

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное учреждение науки
«Санкт-Петербургский Федеральный исследовательский центр
Российской академии наук» (СПб ФИЦ РАН)

С.В. Кулешов, А.А. Зайцева, А.Ю. Аксенов,
А.А. Карпов, И.С. Кипяткова, И.В. Ватаманюк

3D-ТЕХНОЛОГИИ В СОВРЕМЕННЫХ
ИНФОРМАЦИОННЫХ СИСТЕМАХ:
ТЕОРИЯ И ПРАКТИКА

Учебно-методическое пособие

Санкт-Петербург

2021

УДК 531.7
ББК 32.813
К42

Рецензенты:

федеральное государственное автономное образовательное учреждение
высшего образования «Санкт-Петербургский политехнический университет
Петра Великого», доцент *В.В. Потехин*

Федеральное государственное бюджетное учреждение науки
«Санкт-Петербургский Федеральный исследовательский центр Российской
академии наук», главный научный сотрудник *И.С. Лебедев*

Утверждено Ученым советом СПб ФИЦ РАН
в качестве учебно-методического пособия

**Кулешов С.В., Зайцева А.А., Аксенов А.Ю., Карпов А.А.,
Кипяткова И.С., Ватаманюк И.В.**

3D-технологии в современных информационных системах: теория и
практика. Учеб.-метод. пособие /СПб ФИЦ РАН. СПб., 2021. 83с.: ил. 61
ISBN 978-5-6047036-1-8

Учебно-методическое пособие дает базовые понятия и знания о
современном быстро развивающемся направлении 3D-сканирования и
3D-печати, которое является частью более сложных задач 3D-
копирования, 3D-архивации и 3D-прототипирования сложных
пространственных объектов.

В пособии приводятся теоретические сведения о способах
представления пространственных объектов, форматах представления и
методах сжатия таких представлений, а также практические
рекомендации по использованию 3D-сканеров и 3D-принтеров.
Материал данного пособия может быть использован при разработке
информационных систем построения цифровых двойников, цифровых
производств в рамках Internet of Things и концепции «Индустрия 4.0».

Предназначено для студентов технических ВУЗов, аспирантов,
обучающихся по направлению «09.06.01 Информатика и
вычислительная техника», а также для всех читателей, интересующихся
современными технологиями, 3D-сканированием и 3D-печатью.

©Авторы, 2021
©СПб ФИЦ РАН, 2021

ISBN 978-5-6047036-1-8

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ.....	5
1 СПОСОБЫ ПРЕДСТАВЛЕНИЯ ЦИФРОВЫХ МНОГОМЕРНЫХ ДАННЫХ	6
1.1 Общее представление о процессе перевода объектов физического мира в цифровой образ	6
1.2 Методы получения цифровых трехмерных данных.....	8
1.3 Способы представления трехмерных объектов.....	13
1.4 Контрольные вопросы.....	23
2 МЕТОДЫ КОМПРЕССИИ (СЖАТИЯ) ТРЕХМЕРНЫХ ДАННЫХ	25
2.1 Методы сжатия данных без потерь.....	26
2.2 Методы сжатия данных с потерями.....	29
2.3 Контрольные вопросы.....	31
3 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ СЖАТИЯ ОБЛАКОВ ТОЧЕК	32
3.1 Виды разверток.....	34
3.2 Заполняющие пространство кривые и дискретные пространства	36
3.3 Динамическое разбиение и масштабирование пространства облаков точек	42
3.4 Модель представления пространственных объектов.....	49
3.5 Алгоритм сжатия пространственных объектов без потерь	50
3.6 Контрольные вопросы.....	52
4 ОСОБЕННОСТИ И ОГРАНИЧЕНИЯ РЕПЛИЦИРОВАНИЯ 3D-ОБЪЕКТОВ.....	54

4.1 Основы практики 3D-сканирования	54
4.2 Основы практики 3D-печати	65
4.3 Контрольные вопросы.....	75
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	77

ПРЕДИСЛОВИЕ

Развитие технологий дало возможность человечеству пройти путь от ремесленного ручного производства материальных ценностей через ряд этапов, включая машинное производство, производство с использованием ЧПУ-оборудования, до программно-определяемого производства. В идеальной модели такого производства любой объект описывается с помощью универсального цифрового представления, которое интерпретируется универсальными производственными машинами, способными воспроизвести любой объект реального мира. На практике все оказывается немного сложнее. На настоящем уровне развития информационных технологий пока не существует единого универсального формата цифрового представления объектов реального мира для последующего воспроизводства и прототипирования. Многообразие видов оборудования для реализации программно-определяемого производства обусловило появление большого разнообразия форматов представления и методов получения исходных данных. При этом развитие цифровых производств в рамках Internet of Things и концепции «Индустрия 4.0» требует появления единых стандартов и подходов к формированию цифрового представления объектов реального мира.

Настоящее учебно-методическое пособие дает базовые понятия и знания о современном быстро развивающемся направлении 3D-сканирования, которое является частью более сложных задач 3D-копирования, 3D-архивации и 3D-прототипирования сложных пространственных объектов, систематизирует подходы к процессам их получения.

В пособии приводятся теоретические сведения о способах представления пространственных объектов, форматах представления и методах их сжатия, а также практические рекомендации по использованию 3D-сканеров и 3D-принтеров. Материал данного пособия может быть использован при разработке информационных систем построения цифровых двойников, цифровых производств в рамках Internet of Things и концепции «Индустрия 4.0».

1 СПОСОБЫ ПРЕДСТАВЛЕНИЯ ЦИФРОВЫХ МНОГОМЕРНЫХ ДАННЫХ

1.1 Общее представление о процессе перевода объектов физического мира в цифровой образ

На текущем этапе развития информационных технологий в области быстрого прототипирования возникает потребность унификации данного процесса, что в свою очередь вносит необходимость формирования новых требований и ограничений, обусловленных выявленными особенностями и спецификой решаемых задач. Для понимания общего направления развития данного процесса опишем примерный жизненный цикл 3D-объекта в задачах репликации/модификации/копирования (рисунок 1.1).



Рисунок 1.1 – Жизненный цикл 3D-объекта

Как указано в [1], типичный жизненный цикл 3D-объекта включает в себя следующие этапы:

- 1) 3D-сканирование исходного материального объекта;
- 2) восстановление (построение) цифрового образа с использованием программного обеспечения (ПО) в автоматическом или 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.

На конечный результат также оказывает влияние человеческий фактор, проявляющийся как на этапе сканирования (если перемещение сканера не роботизировано), так и на этапах работы с цифровым образом.

В тех случаях, когда сканирование производится в «полевых условиях», что часто происходит, например, при работе с археологическими, архитектурными объектами, а также произведениями искусства, требуется сохранять и передавать большие объемы данных в виде 3D-сканов, полученных на 1 и 2 этапах, для того чтобы произвести этап 3 в условиях наличия больших вычислительных мощностей с целью уменьшения влияния человеческого фактора.

В этом случае актуальна задача сжатия (уменьшения объема битового представления) без потерь полученного цифрового образа 3D-объекта перед сохранением или передачей данных между 2 и 3 этапами.

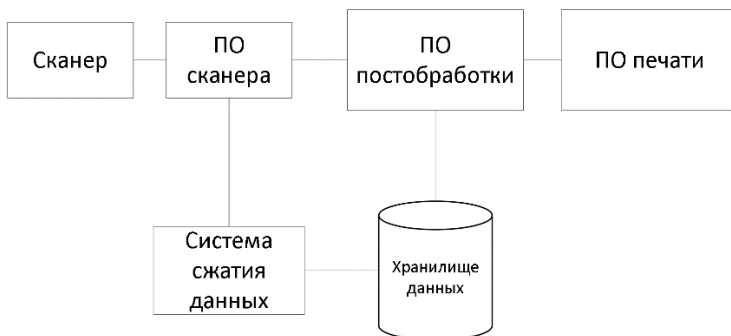


Рисунок 1.2 – Место системы сжатия в процессе 3D-репликации

Возможен также случай, когда этапы 1–3 заменяются одним этапом искусственного синтеза 3D-объекта. Объектом такого синтеза может быть как художественный или технический объект, так и результат визуализации физического или математического моделирования [1]. 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 по совершенствованию методов постобработки, 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 цифровых трехмерных данных

Согласно определению [2]: «Пространственные данные — это цифровые данные о пространственных объектах, включающие сведения об их местоположении, форме и свойствах, представленные в координатно-временной системе...».

Далее мы будем рассматривать данные без временной компоненты.

Методы получения 3D-данных можно разделить на три группы (рисунок 1.3):

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

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 и по элементам построения [3].

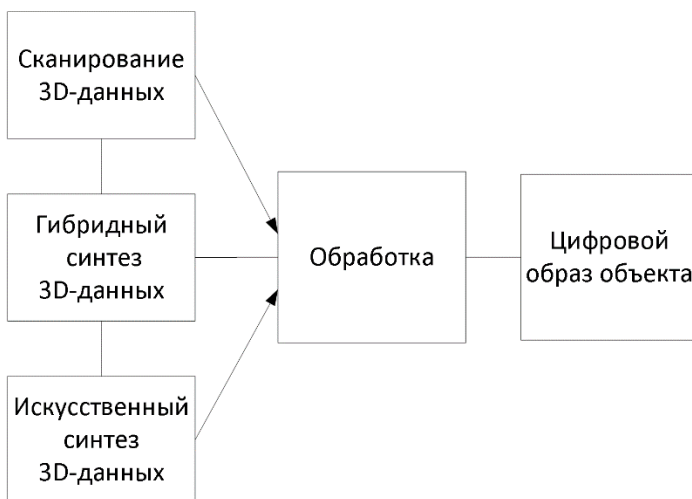


Рисунок 1.3 – Источники получения 3D-данных для 3D-печати

По наличию истории построения объекта 3D-моделирование делится на:

- параметрическое моделирование – по набору заданных варьируемых параметров операций. Примером ПО является САПР;
- непараметрическое моделирование – без сохранения параметров построения (истории построения). Примером ПО, использующего непараметрическое моделирование, является программа Rhino;
- комбинированное моделирование – история построения объектов по умолчанию сохраняется, но ее в любой момент можно отключить (и удалить). Примером ПО с комбинированным моделированием являются программы Alias Studio Tools, Grasshopper, Rhino.

Классификация 3D-моделирования по элементам построения приведена в таблице 1.1.

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 [1].

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.

Таблица 1.1 – Виды 3D-моделирования по элементам построения [3]

№	Виды моделирования	Элементы построения	ПО
1.	Полигональное	Полигоны, кривые	Alias, 3DS Max, Maya, Rhino
2.	Каркасное	Точка и линия	Alias ST, AutoCAD, CATIA, SolidWorks, IsemSurf
3.	Поверхностное	Точка, линия, поверхность	Alias ST, AutoCAD, CATIA, SolidWorks
4.	Твердотельное	Твердое тело (solid)	Alias ST, AutoCAD, CATIA, SolidWorks
5.	Конечно-элементное	Узел, конечный элемент, сетка	Ansys, Rhino+Kangaroo
6.	Генеративное	Компоненты, связи между компонентами	Rhino+Grasshopper

Часто 3D-сканирование используется в целях сопоставления реплицированного объекта с его цифровым образом, что играет особую роль в задачах медицинского протезирования [4, 5].

Основные области применения 3D-сканирования [6–10]:

- 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

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 [11].

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. На базе этих координат строится трехмерная модель сканируемого объекта с точностью 0,05мм.

Бесконтактные 3D-сканеры являются более сложными приборами, основанными на бесконтактных датчиках, работающих на различных физических принципах, и во многом зависящими от примененных алгоритмов определения пространственных координат и восстановления форм объектов.

