получить с помощью системы машинного зрения в виде облака точек. Активное развитие технологий трёхмерного сканирования способствует возрастанию количества исследований, посвящённых задачам анализа и классификации получаемых моделей человеческих лиц. Решение таких задач актуально для многих приложений, связанных с биометрической идентификацией по лицу [39, 62, 68, 76, 80, 84], распознаванием черт лица [94], медицинской диагностикой, планированием и оценкой результатов операций в косметологии [21] и ортодонтии [5, 66].

В данной главе рассматривается ряд задач анализа трёхмерных моделей человеческих лиц, в которых возникают проблемы сравнения и подгонки поверхностей: количественная оценка асимметрии 3d модели лица, построение пространственной модели лица и челюстей для задач ортодонтии, сегментация 3d модели лица на статические и динамические области по трёхмерной видеопоследовательности. В рамках исследований по диссертации перечисленные задачи были рассмотрены в качестве приложений предложенных мер и подхода для сравнения поверхностей. Далее приводятся постановки прикладных задач, предлагаемые методы их решения и результаты вычислительных экспериментов.

Поверхность лица рассматривается как однозначная функция двух переменных $z = F(x, y)$ в некоторой системе координат, в которой ось Oz на-

правлена вдоль линий визирования. Тогда меры для сравнения поверхностей основываются на сравнении двух функций, заданных на разных нерегулярных двумерных сетках.

3.1. Задача оценки асимметрии лица по 3d модели

Известно, что лицо человека обладает лишь *приближенной* зеркальной (двусторонней) симметрией относительно плоскости, делящей его на правую и левую половины. В данном разделе предлагается метод, позволяющий оценивать степень такого приближения, т. е. вычислять количественную оценку *асимметрии*.

Задача оценки асимметрии лица возникает во многих медицинских исследованиях (в офтальмологии для предклинической диагностики и профилактики развития аномалий зрения у детей [16, 82] и оценки асимметрии глазниц; в ортодонтии, косметологии и лицевой хирургии для оценки асимметрии до и после операции), при биометрической идентификации личности [78], распознавании эмоций [79, 93], определении пола [74] по модели лица.

Асимметрия объекта входит в его структурное описание, и оценка асимметрии не может быть получена с помощью каких-либо локальных критериев. Для получения такой оценки необходимо анализировать объект целиком и, кроме того, уметь определять *плоскость симметрии*, которая делит лицо на левую и правую части.

Сами понятия симметрии и асимметрии лица нуждаются в формализации. Далее вводится такая формализация и предлагается новый метод вычисления оценки асимметрии, основанный на сравнении исходной и отражённой моделей лица.

Пусть $\rho(S_1, S_2)$ — мера, позволяющая оценивать различие между двумя моделями поверхностей S_1, S_2 , заданных облаками точек. Будем считать

модель лица *симметричной* относительно некоторой пересекающей модель плоскости, если мера различия ρ между исходной поверхностью лица и поверхностью, зеркально отражённой относительно данной плоскости, равна 0. С точки зрения введённого понятия симметрии, модели человеческих лиц не являются симметричными.

Пусть S — исходная модель лица, заданная облаком точек. Обозначим через S_P модель, полученную отражением S относительно плоскости P .

Плоскостью симметрии P^* модели лица S будем называть такую плоскость P , при которой мера различия между исходной и зеркально отражённой относительно этой плоскости моделями минимальна (то есть две модели совпадают наилучшим образом):

$$P^* = \arg \min_P \rho(S, S_P). \quad (3.1)$$

При оценке *асимметрии* лица предлагается сравнивать исходную модель и модель, отражённую относительно плоскости симметрии P^* . В качестве количественной оценки асимметрии лица предлагается использовать значение меры различия $\rho(S, S_{P^*})$.

Далее показывается, какое влияние оказывает определение плоскости симметрии на оценку, и предлагается алгоритм поиска плоскости симметрии. При этом решается экстремальная задача (3.1): наиболее точно найденной плоскости соответствует наименьшая оценка.

В перечисленных выше работах [74, 78, 79, 93] используются модели, приведённые к *регулярным* сеткам (см. рис. 3.1): поэтому в качестве оценки асимметрии в них предлагается напрямую вычислять сумму или среднее значение по всем разностям высот между исходной и отражённой моделями (левой и правой половинами модели). Обычно модели лица, полученные с помощью трёхмерных сканеров, заданы на таких сетках, что точки, в которых известны значения высот поверхности, расположены несимметрично относительно

плоскости симметрии модели. Такие сетки могут быть как регулярными, так и *нерегулярными* (см. рис. 3.2). Кроме того, методы, позволяющие сравнивать только модели, заданные на регулярных сетках, не позволяют уточнять плоскость симметрии (в рамках самой модели), так как «шевеление» плоскости сразу же приводят к нерегулярностям. Поэтому в данной задаче требуется введение более сложной оценки для вычисления количественного значения асимметрии.

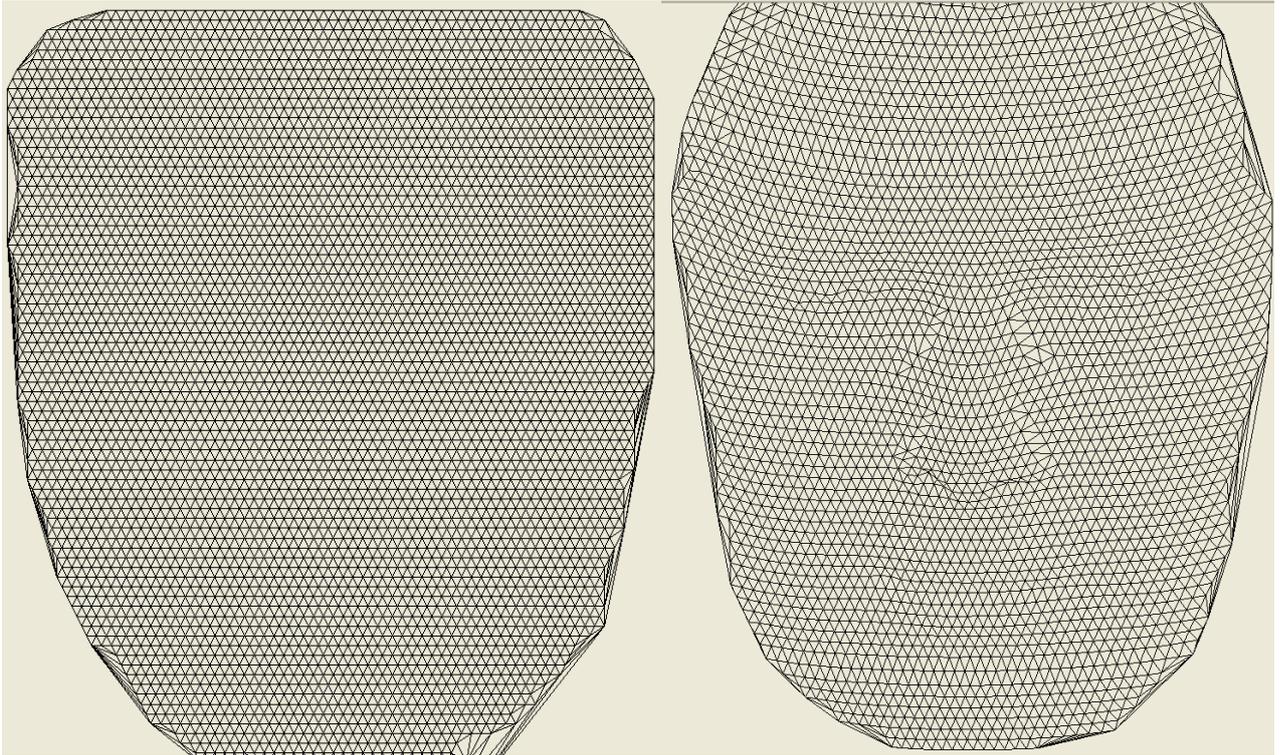


Рис. 3.1. Регулярная треугольная сетка. **Рис. 3.2.** Нерегулярная треугольная сетка.

Модель поверхности лица S , полученную методом трёхмерного сканирования, можно рассматривать как однозначную поверхность (см. рис. 3.3), которой соответствует функция f , заданная набором высот на дискретном множестве точек на плоскости.

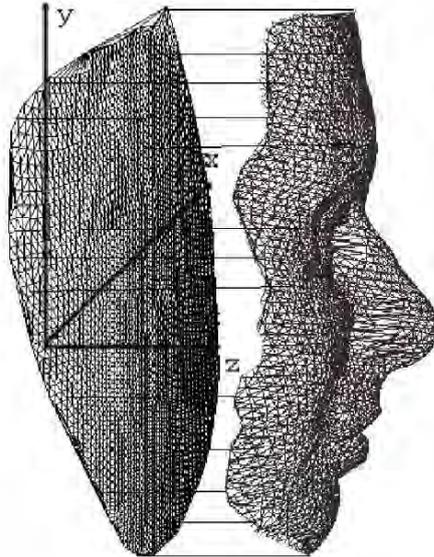


Рис. 3.3. Проецирование поверхности модели лица на плоскость.

3.1.1. Описание базы моделей лиц

Трёхмерные модели для данного исследования были получены с помощью 3d камеры, разработанной в компании Artec Group¹. Была собрана база, содержащая 191 модель лиц 8 разных людей. При этом съёмка в фас происходила при «нейтральном» выражении лиц снимающихся.

Каждая модель базы содержит набор точек, заданных своими координатами (x, y, z) в пространстве E^3 . Координаты точек предоставлены в масштабе 1 : 1, т.е. соответствуют реальным размерам лица человека, с точностью до 10^{-6} мм. Количество точек в моделях варьируется от 1 000 до 3 000, а в среднем по базе составляет 1 500–2 000.

В системе координат $Oxyz$ модель базы расположена таким образом, что кончик носа соответствует началу координат, ось Oz направлена вдоль оси визирования в направлении позиции съёмки, а ось Oy проходит вдоль лица вверх(см. рис. 3.4). Следует отметить, что такое расположение модели в системе координат является приближённым и будет уточнено в процессе получения количественной оценки асимметрии с помощью преобразований

¹ <http://www.artec-group.com>.

