

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Нижегородский государственный архитектурно-строительный университет»

На правах рукописи



Роменский Сергей Александрович

**АЛГОРИТМЫ ПРЕОБРАЗОВАНИЯ
ЧЕРТЕЖНО-КОНСТРУКТОРСКОЙ ДОКУМЕНТАЦИИ
В ТРЕХМЕРНУЮ КАРКАСНУЮ МОДЕЛЬ**

2.5.1. «Инженерная геометрия и компьютерная графика.
Цифровая поддержка жизненного цикла изделий»

Диссертация на соискание ученой
степени кандидата технических наук

Научный руководитель д.т.н., проф. Ротков С.И.

Нижний Новгород - 2025

Содержание

ВВЕДЕНИЕ.....	4
ГЛАВА 1. СИНТЕЗ ТРЕХМЕРНОЙ КАРКАСНОЙ МОДЕЛИ ИЗ ДВУМЕРНЫХ ПРОЕКЦИОННЫХ ИЗОБРАЖЕНИЙ.....	13
1.1 Постановка задачи.....	13
1.2 Обзор существующих решений.....	20
1.3 Основные понятия.....	37
1.4 Выводы по первой главе.....	42
ГЛАВА 2. ПОЛУЧЕНИЕ ГЕОМЕТРО-ГРАФИЧЕСКОЙ ИНФОРМАЦИИ ИЗ РАЗЛИЧНЫХ ИСТОЧНИКОВ И ПОДГОТОВКА К ЕЁ ИСПОЛЬЗОВАНИЮ.....	43
2.1 Получение геометро-графической информации из различных источников.	43
2.2 Получение геометро-графической информации при помощи СОМ-объекта на примере САПР Компас-3D.....	51
2.3 Поиск и контроль ошибок проектирования/распознавания/векторизации и алгоритмы их исправления.....	60
2.4 Выделение компонент связности на чертежах.....	67
2.5 Выводы по второй главе.....	70
ГЛАВА 3. АЛГОРИТМЫ ФОРМИРОВАНИЯ ТРЕХМЕРНОЙ КАРКАСНОЙ МОДЕЛИ.....	71
3.1 Выделение проекционных видов и восстановление проекционных связей.	71
3.2 Анализ проекционных изображений.....	78
3.3 Получение точечной модели.....	81
3.4 Формирование каркасной модели.....	84
3.5 Выводы по третьей главе.....	90
ГЛАВА 4. ИССЛЕДОВАНИЕ ВЫЧИСЛИТЕЛЬНЫХ ХАРАКТЕРИСТИК АЛГОРИТМА ВОССТАНОВЛЕНИЯ КАРКАСНЫХ МОДЕЛЕЙ ПО ЧЕРТЕЖНО-КОНСТРУКТОРСКОЙ ДОКУМЕНТАЦИИ.....	91
4.1 Исследование временных характеристик алгоритма восстановления каркасной модели по проекционным изображениям.....	91
4.2 Программная реализация представленного алгоритма.....	94

4.3 Сравнение ручного построения трехмерной детали и автоматического.....	96
4.4 Сравнение времени чтения геометрической информации из внешнего источника и времени восстановления модели.....	97
4.5 Восстановление трехмерных каркасных моделей на примере чертежей, использующихся на производстве.....	102
4.6 Выводы по четвертой главе.....	109
ГЛАВА 5. ОРГАНИЗАЦИЯ СИСТЕМЫ УПРАВЛЕНИЯ ДАННЫМИ ДЛЯ ОПЕРАТИВНОЙ РАБОТЫ С ГЕОМЕТРО-ГРАФИЧЕСКОЙ ИНФОРМАЦИЕЙ	110
5.1 Постановка задачи.....	110
5.2 Обзор существующих решений.....	112
5.3 Макет системы управления для хранения геометрической информации....	122
5.4 Описание хранения двумерной ГГИ.....	126
5.5 Описание хранения трехмерной ГГИ.....	128
5.6 Выводы по пятой главе.....	134
ЗАКЛЮЧЕНИЕ.....	135
СПИСОК ЛИТЕРАТУРЫ.....	139
ПРИЛОЖЕНИЕ А. АКТЫ О ВНЕДРЕНИИ.....	152
ПРИЛОЖЕНИЕ Б. СВИДЕТЕЛЬСТВА О РЕГИСТРАЦИИ ПРОГРАММ ДЛЯ ЭВМ.....	154

ВВЕДЕНИЕ

Актуальность темы исследования. Активно идущий процесс всеобщей цифровизации различных отраслей науки, техники и технологий предполагает перевод всех проектов с аналогового представления информации в цифровой вид. Об этом еще пятьдесят лет тому назад писал академик В.М.Глушков в своей монографии «Основы безбумажной информатики». Это обстоятельство также относится и к процессам конструирования объектов, которые ведутся на основе объектов-аналогов, ранее разработанных в той или иной организации. Аналоги хранятся либо в виде бумажного архива чертежно-конструкторской и технологической документации, либо в виде трехмерных объектов-прототипов. В любом случае информация об объекте проектирования или производства должна быть сохранена и составляет информационную поддержку жизненного цикла изделия (ЖЦИ). Данная поддержка осуществляется при помощи совокупности различных систем, объединенных единой базой данных и получила название CALS (Continuous Acquisition and Lifecycle Support) для машиностроительного комплекса и BIM (Building Information Modelling) для строительного комплекса.

Возможность автоматически восстанавливать образы объектов в процессе автоматического проектирования полностью соответствует современной идеологии CALS-технологий. Современные системы геометрии и графики, используемые на различных этапах поддержки жизненного цикла изделия, должны работать одновременно в двух пространствах измерений таким образом, чтобы между моделью объекта (3-х мерное пространство, далее 3D) и его изображением (2-мерное пространство, далее 2D) была двухсторонняя взаимно однозначная информационная связь (далее 3D <-> 2D). Однако подавляющее большинство систем предоставляют только связь 3D->2D, что существенно ограничивает возможности проектирования и не даёт конструктору пользоваться традиционными методами проектирования. Так же необходимо учитывать тот факт, что технический чертёж содержит всю информацию о проектируемом объекте в том виде, который естественен для конструктора, полностью соответствует государственному стандарту и легко может быть создан в

электронном виде.

В современных САПР, поддерживающих технологии CALS и BIM, возможности формирования геометрических моделей трёхмерных объектов сводятся к четырём базовым операциям: трём кинематическим (линейное перемещение, вращение, движение по заданной траектории) и построению модели через сечения. В основе этих функциональных операторов лежит каркасно-кинематический метод. Все эти методы опираются на входные двумерные данные, такие как плоские контуры, заданные либо на бумаге, либо на экране компьютера, что ограничивает гибкость работы с объёмными формами.

При этом в промышленной практике проектирование до сих пор основывается на бумажных или, в лучшем случае, цифровых чертежах, содержащих преимущественно двумерную информацию. Архивы конструкторской документации предприятий могут насчитывать миллионы листов формата А0, что создаёт значительные сложности при их цифровизации.

Создание системы, выполняющей автоматическую реконструкцию трехмерных моделей по техническому чертежу, позволило бы во много раз сократить время проектирования объектов различного назначения. Поэтому необходимо разрабатывать математическое и программное обеспечение системы автоматической реконструкции трехмерных моделей объектов по техническому чертежу.

Степень разработанности темы исследования. Первые исследования в области восстановления 3D моделей по проекциям были проведены Х.И. Тани (1964), В.А. Щеколдиным (1968), и В.С. Полозовым в его докторской диссертации (1983). Аналогичные исследования были проведены профессором Masanori Idesawa в 1973 году и были представлены в его работе «A system to generate a solid figure from three views». В.С. Полозовым (диссертация на соискание степени д.т.н., 1983) была сформулирована и поставлена проблема автоматизации процесса преобразования двумерной геометрической модели в трехмерную конструктивную модель.

Так же решением данной проблемы занимались Ротков С.И., Тюрина В.А.

Ими были предложены методы и алгоритмы автоматизации процесса преобразования двумерного технического чертежа в трехмерную каркасную модель, причем проблема удаления ложных ребер каркасной модели решалась путем построения графа трехмерной модели и применением фундаментальных циклов.

Попытка решения поставленной проблемы как обратной задачи геометрии и графики была осуществлена в системе геометрии и графики “Китеж”. Данная система включает в себя реализованный алгоритм преобразования технического чертежа в трехмерную каркасную модель с существенными ограничениями по количеству вершин в каркасной модели (256) и строгим контролем входных данных (в частности, это касается наличия линий проекционной связи между различными видами изделия на исходном чертеже), обусловленные возможностями вычислительной техники, языков программирования, используемых при реализации, а также уровнем технологий программирования того периода времени.

Несмотря на имеющийся массив исследований, все ещё остаются некоторые нерешенные проблемы (например, ограниченная автоматизация, зависимость от качества входных данных, ограниченная функциональность и т.д.), попытка решить которые и делается в данной работе.

Объектом исследования являются двумерные и трехмерные геометрические модели объектов класса “Деталь”.

Предметом исследования являются алгоритмы преобразования двумерных геометрических моделей, заданных на поле многовидового технического чертежа, в трехмерные каркасные модели, содержащие ребра, описываемые уравнениями до второго порядка включительно.

Цель исследования. Основная цель данной работы заключается в развитии метода синтеза 3D каркасных моделей пространственных объектов по многовидовому чертежу путём расширения класса моделируемых тел.

Задачи исследования:

1. Разработать структурную схему автоматизированного процесса

преобразования двумерного многовидового чертежа в трехмерную модель на основе анализа существующих методов перехода 2D-геометрии в 3D для нахождения эффективных решений обратной задачи начертательной геометрии.

2. Разработать алгоритмы обработки двумерной геометрической модели для устранения различных погрешностей ручного ввода и улучшения качества входных данных для алгоритма преобразования технического чертежа в трехмерную каркасную модель.
3. Разработать геометрический алгоритм преобразования двумерных проекционных изображений в трехмерную каркасную модель и подтвердить экспериментально его временные характеристики, определенные теоретически.
4. Разработать макет системы управления и хранения для оперативной работы с геометро-графической информацией (ГГИ) в исследуемой проблеме и реализовать модуль импорта ГГИ в неё.

Научная новизна исследования:

1. Предложена структурная схема решения обратной задачи начертательной геометрии, отличительной особенностью которой является наличие этапа автоматизированного синтеза трехмерной модели с применением алгоритма предварительной обработки входной геометро-графической информации.
2. Разработан алгоритм предварительной обработки двумерных геометрических данных, ориентированный на устранение противоречия между входным предметно-манипулятивным чертежом и выходным эпюром Монжа и основанный на анализе проекционных изображений, результатом которого является восстановление топологических и метрических связей.
3. Разработан геометрический алгоритм преобразования двумерных проекций в трехмерную каркасную модель, основанный на анализе топологических и метрических взаимосвязей элементов чертежа. Отличительная особенность алгоритма заключается в его структуре, оптимизированной для

реконструкции каркасной модели.

Теоретическая значимость диссертационного исследования заключается в развитии и систематизации научных представлений в области геометрического моделирования и компьютерной графики применительно к задачам автоматизированного преобразования конструкторской документации. Работа вносит вклад в теоретические основы реконструкции трехмерных геометрических моделей по двумерным проекциям, анализируя и формализуя принципы, заложенные в традиционной инженерной графике.

