

Федеральное государственное бюджетное учреждение науки  
Санкт-Петербургский институт информатики и автоматизации  
Российской академии наук (СПИИРАН)

*На правах рукописи*



АКСЕНОВ  
Алексей Юрьевич

МОДЕЛИ И МЕТОДЫ ОБРАБОТКИ И ПРЕДСТАВЛЕНИЯ  
СЛОЖНЫХ ПРОСТРАНСТВЕННЫХ ОБЪЕКТОВ

*Специальность 05.13.01 – Системный анализ, управление и обработка  
информации (технические системы)*

Диссертация на соискание ученой степени  
кандидата технических наук

Научный руководитель:  
доктор технических наук,  
Кулешов С.В.

Санкт-Петербург, 2015

## СОДЕРЖАНИЕ

Список используемых сокращений .....	4
Введение .....	5
Положения, выносимые на защиту .....	16
Глава 1. Анализ процессов получения и представления пространственных данных .....	17
1.1 Общее понятие о процессе перевода пространственных объектов в трехмерный цифровой образ.....	17
1.2 Методы получения пространственных трехмерных данных .....	20
1.2.1 Программные продукты для синтеза 3D-объектов .....	21
1.2.2 3D-сканирование пространственных форм.....	22
1.2.3 Компьютерные томографы .....	26
1.3 Анализ существующих моделей и методов представления трехмерных объектов.....	29
1.3.1 Модели на основе полигональных сеток.....	31
1.3.2 Модели на основе вокселей .....	33
1.3.3 Модели, основанные на картах глубины.....	34
1.3.4 Модели на основе точечных представлений (облака точек).....	36
1.4 Форматы представления трехмерных данных .....	37
Выводы по главе 1.....	38
Глава 2. Методы компрессии трехмерных данных .....	41
2.1 Анализ методов компрессии без потерь .....	43
2.2 Анализ методов компрессии с потерями.....	47
Выводы по главе 2.....	50
Глава 3. Разработка модели, методов и алгоритма компрессии для хранения и передачи трехмерных пространственных данных.....	51
3.1 Виды разверток.....	53
3.2 Заполняющие пространство кривые и дискретные пространства .....	56

3.3	Динамическое разбиение и масштабирование пространства облаков точек	63
3.4	Разработка модели представления пространственных объектов.....	70
3.5	Разработка метода и алгоритма сжатия пространственных объектов.....	72
3.6	Программная реализация системы сжатия пространственных объектов .....	75
	Выводы по главе 3.....	77
Глава 4	Экспериментальная апробация и оценка эффективности метода сжатия.	79
4.1	Виды объектов.....	81
4.2	Исследование особенностей технической системы «3D-сканер Artec Spider – программное обеспечение сканера – система сжатия».....	82
4.3	Оценка эффективности метода компрессии.....	90
	Выводы по главе 4.....	93
	Заключение .....	94
	Список литературы .....	96
Приложение А.	Фрагменты программного кода системы для обработки облаков точек .....	106
Приложение Б.	Акты внедрения.....	108

## СПИСОК ИСПОЛЬЗУЕМЫХ СОКРАЩЕНИЙ

**3D** — (от англ. 3-dimensional), трехмерный

**CAD** — Computer-Aided Design — Система автоматизированного проектирования

**JPEG** — англ. Joint Photographic Experts Group, – формат представления графических данных

**JSON** — (англ. JavaScript Object Notation) — текстовый формат обмена данными

**LDI** — Layered Depth Images – многослойные изображения с глубиной

**PLY, STL, XYZ, OBJ, PTX, E57** — форматы представления 3D-данных

**PPM** — (англ. Prediction by Partial Matching) — предсказание по частичному совпадению

**RAW** — (англ. raw — сырой, необработанный) — формат представления необработанных данных

**RLE** — (англ. run-length encoding) – кодирование длин серий, групповое кодирование

**TIN** — Triangulated Irregular Network, нерегулярная триангуляционная сеть неперекрывающихся треугольников

**XML** — (англ. eXtensible Markup Language — расширяемый язык разметки, универсальный формат представления данных

**ZIP** — формат сжатия данных и архивации файлов

**АЦП** — аналогово-цифровой преобразователь

**ЗПК** — заполняющая пространство кривая (zpk)

**КТ** — компьютерная томография

**ПО** — программное обеспечение

**ПОС** — программное обеспечение 3D-сканера

## ВВЕДЕНИЕ

**Актуальность темы диссертации.** На протяжении многих лет человечество искало пути представления пространственного положения объектов, а также способы описания самих этих объектов. Начиная с Р.Декарта и Г.Монжа [1–2] сформировались парадигмы прямоугольного представления координат в пространстве и способы описания объектов средствами начертательной геометрии путем построения моделей по проекциям [3].

С появлением нового класса цифровых устройств (цифровых камер, цифровых томографов, 3D-сканеров), способных создавать «цифровые сканы» (копии) материальных объектов основным способом представления стала являться запись свойств и координат отдельных точек объекта. Цифровое сканирование – потенциально новая парадигма, которая расширяет возможности проектирования объемных объектов, переводя процесс проектирования от работы с проективной геометрией к непосредственной работе с точками объекта в трехмерном пространстве, тем самым упрощая работу в тех областях деятельности, где проективный подход был затруднен (например, медицинские исследования, томография, ландшафтный дизайн и т.п.).

Независимо от физического принципа работы таких устройств и спектральной области их чувствительных элементов (оптические сканеры, рентгеновская томография, ультразвуковое сканирование) они формируют «цифровой скан» объекта, позволяющий работать с ним, абстрагируясь от способов его получения.

Основным недостатком цифрового скана является его большой битовый объем, так как в его цифровом представлении приходится перечислять все элементы сетки, относящиеся к объекту и не относящиеся к нему. Особенно сильно увеличение объема проявляется при трехмерном представлении сложных пространственных объектов с высокой детализацией [4].

Особо остро этот недостаток проявляется в тех случаях, когда сканирование производится в «полевых условиях», что часто происходит, например, при работе с объектами искусства, а также в задачах дистанционного мониторинга, сканирования и необходимости дистанционной обработки «сырых» цифровых сканов.

В практических задачах для экономии памяти или ресурсов каналов связи применяется компрессия данных. В методах сжатия применяются теоретические основы, разработанные Ziv J., Lempel A., Welch T.A., Katz P. W. [5–6], Huffman D.A. [7], В. В. Александровым [8], Д.С. Ватолиным [9] и др.

Все методы сжатия можно разделить на сжатие с потерями (в этом случае не гарантируется полное восстановление исходного описания объекта) и сжатие без потерь, когда такое восстановление гарантируется. Следует отметить, что если для изображений и видеоданных разработано значительное количество методов сжатия [10–15], многие из которых стали стандартом, то для 3D-данных готовых решений для сжатия на данный момент практически не существует, все доступные методы представлены в виде экспериментальных прототипов. При этом за последние годы наблюдается повышение доступности технологий 3D-сканирования. В связи с этим разработка методов сжатия облаков точек, а также предложения отраслевого стандарта для такого сжатия являются актуальной задачей.

В данной диссертационной работе разрабатывается метод сжатия без потерь, ориентированный на работу с цифровыми сканами — данными, получаемыми в результате 3D-сканирования и предназначенными для последующей обработки и репликации на устройствах 3D-печати.

Анализ актуальных исследований выявил практически полное отсутствие готовых комплексных решений для оптимизации цифрового скана сложного пространственного объекта. Причиной этого является отсутствие эффективных методов сжатия без потерь трехмерных данных, получаемых в результате 3D-сканирования. Решение указанных задач, применительно к реализации метода

уменьшения битового объема без потерь для отсканированных 3D-объектов в пределах разрешающей способности, представленных в форме облаков точек, и составляет суть диссертационной работы.

**Целью работы** является исследование методов цифрового сканирования и разработка методов представления и обработки с целью уменьшения объема и хранения цифровых сканов, полученных в результате 3D-сканирования. Для достижения поставленной цели в диссертационной работе поставлены и решены следующие задачи:

1. Анализ возможностей цифровых систем получения, представления и обработки пространственных данных.
2. Разработка метода динамического масштабирования пространства облаков точек.
3. Разработка метода переупорядочения в битовый поток, имеющий упорядоченную структуру, обеспечивающую эффективное вторичное сжатие для передачи и хранения.
4. Разработка алгоритма и формата представления цифровых сканов с целью уменьшения битового объема их цифрового представления.
5. Разработка интерактивной программной системы для обработки облаков точек, представляющих собой цифровые сканы.

**Объектом исследований** является процесс репликации материальных объектов с использованием цифровых программируемых технологий.

**Предметом исследований** являются методы и алгоритмы представления и сжатия цифровых сканов сложных пространственных объектов.

**Основные методы исследования.** Для решения поставленных задач в работе используются системный и инфокоммуникационный подход, методы компьютерной графики, 3D-моделирования, теория множеств и отношений. При разработке архитектуры программной интерактивной системы применены компонентно- и объектно- ориентированные подходы.

**Научная новизна** предлагаемой диссертации состоит в следующем:

1. Предложен метод переупорядочения облака точек в битовый поток, имеющий упорядоченную структуру, отличающийся сохранением локальных особенностей областей пространств, что позволяет использовать методы группового кодирования для уменьшения битового объема.
2. Разработан метод динамического разбиения и масштабирования пространства облаков точек, учитывающий специфику технологии бесконтактного оптического трехмерного сканирования.
3. Разработан алгоритм сжатия облака точек, отличающийся применением заполняющей пространство кривой для переупорядочения (трансформации) облаков точек и не требующий для работы восстановленной поверхности объекта (набора полигонов).
4. Разработана модель представления пространственных объектов, использующая упорядоченное одномерное представление облаков точек на основе заполняющей пространство кривой.
5. Разработана интерактивная система сжатия облаков точек, отличающаяся применением динамического разбиения и масштабирования пространства и заполняющей пространство кривой.

**Обоснованность и достоверность** научных положений, основных выводов и результатов диссертации обеспечиваются всесторонним тщательным анализом состояния исследований в данной области на сегодняшний день. Корректность предложенных методов и алгоритмов подтверждается согласованностью теоретических положений диссертационной работы и результатов, полученных при практической реализации этих методов и алгоритмов, а также апробацией основных теоретических положений диссертации в печатных трудах и докладах на российских и международных научных конференциях.

**Практическая ценность работы** заключается в создании интерактивной программной системы, реализующей теоретические результаты работы и предназначенной для работы с облаками точек и их сжатия.



Предложенные в диссертационной работе подходы, методы и алгоритмы позволяют повысить (в 2 и более раз по сравнению с существующими аналогами) эффективность представления и передачи пространственных данных, полученных при 3D-сканировании.

**Реализация результатов работы.** Представленные в работе методы и алгоритмы были программно реализованы в виде объектно-ориентированной библиотеки классов на языке Java и других вспомогательных программ. Библиотека нашла применение в рамках проектов ОНИТ РАН, НОЦ Курск и учебных курсов.

**Апробация результатов работы.** Научные результаты и основные положения работы представлялись на конференциях: Всероссийская конференция «Информационные технологии в управлении» (ИТУ-2014), Санкт-Петербург, 2014; International Conference of Young Scientists AUTOMATION & CONTROL (Saint-Petersburg, State Polytechnical University, 2013); «Информационные технологии в управлении» (ИТУ–2012), г. Санкт-Петербург, 2012; VII Международная научно-практическая конференция «Регионы России: стратегии и механизмы модернизации, инновационного и технологического развития», г. Москва, 2011 г.; 10-я Международная конференция «Распознавание образов и анализ изображений: новые информационные технологии» РОАИ-10-2010, г. Санкт-Петербург, 2010, VIII всероссийская научно-практическая конференция с международным участием "Современные информационные технологии в науке, образовании и практике", Оренбург, 2009; 9-я Санкт-Петербургская международная конференция «Региональная информатика – 2004», СПб, 2004.

**Публикации.** Автором опубликовано по теме диссертации 11 печатных работ, среди них 6 работ в рецензируемых журналах из перечня ВАК [16–27].

**Структура и объем диссертационной работы.** Диссертация объемом 110 машинописных страниц содержит введение, 4 главы и заключение, список литературы (95 наименований), 57 рисунков, 7 таблиц и 2 приложения.

## Содержание работы

**В первой главе** проводится анализ современных методов получения пространственных данных.

С развитием цифровых программируемых инфокоммуникационных технологий наметилась тенденция к сведению накопленного опыта по созданию цифровых образов материальных объектов в унифицированные технологические процессы, которые постепенно становятся доступными массовому потребителю. Как и в любой новой технологии, данный этап развития характеризуется накоплением опыта, разработкой новых сервисных функций, стандартизацией и унификацией отдельных компонентов, образующих единый жизненный цикл данной технологии.

Одним из путей создания цифровых сканов сложных пространственных объектов является 3D-сканирование (трехмерное сканирование), это направление в последние годы сделало качественный скачок в развитии оборудования и программного обеспечения в рамках программируемых технологий.

Трёхмерное сканирование используется в задачах реверс-инжиниринга, при проектировании оснастки, приспособлений и запасных частей, особенно в условиях отсутствия оригинальной документации на изделие, а также при оцифровке сложных пространственных форм. Часто 3D-сканирование используется в целях сопоставления реплицированного объекта с его цифровым образом, что играет особую роль в задачах медицинского протезирования.

Современные 3D-сканеры являются комбинированными приборами. Во многих из них применяется совмещенная двойная система получения координат 3D-объекта. В дополнение к лазерным датчикам (являющимся развитием идеи механического «щупа» в контактных устройствах) используется цифровая камера, обеспечивающая получение текстурной информации об объекте.

Вместо одиночных лазерных сенсоров могут применяться более сложные системы, такие как системы 3D-сканирования на базе ультразвуковых сенсоров, преимуществом которых является наличие режима сканирования объектов с

внутренней структурой. Имеются также опытные образцы магнитных сканеров, использующих изменение пространственного магнитного поля объекта для локализации его пространственных координат.

Многообразие устройств сканирования создает проблему разработки специализированного программного обеспечения для работы с такими устройствами и интерпретации получаемых с них данных, а также для обеспечения унифицированного человеко-машинного взаимодействия.

С появлением перечисленных устройств, способных создавать цифровые копии реальных объектов основным способом представления данных стала запись свойств и пространственного положения точек, принадлежащих объекту. Основным недостатком такого вида представления является его большой битовый объем.

Следует заметить, что в случае использования портативного 3D-сканера в «полевых условиях» без возможности применения хранилищ данных и серверного оборудования проблема компактного хранения объема отсканированных данных становится определяющей. Особенно важно не допускать потерь данных или ухудшения точности представления в результате применения сжатия с потерями, чтобы иметь возможность корректной постобработки 3D-объекта без вносимых методом сжатия артефактов. В качестве примера необходимости такого подхода можно привести рекомендации в профессиональной фотографии использовать RAW-формат вместо формата сжатия с потерями JPEG, используемого в любительской фотографии.

В зависимости от способа получения 3D-данных занимаемый ими объем битового представления будет различным.

Детальное изучение существующих моделей представления пространственных данных в цифровом виде позволяет классифицировать их следующим образом:

- 1) Полигональные сетки (представление в виде списка граней) являются наиболее распространенным представлением, для которого доступно большое

количество программных продуктов для редактирования и визуализации объектов.

Важной особенностью моделей на основе полигональных сеток является связность, которая упрощает генерацию искусственных поверхностей.

Соответственно применение полигональных моделей требует в ряде задач предварительного этапа определения связности отдельных точек перед дальнейшей обработкой. В связи с этим полигональные модели не могут использоваться для непосредственного хранения цифровых сканов, так как в этом случае требуется решение неоднозначной задачи восстановления поверхностей.

2) Воксельные модели представляют собой регулярный трехмерный массив, каждому элементу которого сопоставлены характеристики цвета и прозрачности. В связи с тем, что в воксельном представлении хранится описание всей области пространства, содержащей в себе 3D-объект, то даже небольшие по геометрическим размерам модели занимают значительный битовый объем.

3) Модели, основанные на картах глубины. Включают в себя модели, основанные на изображениях с картами глубины и многослойных изображениях с глубиной. Сложность реализации методов визуализации и поддержки многомасштабности ограничивают возможности применения этих моделей.

4) Точечные представления (облака точек). Объекты моделируются как набор трехмерных координат точек принадлежащих поверхностям объекта. Основной проблемой таких представлений является отсутствие данных о связности точек и их принадлежности к непрерывным поверхностям сканируемого 3D-объекта.

Основным недостатком большинства рассмотренных в главе моделей, является большой объем их битового представления при программной реализации. Использование неструктурированного набора точек при увеличении объема обуславливает необходимость применения методов упрощения и сжатия, имеющих высокую вычислительную сложность или требующих дополнительно сформированных данных.

**Во второй главе** рассматриваются актуальные решения задачи сжатия 3D-данных, имеющей большое значение для архивации, интерфейсной коммуникации и быстрого прототипирования.

В настоящее время известные методы сжатия данных, полученных в результате 3D-сканирования, делятся на 2 группы:

- методы, требующие для работы предварительного перевода облака точек в набор полигонов (восстановление поверхностей объекта);
- методы, ориентированные на работу непосредственно с облаками точек без предварительной интерполяции поверхности набором полигонов.

Однако в связи с тем, что при сканировании восстановление полигонов производится с помощью прикладного программного обеспечения и является приближенным, то более корректными и перспективными для задач сжатия без потерь являются методы, работающие непосредственно с облаками точек.

Анализ существующих методов сжатия, работающих непосредственно с облаками точек, показал, что все они используют различные алгоритмы обхода точек пространства принадлежащих сканируемому объекту и дальнейшего представления их в виде древовидной структуры (которыми и отличаются друг от друга) и дальнейшего сжатия методами группового кодирования, тем самым используя элементы методов восстановления поверхностей. Главным недостатком является снижение степени сжатия для объектов имеющих сложный рельеф поверхности и внутреннюю структуру.

**В третьей главе** разработана модель представления пространственных объектов, использующая упорядоченное одномерное представление облаков точек на основе заполняющей пространство кривой.

На основе разработанной модели предлагается метод эффективного представления пространственных данных, учитывающий особенности 3D-

сканирования, предназначенный для повышения компактности хранения 3D-моделей в цифровых архивах и упрощения передачи 3D-данных по каналам связи.

**В четвертой главе** исследуются характеристики и свойства технической системы (3D-сканер — программное обеспечение сканера — система компрессии), проводится оценка эффективности предложенного метода компрессии, приводятся результаты проведенных испытаний.

Процесс сканирования 3D-объекта включает в себя ряд последовательных действий, каждое из которых влияет на конечный результат сканирования. В связи с этим программное обеспечение 3D-сканера (ПОС), осуществляющее реконструкцию облака точек из набора изображений, полученных с нескольких камер 3D-сканера, а также его постобработку имеет смысл рассматривать как техническую систему «3D-сканер – ПОС».

Влияние ПОС проявляется в первую очередь в изменении (часто искажении) формы объекта при попытке компенсировать ошибки в работе сканера.

Предложенный метод компрессии пространства точек на основе разбиения областей пространства на элементы размером  $256 \times 256 \times 256$ , с последующим их преобразованием в битовые последовательности с использованием обхода в соответствии с алгоритмом заполняющей пространство кривой и вторичного сжатия дает положительный эффект, что подтверждается результатами работы программы.

Предложенный метод может быть использован в программном обеспечении, предназначенном для работы с 3D-сканерами и 3D-принтерами, пакетах пост-обработки сложных пространственных форм и в системах когнитивного программирования для повышения эффективности использования систем хранения данных 3D-объектов, создания библиотек цифровых сканов и сокращения издержек на архивацию и интерфейсную коммуникацию при их передаче.

**В приложении А** приводятся фрагменты программного кода для обработки облаков точек.

**В заключении** содержится перечень задач, которые были решены в результате диссертационного исследования, а также сведения об апробации и внедрении результатов работы.

## **Положения, выносимые на защиту**

На основе проведенных исследований и экспериментальной апробации результатов на защиту выносятся следующие положения:

1. Метод переупорядочения облака точек в битовый поток, имеющий упорядоченную структуру, обеспечивающую эффективное вторичное сжатие, и последующей компрессии этого потока.
2. Метод динамического разбиения и масштабирования пространства облаков точек для возможности применения заполняющей пространство кривой.
3. Модель представления пространственных объектов, использующая упорядоченное одномерное представление облаков точек на основе заполняющей пространство кривой.
4. Алгоритм сжатия облаков точек, полученных в результате трехмерного сканирования.
5. Алгоритм и интерактивная система сжатия цифрового представления пространственных объектов, полученных в результате трехмерного сканирования.



## Глава 1. Анализ процессов получения и представления пространственных данных

### 1.1 Общее понятие о процессе перевода пространственных объектов в трехмерный цифровой образ

До последнего времени перевод материальных объектов в цифровое 3D-представление встречался в основном в качестве решений отдельных технических задач, причем подходы зачастую не опирались на опыт предыдущих решений. С развитием цифровых программируемых инфокоммуникационных технологий наметилась тенденция к сведению накопленного опыта по созданию цифровых образов материальных объектов в унифицированные технологические процессы, которые постепенно становятся доступными массовому потребителю. Как и в любой новой технологии, данный этап развития характеризуется накоплением опыта, разработкой новых сервисных функций, стандартизацией и унификацией отдельных компонентов, образующих единый жизненный цикл данной технологии.

На данном этапе также формируются требования и ограничения, обусловленные выявленными особенностями и спецификой решаемых задач.

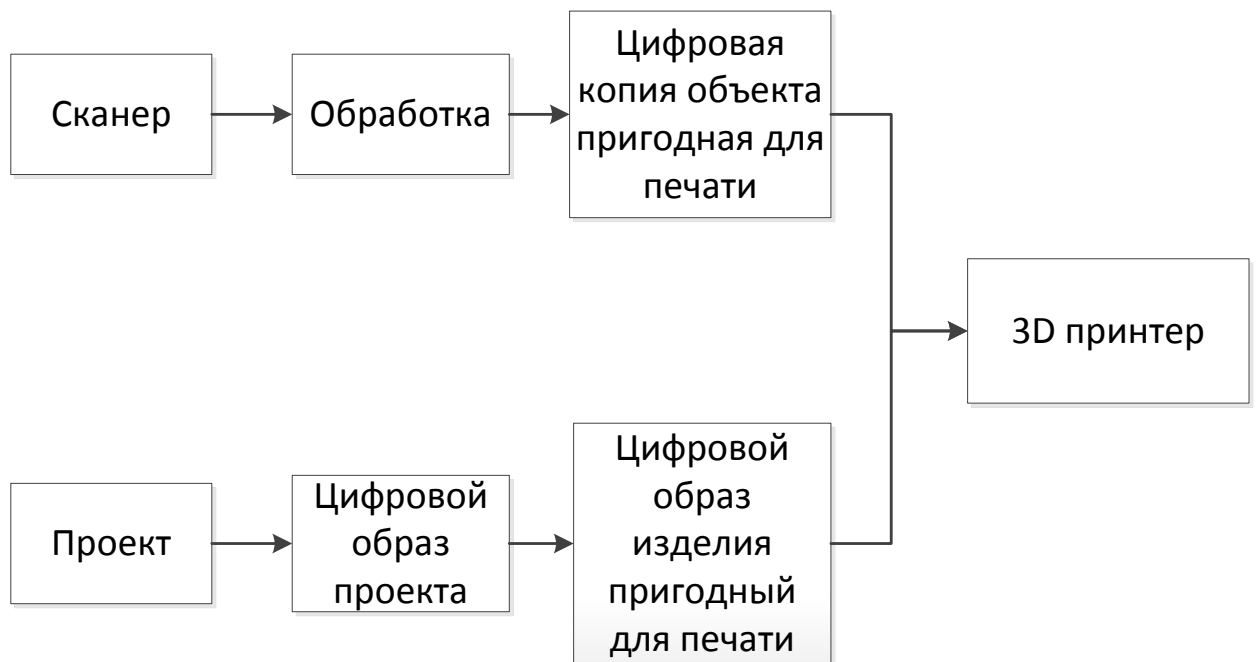


Рисунок 1.1 – Жизненный цикл 3D-объекта

Анализ работ [28–35] показал, что типичный жизненный цикл 3D-объекта в задачах репликации имеет следующие этапы (рисунок 1.1):

- 1) 3D-сканирование исходного материального объекта;
- 2) Восстановление (построение) цифрового образа с использованием программного обеспечения (ПО) в полуавтоматическом режиме;
- 3) Обработка цифрового образа в ручном режиме с использованием ПО (удаление артефактов, корректировка текстур, восстановление формы);
- 4) Предпечатная подготовка (масштабирование, верификация на предмет возможности печати, формирование выходного файла для печати в формате 3D-принтера) с использованием ПО в полуавтоматическом режиме;
- 5) 3D-печать материальной копии цифрового образа;
- 6) Сохранение и накопление цифровых образов в электронных библиотеках 3D-объектов.

Следует заметить, что на конечный результат оказывает влияние человеческий фактор, проявляющийся как на этапе сканирования, так и на неавтоматизированных и слабо автоматизированных этапах работы с цифровым образом.

В тех случаях, когда сканирование производится в «полевых условиях», что часто происходит, например, при работе с объектами искусства, требуется сохранять большие объемы 3D-сканов, полученных на 1 и 2 этапах, для того чтобы произвести этап 3 в «стационарных условиях» с целью уменьшения влияния человеческого фактора.

В этом случае актуальна задача компрессии (уменьшения объема битового представления) 3D-объекта перед сохранением или передачей данных между 2 и 3 этапами.