системы координат — малых сдвигов и поворотов на малые углы. Будем называть такое расположение модели в системе координат *нормализованным*, а процесс приведения модели лица в такую систему координат — *нормализацией*. При описанном расположении плоскость Oyz можно принять за первое приближение плоскости симметрии модели.

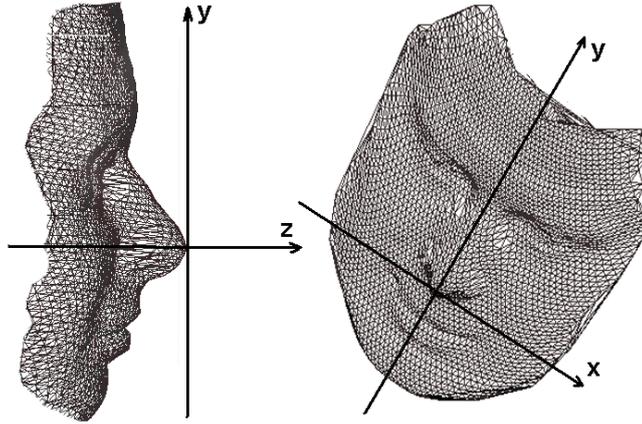


Рис. 3.4. Нормализация трёхмерной модели лица в системе координат.

3.1.2. Постановка задачи

Содержание математической задачи, решение которой будет использовано для оценки асимметрии модели, заключается в следующем. Дана модель лица S , заданная облаком из N точек с координатами в системе координат $Oxyz$. Пусть модели S соответствует функция f_1 , а модели S_{Oyz} , полученной отражением модели S относительно плоскости Oyz , — функция f_2 . Сравнение двух моделей сводится к сравнению соответствующих им функций.

Пусть функции f_1 и f_2 заданы наборами значений $\{f_1^i\}_{i=1}^N$, $\{f_2^i\}_{i=1}^N$ в узлах некоторых двумерных сеток g_1 и g_2 соответственно:

$$g_1 = \{(x^i, y^i)\}_{i=1}^N, \quad g_2 = \{(-x^i, y^i)\}_{i=1}^N,$$

$$f_1^i = f_1(x^i, y^i), \quad f_2^i = f_2(-x^i, y^i).$$

Пусть ρ — мера для сравнения поверхностей (например, ρ_M (2.4), ρ_{V_μ} (2.7) или ρ_V (2.9) из подраздела 2.1.3 главы 2).

Требуется вычислить меру $\rho(f_1, f_2)$.

3.1.3. Метод оценки асимметрии модели

Предлагаемый подход к получению оценки асимметрии лица состоит из следующих этапов:

- вычисление начальной количественной оценки асимметрии;
- поиск плоскости симметрии модели;
- вычисление количественной оценки.

Общая схема подхода изображена на рис. 3.5. Рассмотрим основные этапы более подробно.

3.1.3.1. Получение начальной количественной оценки

На первом этапе метода оценки асимметрии используется алгоритм сравнения однозначных поверхностей, описанный в разделе 2.2 главы 2. При этом сравниваются поверхности, заданные функциями f_1 (соответствующей исходной маске) и f_2 (соответствующей маске, отражённой относительно плоскости Oyz). На рис. 3.6 участки лица, на которых отражённая маска выше исходной, выделены более тёмным цветом. Полученное значение функции расстояния $\rho(f_1, f_2)$ будем считать *начальной количественной оценкой* асимметрии модели.

3.1.3.2. Уточнение плоскости симметрии

На следующем этапе алгоритма варьируется гипотеза о положении плоскости симметрии. Система координат связана с этой гипотезой, так как за плоскость симметрии принята плоскость Oyz . Поэтому такие вариации вы-

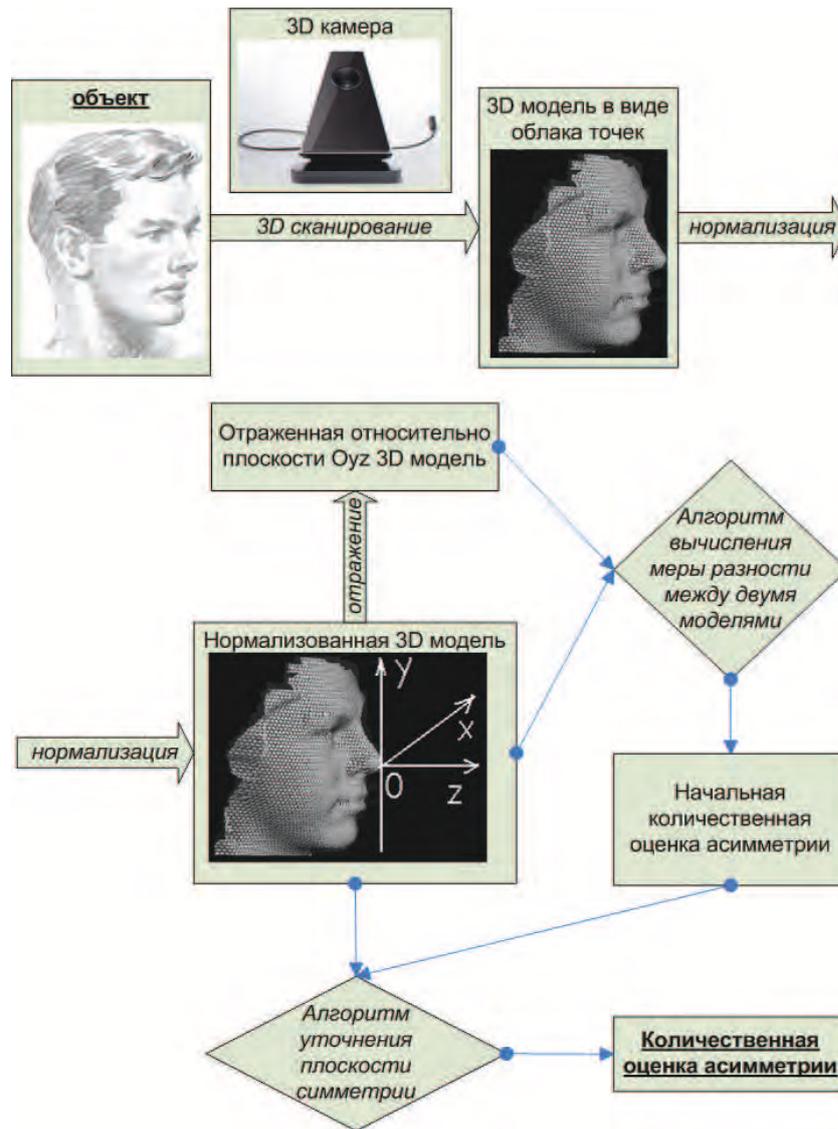


Рис. 3.5. Общая схема предлагаемого подхода.

ражаются в рамках принятой модели в преобразовании системы координат. Так как нормализация моделей в системе координат не является абсолютно точной, а лишь приближённой, мы пробуем преобразовать координаты с помощью элементарных преобразований: небольших сдвигов по осям координат и поворотов вокруг осей на малые углы. Уточнение происходит с целью нахождения такого положения плоскости симметрии Oyz , при котором значение количественной оценки асимметрии минимально (то есть, при котором две маски совпадают наилучшим образом, см. рис. 3.7). При этом минимизация будет происходить не по всем шести параметрам преобразований (как

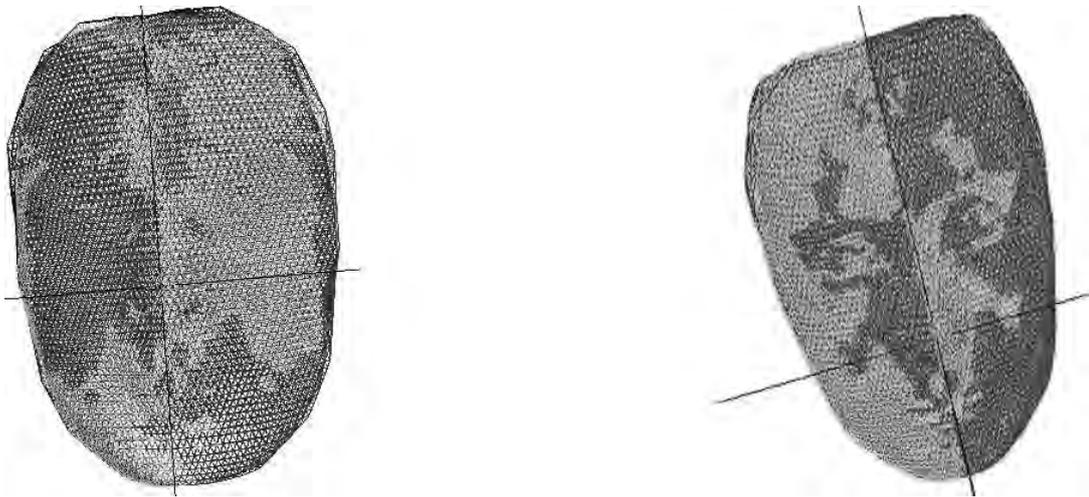


Рис. 3.6. Примеры разных исходных масок и их отражений относительно плоскостей симметрии.

было рассмотрено в разделе 2.4 главы 2), а лишь по трём: сдвиги по осям Oy и Oz не будут давать вклад в оценку асимметрии, поворот вокруг оси Ox не рассматривается в силу того, что лица снимаются в фас. Минимизация будет происходить по трём оставшимся параметрам: сдвигу по оси Ox и поворотам вокруг осей Oy и Oz .

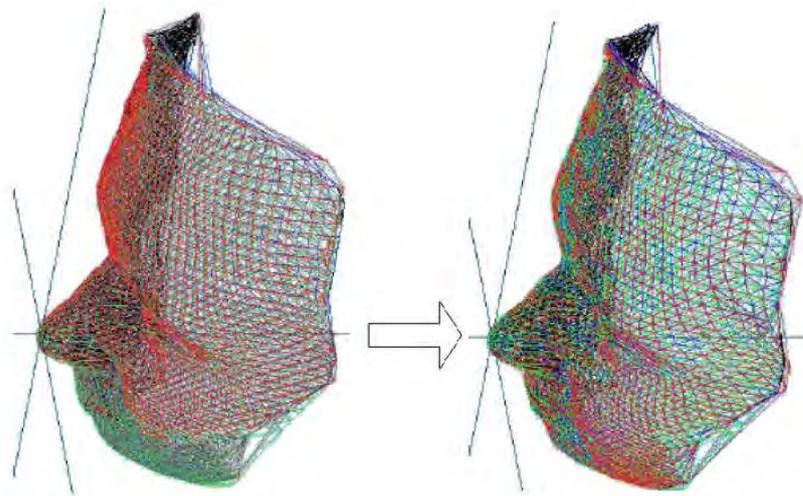


Рис. 3.7. Поиск наилучшего совпадения между поверхностями.