Практическая значимость работы заключается в разработке отдельных этапов промышленной информационной технологии преобразования архивных чертежно-конструкторских документов на бумажных носителях в электронную модель 3D объекта класса «Деталь», не зависящей от конкретной CAD/CAM/CAE/PDM/PLM системы.

Результаты данной работы могут быть применены для автоматизации процесса преобразования двумерных технических чертежей в трехмерные каркасные модели на производстве, использующем бумажный или электронный архив двумерных чертежей.

Результаты диссертационной работы внедрены в практическую деятельность следующих организаций: АО ЦНИИ Буревестник (акт внедрения результатов работ от 30.05.2025), ООО «Лаборатория КОМПАС» компании АО Аскон (акт о внедрении результатов работ от 29.12.2020).

Исследование выполнялось в рамках грантов:

- РФФИ:

- № 15-07-05110 «Исследование методов генерации электронных 3D моделей геометрических объектов по изображениям, поиска и определения ложных данных для систем виртуальной реальности и управления научно-техническими данными»,

- №17-07-00543 «Исследование и разработка методов и средств геометрического моделирования и отображения сложноструктурированных поверхностей для уникальных конструкций»,

- №19-07-01024 «Исследование и разработка алгоритмов автоматизации преобразования двумерных геометрических моделей в трехмерные каркасные модели с интегрированными средствами обмена данными между системами геометрии и графики и прикладными программами».

- РНФ:

- № 25-21-00003 «Новая концепция твердотельного моделирования геометрических объектов в точечном исчислении».

Методология и методы исследования. Исследование опирается на фундаментальные принципы начертательной геометрии, являющейся теоретической базой для построения и чтения технических чертежей. В работе используются и развиваются методы построения и анализа ортогональных проекций, определения истинных размеров и форм геометрических объектов, а также восстановления их пространственного положения по плоским изображениям. Формализация этих принципов в виде вычислительных алгоритмов позволяет перевести интуитивные или графические операции инженера в область автоматизированной обработки данных.

Активно применяются и развиваются методы компьютерной графики и вычислительной геометрии. Это включает использование различных способов представления геометрической информации (например, каркасные, поверхностные или твердотельные модели), алгоритмы геометрических преобразований (вращение, перенос, масштабирование), методы анализа топологических связей между геометрическими примитивами. Исследование демонстрирует, как эти методы могут быть адаптированы и объединены для решения сложной задачи интерпретации символьного и графического языка инженерных чертежей.

Экспериментальные исследования выполнены с помощью разработанного программного комплекса, реализованного на языке программирования C++.

Соответствие паспорту специальности 2.5.1. По своему содержанию диссертационная работа соответствует следующим пунктам научной

специальности 2.5.1. «Инженерная геометрия и компьютерная графика. Цифровая поддержка жизненного цикла изделий»:

П.2 Теория и практика непрерывного и дискретного геометрического моделирования. Конструирование кривых линий, поверхностей и тел по заданным требованиям.

П.11 Разработка геометрических и других научных основ построения систем и средств цифровой поддержки процессов ЖЦИ, разработка и исследование методов, моделей и алгоритмов синтеза и анализа решений различного уровня, включая конструкторские и технологические решения.

П.13. Разработка геометрических и других научных основ построения средств цифрового документирования, безбумажного документооборота, процессов работы электронных архивов технической документации, взаимодействия с изготовителем и потребителем изделий.

Положения, выносимые на защиту.

1. Структурная схема решения обратной задачи начертательной геометрии.
2. Автоматизация предварительной обработки двумерных геометрических данных, позволяющая устранить противоречия между входным предметно-манипулятивным чертежом и выходным эпюром Монжа.
3. Расширение возможностей метода синтеза 3D моделей пространственных объектов по многовидовому чертежу путём расширения класса моделируемых тел.

Степень достоверности и апробация результатов. Достоверность полученных результатов обеспечивается корректным использованием аппарата алгебры, логики, теории геометрических построений, параметрической геометрии и др., экспериментальной проверкой алгоритмов, результатами опытной эксплуатации разработанных программных средств.

Результаты исследований были представлены и обсуждены на научных семинарах кафедры «Инженерной графики и информационного моделирования» ННГАСУ, а также на следующих конференциях:

- XV и XVI Международные конференции "Системы проектирования,

технологической подготовки производства и управления этапами жизненного цикла промышленного продукта (CAD/CAM/PDM-2015, 2016)" (Москва, ИПУ РАН, 2015,2016);

- II, III и IV Международные научно-практические конференции «Виртуальное моделирование, прототипирование и промышленный дизайн» (Тамбов, 2015, 2016, 2017);
- 26-я,27-я,28-я,29,30-я и 32-я Международная конференция и школа-семинар по компьютерной графике, обработке изображений и машинному зрению, системам визуализации и виртуального окружения GraphiCon (Нижний Новгород, 2016, Пермь, 2017, Томск, 2018, Брянск, 2019, Санкт-Петербург, 2020, Рязань, 2022)
- Третья Международная конференция «Ситуационные центры и геоинформационные системы для задач мониторинга и безопасности: 10 лет НеоГеографии (SC-NeoGeo-VRTerro2015)» (Пушино, АНО ИФТИ, 2015);
- Международная научно-практическая конференция «Информатизация инженерного образования» (Москва, МЭИ, 2016);
- Международная научная Школа-семинар Московского физико-технического института (государственного университета) Института физико-технической информатики SCVRT1516 (Протвино, АНО ИФТИ, 2016);
- 9-я Международная конференция «Физико-техническая информатика» СРТ-2021 (Пушино, АНО НИЦ ФТИ, 2021);
- VI Международная конференция и молодежная школа «Информационные технологии и нанотехнологии» (ИТНТ-2020) (Самара,2020);
- на региональных конференциях: «XX Нижегородская сессия молодых ученых. Естественные, математические науки» (Арзамас, Нижегородская обл., 2015); «XXI Нижегородская сессия молодых ученых. Технические науки» (Арзамас, Нижегородская обл., 2016); «XXI Нижегородская сессия молодых ученых. Естественные, математические науки» (Арзамас, Нижегородская обл., 2016).

Публикации по теме диссертации. Основные материалы диссертации опубликованы в 24 статьях, в том числе две – в изданиях, рекомендованных ВАК по научной специальности 2.5.1 уровня К2, две – в изданиях, рекомендованных ВАК по смежным специальностям, одна - уровня К1, одна - уровня К3, одна – проиндексированы в Scopus. Получены три свидетельства о регистрации программ для ЭВМ.

Диссертация состоит из введения, пяти глав, заключения, списка используемой литературы и двух приложений общим объемом 157 страниц. Содержит 44 рисунка, четыре таблицы. Список литературы содержит 103 источника.

ГЛАВА 1. СИНТЕЗ ТРЕХМЕРНОЙ КАРКАСНОЙ МОДЕЛИ ИЗ ДВУМЕРНЫХ ПРОЕКЦИОННЫХ ИЗОБРАЖЕНИЙ

1.1 Постановка задачи

Начертательная геометрия (далее НГ), согласно определению проф. И.И.Котова [21], это раздел геометрии, изучающий методы построения чертежей пространственных фигур и способы решения на этих чертежах различных геометрических задач. Из этого определения следует, что как наука НГ имеет две основные задачи, прямую и обратную, которые заключаются в следующем:

- прямая задача - построение изображения пространственного объекта при заданном аппарате проецирования;
- обратная задача - создание геометрической модели тела по данному чертежу, то есть задача чтения чертежа.

Здесь и далее под геометрической моделью будем понимать некоторый способ математического описания формы объекта, а также совокупность данных, представляющих её в памяти ЭВМ.

В настоящее время начертательная геометрия и смежные с ней прикладные науки - вычислительная геометрия и компьютерная графика - приобрели особое значение в области автоматизации конструкторской деятельности. Теперь в любой конкурентоспособной системе автоматизации проектирования наряду с графическим редактором для создания чертежа и системой оформления чертёжно-конструкторской документации обязательно должен присутствовать пакет программ для создания геометрической модели тела (детали или механизма), поскольку информация о геометрических характеристиках необходима:

- для расчёта механических характеристик тела, таких, как масса, объём, моменты инерции и др.);
- для прочностного расчёта тела или конструкции методом конечных элементов;
- для автоматического построения чертежа при заданном аппарате

проецирования, а также построения разрезов и сечений;

- для изготовления детали или макета на станках с ЧПУ или 3d принтере;
- для визуализации объекта на экране дисплея с наложением текстуры, полей освещённости, с возможностью выбора точки зрения и т.д.

Известны различные методы конструирования пространственных (далее будем придерживаться обозначения 3D, от английского three dimensional - трёхмерный, в отличие от 2D - двумерных, плоских) геометрических моделей, например:

- метод обобщённого вращения плоского контура вокруг заданной оси;
- метод обобщённого сдвига контура вдоль заданного вектора (моделирование цилиндрических и конических тел);
- метод синтеза 3D модели по заданным проекциям тела (по многовидовому чертежу);
- метод теоретико-множественных операций (ТМО) объединения, пересечения, вычитания над ранее полученными объектами или над простыми телами, такими как параллелепипед, цилиндр, шар и т.п.

Первые три способа являются частными случаями решения обратной задачи НГ, так как 3D модель создается, исходя из имеющегося плоского чертежа. В этих способах реализуется каркасно-кинематический способ генерации [21,26,27] данных о пространственном объекте.

В общем виде обратная задача НГ трудно формализуема. Как было указано в [43] на сегодняшний день никто не доказал ни существования алгоритма общего решения, ни его отсутствия. Возможность решения обратной задачи НГ заложена в правильно построенном чертеже, поскольку согласно принципу обратимости чертежа он должен нести информацию необходимую и достаточную для мысленного восстановления тела.

В любом алгоритме создания 3D модели по плоскому изображению могут быть выделены следующие части [44]:

1. анализ изображения и подготовка его к синтезу модели;

2. синтез вершин пространственной модели;
3. создание каркасной модели, анализ на соответствие исходному изображению и пространственному объекту;
4. создание граничной модели, анализ на соответствие объекту;
5. создание конструктивной модели тела, которую можно использовать в ТМО.

Метод синтеза по многовидовому чертежу является наиболее сложным, поскольку информацию о геометрии 3D объекта необходимо получить по его плоским изображениям и информации о связях между ними, заданной неявно, в виде правил построения чертежей.

Как показывает опыт создания и использования СИГМ КИТЕЖ, преимуществом метода является возможность построения тел более сложной формы, для конструирования которых с помощью ТМО потребовалось бы несколько операций, за время, сравнимое со временем выполнения одной-двух операций [44].

Одной из технологических проблем в программной реализации метода синтеза 3D объекта является проблема ввода исходных геометрических данных. Во всех ранее известных программных реализациях эта проблема никак не затрагивалась. Данные вводились вручную в виде структурированного массива чисел, что делало программу не пригодной для промышленного использования в рамках той или иной системы проектирования. Подготовка данных в этом случае требовала большого ручного труда и значительного времени подготовки.

Технические чертежи служат универсальным языком для передачи инженерных решений и конструкторских идей. Изначально их создание требовало ручного труда с помощью специализированных инструментов вроде рейсшин, циркулей и чертёжных ручек. С развитием технологий эту функцию взяли на себя системы автоматизированного проектирования (CAD), которые не только ускорили процесс создания 2D- и 3D-моделей, но и позволили детализировать проекты до мелочей[26]. Однако многие предприятия до сих пор сталкиваются с проблемой: у них есть обширные архивы готовых изделий, но отсутствуют

связанные с ними CAD-файлы.