Как правило, современные сканеры имеют проприетарное закрытое встроенное программное обеспечение и представляют собой систему сканер – программное обеспечение сканера.

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 1.4).

Современные 3D-сканеры являются комбинированными приборами, содержащими несколько типов датчиков. 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 [12].

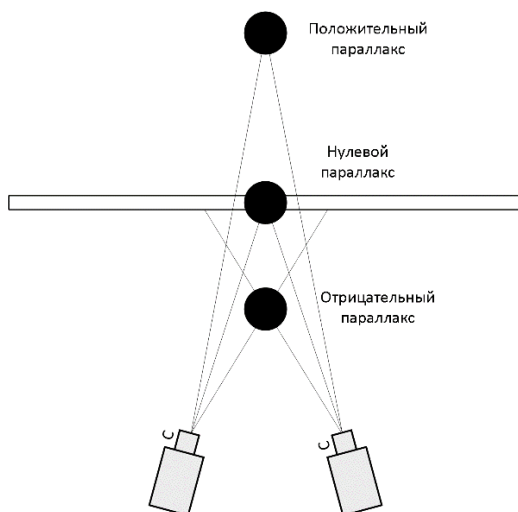


Рисунок 1.4 – Бесконтактный пассивный 3D-сканер

Для ускорения процесса и повышения точности сканирования возможно использование структурированной подсветки, когда источник излучения (проектор) наносит на объект полосу или сетку (рисунок 1.5). Камера, смещенная относительно проектора, воспринимает отражение этой сетки и по обнаруженным искажениям линий вычисляет расстояние до каждой точки в поле зрения. За счет этого достигается и более высокая скорость работы (можно анализировать всё поле зрения сразу), и лучшая точность. Именно такой способ используется в 3D-сканерах Artec [13].

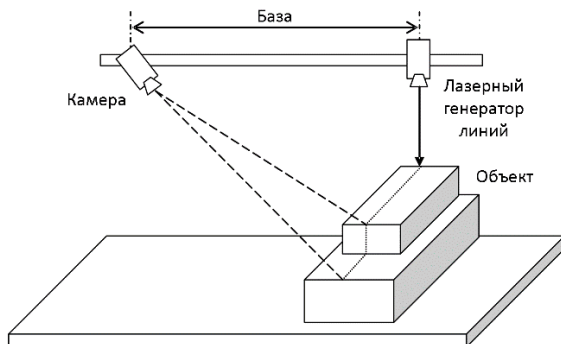


Рисунок 1.5 – Технология активного 3D-сканирования

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 [14].

Компьютерные томографы. Разновидностью технологии получения 3D-модели с внутренней структурой является компьютерная томография [15].

Компьютерная томография (КТ) — метод лучевой диагностики, позволяющий получить послойное изображение любой области человеческого тела или иного физического объекта, материал которого способен ослаблять рентгеновское излучение. Типичное значение толщины среза в популярных томографах находится в пределах от 0,5мм до 10мм.

1.3 Способы представления трехмерных объектов

Способ представления 3D-объекта в цифровом виде (модель представления) определяется структурой данных, то есть форматом ее битового описания. 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 [16].

Проведем сравнительный анализ различных представлений 3D-объектов (рисунок 1.6).

Вершинное представление (рисунок 1.7) описывает объект как множество соединенных между собой вершин, является наиболее простым, информация о гранях и ребрах выражена неявно, поэтому для восстановления исходного объекта требуется обход всех вершин и составление списка граней.

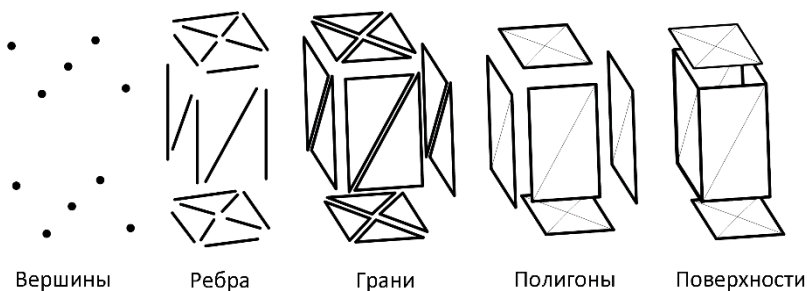


Рисунок 1.6 – Элементы моделирования 3D-объекта

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	0.5,0.5,0	v5 v6 v7 v8
v9	0.5,0.5,1	v0 v1 v2 v3

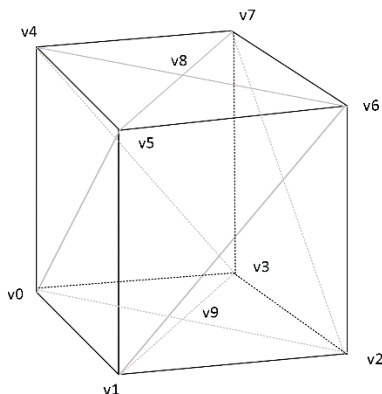


Рисунок 1.7 – Вершинное представление

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

Представление в виде списка граней больше подходит для моделирования, чем вершинное представление, так как позволяет осуществлять прямой поиск вершин и граней, прилегающих к ним. На рисунке 1.8 приведен пример представления параллелепипеда с использованием 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 сопоставимое время. Однако, ребра в таком представлении не заданы явно, поэтому для поиска всех граней, прилегающих к заданной грани, требуется осуществлять дополнительные операции поиска.

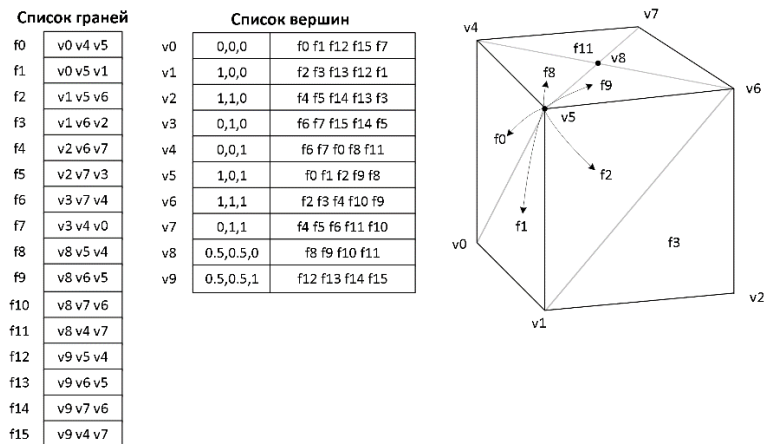


Рисунок 1.8 – Представление в виде списка граней

В задачах, где требуется быстрый доступ ко всем элементам представления трехмерного объекта (вершины, грани, ребра) используется «крылатое» представление, показанное на рисунке 1.9 [17].

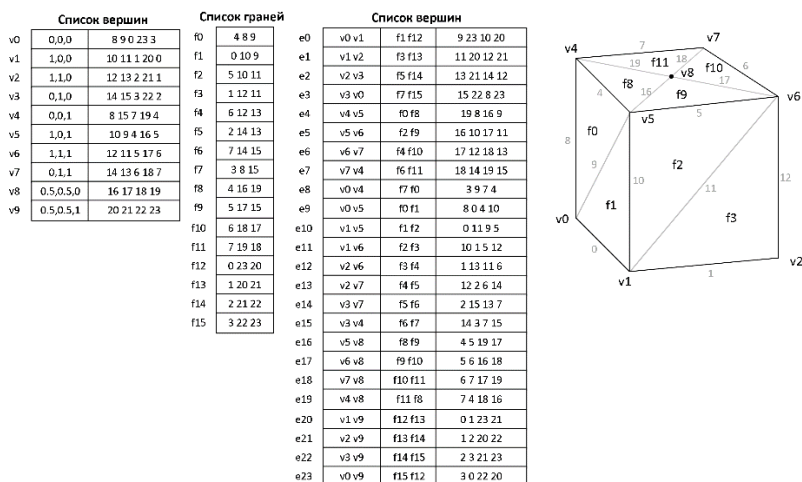


Рисунок 1.9 – Представление в виде списка граней и ребер («крылатое» представление)

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 3D-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 (структура данных типа «облако 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 представления и хранения цифровых сканов, так как в этом случае требуется решение неоднозначной задачи восстановления поверхностей [18–19]. Более того, работа с сетками больших размеров затруднена из-за увеличивающегося объема данных для поддержания связности, что, в свою очередь, требует применения методов упрощения.

Современные методы упрощения сеток в основном используют комбинации описанных ниже подходов [20–22].

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 представления при увеличении количества элементов в модели без упрощения и соответствующего ухудшения качества представления.

Модели на основе вокселей

Воксельные модели представляются в виде регулярных трехмерных массивов, элементам которых сопоставляются характеристики цвета и прозрачности [23–24].

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 [23], 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.

Модели, основанные на картах глубины

Данные модели (англ, Image-Based Modeling and Rendering, IBMR) [25] 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.

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 пространственного положения камеры части объекта.

Использование многослойных изображений с глубиной [26].

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.

Модели на основе точечных представлений (облака точек)

В точечных представлениях объекты представляются как набор трехмерных координат точек, принадлежащих поверхностям исходного объекта (рисунок 1.10).

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 [27].

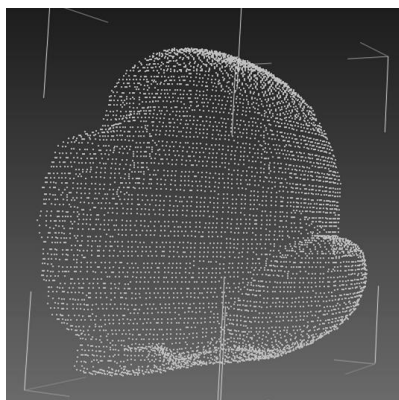


Рисунок 1.10 – Представление трехмерного объекта в виде облака точек

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 или сжатия данных, имеющих высокую вычислительную сложность или требующих дополнительно сформированных структур 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.

Особенности существующих моделей представления пространственных данных обобщены в таблице 1.2.

Каждая из моделей представления ориентирована на свою специализированную область применения, что не позволяет использовать их в качестве универсального представления 3D-объектов. Например, модели на основе полигональных сеток, были специально разработаны для работы в виртуальном пространстве и ориентированы на работу с CAD-подобными средствами проектирования.

Таблица 1.2 – Особенности существующих моделей представления пространственных данных.

Модель представления	Преимущества модели	Недостатки модели
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.

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 [28, 29].

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.

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.

Для задач репликации и хранения объектов 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 [30].

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 [31].

1.4 Контрольные вопросы

- 1) Какие существуют варианты получения 3D-объектов:
 - а) 3D-сканирование;
 - б) синтез в 3D-редакторе;
 - в) фотограмметрия;
 - г) все перечисленные варианты.
- 2) Каркасная модель представления 3D-объектов предполагает использование элементов построения:
 - а) полигон, кривая;
 - б) точка, линия;
 - в) точка, линия, поверхность;
 - г) узел, конечный элементы, сетка.
- 3) Внутреннюю структуру 3D-объекта позволяет получать метод сканирования с использованием:
 - а) компьютерной томографии;
 - б) контактного 3D-сканирования;
 - в) бесконтактного оптического 3D-сканирования.

- 4) Данных с 3D-сканера без применения дополнительных алгоритмов достаточно для построения модели на основе:
- а) полигональных сеток;
 - б) облаков точек;
 - в) списков граней;
 - г) списков граней и ребер.
- 5) Алгоритмы сэмплирования применяются для:
- а) преобразования вершин в грани;
 - б) упрощения геометрии модели;
 - в) повышения качества текстур;
 - г) формирования списков граней.
- 6) Триангуляция Делоне использует базовый элемент в виде:
- а) точки;
 - б) линии;
 - в) треугольника;
 - г) тетраэдра.