Обозначим через $S^{(x,\varphi,\psi)}$ модель лица, полученную из исходной модели S с помощью следующих преобразований: сдвига на x по оси Ox , поворот на угол φ вокруг оси Oy и на угол ψ вокруг оси Oz . Будем считать,

что параметры преобразования малы и не выводят модель $S^{(x,\varphi,\psi)}$ из класса однозначных поверхностей в системе координат $Oxyz$. Пусть $f_1^{(x,\varphi,\psi)}$ — функция, соответствующая модели $S^{(x,\varphi,\psi)}$, а $f_2^{(x,\varphi,\psi)}$ — функция, соответствующая отражённой модели $S_{Oyz}^{(x,\varphi,\psi)}$.

Для поиска плоскости симметрии решается следующая оптимизационная задача:

$$\rho(f_1^{(x,\varphi,\psi)}, f_2^{(x,\varphi,\psi)}) \rightarrow \min_{x,\varphi,\psi}. \quad (3.2)$$

Количественной оценкой асимметрии исходной модели является решение задачи (3.2) — значение ρ^* меры между исходной моделью и её отражением относительно плоскости симметрии:

$$\rho^* = \arg \min_{x,\varphi,\psi} \rho(f_1^{(x,\varphi,\psi)}, f_2^{(x,\varphi,\psi)}). \quad (3.3)$$

Для решения задачи (3.2) может быть использован метод Нелдера-Мида (см. подраздел 2.4.2 главы 2).

В качестве первоначального приближения берётся точка, соответствующая совпадению плоскости симметрии с плоскостью Oyz в исходной системе координат, т. е. точка $M_0(x_0, \varphi_0, \psi_0) = (0, 0, 0)$.

При минимизации учитывается, что функция является овражной: изменение Δ переменных φ и ψ приводит к большему изменению значения функции, чем такое же изменение Δ переменной x .

3.1.4. Локальная асимметрия модели

Предложенный метод позволяет анализировать асимметрию модели локально (по зонам). Таким образом можно определить, какие участки лица наиболее асимметричны с точки зрения предложенной оценки. Для этого для каждого треугольника общей триангуляции Делоне T вычисляется значение отношения объёма разности между двумя функциями на этом треугольнике к его площади.

На рис. 3.8 изображены примеры нескольких раскрашенных моделей лиц: каждая грань общей триангуляции окрашена в соответствии с нормированным значением указанного отношения — наиболее светлые участки являются наиболее асимметричными. Зоны наибольшей и наименьшей симметричности сходны для разных моделей лиц, принадлежащих одному и тому же человеку, и отличаются для моделей лиц разных людей.

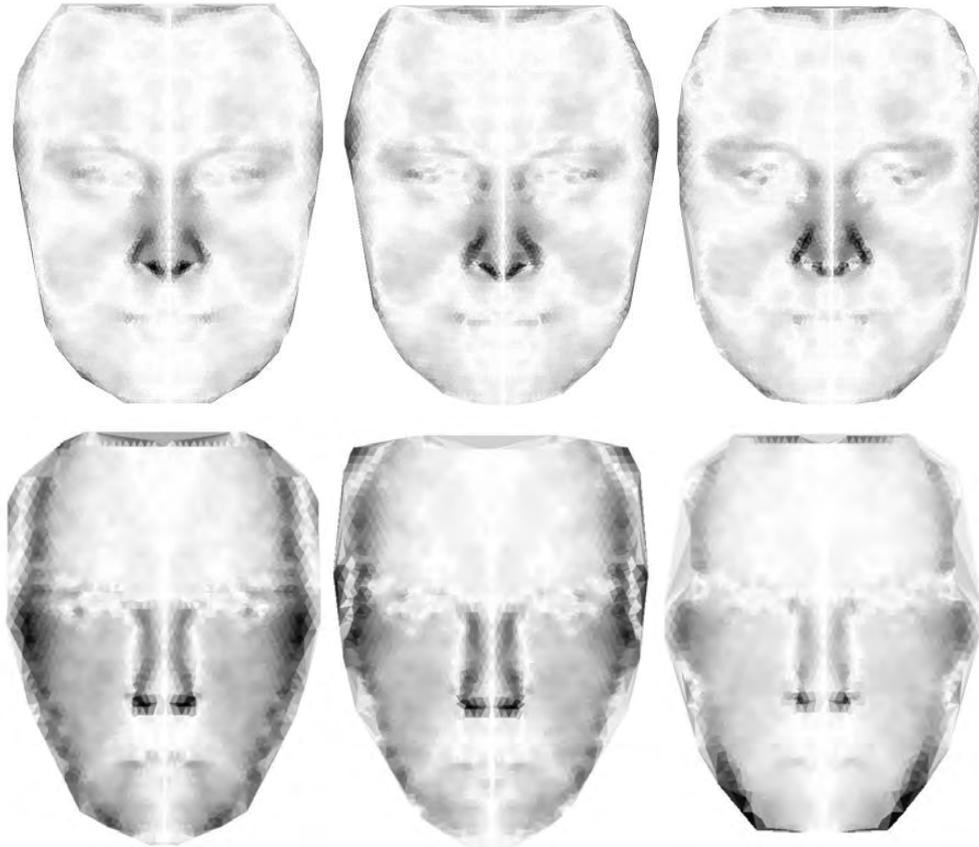


Рис. 3.8. Сверху: три разные модели лица одного человека, раскрашенные в соответствии с оценкой асимметрии, снизу: три разные модели лица другого человека.

3.1.5. Эксперименты

В вычислительных экспериментах для данного исследования использовалась следующая мера для сравнения поверхностей:

$$\rho(f_1, f_2) = \sum_{\Delta ABC \in T} V(A, B, C, f_1, f_2) = \sum_{\Delta ABC \in T_{\Delta ABC}} \iint |f_1 - f_2| dx dy. \quad (3.4)$$

Номер модели	<i>I</i>	<i>II</i>	<i>III</i>	<i>IV</i>
Начальная оценка асимметрии	47 466,361	49 192,110	43 473,767	46 280,040
Уточнённая оценка асимметрии	24 072,518	25 205,272	24 421,316	22 263,813

Таблица 3.1. Объём разности между моделью и её отражением (все значения приведены в куб. мм.) для четырёх разных моделей одного и того же человека.

Физический смысл меры (3.4) — объём разности между исходной и отражённой масками, единица измерения — куб. мм.

В таблице 3.1 приведены значения начальной количественной оценки асимметрии и оценки асимметрии после уточнения плоскости симметрии для 4 моделей лица одного и того же человека:

Чтобы пояснить, насколько значимыми являются величины, указанные в таблице, отметим, что величина уточнённой оценки асимметрии, измеренная в столовых ложках жидкости, составляет примерно полторы столовые ложки².

При этом оптимальный сдвиг по оси Ox для этих моделей составлял от 2,4 до 2,6 мм, а оптимальные углы поворота вокруг осей координат Oy и Oz — примерно 0,015 рад.

3.1.5.1. Устойчивость измеренной оценки на базе лиц

В рамках исследований были проведены вычислительные эксперименты на описанной базе моделей лиц для проверки устойчивости предложенной оценки.

Эксперименты показали, что начальная количественная оценка асиммет-

² Объём столовой ложки составляет 15 000 куб. мм.

Номер модели	Начальная оценка асимметрии	Уточнённая оценка асимметрии
I	92 628,068	58 228,725
II	97 199,64	62 139,821
III	89 392,86	59 263,088
IV	84 066,166	58 931,266
V	71 079,908	53 289,94
VI	101 749,542	61 009,302

Таблица 3.2. Значение начальной и уточнённой оценки асимметрии для 6 моделей лиц одного и того же человека (все значения приведены в куб. мм.).

рии для разных моделей лиц одного и того же человека сильно изменяется и не обладает устойчивостью. Однако, этап уточнения плоскости симметрии модели увеличивает устойчивость предложенной оценки.

В таблице 3.2 приведены результаты для нескольких моделей лиц одного и того же человека из базы.

По результатам вычисления начальной и уточнённой оценок для 30 моделей лиц, относящихся к одному снимаемому объекту (человеку), можно установить, что в среднем разброс значений начальной оценки асимметрии составляет 36 000–40 000 куб. мм., а разброс значений окончательной (уточнённой) оценки значительно меньше и составляет в среднем 9 000–11 000 куб. мм.

Таким образом, вычислительные эксперименты показали, что алгоритм уточнения плоскости симметрии достоверен, работает корректно и значительно повышает устойчивость предложенной оценки асимметрии лица.

3.1.5.2. Оценка точности аппроксимации модели

Для оценки точности аппроксимации модели лица была проведена серия экспериментов по упрощению поверхности с помощью прореживания исходного облака точек (выкидывания из модели случайно выбранных точек).

В рамках эксперимента исследовалась зависимость между оценкой асимметрии лица и количеством точек в исходной 3d модели лица (см. рис. 3.9). Как видно из графика, для моделей из 3 000 точек оценка асимметрии начинает сильно возрастать при удалении из исходного облака более 2/3 точек.