Возможность автоматически восстанавливать образы объектов в процессе автоматического проектирования полностью соответствует современной идеологии CALS-технологий.

CALS (Continuous Acquisition and Lifecycle Support) — концепция, объединяющая принципы и технологии информационной поддержки жизненного цикла продукции на всех его стадиях, основанная на использовании интегрированной информационной среды (единого информационного пространства), обеспечивающая единообразные способы управления процессами и взаимодействия всех участников этого цикла: заказчиков продукции (включая государственные учреждения, и ведомства), поставщиков (производителей) продукции, эксплуатационного и ремонтного персонала, реализованная в соответствии с требованиями системы международных стандартов, регламентирующих правила указанного взаимодействия преимущественно посредством электронного обмена данными [23,43].

Это же определение справедливо и для BIM (Building Information Modelling) – Информационное моделирование зданий и сооружений.

Это определение также означает, что основой электронной модели изделия, обеспечивающей все информационные процессы в рамках обсуждаемой концепции, является геометрическая модель проектируемого или изготавливаемого объекта.

Оцифровка бумажных чертежей остаётся актуальной задачей, поскольку большинство производственных архивов содержат именно такие документы — иногда в количестве миллионов листов[11]. Обычно их сканируют и хранят как растровые изображения, но этого недостаточно для модернизации или адаптации деталей под новые требования. Современные CAD-системы не могут автоматически преобразовывать такие сканы в точные 3D-модели, что особенно касается сложных проекций и сборок. Эта проблема подчёркивает необходимость разработки специализированных алгоритмов, способных корректно интерпретировать геометрию и топологию объектов на основе ортогональных

видов, сохраняя при этом все конструкторские особенности исходного чертежа.

В качестве примера приведем данные из диссертации Широкого [55]. На Горьковском автомобильном заводе за 90 лет работы предприятия накопился архив бумажной конструкторско-технологической информации в объеме 60 млн. листов формата А0, разбросанных по различным производственным площадкам. Если привести все листы к формату А4, как это предусмотрено ГОСТ, и собрать все вместе, то это будет куб с ребром в 100 м.

Как отмечает Г.Б.Широкий в диссертации «Средства архивизации геометро-графической информации...» [55], попытки массового сканирования таких массивов не решают проблему:

- полученные растровые изображения не содержат данных о взаимосвязях геометрических элементов, что требует последующей ручной векторизации. Существующие инструменты конвертации «растр-вектор» не анализируют семантику чертежей, а автоматизация перевода бумажных документов в трёхмерные модели остаётся нереализованной задачей. Даже при использовании сканеров процесс оцифровки сохраняет высокую трудоёмкость, так как создание 3D-объектов из отсканированных данных по-прежнему выполняется исключительно вручную;

- отсканированный лист формата А0 при стандартных параметрах сканирования 300dpi занимает объем 15 Мб без использования программных средств сжатия изображения без потери качества данных. Кроме того, процесс сканирования требует значительных вычислительных ресурсов для хранения и обработки чертежей, не говоря уже о требуемых людских, временных и финансовых затратах.

Подводя итог вышеперечисленного, требуется разработать структурную схему технологии решения обратной задачи начертательной геометрии, которая учитывала бы наличие бумажных чертежей в цепочке производства трехмерной детали. На блок-схеме ниже (рисунок 1.1) представлена предлагаемая структурная схема преобразования бумажного чертежа в трехмерную модель. Данная технология предполагает следующие этапы:

1. Сканирование бумажного чертежа[1]. Бумажный чертеж оцифровывается с помощью сканера, что позволяет получить растровое изображение. Происходит обработка и устранение недостатков, связанных с устареванием бумажного носителя.
2. Получение векторизованного чертежа на основании сканированного изображения [5]. Растровое изображение преобразуется в векторный формат, что делает возможным анализ и обработку геометрических данных.
3. Перенос векторной геометро-графической информации в специализированную базу данных для оперативной работы, более подробно данный процесс описывается далее, в главе 5.
4. Выделение внутри чертежа проекционных видов и формирование на их основе трехмерной каркасной модели. Данный алгоритм расписан подробно в главах 2 и 3.
5. Поиск и удаление ложных ребер и вершин, формирование граничной модели и перевод её в конструктивную модель. Данный алгоритм разработан и реализован Тюриной В.А. в её кандидатской диссертации [51, 37, 43, 46]. Готовая трехмерная модель сохраняется и может быть использована для дальнейшего анализа или проектирования.

Эта технология позволяет автоматизировать процесс преобразования бумажных чертежей в трехмерные модели, что значительно упрощает работу с архивными данными и ускоряет процесс проектирования [32]. Все данные, включая исходные чертежи, промежуточные модели и конечную трехмерную модель, сохраняются в архиве для последующего использования.

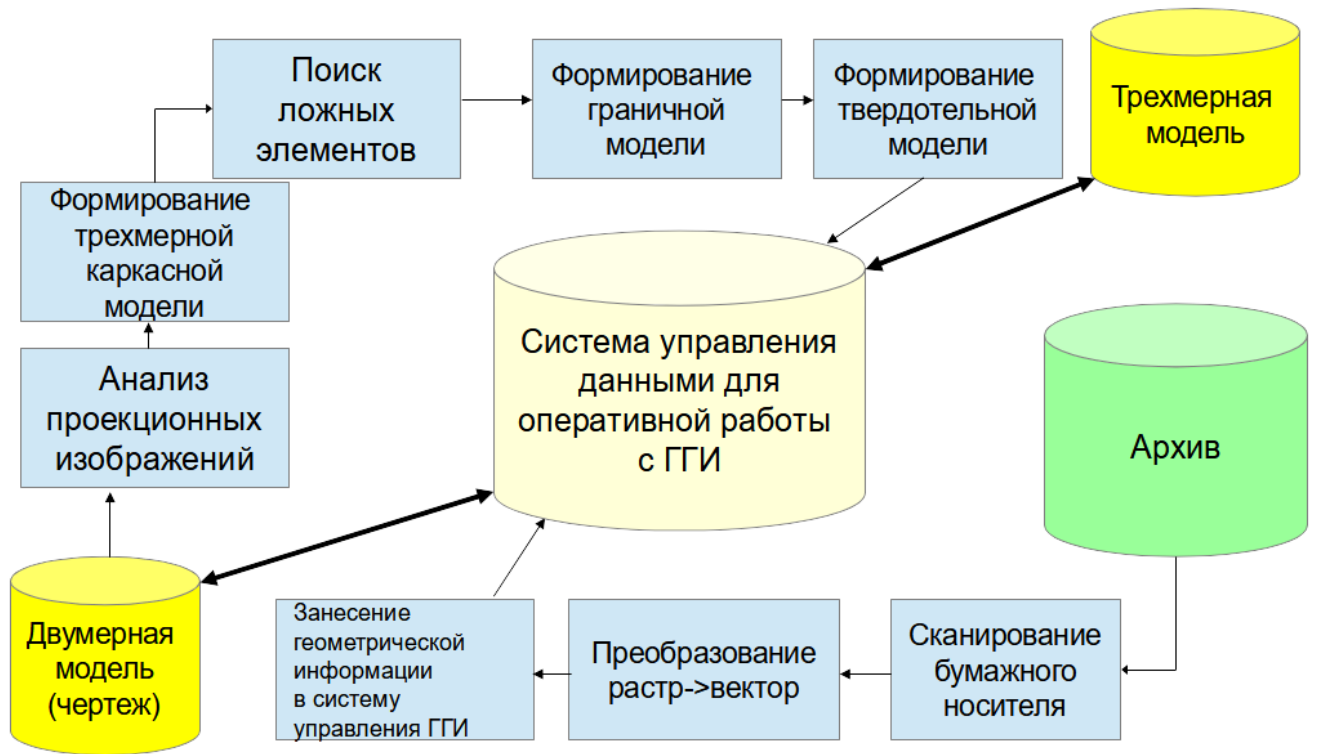


Рисунок 1.1 - Предлагаемая структурная схема решения обратной задачи начертательной геометрии.

1.2 Обзор существующих решений

Автоматическая реконструкция 3D CAD-моделей из 2D проекционных изображений является давней проблемой, начиная с конца 1960-х годов. Первые исследования в области восстановления 3D моделей по проекциям были проведены Х.И. Тани (1964)[49], В.А. Щеколдиным (1968)[54], и В.С. Полозовым (1966) [30]. Аналогичные исследования были проведены профессором Masanori Idesawa в 1973 году и были представлены в его работе «A system to generate a solid figure from three views»[72]. В.С. Полозовым была сформулирована и поставлена [20, 31] Подход преобразования 2D чертежей в граничные модели был так же формализован в работах [83,95]. С тех пор исследовались различные идеи, которые обсуждаются в этом разделе. Они включают в себя построение промежуточной каркасной модели, которая затем используется для реконструкции 3D CAD-моделей.

Относительно недавно появилось новое направление по созданию трехмерных моделей - при помощи нейронных сетей. данный подход характеризуется созданием облака точек на основе проекционных изображений, в котором потом нейронной сетью распознается та или иная геометрическая фигура.

1.2.1 Реконструкция 3D-каркасных моделей

Начальный этап подхода к 3D-реконструкции - автоматическое преобразование двумерных чертежей в трехмерную каркасную модель. Некоторые методы по реконструкции каркасной модели были сосредоточены на генерации многогранников из их ортогональных проекций.

Например, в статье [100] описывается алгоритм для восстановления многогранников по двумерным ортогональным проекциям. Алгоритм способен обрабатывать различные случаи и эффективно определять все возможные корректные решения. В нём используются методы для построения каркасных структур-кандидатов, обработки прерывистых линий во входных данных, генерации контуров граней, определения разделяющих рёбер и вершин,

построения контуров тела и сборки этих контуров в возможные объекты.

Furferi и др. [66] представили автоматизированный метод преобразования 2D ортогональных чертежей в 3D псевдокаркасные модели на основе граничного подхода. Авторы разработали систематическую процедуру, использующую DXF-файлы в качестве входных данных, которая включает этапы: предобработку (сегментацию рёбер, проверку коллинеарности), построение топологической структуры вершин и рёбер, а также генерацию 3D каркаса. Метод ориентирован на многогранники, что упрощает реализацию и минимизирует взаимодействие с пользователем. Алгоритм реализован в MatLab и протестирован на ряде примеров, подтвердив свою эффективность. Основное преимущество — автоматизация и чёткая структура, упрощающая интеграцию в исследовательские задачи. Однако метод не поддерживает криволинейные объекты, а его дальнейшее развитие предполагает переход к реконструкции полноценных твердотельных моделей. Работа служит основой для разработки инструментов преобразования инженерных чертежей в 3D CAD-модели.