2 МЕТОДЫ КОМПРЕССИИ (СЖАТИЯ) ТРЕХМЕРНЫХ ДАННЫХ

Основным недостатком большинства рассмотренных способов представления трехмерных данных является большой объем их битового представления при сохранении файла. 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.

Основные идеи, лежащие в основе существующих подходов и реализаций сжатия пространственных данных с потерями и без потерь

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.

Существующие методы сжатия, работающие непосредственно с облаками точек, используют различные алгоритмы обхода точек пространства принадлежащих сканируемому объекту и дальнейшего представления их в виде древовидный структур (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.

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).

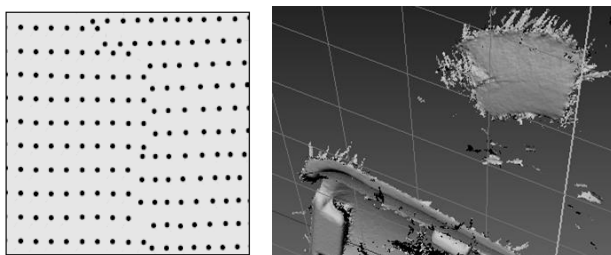


Рисунок 2.1 – Артефакты при автоматической склейке отсканированных поверхностей

2.1 Методы сжатия данных 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) [32–33].

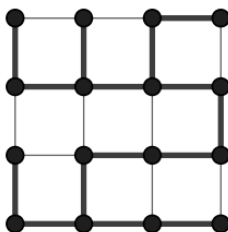


Рисунок 2.2 – Пример остовного дерева

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 сжатию в 5–17 раз, в зависимости от количества точек в облаке. При этом качество сжатия ухудшается с увеличением количества точек [34].

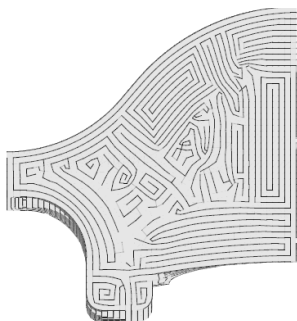


Рисунок 2.3 – Пример описания поверхности объекта при помощи остоного дерева

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 (рисунок 2.4).

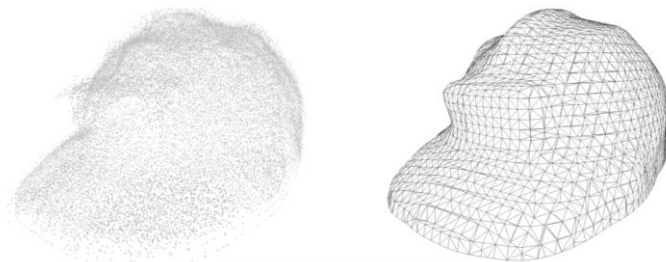


Рисунок 2.4 – Облако точек (слева) преобразованное в полигональную сетку (справа)

Существуют методы сжатия 3D-данных, обеспечивающие сжатие без потерь, которые основаны на представлении точек на поверхности в виде остоного дерева (ациклического связного подграфа для связного неориентированного графа, в который входят все его вершины). Способ построения остоного 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.

Другим способом сжатия являются иерархические представления. Например, существует 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 [35]. Например, для воксельных иерархических моделей используют октодеревья, которые подходят для представления 3D-объектов, заданных на регулярной сетке (рисунок 2.5).

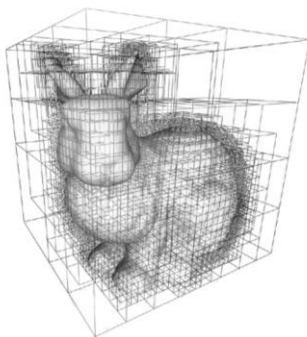


Рисунок 2.5 – 3D-модель объекта, представленная в виде октодерева

Терминальные узлы октодерева представляют собой модель объекта 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 в таком случае возможно при рекурсивном обходе октодерева от корневого узла к

терминальным. Сохранение структуры без части терминальных узлов дает уменьшение точности при соответствующем уменьшением информационной избыточности.

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

2.2 Методы сжатия данных с потерями

В качестве примера сжатия трехмерных данных с потерями можно привести метод, 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.

В ряде известных алгоритмов сжатие производится за счет устранения корреляционной зависимости между последовательными двумерными массивами (слоями). Например, 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).

Алгоритм сжатия данных с потерями имеет сходную структуру, с описанным в [36] и основан на разбиении трехмерной области томографических данных (объединенной

последовательности срезов) на элементы $N \times N \times N$ точек, каждый из которых компрессируется независимо (рисунок 2.6).



Рисунок 2.6 – Последовательность этапов при сжатии с потерями 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 [37].

В качестве алгоритмов нормализации может использоваться ассоциативно-пирамидальная развертка [38], многомерное дискретное косинусное преобразование [37] и ряд других, обладающих указанными свойствами.

Этап квантования повышает степень компрессии в случае, если задача допускает сжатие с потерями [39] (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 [40]).

Значение величины сжатия для такого метода сжатия варьируется от 10 до 120 раз в зависимости от степени квантования (количества вносимых артефактов).

2.3 Контрольные вопросы

- 1) Необходимость использования компрессии данных обусловлена:
 - а) низкой точностью данных;
 - б) большим битовым размером данных;
 - в) высокой скоростью доступа к данным;
 - г) требованиями защиты данных.
- 2) Представление данных в виде остовного дерево является:
 - а) методом повышения точности данных;
 - б) методом сжатия данных с потерями;
 - в) методом сжатия данных без потерь.
- 3) Представление данных, основанное на последовательности плоских срезов является результатом:
 - а) сканирования контактным 3D-сканером;
 - б) восстановления полигональной сетки 3D-модели;
 - в) сканирования 3D-модели компьютерным томографом;
 - г) интерполяции вокселей в 3D-модели.
- 4) На этапе квантования данных осуществляется:
 - а) уточнение данных;
 - б) прогнозирование данных;
 - в) закругление данных;
 - г) квантовые вычисления данных.
- 5) Методы, используемые на этапе нормализации должны обладать свойствами:
 - а) взаимно-однозначного обратного восстановления;
 - б) способность данных к квантованию;
 - в) данные после этого метода более эффективно сжимаются методами вторичного сжатия;
 - г) все перечисленные свойства.
- 6) Алгоритмы Хаффмана и Лемпеля-Зива-Велча могут использоваться на этапах:
 - а) нормализации;
 - б) квантования;
 - в) вторичного сжатия;
 - г) все перечисленные варианты.

3 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ СЖАТИЯ ОБЛАКОВ ТОЧЕК

В главе даются теоретические основы представления данных о 3D-объектах в виде облаков точек, а также способов уменьшения информационной избыточности в таких представлениях. Использование априорных знаний о структуре исходных данных (наличие компактных областей точек пространства, соответствующих поверхностям сканируемого объекта), позволяет применять развертки сложного вида для обхода точек трехмерного пространства.

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

Особенностями массива **М** являются компактные области точек пространства, соответствующие поверхностям сканируемого объекта, соответственно для этапа нормализации целесообразно использовать алгоритм, основанный на переупорядочивании точек, сохраняющем их локальную близость.

Обобщенная схема построения систем сжатия данных, учитывающая структуру большинства методов сжатия, как с потерями, так и без потерь приведена на рисунке 3.1. Путем исключения или замены некоторых из ее элементов можно управлять свойствами алгоритма сжатия [41].

Согласно обобщенной схеме построения систем компрессии данных, чтобы при компрессии многомерных данных обеспечить отсутствие потерь при удовлетворительном уровне сжатия требуется пропустить этап семантической редукции (этап, снижающий точность представления данных за счет закругления или уменьшения разрядности представления значений). В качестве реализации этапа нормализации могут быть использованы разнообразные методы переупорядочения данных.

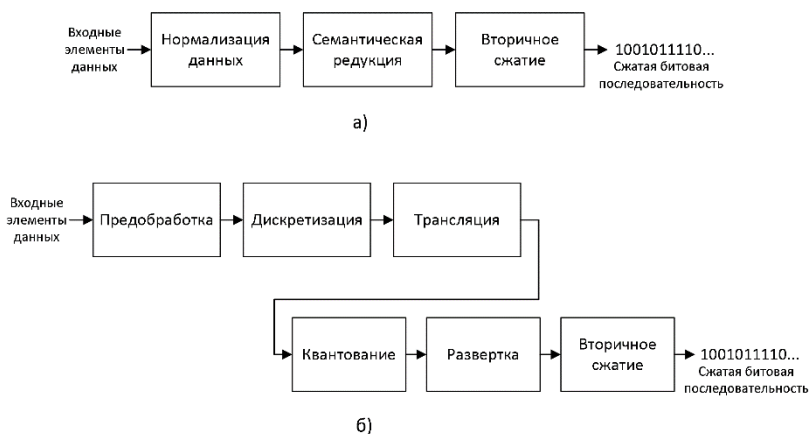


Рисунок 3.1 – Общая схема алгоритма компрессии:
а) обобщенная, б) детализированная

Проблема упорядочения точек многомерного пространства обусловлена принципом построения n -мерного декартова пространства [42], которое определено как прямое произведение: $\mathbf{R}^n = \mathbf{R}_1^1 \times \mathbf{R}_2^1 \times \dots \times \mathbf{R}_i^1 \times \mathbf{R}_n^1$, где однозначно определён порядок только на каждой координате $\mathbf{R}_i^1 \in \mathbf{R}^1$.

Возможность же однозначного упорядочения точек в $\mathbf{R}^n$ 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 $\mathbf{R}^n$ эквивалентна построению отображения $f: \mathbf{R}^n \rightarrow \mathbf{R}^1$, поскольку множества $\mathbf{R}^n$ и $\mathbf{R}^1$ равномощны, но только на $\mathbf{R}^1$ однозначно определен порядок [43].

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

Разные виды отображения $f: \mathbf{R}^n \rightarrow \mathbf{R}^1$ в различной степени сохраняют локальные и глобальные свойства пространства $\mathbf{R}^n$ [42].

3.1 Виды разверток

Рассмотрим наиболее часто применяемые типы разверток, которые можно разбить на три класса [44]:

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.

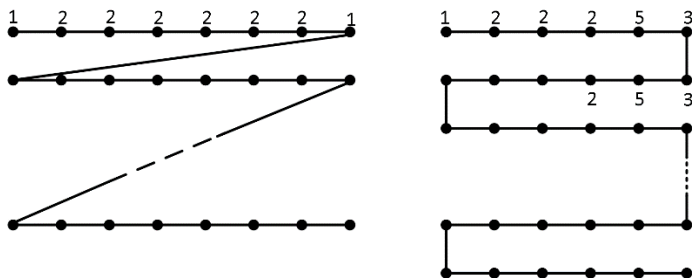


Рисунок 3.2 – Примеры линейных разверток

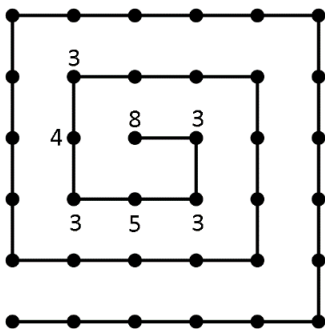


Рисунок 3.3 – Спиральная развертка

2) Рекурсивные развертки (рисунок 3.4) являются приближением к некоторой заполняющей пространство [45–47]. Закон построения этих разверток полностью определяется первым и вторым приближением.