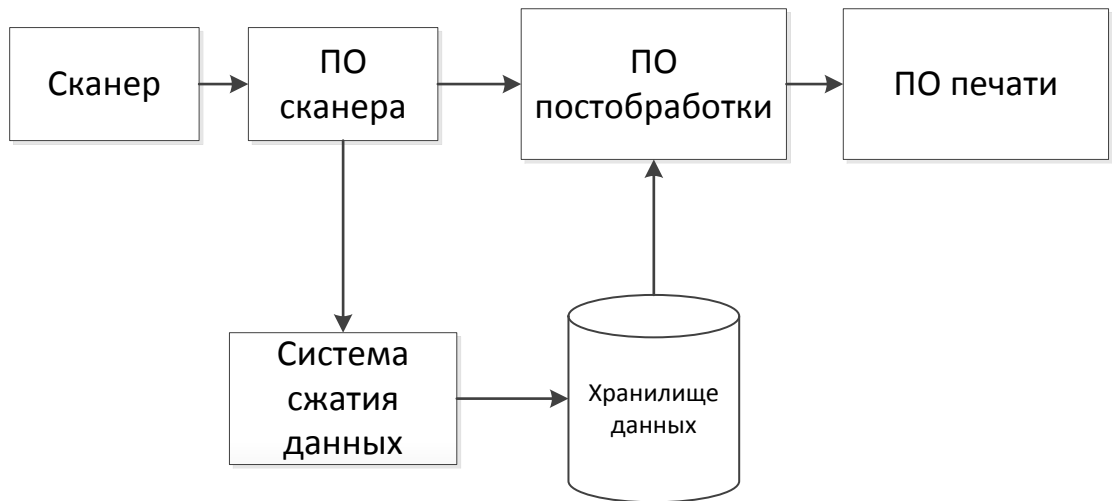


Рисунок 1.2 – Место разрабатываемой системы сжатия в процессе 3D-репликации

Возможен также случай, когда этапы 1–3 заменяются одним этапом искусственного синтеза 3D-объекта (рисунок 1.1). Объектом такого синтеза может быть как художественный или технический объект [4], так и результат визуализации физического или математического моделирования [36–37]. Искусственно-синтезированные объекты, как правило, отличаются повышенной детализацией, что ведет к увеличению количества точек-вершин, а, соответственно, длины битового описания такого объекта. В этом случае задача сжатия 3D-объектов также актуальна.

Кроме того, применительно к процессу «3D-сканирование → 3D-обработка → 3D-печать» (рисунок 1.2), человеческий фактор оказывает важное влияние на этапах сканирования и постобработки. Можно с уверенностью утверждать, что в будущем будут вестись работы, минимизирующие негативное влияние человеческого фактора. Предлагаемое исследование способствует минимизации указанного негативного влияния, отделяя этап сканирования, имеющий, как правило, жесткие временные ограничения (что повышает негативное влияние человеческого фактора) от этапа постобработки, выполняемого в более комфортных условиях с меньшими ограничениями по времени.

## 1.2 Методы получения пространственных трехмерных данных

Согласно определению [38]: «Пространственные данные — это цифровые данные о пространственных объектах, включающие сведения об их местоположении, форме и свойствах, представленные в координатно-временной системе...».

В данной работе будут рассматриваться данные без временной компоненты.

Все методы получения 3D-данных можно разделить на 3 группы (рисунок 1.3):

- 1) 3D-сканирование исходного материального объекта;
- 2) искусственный синтез 3D-объекта;
- 3) гибридные методы получения цифровых 3D-объектов.



Рисунок 1.3 – Источники получения 3D-данных для 3D-печати

### 1.2.1 Программные продукты для синтеза 3D-объектов

Программные пакеты, позволяющие создавать и моделировать объекты виртуальной реальности и создавать на основе этих моделей изображения, можно классифицировать по наличию истории построения объекта (таблица 1.1) и по элементам построения (таблица 1.2) [39].

Таблица 1.1 – Виды 3D-моделирования по наличию истории построения объекта

Виды	Описание	Пример программы
Параметрическое моделирование	Моделирование по набору заданных варьируемых параметров операций	CATIA
Непараметрическое моделирование	Моделирование без сохранения параметров построения (истории построения)	Rhinoceros
Комбинированное моделирование	Историю построения в любой момент можно удалить/отключить	Alias Studio Tools, Rhinoceros + Grasshopper

Таблица 1.2 – Виды 3D-моделирования по элементам построения

№	Виды	Элементы построения	Программы	Модель
1.	Полигональное моделирование	Полигон, кривая, (poly, nurbs, mesh)	Alias, 3ds Max, Maya, Rhino	Полигональная
2.	Каркасное моделирование	Точка и линия (line & point)	Alias ST, AutoCAD, CATIA, IsemSurf	Каркасная
3.	Поверхностное моделирование	Точка, линия, поверхность (surface)	Alias ST, AutoCAD, CATIA, IsemSurf, SolidWorks	Поверхностная

Таблица 1.2 (продолжение) – Виды 3D-моделирования по элементам построения

№	Виды	Элементы построения	Программы	Модель
1.	Полигональное моделирование	Полигон, кривая, (poly, nurbs, mesh)	Alias, 3ds Max, Maya, Rhino	Полигональная
2.	Каркасное моделирование	Точка и линия (line & point)	Alias ST, AutoCAD, CATIA, IsemSurf	Каркасная
3.	Поверхностное моделирование	Точка, линия, поверхность (surface)	Alias ST, AutoCAD, CATIA, IsemSurf, SolidWorks	Поверхностная
4.	Твердотельное моделирование	solid (твердое тело)	Alias ST, AutoCAD, CATIA, IsemSurf, SolidWorks	Твердотельная
5.	Конечно-элементное моделирование	узел, конечный элемент, сетка (point, lines, mesh)	Ansys, Rhino + Kangaroo	Конечно-элементная
6.	Генеративное моделирование	компоненты, связи между компонентами	Rhino Grasshopper +	Генеративная

### 1.2.2 3D-сканирование пространственных форм

Под 3D-сканированием (трёхмерным сканированием) понимается процесс преобразования физической формы материального объекта в цифровой вид (цифровой скан), то есть формирование компьютерной 3D-модели.

Основные области применения [40–44]:

- Инженерный анализ
- Контроль качества и инспекция
- Разработка упаковки
- Цифровое архивирование
- Промышленный дизайн
- Развлечения и игры

- Рынок аксессуаров
- Репродуцирование и изготовление на заказ
- Медицина и ортопедия

Трёхмерное сканирование используется в задачах реверс-инжиниринга, при проектировании оснастки, приспособлений и запасных частей, особенно в условиях отсутствия оригинальной документации на изделие, а также при оцифровке сложных пространственных форм. Часто 3D-сканирование используется в целях сопоставления реплицированного объекта с его цифровым образом, что играет особую роль в задачах медицинского протезирования [16, 45].

Этот быстрый, точный и экономичный способ сбора физических данных позволяет инженерам создавать более качественные решения, уменьшить ошибки прототипирования, усовершенствовать производственные процессы и, тем самым, выйти на новый уровень качества производства [46].

Методы 3D-сканирования можно классифицировать по принципу работы сканеров:

1. Контактный метод;
2. Бесконтактный (активный, пассивный) метод.

#### *Контактное 3D-сканирование*

Основным принципом данного метода является обводка сканируемого объекта контактной головкой, оснащенной активным пьезосенсором. Головка движется по поверхности объекта и в компьютер заносятся координаты о его положении (рисунок 1.4). На базе этих координат строится трехмерная модель сканируемого объекта с точностью 0,05мм. Этот метод позволяет значительно сократить время, необходимое для разработки новых изделий и их запуска в производство.



Рисунок 1.4 – Контактный 3D-сканер

*Бесконтактные 3D-сканеры* являются более сложными приборами, во многом зависящими от примененных алгоритмов определения пространственных координат и восстановления форм объектов.

Как правило, современные сканеры имеют коммерчески закрытое встроенное программное обеспечение и представляют собой систему сканер – программное обеспечение сканера «3D-Сканер — ПОС».

Сканирование объекта ведется съёмкой объекта с разных ракурсов. Две камеры позволяют получить восстановленную поверхность по относительному смещению фрагментов изображения объекта, которые видны обеим камерам (пассивный способ, рисунок 1.5).

Современные 3D-сканеры являются комбинированными приборами. Во многих из них применяется совмещенная двойная система получения координат 3D-объекта. В дополнение к лазерным датчикам (являющимся развитием идеи механического «щупа» в контактных устройствах) используется цифровая камера, обеспечивающая получение текстурной информации об объекте [47].



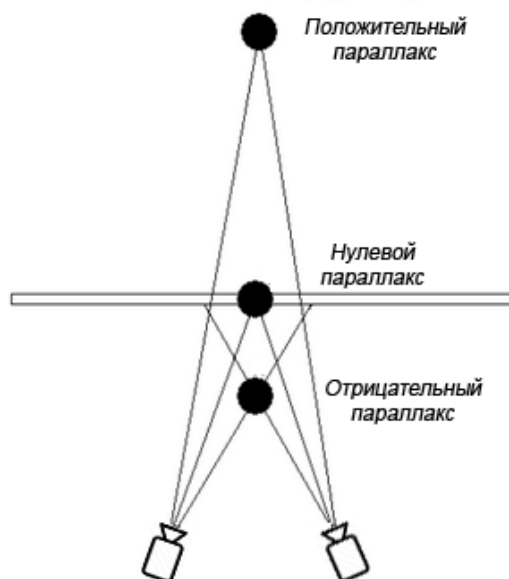


Рисунок 1.5 – Бесконтактный пассивный 3D-сканер

Для ускорения процесса возможно использование структурированной подсветки, когда источник излучения (проектор) наносит на объект полосу или сетку (рисунок 1.6). Камера, смещенная относительно проектора, воспринимает отражение этой сетки и по обнаруженным искажениям вычисляет расстояние до каждой точки в поле зрения. За счет этого достигается и высокая скорость (можно анализировать всё поле зрения сразу), и лучшая точность. Именно такой способ используется в 3D-сканерах Artec [48].

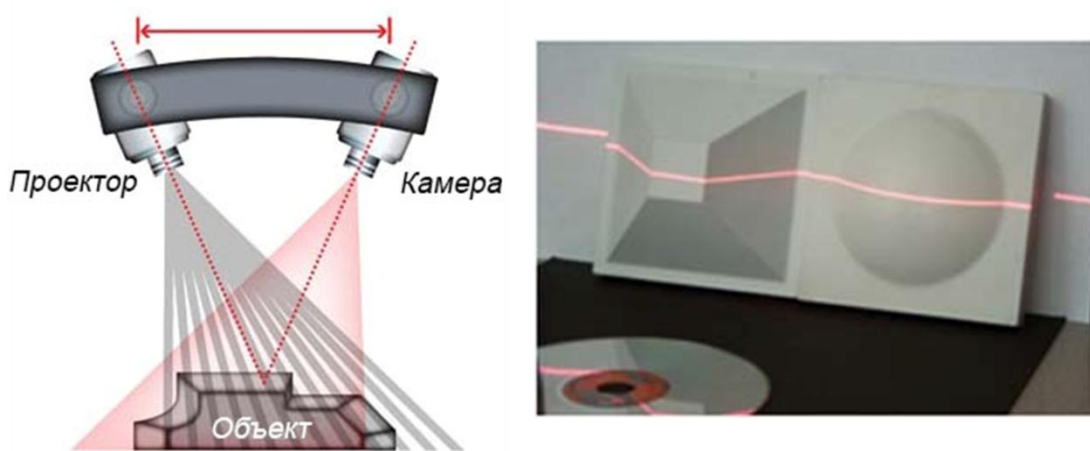


Рисунок 1.6 – технология активного 3D-сканирования

Вместо одиночных лазерных сенсоров могут применяться более сложные системы, такие как системы 3D-сканирования на базе ультразвуковых сенсоров, преимуществом которых является наличие режима сканирования объектов с внутренней структурой. Имеются также опытные образцы магнитных сканеров, использующих изменение пространственного магнитного поля объекта для локализации его пространственных координат.

Многообразие устройств сканирования создает проблему разработки специализированного программного обеспечения для работы с такими устройствами и интерпретации получаемых с них данных, а также для обеспечения унифицированного человеко-машинного взаимодействия [49].

### **1.2.3 Компьютерные томографы**

Разновидностью технологии получения 3D-модели с внутренней структурой является компьютерная томография [50].

Компьютерная томография (КТ) — метод лучевой диагностики, позволяющий получить послойное изображение любой области человеческого тела толщиной среза от 0,5мм до 10мм.

Принцип работы рентгеновского компьютерного томографа основывается на круговом просвечивании исследуемой области тонким пучком рентгеновских лучей перпендикулярным оси тела, регистрации ослабленного излучения с противоположной стороны системой детекторов и преобразование его в электрические сигналы: проходя через тело человека, рентгеновские лучи поглощаются различными тканями в разной степени. После этого рентгеновские лучи попадают на специальную чувствительную матрицу, данные с которой считываются компьютером.

Обычно используется порядка 512–1024 детекторов. Система трубка–детекторы вращается на  $360^{\circ}$  вокруг исследуемого объекта. При этом происходит сканирование 360–1200 проекций (ракурсов). Время полного поворота системы 2–10 сек. Полученные сигналы поступают на коммутатор, с которого подаются на

АЦП, и передаются в специализированную компьютерную систему. Принятые сигналы проходят специальную обработку, в результате которой получаются изображения, которые и подаются на монитор КТ и сохраняются на диске компьютерной системы [51].

### *Построение 3х мерного изображения*

При наличии достаточно большого количества срезов (рисунок 1.7) и зная шаг, с которым они производились, можно восстановить трехмерное изображение исследуемого объекта.

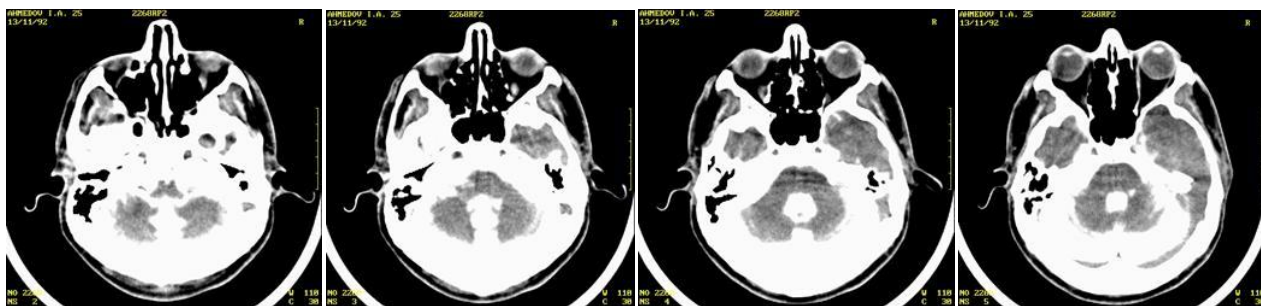


Рисунок 1.7 – Пакет срезов для восстановления трехмерного изображения

Общая схема построения трехмерного изображения представлена на рисунке 1.8. В связи с тем, что расстояние между срезами значительно больше, чем расстояние между точками (пикселями) на самом срезе, то используется алгоритм построения дополнительных, промежуточных срезов. Дополнительные срезы получаются путем нахождения среднеарифметических значений точек предыдущего и последующего срезов. В результате получается трехмерный массив, с помощью которого восстанавливается трехмерное изображение. Трехмерный массив позволяет производить дополнительные операции (срезы, вырезы, перемещение в пространстве) для отображения внутренней структуры исследуемого объекта.

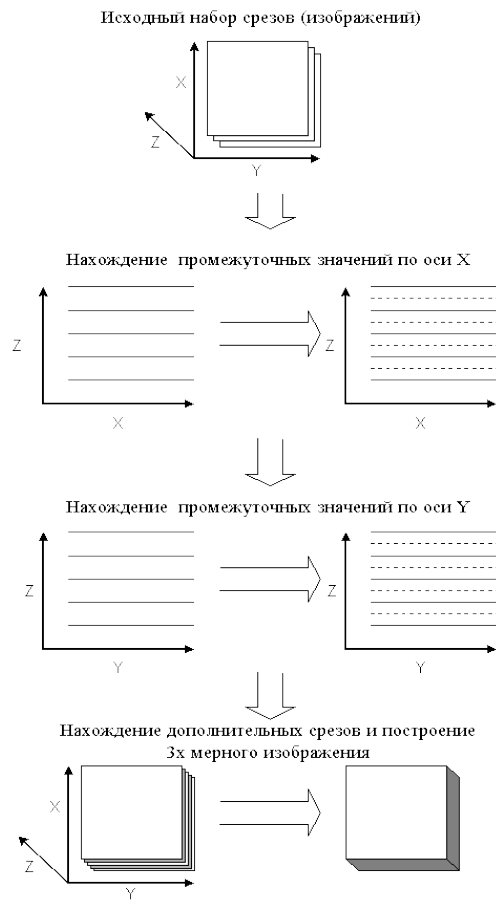


Рисунок 1.8 – Структурная схема построения трехмерного изображения [51]

### 1.3 Анализ существующих моделей и методов представления трехмерных объектов

Способ представления 3D-объекта в цифровом виде определяется структурой данных, то есть форматом ее битовой последовательности. Каждая из моделей представления ориентирована на свою специализированную область применения, а исходные точки поверхности распределяются различными способами: по узлам регулярной сетки, по структурным линиям рельефа или хаотично [3].

Проведем сравнительный анализ применимости различных представлений 3D-объектов для решения задачи компрессии цифрового скана (рисунок 1.9).

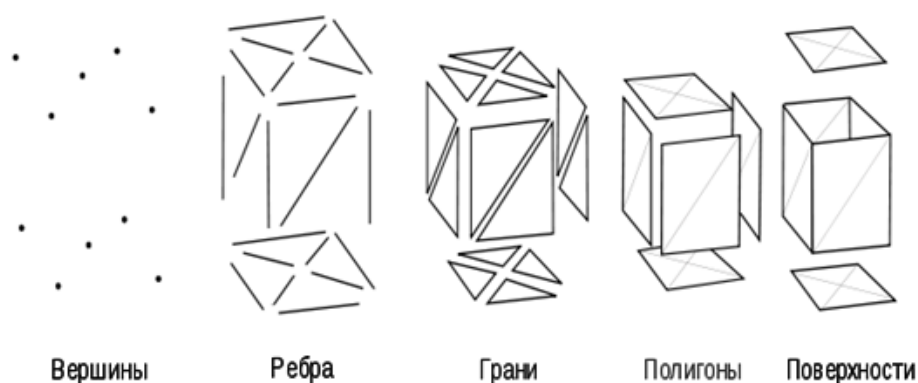


Рисунок 1.9 – Элементы моделирования 3D-объекта

Вершинное представление описывает объект как множество соединенных между собой вершин, является наиболее простым, в котором информация о гранях и ребрах не выражена явно. Поэтому для восстановления исходного объекта требуется обход всех вершин и составление списка граней.

**Список вершин**

v0	0,0,0	v1 v5 v4 v3 v9
v1	1,0,0	v2 v6 v5 v0 v9
v2	1,1,0	v3 v7 v6 v1 v9
v3	0,1,0	v2 v6 v7 v4 v9
v4	0,0,1	v5 v0 v3 v7 v8
v5	1,0,1	v6 v1 v0 v4 v8
v6	1,1,1	v7 v2 v1 v5 v8
v7	0,1,1	v4 v3 v2 v6 v8
v8	.5,5,0	v5 v6 v7 v8
v9	.5,5,1	v0 v1 v2 v3

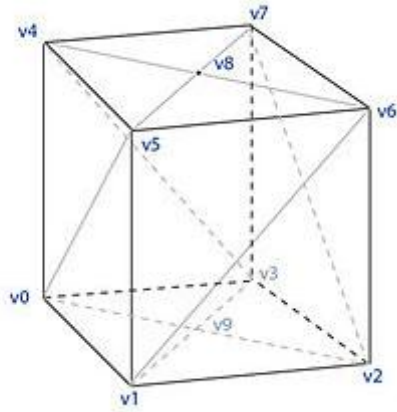


Рисунок 1.10 – Вершинное представление

Рисунок 1.10 показывает пример представления параллелепипеда с использованием вершинного представления [52].

Модель представления с использованием списка граней позволяет представить объект в виде множества граней и множества вершин. Это самое широко используемое представление.

Представление в виде списка граней более адекватно подходит для моделирования, чем вершинное представление, так как он позволяет осуществлять прямой поиск вершин и граней, прилегающих к ним. На рисунке 1.11 приведен пример представления параллелепипеда с использованием списка граней.

Список граней		Список вершин	
f0	v0 v4 v5	v0	0,0,0 f0 f1 f12 f15 f7
f1	v0 v5 v1	v1	1,0,0 f2 f3 f13 f12 f1
f2	v1 v5 v6	v2	1,1,0 f4 f5 f14 f13 f3
f3	v1 v6 v2	v3	0,1,0 f6 f7 f15 f14 f5
f4	v2 v6 v7	v4	0,0,1 f6 f7 f0 f8 f11
f5	v2 v7 v3	v5	1,0,1 f0 f1 f2 f9 f8
f6	v3 v7 v4	v6	1,1,1 f2 f3 f4 f10 f9
f7	v3 v4 v0	v7	0,1,1 f4 f5 f6 f11 f10
f8	v8 v5 v4	v8	.5,5,0 f8 f9 f10 f11
f9	v8 v6 v5	v9	.5,5,1 f12 f13 f14 f15
f10	v8 v7 v6		
f11	v8 v4 v7		
f12	v9 v5 v4		
f13	v9 v6 v5		
f14	v9 v7 v6		
f15	v9 v4 v7		

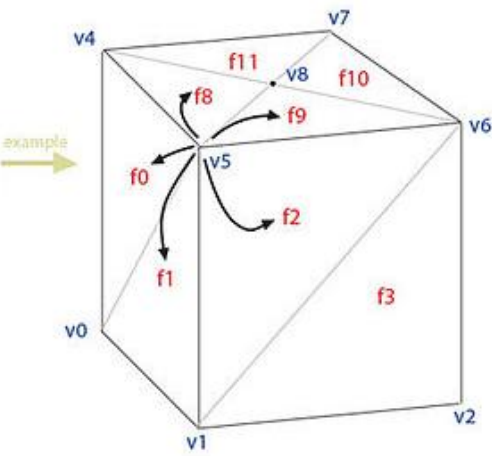


Рисунок 1.11 – Представление в виде списка граней

В отличие от вершинного представления, в списке граней и грани и вершины представлены в явном виде, так что нахождение соседних граней и вершин занимает одинаковое время. Однако, ребра в таком представлении не заданы явно, поэтому для поиска всех граней, прилегающих к заданной грани, требуется осуществлять дополнительные операции поиска.

В задачах, где требуется быстрый доступ ко всем элементам представления трехмерного объекта (вершины, грани, ребра) используется представление, показанное на рисунке 1.12 [53].

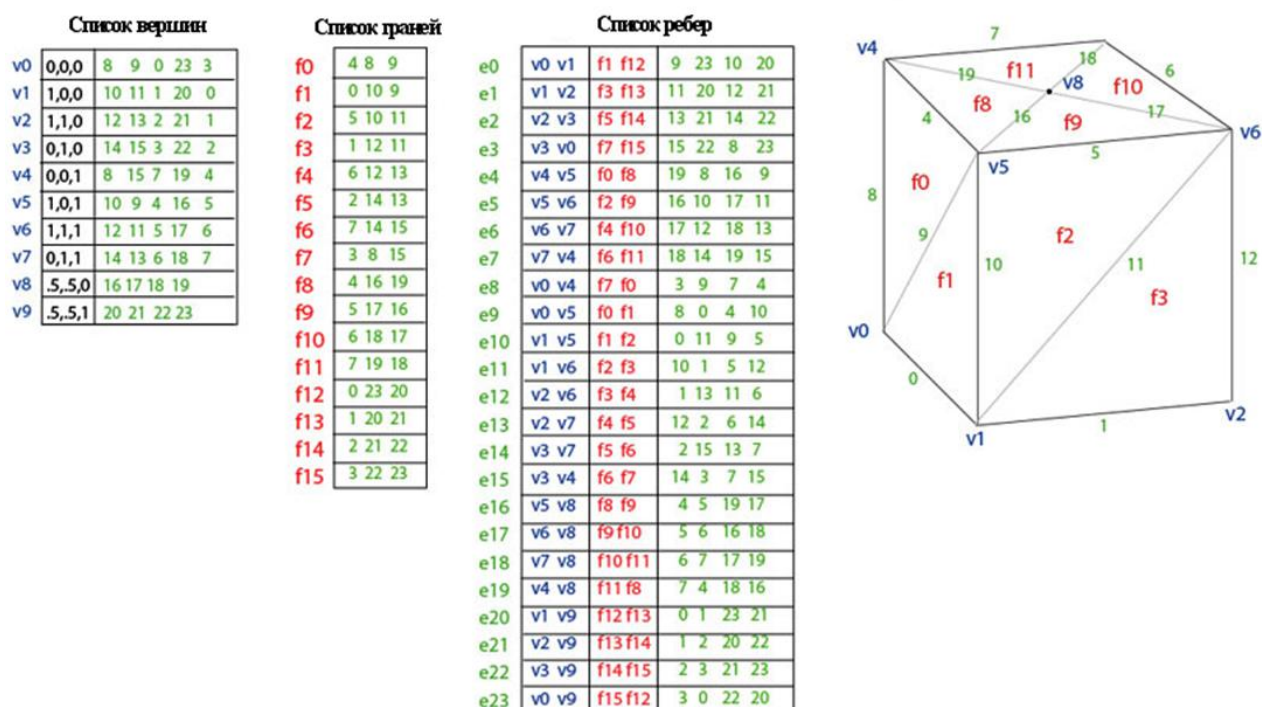


Рисунок 1.12 – Представление в виде списка граней и ребер («крылатое» представление)

### 1.3.1 Модели на основе полигональных сеток

Полигональные сетки (представление в виде списка граней) являются наиболее распространенным представлением, для которого доступно большое количество программных продуктов для редактирования и визуализации объектов.

Важной особенностью моделей на основе полигональных сеток является связность, которая упрощает генерацию искусственных поверхностей.

Однако, данные, полученные с 3D-сканеров представляют собой набор близко расположенных точек и не содержат сведений о принадлежности точек конкретным поверхностям. Такие ограничения следуют из физического принципа работы сканера, обладающего дискретным шагом конечного разрешения.

Соответственно применение полигональных моделей требует в ряде задач предварительного этапа определения связности отдельных точек перед дальнейшей обработкой. В связи с этим полигональные модели не могут использоваться для непосредственного хранения цифровых сканов, так как в этом случае требуется решение неоднозначной задачи восстановления поверхностей [54–55]. Более того, работа с сетками больших размеров затруднена из-за увеличивающегося объема данных для поддержания связности, что, в свою очередь, требует применения методов упрощения.