Gorgani и др. [70] исследуют процесс получения 3D моделей из 2D чертежей, используя метод анализа поверхности с применением нечеткой логики. Авторы подчеркивают, что традиционные методы требуют три ортогональных вида, в то время как их метод позволяет использовать только два, что упрощает процесс и уменьшает неопределенности, связанные с различиями в представлениях. Улучшение точности и сокращение затрат на проектирование и производство делают этот подход более эффективным, способствуя устойчивому развитию в экономической, социальной и экологической сферах. В статье также отмечается, что данный метод применим только к моделям с плоскими гранями и предлагает направления для будущих исследований, направленных на расширение его применения к изогнутым поверхностям. Вышеперечисленные методы не учитывали построение каркасной модели с криволинейными рёбрами. Поскольку криволинейные рёбра и поверхности являются важными частями CAD-моделей, необходимо уделять особое внимание их обработке.

Значительный вклад в реконструкцию криволинейных каркасных моделей

был представлен Liu и др. [87]. Объединив геометрические свойства коник с аффинными свойствами, их метод может устранить ограничения, накладываемые на оси коник другими методами. Учитывая, что двух сопряженных диаметров достаточно для определения уникального конического сечения, метод сначала определяет два сопряженных диаметра пространственной коники из ее проецируемых коник, а затем генерирует 3D коническое ребро из двух сопряженных диаметров. Однако это сложная процедура, требующая различных техник извлечения геометрических параметров для разных типов коник, в которой участвует большое количество проверки пересечений и аффинных преобразований.

Lu и др. [82] предложили автоматический метод создания трехмерной модели на основе техники подгонки квадратичной поверхности для извлечения полных сетей характеристических кривых (FCN) из 3D-сеток поверхности, а также для нахождения циклов и генерации высококачественной сегментации. В начальной выборке шумовых и фрагментированных характеристических кривых авторы сначала подгоняют квадратичные поверхности к каждой кривой и соответствующим соседним вершинам, чтобы отфильтровать незначительные или шумовые кривые признаков. Затем проводят этап расширения признаков, чтобы решить проблемы пересечений кривых и сформировать замкнутую FCN. В итоге рассматриваются круговые кривые как циклы в полной FCN и полученная сеть сегментируется на патчи, чтобы раскрыть высокоструктурированное представление исходной геометрии. Из недостатков - данный метод работает только с трехмерной точечной моделью, получаемой при помощи трехмерного сканера.

Статья Zhang и др. [103] посвящена разработке собственного подхода к реконструкции трехмерных каркасных моделей криволинейных объектов на основе трех ортогональных проекций. Рассматриваемый авторами алгоритм не накладывает строгих ограничений на ориентацию квадратичных поверхностей, что позволяет обрабатывать более широкий диапазон объектов по сравнению с предыдущими методами, которые часто требовали, чтобы объекты были

ориентированы вдоль координатных осей. Методология статьи включает генерацию 3D конических ребер из 2D проекций, использование пятиточечного метода для получения алгебраических представлений, а также эффективные алгоритмы для повышения производительности. Реализация представленного алгоритма выполнена с акцентом на обработку именно криволинейных элементов, что делает подход универсальным для разных геометрических форм.

Gong и др. [68] предложили метод реконструкции 3D-каркасов криволинейных объектов на основе деревьев решений. Деревья решений ищут проективные свойства в трёх видах и затем строят 3D-рёбра, но сложность алгоритма высока из-за дублирования деревьев. Это достаточно эффективный метод реконструкции 3D криволинейных каркасных моделей без ограничений на расположение пространственных коник. Метод может сократить пространство поиска для выявления всех совпадающих проекций в соответствии с проекционными свойствами пространственных рёбер. Также он может легко генерировать различные типы коник в произвольных позициях, используя рациональную форму Безье для коник.

1.2.2. Реконструкция 3D CAD-моделей

В работах [67,69] авторы предлагают B-гер-ориентированный подход, использующий гибридную каркасную модель и граф связей (LRG) для распознавания квадратичных поверхностей (например, сфер, цилиндров, конусов) через сопоставление шаблонов на основе подсказок из проекций. Метод эффективно снижает время обработки, избегая комбинаторного поиска и перебора, характерных для традиционных методов, и способен обрабатывать сложные случаи взаимодействия квадратичных поверхностей. Алгоритм был реализован в системе TiGEMS® 6.0 и успешно протестирован на реальных механических деталях, демонстрируя высокую скорость и точность восстановления. Основные преимущества включают расширение области применения за счет поддержки квадрик кривых и упрощение процедуры за счет одновременного восстановления геометрии и топологии поверхностей. Высокая алгоритмическая сложность и ограничения в обработке криволинейных рёбер

делают эти методы непрактичными.

В статье [91] рассматриваются преимущества и ограничения известных подходов восстановления граней по каркасной модели, а также представляются результаты тестов, которые иллюстрируют достоинства и недостатки широко используемого метода, основанного на алгоритме Дейкстры. Авторы пришли к выводу, что основной причиной многих неудачных случаев является предположение алгоритма о том, что стоимость прохождения ребра является фиксированной. Предлагается новый алгоритм полиномиальной сложности для нахождения граней в каркасных моделях на основе алгоритма кратчайшего пути. Этот алгоритм может быть адаптирован для любых задач нахождения наименьшего контура в графах, где стоимость прохождения ребра не фиксирована, а зависит от контекста.

Zakharov и др. [102] представили алгоритм спектрального сопоставления графов для построения 3D-граней моделей из каркасов. В этом методе грани формируются на основе замкнутых петель. Однако эти методы не решают проблему копланарности двух вложенных петель. Представленный ими алгоритм содержит вероятностный подход для трехмерной реконструкции визуальной среды различных сцен на основе спутниковых и аэрофотоснимков. В описанной модели используются условные вероятности появления узнаваемых признаков и реконструируемых объектов.

Некоторые из этих подходов [70,91,100] работают только для 3D-моделей с плоскими поверхностями - с многогранниками.

В работе [62] авторы предлагают метод автоматической реконструкции 3D-моделей из 2D-орфографических проекций с использованием операций выдавливания и вращения. Алгоритм работает в два этапа: на первом этапе преобразуются внутренние геометро-графические данные (извлеченные из формата DXF) в ортогональные проекции, а на втором — выполняется выдавливание внешних профилей и их последующая обработка с использованием операций вычитания и пересечения. Результирующая модель получается путем объединения обработанных объемных элементов. Метод не требует

вмешательства пользователя и демонстрирует высокую эффективность на различных примерах, включая сложные детали. Основные преимущества — автоматизация процесса и возможность обработки объектов как с внутренними элементами, так и без них.

В статье [78] представлен подход для восстановления 3D-моделей пересекающихся тел вращения (например, цилиндров, конусов) из двумерных ортогональных проекций. Примером такой детали является тройник для трубного соединения. В отличие от традиционных CSG-ориентированных методов (CSG - constructive solid geometry, конструирование трехмерных моделей при помощи булевых операций над геометрическими примитивами и их производными), которые ограничены изолированными объектами, предложенный подход использует подсказки для распознавания взаимодействующих тел вращения и их оптимальной реконструкции. В этом подходе окружности на главном виде служат ключами, а вершины и рёбра в других видах сопоставляются с окружностями для построения объектов. Метод основан на геометрическом анализе проекций и позволяет автоматизировать процесс создания 3D-моделей, сокращая время проектирования и улучшая качество конечного результата. Основное достоинство метода — расширение области применения CSG-подходов за счет поддержки сложных случаев пересечения объектов, что делает его применимым в промышленном проектировании и обработке устаревших чертежей.

В [94] авторы представили подход для автоматического восстановления 3D-моделей из 2D ортогональных проекций инженерных чертежей, основанный на извлечении элементов знаний и методов теории графов для представления 3D-моделей.. Метод сочетает элементы вариационной геометрии, теории графов и интеллектуального анализа данных, чтобы имитировать процесс, который использует опытный инженер. Основные этапы включают создание локальных графов, представляющих схемы размеров для каждого вида, их объединение в глобальный граф размерной схемы и преобразование данных из глобального графа в 3D-модель. Ключевые особенности — использование топологических отношений и анализа размерных схем, что позволяет автоматизировать процесс и

повысить точность восстановления. Метод применим для обработки инженерных чертежей и может быть использован в CAD/CAM системах для ускорения проектирования и обработки устаревших данных. Однако создание системы знаний, включающей все типы особенностей в 2D чертежах, затруднительно.

В статье [89] представлен метод автоматического восстановления 3D-моделей из эскизов, основанный на извлечении признаков из линейных чертежей. В отличие от традиционных методов маркировки линий, которые сложно применимы к криволинейным объектам, предложенный подход использует предопределенные признаки (например, кубоиды, цилиндры, отверстия и фаски) для их распознавания и преобразования в 3D-модели. Метод имитирует процесс распознавания, который использует человек, анализируя примитивы и сложные формы на эскизах. Ключевые преимущества — возможность обработки криволинейных объектов, которые трудно восстановить с помощью традиционных методов, и расширение области применения за счет работы с более сложными сценами.

В статье [71] представлен подход для автоматического восстановления 3D-моделей из фотографий 2D-чертежей с использованием библиотеки OpenCV. Метод позволяет извлекать геометрические признаки из изображений и преобразовывать их в 3D CAD модели в формате SCAD, которые затем могут быть переведены в облака точек для дальнейшей обработки, например, для FEA-анализа или производства. Основная новизна работы заключается в использовании фотографий вместо сканированных или электронных чертежей, что особенно актуально для устаревших инженерных проектов, где исходные чертежи могут быть утеряны. Алгоритм реализован в виде Jupyter Notebooks и доступен на GitHub, что делает его удобным для использования и дальнейшего развития. Ограничением метода является пока невозможность обработки невидимых линий на чертежах, но за счет своей универсальности и простоты он демонстрирует высокий потенциал для применения в механической, аэрокосмической и биомедицинской областях.

Работа [60] посвящена проблеме преобразования рукописных и

отсканированных инженерных чертежей в 3D-модели с использованием неконтролируемых и гибридных методов векторизации. Основное внимание уделено разделению слоев чертежей (например, прямых линий, дуг, окружностей) с помощью глубокого обучения и последующему восстановлению 3D-моделей. Предложен алгоритм предобработки, который включает кластеризацию видов, удаление шума с использованием U-net и скелетизацию изображений для улучшения процесса векторизации. Гибридный подход (сочетание глубокого обучения и традиционных методов) позволил достичь 82% точности по метрике IoU при разделении чертежей на четыре слоя. Основные достижения — разработка устойчивого метода для автоматизированной векторизации и постобработки, способного обрабатывать сложные чертежи с пересекающимися геометрическими примитивами. Работа демонстрирует потенциал применения этих методов в CAD-системах и реконструкции устаревших чертежей. Однако методы на основе распознавания особенностей недостаточно универсальны и работают только с ограниченными типами 2D чертежей и 3D-форм. Эти ограничения мешают эффективной обработке сложных CAD-моделей.

Статья С. И. Вяткина и Б. С. Долговесова [9] предлагает метод реконструкции 3D-объектов по данным одного ракурса изображения. В отличие от существующих подходов, этот метод использует силуэт объекта и многоуровневую карту высот, что позволяет избежать ограничений по глубине и значительно сократить объём вычислений. Алгоритм основан на минимизации энергетического функционала и позволяет генерировать 3D-модели с высоким разрешением в интерактивном режиме. Метод был реализован с использованием GPU, что обеспечило быстрое выполнение вычислений. В статье также обсуждаются преимущества предложенного подхода, такие как двузначная точность разрешения, оптимальность вычислений и снижение требований к памяти. Этот метод особенно эффективен для интерактивного 3D-моделирования и превосходит многие известные подходы по правдоподобности реконструкции и минимальности пользовательского ввода. Из существенных ограничений данного подхода стоит отметить, что авторы предполагают симметричность объекта

относительно плоскости изображения (иначе алгоритм требует дополнительных обработок, например, зеркального отражения карты глубины). Метод особенно эффективен для быстрого моделирования и веб-приложений, где важны скорость и минимальный пользовательский ввод.