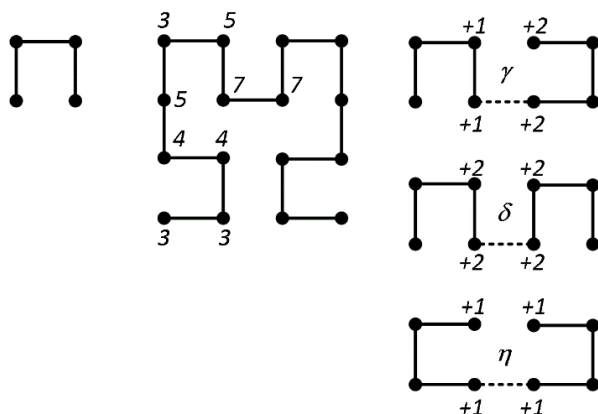


Рисунок 3.4 – Построение рекурсивной развертки

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.

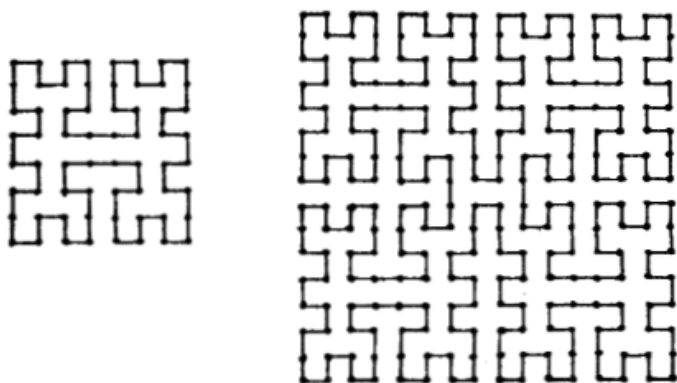


Рисунок 3.5 – Пример блочной развертки замкнутого типа

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 [48]) и глобальные свойства (сохраняющие меру). Они были сконструированы в 1890 г. Дж. Пеано [49]. Впоследствии отображение $f: \mathbf{R}^n \rightarrow \mathbf{R}^1$ 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 $\mathbf{R}^1$, соответствуют точки, близкие в $\mathbf{R}^n$ то есть при $f^{-1}: \mathbf{R}^1 \rightarrow \mathbf{R}^n$ 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 $\mathbf{R}_m^n$ и в $\mathbf{R}_m^1$. При этом свойство сохранения пространственных отношений и условие Гёльдера выполняются с точностью $\varepsilon(m)$, зависящего от номера приближения, условие взаимной однозначности выполняется полностью [43].

Такое представление в дискретном виде обладает наибольшей эффективностью для одномерного представления многомерных множеств.

3.2 Заполняющие пространство кривые и дискретные пространства

Каждая ЗПК представлена в виде непрерывной, нигде не дифференцируемой кривой, проходящей через все точки $\mathbf{R}^n$ (единичного гиперкуба). Алгоритм построения ЗПК для случая $n=2$ впервые был дан в геометрической форме в работе [50].

Собственно ЗПК изобразить нельзя, возможно лишь как это сделано [50], 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 $\mathbf{R}^n$, то задание ее эквивалентно заданию соответствия между точками единичного отрезка $\mathbf{R}^1$ и единичного гиперкуба $\mathbf{R}^n$, и вопрос состоит в ее алгоритмическом описании.

Отметим, что при решении практических задач никогда не требуется знать точного соответствия координат точки в $\mathbf{R}^n$ и ее координат в $\mathbf{R}^1$. 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 $\mathbf{R}^n$ и $\mathbf{R}^1$ применять не собственно ЗПК, а лишь ее соответствующее приближение, что значительно упрощает решение практических задач, а число отличных друг от друга элементов пространства $\mathbf{R}^n$ становится конечным. Такой квантованный единичный гиперкуб $\mathbf{R}^n$ будем называть дискретным пространством.

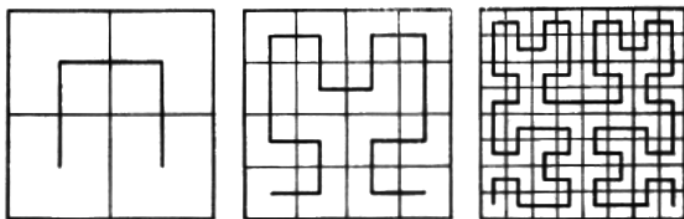


Рисунок 3.6 – Первое, второе и третье приближение ЗПК

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.

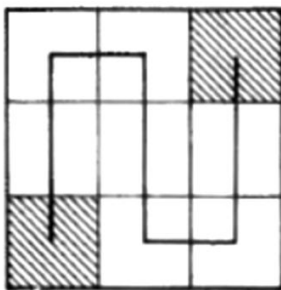


Рисунок 3.7 – Пример эталонной ячейки

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 описано семейство эталонных

ячеек для $k=2$ и произвольной размерности пространства n , которое получается из очень простой общей формулы. Назовем его α -семейством.

Ячейки α -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 $\mathbf{R}^n$, $n=2,3,4$, опущены. Если записать последовательности, соответствующие эталонным ячейкам этого семейства, то получим

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

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

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

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

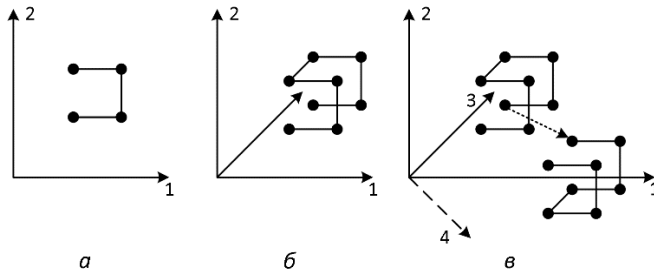


Рисунок 3.8 – Эталонные ячейки α -семейства

Формула для g -го члена последовательности (2), обозначенной $F(g)$, $g = 0, 2^n - 1$ при разложении (2) по системе функций следующего специального вида:

$$\begin{aligned} \psi_1(g) &= 1, 0, -1, 0, 1, 0, -1, \dots \\ \psi_2(g) &= 0, 1, 0, 0, 0, -1, 0, 0, 0, 1, 0, \dots \\ &\dots \\ \psi_i(g) &= 0 \dots 0, 1, 0 \dots 0, 1, 0 \dots 0, 1, \dots \\ &\quad 2^{j-1}-1 \quad 2^{j-1}-1 \quad 2^{j-1}-1 \end{aligned}$$

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

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

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

$$\psi_j(g) = \frac{1}{2} \left((-1)^{\text{entier}\left(\frac{g}{2^j} + 0.5\right)} - (-1)^{\text{entier}\left(\frac{(g+1)}{2^j} + 0.5\right)} \right), \quad (4)$$

$$j = \overline{1, n}$$

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

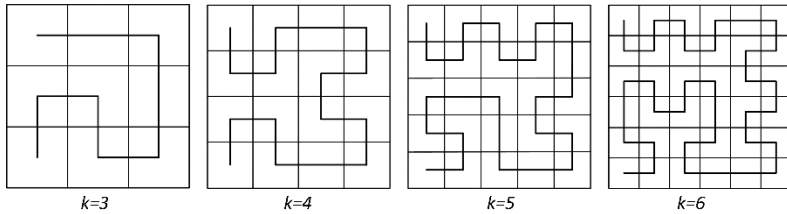


Рисунок 3.9 – Пример семейства эталонных ячеек

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):

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

$n = 3$ 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;

$n = 4$ 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, 4, -3, -4, -4.