Существующие методы упрощения сеток в основном используют комбинации описанных ниже подходов [56–58].

Алгоритмы сэмплирования (от англ. *sampling* – упрощение) применяются для упрощения геометрии модели, при этом используется подмножество исходных точек или их округление до ближайшей точки на выбранной трехмерной сетке. Такие алгоритмы наиболее эффективны для работы с гладкими поверхностями.

При использовании адаптивного разбиения (англ. *adaptive subdivision*) для аппроксимации исходной модели сначала определяется базовая сетка, затем производится ее рекурсивное уточнение в областях, требующих повышенной точности. Применение такого подхода оправдано в случаях простой базовой модели (прямоугольник для фрагмента ландшафта). Для произвольного объекта может потребоваться создание базовой модели, отражающей важные свойства исходного объекта, что является нетривиальной задачей.



Алгоритмы прореживания (англ, decimation) работают по принципу итеративного удаления граней или вершин из полигональной сетки, заново производя триангуляцию на каждом шаге.

Метод слияния вершин выполняет объединение нескольких близлежащих вершин в одну, которая может быть также объединена с другими вершинами на следующих шагах упрощения модели. При этом сокращается общее количество треугольников модели.

При слиянии ребер (edge collapse) производится объединение вершин, разделяющих общую грань. Это сохраняет локальную топологию.

Эти методы имеют высокую сложность вычислений, а их использование не всегда оправдано, поскольку сложность является следствием необходимости поддерживать связность модели.

#### **Преимущества и недостатки:**

- наличие аппаратной поддержки;
- требуется искусственная поддержка связности;
- снижение эффективности представление при увеличении количества элементов в модели без упрощения и соответствующего ухудшения качества представления.

### **1.3.2 Модели на основе вокселей**

Воксельные модели представляются в виде регулярных трехмерных массивов, элементам которых сопоставляются характеристики цвета и прозрачности [59–60].

В связи с тем, что в воксельном представлении хранится описание всей области пространства, содержащей в себе 3D-объект, то объемы данных в воксельных представлениях значительны даже для небольших моделей. Для уменьшения объема представления наиболее часто используются древовидные иерархии. Существует вариант иерархического задания многомасштабного представления октодеревом [59], в котором элементы (node) содержат значения

усредненного цвета и усредненной прозрачности для всех элементов, стоящих ниже по иерархии. Кроме того, каждый элемент также содержит разность исходного значения для конкретного элемента и среднего значения элемента, стоящего выше по иерархии.

Воксельные представления ориентированы на визуализацию технических объектов, виртуальную реальность, хранение медицинских данных, а также в других областях.

#### **Преимущества и недостатки:**

- простая регулярная структура, наличие аппаратной поддержки;
- требуется хранить весь объем, содержащий в себе 3D-объект;
- требуется применение специальных многомасштабных структур для работы со сложными объектами.

### **1.3.3 Модели, основанные на картах глубины**

Данные модели (англ, Image-Based Modeling and Rendering, IBMR) [61] не используют промежуточные структуры данных, и представляют 3D-объект с помощью двумерных изображений, дополненных компонентом, определяющим глубину для каждой точки. Это значит, что результирующее для заданных параметров виртуальной камеры и точки наблюдения изображение возможно составить по конечному набору исходных изображений сцены.

#### *Использование изображений с картами глубины*

Базовой структурой данных, используемой в IBMR является набор изображений с картами глубины. Пара «изображение – карта глубины» определяется как RGB-изображение, которому сопоставлено изображение grayscale того же размера, значение каждой точки которого определяется расстоянием от камеры до поверхности объекта.

Заметим, что данные с некоторых современных бесконтактных сканеров можно получать непосредственно в виде карт глубины, более того, в некоторых

случаях сразу доступна и текстурная информация об объекте. Следовательно, такое представление может быть использовано в качестве необработанного внутреннего представления в программном обеспечении системы 3D-сканирования. При этом пара «изображение – карта глубины» однозначно определяет описание фрагмента поверхности, а точность приближения зависит от разрешения 3D-сканера и выбранного положения его камеры.

Для восстановления исходного объекта необходимо использовать набор карт глубины, описывающих объект с разных ракурсов, полученных в процессе перемещения 3D-сканера относительно сканируемого объекта, так как каждая карта глубины хранит информацию о видимой из только из конкретного положения камеры части объекта.

*Использование многослойных изображений с глубиной [62].*

Многослойные изображения с глубиной содержат все пересечения воображаемого луча, «просвечивающего» 3D-объект, для каждого пикселя карты цвета, что обуславливает возможность использования только одного такого изображения для хранения исчерпывающей информации, описывающей 3D-объект.

Под LDI (Layered Depth Images) понимается трехмерная структура данных, представляющая собой массив, каждый элемент которого представлен списком точек, каждая из которых содержит данные о глубине (расстояние от опорной плоскости) и атрибутах (например, цвет или прозрачность).

При использовании LDI-структур имеются некоторые ограничения, обусловленные наличием базовой плоскости, на которую производится проекция проекцией всех точек. Еще одно ограничение связано с необходимостью использования для создания LDI-структуры сторонних программных средств.

### 1.3.4 Модели на основе точечных представлений (облака точек)

В точечных представлениях объекты моделируются как набор трехмерных координат точек принадлежащих поверхностям объекта (рисунок 1.13).



Рисунок 1.13 – Представление трехмерного объекта в виде облака точек

Основной проблемой таких представлений является отсутствие данных о связности точек и их принадлежности к непрерывным поверхностям сканируемого 3D-объекта. В связи с неоднозначностью решения задачи восстановления непрерывной поверхности возможно появление разрывов.

Возможные трудности с визуализацией можно решать, например, применением метода сплэттинга (англ, splatting) — расчета формы проекции точки на плоскость экрана. Такой подход используется, в частности, в [63].

Другим недостатком точечных представлений является необходимость хранения большого объема данных, что также усиливает проблемы визуализации. Использование неструктурированного набора точек при увеличении объема обуславливает необходимость применения методов упрощения и сжатия, имеющих высокую вычислительную сложность или требующих дополнительно сформированных данных.

#### **Преимущества и недостатки:**

- легкость получения и моделирования;

- простота визуализации моделей небольшого объема;
- неоднозначность результата при восстановлении связности объекта;
- определенные сложности при реализации многомасштабности;
- большие объемы данных.

#### 1.4 Форматы представления трехмерных данных

Наиболее распространенными форматами представления сканированных данных являются: ply, stl, xyz, obj, ptx, e57 и ряд других. В них в текстовой или бинарной форме содержится список координат точек, принадлежащих трехмерному объекту (рисунок 1.14) на примере формата OBJ.

```
#####
#
# OBJ File Generated by Meshlab
#
#####
# Object Cube.obj
#
# Vertices: 64848
# Faces: 21616
#
#####
v 744.931458 15.247781 20.000000
v 745.931458 15.247781 20.000000
v 744.931458 15.247781 19.000000
v 745.931458 15.247781 20.000000
v 745.931458 15.247781 19.000000
v 744.931458 15.247781 19.000000
v 762.942871 15.247781 20.000000
v 763.931458 15.247781 20.000000
v 762.942871 15.247781 19.000000
v 763.931458 15.247781 19.000000
```

Рисунок 1.14 – Пример содержимого форматов цифровых сканов на примере формата OBJ

Для задач репликации наиболее востребованным оказался формат STL. В этом формате используется способ аппроксимации любых поверхностей системой стыкуемых друг с другом плоских треугольников. Таким образом, поверхность объекта аппроксимируется поверхностной сеткой с треугольными ячейками. При этом формируется TIN-поверхность (Triangulated Irregular Network) –

нерегулярная триангуляционная сеть неперекрывающихся треугольников, соответствующая так называемой триангуляции Делоне [64].

Вершинами треугольников являются опорные точки, соответствующие исходной поверхности. Треугольники формируются с учетом направления обхода вершин по часовой стрелке.

Например, рельеф может быть представлен в виде полиномиальной поверхности, коэффициенты которой задаются значениями в вершинах граней треугольников. При правильном построении эта сеть не имеет разрывов и наложений, а представление 3D-объекта является списком параметров треугольных ячеек, описывающих эту поверхность.

Применение STL-формата в технологии быстрого прототипирования обусловлено простотой алгоритма построения горизонтальных сечений и соответствующей вычислительной процедуры, которая сводится к определению треугольников, пересекаемых горизонтальной плоскостью сечения (пара вершин каждого такого треугольника должна лежать по разные стороны секущей плоскости), после этого определяется линия пересечения с плоскостью, т.е. участок границы контура, из которых и формируется граница полного горизонтального сечения [65].

### **Выводы по главе 1**

В главе приведен анализ процессов получения и представления пространственных данных с целью построения цифровых копий реальных объектов, проанализированы особенности существующих моделей представления пространственных данных, основные характеристики которых обобщены в таблице 1.3.

Таблица 1.3 – Особенности существующих моделей представления пространственных данных.

<b>Модель представления</b>	<b>Преимущества модели</b>	<b>Недостатки модели</b>
Модели на основе полигональных сеток	- наличие аппаратной поддержки;	- требуется искусственная поддержка связности; - снижение эффективности представление при увеличении количества элементов в модели без упрощения и соответствующего ухудшения качества представления.
Модели на основе вокселей	- простая регулярная структура, - наличие аппаратной поддержки;	- требуется хранить весь объем, содержащий в себе 3D-объект; - требуется применение специальных многомасштабных структур для работы со сложными объектами.
Модели, основанные на картах глубины	- простота представления рельефа поверхности;	- не могут быть напрямую получены с устройств 3D-сканирования, необходимо использование дополнительных программных средств.
Модели на основе точечных представлений (облака точек)	- легкость получения и моделирования; - простота визуализации моделей небольшого объема;	- неоднозначность результата при восстановлении связности объекта; - определенные сложности при реализации многомасштабности; - большие объемы данных.

Каждая из моделей представления ориентирована на свою специализированную область применения, что не позволяет использовать их в качестве универсального представления. Например, модели на основе полигональных сеток, были специально разработаны для работы в виртуальном пространстве и ориентированы на работу с CAD-подобными средствами проектирования.

С точки зрения простоты получения данных с бесконтактных сканеров, поддержки масштабируемости и аппаратной поддержки визуализации,

наилучшими характеристиками обладают точечные и основанные на картах глубины представления. Описание в виде облаков точек являются удобным способом представления сканированных данных без дополнительной обработки.

Следует отметить особенность получения точечных представлений: программному обеспечению сканера требуется производить выделение опорных точек на объекте и их сопоставления для серии кадров, полученных с камер 3D-сканера. В случае возникновения ошибок в этом процессе возможно появление артефактов сканирования, выражающихся в появлении точек, не соответствующих реальной форме объекта. Разработка методов сопоставления опорных точек и повышение их эффективности является отдельной большой областью исследований [66, 67].

В случае использования портативного 3D-сканера в «полевых условиях» без возможности применения хранилищ данных и серверного оборудования проблема увеличения объема отсканированных данных становится определяющей. Особенно важно не допускать потерь данных или их ухудшения в результате применения сжатия с потерями, чтобы иметь возможность корректной постобработки 3D-объекта без вносимых методом сжатия артефактов.

Таким образом, задача диссертационного исследования состоит в следующем: разработка модельно-алгоритмического обеспечения для компактного описания и гарантированного восстановления исходного образа пространственного объекта произвольной формы.

Для трансляции содержательной постановки в программную реализацию предполагается разработка функциональной модели в виде алгоритма сжатия облаков точек.



## Глава 2. Методы компрессии трехмерных данных

Основным недостатком большинства рассмотренных в предыдущем разделе моделей, является большой объем их битового представления при программной реализации. Использование неструктурированного набора точек при увеличении объема обуславливает необходимость применения методов упрощения и сжатия, использовать которые в реальном времени не представляется возможным без введения дополнительных структур данных, ресурсов памяти и процессорного времени.

В зависимости от способа получения 3D-данных, занимаемый ими объем битового представления будет различным.

Детальное изучение существующих подходов и реализаций компрессии пространственных данных с потерями и без потерь позволило выделить основные идеи, лежащие в их основе.

В настоящее время, известные методы сжатия данных, полученных в результате 3D-сканирования, делятся на 2 группы:

- методы, требующие для работы предварительного перевода облака точек в набор полигонов (восстановление поверхностей объекта);
- методы, ориентированные на работу непосредственно с облаками точек без предварительной интерполяции поверхности набором полигонов.

Однако в связи с тем, что при сканировании восстановление полигонов производится с помощью прикладного программного обеспечения и является приближенным, то более корректными и перспективными для задач сжатия без потерь являются методы, работающие непосредственно с облаками точек.

Анализ существующих методов сжатия, работающих непосредственно с облаками точек, показал, что все они используют различные алгоритмы обхода точек пространства принадлежащих сканируемому объекту и дальнейшего представления их в виде древовидной структуры (которыми и отличаются друг от друга) и дальнейшего сжатия методами группового кодирования, тем самым используя элементы методов восстановления поверхностей. Главным недостатком

является падение эффективности сжатия для объектов имеющих сложный рельеф поверхности и внутреннюю структуру.

В таблице 2.1 представлены сравнительные объемы исходного (несжатого) битового представления различных 3D-объектов. Из нее видно, что для эффективного хранения 3D-объектов необходимы форматы представления этих объектов в сжатом виде, ориентированные на конкретный класс представления [16].

Таблица 2.1 – Ориентировочное сравнение объемов битового представления различных 3D-объектов

Вид объекта	Способ получения	Размер, байт
Барельеф	3D-сканирование	16 757 334
Подшипник	3D-сканирование	362 982 784
Анатомическая модель «мозг»	3D-сканирование	4 567 684
Декоративная фигурка	3D-сканирование	5 950 884
Барельеф2	3D-сканирование	10 866 084
Вертолет (модель)	синтезирован в 3D Studio Max	5 505 893
Стереопара в HD-качестве (фрагмент фильма)	Стереокамера	12 441 600

Кроме того, в задаче представления и компрессии 3D-данных важно учитывать характерные особенности 3D-сканирования, возникающие на этапах постобработки (например, склейки слоев, удаление пустых областей и т.д.). Эти особенности (артефакты) связаны с относительным взаимным смещением координатных сеток, по которым происходит привязка вокселей отсканированных поверхностей (рисунок 2.1). Более подробно исследование особенностей

технической системы 3D-сканер — программное обеспечение сканера и причины возникновения артефактов будут рассмотрены в главе 4.

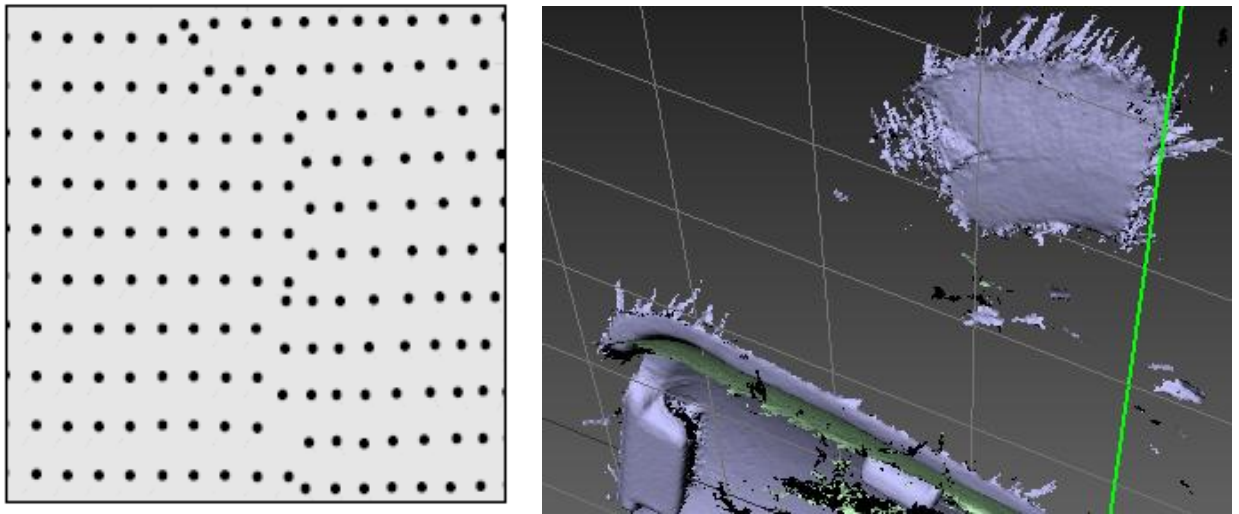


Рисунок 2.1 – Артефакты при автоматической склейке отсканированных поверхностей

## 2.1 Анализ методов компрессии без потерь

Методы сжатия без потерь основаны на описании точек в виде регулярных структур, например в представлении точек на поверхности в виде остовного дерева, приведенного на рисунке 2.2 (ациклического связного подграфа для связного неориентированного графа, в который входят все его вершины) [68–69].

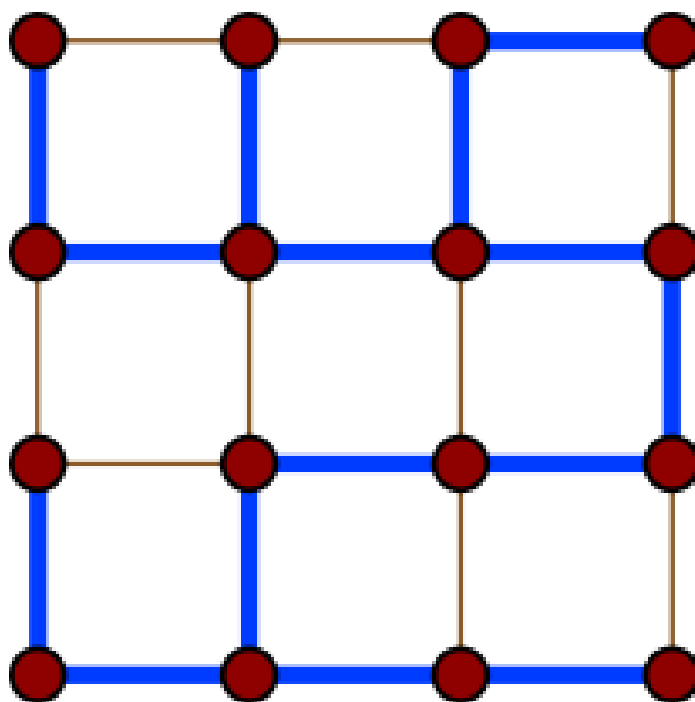


Рисунок 2.2 – Пример остовного дерева

Такое остовное дерево может кодироваться при помощи последовательностей символов B, L, R, F, T (base, left, right, forward, terminal), способ построения которых основан на алгоритме с использованием линейных предсказателей. Рекурсивное описание точек, принадлежащих поверхности, путем включения их в описание уже построенных деревьев позволяет уменьшать битовую длину представления исходного объекта (рисунок 2.3). Теоретически, использование метода позволяет сократить длину битового представления до 2–3 бит на точку, что эквивалентно компрессии в 5–17 раз, в зависимости от количества точек в облаке. При этом качество компрессии ухудшается с увеличением количества точек [70].

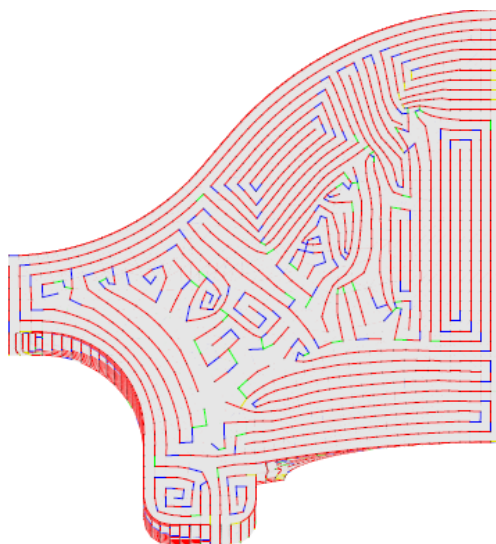


Рисунок 2.3 – Пример описания поверхности при помощи остовного дерева

Существует группа методов сжатия 3D-данных, в основе которых лежит построение полигональных сеток, что обуславливает сжатие 3D-данных с потерями, так как такие преобразования являются необратимыми и приводят к упрощению 3D-модели. Такое сжатие 3D-данных с потерями не подходит для целей диссертационного исследования, так как при 3D-сканировании для качественной дальнейшей обработки 3D-объектов в стационарных условиях чаще всего требуется сохранять все элементы облака точек (рисунок 2.4).

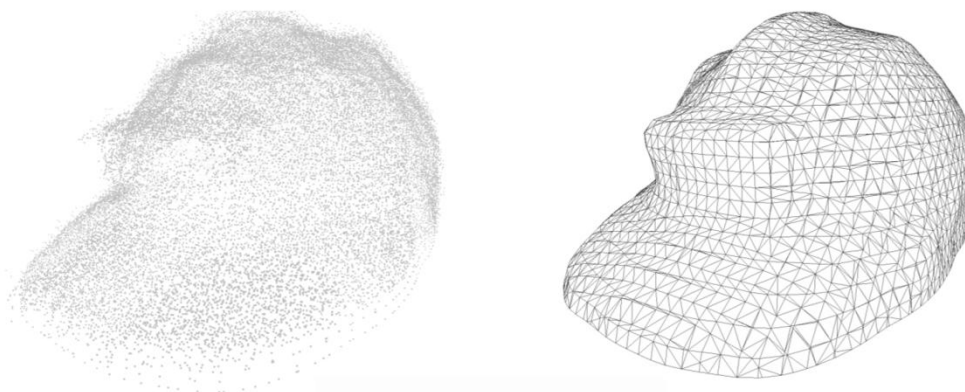


Рисунок 2.4 – Облако точек (слева) преобразованное в полигональную сетку (справа)

Существуют методы сжатия 3D-данных, обеспечивающие сжатие без потерь, которые основаны на представлении точек на поверхности в виде

остовного дерева (ациклического связного подграфа для связного неориентированного графа, в который входят все его вершины). Способ построения остовного дерева основан на алгоритме с применением линейных предсказателей. Подобные методы показывают очень хорошие результаты на обработанных, сглаженных облаках точек без артефактов и на не детализированных объектах без сложной внутренней структуры с малым количеством артефактов на поверхности.

Еще одним способом сжатия являются иерархические представления. Известен способ построения иерархий с использованием древовидных структур, таких как восьмеричные деревья или kd-деревья, трехмерных деревьев на базе иерархий ограничивающих сфер и кубов. Свойства иерархий зависят от базового представления, при этом сохраняя общие принципы построения [71]. Например, для воксельных иерархических моделей используют октодеревья, которые подходят для представления 3D-объектов, заданных на регулярной сетке (рисунок 2.5).

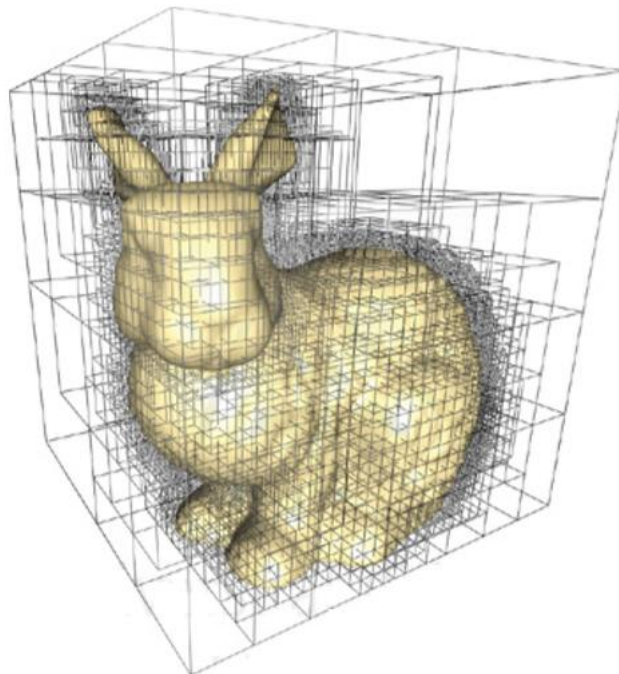


Рисунок 2.5 – 3D-модель, представленная в виде октодерева

Терминальные узлы октодерева представляют собой модель с исходной (максимальной) детализацией. При подъеме по иерархии вверх значения в узлах усредняются с соответствующим снижением детализации, вплоть до вырождения в единственное значение в корневом узле дерева. Восстановление объекта таком случае возможно при рекурсивном обходе октодерева от корневого узла к терминальным.

Так же иерархические структуры могут применяться к полигональным сеткам для динамического контроля уровня детализации и сложности модели.

## **2.2 Анализ методов компрессии с потерями**

В качестве примера компрессии трехмерных данных с потерями можно привести метод, описанный в [72], ориентированный на сжатие пространства точек, полученных как серию срезов (двухмерных измерений, сделанных под различными углами), однозначно определяет трехмерную структуру объекта. Такой тип данных может быть сформирован различным оборудованием, в том числе рентгеновскими системами и электронными микроскопами.

Необходимость обеспечить долгосрочное хранение (архивирование) томографических данных, передачу данных и удаленный доступ к ним требует реализации компрессии (сжатия) томографических данных.

Большинство известных методов компрессии объемных томографических данных основаны на реконструкции объемного представления как серии двумерных массивов, причем обработка каждого двумерного массива производится независимо, что ведет к потере структуры связности локальной топологии данных. Так разработанные в ИППИ РАН [73] методы основаны на дискретном вейвлет-преобразовании, применяемом к каждому срезу и дают компрессию 1,9—3,0 раз.

В ряде известных алгоритмов сжатие производится за счет устранения корреляционной зависимости между последовательными двумерными массивами (слоями). В работе [74] показано, что для медицинских исследований требуется

сохранение локальных особенностей данных. Обработка многомерных данных в виде плоскостного послойного сканирования массивов не обеспечивает трехмерной связности (не обеспечивается изотропия данных по направлениям). Соответственно, потенциально более эффективными являются методы, основанные на сохранении локальных особенностей данных (обеспечивая трехмерную связность данных).