Статья [99] тоже предлагает метод 3D-реконструкции зданий с использованием технологии монокулярного зрения, что особенно полезно для мониторинга строительства и оценки повреждений в гражданском строительстве. Авторы, Boqiang Xu и Chao Liu, предлагают подход, который использует геометрические характеристики зданий для нормализации модели облака точек, делая её длину, ширину и высоту параллельными осям мировой системы координат, а также восстанавливая абсолютный масштаб модели. Этот метод, основанный на цифровой обработке изображений, обеспечивает высокую точность и автоматизацию, устраняя необходимость ручной калибровки камеры. В статье представлены теоретические основы, шаги реализации и результаты полевого эксперимента, подтверждающие эффективность метода. Этот подход предлагает удобное и надёжное решение для 3D-реконструкции зданий с большим потенциалом для практического применения. Метод сочетает компьютерное зрение, геометрические ограничения зданий и данные с дронов для автоматизированной, точной реконструкции с сохранением масштаба.

Недостатки предложенных подходов:

1. Ограниченная автоматизация:

a. Ручная обработка данных (например, коррекция чертежей, уточнение размеров) требуется на разных этапах (например, [94] использует вариационную геометрию и графовые методы, что требует ручного вмешательства для анализа топологических отношений).

b. Трудоемкость процесса, особенно для сложных объектов (например, [67,69] использует гибридные каркасные модели, которые требуют ручного уточнения для обработки несовершенств чертежей).

c. Отсутствие поддержки криволинейных ребер (например, в [70])

d. Необходимость ручного ввода (например, в [99] существует

необходимость ручного ввода масштабного коэффициента, хоть этот метод и может восстанавливать абсолютный масштаб через данные дронов или высоту этажа, эта информация (масштабный коэффициент) требует предварительного задания (например, известной высоты этажа) или дополнительных шагов, иначе модель остаётся масштабируемой лишь относительно)

2. Зависимость от качества входных данных:

а. Недостаток гибкости при обработке шумных или некачественных чертежей (например, [69] требует идеальных чертежей для корректного распознавания квадратических поверхностей).

б. Точность восстановления снижается при наличии ошибок или недостающих элементов в чертежах (например, [82] обрабатывает зашумленные данные, но требует локальных параметров для аппроксимации поверхностей).

3. Ограниченная функциональность:

а. Сложности с обработкой сложных геометрий (например, [69] ограничен квадратическими поверхностями и не справляется с нестандартными формами или [99], использующий предположение о наличии строгих геометрических закономерностей (параллельность/перпендикулярность линий, равномерная этажность) и требующий доработки для зданий с криволинейными формами или сложной архитектурой).

б. Недостаток поддержки сложных взаимодействий граней (например, [94] использует топологические методы, которые ограничены простыми объектами).

с. Поддержка только специализированным ПО (например, в [66] весь алгоритм создан и протестирован для пакета MatLab)

4. Вычислительная сложность:

а. Комбинаторный поиск и перебор требуют высоких вычислительных ресурсов (например, [69] использует сложные алгоритмы

для анализа промежуточных каркасов, что увеличивает время обработки).

б. Медленная обработка больших или сложных объектов (например, [67] требует дополнительных шагов для восстановления граничных моделей).

5. Требования к пользователю:

а. Необходимость глубоких знаний в геометрическом моделировании (например, [94] требует опытного пользователя для анализа вариационной геометрии).

1.2.3. Модели глубокого обучения

В последнее время растёт интерес к применению методов глубокого обучения для трехмерного моделирования.

Li и др. [79,80] предлагают использовать следующий подход: пользователь делает набросок итоговой формы, а их разрабатываемая система преобразует входные штрихи в последовательность команд, выраженных на упрощенном языке САПР. При выполнении эти команды воспроизводят набросанный объект. Они интерпретируют моделирование САПР на основе эскизов как задачу перевода последовательностей, для чего авторами используется нейронная сеть Transformers[61]. Учитывая последовательность штрихов в качестве входных данных, вводится задача группировки штрихов, соответствующих отдельным операциям САПР. Авторы объединяем группировку штрихов с геометрической подгонкой параметров операций, так что промежуточные группы геометрически корректируются перед повторным использованием в качестве контекста для последующих шагов в выводе последовательности.

Wu и др. [98] представляют нейросетевую модель для генерации CAD-моделей в виде последовательности CAD-операций, таких как выдавливание, вращение, булевы операции, на основе архитектуры Transformer. В отличие от традиционных 3D-генеративных моделей, работающих с вокселями, облаками точек и мешами, их нейросеть DeepCAD воспроизводит процесс проектирования, создавая редактируемые и параметрические модели, готовые к использованию в CAD-системах (SolidWorks, AutoCAD). Для обучения модели был создан большой

датасет из 178,238 CAD-моделей с их историей операций, который доступен для дальнейших исследований. Основные преимущества — возможность создания чистых, детализированных и разнообразных 3D-форм, а также прямая интеграция с CAD-инструментами.

В работах [74,77,96] Willis и др. предлагают новый подход к автоматической генерации технических эскизов (CAD sketches) с использованием глубоких нейронных сетей. Авторы обучают модель на большом датасете CAD-операций, включающем такие параметры, как размерность, геометрические ограничения и булевы операции, что позволяет модели создавать реалистичные эскизы, которые могут быть использованы для параметрического CAD-моделирования. Ключевая особенность работы — возможность восстановления CAD-моделей из зашумленных 3D-сканов через генерацию промежуточных эскизов, что значительно упрощает процесс реверс-инжиниринга. Метод также может быть применен для автодополнения пользовательских команд и ускорения создания сложных инженерных чертежей. Основные достоинства — генерация редактируемых и параметрических эскизов, что делает модель полезной как для инженеров, так и для дизайнеров. Работа открывает новые перспективы для автоматизации CAD-процессов и интеграции искусственного интеллекта в промышленное проектирование.

В [97] авторами представлен крупный датасет CAD-моделей с их историей операций, созданный на основе данных из Autodesk Fusion 360, и среда для программной реконструкции этих моделей через машинное обучение. Датасет включает 21,000 моделей с последовательностями операций, такими как выдавливание, вращение и булевы операции, что делает его полезным для обучения моделей, способных автоматизировать процесс CAD-моделирования. Авторы также предоставляют инструменты для анализа и визуализации данных, включая графы зависимостей операций, что открывает новые возможности для обучения с подкреплением и генеративных моделей в области проектирования. Основное значение работы — создание базовой инфраструктуры для исследований в области программной реконструкции CAD-моделей, что может

привести к ускорению разработки и улучшению качества проектов в механической, аэрокосмической и производственной отраслях.

Wang и др. [93] представили метод для восстановления 3D-граничных CAD-моделей из 2D-проекций, основанный на глубоком обучении и топологическом анализе. Основная задача — реконструировать геометрические и топологические отношения между ребрами и гранями в 3D-модели, используя только одну или несколько 2D-проекций. Для обучения модели используется гибридный подход, включающий трехмерные граничные сети и задачи предсказания циклов, что позволяет модели эффективно обрабатывать сложные случаи, такие как вложенные циклы и многозначные проекции

Статьи [90,101] представляют метод автоматической реконструкции поверхностей на основе классификации инвариантности (например, плоскостей, цилиндров, сфер) с использованием трансферного обучения на предобученной CNN (AlexNet). Входные данные — облака точек из Skin Model Shapes или реальных измерений — разделяются на части через поверхностные вариации, затем каждая часть классифицируется и аппроксимируется в идеальную поверхность. Ключевые преимущества — низкие вычислительные затраты, автоматизация процесса и возможность работы с зашумленными данными без необходимости в локальных параметрах (например, нормалях). Метод тестируется на открытом датасете и демонстрирует высокую точность в задачах обратного проектирования и геометрической обработки.

В статьях [8, 75] описывается подход к восстановлению трехмерной модели твердых объектов по одному аэрокосмическому изображению, состоящий из двух основных этапов: извлечения семантической информации (получаемой при помощи технологий машинного обучения) и восстановления геометрии. Авторы рассматривают применение искусственных нейронных сетей для семантической сегментации компонентов зданий и эталонных объектов, а также описывают математические преобразования для расчета фотометрических параметров и преобразования пространственных координат. Предложенные алгоритмы учитывают различные условия наличия определенных классов на изображении,

выявленные в результате машинного обучения, а также условия отсутствия метаданных о пространственном разрешении, пространственной ориентации съемки и положении Солнца (угол съемки, азимут сканирования и т.д.). В работе также представлены примеры применения метода для восстановления 3D-моделей зданий по спутниковым и аэрофотоснимкам, обсуждаются уровни детализации моделей (LoD1(только призматическая форма реконструируемых объектов) и LoD2(содержащая криволинейные поверхности)) и их применение в задачах городского планирования. Особое внимание уделено подходу, основанному на данных, который использует сверточные нейронные сети для сегментации и восстановления сложных геометрических форм. Авторы также отмечают возможности метода при работе с изображениями различного качества и спектральных характеристик, подчеркивая его перспективность для автоматизации задач 3D-реконструкции. Этот подход позволяет восстанавливать 3D-модели даже при ограниченных данных, что делает его универсальным и практичным для различных задач.

Статья [81] посвящена разработке системы автоматизированного создания трехмерных моделей строительных объектов на основе глубокого обучения. Автор рассматривает технологию 3D-реконструкции как ключевой инструмент для упрощения понимания архитектурных чертежей и повышения эффективности управления строительными проектами. В статье представлен алгоритм, который преобразует двумерные архитектурные планы (DXF) в трехмерные модели (3DS), используя методы компьютерного зрения и искусственного интеллекта. Система, разработанная в рамках исследования (DLR-P), интегрируется с технологией BIM (Building Information Modeling) для автоматизации контроля за ходом строительства. Практическая проверка системы на примере строительной площадки в Китае показала, что среднее время реконструкции составляет 61 секунду, что делает её пригодной для задач управления строительством. Результаты демонстрируют, что предложенный подход экономически эффективен и упрощает процесс контроля строительных работ по сравнению с традиционными методами.

В предложенных методах реконструкции трехмерной детали при помощи систем машинного обучения есть следующие недостатки:

1. Ограниченная обобщаемость:

а. Нейронные сети часто специализируются на конкретных классах объектов или типах поверхностей, что ограничивает их применение в широком спектре задач (например, в [80] модели работают только с параметрическими операциями, такими как выдавливание и скругление, а метод, описанный в [8, 75] в основном позволяет создавать модели, содержащие только призматические формы)

б. Требуется большое количество данных для обучения, что может быть трудоемким и дорогостоящим (например, в [98] использовался датасет из 178,238 моделей).

2. Проблемы с точностью:

а. Нейронные сети часто генерируют приближенные результаты (например, облака точек или сетчатые поверхности), которые требуют дополнительной постобработки для получения точностей CAD-моделей (например, [79] преобразует изображения в облака точек, что не подходит для прямого использования в CAD-системах).