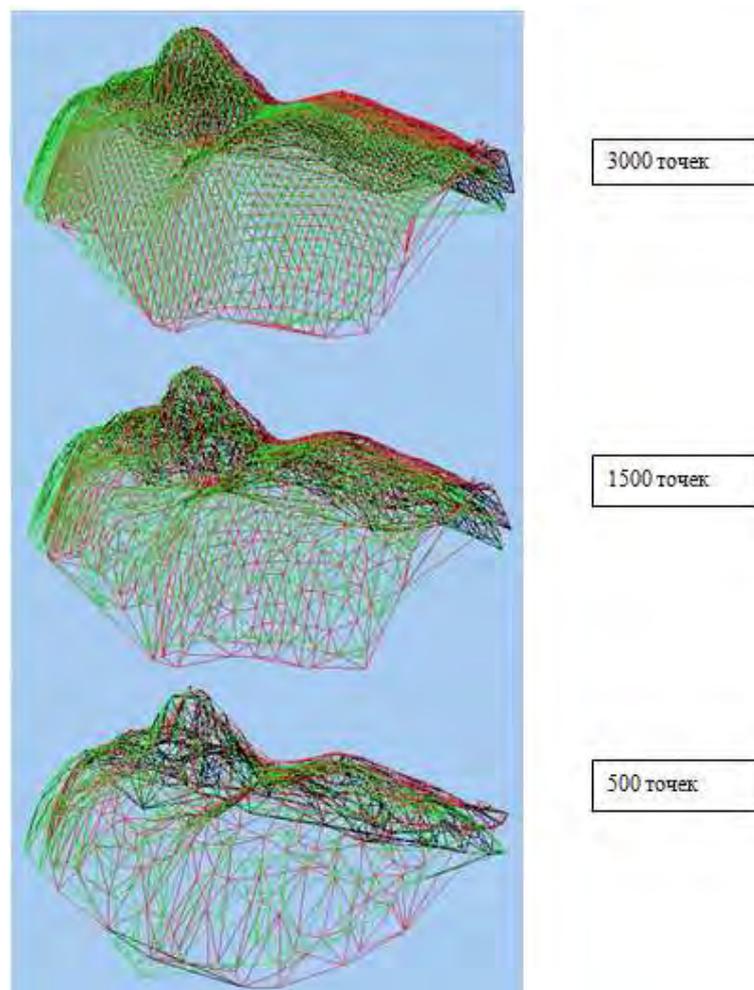


Рис. 3.9. Исходная и обратная маски при прореживанием облака точек.

Результаты показаны на графике 3.10.

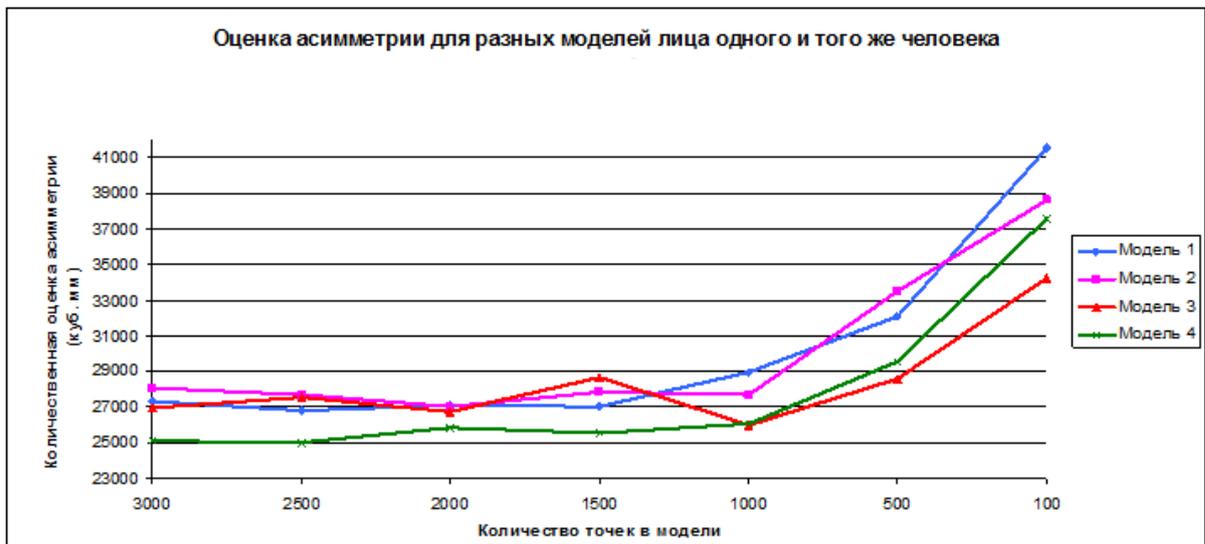


Рис. 3.10. График зависимости между оценкой асимметрии лица и количеством точек в модели.

3.2. Построение совместной пространственной модели лица и зубов для задач ортодонтии

В стоматологии существует методика сопоставления моделей лица и зубов человека с использованием *реперного объекта*, т.е. объекта, позволяющего определить взаимное пространственное расположение других объектов, связанных с ним [22]. Методика заключается в следующем. С помощью трёхмерного сканера производится съёмка 3d модели лица (модель 1) и 3d модели лица вместе с реперным объектом (модель 2); с помощью стоматологического сканера, обладающего высокой точностью, производится съёмка 3d модели гипсовой челюсти (модель 3) и 3d модели гипсовой челюсти вместе с реперным объектом (модель 4) (см. рис. 3.11).

Тогда задача позиционирования модели челюсти в модели лица [5] заключается в сопоставлении моделей 1 и 3. Задача сводится к подгонке модели 1 к модели 2, модели 3 к модели 4 и дальнейшему совмещению моделей 2 и 4 по реперному объекту. Сложность такого совмещения заключается в том, что реперный объект в моделях 2 и 4 имеет разные уровни детализации, так как

получен сканерами разной точности. В этом случае может быть использована мера $\rho_{\partial V}$ (2.10).

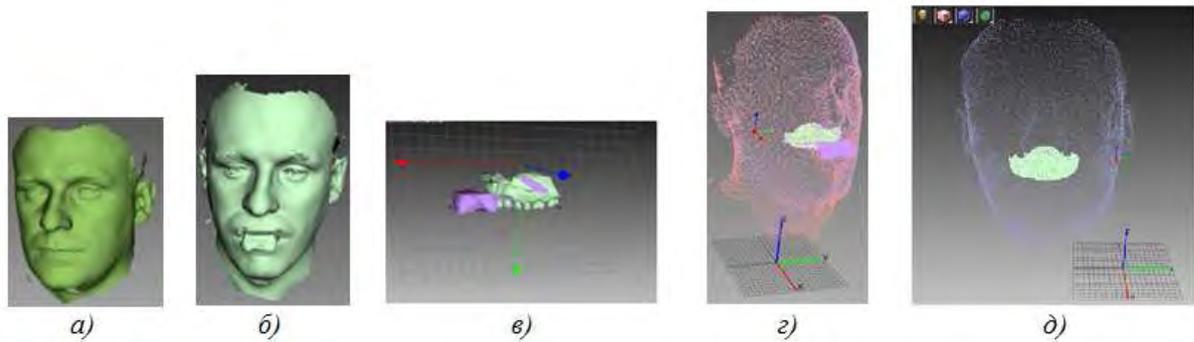


Рис. 3.11. Совместная пространственная модель лица и зубов: *a)* 3d модель лица; *б)* 3d модель лица с реперным объектом; *в)* 3d модель челюсти с реперным объектом; *г)* совмещение моделей по реперному объекту; *д)* позиционирование модели челюсти в модели головы.

3.3. Задача оценки динамики движения нижней челюсти по трёхмерной видеопоследовательности

Современные трёхмерные сканеры позволяют не только получить модель лица необходимой точности, но и произвести съёмку изменений лица в режиме реального времени, захватывая любые мимические движения, в том числе движения нижней челюсти, например, во время жевания или разговора.

В последние годы технологии трёхмерного моделирования получили широкое распространение в стоматологии [5, 66] и косметологии [21]: работа с 3d моделями позволяет производить предклинические исследования, планировать операции и оценивать возможные результаты лечения. В [70] предлагается метод лазерного сканирования и система визуализации для виртуального планирования операций. В [72] предлагается использование фотограмметрической системы для получения трёхмерных моделей челюстей и лица и дальнейшего определения взаимного расположения нижней и верхней челю-

стей. В [53] предлагается метод моделирования челюсти пациента по снимкам, полученным компьютерным томографическим сканированием.

Задачи анализа динамики движения челюсти и построения модели такого движения актуальны и востребованы для исследований в областях хирургической стоматологии и челюстно-лицевой хирургии [71]. В реконструктивной хирургии при планировании операций используются снимки лица в движении для более точного последующего анализа черепа пациента [32].

В данном разделе рассматривается задача регистрации изменений модели лица и нижней челюсти по трёхмерной видеопоследовательности, полученной при съёмке жующего человека. Для этого в модели выделяются статические и динамические относительно исследуемого видеоряда области.

Анализируя данные, полученные в результате обработки последовательности кадров, необходимо произвести сегментацию модели и описать движение динамической области относительно статической в процессе съёмки. Далее предлагается подход для решения этой задачи, основанный на отдельной подгонке верхних и нижних частей модели лица.

3.3.1. Постановка задачи

Рассмотрим задачу сегментации модели лица человека по трёхмерному видеоряду процесса жевания — серии последовательных трёхмерных изображений. Под *сегментацией* будем понимать разбиение поверхности лица на статические и динамические (относительно исследуемого видеоряда) области. При этом будет производиться сегментация статичной трёхмерной модели лица, полученной при «нейтральном» выражении лица снимающегося.

Исходными данными в задаче являются модели, заданные в виде облаков точек:

- статичная 3d модель лица S ;

— видеопоследовательность из трёхмерных изображений лица D_1, \dots, D_n .

Требуется разделить S на части, статические и динамические относительно заданной видеопоследовательности.

3.3.2. Метод сегментации модели на статические и динамические области

Пусть каждая из исходных моделей $I \in \{S, D_1, \dots, D_n\}$ задана в трёхмерном пространстве набором координат точек: $\{x_k, y_k, f_I(x_k, y_k)\}_{k=1}^{N_I}$, где N_I — количество точек в модели I . Обозначим набор точек плоскости, на котором задаётся поверхность модели I , через g_I : $g_I = \{x_k, y_k\}_{k=1}^{N_I}$.

3.3.2.1. Стандартная система координат

На первом шаге работы происходит нормализация моделей: с каждой из моделей S, D_1, \dots, D_n связывается своя система координат.