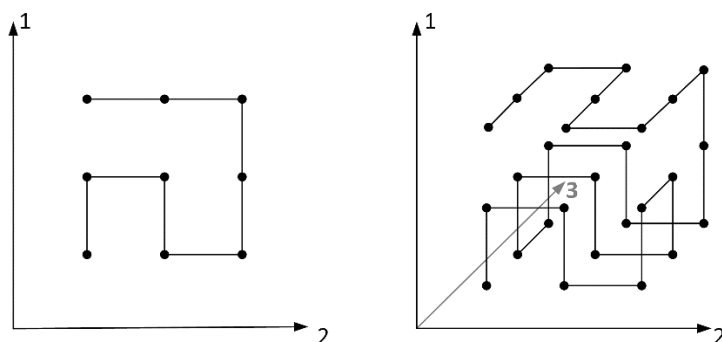


Рисунок 3.10 – Пример семейства эталонных ячеек

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 α -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 [51], 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);

```

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 требуемого 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 [52].

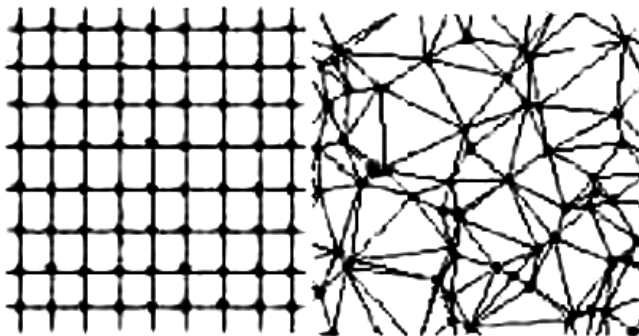


Рисунок 3.11 – Сетки с регулярной (слева) и нерегулярной структурой (справа)

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 [30].

При использовании ЗПК для успешного решения задачи взаимно-однозначного соответствия координат точки в пространстве $\mathbf{R}^3$, представленной в дискретном виде с конечным числом десятичных знаков, координатам пространства $\mathbf{R}^1$ требуется производить динамическое разбиение и масштабирование пространства исходных облаков точек пространства $\mathbf{R}^3$.

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).

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

$$x_0 = \min_i x_i$$

$$y_0 = \min_i y_i$$

$$z_0 = \min_i z_i$$

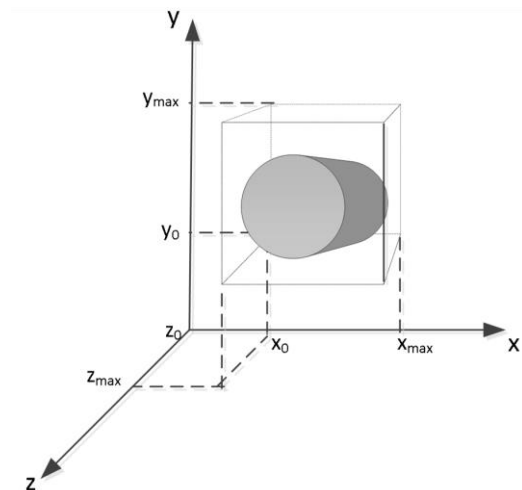


Рисунок 3.12 – Область пространства (параллелепипед), ограничивающая облако точек

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 [53].

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 (рисунок 3.14).

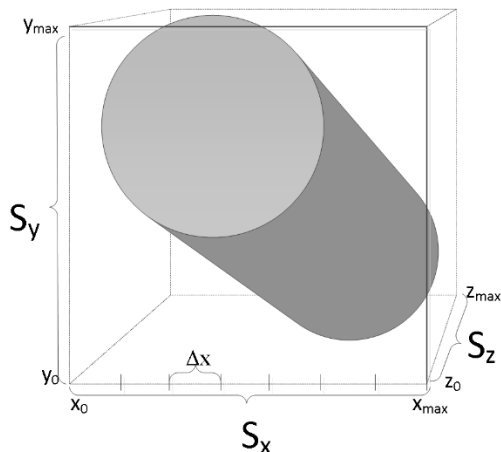


Рисунок 3.13 – 3D-объект в системе координат ограничивающего параллелепипеда

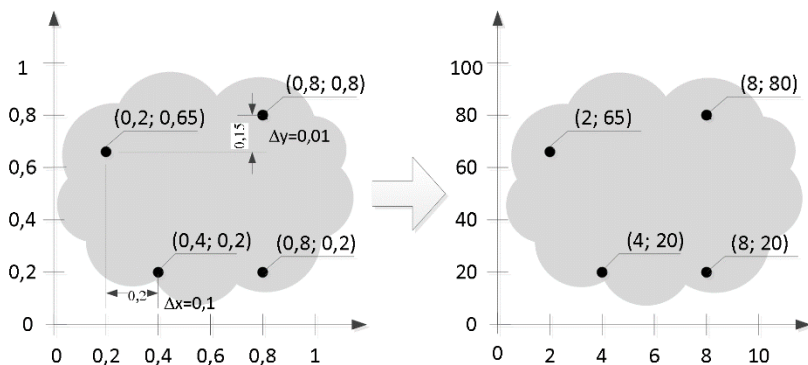


Рисунок 3.14 – Иллюстрация преобразования координат в облаке точек

В общем случае, шаг квантования Δx 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.

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 3.14).

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, 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:

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

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

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

где S_x, S_y, S_z – размер параллелепипеда, ограничивающего 3D-объект по осям, sx, sy, sz – линейные размеры элементов

разбиения, а $[x] = \min\{n \in \mathbb{Z} \mid n \geq x\}$ — операция округления к большему. Тогда $N = N_x \times N_y \times N_z$ — общее количество элементов разбиения для 3D-объекта (рисунок 3.15).

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:

$$\begin{cases} x' = k_x(x - x_0) - i \times sx \\ y' = k_y(y - y_0) - i \times sy \\ z' = k_z(z - z_0) - i \times sz \end{cases}$$

Где i, j, l — позиция текущего элемента разбиения 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.

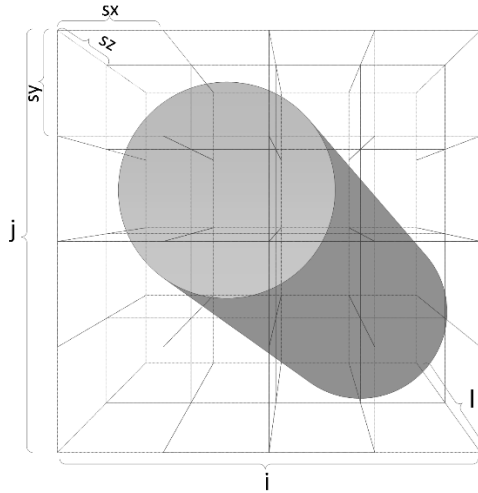


Рисунок 3.15 – К понятию динамического разбиения и масштабирования пространства облаков точек

3.4 Модель 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 [54]. При увеличении объема исходных данных такая модель требует коррекции путем задания порядка перечисления элементов представления для снижения информационной избыточности.

Для решения 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 $x', y', z' \in \mathbb{Z}$, $p'(x', y', z') \in S_n$.

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 Алгоритм сжатия пространственных объектов без потерь

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 3.16).

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 сжат [4].

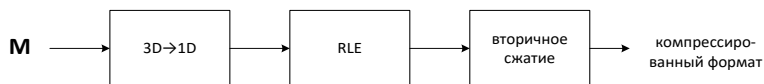


Рисунок 3.16 – Общая схема алгоритма сжатия

Шаг 1 алгоритма: преобразование 3D→1D

Основным этапом сжатия является преобразование трехмерного массива **М** в линейную бинарную последовательность (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 3.17.

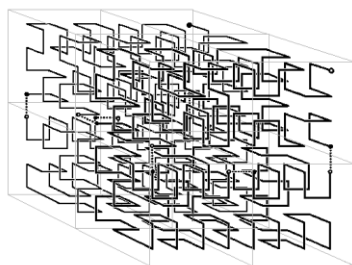


Рисунок 3.17 – Заполняющая пространство кривая для трехмерного случая с размерностью $8 \times 8 \times 8$

Шаг 2 алгоритма: групповое кодирование

После преобразования массива **М** 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 [55]. При этом формируются пары вида <пропустить, число>, 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 [56].

В предложенном методе используется модифицированный (семанτικο-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, 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 [57], сжатие с предсказанием по частичному совпадению (PPM) [55], группа алгоритмов сжатия LZxx [55], сжатие Хаффмана [58] и ряд других, что позволяет уменьшить среднюю длину кодового слова для символов алфавита.

3.6 Контрольные вопросы

- 1) На этапе дискретизации данных осуществляется:
 - а) дублирование данных;
 - б) разбиение данных на блоки;
 - в) закругление данных;
 - г) дискретные вычисления данных.
- 2) Задача упорядочения точек R^n эквивалентна:
 - а) построению отображения $f: R^1 \rightarrow R^n$;
 - б) построению отображения $f: R^n \rightarrow R^1$;
 - в) построению отображения $f: R^n \rightarrow R^n$.
- 3) Кривая Пеано может определять развертку следующего вида:
 - а) спиральную развертку;
 - б) линейную развертку;
 - в) блочную развертку;
 - г) все перечисленные виды разверток.
- 4) Потеря точности представления данных при использовании заполняющей пространству кривой обусловлена:

- а) отбрасыванием незначащих разрядов в числах;
 - б) округлением координат точек по опорным узлам сетки;
 - в) округлением номеров отрезков в кривой;
 - г) дискретизацией элементов кривой.
- 5) Свойство нерегулярной сетки:
- а) регулярная упорядоченная структура отсутствует;
 - б) нет двух одинаковых элементов в сетке;
 - в) не используется система координат;
 - г) шаг квантования меняется от элемента к элементу.
- 6) Модель облака точек описывает:
- а) структурированный набор точек;
 - б) набор областей связности точек;
 - в) неструктурированный набор точек;
 - г) структуру граней в объекте.

4 ОСОБЕННОСТИ И ОГРАНИЧЕНИЯ РЕПЛИЦИРОВАНИЯ 3D-ОБЪЕКТОВ

4.1 Основы практики 3D-сканирования

Процедура сканирования состоит из трех этапов: этапа съемки, этапа обработки и, по завершении, экспорта. В результате будет сравнительно быстро получена качественная трехмерная модель объекта.

В качестве основного программного обеспечения для обработки облака точек в рамках данного пособия рассматривается поставляемая производителем сканеров Artec программа Artec Studio. Данное ПО позволяет сканировать и получать 3D-модели, используя как сканеры Artec, так и устройства сторонних производителей (Microsoft Kinect, Intel RealSense и PrimeSense Carmine и др.) [59].

На рисунке 4.1 приведен общий вид интерфейса. Все сканируемые данные отображаются в окне 3D вида (3D-view) [59].

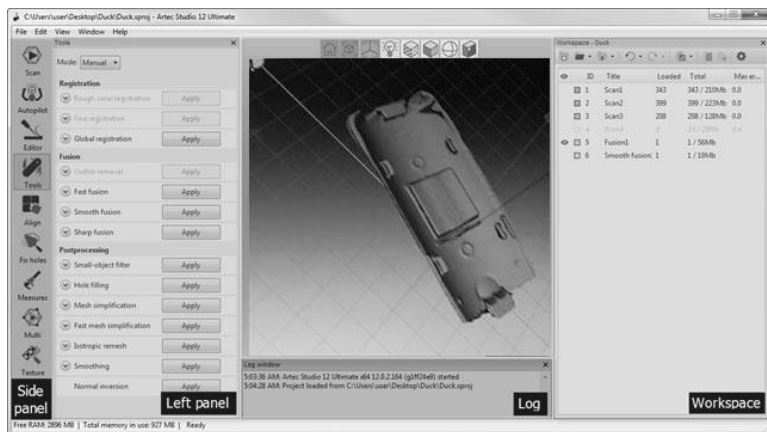


Рисунок 4.1 – Главное окно Artec Studio

Слева от окна 3D вида (Side Panel) находится боковая панель с иконками различных режимов работы приложения, таких как: Съемка, Автопилот, Редактор, Команды, Сборка, Склейка, Измерения, Текстура, Viewshape. В каждый момент времени приложение может работать только в одном из этих режимов, за

исключением Автопилота. Основными режимами работы являются: Съемка, Автопилот, Редактор, Команды и Сборка.

В верхней части окна 3D вида расположена контекстнозависимая панель 3D-инструментов, включающая следующие команды: Исходная позиция, Масштабировать вид, Сетка, Освещение, Цвет, Режим отрисовки, Обратная сторона. В режиме «Редактор» появляются дополнительные инструменты: Сквозное выделение, Позиционирование, Сглаживающая кисть, Переместить/вращать/масштабировать, Ластик, Кисть удаления деталей и Кисть восстановления текстуры.

Панель «Рабочая область» используется для отображения и управления всеми данными, загруженными в приложение. Здесь отображаются сканы, а также команды работы с проектами, такие как сохранение, удаление, перемещение, переименование.

В нижней части окна приложения расположена панель «Журнал». Журнал представляет собой отчет программы о выполненных командах с указанием времени выполнения операции и дополнительных сведений. В журнал выводятся сообщения об ошибках и диагностические сообщения, выдаваемые алгоритмами обработки.

В таблице 4.1 приведены этапы съемки в соответствии с инструкцией [59].

Таблица 4.1 – Этапы съемки

1. Start Для начала процедуры сканирования необходимо нацелить 3D сканер Artec на нужный объект. После этого нажать кнопку начала работы для запуска процедуры сканирования. Для обеспечения правильного сканирования при возникновении каких-либо затруднений или ошибок система автоматически подает визуальные и звуковые сигналы.	
--	---

2. Движение сканера вокруг объекта

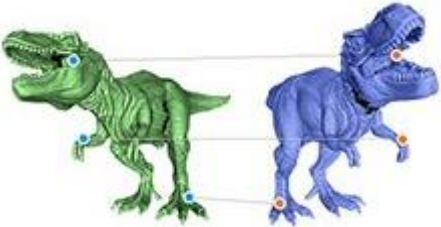
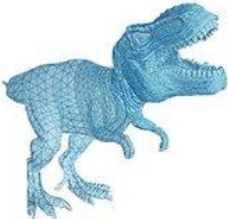