б. Обработка скрытых линий и сложных взаимодействий граней остается проблемой, что снижает качество реконструкции (например, в [93] отмечаются сложности с вложенными циклами).

с. Зависимость от эталонных объектов. В [8, 75] авторы отмечают, что в случаях отсутствия метаданных метод использует эталонные объекты для расчета параметров. Это может привести к ошибкам, если эталонные объекты не соответствуют реальным условиям.

3. Зависимость от качества входных данных:

а. Шумные или некачественные входные данные (например, ручные эскизы, отсканированные чертежи) могут привести к ошибкам в реконструкции (например, в [80] отмечается необходимость чистого ввода для точного распознавания примитивов, в [8, 75] авторы отмечают, что при

недостаточном разрешении, плохих условиях освещения или отсутствии метаданных точность реконструкции значительно снижается, а в [81] эффективность работы алгоритма целиком зависит от качества и углов съемки изображений - при плохом освещении, заслонениях или низком разрешении камер могут возникать ошибки в реконструкции).

б. Недостаток структурированных данных в чертежах (например, отсутствие размеров или корректных проекций) может усложнить процесс восстановления (например, в [79] отсутствие скрытых линий на чертежах требует ручной коррекции).

4. Вычислительная сложность:

а. Обучение и инференс моделей требуют значительных вычислительных ресурсов, особенно для сложных объектов или больших датасетов (например, в [98] используется архитектура Transformer, что требует дорогостоящего оборудования, в [81] автор также упоминает, что для обучения и работы модели MVSNet, а также для обработки данных BIM требуются производительные компьютеры).

б. Время обработки может быть заметным, что затрудняет использование в реальном времени.

5. Ограничения в функциональности:

а. Нейронные сети часто поддерживают ограниченный набор операций (например, [98] работает только с базовыми операциями, такими как выдавливание и булевы операции).

б. Недостаток гибкости в обработке нестандартных форм или сложных геометрий (например, [96] ограничен генерацией технических эскизов и не поддерживает свободные формы, аналогично и [8, 75]).

6. Проблемы интеграции с CAD-системами:

а. Генерация непараметрических моделей (например, мешей или облаков точек) затрудняет их редактирование в CAD-программах (например, [80] генерирует параметрические модели, но требует дополнительных шагов для их интеграции).

б. Отсутствие совместимости с некоторыми CAD-форматами или инструментами может ограничить применение моделей в промышленных задачах.

7. Требования к пользователю:

а. Некоторые методы требуют значительного участия пользователя для коррекции или уточнения результатов (например, в [97] отмечается необходимость ручного вмешательства для обработки сложных случаев).

б. Недостаток интуитивности в интерфейсах может затруднить использование моделей непрофессионалами.

Методы на основе глубокого обучения демонстрируют впечатляющие результаты, но, насколько известно автору, ни одна модель не принимает только чертежи, содержащие ортогональные проекции на вход. Кроме того, такие методы не достигают 100% успешной реконструкции (например, создание выдуманных моделей, пропуск граней), и иногда использование сгенерированных моделей в промышленных процессах сложнее, чем создание с нуля.

1.3 Основные понятия

1.3.1 Каркасная модель как способ описания формы тела в R^3 .

Считаем, что тело конечно и ограничено кусками гладких поверхностей, которые в общем случае могут не иметь аналитического описания. Любой отрезок прямой или кривой, являющийся общей частью границы каких-либо двух кусков, назовём ребром каркасной модели. Если при переходе через ребро внешняя нормаль к поверхности тела как вектор-функция координат $n(x,y,z)=(n_x(x,y,z),n_y(x,y,z),n_z(x,y,z))$ терпит разрыв, то такое ребро называют конструктивным ребром каркаса. Соответственно, ребро называют неконструктивным, если при переходе через него нормаль изменяется непрерывно. Общие точки нескольких рёбер назовём вершинами каркаса. Очевидно, что любая вершина является общей граничной точкой нескольких кусков поверхности тела. Совокупность множества вершин и множества рёбер назовем каркасной моделью пространственного геометрического тела.

Иногда введенная таким образом каркасная модель недостаточно полно отражает форму тела. В этом случае некоторые куски поверхности нужно разбить на части лежащими на них характерными линиями, которые позволяют лучше распознать форму моделируемого тела (в частности, таковыми могут являться линии очерка на криволинейных поверхностях, то есть линии, в каждой точке которых плоскость, касательная к поверхности, параллельна заданному вектору проецирования). Каркасная модель при этом дополняется новыми вершинами и неконструктивными рёбрами.

Теперь перейдем от данного эмпирического определения каркасной модели к её математическому заданию. Чтобы описать геометрические характеристики каркаса, нужно задать координаты в R^3 всех его вершин и в том или ином виде (например, параметрически) уравнения линий, отрезками которых являются рёбра каркаса (далее будем называть эти линии несущими линиями рёбер). Но, кроме этого, важны топологические связи между множеством вершин и множеством рёбер каркаса. С этой точки зрения каркасную модель удобно описывать как граф.

Множество вершин этого графа есть множество точек в R^3 , являющихся вершинами каркаса согласно эмпирическому определению. Две вершины V_i и V_j графа смежны, если как точки в R^3 они принадлежат одному ребру каркаса; тогда пара (V_i, V_j) -ребро графа. Очевидно взаимно-однозначное соответствие между рёбрами каркаса и рёбрами графа; поэтому, говоря далее о смежности вершин или инцидентности вершины каркаса ребру, будем понимать под этим соответствующие отношения между элементами графа, определяющего топологию модели.

Окончательно можно определить каркасную модель тела следующим образом:

$$K = \{V, L, E\}, (1)$$

где $V = \{V_i = (X_i, Y_i, Z_i) \in R^3, i = 1 \dots N_v\}$ - множество вершин модели;

$L = \{L_i = (X_j(t), Y_j(t), Z_j(t)) \in R^3, T_{0j} \leq t \leq T_j, j = 1 \dots N_L\}$ - множество несущих линий, заданных параметрически;

$E = \{E_k = (V_{1k}, V_{2k}, L_k), V_{1k}, V_{2k} \in V, L_k \in L, k = 1 \dots N_E\}$ - множество рёбер модели.

1.3.2 Описание проекции тела на некоторую плоскость

Рассмотрим проекцию тела на некоторую плоскость, называемую далее картинной плоскостью или плоскостью проекции. Конструктивные рёбра каркаса и рёбра очерка криволинейных поверхностей проецируются в отрезки некоторых линий на картинной плоскости, которые могут накладываться друг на друга или пересекаться; линейные ребра (отрезки прямых), параллельные вектору проецирования, вырождаются в точку. Вершины каркаса проецируются в некоторые точки. При проецировании инцидентность вершины и ребра, а следовательно, и смежность вершин, не нарушаются, но проекции вершин могут также накладываться друг на друга. Из всего этого следует, что 2D модель проекции отличается от 3D каркасной модели лишь размерностью пространства, в котором заданы координаты точек-вершин и параметры несущих линий рёбер, и смысловым наполнением понятий «вершина» и «ребро». Под вершиной проекции будем понимать некоторую точку, заданную своими координатами на картинной плоскости. Под ребром проекции понимаем отрезок прямой или кривой линии на

этой плоскости (геометрический смысл), соединяющий две вершины проекции (топологический смысл).

Математическая модель проекции:

$$PR = \{P, C, F\}, (2)$$

где $P = \{P_i = X_i, Y_i \in R^2, i = 1..M_p\}$ - множество вершин проекции;

$C = \{C_j = (X_j(t), Y_j(t), Z_j(t)) \in R^2, T_{0j} \leq t \leq T_j, j = 1..M_c\}$ - множество несущих линий;

$F = \{F_k = (P_{1k}, P_{2k}, C_k), P_{1k}, P_{2k} \in P, C_k \in C, k = 1..M_f\}$ - множество ребер проекции

1.3.3 Моделируемые тела

Зададим класс тел, каркасная модель которых может быть синтезирована разрабатываемым алгоритмом:

1. будем считать, что тело ограничено кусками плоскостей и криволинейных поверхностей второго порядка, а именно: эллипсоидов, эллиптических цилиндров и эллиптических конусов;

2. пусть несущими линиями ребер каркасной модели, определенных выше как общие границы различных отсеков поверхности, будут лишь прямые и эллипсы, плоскость которых произвольно ориентирована в пространстве.

При прямолинейном ортогональном проецировании такого тела несущими линиями на плоскости проекции будут также прямые и эллипсы произвольной ориентации. Таким образом, определив класс моделируемых тел, мы выявили класс исходных данных для задачи: ребрами каждой проекции могут быть только отрезки прямых и эллиптические дуги.

В системе геометрии и графики “Китеж” [19, 47] реализован алгоритм, синтезирующий каркасную модель для тел, ограниченных плоскостями и круговыми коническими и цилиндрическими поверхностями, причем ось любой такой поверхности должна быть перпендикулярна одной из координатных плоскостей. Ребрами каркаса могут быть лишь отрезки прямых и дуги окружностей; последние получаются при сечении цилиндра или конуса плоскостью, перпендикулярной оси. Ребрами проекций также могут являться только отрезки и дуги окружностей.

Вышеуказанные ограничения на входные данные - это попытка расширить

класс моделируемых тел. В частности, алгоритм должен создавать модели тел, частью границы которых является поверхность сферы, наклонного цилиндра или конуса.

1.3.4 Информационная модель задачи.

Модель проекций (исходные данные).

Координаты вершин проекций на чертеже:

$$P_i = (X_i, Y_i), i = 1 \dots M_p. (3)$$

Уравнения несущих линий 2-го порядка (эллипсов):

$$EL_j: A_{11j}(X - X_{cj})^2 + 2A_{12j}(X - X_{cj})(Y - Y_{cj}) + A_{22j}(Y - Y_{cj})^2 = 1 (4)$$

или

$$\begin{aligned} X &= a_j \cos \Phi_j \cos t - b_j \sin \Phi_j \sin t + X_{cj}, \\ Y &= a_j \sin \Phi_j \cos t + b_j \cos \Phi_j \sin t + Y_{cj}, j = 1 \dots M_{el}. \end{aligned} (5)$$

Параметры ребер проекций:

$$R_k = (N_{p1k}, N_{p2k}, N_{elk}), 1 \leq N_{p1k}, N_{p2k} \leq M_p, 0 \leq N_{elk} \leq M_{el}, 1 \leq k \leq M_r, (6)$$

где N_{p1k}, N_{p2k} - номера вершин, инцидентных ребру,

N_{elk} - номер несущей линии (0 - прямая).

Модель каркаса (выходные данные).

Координаты вершин каркаса:

$$V_i = (X_i, Y_i, Z_i), i = 1 \dots N_v, (7)$$

Уравнения несущих линий 2-го порядка (эллипсов):

L_j : уравнение плоскости + уравнение поверхности второго порядка (эллипсоида, эллиптического цилиндра или конуса) или параметрически:

$$\begin{aligned} X &= X_j(t), \\ Y &= Y_j(t), \\ Z &= Z_j(t) \end{aligned} (8)$$

где $X_j(t), Y_j(t), Z_j(t)$, - линейные комбинации функций $\{1, \sin t, \cos t\}$, коэффициенты которых связаны между собой,

или в векторном виде:

$$r_j(t_1, t_2) = r_{cj} + t_1 r_{1j} + t_2 r_{2j}, t_{12} + t_{22} = 1, (9)$$

где r_{cj} - радиус-вектор центра эллипса,

r_{1j} и r_{2j} - векторы полуосей эллипса, $j = 1 \dots N_l$.