Съёмка лиц для данного исследования производилась в фас, с помощью сканера были получены модели в виде облаков точек в системе координат, в которой ось Oz проходит вдоль оси визирования и направлена в камеру, ось Oy проходит вдоль лица по направлению от подбородка ко лбу, ось Ox — поперёк лица, от правой щеки к левой. В дальнейшем будем подразумевать, что во всех обрабатываемых моделях лиц оси координат направлены таким же образом, и под *верхней* и *нижней* частями модели понимаются части с большими и, соответственно, меньшими значениями координаты по оси Oy .

В процессе нормализации первоначальная система координат каждой из моделей $I \in \{S, D_1, \dots, D_n\}$ преобразовывается с помощью параллельного переноса в новую таким образом, чтобы для модели были выполнены следу-

ющие условия:

$$\begin{cases} \sum_{(x,y) \in g_I} x = \sum_{(x,y) \in g_I} y = 0; \\ \max_{(x,y) \in g_I} f_I(x, y) = 0. \end{cases} \quad (3.5)$$

В (3.5) суммирование производится по всем точкам (x, y) из g_I . Систему координат модели I , для которой выполнены условия (3.5), будем называть *стандартной* системой координат модели I .

3.3.2.2. Сегментация модели

При движении нижней челюсти на снимках видеоряда наибольшей статической частью является верхняя часть лица, к которой относится лоб и нос; нижняя часть лица (подбородок, губы) относятся к динамическим частям.

Для верхней и нижней частей лица строятся собственные местные системы декартовых координат. Будем называть их соответственно *верхней и нижней системами координат*. Для модели S верхняя и нижняя, а также стандартная системы координат совпадают, а для моделей D из видеоряда, сопоставленных по верхней части с моделью S , они различаются. Задача состоит в регистрации этих различий. Преобразование нижней системы координат в верхнюю для каждого кадра будет описывать динамику движения нижней челюсти.

Для сравнения моделей лица с двух снимков предлагается подход, заключающийся в 1) определении статической части для обоих снимков методом подгонки и 2) описании положения динамической части относительно статической.

Рассмотрим статичную модель S и модель видеоряда $D \in \{D_1, \dots, D_n\}$. Пусть S_B, D_B — статические (верхние) части моделей S и D соответственно, а S_H, D_H — динамические (нижние) части этих моделей (см. рис. 3.12). Средняя часть моделей, расположенная между верхней и нижней частями,

не рассматривается.

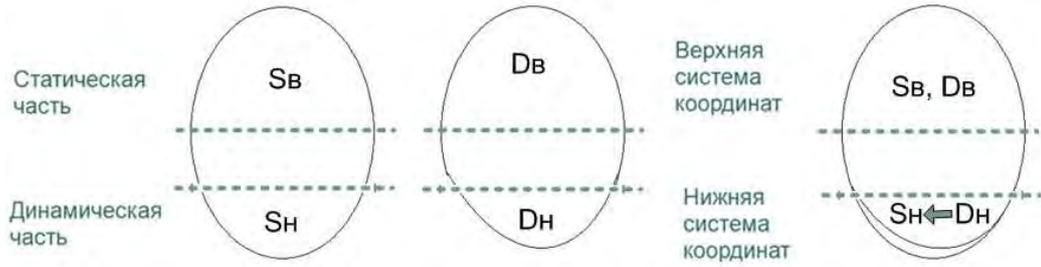


Рис. 3.12. Подгонка статических частей двух моделей. S_B, S_H — верхняя и нижняя части статичной модели S ; D_B, D_H — верхняя и нижняя части модели D из видеоряда.

Модель D приводится к стандартной системе координат модели S и статические части S_B, D_B сопоставляются друг с другом методом подгонки так, чтобы мера различия ρ между ними была наименьшей. При этом динамические части моделей S_H и D_H будут отличаться. Задача описания динамики движения подвижной части модели относительно статической состоит в регистрации таких различий для каждого снимка D из видеоряда.

Для выделения статических частей модели разделяются горизонтальной плоскостью P , первоначально совпадающей с плоскостью Oxz , верхние части моделей сравниваются, далее плоскость P смещается вдоль оси Oy с некоторым шагом, и отделяемые ей верхние части моделей снова сравниваются. При этом могут быть применены меры различия ρ , учитывающие площадь, на которой задана верхняя часть модели, отсекаемая плоскостью P . Смещение плоскости P происходит до тех пор, пока мера различия поверхностей уменьшается.

Для выделения динамических частей используется аналогичный метод. Первоначальное положение плоскости P устанавливается на некотором уровне (отсекающем заранее определённую нижнюю часть модели — с наименьшими координатами по оси Oy). Затем плоскость P смещается с некоторым шагом до тех пор, пока мера различия ρ уменьшается. Эти смещения производятся в обоих направлениях по оси Oy , а в динамическая часть определяется

как часть с минимальным значением меры различия ρ из полученных двух.

При этом в качестве метода сравнения поверхностей используется метод, описанный в разделе 2.2 главы 2.

Используемые меры различия между поверхностями двух моделей и метод подгонки описан в следующем подразделе. Подгонка поверхностей состоит в нахождении такого движения, при котором мера различия между поверхностями, описывающими лица, минимальна (см. рис. 3.13).

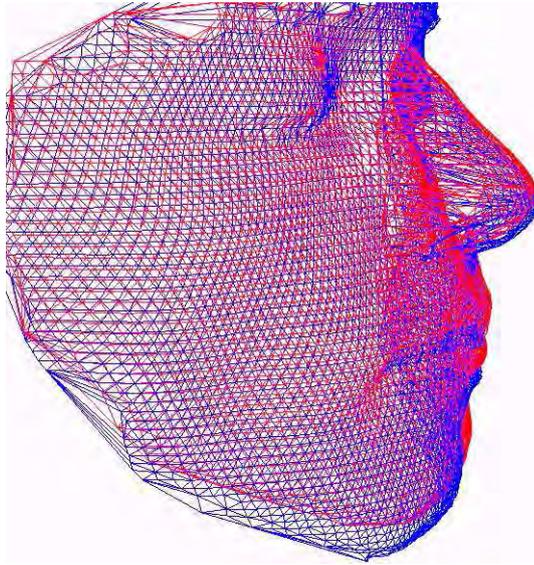


Рис. 3.13. Пример подгонки двух моделей из видеоряда. Слабое отклонение в статических частях — нос и лоб, сильное отклонение в динамической части — челюсти.

3.3.3. Подгонка статических частей моделей

Пусть M — движение в пространстве \mathbb{E}^3 . Далее будем рассматривать его как композицию последовательных преобразований: поворотов на углы $\alpha_M, \beta_M, \gamma_M$ вокруг осей Ox, Oy, Oz соответственно и параллельного переноса на вектор $(\Delta x_M, \Delta y_M, \Delta z_M)$.

Обозначим через f^M образ f при движении. Пусть статические части S_B и D_B исходных моделей представлены облаками точек с координатами $\{x_i, y_i, f_{S_B}(x_i, y_i)\}_{i=1}^{N_1}$ и $\{x_j, y_j, f_{D_B}(x_j, y_j)\}_{j=1}^{N_2}$.

Подгонка двух статических частей S_B и D_B моделей заключается в минимизации меры различия ρ между ними. Для этого в пространстве преобразований \mathbb{E}^6 решается оптимизационная задача, аналогичная (2.18):

$$\rho(f_{S_B}, f_{D_B}^M) \rightarrow \min_M, \quad (3.6)$$

где поиск минимума происходит по 6 параметрам преобразования, которыми определяется движение M : $\alpha_M, \beta_M, \gamma_M, \Delta x_M, \Delta y_M, \Delta z_M$.

В качестве меры ρ могут быть использованы меры ρ_M (2.4), ρ_{V_μ} (2.7) или ρ_V (2.9), введённые в подразделе 2.1.3 главы 2.

Выбор начального приближения для метода минимизации является критичным в рассматриваемой задаче. Описанное выше приведение моделей к стандартной системе координат является эффективным решением этой проблемы.

Для решения задачи (3.6) можно использовать метод минимизации Нелдера-Мида (см. подраздел 2.4.2 главы 2).

3.3.4. Эксперименты

Съёмка для настоящего исследования производилась трёхмерным сканером BroadwayTM компании Artec Group, позволяющим делать до 15 снимков в секунду.

В рамках исследования были проведены вычислительные эксперименты на видеопоследовательности из 83 моделей, полученном при съёмке жующего человека. Количество точек в моделях варьировалось от 4 000 до 5 000. Статичная модель лица содержала около 100 000 точек.

Проведённые эксперименты показали, что самая быстрая оптимизация меры различия достигается в случае использования среднего осевого расстояния ρ_M (2.4) и метода минимизации Нелдера-Мида.

3.3.4.1. Выделение статических частей

В процессе жевания большая часть верхней части человеческого лица остаётся неподвижной, и алгоритм подгонки по этим частям даёт хорошие результаты.

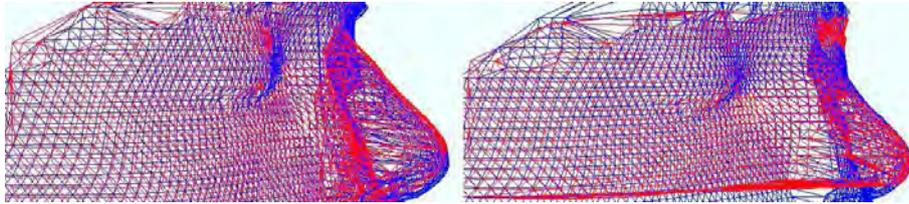


Рис. 3.14. Сравнение качества выравнивания верхних частей лица по всей модели (слева) и только по сегментированной статической части (справа).

Несмотря на неплохое качество выравнивания верхних частей лица при подгонке по всей модели, на рис. 3.14 видно заметное улучшение при подгонке отдельно по выделенной статической области.

3.3.4.2. Выделение динамических частей

Для определения начального положения разделяющей плоскости P был проведён вычислительный эксперимент на моделях видеоряда D_1, \dots, D_n . Динамическая область задаётся точками с координатой по оси Oy меньше определённого порогового значения. На рис. 3.15 представлен график минимальной меры различия между динамическими областями в зависимости от доли отсекаемой плоскостью P части лица (от 1% до 100% с шагом в 1%).