На общей схеме обработки томографических данных (рисунок 2.6) показано место компрессора/декомпрессора (кодека) в потоке данных, получаемых с компьютерного томографа.

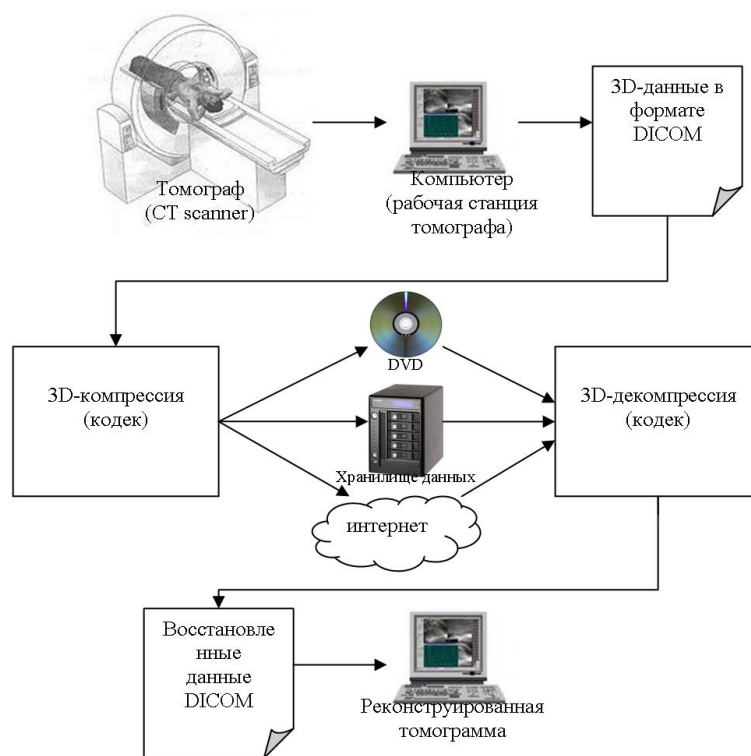


Рисунок 2.6 – Структурная схема метода сжатия с потерями для 3D-данных, полученных с использованием компьютерной томографии

Метод компрессии имеет структуру, сходную с True3D Vision, предложенную в [75] и основан на разбиении трехмерного пространства данных томографических данных (объединенной последовательности срезов) на элементы  $N \times N \times N$  точек, каждый из которых компрессируется независимо (рисунок 2.7).





Рисунок 2.7 – Последовательность компрессии томографических данных

На первом этапе выполняется нормализация данных, то есть преобразование данных, которое обладает следующими свойствами:

- 1) возможность взаимно-однозначного обратного восстановления,
- 2) способность к квантованию (загрубление или отбрасывание некоторых элементов данных с целью уменьшения битового объема),
- 3) полученные после нормализации данные более эффективно сжимаются методами вторичного сжатия [76].

В качестве алгоритмов нормализации может использоваться ассоциативно-пирамидальная развертка [21], многомерное дискретное косинусное преобразование [76] и ряд других, обладающих указанными свойствами.

Этап квантования повышает степень компрессии в случае, если задача допускает сжатие с потерями [77] (которое в случае томографических данных проявляется как появление дополнительных артефактов на изображении). Квантование позволяет представить данные с минимальной точностью, которая обеспечивает требуемый уровень качества изображения.

После этапов нормализации и квантования, полученные данные сжимаются с помощью алгоритмов вторичного сжатия (алгоритмы Хаффмана, арифметического кодирования, кодирования длинных последовательностей (RLE) и алгоритм сжатия на основе терминальных программ [78]).

Величина компрессия такого метода показала возможность сжатия от 10 до 120 раз в зависимости от степени квантования (количества вносимых артефактов).

## **Выводы по главе 2**

В главе рассмотрено современное состояние подходов к решению задачи сжатия 3D-данных, имеющей большое практическое значение для развития цифровых программируемых технологий в области компьютерного моделирования и быстрого прототипирования.

Рассмотрены методы компрессии без потерь на основе регулярных структур (остовного дерева) и в виде иерархических представлений, а также метод с потерями на основе многомерного дискретного косинусного преобразования для трехмерных данных, полученных с различных источников.

Показано, что на текущем этапе развития технических средств процедура обработки и упрощения данных является ресурсоемкой задачей и требует непосредственного участия человека. Поэтому важно не допускать потерь данных или их ухудшения в результате применения сжатия с потерями, чтобы иметь возможность корректной постобработки 3D-объекта без вносимых методом сжатия артефактов и безвозвратной потери данных.

В связи с этим наиболее целесообразным является разработка новых методов сжатия без потерь, не требующих предварительного упрощения модели или перевода ее в полигональную форму представления. Соответственно, в качестве исходных данных, удовлетворяющих условию наименьшей предварительной обработки, целесообразно использовать модель представления в виде облаков точек.

### Глава 3. Разработка модели, методов и алгоритма компрессии для хранения и передачи трехмерных пространственных данных

С целью обеспечения требований к задаче компрессии трехмерных пространственных данных без потерь, сформулированных в предыдущих главах, требуется разработать метод предназначенный для повышения компактности хранения 3D-моделей и упрощения передачи объемных данных по каналам связи в рамках цифровой программируемой технологии, учитывающий особенности процесса 3D-сканирования.

В связи с этим целесообразно решить научно-техническую задачу нахождения метода сжатия  $C$ , формирующего из облака точек  $S$ , битовую последовательность  $\{a_i\}$ , более короткую чем длина его исходного битового представления  $\{b_i\}$  при способе записи (формате)  $F: S \xrightarrow{C} \{a_1, a_2, \dots, a_q\}$ , где  $S \xrightarrow{F} \{b_1, b_2, \dots, b_n\}$  при котором  $n \geq q$ .

Учитывая априорные знания о структуре исходных данных (наличие компактных областей точек пространства, соответствующих поверхностям сканируемого объекта), было сделано предположение об эффективности использования алгоритма, основанного на применении развертки сложного вида для обхода точек трехмерного пространства.

В процессе работы 3D-сканер формирует наборы изображений с нескольких камер, которые с помощью программной обработки преобразуются в набор пространственных точек с измеренными расстояниями от них до сканера с общей системой координат, который может быть представлен в виде многомерного массива  $M$ , координаты элементов которого соответствуют координатам пространства, а значения (0 или 1) являются признаком наличия точки поверхности объекта по данным координатам.

Особенностями массива  $M$  являются компактные области точек пространства, соответствующие поверхностям сканируемого объекта,

соответственно для этапа нормализации целесообразно использовать алгоритм, основанный на переупорядочивании точек, сохраняющем их локальную близость.

В исследовании [76] была предложена обобщенная схема построения систем компрессии данных, учитывающая структуру как уже существующих, так и разрабатываемых методов, как с потерями, так и без потерь (рисунок 3.1). Особенностью схемы является возможность управления ожидаемыми свойствами проектируемой системы путем исключения или замены некоторых из ее элементов [79].

Согласно обобщенной схеме построения систем компрессии данных, чтобы при компрессии многомерных данных обеспечить отсутствие потерь при удовлетворительном уровне сжатия требуется пропустить этап семантической редукции. В качестве реализации этапа нормализации могут быть использованы разнообразные методы переупорядочения данных.

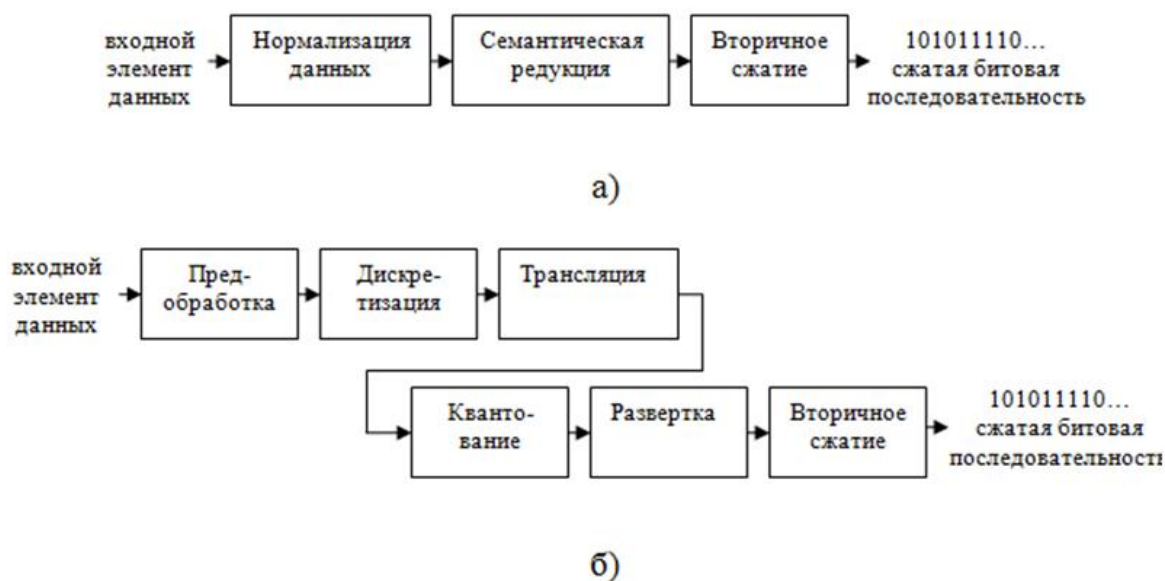


Рисунок 3.1 – Общая схема алгоритма компрессии:

а) обобщенная, б) детализированная

Проблема упорядочения точек многомерного пространства обусловлена принципом построения  $n$ -мерного декартова пространства [80], которое определено как прямое произведение:

$R^n = R_1^1 \times R_2^1 \times \dots \times R_i \times R_n^1$ , где однозначно определён порядок только на каждой координате  $R_i \in R^1$ .

Возможность же однозначного упорядочения точек в  $R^n$  принципиально связана с необходимостью априорного введения дополнительных условий и ограничений, в качестве которых чаще всего выступают различные функциональные зависимости, заданные формулой или геометрически, алгоритмы направленного перебора точек и прочие. Эти ограничения задают порядок обхода точек. По мере роста числа координат отношение информативной области ко всей рассматриваемой области имеет тенденцию к уменьшению.

Задача упорядочения точек  $R^n$  эквивалентна построению отображения  $f: R^n \rightarrow R^1$ , поскольку множества  $R^n$  и  $R^1$  равномощны, но только на  $R^1$  однозначно определен порядок [81].

При этом операция упорядочения должна производиться таким образом, чтобы точки близкие в пространстве  $R^n$  оказались близки и в пространстве  $R^1$ .

Разные виды отображения  $f: R^n \rightarrow R^1$  в различной степени сохраняют локальные и глобальные свойства пространства  $R^n$  [80].

### 3.1 Виды разверток

Рассмотрим наиболее часто применяемые типы разверток, которые можно разбить на три класса [82]:

1) Простейшие линейные развертки, упорядочение элементов двумерного массива в этом случае производится по строкам (столбцам) (рисунок 3.2). К этим же разверткам можно отнести и спиральную развертку (рисунок 3.3), осуществляющую упорядочение по последовательно убывающим и чередующимся строкам и столбцам.

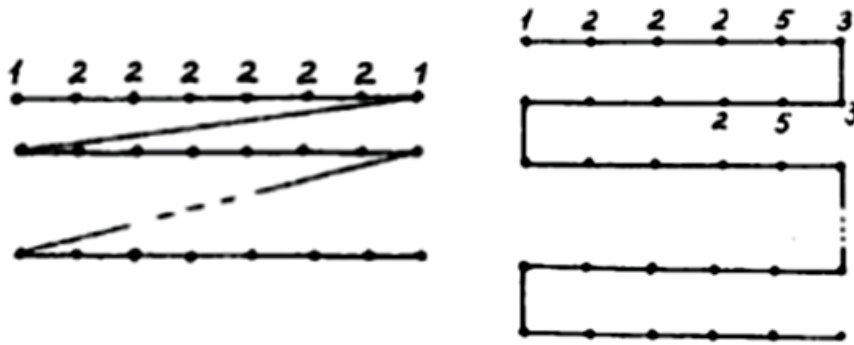


Рисунок 3.2 – Примеры линейных разверток

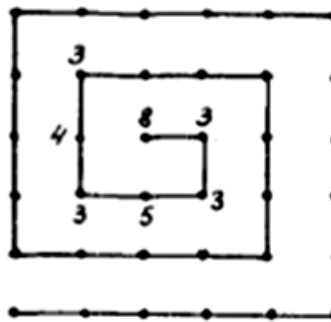


Рисунок 3.3 – Спиральная развертка

2) Рекурсивные развертки (рисунок 3.4) являются приближением к некоторой заполняющей пространство кривой [83–85]. Закон построения этих разверток полностью определяется первым и вторым приближением.

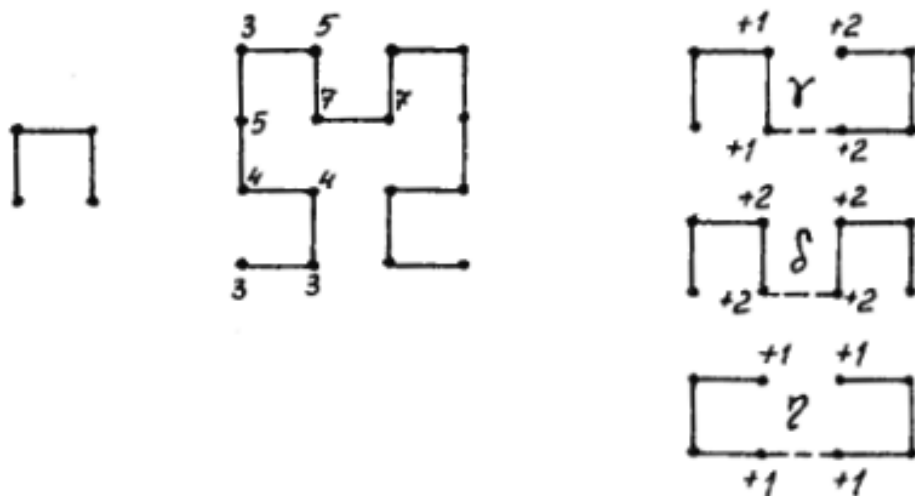


Рисунок 3.4 – Построение рекурсивной развертки

3) Замкнутые блочные развертки (рисунок 3.5). В этих развертках, в отличие от рассмотренных выше, последующее приближение получается из предыдущих не в соответствии с первым приближением, а за счет использования связей, обеспечивающих замыкание развертки.

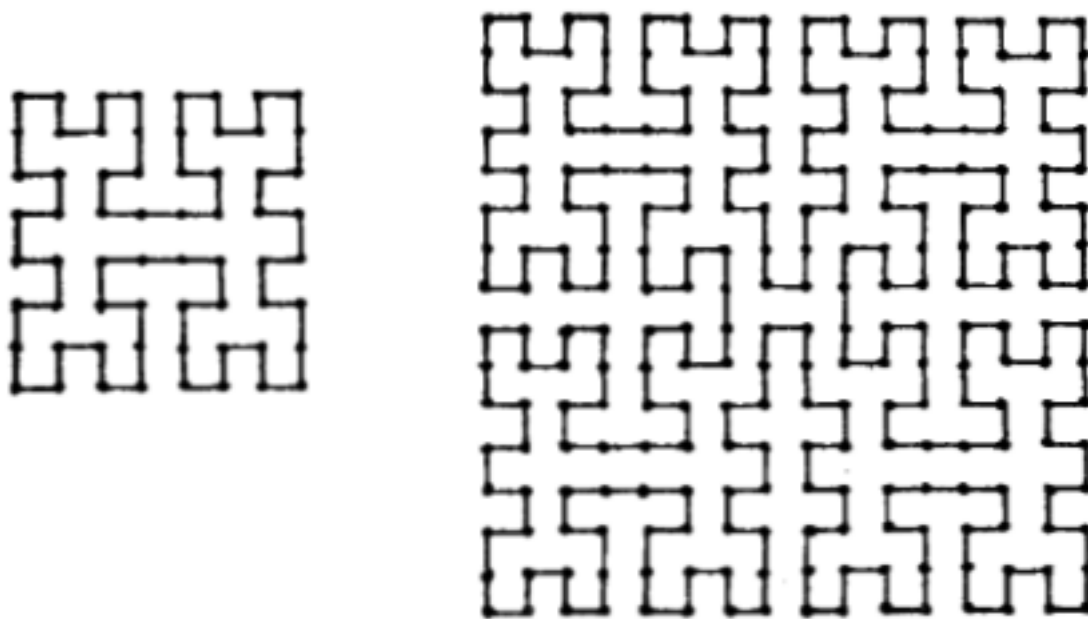


Рисунок 3.5 – Пример блочной развертки замкнутого типа

Такие развертки позволяют рассматривать группы близких элементов в пространстве и сохранять топологическую близость элементов исходного пространства и элементов полученного в результате развертки одномерного массива.

В аналитической математике известен класс отображений, хорошо сохраняющих локальные свойства (обладающие свойством непрерывности и удовлетворяющие условию Гёльдера [86]) и глобальные свойства (сохраняющие меру). Они были сконструированы в 1890 г. Дж. Пеано [87]. Впоследствии отображение  $f: \mathbb{R}^n \rightarrow \mathbb{R}^1$  стали называть пеановской кривой (или заполняющей пространство кривой – ЗПК).

Кривая Пеано представляет собой одновременно и объём, и площадь, и линию, обеспечивая при этом однозначный способ упорядочения точек в  $n$ -

мерном пространстве. Обход точек (упорядочение) происходит так, что прежде чем покинуть некоторую ограниченную область и перейти к следующей, будут пройдены все точки текущей области.

Точкам, близким в  $R^1$ , соответствуют точки, близкие в  $R^n$  то есть при  $f^{-1}: R^1 \rightarrow R^n$  сохраняются пространственные связи и отношения с ограниченным искажением метрики, подчинённым условию Гёльдера.

Всё сказанное в равной мере относится к приближениям пеановской кривой, если рассматривать окрестности точек специального вида — в виде вложенных областей с заданной точностью, определяемой номером приближения.

Конечное  $m$ -ое приближение пеановской кривой задает соответствие дискретных областей в  $R_m^n$  и в  $R_m^1$ . При этом свойство сохранения пространственных отношений и условие Гёльдера выполняются с точностью  $\varepsilon(m)$ , зависящего от номера приближения, условие взаимной однозначности выполняется полностью [81].

Такое представление в дискретном виде, согласно исследованиям, проведенным в работе [82], обладает наибольшей эффективностью для одномерного представления многомерных множеств. Способ построения этой кривой описан в [80].

### 3.2 Заполняющие пространство кривые и дискретные пространства

Каждая ЗПК представлена в виде непрерывной, нигде не дифференцируемой кривой, проходящей через все точки  $R^n$  (единичного гиперкуба). Алгоритм построения ЗПК для случая  $n = 2$  впервые был дан в геометрической форме в работе [88]. Собственно ЗПК изобразить нельзя, возможно лишь как это сделано [88], дать последовательность кривых, которые в пределе сходятся к ней.

Каждая такая кривая называется «приближением» ЗПК и имеет номер, определяемый ее местом в последовательности, например кривая (рисунок 3.6) есть первое, второе и третье приближение ЗПК соответственно. С геометрической



точки зрения  $m$ -е приближение ЗПК есть повторение некоторого числа раз  $(m - 1)$ -го приближения в соответствии с законом, определяемым первым приближением, поэтому ЗПК называют также самоподобными кривыми.

Поскольку любая ЗПК задает способ упорядочения точек пространства  $R^n$ , то задание ее эквивалентно заданию соответствия между точками единичного отрезка  $R^1$  и единичного гиперкуба  $R^n$ , и вопрос состоит в ее алгоритмическом описании.

Отметим, что при решении практических задач никогда не требуется знать точного соответствия координат точки в  $R^n$  и ее координат в  $R^1$ . Как правило, необходимым является лишь вполне определенное число десятичных знаков (что, в свою очередь, позволяет масштабировать значения координат до целых значений). Ограничение точности дает возможность для поиска соответствия точек  $R^n$  и  $R^1$  применять не собственно ЗПК, а лишь ее соответствующее приближение, что значительно упрощает решение практических задач, а число отличных друг от друга элементов пространства  $R^n$  становится конечным. Такой квантованный единичный гиперкуб  $R^n$  будем называть дискретным пространством.

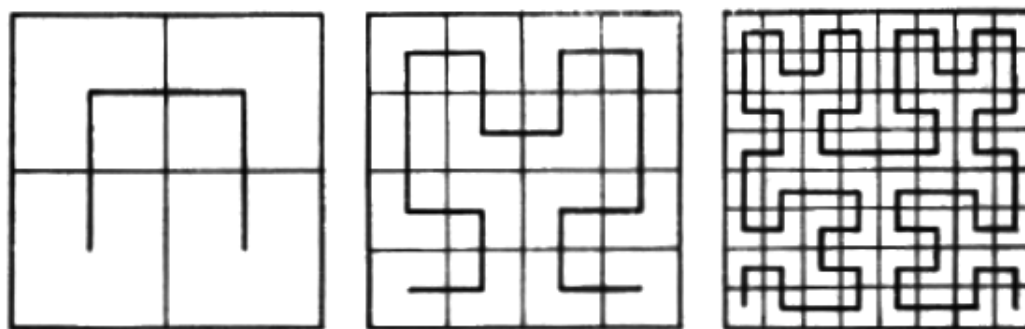


Рисунок 3.6 – Первое, второе и третье приближение ЗПК

### **Аналитическое описание отображения $R^1$ на $R^n$ и $R^n$ на $R^1$**

Для описания какой-либо ЗПК необходимо знать вид ее эталонной ячейки. Будем предполагать квазинепрерывность всех приближений ЗПК, это означает,

что все участки  $m$ -го приближения ЗПК имеют длину  $1/k^m$  и направлены параллельно осям системы координат. Предлагается следующий способ описания прохождения приближения ЗПК через центры соответствующих квантов. Будем записывать последовательность символов, каждый из которых соответствует участку кривой длиной  $1/k^m$ , соединяющему центры двух последовательно проходимых квантов. Абсолютная величина символа равна номеру координатной оси, которой параллелен этот участок кривой. Знак символа «+», если проход участка кривой осуществляется в направлении выделенной координатной оси, и «-» в противоположном случае.

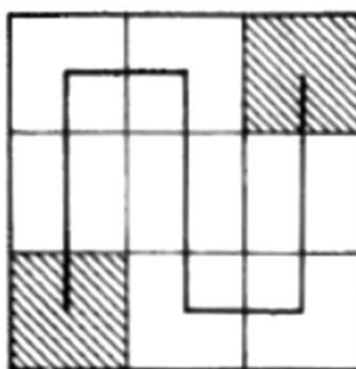


Рисунок 3.7 – Пример эталонной ячейки

Так, например, эталонную ячейку на рисунке 3.7 для  $k = 3$  можно описать следующей последовательностью:  $2, 2, 1, -2, -2, 1, 2, 2$  (принимая, что ось координат, параллельная строкам листа, направлена вправо и имеет номер 1, а перпендикулярная ей ось направлена вверх листа и имеет номер 2). Эталонную ячейку любой размерности и основания разбиения можно описать при помощи такой последовательности цифр. В принципе, возможно выбирать вид эталонной ячейки и основание разбиения для нее различными способами, однако экономное и легко алгоритмизируемое описание для эталонной ячейки получить не всегда легко. Ниже будет описано семейство эталонных ячеек для  $k = 2$  и произвольной размерности пространства  $n$ , которое получается из очень простой общей формулы. Назовем его  $\alpha$ -семейством.

Ячейки  $\alpha$ -семейства для  $n = 2, 3, 4$  показаны на рисунке 3.8, причем для  $n = 4$  (рисунок 3.8, в) пунктиром обозначен отрезок кривой, уходящий в четвертое измерение, контуры куба  $R^n$ ,  $R = 2, 3, 4$ , опущены. Если записать последовательности, соответствующие эталонным ячейкам этого семейства, то получим

$$\begin{aligned} n = 2 & \quad 1, 2, -1; \\ n = 3 & \quad 1, 2, -1, 3, 1, -2, -1; \\ n = 4 & \quad 1, 2, -1, 3, 1, -2, -1, 4, 1, 2, -1, 3, 1, -2, -1; \end{aligned} \quad (1)$$

Для любого  $n > 4$  можно написать аналогичную (1) последовательность

$$1, 2, -1, 3, 1, 2, -1, 4, 1, \dots, -1, n, 1, \dots, 1, -2, -1. \quad (2)$$

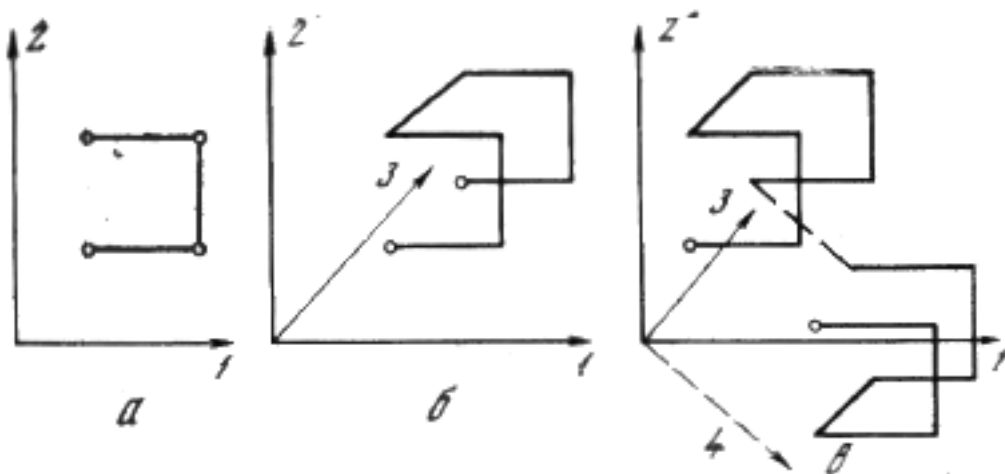


Рисунок 3.8 – Эталонные ячейки  $\alpha$ -семейства

Выведем формулу для  $g$ -го члена последовательности (2), которую обозначим  $F(g)$ ,  $g = \overline{0, 2^n - 1}$ . Разложим (2) по системе функций следующего специального вида:

$$\begin{aligned} \psi_1(g) &= 1, 0, -1, 0, 1, 0, -1 \dots \\ \psi_2(g) &= 0, 1, 0, 0, 0, -1, 0, 0, 0, 1, 0, \dots \end{aligned}$$

...

$$\psi_i(g) = \underbrace{0 \dots 0}_{2^{j-1}-1}, \underbrace{1, 0 \dots 0}_{2^{j-1}-1}, \underbrace{1, 0 \dots 0}_{2^j-1}, 1, \dots$$

Тогда

$$F(g) = \sum_{l=1}^n l \psi_l(g), \quad (3)$$

так как при каждом  $l$  только одна из функций  $\psi_l$ ,  $l = \overline{1, n}$ , имеет значение, отличное от нуля.