Параметры ребер каркаса:

$$Ek = (N_{v1k}, N_{v2k}, N_{1k}), 1 \leq N_{v1k}, N_{v2k} \leq M_p, 0 \leq N_{1k} \leq N_l, 1 \leq k \leq N_e, (10)$$

где N_{v1k}, N_{v2k} - номера вершин, инцидентных ребру,

N_{1k} - номер несущей линии (0 - прямая).

Уравнения несущих линий 1-го порядка (прямых) легко получить исходя из координат, инцидентных ребру вершин и поэтому эти уравнения не включены в модели каркаса и проекций явно, а все ребра-отрезки имеют одинаковый номер несущей линии, равный 0.

1.4 Выводы по первой главе

1. Разработана структурная схема решения обратной задачи начертательной геометрии, отличительной особенностью которой является наличие этапа автоматизированного синтеза трехмерной модели с применением алгоритма предварительной обработки входной геометро-графической информации.

2. Составлен аналитический обзор существующих решений, выявлены их недостатки: алгоритмические методы либо существенно ограничивают класс моделируемых тел (например, многогранниками), либо имеют ограниченную автоматизацию (с необходимостью ручной корректировки модели в процессе формирования), либо очень сильно зависят от качества входных данных; варианты же реконструкции при помощи нейросетевых алгоритмов обладают ограниченной обобщаемостью (результат целиком зависит от обучающего датасета), проблемами с точностью реконструируемого объекта (часто генерируются приближенные результаты, а также подобные алгоритмы имеют склонность к формированию вариантов решения, не имеющих общего с исходными данными), огромнейшей вычислительной сложностью (чем больше будет входных параметров, тем больше вычислительной мощности будет требовать нейросеть).

3. Выделен класс моделируемых тел и, как следствие, класс данных, содержащих исходную информацию для алгоритма преобразования проекционных изображений в каркасную модель, сформулирована информационная модель задачи.

ГЛАВА 2. ПОЛУЧЕНИЕ ГЕОМЕТРО-ГРАФИЧЕСКОЙ ИНФОРМАЦИИ ИЗ РАЗЛИЧНЫХ ИСТОЧНИКОВ И ПОДГОТОВКА К ЕЁ ИСПОЛЬЗОВАНИЮ

2.1 Получение геометро-графической информации из различных источников

Любой производственный объект на протяжении всего срока своего существования требует информационного сопровождения, охватывающего все этапы жизненного цикла. Эта задача решается с помощью комплекса взаимосвязанных систем, использующих общую базу данных. Однако при модификации изделия в различных программных средах возникает сложность обмена информацией между ними, так как каждая система может использовать уникальные форматы данных.

Для устранения подобных проблем разработаны международные стандарты IGES [86] и STEP [73], предназначенные для унификации обмена данными о продукте. Однако при работе с геометрическими моделями эти стандарты оказываются неэффективными из-за избыточности информации, необходимости двукратного преобразования форматов и связанных с этим ошибок. Дополнительную сложность создает доступ к внутренним форматам САПР: разработчикам приходится либо изучать официально опубликованные спецификации, либо применять незаконные методы анализа структуры данных, если спецификации закрыты для сторонних пользователей.

Электронная модель любого изделия, используемая в CALS-технологиях, на 80-90% состоит из геометро-графической информации, которая создается конструктором или технологом с помощью соответствующих программных средств, входящих в состав систем автоматизированного проектирования и технологической подготовки производства. Каждая из таких систем имеет свой, только ей присущий, экранный интерфейс, свою внутреннюю структуру данных и свои алгоритмы решения элементарных геометро-графических задач. Это приводит к проблемам передачи данных между составными частями систем

автоматизированного проектирования, к трудностям при составлении прикладных программ конкретного пользователя, к проблемам чтения и интерпретации данных из внутренней структуры конкретной системы. Кроме того, внутренняя структура данных каждой системы является предметом коммерческой тайны фирмы-разработчика системы и тщательно охраняется, и не публикуется.

Для обеспечения информационной совместимости между различными системами существуют два основных способа передачи данных[39]:

- через программы – конвертеры данных, создаваемые по принципу «каждый с каждым», что приводит к необходимости создавать дополнительную программу связи (рисунок 2.1);

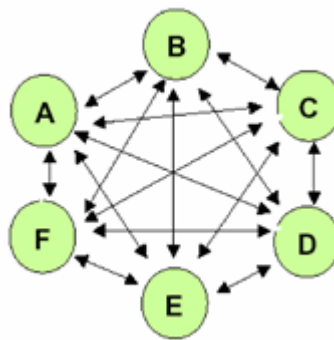


Рисунок 2.1 - Диаграмма связи форматов хранения различных систем (А,В,С - различные форматы хранения САПР).

- через промежуточный нейтральный формат типа IGES [86] или STEP[73] (рисунок 2.2).

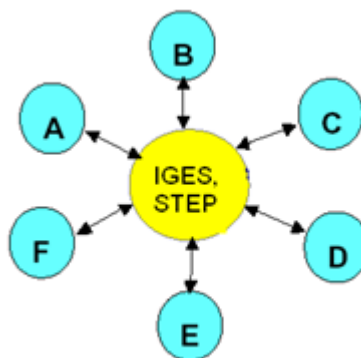


Рисунок 2.2 - Диаграмма связи форматов хранения различных систем при использовании международных стандартов передачи (А,В,С - различные форматы хранения САПР).

При этом фирма-разработчик оставляет за собой право реализовывать то или иное подмножество нейтрального формата, что приводит к разночтениям при

передаче и анализе данных в системе-приемнике.

В то же время в состав любой САПР входит система управления базами данных (СУБД), которую можно использовать для целей оперативного хранения, анализа и передачи геометро-графических данных. Это означает, что внутреннюю структуру данных системы геометрии и графики необходимо привести в соответствие с правилами создания баз данных.

Возникает проблема создания прикладной программы, использующей внутренние форматы хранения данных. Для преодоления противоречия необходимо иметь операторы чтения и записи из внутренней структуры. Их предоставляет либо фирма-разработчик САПР либо данные операции происходят на уровне текстовых фирменных форматов[40].

Далее рассмотрим перечень наиболее значимых форматов в контексте конвертации их в систему поддержки ЖЦИ.

Формат хранения .cdw[10]

Используется в российской САПР «Компас», разработан компанией «Аскон». Предназначен для хранения и использования внутри данной САПР, содержит информацию о чертеже в бинарном виде. Может содержать двухмерные и трехмерные данные, которые представляют собой упорядоченный набор геометрических параметров таких, как: координаты точек, коэффициенты уравнений линий, в том числе высоких порядков, информацию о типе линий и другие неграфические атрибуты.

Достоинством внутреннего формата системы КОМПАС являются компактность хранения, т.е. сравнительно малый объем файла чертежа или модели и отсутствие дублирования данных.

Недостатками формата является то, что все данные о чертеже или модели хранятся в бинарном виде, что неудобно для расшифровки без спецификации, и, следовательно, приводит к трудностям при организации обмена данными с API программами.

Формат хранения .dwg[15]

«DWG (от англ. drawing — чертеж) — бинарный формат файла,

используемый для хранения двухмерных (2D) и трёхмерных (3D) проектных данных и метаданных. Является основным форматом для некоторых САПР-программ (прямая поддержка — например, AutoCAD, nanoCAD, IntelliCAD и его вариаций, Caddie).»[88]

Данный формат разработан компанией Autodesk, спецификации доступны лишь авторизованным группам разработчиков.

Достоинством внутреннего формата системы Autocad является его широкая применяемость среди продуктов компании Autodesk, к недостаткам же относится закрытая спецификация, что делает его трудноприменимым к поставленной задаче.

Формат хранения .dxf[29]

«DXF (англ. Drawing eXchange Format) — открытый формат файлов для обмена графической информацией между приложениями САПР. Был создан фирмой Autodesk для системы AutoCAD. Поддерживается практически всеми CAD-системами на платформе PC.»[64]

К числу достоинств данного формата можно отнести хранение геометро-графической информации в текстовом виде, открытую спецификацию, что облегчает чтение данных прикладной программой, распространенность формата среди различных САПР, последовательность хранения данных внутри файла в виде записей фиксированного формата и объема (80 байт).

К недостаткам данного формата можно отнести закрытую спецификацию хранения трехмерных данных, постоянно расширяющуюся спецификацию хранения двухмерных данных. Из особенностей внутренней структуры хранения данных, минусом формата является то, что в каждой записи содержится один символ или число, что дает разреженность данных и, как следствие, большой объем выходного файла.

Формат хранения .svg[65]

«SVG (от англ. Scalable Vector Graphics — масштабируемая векторная графика) — язык разметки масштабируемой векторной графики, созданный Консорциумом Всемирной паутины (W3C) и входящий в подмножество

расширяемого языка разметки XML, предназначен для описания двумерной векторной и смешанной векторно/растровой графики в формате XML. Поддерживает как неподвижную, так и анимированную интерактивную графику — или, в иных терминах, декларативную и скриптовую. Не поддерживает описание трёхмерных объектов (не путать с имитацией трёхмерности путём светотени). Это открытый стандарт, который является рекомендацией консорциума W3C — организации, разработавшей такие стандарты, как HTML и XHTML. В основу SVG легли языки разметки VML и PGML. Разрабатывается с 1999 года. В 2001 году вышла версия 1.0, в 2011 - версия 1.1, которая остается актуальной до сегодняшнего дня. В настоящее время в активной разработке находится версия 2.»[88]

Достоинствами данного формата являются удобная для программиста структура хранения объектов при помощи тегов, открытая спецификация, возможность сжатия конечного файла в формат .svgz.

Данный формат хранения наследует все недостатки своего «предка» - xml, а так же сложен при чтении больших объектов (например, в картографии), так как для отображения маленького участка необходимо прочесть весь файл.

В [55] описаны некоторые существующие форматы хранения геометро-графической информации, а также проведен анализ объема выходного файла у различных систем для одинакового чертежа. Часть данных приведена на рисунке 2.3 и в таблице 2.1.