Как показали эксперименты, в промежутке от 2% до 8% локальные минимумы функционала изменяются мало, что позволяет сделать предположение о том, что в качестве приближения динамической части можно использовать нижние 8% точек исходной модели (см. рис. 3.16).

На рис. 3.17 показан результат выделения верхней и нижней частей модели для трёх снимков из видеоряда.

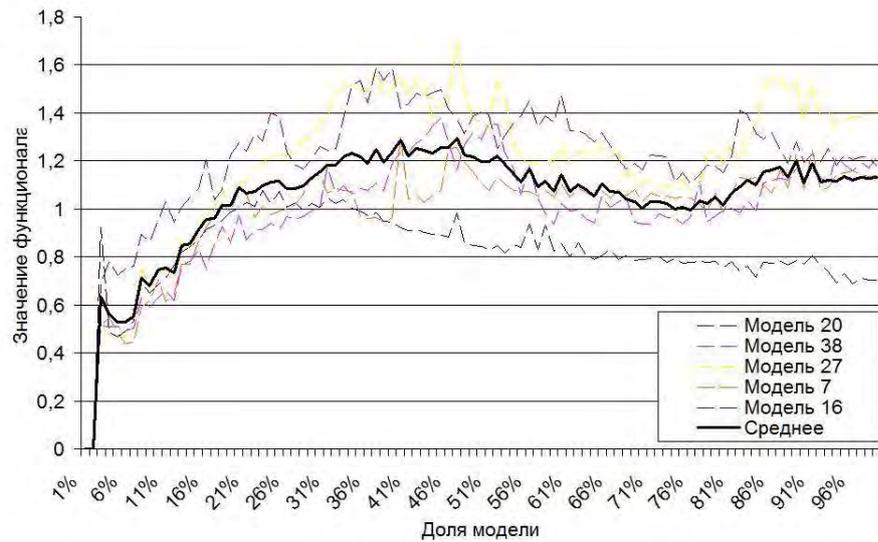


Рис. 3.15. Зависимость значения минимальной меры различия между динамическими областями от доли количества точек в динамической области от общего количества точек в модели лица.

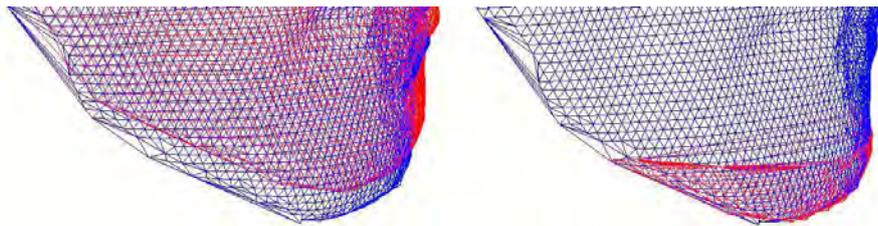


Рис. 3.16. Сравнение качества выравнивания нижней челюсти по всей модели (слева) и только по сегментированной динамической части (справа).

Преобразование системы координат, связанной с динамической областью, в систему координат, стандартную для исходной статичной модели S , описывает динамику движения нижней челюсти. Формальное описание движения было получено в виде серии матриц преобразования и может быть представлено как анимация движения базисных векторов нижней системы координат при фиксированном положении базисных векторов верхней системы исходной статичной модели (см. рис. 3.18). По полученным данным могут быть определены максимальные сдвиги нижней челюсти, характер движения и другие параметры для исследуемого видеоряда.

Проведённые эксперименты обосновывают реализуемость и корректность

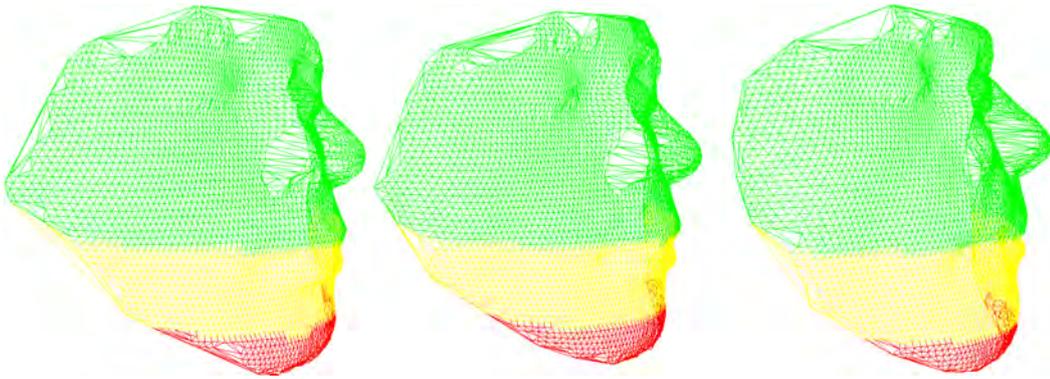


Рис. 3.17. Выделенные предложенным методом верхняя (зелёный цвет), средняя (жёлтый цвет) и нижняя (красный цвет) части моделей для трёх снимков из видеоряда.

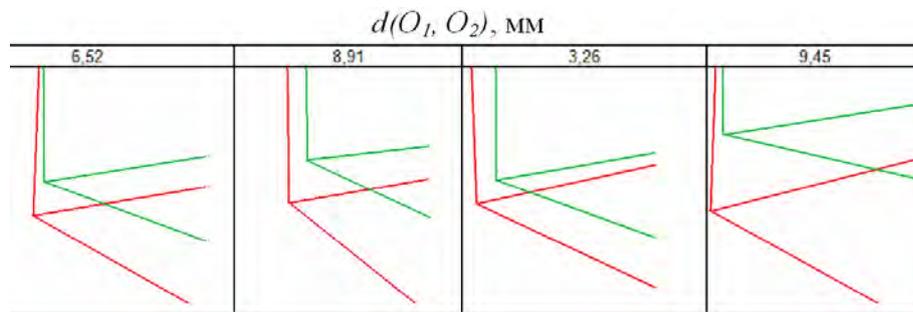


Рис. 3.18. Положение нижней системы координат с центром O_2 (красный цвет) относительно верхней системы координат с центром O_1 (зелёный цвет) для четырёх снимков из видеоряда. Числа соответствуют евклидову расстоянию между O_1 и O_2 в мм.

предложенного метода сегментации модели лица.

3.4. Основные выводы

1. Задача оценки асимметрии человеческого лица по данным, полученным в результате трёхмерного сканирования, может быть сведена к сравнению двух поверхностей, заданных нерегулярными облаками точек. Сравнению подлежат исходное облако и его копия, полученная путём зеркального отражения относительно какой-либо плоскости. В качестве меры асимметрии предлагается использовать расстояние между этими поверхностями, полученное в результате подгонки с использованием ка-

кой-либо из предложенных мер. Процесс подгонки сводится к варьированию параметров плоскости, используемой для получения зеркально отражённой модели лица.

2. Полученное в процессе подгонки положение исходной и отражённой поверхностей даёт возможность оценить меру их близости, которая может служить адекватной мерой асимметрии лица. Кроме того, положение плоскости зеркального отражения, соответствующей наибольшему совпадению исходной и отражённой моделей, может рассматриваться в качестве плоскости симметрии лица.
3. Сформулирована численная постановка задачи оценки асимметрии лица по трёхмерной модели: введены формальные понятия симметрии и асимметрии для модели, предложена количественная оценка асимметрии.

Разработан алгоритм её вычисления, основанный на сравнении исходной и отражённой масок лица и поиске оптимальной плоскости симметрии исходной маски. Получено экспериментальное подтверждение устойчивости предложенной оценки.

4. Сформулирована постановка задачи сегментации трёхмерной модели лица на статические (верхняя часть лица) и динамические (нижняя челюсть) части по трёхмерной видеопоследовательности процесса жевания. Предложен метод, позволяющий сегментировать трёхмерную модель и описывать динамику движения подвижной части относительно статической. Метод основывается на отдельной подгонке верхней и нижней частей моделей лиц. Проведены вычислительные эксперименты на реальных данных, показывающие реализуемость и корректность предложенной модели.

Заключение

В работе рассмотрена задача сравнения поверхностей объектов, представленных неструктурированными облаками точек в трёхмерном пространстве. Предложены меры для сравнения таких поверхностей и новые алгоритмы вычисления мер, сохраняющие исходную нерегулярность множеств точек и обладающие высокой вычислительной эффективностью.

Предложенный в работе подход основан на вычислении и подгонке кусочно-линейных моделей исходных поверхностей на триангуляциях Делоне. Основным этапом подхода является разработанный оригинальный метод локализации узлов триангуляций Делоне друг в друге с помощью построения и обхода минимальных остовных деревьев триангуляций.

В работе введена мера сравнения двух моделей поверхностей для случая большого различия в уровнях их детализации. Предложен эффективный алгоритм вычисления меры, основанный на перечислении множества интерфейсных граней объединённой триангуляции Делоне. Доказана линейная по общему количеству точек в двух моделях оценка сложности алгоритма

Для обоснования реализуемости предложенного решения в работе рассмотрены приложения разработанных методов для некоторых задач анализа моделей человеческих лиц, полученных методами трёхмерного сканирования. Разработан метод оценки асимметрии трёхмерной модели лица и метод сегментации поверхности модели на статические и динамические области по трёхмерному видеоряду.

Предложенный подход обоснован математическими доказательствами и проверен в вычислительных экспериментах с помощью разработанной программной реализации.