Записать функции  $\psi_j$ ,  $j = \overline{1, n}$ , аналитически можно разными способами, например, при помощи функции взятия целой части:

$$\psi_j(g) = \frac{1}{2} \left( (-1)^{\text{entier}(g/2^j+0,5)} - (-1)^{\text{entier}((g+1)/2^j+0,5)} \right), \quad j = \overline{1, n} \quad (4)$$

Таким образом, при подстановке  $g = 0, 1, 2, \dots, 2^n - 1$  и конкретном  $n$  из (3) получаем последовательность (2), являющуюся описанием эталонной ячейки  $\alpha$ -семейства  $n$ -й размерности. При других значениях  $k$  также можно строить различные эталонные ячейки. Например, для  $n = 2$  и  $k = 3, 4, \dots$  на рисунке 3.9 показано семейство эталонных ячеек, являющееся обобщением ячейки  $\alpha$ -семейства для  $n = 2$ ,  $k = 2$  (рисунок 3.8, а).

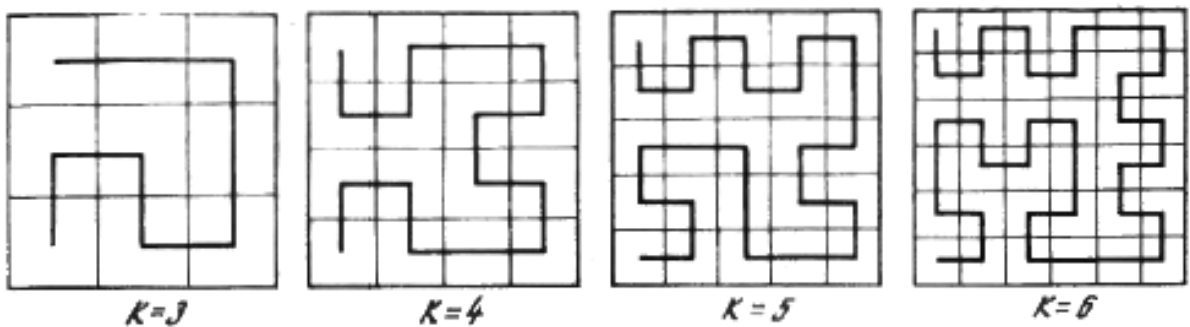


Рисунок 3.9 – Пример семейства эталонных ячеек

Представление ячеек при  $k > 2$  формулами, аналогичными (3), несколько сложнее. Кроме того, с увеличением  $k$  увеличивается произвол в построении

конкретных ячеек. Приведем пример для  $k = 3$  (рисунок 3.10),  $n = 2$  (а) и  $n = 3$  (б) по аналогии с рисунком 3.8 и формулами (1):

$$n = 2 \quad 1, 2, -1, 2, 1, 1, -2, -2;$$

$$n = 3 \quad 1, 2, -1, 2, 1, 3, -1, -2, 1, -2, -1, 3, 1, 2, -1, 2, 1, 1, -3, -3, -2, \\ 3, 3, -2, -3, -3;$$

$$n = 4 \quad 1, 2, -1, 2, 1, 3, -1, -2, 1, -2, -1, 3, 1, 2, -1, 2,$$

$$1, 4, -1, -2, 1, -2, -1, -3, 1, 2, -1, 2, 1, -3, -1,$$

$$-2, 1, -2, -1, 4, 1, 2, -1, 2, 1, 3, -1, -2, 1, -2,$$

$$-1, 3, 1, 2, -1, 2, 1, 1, -4, -4, -3, 4, 4, -3, -4,$$

$$-4, -2, 4, 4, 3, -4, -4, 3, 4, 4, -2, -4, -4, -3,$$

$$4, 4, -3, -4, -4.$$

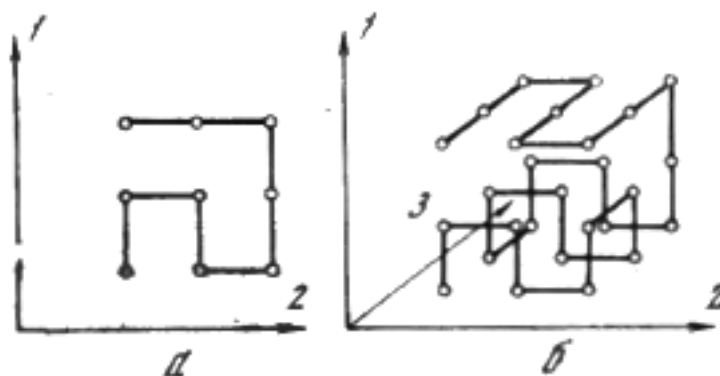


Рисунок 3.10 – Пример семейства эталонных ячеек

По аналогии принципиально можно получить выражения типа (1) или (2) для произвольных пар, однако необходимо отметить нарастающую сложность построений, необходимых для последующего обобщения.

Отметим, что, кроме описанного здесь  $\alpha$ -семейства, существует и может быть предложен ряд других ЗПК, сравнительный анализ которых может явиться предметом самостоятельного исследования.

### Алгоритмическое описание ЗПК

Алгоритмически описать ЗПК возможно с помощью рекурсивной процедуры [89], которую в виде псевдокода можно представить в следующем виде:

```

procedure hilbert(x, y, xi, xj, yi, yj, n)
if (n <= 0) then
    LineTo(x + (xi + yi)/2, y + (xj + yj)/2);
else
    {
    hilbert(x, y, yi/2, yj/2, xi/2, xj/2, n-1);
    hilbert(x+xi/2, y+xj/2, xi/2, xj/2, yi/2, yj/2, n-1);
    hilbert(x+xi/2+yi/2, y+xj/2+yj/2, xi/2, xj/2, yi/2, yj/2, n-1);
    hilbert(x+xi/2+yi, y+xj/2+yj, -yi/2, -yj/2, -xi/2, -xj/2, n-1);
    }
end procedure;

```

Инициализация построения ЗПК выполняется запуском описанной выше процедуры с начальными параметрами:

```

hilbert(0, 0, 1, 0, 0, 1, 1);

```

Практическая реализация генератора ЗПК на языке Java приведена в приложении А. Сравнительная оценка времени выполнения и требуемого объема памяти для построения ЗПК различного размера приведена в таблице 3.1 (для ЗПК размера меньше 64x64x64 используется 16 бит на элемент, для остальных – 32 бита).

Таблица 3.1 – Сравнительная оценка времени выполнения и потребного количества памяти при генерации ЗПК

Размер ЗПК	Время выполнения, мс	Объем памяти для хранения ЗПК
32x32x32	16	64кб
64x64x64	117	1 Мб
128x128x128	221	8 Мб
256x256x256	1152	64 Мб
512x512x512	8565	512 Мб
1024x1024x1024	120548	4 Гб

### 3.3 Динамическое разбиение и масштабирование пространства облаков точек

Данные, полученные от трёхмерного сканера после сканирования поверхности объекта, 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.

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

Облаком точек называется любой набор точек в пространстве  $E^d$  или набор проекций этих точек в подпространстве более низкой размерности, где  $E^d$  — это  $d$ -мерное евклидово пространство.

Под облаком точек будем понимать набор вершин в трёхмерной системе координат для представления поверхности объекта.

Для моделирования поверхностей используются два основных способа представления: задание поверхности на регулярной (структурированной) и

нерегулярной (хаотической) сетках (рисунок 3.11). Оба способа имеют свои преимущества и недостатки [90].

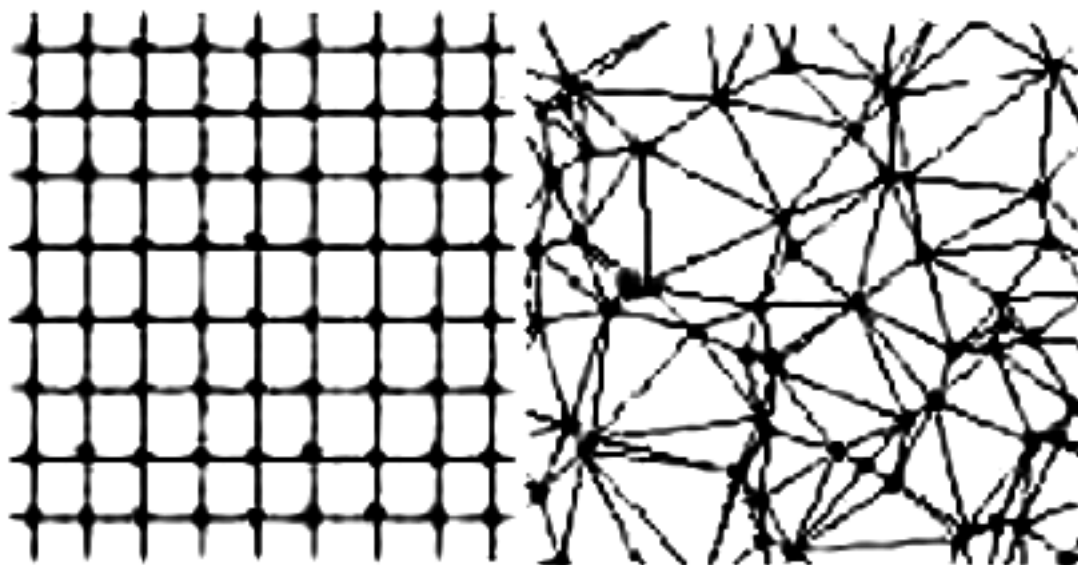


Рисунок 3.11 – Сетки с регулярной (слева) и нерегулярной структурой (справа)

### **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.

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 (грани) 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.

### **Сетки нерегулярной структуры**

Сетку, не являющуюся регулярной, будем называть нерегулярной. В нерегулярной сетке точки могут располагаться произвольно, то есть регулярная упорядоченная структура отсутствует. Точки поверхностей объектов, полученные методами трёхмерного сканирования, 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. При моделировании поверхностей с помощью нерегулярных сеток используют триангуляционные сетки Делоне.

Как было отмечено в п.4.3, при использовании ЗПК для успешного решения задачи взаимно-однозначного соответствия координат точки в пространстве  $R^3$ , представленной в дискретном виде с конечным числом десятичных знаков, координатам пространства  $R^1$  требуется производить динамическое разбиение и масштабирование пространства исходных облаков точек пространства  $R^3$ .

На начальном этапе определим область пространства, содержащего все облако точек сканированного объекта. Для этого путем нахождения максимальных и минимальных значений координат точек облака зададим параметры параллелепипеда, который охватывает все облако точек сканированного объекта (рисунок 3.12).

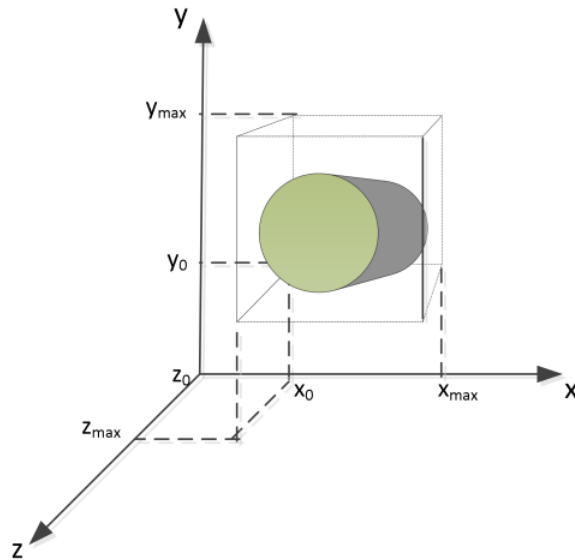


Рисунок 3.12 – Область пространства (параллелепипед), ограничивающая облако точек

Пусть  $x_0$ ,  $y_0$ ,  $z_0$  – смещение параллелепипеда относительно глобальной системы координат (система координат сканера), тогда:

$$x_0 = \min_i x_i$$

$$y_0 = \min_i y_i$$

$$z_0 = \min_i z_i.$$

Длины сторон параллелепипеда  $S_x$ ,  $S_y$ ,  $S_z$  определяются следующим образом:

$$S_x = \max_{i \neq j} |x_i - x_j|$$

$$S_y = \max_{i \neq j} |y_i - y_j|$$

$$S_z = \max_{i \neq j} |z_i - z_j|$$

Все преобразования над облаком точек будут осуществляться в локальной системе координат параллелепипеда, охватывающего облако точек (рисунок 3.13). Предложенный метод является адаптированной к решаемой задаче вариацией метода [91].

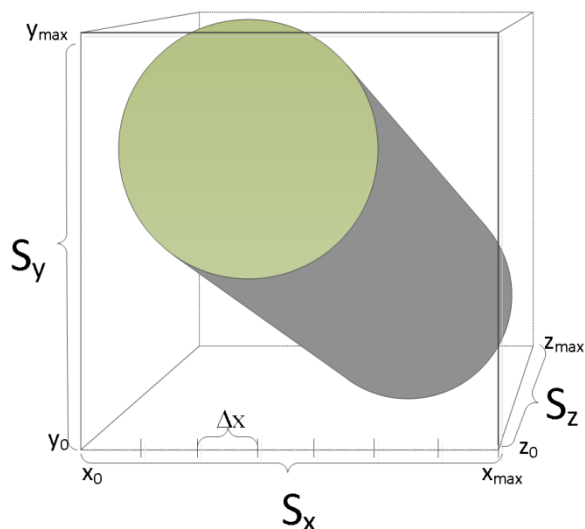


Рисунок 3.13 – 3D-объект в системе координат ограничивающего параллелепипеда

На следующем этапе определяется шаг квантования и коэффициенты, обеспечивающие представление облака точек на регулярной сетке без потери пространственного разрешения путем перевода координат в целочисленные значения (рисунки 3.14 и 3.15).

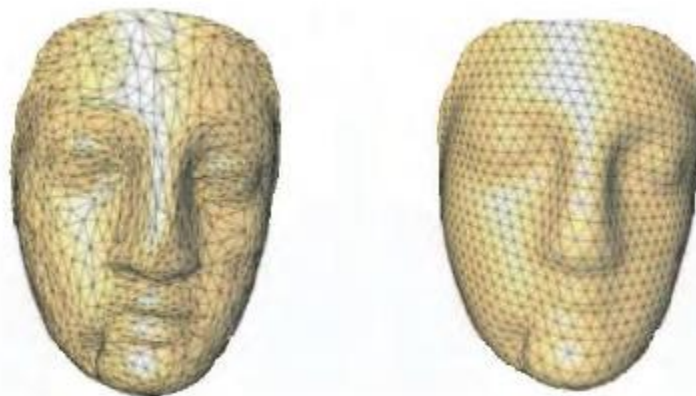


Рисунок 3.14 – Пересчёт нерегулярной сетки в регулярную [90]

В общем случае, шаг квантования  $\Delta x$  нужно искать исходя из условия нахождения целого количества шагов между любыми двумя координатами:  $\exists x_i, x_j : x_i = x_j \frac{1}{\Delta x} p, p \in \mathbb{Z}$ , выражения для  $\Delta y, \Delta z$  аналогичны.

В данной работе шаг квантования берется равным весу младшего разряда числовых значений координат полученных системой «сканер–ПО» и не изменяется в процессе работы,  $\Delta x = \delta$ , где  $\min \delta$  такое что  $10^\delta x \in \mathbb{Z}$  (рисунок 3.15).

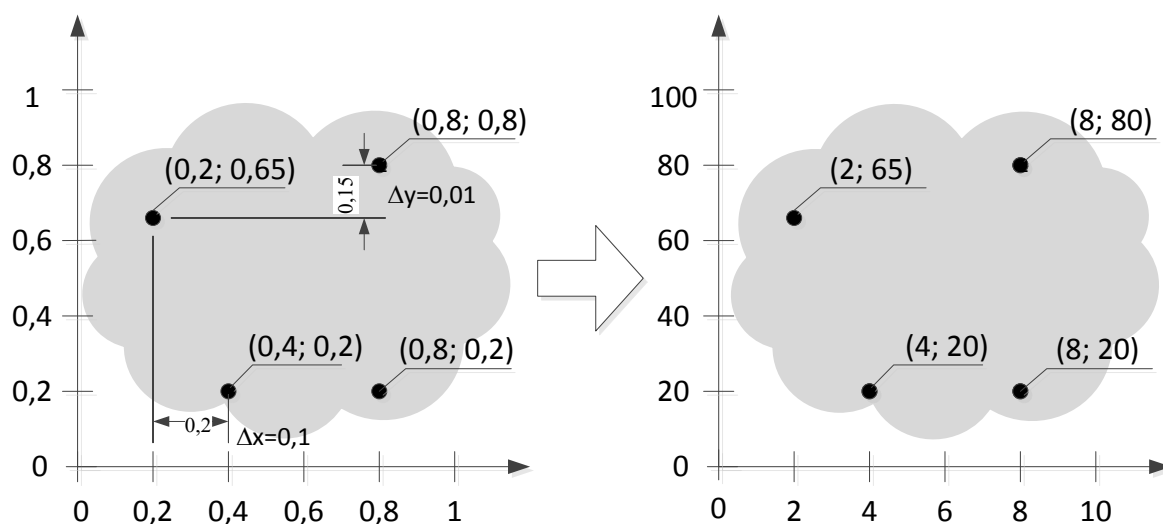


Рисунок 3.15 – Иллюстрация преобразования координат в облаке точек

Коэффициенты для перевода координат в целочисленные значения соответственно равны  $k_x = \frac{1}{\Delta x}$ ,  $k_y = \frac{1}{\Delta y}$  и  $k_z = \frac{1}{\Delta z}$ .

На практике объемы облаков точек, полученных после трехмерного сканирования, достигают сотен тысяч точек. Процедура преобразования координат облака точек к целочисленным значениям на регулярной сетке без потери разрешения приводит к увеличению пространства, требующегося для обхода по ЗПК. В связи с тем, что алгоритмическое построение последовательности точек ЗПК имеет степенную зависимость сложности (времени выполнения или объема памяти) от объема пространства (раздел 3.3, таблица 3.1), необходимо разбиение пространства на элементы меньшего размера. С целью ограничения разрядности для значения координат внутри каждого элемента разбиения до 8 бит было принято решение использовать элементы разбиения размером  $256 \times 256 \times 256$  ( $2^{8 \times 3}$  точек).

Для нахождения необходимого количества элементов разбиения для заданного объекта определим количество элементов разбиения по каждой оси координат:

$$N_x = \left\lceil \frac{S_x}{s_x \times \Delta x} \right\rceil$$

$$N_y = \left\lceil \frac{S_y}{s_y \times \Delta y} \right\rceil$$

$$N_z = \left\lceil \frac{S_z}{s_z \times \Delta z} \right\rceil,$$

где  $S_x, S_y, S_z$  – размер параллелепипеда, ограничивающего 3D-объект по осям,  $s_x, s_y, s_z$  – линейные размеры элементов разбиения, а  $\lceil x \rceil = \min\{n \in \mathbb{Z} | n \geq x\}$  — операция округления к большему. Тогда  $N = N_x \times N_y \times N_z$  – общее количество элементов разбиения для 3D-объекта (рисунок 3.16).

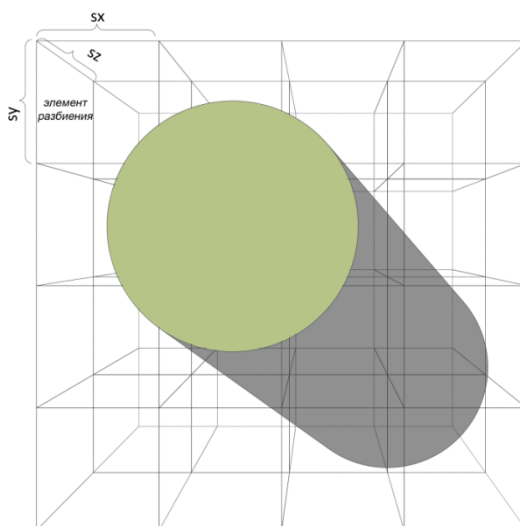


Рисунок 3.16 – К понятию динамического разбиения и масштабирования пространства облаков точек

Используя вышеописанное динамическое разбиение пространства, содержащего 3D-объект, и его масштабирование в пространстве, координаты точки  $(x', y', z')$  внутри элемента разбиения  $(i, j, l)$  вычисляются согласно выражению:

$$\begin{cases} x' = k_x(x - x_0) - i \times sx \\ y' = k_y(y - y_0) - j \times sy, \text{ где } i, j, l - \text{ позиция текущего элемента разбиения} \\ z' = k_z(z - z_0) - l \times sz \end{cases}$$

пространства,  $x_0, y_0, z_0$  – начальное смещение облака точек относительно нуля системы координат,  $k_x, k_y, k_z$  – коэффициенты масштабирования по ширине, высоте и глубине соответственно,  $sx \times sy \times sz$  – размер единичного элемента, к которому применяется компрессия данных.

### 3.4 Разработка модели представления пространственных объектов

При отсутствии технической или организационной возможности и ресурсов для преобразования первичного цифрового скана в 3D-модель отражающую сканируемый объект с заданным качеством, требуется сохранение исходного массива данных чаще всего представляющего собой список координат всех точек принадлежащих объекту (облако точек).

В данной модели представления поверхности 3D-объекта каждая точка исходного облака точек  $p(x, y, z) \in S$  представлена в декартовой системе координат  $x, y, z \in \mathbb{R}$  относительно некоторого выбранного начала координат с масштабом, обеспечиваемым разрешающей способностью аппаратной части сканера.

Встроенное ПО сканера позволяет преобразовывать исходное облако точек в полигональную модель представления  $\{S, F\}$  путем введения дополнительной связности точек, где  $S$  — облако точек, а  $F$  — множество граней  $f(p_i, p_j, p_k) \in F$ , каждая грань представлена треугольником с вершинами  $p_i, p_j, p_k$ . Однако, задача построения множества граней  $F$  для отсканированного 3D-объекта во-первых не всегда имеет однозначное решение, а во-вторых использование такой модели представления не дает преимуществ для последующего сжатия без потерь, так как прирост избыточности не компенсируется введением частичной упорядоченности, обусловленной введением произвольной связности.

Модель облака точек по своему определению описывает неструктурированный набор точек, что обуславливает наличие информационной избыточности представления [17]. При увеличении объема исходных данных такая модель требует коррекции путем задания порядка перечисления элементов представления для снижения информационной избыточности.

В настоящем исследовании для решения поставленной задачи разработки метода сжатия цифровых сканов без потерь разрабатывается модель представления пространственных объектов, использующая упорядоченное одномерное представление облаков точек 3D-объекта:  $\{S, G, R(G)\}$ , где  $S$  – облако точек,  $G$  – кривая, задающая порядок точек,  $R(G)$  – функция развертки, соответствующая порядку обхода точек  $G$ . Приведенное ниже функциональное описание является основой модели представления пространственных объектов, использующей упорядоченное одномерное представление.

Функционально процесс сжатия без потерь можно представить в следующем виде. Пусть  $S$  – множество точек, описывающих исходный объект в форме облака точек. Используя отображение  $S \xrightarrow{N} S_N$  производится нормализация координат, с целью получить целые значения координат  $x', y', z' \in Z$ ,  $p'(x', y', z') \in S_N$ .

После этого, используя отображение  $S \xrightarrow{G} S_G$  с использованием заполняющей пространство кривой  $G$ :  $q = f(x, y, z)$ , задавая отношение линейного порядка для элементов  $p'$  формируется упорядоченный кортеж точек. Здесь  $f(x, y, z)$  — функция порядкового номера  $q$  элемента  $p'(x', y', z') \in S_N$  в кортеже  $S_G = \{p'_G\}$

$$\{a_q\}, a_q = \begin{cases} 0, & \text{если } p(x, y, z) \notin S, q = f(x, y, z) \\ 1, & \text{если } p(x, y, z) \in S, q = f(x, y, z) \end{cases}$$

Представив полученный кортеж  $\{a_q\}$  как битовую последовательность длиной  $|S|$ , к ней применяются описанные ниже операции группового кодирования и вторичного сжатия, после которых формируется окончательный битовый поток.

### 3.5 Разработка метода и алгоритма сжатия пространственных объектов

Для реализации алгоритма все пространство, содержащее объект, разбивается на трехмерные ячейки размером (элементы разбиения), каждая из которых сжималась независимо от других, а результаты сжатия объединялись в единый выходной поток.

Как было указано выше, каждый элемент разбиения может быть представлен в виде многомерного массива  $\mathbf{M}$ , координаты элементов которого соответствуют координатам пространства элемента разбиения, а значения (0 или 1) являются признаком наличия точки поверхности объекта по данным координатам. Для эффективного представления массива  $\mathbf{M}$  предлагается следующий алгоритм (рисунок 3.17).

Используем тот факт, что облако точек является трехмерным битовым массивом  $\mathbf{M}$ , который при конечном разрешении, обеспечиваемом сканером, является разреженным и может быть достаточно эффективно скомпрессирован [16].

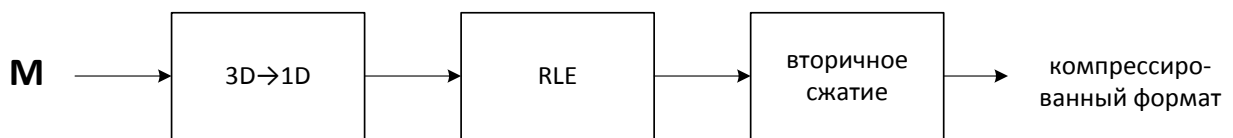


Рисунок 3.17 – Общая схема алгоритма компрессии

#### Шаг 1 алгоритма: преобразование $3D \rightarrow 1D$ .

Основным этапом сжатия является преобразование трехмерного массива  $\mathbf{M}$  в линейную бинарную последовательность (в терминологии работ, проводимых в лаборатории автоматизации научных исследований СПИИРАН этот этап является нормализацией, ориентированной на конкретный тип данных).