Правильная процедура сканирования предполагает перемещение 3D сканера Artec вокруг всего объекта. Система запускает процесс выравнивания сканируемой поверхности в режиме реального времени, что позволяет запустить сканирование точно и правильно и определить необходимый объем работы. При этом пользователь может выполнять сканирование в любой удобной последовательности. Во время сканирования рекомендуется больше внимания обращать на изображение объекта на экране, чем на реальный объект.



3. Получение сканов

Пользователь может сделать любое нужное число сканов, чтобы обеспечить полный захват сканируемого объекта. Если необходимо, можно сначала отсканировать одну сторону объекта, затем остановить сканирование с помощью кнопки Stop, а затем, повернув объект, завершить сканирование.



4. Процедура совмещения сканов Для получения единой модели система должна выровнять все отсканированные изображения, а если съемка не выполнена до конца, то необходимо выполнить повторную процедуру сканирования нужного участка.	
5. Процедура объединения в 3D-модель Целую 3D-модель можно получить после совмещения всех сканов с помощью специального алгоритма совмещения.	
6. Процедура сглаживания и оптимизации сетки ПО сканера позволяет производить оптимизацию сетки, сглаживание поверхности и заполнение полостей.	
7. Процедура добавления текстуры Наложение текстуры на объект производится автоматически.	
8. Процедура экспорта результатов ПО позволяет экспортировать результаты сканирования в самые распространенные 3D-форматы: ASCII, AOP, STL, PLY, VRML, OBJ.	

Для проведения экспериментов по определению возможностей репликации 3D-сканирования используются следующие виды данных:

1) полученные с помощью 3D-сканера Artec Spider облака точек (пример на рисунке 4.2). Параметры сканирования: процесс сканирования производится в «реальном времени». Облако точек получено за один проход сканирования;

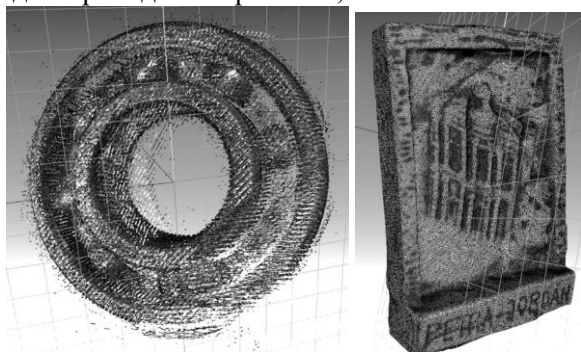


Рисунок 4.2 – Исходные облака точек с 3D-сканера Artec Spider

2) искусственно синтезированные в программах 3DS Max, AutoCAD (рисунок 4.3);

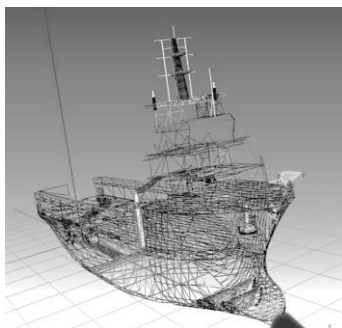


Рисунок 4.3 – Искусственно синтезированный 3D-объект

3) стандартные коллекции 3D-объектов из Интернет-источников (пример на рисунке 4.4).

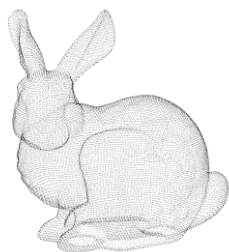
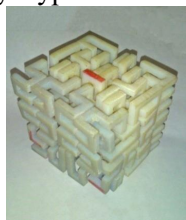


Рисунок 4.4 – 3D-объект из стандартных коллекций

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.

Виды объектов для сканирования [1]

Исходные объекты 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)



Ограничения 3D-сканирования

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 (ПО), осуществляющее реконструкцию облака точек из набора изображений, полученных с нескольких камер 3D-сканера, а также их постобработку имеет смысл рассматривать как единую техническую систему. Влияние ПО проявляется в первую очередь в изменении (часто искажении) формы объекта (рисунок 4.5) при попытке компенсировать ошибки в работе сканера.

Часто попытка ПО 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 4.6) за счет попытки улучшения путем сглаживания поверхности ПО сканера.

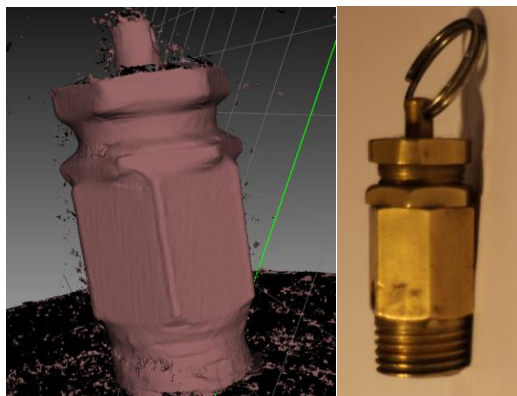
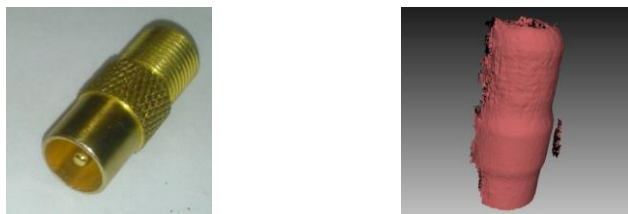


Рисунок 4.5 – Влияние особенностей технической системы 3D-сканер+ПО на изменение формы объекта (слева – отсканированный объект, справа – фотография исходного объекта)



*Рисунок 4.6 – 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 4.7 приведен пример физического объекта, для которого не удалось сразу получить хороший 3D-скан из-за особенностей сканируемой поверхности.



Рисунок 4.7 – Невозможность получения цифрового скана из-за особенностей сканированной поверхности

На некоторых сканах фиксируется отсутствие мелких отверстий, вызванное ошибочным применением сглаживания поверхности в ПО (рисунок 4.8).

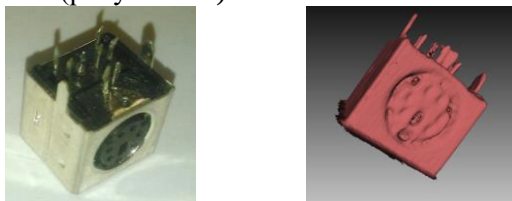


Рисунок 4.8 – Отсутствие мелких отверстий.

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 4.9.



Рисунок 4.9 – Артефакты сканирования.

Слева – фотография объекта, справа – визуализация цифрового скана

На рисунке 4.10 приведен цифровой скан, на котором при хорошем качестве 3D-сканирования элементов большого размера и малой детализации присутствует множество артефактов сканирования на тонких элементах конструкции.

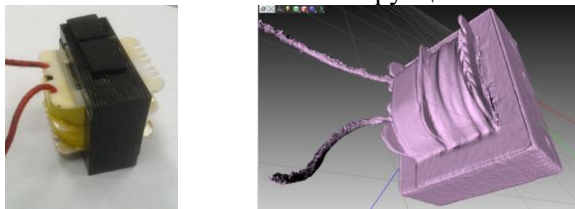


Рисунок 4.10 – Артефакты сканирования на тонких элементах конструкции. Слева – фотография объекта, справа – визуализация цифрового скана

Отсутствие мелких деталей и материал объекта, не дающий световых бликов, повышают вероятность получения высокого качества цифрового скана с хорошей детализацией (рисунок 4.11).

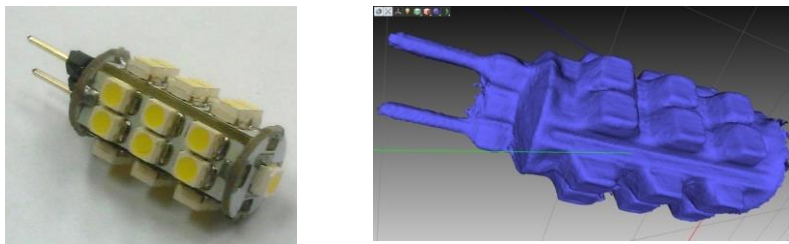


Рисунок 4.11 – Влияние материала поверхности объекта на качества 3D-скана. Слева – фотография объекта, справа – визуализация цифрового скана

На рисунке 4.12 приведен пример цифрового скана с дефектами поверхности (отсутствие точек), вызванными световыми бликами.

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 4.13).



Рисунок 4.12 – Дефекты поверхности вызванные световыми бликами. Слева – фотография объекта, справа – визуализация цифрового скана

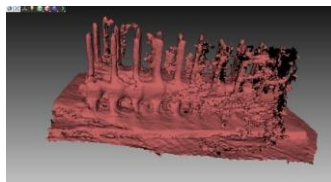


Рисунок 4.13 – Артефакты сканирования, вызванные особенностями формы сканируемого объекта. Слева – фотография объекта, справа – визуализация цифрового скана

Сканирование ряда объектов позволило выявить анизотропию разрешающей способности 3D-сканера в вертикальном и горизонтальном направлениях (рисунок 4.14). Во втором примере видна попытка ПО компенсировать недостаток точек поверхности, что привело к ухудшению качества цифрового скана.

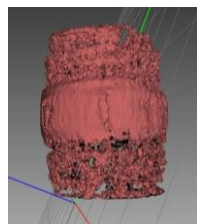
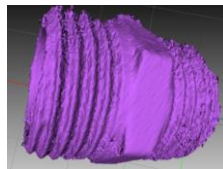


Рисунок 4.14 – Влияние анизотропии разрешающей способности системы. Слева – фотография объекта, справа – визуализация цифрового скана

Система сканирования не позволяет полноценно воспроизводить в цифровом скане перекрывающиеся поверхности объектов, что демонстрируется на рисунке 4.15.

Таким образом, при недостатках внешнего освещения объекта (световые блики, низкий уровень общей освещенности) проявляются ошибки в работе сканера в виде дефектов поверхности, которые не компенсируются ПО.

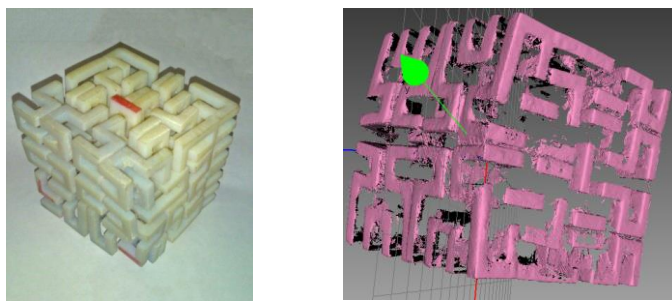


Рисунок 4.15 – Цифровой скан объекта с перекрывающимися поверхностями. Слева – фотография объекта, справа – визуализация цифрового скана

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 ПО 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 ПО для некоторых видов поверхностей приводит к отсутствию мелких отверстий.

Реальная разрешающая способность 3D-сканера Artec составляет 1 мм по вертикальной оси и 0.5 мм по горизонтальной.

С развитием программ сканирования многие из описанных ограничений постепенно снимаются.

4.2 Основы практики 3D-печати

3D-печать – это финальный этап в процессе репликации 3D-объекта. Собственно 3D-печать производится специализированным оборудованием на основе числового программного управления (ЧПУ), упрощенная схема которого приведена на рисунке 4.16.

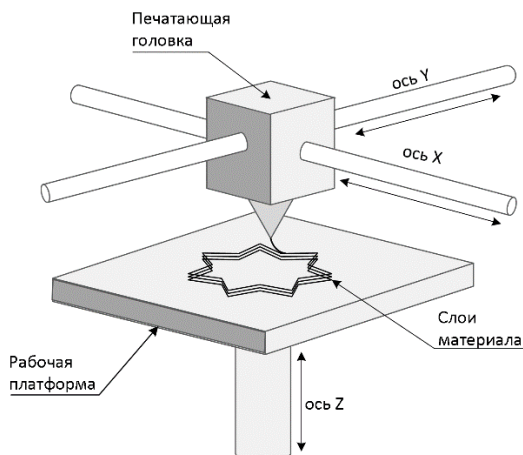


Рисунок 4.16 – Схема работы 3D-принтера с подвижной печатающей головкой (слева-направо, взад-вперед) [60]

Для формирования управляющего кода для 3D-принтера применяется программа-слайсер [60]. По определению, слайсер — это программное обеспечение для 3D-печати, которое преобразует цифровые 3D-модели в инструкции по печати для 3D-принтера. Слайсер «разрезает» 3D-модель на горизонтальные слои на основе выбранных настроек и вычисляет, сколько материала и времени потребуется принтеру для экструзии. Вся эта информация затем сохраняется в файл в формате GCode, который отправляется на 3D-принтер. По сути программное обеспечение для 3D-печати выступает посредником между 3D-моделью и 3D-принтером (рисунки 4.17, 4.18).

GCode — разновидность языков программирования устройств с числовым программным управлением (ЧПУ), содержащая команды на перемещение рабочих органов оборудования, выполнения типовых операций и управления параметрами инструмента.



Рисунок 4.17 – Иллюстрация работы слайсера [61]

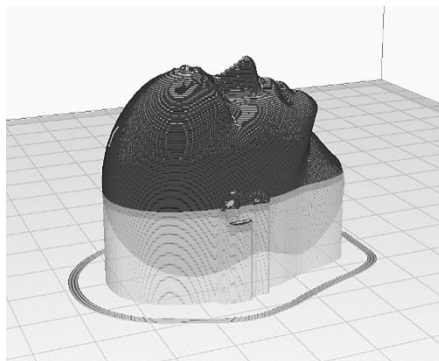


Рисунок 4.18 – Внешний вид рабочего окна программы слайсера Ultimaker Cura [62]

Процесс слайсинга 3D-объектов для печати состоит из следующих этапов:

- «нарезка» модели на слои;
- генерация заполнителей;
- генерация поддержки;
- формирование GCode для принтера;
- управление режимами печати;
- реализация функции «3D-принтер хост».