Литература

- [1] *Амелин В. В., Князь В. А.* Объединение фрагментов трёхмерной модели объекта // Труды 12й международной конференции по компьютерной графике и машинному зрению ГрафиКон'2002. — Нижний Новгород, 2002. — С. 99–103.
- [2] *Ахо А., Хопкрофт Дж., Ульман Дж.* Построение и анализ вычислительных алгоритмов: Пер. с англ. А. О. Слисенко. — Москва: Издательство «Мир», 1979. — 536 с.
- [3] *Гордеев Д. В., Дышкант Н. Ф.* Построение модели динамики движения челюсти человека в процессе жевания по серии трёхмерных изображений // Труды 19й международной конференции по компьютерной графике и машинному зрению ГрафиКон'2009. — Москва, МГУ, 2009. — С. 348–352.
- [4] *Гордеев Д. В., Дышкант Н. Ф.* Сегментация модели лица на статические и динамические области по трёхмерной видеопоследовательности // Докл. всеросс. конф. Математические методы распознавания образов-14. — М: МАКС Пресс, 2009. — С. 329–332.
- [5] *Дзараев Ч. Р., Персин Л. С., Порохин А. Ю.* Применение 3D сканеров при диагностики зубочелюстных аномалий (МГМСУ) // Доклады Всероссийского научно-практического форума «Дентал-Ревю — 2010». — Москва, 2010.
- [6] *Делоне Б. Н.* О пустоте сферы // Изв. АН СССР, ОМЭН. — 1934. — № 4. — С. 793–800.
- [7] *Дышкант Н. Ф., Местецкий Л. М.* Сравнение 3D портретов при распозна-

- вании лиц // Докл. всеросс. конф. Математические методы распознавания образов-13. — М: МАКС Пресс, 2007. — С. 314–316.
- [8] *Дышкант Н. Ф.* Операции над функциями, заданными на разных нерегулярных двумерных сетках // Сборник тезисов XV Международной научной конференции студентов, аспирантов и молодых учёных «Ломоносов-2008». — М: МАКС Пресс, 2008. — С. 32.
- [9] *Дышкант Н. Ф., Местецкий Л. М.* Оценка асимметрии лица по трёхмерному портрету // Интеллектуализация обработки информации (ИОИ-2008): Тез. докл. Междунар. науч. конф. — Симферополь: Крымский НЦ НАН Украины, 2008. — С. 94–96.
- [10] *Дышкант Н. Ф., Местецкий Л. М.* Оценка асимметрии лица по трёхмерному портрету // Таврический вестник информатики и математики. — 2008. — № 1. — С. 189–198.
- [11] *Дышкант Н. Ф.* Метод сравнения формы пространственных объектов // Сборник тезисов лучших дипломных работ 2008 года. — Москва: Изд. отдел ф-та ВМК МГУ, 2008. — С. 69–70.
- [12] *Дышкант Н. Ф.* Оценка мимической динамики движения челюсти в процессе жевания по трёхмерному видеоряду // Сборник тезисов XVI Международной научной конференции студентов, аспирантов и молодых учёных «Ломоносов-2009». — М: МАКС Пресс, 2009. — С. 28.
- [13] *Дышкант Н. Ф.* Сравнение и подгонка поверхностей при решении прикладных задач анализа 3d портретов человеческих лиц // Тез. докл. конф. Техническое зрение в системах управления-2011. — Москва, ИКИ РАН, 2011. — С. 76–77.

- [14] *Дышкант Н. Ф., Местецкий Л. М.* Сравнение однолистных поверхностей полученных при 3D сканировании // Труды 18й международной конференции по компьютерной графике и машинному зрению Графи-Кон'2008. — Москва, МГУ, 2008. — С. 270–277.
- [15] *Дышкант Н. Ф.* Сравнение поверхностей, заданных на неструктурированных сетках и сетках разной плотности // Доклады 8-й Международной конференции «Интеллектуализация обработки информации» (ИОИ-2010). — М.:МАКС Пресс, 2010. — С. 339–342.
- [16] *Князь В. А., Крыченков В. Ф., Матвеев В. А., Мурынин А. Б., Салтыкова Е. Б.* Исследования корреляций между асимметриями лица и патологиями зрения человека методами трёхмерного моделирования // Труды института Системного анализа РАН, Динамика неоднородных систем. — 2007. — № 4. С. 223–228.
- [17] *Колесов А., Павлова О.* Пакет Surfer — обработка и визуализация двумерных функций // КомпьютерПресс. — 1999. — № 2/99.
- [18] *Кормен Т., Лейзерсон Ч., Ривест Р.* Алгоритмы: построение и анализ. — М.: МЦНМО: БИНОМ. Лаборатория знаний, 2004. — 960 с.
- [19] *Костюк Ю. Л., Фукс А. Л.* Визуально гладкая аппроксимация однозначной поверхности, заданной нерегулярным набором точек // Геоинформатика–2000: Труды международной научно-практической конференции. — Томск: Изд-во Томского ун-та, 2000. — С. 41–45.
- [20] *Костюк Ю. Л.* Графический поиск с использованием триангуляции и клеточного разбиения // Вестник Томского гос. ун-та. — 2002. — № 275. — С. 147–152.

- [21] *Кэмп М. С., Вонг В. В., Филип Э., Картер К. С., Губта С С.* Старение периорбитальной области: количественный анализ // Инъекционные методы в косметологии. — 2010. — № 4.
- [22] *Левицкий В. В.* Разработка системы трёхмерной визуализации лица и зубных рядов и её применение в стоматологической клинике: Автореф. дис. . . канд. мед. наук: 14.00.21 / ЦНИИС и ЧЛХ. — М., 2008. — 23 с.
- [23] *Марков К. Н., Ширков П. Д.* Алгоритмы сглаживания поверхностей, заданных на нерегулярных сетках // Матем. моделирование. — 2009. — Т. 21, № 6. — С. 69–78.
- [24] *Местецкий Л. М.* Непрерывная морфология бинарных изображений: фигуры, скелеты, циркуляры. — М.: ФИЗМАТЛИТ, 2009. — 288 с.
- [25] *Местецкий Л. М., Царик Е. В.* Триангуляция Делоне: рекурсия без пространственного разделения точек // Труды международной конференции по компьютерной графике и машинному зрению ГрафиКон'2004. — Москва, МГУ, 2004. — С. 267–270.
- [26] *Местецкий Л. М., Царик Е. В.* Слияние неразделённых триангуляций Делоне // Сложные системы: обработка информации, моделирование и оптимизация: Сборник научных трудов. Вып. 2. — Тверь: Тверской гос. университет, 2004. — С. 216–231.
- [27] *Онопрыйко М. Д., Попов Е. В.* Создание NURBS поверхностей в системе трёхмерного компьютерного моделирования КЗ // Труды международной конференции по компьютерной графике и машинному зрению ГрафиКон'2001 — Нижний Новгород, 2001. — С. 145–149.
- [28] *Препарата Ф., Шеймос М.* Вычислительная геометрия: Введение: Пер. с англ. — М.: Мир, 1989. — 478 с.

- [29] *Скворцов А. В.* Обзор алгоритмов построения триангуляции Делоне // Вычислительные методы и программирование. — 2002. — № 3 — С. 14–39.
- [30] *Скворцов А. В.* Триангуляция Делоне и её применение. — Томск: Изд-во Томского ун-та, 2002. — 128 с.
- [31] *Скворцов А. В., Костюк Ю. Л.* Эффективные алгоритмы построения триангуляции Делоне // Геоинформатика. Теория и практика. — № 1. — Томск: Изд-во Томского ун-та, 1998. — С. 22–47.
- [32] *Форсайт Д., Понс Ж.* Компьютерное зрение. Современный подход. — Изд-во Вильямс, 2004. — 928 с.
- [33] *Яблонский С. В.* Введение в дискретную математику: Учеб. пособие для вузов. — 2-е изд., перераб. и доп. — М.: Наука. Гл. ред. физ.-мат. лит., 1986. — 394 с.
- [34] *Alliez P., Ucelli G., Gotsman C., Attene M.* Recent Advances in Remeshing of Surfaces // Shape Analysis and Structuring: Mathematics and Visualization:. — 2008. — Pp. 53–82.
Pierre Alliez, Giuliana Ucelli, Craig Gotsman and Marco Attene
- [35] *Bern M., Eppstein D.* Mesh generation and optimal triangulations — In D. Z. Du and F.K. Hwang, editors, Computing in Euclidean Geometry. World Scientific Publishing Co., 1992. — 78 p.
- [36] *Besl P., McKay H.* A method for registration of 3-d shapes // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 1992. — Vol. 14, no. 2. — Pp. 239–256.
- [37] *Boissonnat J.-D., Teillaud M.* Effective Computational Geometry for Curves and Surfaces — Springer-Verlag Berlin Heidelberg, 2006. — 343 p.

- [38] *Bose P., Devroye L.* Intersections with Random Geometric Objects // Computational Geometry: Theory and Applications. — 1998. — Vol. 10, no. 3. — Pp. 139–154.
- [39] *Brunelli R., Poggio T.* Face recognition: features versus templates // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 1993. — Vol. 15, no. 10. — Pp. 1042–1052.
- [40] *Brunnstrom;K., Stoddart A. J.* Genetic algorithms for free-form surface matching // Proc. ICPR. — 1996. — Pp. 689–693.
- [41] *Chen Y., Medioni G.* Object modelling by registration of multiple range images // Image and Vision Computing. — 1992. — Vol. 10, no. 3. — Pp. 145–155.
- [42] *Cheriton D., Tarjan R. E.* Finding minimum spanning trees // SIAM J. Comput. — 1976. — Vol. 5, no. 4. — Pp. 724–742.
- [43] *Cignoni P., Rocchini C., Scopigno R.* Metro: measuring error on simplified surfaces // Computer Graphics Forum, Blackwell Publishers. — 1998. — Vol. 17, no. 2. — Pp. 167–174.
- [44] *Clarkson K.* A randomized algorithm for closest point queries // SIAM J. Computing. — 1998. — Vol. 17. — Pp. 830–847.
- [45] *Devillers O., Pion S., Teillaud M.* Walking in a triangulation // Internat. J. Found. Comput. Sci. 13. — 2002. — Pp. 181–199.
- [46] *Devroye L., Lemaire C., Moreau J.* Expected time analysis for Delaunay point location // Computational geometry. — 2004. — Vol. 29, no. 2. — Pp. 61–89.