Для случая блока  $8 \times 8 \times 8$ , порядок обхода точек при преобразовании  $R^3 \rightarrow R^1$  с использованием ЗПК будет иметь вид, представленный на рисунке 3.18.



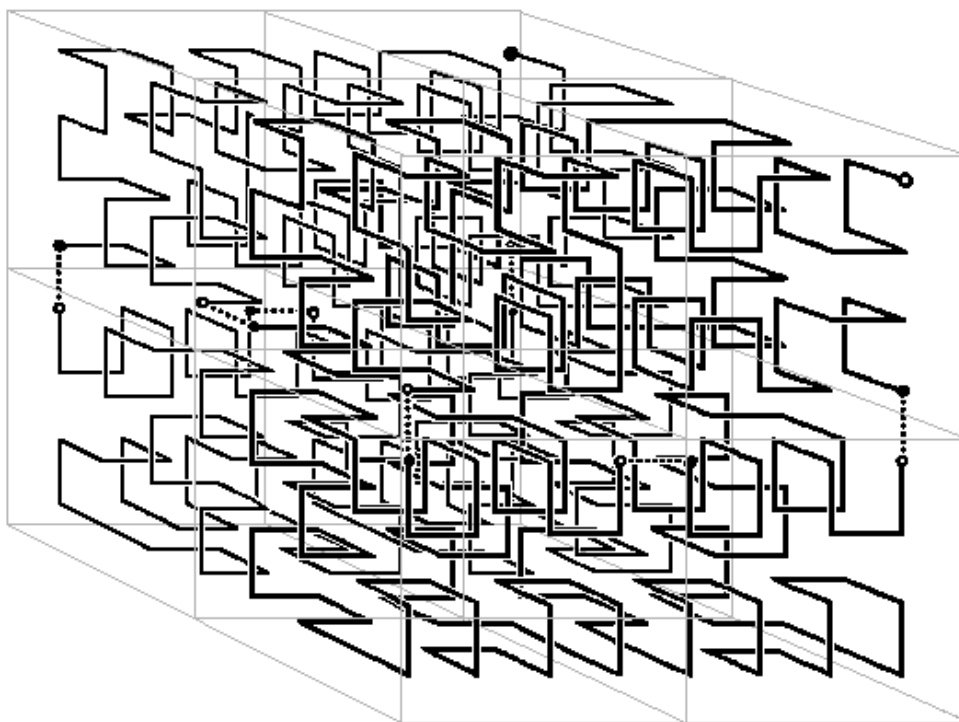


Рисунок 3.18 – Заполняющая пространство кривая для трехмерного случая с размерностью  $8 \times 8 \times 8$

### **Шаг 2 алгоритма: групповое кодирование.**

После преобразования массива  $\mathbf{M}$  из  $R^3 \rightarrow R^1$  в сформированной битовой последовательности присутствуют длинные последовательности нулей, которые при использовании группового кодирования типа RLE (англ. Run-length encoding, RLE) позволяет достичь уменьшения битового объема.

Групповое кодирование длинных последовательностей нулей осуществляется с помощью вариации алгоритма RLE [92]. При этом формируются пары вида  $\langle \text{пропустить, число} \rangle$ , где «пропустить» является счетчиком пропускаемых нулей, а «число» — значение, которое необходимо поставить в следующую ячейку. Так, вектор  $(42, 3, 0, 0, 0, -2, 0, 0, 0, 0, 1)$  будет свернут в пары  $(0,42) (0,3) (3,-2) (4,1)$ , а конкретное битовое представление будет зависеть от реализации [74].

В предложенном методе используется модифицированный (семантико-ориентированный) вариант алгоритма RLE, ориентированный на кодирование длинных последовательностей нулей, а прочие значения кодируются непосредственно собственным значением.

При этом битовая последовательность: (0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0) будет преобразована в вид (0,5), 1, (0,3), 1, 1, (0,1), 1, (0,3) и представлена в виде последовательности байтов (0, 5, 1, 0, 3, 1, 1, 0, 1, 1, 0, 3), при этом значение байта "0" зарезервировано для кодирования последовательностей нулей.

### Шаг 3 алгоритма: Вторичное сжатие

В качестве заключительного этапа вторичного сжатия может быть использован один из методов энтропийного сжатия, например, арифметическое сжатие [93], сжатие с предсказанием по частичному совпадению (PPM) [92], группа алгоритмов сжатия LZxx [92], сжатие Хаффмана [94] и ряд других, что позволяет уменьшить среднюю длину кодового слова для символов алфавита. Сравнительная эффективность реализаций различных алгоритмов (на примере данных после этапа группового кодирования) приведена в таблице 3.2 и на рисунке 3.19.

Таблица 3.2 – Сравнение эффективности методов компрессии для вторичного сжатия

Метод	Сжатие, раз	Время выполнения, с
BZip2	645	0,4
PPMD	584	0,31
LZMA	573	1,02
LZMA2	571	1,05
Zip	366	0,81
DEFLATE	356	0,93
DEFLATE64	191	0,94

В качестве базового варианта этапа вторичного сжатия был выбрана реализация DEFLATE как компромиссный вариант по производительности и эффективности сжатия.

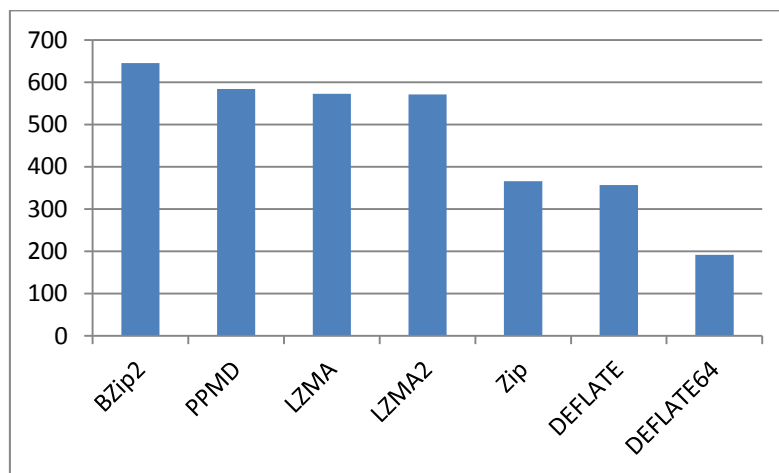


Рисунок 3.19 – Сравнение методов вторичного сжатия для структур данных, полученных на этапе нормализации

### 3.6 Программная реализация системы сжатия пространственных объектов

Программная реализация интерактивной системы сжатия цифрового представления пространственных объектов является кроссплатформенным (Windows и Linux) приложением на языке JAVA и работает среде выполнения Java Runtime Environment версии не ниже 1.6.

Разработанная система построена по модульной архитектуре. Структурная схема взаимодействия модулей интерактивной программной системы компрессии данных приведена на рисунке 3.20.

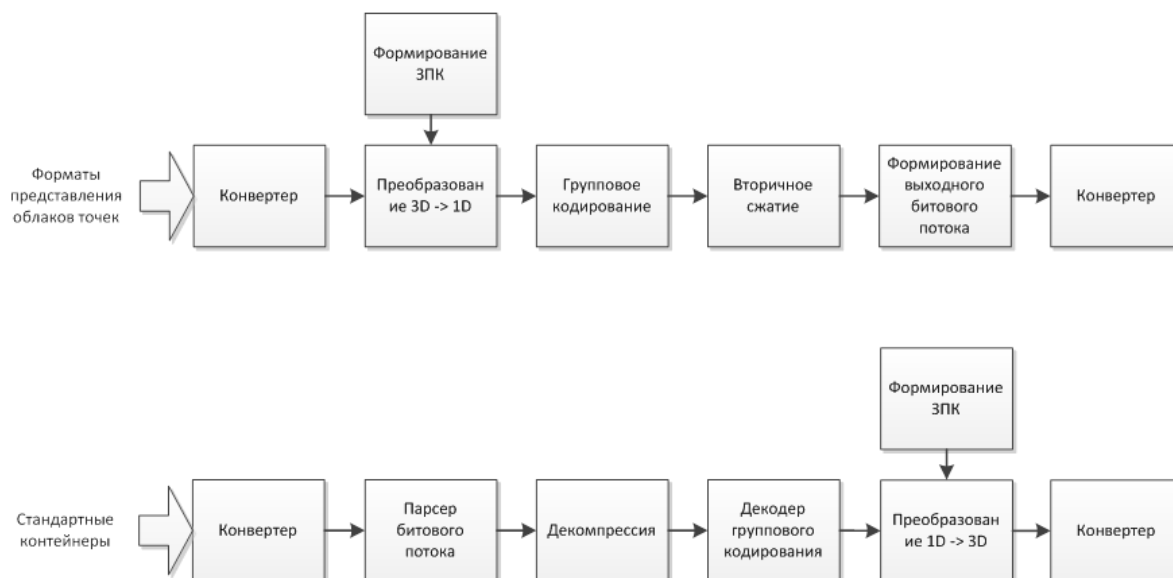


Рисунок 3.20 – Структурная схема интерактивной программной системы компрессии 3D-данных

Модуль «Конвертер». Модуль реализует функцию преобразования различных стандартных форматов файлов в формат внутреннего представления (массив  $\mathbf{M}$ , раздел 3.5) и обратно.

Модуль формирования ЗПК. Модуль служит для генерации последовательности обхода точек объема пространства по закону 3D-ЗПК (возможен вариант использования заранее подготовленных данных из файла). Фрагмент программного кода генератора ЗПК приведен в Приложении А.

Модуль преобразования  $R^3 \rightarrow R^1$ . Модуль служит для организации процесса выборки элементов трехмерного массива точек, соответствующего элементу разбиения в линейную последовательность значений, причем закон выборки определяется формой кривой ЗПК. Модуль использует данные о последовательности обхода точек объема пространства, сформированные модулем формирования ЗПК. Фрагмент программного кода для преобразования  $R^3 \rightarrow R^1$  с формированием структуры данных, подлежащей дальнейшей компрессии приведен в Приложении А.

Модуль группового кодирования. Модуль реализует байт-ориентированный метод группового кодирования длинных последовательностей нулей из битового потока.

Модуль вторичного сжатия. Модуль использует алгоритм сжатия данных Deflate (стандарт RFC 1951 [95]) для компрессии битового потока, полученного с выхода модуля группового кодирования, формируя битовый поток меньшего объема.

Модуль формирования выходного потока. Модуль служит для преобразования битового потока, полученного после применения всех этапов компрессии данных, в транспортный поток, предназначенный:

- для передачи по каналам связи;
- для сохранения в файл путем снабжения заголовками и метаданными;
- для размещения полученного потока в стандартных типах контейнеров (XML, JSON и т.п.).

Кроме того, модуль маркирует фрагменты битового потока, соответствующего конкретным элементам разбиения, помечая пустые (не содержащие точек) элементы разбиения специальным маркером.

### **Выводы по главе 3**

В главе разработана модель представления 3D-объектов, полученных в результате 3D-сканирования. Предложен метод и алгоритм компрессии для эффективного представления 3D-объектов, учитывающий особенности 3D-сканирования и предназначенный для повышения компактности их хранения в цифровых хранилищах, а также для упрощения передачи объемных данных по каналам связи в рамках цифровой программируемой технологии.

Верификация разработанной модели обеспечивается корректностью выполнения всех этапов построения модели и тестирования ее программной реализации, а также корректным восстановлением цифрового образа исходного объекта.

Валидация модели обеспечивается неизменностью геометрических свойств 3D-объекта при уменьшении битового объема представления.

Сравнительный анализ эффективности сжатия разработанного алгоритма в различных режимах работы приведен на рисунке 3.21.

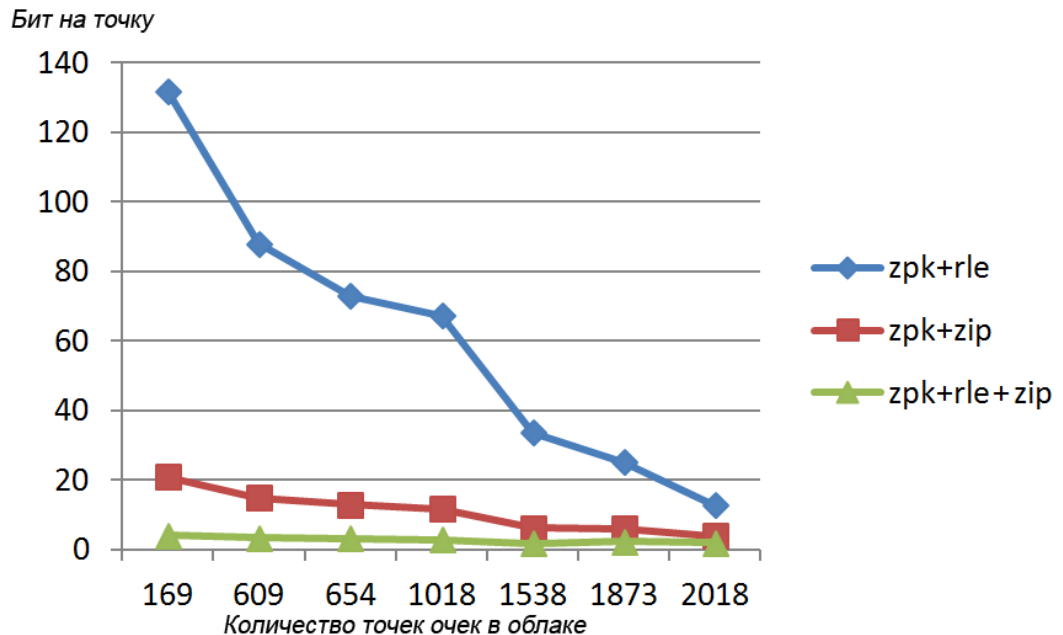


Рисунок 3.21 – Сравнительный анализ эффективности разработанного алгоритма для различных режимов работы

На рисунке 3.21 используются следующие обозначения режимов работы алгоритма:

- zpk+rle – вариант алгоритма с использованием ЗПК с групповым кодированием без энтропийного сжатия;
- zpk+zip – вариант алгоритма с использованием ЗПК с энтропийным сжатием без группового кодирования;
- zpk+zip+rle – вариант алгоритма с использованием ЗПК с энтропийным сжатием после группового кодирования.

## Глава 4 Экспериментальная апробация и оценка эффективности метода сжатия

В данном разделе исследуются характеристики и свойства технической системы «3D-сканер — программное обеспечение сканера (ПОС) — система компрессии».

Для проведения экспериментов использовались следующие виды данных:

- 1) Полученные с помощью 3D-сканера Artec Spider облака точек (пример на рисунке 4.1). Параметры сканирования: процесс сканирования производился в «реальном времени». Облако точек получено за один проход сканирования.

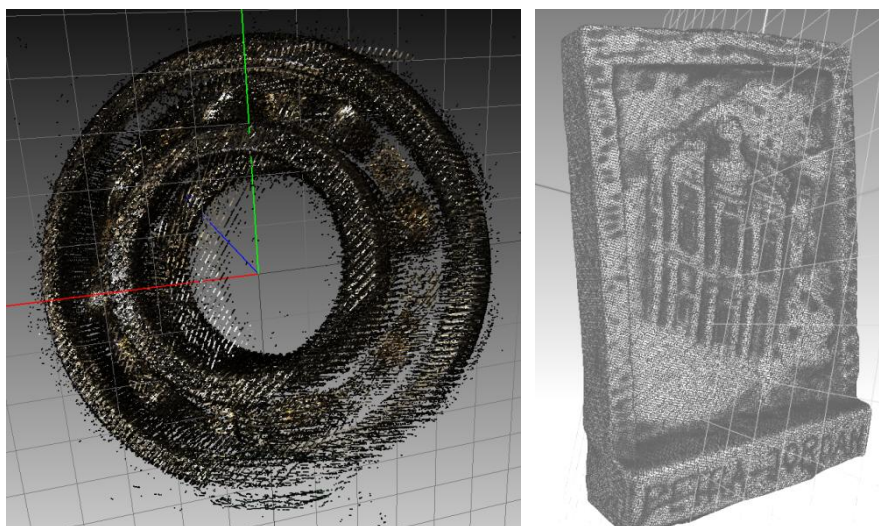


Рисунок 4.1 – Исходные облака точек с 3D-сканера Artec Spider

- 2) Искусственно синтезированные в программах 3DS Max, AutoCAD (рисунок 4.2);

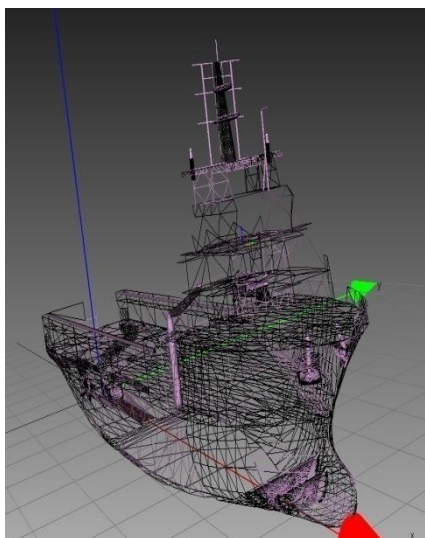


Рисунок 4.2 – Искусственно синтезированный 3D-объект

- 3) Стандартные коллекции 3D-объектов из Интернет-источников (пример на рисунке 4.3).

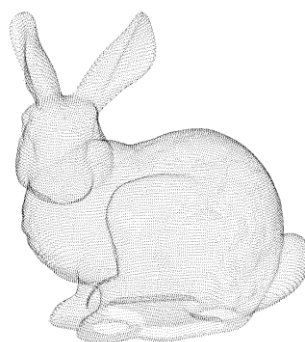


Рисунок 4.3 – 3D-объект из стандартных коллекций

Для эффективного использования средств 3D-сканирования необходимо провести исследование влияния свойств поверхностей и положения сканера на результаты сканирования; проработать различные алгоритмы сканирования применительно к разным типам сканируемых объектов.

На основе экспериментальных данных было проведено сравнение эффективности разработанного метода с известными существующими методами (в рамках данных, предоставленных авторами методов) (раздел 2.2), а также определены особенности и выявлены ограничения

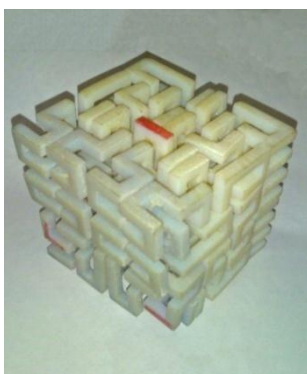


технологии 3D-сканирования, которые могут стать основой технического задания к дальнейшим исследованиям.

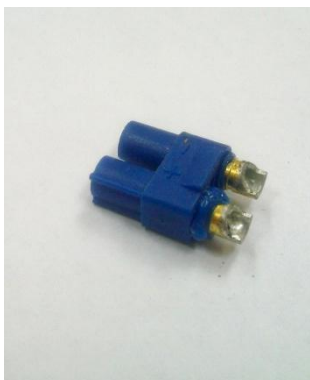
#### 4.1 Виды объектов

Исходные объекты для экспериментов можно классифицировать по наличию внутренней структуры:

- с внутренней структурой



- без внутренней структуры



по характеру поверхности:

- гладкие



- со сложной поверхностью (мелкие детали с размером превышающим разрешающую способность сканера)



- с зашумленной поверхностью (мелкие детали с размером меньше разрешающей способности сканера)



#### **4.2 Исследование особенностей технической системы «3D-сканер Artec Spider – программное обеспечение сканера – система сжатия»**

Как было отмечено в главе 1, процесс сканирования 3D-объекта включает в себя ряд последовательных действий, каждое из которых влияет на конечный результат сканирования. В связи с этим программное обеспечение сканера (ПОС), осуществляющее реконструкцию облака точек из набора изображений, полученных с нескольких камер 3D-сканера, а также их постобработку имеет смысл рассматривать как техническую систему «3D-сканер – ПОС». В качестве ПОС в экспериментах использована «Artec Studio версии 9.2».

Влияние ПОС проявляется в первую очередь в изменении (часто искажении) формы объекта (рисунок 4.4) при попытке компенсировать ошибки в работе сканера.

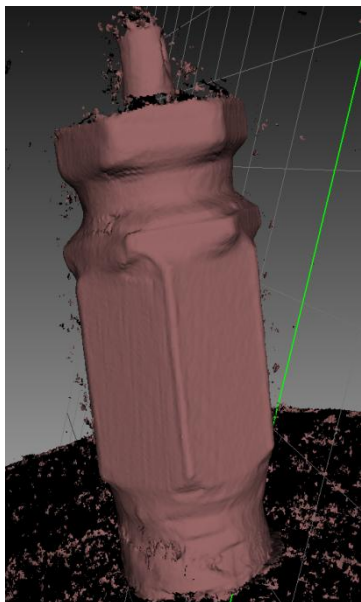


Рисунок 4.4 – Влияние особенностей технической системы 3D-сканер–ПОС на изменение формы объекта (слева – отсканированный объект, справа – фотография исходного объекта)

Часто попытка ПОС компенсировать недостаток точек поверхности в области точек приводит к ухудшению качества цифрового скана.

Для объектов, имеющих большое количество элементов поверхности, размер которых не может быть воспроизведен в связи с ограничением разрешающей способности системы цифрового сканирования возможна потеря точности мелких элементов (витков резьбы на рисунке 4.5) за счет попытки улучшения путем сглаживания поверхности в системе "сканер+ПО".

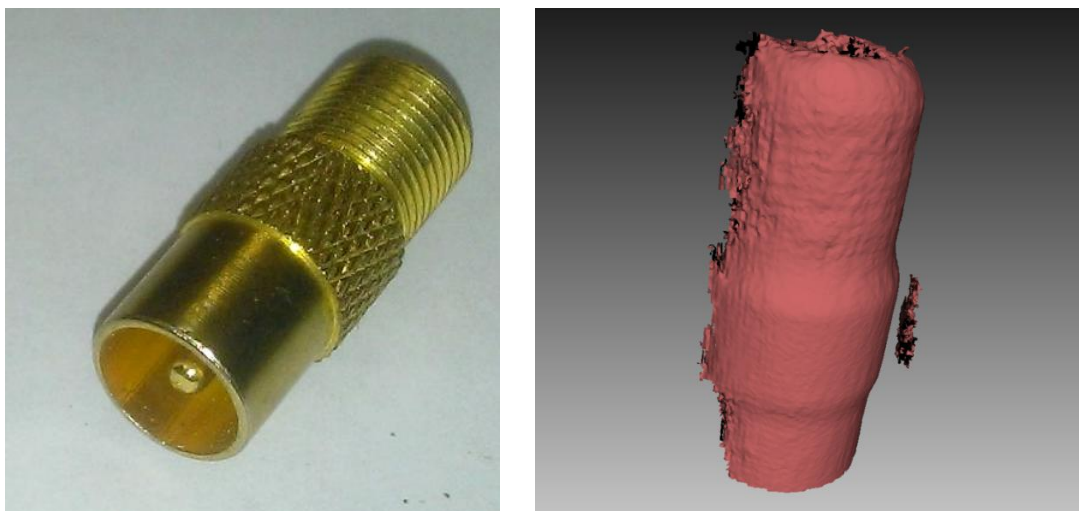


Рисунок 4.5 – Потеря мелких элементов поверхности.

Слева – фотография объекта, справа – визуализация цифрового скана.

Существуют типы поверхности (в первую очередь прозрачные и зеркальные), которые не удастся отсканировать из-за особенностей оптического принципа работы 3D-сканера.

На рисунке 4.6 приведены примеры объектов, для которых не удалось получить 3D-скан из-за особенностей сканируемой поверхности.

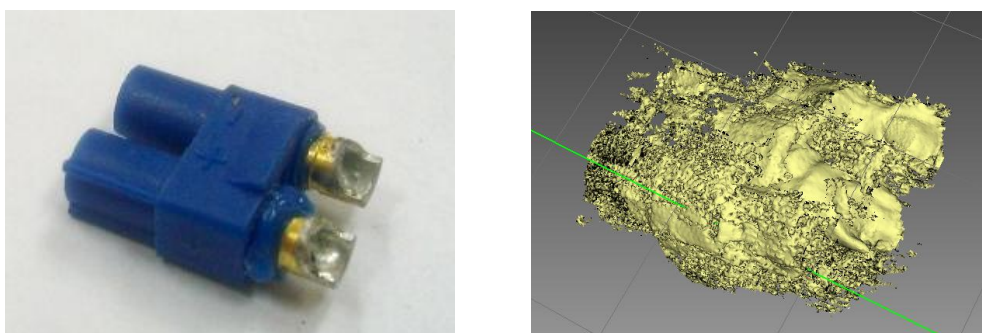


Рисунок 4.6 – Невозможность получения цифрового скана из-за особенностей сканированной поверхности

На некоторых сканах фиксируется отсутствие мелких отверстий, вызванное ошибочным применением сглаживания поверхности в ПО (рисунок 4.7).



Рисунок 4.7 – Отсутствие мелких отверстий. Слева – фотография объекта, справа – визуализация цифрового скана

Артефакты сканирования в виде дублирования отдельных элементов сканирования, вызванные потерей ключевых точек в процессе сканирования за счет симметричности объекта сканирования приведены на рисунке 4.8.