В зависимости от используемой технологии печати (экструдирование, напыление, и др.) применяются различные стратегии формирования GCode, в соответствии с разными топологиями слайсинга (то есть особенностей разделения пространственного объекта на слои). Наиболее распространенные топологии слайсинга [63]:

1) Для верхней и нижней граней объекта применяют сплошную заливку материалом в 3-4 слоя, иногда может требоваться до 8–10 слоев при слишком малой толщине самого слоя и/или повышенных требованиях к прочности.

2) Периметры слайса (то есть боковые стенки 3D-объекта) состоят преимущественно из 2 (иногда до 3–4) слоев для повышенной прочности конечной детали.

3) Внутренняя заливка слайса может изменяться в зависимости от требуемой прочности модели и составляет в среднем 15-30%. Совокупность конструкций, находящихся внутри 3D-объекта называется заполнителем или филлером и может иметь разную внутреннюю структуру (рисунок 4.19).

4) Специально формируемые пространственные структуры, предназначенные для последующего удаления при наличии нависающих элементов, которые не поддерживаны снизу самой формой объекта: суппорты или поддержка, рисунок 4.20).

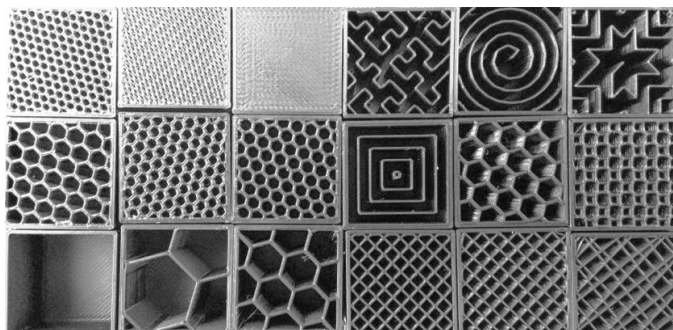


Рисунок 4.19 – Основные виды заполнителей, используемых в FDM-печати [63]



Рисунок 4.20 – Пример 3D-объекта с сформированной поддержкой

Заполнители. Структура, расположенная внутри внешней оболочки напечатанной 3D-модели, называется заполнителем. Одно из преимуществ 3D-печати заключается в том, что детали могут иметь разную степень наполнения материалом. С производственной точки зрения это снижает количество необходимого материала и стоимость, а также вес конечного продукта.

Среднестатистическая величина заполнения составляет порядка 15–30% общего внутреннего объема в зависимости от формы и назначения модели. В некоторых случаях, когда требуется повышенная прочность, возможно увеличение

значения величины заполнения до 50%. Заливка осуществляется внутри модели под внешние периметры [63].

В версии слайсера Ultimaker Cura (4.5) [62] доступно 13 типов заполнения:

- декоративные фигурки (низкая прочность): lines, zig-zag;
- «стандартные» 3D-модели (средняя прочность): grid, triangles, tri-hexagon;
- функциональные 3D-модели (высокая прочность): cubic, cubic subdivision, octet, quarter cubic, gyroid;
- гибкие 3D-модели: concentric, cross, cross 3D.

Дополнительным функционалом слайсера может быть формирование градиентного заполнения, дающего большую плотность заполнения 3D-объекта по периметру. Это дает ту же прочность и жесткость детали при использовании меньшего количества материала. Градиентное заполнение не следует путать с постепенным заполнением, которое обеспечивает нечто подобное, но по оси Z, а не по осям X и Y. Другими словами, этот параметр делает заполнение более плотным в верхней части 3D модели по сравнению с нижней частью. Это может сэкономить материал и время печати, сохраняя при этом достаточную прочность.

В процессе 3D-репликации и при копировании объектов необходимо учитывать ограничения как технологии работы 3D-сканера, о чем было рассказано в разделе 4.1, так и особенности технологии процесса 3D-печати, включающие артефакты, формируемые при слайсинге, а также дефекты 3D-печати на самом принтере (рисунок 4.21).

Рассмотрим основные виды дефектов при 3D-печати, характерные для FDM технологии:

- температурные дефекты;
- проблемы с механикой принтера или с калибровкой узлов принтера;
- несоответствие температурного режима материалу или неправильный выбор температурного режима;
- вобблинг – горизонтальные полосы через равные промежутки;
- дефекты и артефакты программы слайсера;
- прочие виды дефектов.

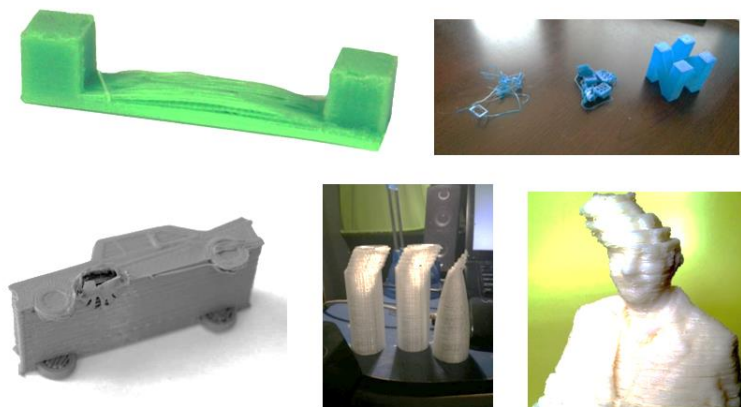


Рисунок 4.21 – К понятию реплицируемости 3D-представления

Температурные дефекты

Причиной возникновения таких дефектов может быть то, что деталь небольшая и печатается в одном экземпляре, так как постоянно находящееся над небольшой деталью или элементом сопло не даёт пластику после печати очередного слоя остыть и затвердеть (рисунок 4.22). Вариантом решения может быть размещение на столе нескольких экземпляров детали, в разных углах стола печати, чтобы была возможность охлаждения после печати слоя.



Рисунок 4.22 – Пример 3D-печати с температурным дефектом [63]

Проблемы с механикой принтера или с калибровкой узлов принтера

Примеры 3D-печати с дефектами, вызванными проблемами с механикой или с калибровкой узлов принтера, приведены на рисунках 4.23-4.25. Причиной возникновения таких дефектов может быть неправильная калибровка по оси Z. Для решения данной проблемы необходимо изменить высоту сопла по отношению к столу.

Пример дефекта печати, вызванного *несоответствием температурного режима материалу или неправильным выбором температурного режима*, приведен на рисунке 4.26. Для предотвращения появления подобных дефектов необходимо устанавливать правильную температуру сопла и стола печати, соответствующую используемому материалу, а также создавать внутри или вокруг 3D-принтера непродуваемую зону.



Рисунок 4.23 – Пример дефекта в виде щели между нитями первого слоя [63]

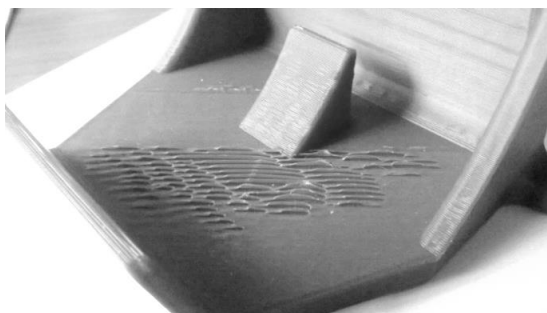


Рисунок 4.24 – Пример дефекта, вызванного переливом расплава, перегревом, неровным столом 3D-принтера [63]

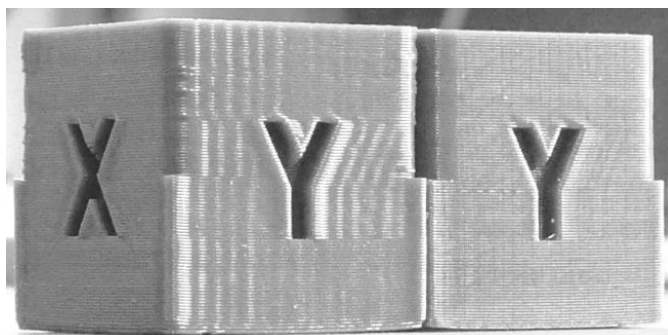


Рисунок 4.25 – Пример дефекта, вызванного колебаниями головки 3D-принтера при движении [63]

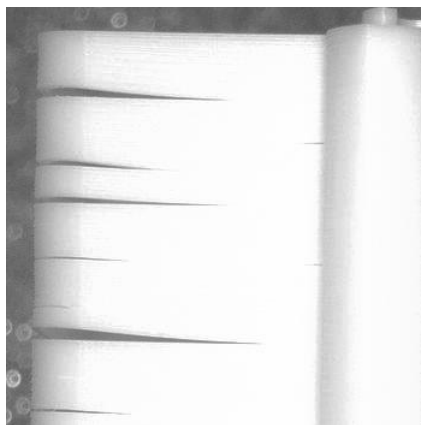


Рисунок 4.26 – Пример дефекта расслоения, вызванного неправильным температурным режимом [63]

Прочие виды дефектов

На рисунках 4.27-4.28 приведены примеры дефектов, вызванных неправильным расчетом скорости движения сопла в программе-слайсере и наличием влаги в материале.

В таблице 4.2 приведены примеры программного обеспечения, работа которого достаточно стабильна для различных моделей 3D-принтеров. В каждом случае необходимо пробовать различное ПО, так как его работа, особенно для оборудования из бюджетного ценового сегмента, зависит от многих факторов, включая температуру и влажность окружающей среды. Перед работой рекомендуется дополнительно ознакомиться с обучающими ресурсами [63-65].



Рисунок 4.27 – Наплывы материала, вызванные неправильным расчетом скорости движения сопла в слайсере [63]

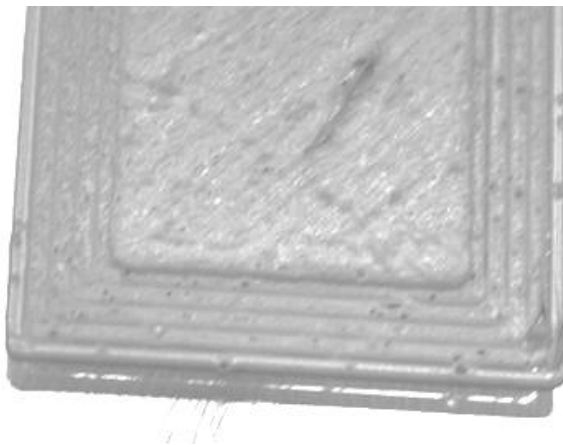


Рисунок 4.28 – Дефекты, вызванные наличием влаги в материале [63]

Библиотеки готовых 3D-моделей для печати: yeggi.com, open3dmodel.com, thingiverse.com, free3d.com.

Таблица 4.2 – Список актуальных программ для использования при 3D-печати

Software	Функции	Уровень	ОС
<u>Cura</u>	Slicer, 3D Printer Host	Beginner	Windows, Mac, Linux
<u>MatterControl 2.0</u>	Slicer, 3D Printer Host, Design	Beginner	Windows, Mac, Linux
<u>3DPrinterOS</u>	STL Editor, STL Repair, Slicer, 3D Printer Host	Beginner	Windows, Mac, Ubuntu, Raspberry Pi
<u>KISSlicer</u>	Slicer	Intermediate	Windows, Mac, Linux
<u>Slic3r</u>	Slicer	Intermediate	Windows, Mac, Linux
<u>SliceCrafter</u>	Slicer	Intermediate	Browser
<u>IceSL</u>	Slicer, Design	Intermediate	Windows, Linux
<u>OctoPrint</u>	Slicer, 3D Printer Host	Intermediate	Windows, Mac, Linux, Raspbian (as OctoPi image)
<u>Repetier-Host</u>	Slicer, 3D Printer Host	Intermediate	Windows, Mac, Linux
<u>3D-Tool Free Viewer</u>	STL Analysis	Intermediate	Windows
<u>MakePrintable</u>	STL Editor, STL Repair	Intermediate	Browser
<u>Meshmixer</u>	STL Editor, STL Repair	Intermediate	Windows, Mac
<u>MeshLab</u>	STL Editor, STL Repair	Professional	Windows, Mac, Linux
<u>Netfabb</u>	STL Repair, Slicer	Professional	Windows
<u>TinkerCAD</u>	Design	Beginner	Browser
<u>3D Slash</u>	Design	Beginner	Browser
<u>Sculptris</u>	Design	Beginner	Windows, Mac

<u>Vectary</u>	Design	Intermediate	Browser
<u>Figuro</u>	Design	Intermediate	Browser
<u>SketchUp Free</u>	Design	Intermediate	Browser
<u>Fusion 360</u>	Design	Intermediate	Windows, Mac
<u>FreeCAD</u>	Design	Intermediate	Windows, Mac, Linux
<u>Blender</u>	Design	Professional	Windows, Mac, Linux
<u>OnShape</u>	Design	Professional	Browser

4.3 Контрольные вопросы

- 1) Виды ошибок, вносимых ПО сканера:
 - а) изменение цвета текстуры;
 - б) искажение формы объекта;
 - в) потеря мелких элементов поверхности;
 - г) все перечисленные варианты.
- 2) На каких поверхностях в принципе невозможно использовать оптический 3D-сканер:
 - а) прозрачные поверхности;
 - б) непрозрачные поверхности;
 - в) поверхности с выпуклостями;
 - г) поверхности с отверстиями.
- 3) Артефакты сканирования – это:
 - а) исходные объекты сканирования;
 - б) результаты сканирования;
 - в) отклонения формы отсканированного объекта от формы исходного объекта.
- 4) Анизотропия разрешающей способности сканера:
 - а) различие разрешающей способности сканеры в разных направлениях;