- [47] *Devroye L., Mucke E. P., Zhu B.* A note on point location in Delaunay triangulations of random points // *Algorithmica*. — 1998. — Vol. 22. — Pp. 477–482.
- [48] *Dyshkant N.* An algorithm for calculating the similarity measures of surfaces represented as point clouds // *Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications*. — 2010. — Vol. 20, no. 4. — Pp. 495–504.
- [49] *Dyshkant N.* Disparity Measure Construction for Comparison of 3D Objects' Surfaces // *Proceedings of the Workshop IMTA*. — Lisbon, Portugal: INSTICC Press, 2009. — Pp. 43–52.
- [50] *Dyshkant N., Mestetskiy L.* Estimation of Asymmetry in 3D Face Models // *Proceedings of International conference on computer vision theory and applications (VISAPP 2009)*. — Lisbon, Portugal: INSTICC Press, 2009. — Pp. 402–405.
- [51] *Dyshkant N.* Measures for Surface Comparison on Unstructured Grids with Different Density // *Lecture Notes in Computer Science: Discrete Geometry for Computer Imagery*. — 2011. — Vol. 6607. — Pp. 501–512.
- [52] *Eggert D. W., Larusso A., Fisher R. B.* Estimating 3-D rigid body transformations: a comparison of four major algorithms // *Machine Vision and Applications*. — 1997. — Vol. 9, no. 5-6. — Pp. 272–290.
- [53] *Enciso R., Memon A., Fidaleo D. A., Neumann U., Mah J.* The Virtual Craniofacial Patient: 3D Jaw Modeling and Animation // *The 11th Annual Medicine Meets Virtual Reality Conference*. — 2003. — Pp. 65–71.
- [54] *Fary I.* On straight-line representations of planar graphs // *Acta a Sci. Math.* (Szeged). — 1948. — Vol. 11. — Pp. 229–233.

- [55] *Fan T., Medioni G., Nevatia R.* Recognizing 3D objects using surface descriptions // IEEE PAMI. — 1989. — Vol. 11, no. 11. — Pp. 1140–1157.
- [56] *Feldmar J., Ayache N., Betting F.* D-2D projective registration of free-form curves and surfaces // CVIU. — 1997. — Vol. 65. — Pp. 403–424.
- [57] *Tarjan R. E.* Data Structures and Network Algorithms — Society for Industrial and Applied Mathematics, 1983. — 131 p.
- [58] *Friedman J. H., Bentley J. L., Finkel R. A.* An Algorithm for Finding Best Matches in Logarithmic Expected Time // ACM Transactions on Mathematical Software. — 1977. — Vol. 3, no. 3. — Pp. 209–226.
- [59] *Gatzke T., Zelinka S., Grimm C., Garland M.* Curvature Maps for Local Shape Comparison // In: Shape Modeling International. — 2005. — Pp. 244–256.
- [60] *Gelfand N., Ikemoto L., Rusinkiewicz S., Levoy M.* Geometrically Stable Sampling for the ICP Algorithm // Fourth International Conference on 3D Digital Imaging and Modeling. — 2003. — Pp. 260–267.
- [61] *Godin G., Rioux M., Baribeau R.* Three-dimensional registration using range and intensity information // Proceedings of the SPIE. — 1994. — Vol. 2350. — Pp. 279–290.
- [62] *Gordon G. G.* Face recognition based on depth maps and surface curvature // In SPIE Geometric Methods in Computer Vision. — 1991. — Vol. 1570. — Pp. 234–247.
- [63] *Gruen A., Akca D.* Least Squares 3D Surface and Curve Matching // ISPRS Journal of Photogrammetry and Remote Sensing. — 2005. — Vol. 59. — Pp. 151–174.

- [64] *Gu X., Gortler S. J., Hoppe H.* Geometry images // Computer graphics proceedings, annual conference series: SIGGRAPH conference proceedings. — 2002. — Pp. 355–361.
- [65] *Guskov I., Vidimce K., Sweldens W., Schroeder P.* Normal meshes // Computer graphics proceedings, annual conference series: SIGGRAPH conference proceedings. — 2000. — Pp. 95–102.
- [66] *Hajeer M. Y., Millet D. T., Ayoub A. F., Siebert J. P.* Applications of 3D imaging in orthodontics // Journal of Ortodontics. — 2004. — Vol. 31, no. 1. — Pp. 62–70.
- [67] *Haran I., Helperin D.* An Experimental Study of Point Location in Planar Arrangements in Cgal // ACM Journal of Experimental Algorithms. — 2009. — Vol. 13, no. 3. — Pp. 1–31.
- [68] *Huang J., Heisele B., Blanz V.* Component-based Face Recognition with 3D Morphable Models // In International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA-03). — 2003. — Pp. 27–34.
- [69] *Kirkpatrick D. G.* Optimal search in planar subdivisions // SIAM J. Comput. — 1983. — Vol. 12, no. 1. — Pp. 28–35.
- [70] *Koidis P., Patias P., Tsioukas V.* 3D Visualization of Dental Data for Virtual Treatment Planning // ISPRS Congress Istanbul 2004, Proceedings of Commission V — 2004. — Pp. 996–1001.
- [71] *Koseki M., Niitsuma A., Inou N., Maki K.* Three-dimensional Display System of Individual Mandibular Movement // Complex Medical Engineering, (Springer), — 2007. — Pp. 117–127.

- [72] *Knyaz V. A., Zheltov S. Yu.* Photogrammetric Techniques for Dentistry Analysis, Planning and Visualisation // ISPRS Congress Beijing 2008, Proceedings of Commission V. — 2008. — Pp. 783–788.
- [73] *Lee D. T., Schachter B. J.* Two Algorithms for Constructing a Delaunay Triangulation // International Journal of Computer and Information Science. — 1980. — Vol. 9, no. 3. — Pp. 219–242.
- [74] *Liu Y., Palmer J.* A Quantified Study of Facial Asymmetry in 3D Faces // Proceedings of the 2003 IEEE International Workshop on Analysis and Modeling of Faces and Gestures, in conjunction with the 2003 International Conference of Computer Vision (ICCV '03). — 2003. — Pp. 222–231.
- [75] *Liu Y., Rodrigues M. A.* Geometrical analysis of two sets of 3D correspondence data patterns for the registration of free-form shapes // J. Int. and Rob. Systems. — 2002. — Vol. 33. — Pp. 409–436.
- [76] *Mccool C., Cook J., Chandran V., Sridharan S.* Feature Modelling of PCA Difference Vectors for 2D and 3D Face Recognition // In Video and Signal Based Surveillance, AVSS '06. IEEE International Conference. — 2006. — Pp. 57.
- [77] *Mitra N J., Guibas L. J., Pauly M.* Partial and approximate symmetry detection for 3D geometry // In ACM SIGGRAPH. — 2006. — Pp. 560–568.
- [78] *Mitra S., Lazar N., Liu Y.* Understanding the Role of Facial Asymmetry in Human Face Identification // Statistics and Computing. — 2007. — Vol. 17. — Pp. 57–70.
- [79] *Mitra S., Liu Y.* Local Facial Asymmetry for Expression Classification // Proceedings the 2004 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04). — 2004. — Vol. 2 — Pp. 889–894.

- [80] *Mohan A., Papageorgiou C., Poggio T.* Example-based object detection in images by components // In IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2001. — Vol. 23, no. 4. — Pp. 349–361.
- [81] *Mucke E., Sias I., Zhu B.* Fast Randomized Point Location Without Preprocessing in Two- And Three-dimensional Delaunay Triangulations // Proceedings of the 11th Annual Symposium on Computational Geometry. — 1996. — Pp. 274–283.
- [82] *Murynin A. B., Krychenkov V. F., Matveev I. A., Kuznetsov V. D.* Estimation of Bilateral Facial Symmetry Deviation Using a Stereoscopic Computer Vision System // Pattern Recognition and Image Analysis. — 2001. — Vol. 11, no. 2. — Pp. 350–352.
- [83] *Nelder J. A., Mead R.* A simplex method for function minimization // Computer Journal. — 1965. — Vol. 7. — Pp. 308–313.
- [84] *Ohbuchi R., Takei T.* Shape-similarity comparison of 3D models using alpha shapes // Proceedings of the 11th Pacific Conference on Computer Graphics and Application. — 2003. — Pp. 293–302.
- [85] *Prim R. C.* Shortest connecting networks and some generalizations // Bell Systems Techn. J. — 1957. — Vol. 36. — Pp. 1389–1401.
- [86] *Shamos M. I.* Computational geometry. — Ph.D. thesis, Dept. of Comput. Sci., Yale Univ. — 1978.
- [87] *Shapiro M.* A note on Lee and Schachter's algorithm for Delaunay triangulation // Inter. Jour. of Comp. and Inf. Sciences. — 1981. — Vol. 10, no. 6. — Pp. 413–418.

- [88] *Schenk T.* Digital Photogrammetry. — Terra-Science, Laurelville, Ohio, 1999. — 428 p.
- [89] *Slingsby A.* An Object-Orientated Approach to Hydrological Modelling using Triangular Irregular Networks // Proceedings of GISRUK03, City University, London, UK. — 2003.
- [90] *Stepanyants D. G., Knyaz V. A.* PC-Based Digital CloseRange Photogrammetric System for Rapid 3D Data Input in CAD Systems // International Archives of Photogrammetry and Remote Sensing. — 2000. — Vol. 33, Part B5. — Pp. 756–763.
- [91] *Szymczak A., Rossignac J., King D.* Piecewise regular meshes: Construction and compression // Graphical Models. — 2002. — Vol. 64, no. 3-4. — Pp. 183–198.
- [92] *Tarjan R. E.* Fibonacci heaps and their uses in improved network optimization algorithms // Journal of the Association for Computing Machinery. — 1987. — Vol. 34, no. 3. — Pp. 596–615.
- [93] *Teng K., Liu Y.* Expression Classification using Wavelet Packet Method on Asymmetry Faces // tech. report CMU-RI-TR-06-03, Robotics Institute, Carnegie Mellon University. — January 2006.
- [94] *Tomaka A.* The application of 3d surfaces scanning in the facial features analysis // Journal of Medical Informatics and Technologies. — 2005. — Pp. 233–240.
- [95] *Turk G., Levoy M.* Zippered polygon meshes from range images // Proc. SIGGRAPH. — 1994. — Pp. 311–318.

- [96] *Vaillant M., Glaunes J.* Surface matching via currents // Lecture Notes in Computer Science: Information Processing in Medical Imaging. — 2005. — Vol. 3565. — Pp. 1–5.
- [97] *Zhang Z.* Iterative point matching for registration of freeform curves and surfaces // International Journal of Computer Vision. — 1994. — Vol. 13, no. 2. — Pp. 119–152.