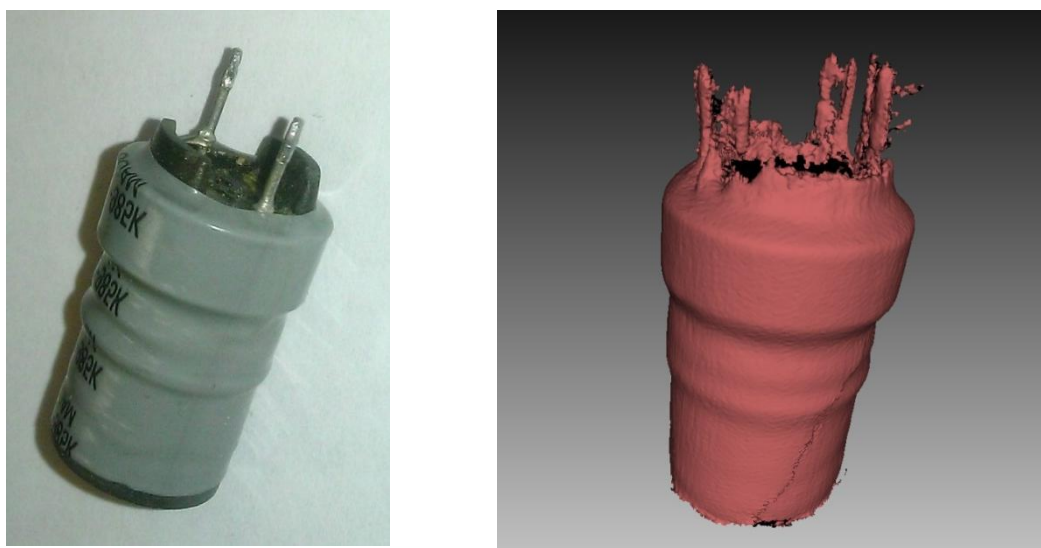


Рисунок 4.8 – Артефакты сканирования. Слева – фотография объекта, справа – визуализация цифрового скана



На рисунке 4.9 приведен цифровой скан, на котором при хорошем качестве 3D-сканирования элементов большого размера и малой детализации присутствует множество артефактов сканирования на тонких элементах конструкции.

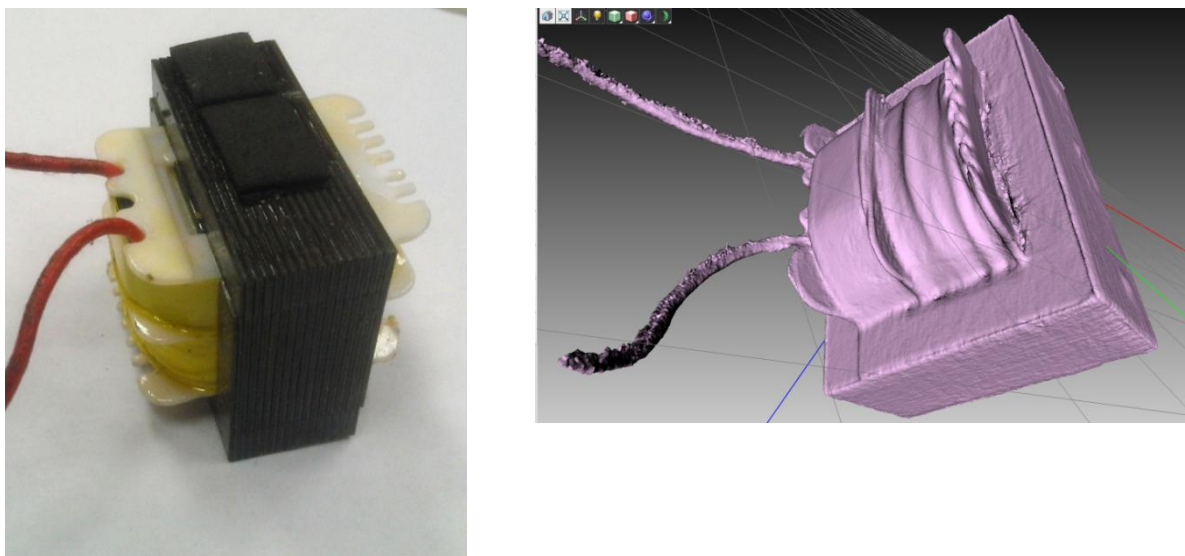


Рисунок 4.9 – Артефакты сканирования на тонких элементах конструкции.

Слева – фотография объекта, справа – визуализация цифрового скана

Отсутствие мелких деталей и материал объекта, не дающий световых бликов, повышают вероятность получения высокого качества цифрового скана с хорошей детализацией (рисунок 4.10).

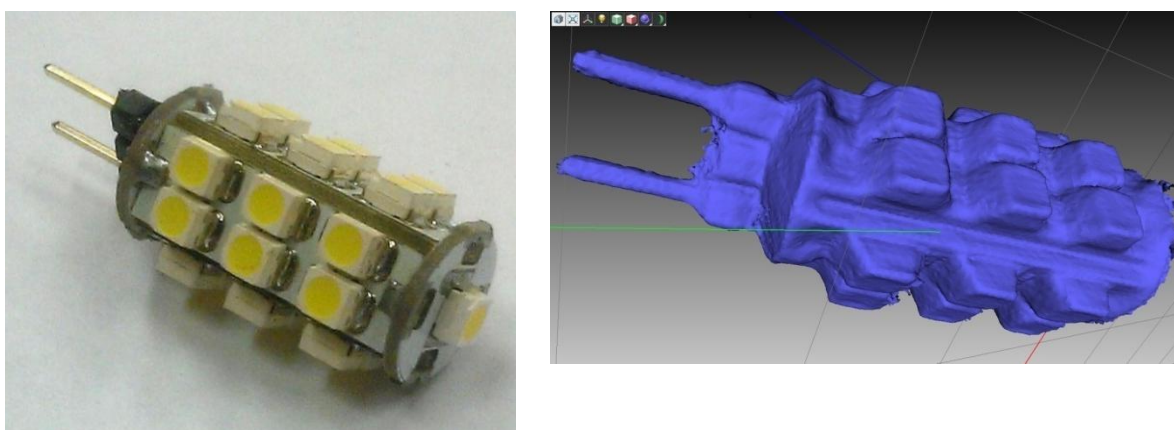


Рисунок 4.10 – Влияние материала поверхности объекта на качества 3D-скана.

Слева – фотография объекта, справа – визуализация цифрового скана

На рисунке 4.11 приведен пример цифрового скана с дефектами поверхности (отсутствие точек), вызванными световыми бликами.

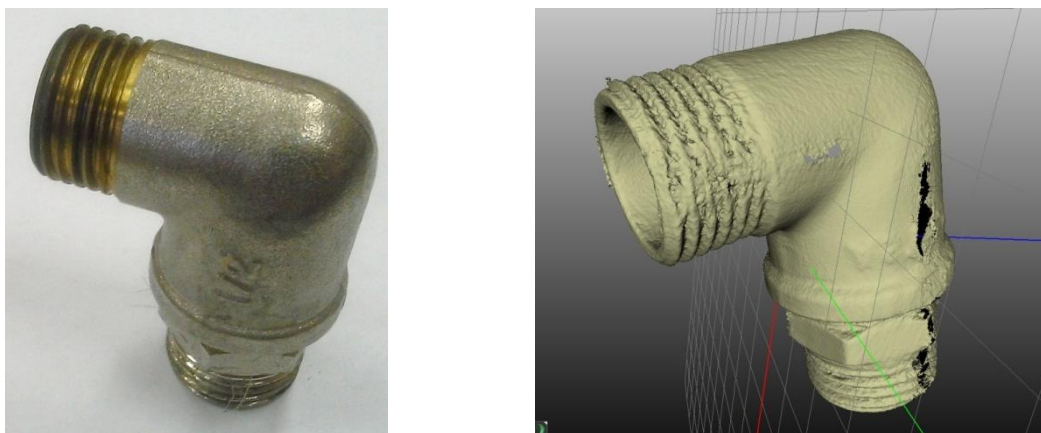


Рисунок 4.11 – Дефекты поверхности вызванные световыми бликами.

Слева – фотография объекта, справа – визуализация цифрового скана

Невозможность долговременного отслеживания опорных точек, вызванная большим количеством тонких элементов, приводит к ошибкам совмещения кадров, что обуславливает появление в цифровом скане большого количества артефактов сканирования (рисунок 4.12).

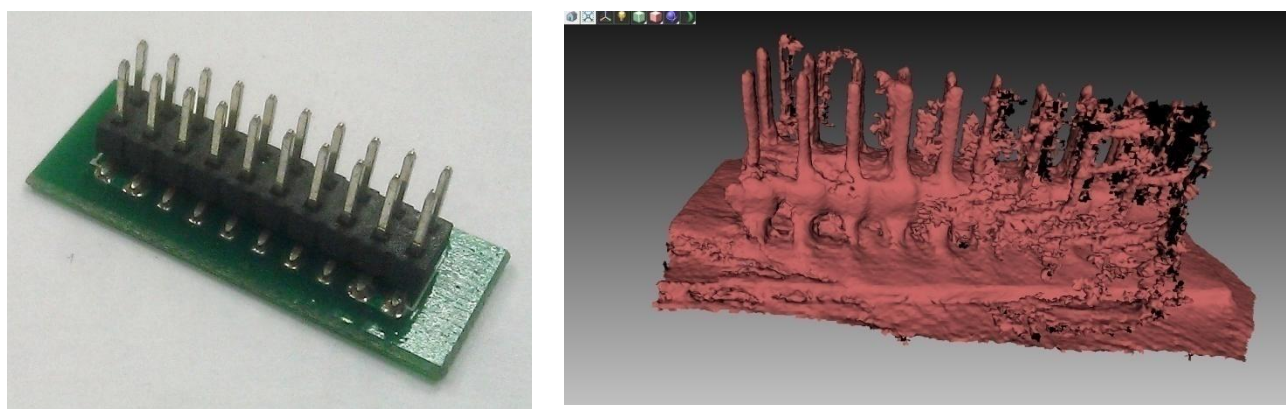


Рисунок 4.12 – Артефакты сканирования, вызванные особенностями формы сканируемого объекта. Слева – фотография объекта, справа – визуализация цифрового скана

Сканирование ряда объектов позволило выявить анизотропию разрешающей способности системы "сканер-ПО" в вертикальном и горизонтальном направлениях (рисунок 4.13). Во втором примере видна попытка ПО компенсировать недостаток точек поверхности, что привело к ухудшению качества цифрового скана.

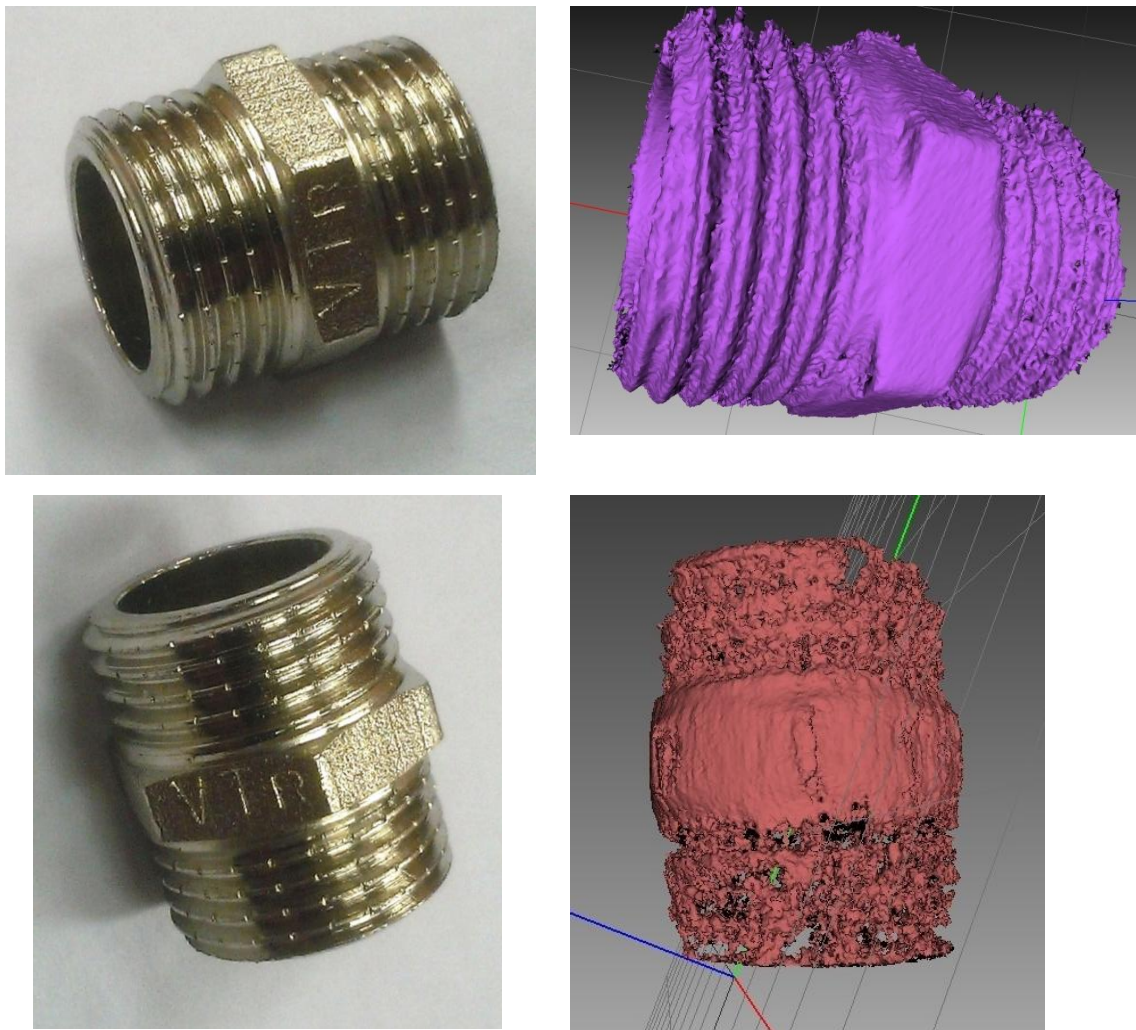


Рисунок 4.13. Влияние анизотропии разрешающей способности системы.

Слева – фотография объекта, справа – визуализация цифрового скана

Система сканирования не позволяет полноценно воспроизводить в цифровом скане перекрывающиеся поверхности объектов, что демонстрируется на рисунке 4.14.



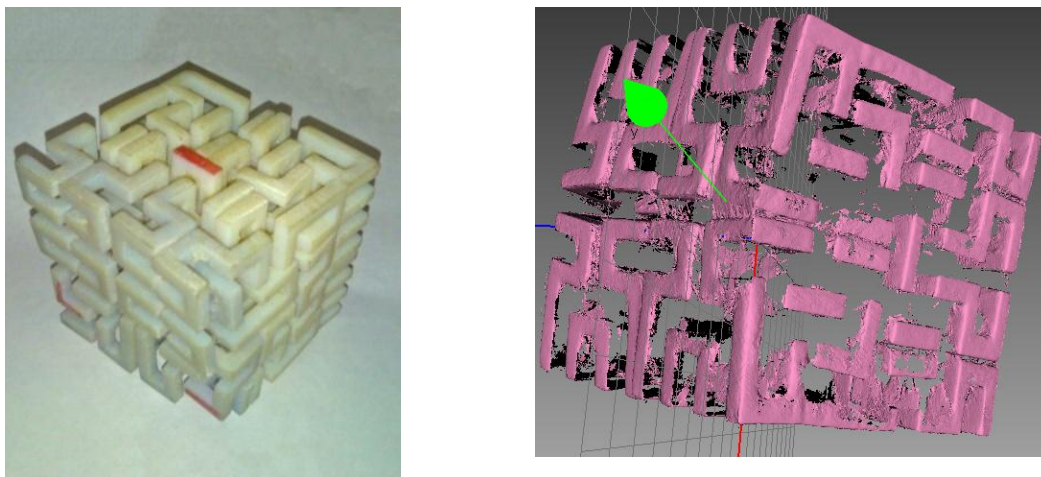


Рисунок 4.14 – Цифровой скан объекта с перекрывающимися поверхностями.

Слева – фотография объекта, справа – визуализация цифрового скана

Таким образом, обобщая результаты проведенных экспериментов, можно сформулировать следующие выводы об особенностях и ограничениях изучаемой системы «3D-сканер–ПОС».

- При недостатках внешнего освещения объекта (световые блики, низкий уровень общей освещенности) проявляются ошибки в работе сканера в виде дефектов поверхности, которые не компенсируются ПОС.
- Отмечен эффект анизотропии разрешающей способности системы «3D-сканер–ПОС» в вертикальном и горизонтальном направлениях.
- В процессе сканирования возможна потеря точности воспроизводства мелких элементов за счет попытки улучшения 3D-скана путем применения сглаживания поверхности.
- Возможно наличие артефактов сканирования, порожденных особенностями формы сканируемого объекта, которые проявляются в связи с невозможностью долговременного отслеживания опорных точек, расположенных на тонких элементах. В этом случае несовершенство ПОС приводит к ошибкам совмещения кадров, проявляющихся в виде многократных повторов мелких фрагментов объекта или всего объекта в

целом. Подобные ошибки возможны и в случаях сканирования симметричных объектов.

- Ошибочное применение сглаживания поверхности в ПОС для некоторых видов поверхностей приводит к отсутствию мелких отверстий.

Следует отметить, что добавление системы сжатия без потерь, на основе предложенного алгоритма не повлияло на характеристики описанных свойств системы «3D-сканер – ПОС».

Кроме того, в результате серии экспериментов по сканированию объектов с различным типом поверхностей, были выявлены некоторые особенности, которые обусловили выбор конкретных значений параметров разработанного алгоритма сжатия (размер элемента разбиения).

В ходе экспериментов удалось определить, что реальная разрешающая способность 3D-сканера Artec составляет 1 мм по вертикальной оси и 0.5 мм по горизонтальной.

#### **4.3 Оценка эффективности метода компрессии**

С учетом реальной разрешающей способности 3D-сканера порядка 1 мм и размера области, попадающей в поле зрения оптической системы равной 250x250x250 мм, при экспериментах достаточно работать с одной пространственной ячейкой размером 256×256×256 точек, так как она полностью перекрывает единовременное поле зрения сканера.

В таблице 4.1 приводятся результаты компрессии ячейки 256×256×256 для разных объектов. Приводится сравнение несжатого битового объема облака точек, заданного в виде координат (RAW-файл), объема RAW-файла, сжатого алгоритмом Deflate, а также результаты применения предложенного алгоритма компрессии с использованием обхода пространства по ЗПК и стандартного построчного преобразования  $R^3 \rightarrow R^1$ .

Таблица 4.1 – Результаты применения метода

Объект	Количество точек в облаке	Исходный формат, байт		Результат сжатия предложенным методом, байт
		Текстовый obj	Бинарный float	
I - коннектор	169	10140	3042	274
Бочонок с резьбой (горизонтально)	609	36540	10962	395
Разъем питания	654	39240	11772	424
Разъем mini Din	1018	61080	18324	512
Светодиодная лампочка	1538	92280	27684	659
Дроссель	1873	112380	33714	758
Штыревой разъем	2018	121080	36324	715
Бочонок с резьбой (вертикально)	4122	247320	74196	916
Уголок с резьбой	5784	347040	104112	1665
Трансформатор	12463	747780	224334	3048

В связи с тем, что облако точек, полученное в результате 3D-сканирования, представляет собой компактные области точек пространства, соответствующие поверхностям сканируемого объекта, то применение ЗПК дает положительный эффект, что подтверждается результатами работы программы, приведенными в таблице 4.1.

На рисунке 4.15 приведена зависимость уровня сжатия (выраженного в битах на точку) от количества точек в исходном объекте для предложенного метода на основе ЗПК (вариант `zpk+rle+zip`) в сравнении с методами сжатия без потерь на основе остовного дерева (метод `Octree`, метод `Spanning tree`) [70–71].

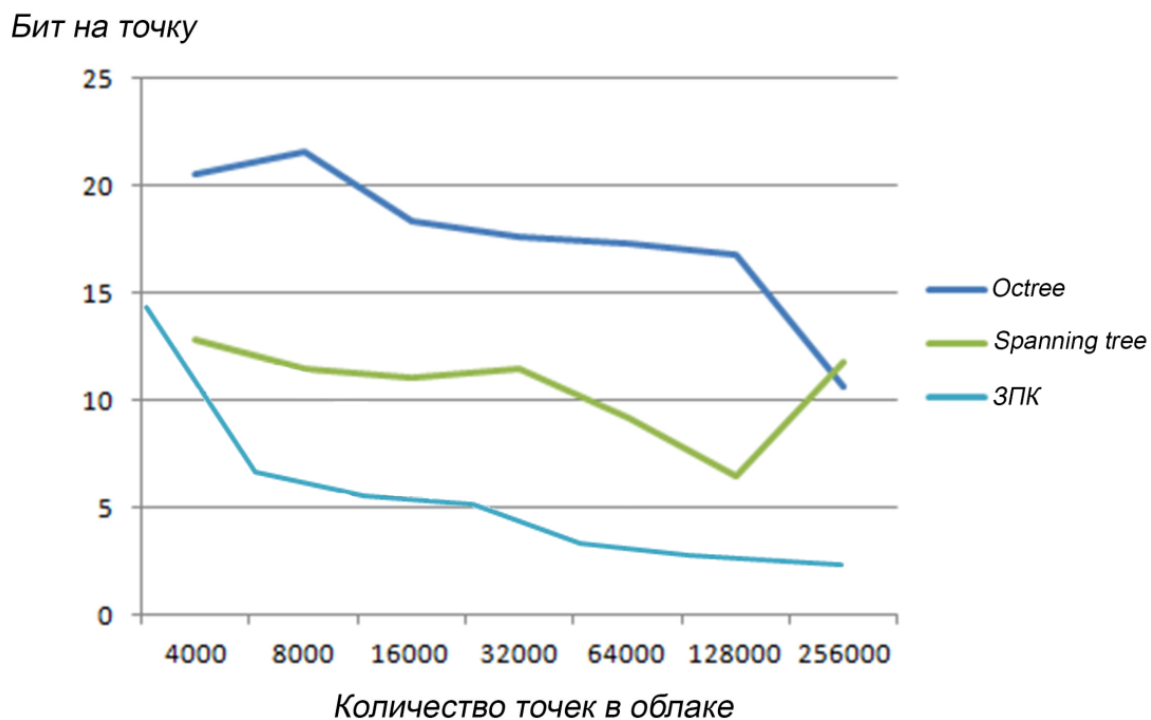


Рисунок 4.15 – Сравнение разработанного метода с существующими методами компрессии

Установлено, что степень сжатия для разработанного метода повышается при увеличении количества точек в облаке, то есть метод более эффективен при повышении точности и разрешающей способности используемых 3D-сканеров.

Предложенный метод может быть использован в программном обеспечении, предназначенном для работы с 3D-сканерами и 3D-принтерами, пакетами пост-обработки сложных пространственных форм и системах когнитивного программирования для повышения эффективности использования систем хранения данных 3D-объектов, создания библиотек объектов и сокращения издержек на связь при передаче 3D-объектов.

#### **Выводы по главе 4**

Установлено, что степень сжатия для разработанного метода повышается при увеличении количества точек в облаке, то есть метод более эффективен при повышении точности и разрешающей способности используемых 3D-сканеров.

Предложенный метод может быть использован в программном обеспечении, предназначенном для работы с 3D-сканерами и 3D-принтерами, пакетами пост-обработки сложных пространственных форм и системах когнитивного программирования для повышения эффективности использования систем хранения данных 3D-объектов, создания библиотек объектов и сокращения издержек на связь при передаче 3D-объектов.

## Заключение

Совокупность разработанных в работе модели, методов и алгоритмов, а также их практическая реализация представляют собой решение актуальной научно-технической задачи представления и сжатия 3D-данных, имеющей большое значение для развития инфокоммуникационных технологий в области компьютерного моделирования и быстрого прототипирования. При этом в ходе решения данной задачи были получены следующие результаты:

1. Проведен анализ современного состояния и особенностей технических систем получения, представления и сжатия пространственных данных.
2. Разработан метод динамического масштабирования пространства облаков точек и их переупорядочения в битовый поток, имеющий упорядоченную структуру, отличающийся способностью сохранять локальные области пространств, что позволяет эффективно применять методы группового кодирования для уменьшения битового объема.
3. Разработана модель представления пространственных объектов, использующая упорядоченное одномерное представление облаков точек на основе заполняющей пространство кривой, и метод динамического разбиения и масштабирования пространства облаков точек, учитывающий специфику технологии бесконтактного оптического трехмерного сканирования.
4. Разработан алгоритм сжатия трехмерных данных, представляющих собой цифровой скан, отличающийся применением заполняющей пространство кривой для переупорядочения облаков точек.
5. Разработана интерактивная система сжатия облаков точек, отличающаяся применением динамического разбиения и масштабирования пространства и заполняющей пространство кривой.

Полученные результаты соответствуют п. 5 «Разработка специального математического и алгоритмического обеспечения систем анализа, оптимизации,

управления, принятия решений и обработки информации», п. 12 «Визуализация, трансформация и анализ информации на основе компьютерных методов обработки информации» паспорта специальности 05.13.01 «Системный анализ, управление и обработка информации (технические системы)».

## Список литературы

- 1 Декарт, Рене. Геометрия: с прил. избр. работ П. Ферма и переписки Декарта /пер., примеч. и ст. А. П. Юшкевича./Р. Декарт. М.; Л.: Гос. объедин. науч.-техн. изд-во НКТП СССР, ред. техн.-теорет. лит., 1938. 288с.
- 2 Монж Г. Начертательная геометрия. Комментарии и редакция Д.И. Каргина. М., Издательство АН СССР, 1947. 256 с.
- 3 Дегтярев В.М. Компьютерная геометрия и графика: учебник для студ. учреждений высш. проф. образования / В. М. Дегтярев. — 2-е изд., стер. — М.: Издательский центр «Академия», 2011. — 192 с.
- 4 Александрова В.В., Симонова И.В., Тарасова О.А. Компьютерное моделирование пространственных форм. В среде – 3D Studio MAX. Спб.: Издательство «Анатолия», 2003 г., 319 с.
- 5 Willard L. Eastman, Abraham Lempel, Jacob Ziv, Martin Cohn. Apparatus and method for compressing data signals and restoring the compressed data signals. U.S. Patent 4 464 650
- 6 Miller, Victor S., Mark N. Wegman. Data compression method. U.S. Patent 4 814 746
- 7 Сэломон Д. Сжатие данных, изображения и звука. — М.: Техносфера, 2004. — 368 с. ISBN 5-94836-027-X.
- 8 Александров В.В., Горский Н.Д. Представление и обработка изображений. Рекурсивный подход. Л., Наука, 1985
- 9 Проекты Дмитрия Ватолина. Электронный ресурс: Доступ — <http://www.compression.ru/dv/>
- 10 Красильников Н. Цифровая обработка 2D- и 3D-изображений. Изд-во: БХВ-Петербург. Серия: Учебная литература для вузов. ISBN 978-5-9775-0700-4. 2011 г.
- 11 Гонсалес Р., Вудс Р. Цифровая обработка изображений. М. Изд-во: Техносфера. Пер. с англ. Л. Рубанов, П. Чочиа. 2012 г. 1104 с.