- б) равенство разрешающей способности сканеры во всех направлениях;
 - в) изменение разрешающей способности сканера во времени;
 - г) постоянство разрешающей способности сканера во времени.
- 5) Регулярная структура объекта (повторяющаяся текстура) может приводить к:
- а) потери синхронизации положения сканера относительно объекта;
 - б) снижению объема работ по сканированию;
 - в) увеличению шума при сканировании.
- 6) Основным назначением программы-слайсера является:
- а) подготовка файла в GCode для последующей печати на 3D-принтере;
 - б) нарезка 3D-моделей на отдельные части для лучшего сжатия;
 - в) конвертация моделей в формат для 3D-вьюверов;
 - г) изготовление уменьшенных копий 3D-моделей.
- 7) Объекты, попавшие в область сканирования, но не относящихся к основному объекту сканирования:
- а) удаляются сканером автоматически;
 - б) невозможно удалить, требуется повторное пересканирование объекта;
 - в) возможно удалить в ПО сканера или при последующей программной обработке данных.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Аксенов А.Ю. Модели и методы обработки и представления сложных пространственных объектов Диссертация на соискание ученой степени кандидата технических наук. Санкт-Петербург. 2015. 110 с.
2. "Географические информационные системы федеральные, региональные, муниципальные. Общие технические требования. ГОСТ Р 52155-2003" (утв. Постановлением Госстандарта РФ от 09.12.2003 N 359-ст)
3. 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
4. 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
5. 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.
6. 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.
7. 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.
8. Мельникова О.Г., Олейников П.П. Информационное моделирование зданий: опыт реконструкции памятников культурного наследия // Социология города. 2013. № 4. С. 72-80.
9. 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.
10. 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.
 11. 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.
 12. Борисенко Б., Ярошенко С. 3D-сканирование в интересах 3D-моделирования // Comprice.ru. Электронный ресурс — Доступ: <http://www.comprice.ru/articles/detail.php?ID=40134>
 13. 3D-сканеры Artec. Трехмерное сканирование на примере модели Eva. Электронный ресурс — Доступ: http://www.ixbt.com/printer/3d/3d_scanner_eva.shtml
 14. Ричардсон Р. Сканируя пространство. // «Экспресс-Электроника», №10. 2003
 15. Телемедицина = Telemedicine : Новые информ. технологии на пороге XXI в. / [Юсупов Р.М., Полонников Р.И., Кувакин В.И. и др.]; Под ред. Р.М. Юсупова и Р.И. Полонникова; Рос. акад. наук. С.-Петерб. ин-т информатики и автоматизации. - СПб., 1998. – 488
 16. 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.
 17. Шикин Е. В., Боресков А. В. Компьютерная графика. Полигональные модели. М.: ДИАЛОГ-МИФИ, 2000. 464 с.
 18. Curless, B., Levoy, M., Volumetric Method for Building Complex Models from Range Images. Proc. SIGGRAPH '96
 19. Turk, G., Levoy, M., Zippered Polygon Meshes from Range Images. Proc. SIGGRAPH '94
 20. Luebke, David P. A Developer's Survey of Polygonal Simplification Algorithms. IEEE Computer Graphics & Applications, 2001.

21. 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
22. Luebke, D., Hallen, B. Perceptually Driven Interactive Rendering, University of Virginia Tech Report #CS-2001-01
23. Laur, D. and Hanrahan, P. Hierarchical Splatting: A Progressive Refinement Algorithm for Volume Rendering. Proc. SIGGRAPH 1991.
24. Westover, L. Footprint Evaluation for Volume Rendering. Proc. SIGGRAPH'90. 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.
25. Debevec, P., Introduction to Image-Based Modeling, Rendering, and Lighting. SIGGRAPH'2000 courses
26. Gortler, S., He, L., Cohen, M., Rendering Layered Depth Images. Microsoft Research, MSTR-TR-97-09
27. 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.
28. 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
29. 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.
30. Триангуляция Делоне и её применение. // Томск: Изд-во Том.ун-та, 2002. 128 с.
31. 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

32. Gumhold, S., Karni, Z., Isenburg, M., and Seidel, H.-P. 2004. Predictive point-cloud compression. In ACM SIGGRAPH Conference Abstracts and Applications
33. 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).
34. 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.
35. Ruwen Schnabel, Reinhard Klein. Octree-based Point-Cloud Compression. In Eurographics Symposium on Point-Based Graphics (2006)
36. 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
37. Кулешов С.В. Пространственно-временное представление, обработка и компрессия видеопотока. // Информационно-измерительные и управляющие системы, №4, т.6, 2008. с. 33–37.
38. 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
39. 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.
40. Александров В.В., Кулешов С.В. Этерификация и терминальные программы // Информационно-измерительные и управляющие системы, №10, т.6, 2008. С. 50–53.
41. 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

42. 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.
43. Александров В.В., Горский Н.Д. Алгоритмы и программы структурного метода обработки данных. Л. : Наука, 1983. 208 с.
44. Горский Н.Д., Мысько С.Н., Сухаричев В.П. Сравнительное исследование некоторых характеристик двумерных разверток. Ленинград, 1982, 24 с.
45. Александров В.В., Горский Н.Д. Структуризация иерархических систем. В кн.: Алгоритмические модели в автоматизации исследований. М. Наука, 1980.
46. 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.
47. Стронгин Р.Г. Численные методы в многоэкстремальных задачах. У.Наука,1970.
48. Математическая энциклопедия. — М.: Советская энциклопедия. И.М. Виноградов 1977—1985.
49. Peano G. Sur une courbe qui remplit toute une aire plene. *Mathemathematieche Annalen*, 36, 1890, p. 157–571.
50. Hilbert D. Uber die stetige Abbildung einer Linie ein Flächenstück. — *Math. Ann.*, 1891, p. 38.
51. Hilbert Curve Concepts & Implementation. Электронный ресурс — Доступ: http://www.fundza.com/algorithmic/space_filling/hilbert/basics/index.html
52. Дышкант Н.Ф. Эффективные алгоритмы сравнения поверхностей, заданных облаками точек. // Диссертация на соискание ученой степени кандидата физико-математических наук. Москва. 2011. 139 с.
53. Кондратюк А.А. К проблеме эффективности компьютерной реализации построения выпуклой оболочки в трехмерном пространстве. // Сборник трудов СПИИРАН.

Интеллектуализация автоматизации проектирования. СПб, 1994, С.55

54. 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
55. Ватолин Д., Ратушняк А., Смирнов М. Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео// М. Диалог-МИФИ. 2003. 384 с.
56. 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)
57. Witten I., Neal R.M., Cleary G. Arithmetic coding for data compression // Comm. ACM. 1987, V.30, No 6. pp. 520-540
58. Huffman, D.A. A method for the construction of minimum redundancy codes. In Proceedings IRE, vol. 40, 1962, pp. 1098-1101.
59. Инструкция пользователя Artec Studio. Электронный ресурс — Доступ: <http://docs.artec-group.com/as/12/ru/>
60. 3D принтер: принцип работы и возможности — KnowHow. Электронный ресурс — Доступ: https://knowhow.pp.ua/3d_printer/
61. Pham G.N., Lee S.-H., Kwon O.-H., Kwon, K.-R. A 3D Printing Model Watermarking Algorithm Based on 3D Slicing and Feature Points. Electronics 2018, 7, 23. <https://doi.org/10.3390/electronics7020023>
62. Ultimaker Cura. Электронный ресурс — Доступ: <https://ultimaker.com/software/ultimaker-cura>
63. Что такое заполнение при 3D-печати и как правильно настраивать его на 3D-принтере. Электронный ресурс — Доступ: https://epo3d.com/blog/31_что-такое-za-polnenie-pri-3d-pechat-i-kak-pr
64. Дефекты FDM 3D печати. URL: https://3deshnik.ru/wiki/index.php/Дефекты_FDM_3D_печати
65. Проблемы качества 3D-печати. URL: <https://3dpt.ru/page/faq>

Учебное издание

Кулешов Сергей Викторович
Зайцева Александра Алексеевна
Аксенов Алексей Юрьевич
Кипяткова Ирина Сергеевна
Карпов Алексей Анатольевич
Ватаманюк Ирина Валерьевна

3D-ТЕХНОЛОГИИ В СОВРЕМЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ: ТЕОРИЯ И ПРАКТИКА

Учебно-методическое пособие

ISBN 978-5-6047036-1-8



Подписано к печати 01.11.21. Формат 60х84 1/16.
Усл. печ. л. 6,45. Уч.-изд. л. 6,93. Тираж 250 экз. Заказ №

Отпечатано с оригинал-макета СПб ФИЦ РАН
199178, Санкт-Петербург, 14-я линия В.О., д. 39
в редакционно-издательском центре ГУАП
190000, Санкт-Петербург, ул. Б. Морская, 67