- 12 Бейтс Р., Мак-Доннелл М. Восстановление и реконструкция изображений. Изд-во: Мир. Пер. с англ.: Б. Кругликов, С. Ярославский, Л. Ярославский. ISBN 5-03-001022-X, 0-19-856176-8. 1989 г.
- 13 Евсютин О., Шелупанов А., Росошек С., Мещеряков Р. Сжатие цифровых изображений. Изд-во: Горячая Линия – Телеком. ISBN 978-5-9912-0357-9. 2013 г.
- 14 Jesse Russell. Сжатие данных. ISBN 978-5-5129-0343-8. 2012 г.
- 15 Кричевский Р. Сжатие и поиск информации. Изд-во: Радио и связь. ISBN 5-256-00325-9; 1989 г.
- 16 Аксенов А.Ю., Александрова В.В., Зайцева А.А. Метод эффективного представления 3D-данных, полученных в результате 3D-сканирования // Информационно-измерительные и управляющие системы, 2014, №6. С. 20-25 (из перечня ВАК)
- 17 Аксенов А.Ю. Метод определения информационной избыточности в аудиоданных на основе использования стандартных психоакустических моделей // Информационно-измерительные и управляющие системы. 2013, т.11, №9. С.39-43 (из перечня ВАК)
- 18 Аксенов А.Ю. Исследование применимости существующих методов сжатия к 3D-видео данным. // Труды СПИИРАН. 2013. Вып. 4(27). С. 73-80 (из перечня ВАК)
- 19 Аксенов А.Ю., Зайцева А.А., Кулешов С.В. О критерии адекватности цифровых трактов передачи данных // Информационно-измерительные и управляющие системы, №7, т.8, 2010. С. 75-77 (из перечня ВАК)
- 20 Аксенов А.Ю., Зайцева А.А. Применение программируемой технологии к обработке сигналов и изображений // Информационно-измерительные и управляющие системы, №11, т.7, 2009. С.63-66. (из перечня ВАК)
- 21 Кулешов С. В., Зайцева А. А., Аксенов А. Ю. Ассоциативно-пирамидальное представление данных // Информационно-измерительные и управляющие системы, №4, т.6, 2008. — с. 14—17. (из перечня ВАК)

- 22 Кулешов С.В., Аксенов А.Ю., Зайцева А.А., Идентификация факта компрессии с потерями в процессе обработки изображений // Труды СПИИРАН. Вып. 5. СПб.: Наука, 2007. (из перечня ВАК)
- 23 Aksenov A., Kuleshov S., Zaytseva A. An application of computer vision systems to solve the problem of unmanned aerial vehicle control // Transport and Telecommunication, 2014, volume 15, no. 3, 209–214 (индексируется в системе Scopus).
- 24 Aksenov A., Kuleshov S., Zaytseva A. Spatiotemporal Video Representation and Compression // Pattern Recognition and Image Analysis, Vol. 23, No. 1, 2013. p.87 (индексируется в системе Scopus).
- 25 Аксенов А.Ю., Александрова В.В., Зайцева А.А. Особенности представления пространственных данных, полученных в результате 3D-сканирования // Материалы конференции «Информационные технологии в управлении» (ИТУ-2014). СПб.: ОАО «Концерн «ЦНИИ «Электроприбор», 2014. С. 440-444.
- 26 Аксенов А.Ю., Зайцева А.А., Кулешов С.В. Критерий  $\epsilon$ -идентифицируемости в обработке аудио и видео данных // Материалы VIII всероссийской научно-практической конференции с международным участием "Современные информационные технологии в науке, образовании и практике". Оренбург, 25-27 ноября 2009.
- 27 Аксенов А.Ю., Макаров А.Н. Цифровая технология анализа и синтеза сигналов. // По пути прогресса – к новым достижениям / ОАО «Научно-производственное предприятие «Радар ммс»/ Сб. материалов под редакцией Генерального директора-Генерального конструктора Г.В.Анцева. СПб.: ООО «Издательство «Логос», 2006. с.188–191.
- 28 Александров В.В., Сарычев В.А. Цифровые программируемые технологии. // «Информационно-измерительные и управляющие системы», №11, т.8, 2010. С. 3–9.

- 29 Александрова В.В., Зайцева А.А. 3D-технология и когнитивное программирование. // Информационно-измерительные и управляющие системы. 2012, т.10, №5. – С. 61–64.
- 30 Александрова В.В., Зайцева А.А., Брысковский Г.В. Цифровые программируемые 3D-технологии // Материалы конференции «Информационные технологии в управлении» (ИТУ–2012) 09–11 октября 2012. С. 547-553
- 31 Александрова В. В., Зайцева А. А., Тыжненко Д. А. Сканирование и редактирование 3D-объекта для прототипирования на 3D-принтере // Информационно-измерительные и управляющие системы. 2013, т.11, №9. — С.53–57.
- 32 Changsoo Je, Sang Wook Lee, and Rae-Hong Park. High-Contrast Color-Stripe Pattern for Rapid Structured-Light Range Imaging. Computer Vision – ECCV 2004, LNCS 3021 (8th European Conference on Computer Vision, Prague, Czech Republic, May 2004, Proceedings, Part I), pp. 95–107, Springer-Verlag Berlin Heidelberg, May 10, 2004
- 33 Лысыч М.Н., Белинченко Р.А., Шкильный А.А. Оборудование и технологии 3D-сканирования // Актуальные направления научных исследований XXI века: теория и практика. 2014. Т. 2. № 4-3 (9-3). С. 210-214.
- 34 Тишкин В.О. Методика сборки и обработки данных, полученных в процессе 3D-сканирования // Научно-технический вестник информационных технологий, механики и оптики. 2011. № 1 (71). С. 87-92.
- 35 Юмашев А.В., Михайлова М.В., Кудерова И.Г., Кристаль Е.А. Варианты использования 3D-сканирования в ортопедической стоматологии. // Вестник новых медицинских технологий. Электронное издание. 2015. № 1. С. 2-6.
- 36 Degtyarev V.M., Gusev M. N. Stereo visualization of model of gas dynamic process in combustion chamber of jet engine // Proc. SPIE 4627, Fifth International Workshop on Nondestructive Testing and Computer Simulations in Science and Engineering, 310

- 37 Degtyarev V.M., Krylov I. P. On classification of cubic and quartic surfaces // Proc. SPIE 5400, Seventh International Workshop on Nondestructive Testing and Computer Simulations in Science and Engineering, 287
- 38 "Географические информационные системы федеральные, региональные, муниципальные. Общие технические требования. ГОСТ Р 52155-2003" (утв. Постановлением Госстандарта РФ от 09.12.2003 N 359-ст)
- 39 Ившин К.С., Башарова А.Ф. Принципы современного трехмерного моделирования в промышленном дизайне. // «Архитектон: известия вузов» №39 Сентябрь 2012. [http://archvuz.ru/2012\\_3/11](http://archvuz.ru/2012_3/11)
- 40 Цапко И.В., Цапко С.Г. Алгоритмы и методы обработки информации в задачах трехмерного сканирования объектов // Известия Томского политехнического университета. 2010. Т. 317. № 5. С. 134-140.
- 41 Самусев С.В., Товмасян М.А., Хлыбов О.С., Дроздов Л.В. Применение фотограмметрии и лазерного 3D-сканирования для измерения профиля инструмента кромкогибочного прессы линии ТЭСА 1420 ОАО «Выксунский металлургический завод» // Производство проката. 2014. № 2. С. 40-42.
- 42 Мельникова О.Г., Олейников П.П. Информационное моделирование зданий: опыт реконструкции памятников культурного наследия // Социология города. 2013. № 4. С. 72-80.
- 43 Гуделайтис А.К., Жукова Л.Т. Изготовление ювелирных изделий методом трехмерного моделирования с использованием современных высокотехнологичных процессов обработки материалов // Дизайн. Материалы. Технология. 2013. Т. 2. № 27. С. 66-72.
- 44 Мотуз В.О., Сарычев Д.С. Применение лазерного сканирования и 3D-моделей в жизненном цикле автомобильных дорог. // САПР и ГИС автомобильных дорог. 2014. № 1 (2). С. 12-15.
- 45 Арсенина О.И., Ряховский А.Н., Сафарова Н.М. Диагностика и планирование ортодонтического лечения пациентов со скученным положением зубов с

- использованием эластомерных корригирующих капп // Стоматология. 2011. Т. 90. № 2. С. 78-80.
- 46 Скворцова А.Ю., Варламов О.О., Сергушин Г.С., Белоусова А.И. Исследование возможностей практического применения технологий виртуальной реальности и угрозы ее развития // Автоматизация и управление в технических системах. 2014. № 2 (10). С. 98-106.
- 47 Борисенко Б., Ярошенко С. 3D-сканирование в интересах 3D-моделирования // Comprice.ru. Электронный ресурс — Доступ: <http://www.comprice.ru/articles/detail.php?ID=40134>
- 48 3D-сканеры Artec. Трехмерное сканирование на примере модели Eva. Электронный ресурс — Доступ: [http://www.ixbt.com/printer/3d/3d\\_scanner\\_eva.shtml](http://www.ixbt.com/printer/3d/3d_scanner_eva.shtml)
- 49 Ричардсон Р. Сканируя пространство. // «Экспресс-Электроника», №10. 2003
- 50 Телемедицина. Новые информационные технологии на пороге XXI века; Р. М. Юсупов, Р. И. Полонников. – СПб.: СПИИРАН, 1998. – 486с. // Физика визуализации изображений в медицине; т. 1 под. ред. С. Уэбба. Москва, «Мир», 1991.
- 51 Марусина М.Я., Казначеева А.О. Современные виды томографии. Учебное пособие. СПб: СПбГУ ИТМО, 2006. – 132 с.
- 52 Smith C. On Vertex-Vertex Meshes and Their Use in Geometric and Biological Modeling // A dissertation submitted to the faculty of graduate studies in partial fulfillment of the requirements for the degree of doctor of philosophy, Calgary, Alberta, 2006. 204 p.
- 53 Шикин Е. В., Боресков А. В. Компьютерная графика. Полигональные модели. М.: ДИАЛОГ-МИФИ, 2000. 464 с.
- 54 Curless, B., Levoy, M., Volumetric Method for Building Complex Models from Range Images. Proc. SIGGRAPH '96
- 55 Turk, G., Levoy, M., Zippered Polygon Meshes from Range Images. Proc. SIGGRAPH '94

- 56 Luebke, David P. A Developer's Survey of Polygonal Simplification Algorithms. IEEE Computer Graphics & Applications, 2001.
- 57 Xia, J.C., El-Sana, J., Varshney, A., Adaptive Real-Time Level-of-Detail- Based Rendering for Polygonal Models. IEEE Trans. on Visualization and Computer Graphics, 1997
- 58 Luebke, D., Hallen, B. Perceptually Driven Interactive Rendering, University of Virginia Tech Report #CS-2001-01
- 59 Laur, D. and Hanrahan, P. Hierarchical Splatting: A Progressive Refinement Algorithm for Volume Rendering. Proc. SIGGRAPH 1991.
- 60 Westover, L. Footprint Evaluation for Volume Rendering. Proc. SIGGRAPH'90.
- 61 Debevec, P., Introduction to Image-Based Modeling, Rendering, and Lighting. SIGGRAPH'2000 courses
- 62 Gortler, S., He, L., Cohen, M., Rendering Layered Depth Images. Microsoft Research, MSTR-TR-97-09
- 63 Levoy, M., Whitted, T. "The Use of Points as a Display Primitive" Technical Report TR 85-022, University of North Carolina at Chapel Hill, 1985.
- 64 Триангуляция Делоне и её применение. // Томск: Изд-во Том.ун-та, 2002. 128 с.
- 65 Челпанов И.Б., Балабан О.М., Аржанухина С.П., Гарибов Р.Б., Кочетков А.В., Янковский Л.В. Задачи, методы и технические средства 3D-моделирования и сканирования в дорожном хозяйстве // Современные наукоемкие технологии. 2012. № 11. С. 37-41
- 66 Петерсон М.В. Кластеризация множества отождествлённых точек на изображениях динамических сцен на основе принципа минимальной длины // Оптический журнал. 2010, №.11, том 77. С.56-62
- 67 Петерсон М.В. Теоретико-информационные критерии и методы оценивания трехмерной структуры сцены и смещений камеры в мобильных системах компьютерного зрения. // Диссертация на соискание ученой степени кандидата физико-математических наук. Санкт-Петербург. 2013. 121 с.

- 68 Gumhold, S., Karni, Z., Isenburg, M., and Seidel, H.-P. 2004. Predictive point-cloud compression. In ACM SIGGRAPH Conference Abstracts and Applications
- 69 Peng, J., and Kuo, C. C. J. 2003. Octree-based progressive geometry encoder. In Internet Multimedia Management Systems IV. Edited by Smith, John R.; Panchanathan, Sethuraman; Zhang, Tong. Proceedings of the SPIE, Volume 5242, pp. 301- 311 (2003).
- 70 Merry B., Marais P., Gain J.: Compression of dense and regular point clouds. In Afrigraph '06: Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa (New York, NY, USA, 2006), ACM Press, pp. 15–20.
- 71 Ruwen Schnabel, Reinhard Klein. Octree-based Point-Cloud Compression. In Eurographics Symposium on Point-Based Graphics (2006)
- 72 Кулешов С.В. Метод 3D-компрессии данных рентгеновской компьютерной томографии // Изв. вузов. Приборостроение. 2011. Т. 54, № 1, с.12–16
- 73 Сушко Д.В., Штарьков Ю.М. О сжатии томографических данных // Информационные процессы. т.8, № 4, 2008. с. 240–255
- 74 Кулешов С.В. Методы и технология построения цифровых программируемых инфокоммуникационных систем. // Дисс. на соискание степени д-ра технических наук. СПб, 2011. 240 с. (Учреждение Российской академии наук Санкт-Петербургский институт информатики и автоматизации РАН)
- 75 Кулешов С.В. Формат представления реальных трехмерных сцен для объемного телевидения (True3D Vision) // Информационно-измерительные и управляющие системы, №4, т.7, 2009. с. 49–52
- 76 Кулешов С.В. Пространственно-временное представление, обработка и компрессия видеопотока. // Информационно-измерительные и управляющие системы, №4, т.6, 2008. с. 33–37.
- 77 Александров В.В., Кулешов С.В., Цветков О.В. Цифровая технология инфокоммуникации. Передача, хранение и семантический анализ текста, звука, видео. // СПб.: Наука. 2008. 244 с.

- 78 Александров В.В., Кулешов С.В. Этерификация и терминальные программы // Информационно-измерительные и управляющие системы, №10, т.6, 2008. С. 50–53.
- 79 Кулешов С.В. Гибридные кодеки и их применение в цифровых программируемых каналах передачи данных. // Информационно-измерительные и управляющие системы. 2012, т.10, №5. С.41–45
- 80 Александров В.В., Лачинов В.М., Поляков А.О. Рекурсивная алгоритмизация кривой, заполняющей многомерный интервал // Известия АН СССР. Техническая кибернетика". 1978. N 1. С. 192–198.
- 81 Александров В.В., Горский Н.Д. Алгоритмы и программы структурного метода обработки данных. Л. : Наука, 1983. 208 с.
- 82 Горский Н.Д., Мысько С.Н., Сухаричев В.П. Сравнительное исследование некоторых характеристик двумерных разверток. Ленинград, 1982, 24 с.
- 83 Александров В.В., Горский Н.Д. Структуризация иерархических систем. В кн.: Алгоритмические модели в автоматизации исследований. М. Наука, 1980.
- 84 Александров В.В., Горский Н.Д., Поляков А.О. Рекурсивные алгоритмы представления и обработки данных. В кн.: Алгоритмы и системы автоматизации исследований и проектирования. М.: Наука, 1980, с.40–78.
- 85 Стронгин Р.Г. Численные методы в многоэкстремальных задачах. У.Наука,1970.
- 86 Математическая энциклопедия. — М.: Советская энциклопедия. И.М. Виноградов 1977—1985.
- 87 Peano G. Sur une courbe qui remplit toute une aire plene. *Mathematiches Annalen*, 36, 1890, p. 157–571.
- 88 Hilbert D. Uber die stetige Abbildung einer Linie ein Flächenstück. — *Math. Ann.*, 1891, p. 38.
- 89 Hilbert Curve Concepts & Implementation. Электронный ресурс — Доступ: [http://www.fundza.com/algorithmic/space\\_filling/hilbert/basics/index.html](http://www.fundza.com/algorithmic/space_filling/hilbert/basics/index.html)



- 90 Дышкант Н.Ф. Эффективные алгоритмы сравнения поверхностей, заданных облаками точек. // Диссертация на соискание ученой степени кандидата физико-математических наук. Москва. 2011. 139 с.
- 91 Кондратюк А.А. К проблеме эффективности компьютерной реализации построения выпуклой оболочки в трехмерном пространстве. // Сборник трудов СПИИРАН. Интеллектуализация автоматизации проектирования. СПб, 1994, С.55
- 92 Ватолин Д., Ратушняк А., Смирнов М. Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео// М. Диалог-МИФИ. 2003. 384 с.
- 93 Witten I., Neal R.M., Cleary G. Arithmetic coding for data compression // Comm. ACM. 1987, V.30, No 6. pp. 520-540
- 94 Huffman, D.A. A method for the construction of minimum redundancy codes. In Proceedings IRE, vol. 40, 1962, pp. 1098-1101.
- 95 RFC 1951: DEFLATE Compressed Data Format Specification version 1.3 // P. Deutsch, Aladdin Enterprises, May 1996.

## Приложение А. Фрагменты программного кода системы для обработки облаков точек

```
public class ZPKgen
{
    //размер ЗПК
    final int sc = 256;
    int n;
    //массив для элементов ЗПК (порядок обхода элементов)
    int q[][][] = new int[sc][sc][sc];

    //шаблоны направлений
    String s[] = {"01234567", "13025746", "32107654", "20316475",
        "15370426", "37261504", "26043715", "04152637",
        "54761032", "75643120", "67452301", "46570213",
        "40625173", "51407362", "73516240", "62734051",
        "10325476", "31207564", "23016745", "02134657",
        "51734062", "73625140", "62407351", "40516273",
        "45670123", "57461302", "76543210", "64752031",
        "04261537", "15043726", "37152604", "26370415"};

    String c="01 004 402 602 267 367 731 531 151 ";

    //рекурсивная функция генерации
    int z(int xc, int yc, int zc, int d, String how)
    {
        if (d == 1)
        {
            q[zc][yc][xc] = n++;
            return 0;
        }

        int d1 = d / 2;
        int ii=0;

        for (int i= 0; i < 32; i++)
            if(s[i].substring(0,2).equals(how))
                {ii=i;break;}
        String c0= s[ii];

        int p=3;
        for (int i = 0; i < 8; i++)
        {
            int c1 = c0.charAt(c.charAt(p)-48)-48;
            p++;

            int x2 = xc + (((c1&1) > 0)?d1:0), y2 = yc +
            (((c1&2)>0)?d1:0), z2 = zc + (((c1&4)>0)?d1:0);
            char real_p[]=new char[2];
            for (int j = 0; j < 2; j++)
            {
                real_p[j]= c0.charAt(c.charAt(p)-48);
                p++;
            }
        }
    }
}
```

```

    }
    z(x2, y2, z2, d1, real_p[0]+""+real_p[1]);
    p++;
  }
  return 0;
}

```

```

//запуск генерации с начальным условием рекурсии
void generate()
{
  n = 0;
  z(0, 0, 0, sc, "01");
}
}

```

**Листинг А.1. Фрагмент класса Java для рекурсивной генерации ЗПК (код оптимизирован для повышения быстродействия)**

```

//чтение файла формата OBJ с облаком точек для одного элемента
разбиения
    BufferedReader f=new BufferedReader(new
InputStreamReader(new FileInputStream(objName)));
    while (f.ready())
    {
        String fn=f.readLine().trim();
        if (fn.length()>0)
            if (fn.startsWith("v "))
            {
                StringTokenizer stringTokenizer=new
StringTokenizer(fn, " ");
                String s1=stringTokenizer.nextToken();
                float i1 = Float.parseFloat(
stringTokenizer.nextToken()*scale;
                float i2 = Float.parseFloat(
stringTokenizer.nextToken()*scale;
                float i3 = Float.parseFloat(
stringTokenizer.nextToken()*scale;

                if ((i1<RANGE+sX) && (i2<RANGE+sY) && (i3<RANGE+sZ)
&& (i1>sX) && (i2>sY) && (i3>sZ))
                {
                    int ii1=Math.round(i1-sX);
                    int ii2=Math.round(i2-sY);
                    int ii3=Math.round(i3-sZ);
                    vcount++;
                }
            }
        //перевод 3D координат точки в индекс в 1D
        int index=zpk.ZPK[ii1][ii2][ii3];
        //установка признака наличия точки
        data[index]=1;
    }
}
}

```

**Листинг А.2. Фрагмент класса Java для преобразования  $R^3 \rightarrow R^1$  с формированием структуры данных, подлежащей дальнейшей компрессии**

## Приложение Б. Акты внедрения

УТВЕРЖДАЮ  
 Заместитель директора  
 по научной работе СПИИРАН  
 д.т.н., профессор  
  
 \_\_\_\_\_ Ронжин А.Л.  
 «10» апреля 2015 года



### Акт

внедрения результатов диссертационной работы  
 Аксенова Алексея Юрьевича

Комиссия в составе: председателя – заведующего лабораторией речевых и многомодальных интерфейсов, доктора технических наук, доцента Карпова Алексея Анатольевича, членов комиссии: старшего научного сотрудника лаборатории автоматизации научных исследований, кандидата технических наук Зайцевой Александры Алексеевны, старшего научного сотрудника лаборатории речевых и многомодальных интерфейсов, кандидата технических наук Кипятковой Ирины Сергеевны, научного сотрудника лаборатории речевых и многомодальных интерфейсов, кандидата технических наук Будкова Виктора Юрьевича, составила настоящий акт, в том, что результаты, полученные в диссертационных исследованиях Аксенова Алексея Юрьевича:

1. алгоритм сжатия трехмерных данных, представляющих собой цифровой скан, отличающийся применением заполняющей пространство кривой для переупорядочения облаков точек;
2. интерактивная программная система для работы с облаками точек, полученными с помощью 3D-сканера, и их обработки

имеют теоретическую значимость и практическую ценность, обладают научной новизной.

Вышеназванные результаты внедрены в деятельность Федерального государственного бюджетного учреждения науки Санкт-Петербургского института информатики и автоматизации Российской академии наук при выполнении НИР «Разработка методологии комплексного мониторинга инфокоммуникационных ресурсов в распределенных сложноорганизованных системах», выполняемой в рамках Программы «Научные основы создания гетерогенных телекоммуникационных и локационных систем и их элементной базы». Данные результаты позволяют сократить затраты на проведение опытно-конструкторских и научно-исследовательских работ, проводимых в организации.

Председатель комиссии:



Карпов Алексей Анатольевич

Члены комиссии



Зайцева Александра Алексеевна



Кипяткова Ирина Сергеевна



Будков Виктор Юрьевич

«10» апреля 2015 года

Федеральная служба по техническому и экспортному контролю  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ УНИТАРНОЕ ПРЕДПРИЯТИЕ**  
**«ГОСУДАРСТВЕННЫЙ НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ**  
**ПРИКЛАДНЫХ ПРОБЛЕМ»**

УТВЕРЖДАЮ

Заместитель генерального  
 директора по научной работе  
 ФГУП «ГосНИИПП»



О.В. Цветков

2015 г.

**А К Т**

о внедрении результатов диссертационных исследований  
**Аксенова Алексея Юрьевича**

Комиссия в составе: председателя – Дегтярева Г.М., д.т.н., ст. научн. сотрудника, и членов комиссии:

- Куценко С.А., начальника лаборатории,
- Колесникова Р.А., заместителя начальника лаборатории,

на основании анализа результатов исследований, представленных в диссертации на соискание ученой степени кандидата технических наук Аксенова А.Ю., материалов отчетов о составных частях НИР, технической и конструкторской документации **установила:**

1. При выполнении научно-исследовательских работ «Сфера», «Трепанг» и опытно-конструкторской работы «Шхера», выполненных по заказам федеральных органов исполнительной власти, ФГУП «ГосНИИПП» были использованы следующие научные результаты, полученные в диссертационном исследовании Аксенова А.Ю:

- модель представления пространственных объектов, использующая упорядоченное одномерное представление облаков точек на основе заполняющей пространство кривой, и метод динамического разбиения и масштабирования пространства облаков точек, учитывающий специфику технологии бесконтактного оптического трехмерного сканирования;
- алгоритм сжатия трехмерных данных, представляющих собой цифровой скан, отличающийся применением заполняющей пространство кривой для переупорядочения облаков точек.

2. Использование перечисленных результатов диссертационного исследования Аксенова А.Ю. при выполнении вышеуказанных НИОКР позволило:

- оптимизировать процессы выполнения операций быстрого прототипирования и создания банка 3D-объектов и 3D-моделей для последующего их использования;
- разработать отдельные компоненты программно-математического обеспечения для сжатия без потерь 3D-объектов, предназначенных для длительного хранения в составе банка 3D-объектов;
- обосновать перспективное направление исследований по совершенствованию методов представления 3D-сканов материальных объектов в процессе опытно-конструкторской деятельности.

Председатель комиссии:

 Г.М. Дегтярев

Члены комиссии:

 С.А. Куценко

 Р.А. Колесников