Таблица 2.1 - Числовые соотношения средних длин файлов систем[55]

Система	КОМПАС	AUTOCAD	CATIA2D	CATIA3D
КОМПАС	⊕	0,45	0,31	0,125
AUTOCAD	2,22	⊕	0,684	0,278
CATIA2D	3,226	1,46	⊕	0,407
CATIA3D	8	3,6	2,457	⊕

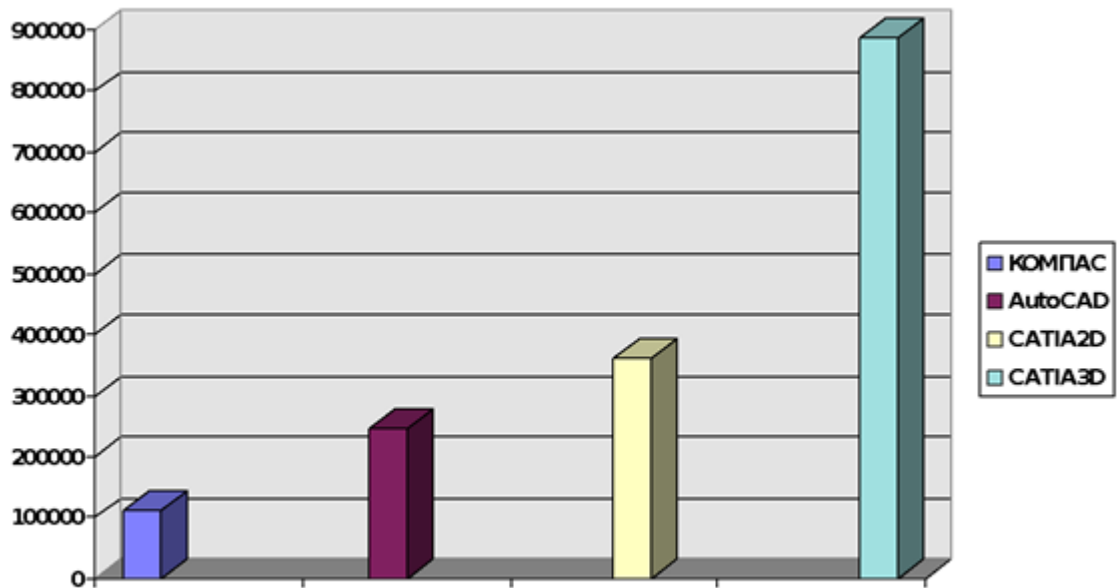


Рисунок 2.3 - Диаграмма соотношения средних длин файлов систем [55]

Исходя из данных, полученных в [55], можно сделать вывод о разреженности внутренней структуры хранения данных у систем Autocad, Catia2d и Catia 3d по сравнению с системой Компас, что увеличивает время доступа к данным и их обработку.

Исходя из поставленной задачи и достоинств и недостатков представленных форматов, выберем два наиболее подходящих - .cdw и .dxf. Рассмотрим подробнее процедуры чтения и записи данных форматов.

Чтение и запись формата .dxf

Так как формат .dxf является текстовым, то и чтение, и запись осуществляются так же, как и у обычного текстового файла с тем отличием, что данный формат имеет определенную структуру, состоящую в основном из 4 секций.

Секция HEADER хранит в себе различные именованные переменные чертежа, например, версию создавшей файл программы, максимальные и минимальные координаты в чертеже и т.д.

Секция TABLES хранит в себе массивы данных, например, таблица слоев со всеми их свойствами, таблица стилей и т.д.

Секция BLOCKS хранит в себе описание примитивов, объединенных в

блоки. Каждый блок имеет уникальный идентификатор, позволяющий вставить данную часть чертежа при помощи команды INSERT в любое место в конечном чертеже. Кроме того, после изменения внутри одного блока, все его включения в основной чертеж так же изменятся.

Секция ENTITIES хранит в себе описание графических примитивов и, по сути, и является отрисовываемой частью.

Все коды графических примитивов можно найти в спецификации формата на сайте компании Autodesk.

Чтение и запись формата .cdw

Чтение бинарного формата .cdw осуществляется из прикладного приложения путем взаимодействия с САПР «Компас» через предлагаемое компанией «Аскон» API и COM-объекты, обеспечивающие связь между двумя запущенными приложениями.

Таким образом, чтение данных из файла формата .cdw возможно произвести лишь средствами САПР «Компас», когда нужный файл открывается внутри САПР и интересующие данные при помощи связующего COM-объекта передаются родительскому процессу прикладного приложения.

Вместе с САПР «Компас» поставляется SDK с примерами использования API, где так же описана процедура чтения данных из открытого документа при помощи обхода всех имеющихся слоев, видов и объектов внутри чертежа.

Запись формата .cdw так же осуществляется при помощи API «Компас» и средствами самой САПР.

Из всего вышеприведенного можно сделать следующий вывод, что при использовании файлов возникает множество препон и ограничений:

1. если файл текстовый (dxf), то он занимает большой объем, по нему неудобно перемещаться для оперативной работы;
2. существуют разночтения при использовании различных версий одного формата (iges,step);
3. если файл бинарный, то вне САПР, в которой он создан, обычно нет возможности вытащить изнутри геометро-графическую информацию (cdw), либо

поддержка будет ограничена старой версией подобного формата (dwg).

Для преодоления описанных недостатков автором предлагается получать геометро-графическую информацию от САПР минуя этап сохранения файла на жесткий диск- например, через COM-объекты[17].

2.2 Получение геометро-графической информации при помощи СОМ-объекта на примере САПР Компас-3D

В случае использования САПР (системы автоматизированного проектирования) «Компас-График» передача геометро-графической информации в прикладную программу может происходить двумя способами: через промежуточные форматы хранения данных или при помощи СОМ-объектов (англ. Component Object Model)[50], осуществляющих взаимодействие двух запущенных приложений в среде Windows. Ниже более подробно рассматривается второй вариант организации передачи данных.

Для упрощения подобного взаимодействия и более эффективной передачи данных компания «Аскон» предоставляет специализированный API(англ. Application Programming Interface) [24]. С его помощью при наличии установленной САПР «Компас-3D» и подключения соответствующих DLL (англ. Dynamic Link Library) становится возможным обмен геометро-графической информацией с прикладной программой без необходимости перекодировки.

Описанный способ взаимодействия позволяет получить геометро-графическую информацию не только из внутреннего формата хранения САПР «Компас-График», но и из любого другого, с которым взаимодействует указанная САПР. После же получения геометро-графической информации прикладной программой её (информацию) можно преобразовать для хранения внутри специализированной базы данных.

Взаимодействие приложений при помощи СОМ-объектов.

С начала 1990 годов корпорация Microsoft разрабатывает технологию, позволяющую создавать гибкие модульные программы таким образом, чтобы отдельные модули можно было писать на разных языках программирования, но чтобы при этом обеспечивалась их полная взаимозаменяемость при использовании в различных программных пакетах. На сегодня эта технология полностью сформирована и называется СОМ (Component Object Model, модель компонентных объектов).

Технология COM описывает методологию реализации компонентов программного обеспечения: объектов, которые могут повторно использоваться, могут быть неоднократно подключены к разным приложениям. Повторное использование компонентов стало логическим следствием эволюции объектно-ориентированного программирования и получило название компонентно-ориентированного подхода. Концепция компонентно-ориентированного программирования предусматривает полное отделение внутренних функций компонента от функций доступа к нему извне. Теперь, обращаясь к компоненту, необязательно знать его внутреннее устройство, для этого достаточно лишь иметь информацию о том, как вызывать его функции. Другими словами, необходимо знать, как взаимодействовать с компонентом и какой у него интерфейс. Такая функциональность в COM достигается за счет поддержки одного или нескольких интерфейсов, которые используются другими программами для доступа к внутренним членам и методам компонента.

Таким образом, интерфейс – это посредник между двумя объектами, обеспечивающий связь между прикладными программами. Интерфейс-ориентированное программирование представляет собой технологию разработки программного обеспечения, жестко нацеленную на использование интерфейсов. Интерфейс является своего рода связующим звеном, предоставленным управляющей программе для доступа к COM-объекту.

Объект COM – конкретный экземпляр COM-класса, заверченный объект с собственными членами данных и методами, который может легко встраиваться в программы или распространяться как отдельный программный продукт. COM-объект представляет собой или DLL-библиотеку или EXE-программу для Windows, которые можно создавать в любой среде программирования, способной поддерживать нужный формат представления. COM-объект может иметь много функций, доступ к которым происходит через его интерфейсы. Любой COM-объект должен иметь по крайней мере один интерфейс IUnknown, хотя на самом деле имеет их значительно больше.

На сегодняшний день технология COM используется практически во всех

серьезных программах. Приведем один пример. Предположим, что пользователю в каком-нибудь отчете нужно поместить электронную таблицу с расчетами, которые ссылаются на определенные параметры в тексте. Чтобы выполнить любые вычисления без использования технологии COM, пришлось бы постоянно переключаться между двумя программами (Word и Excel), а информацию копировать (вырезать и вставлять). При помощи технологии COM можно применять функции электронной таблицы прямо в текстовом редакторе и автоматически форматировать полученный результат. Возможность реализовать операции такого рода называется автоматизацией.

Цель автоматизации состоит в том, чтобы дать возможность программе предоставлять в использование сервисы, которые в ней присутствуют. Основной особенностью автоматизации является возможность комбинировать функции различных специализированных приложений в одном модуле. COM дает возможность программам передавать свою информацию в другие приложения и модули. Если бы каждая программа или приложение-сценарий могли бы поддерживать указатели и процедуру обхода указателей, то проблема была бы решена. Однако в некоторых языках программирования есть определенная трудность с процедурой обхода таблицы указателей. Некоторые из них, например Visual Basic, не поддерживают указатели напрямую. Для решения этой проблемы был разработан специальный интерфейс, который разрешает любым языкам программирования, в том числе таким, как Visual Basic, обращаться к методам COM-компонентов. Этот интерфейс получил название IDispatch.

Реализация операций чтения и записи геометро-графической информации на примере API САПР Компас-График.

Для чтения геометро-графической информации из САПР Компас-График в прикладную программу необходимо проделать следующие операции:

1. Создать COM-объект для взаимодействия между САПР и прикладной программой.
2. Открыть необходимый графический документ для считывания информации в САПР.

3. В активном документе создать итератор для получения доступа ко всем объектам в документе.

4. При помощи итератора обойти все объекты в графическом документе. Блок-схема данного процесса приведена на рисунке 2.4



Рисунок 2.4 - Принцип работы прикладной программы при чтении геометро-графической информации.

2.2.1 Создание СОМ-объекта для взаимодействия с САПР Компас-График

Главным интерфейсом API системы КОМПАС является KompasObject. Получить указатель на этот интерфейс (если быть точным, на интерфейс приложения API 5) можно с помощью экспортной функции CreateKompasObject(). Методы этого интерфейса реализуют наиболее общие функции работы с документами системы, системными настройками, файлами, а также дают возможность получить указатели на другие интерфейсы (интерфейсы динамического массива, работы с математическими функциями, библиотек моделей или фрагментов и различных структур параметров определенного типа).

2.2.2 Открытие графического документа

Другой важный интерфейс API 5 – интерфейс документа модели ksDocument2D.

Получить его можно с помощью функций интерфейса KompasObject:

- ActiveDocument2D – для уже существующего и активного в данный момент документа;
- Document2D – если вы планируете создавать новый двумерный документ.

Для открытия документа с жесткого диска используется метод ksDocument2D::ksOpenDocument.

2.2.3 Создание итератора

Синтаксис: reference ksIterator::CreateIterator(int searchType, reference parent).

Передвижение по модели документа (навигация) производится в соответствии с условиями, заданными в специальном блоке параметров. Он содержит тип движения, определяющий режим перемещения (например, по видам, слоям, всем объектам, объектам заданного типа и т.д.) и указатель комплексного объекта (макроэлемента, слоя, группы) при перемещении по составляющим его объектам. Итератор привязан к конкретному режиму графического редактора (например, документу, виду), поэтому нет возможности использовать один и тот же итератор для навигации в разных видах, штампах и т.п. Итератор сохраняет свое действия до окончания сеанса работы с библиотекой.

2.2.4 Перемещение внутри графического документа

Синтаксис: `reference ksIterator::MoveIterator (reference iterator,unsigned char type);`

Данная функция перемещает текущий указатель итератора (блока параметров навигации) `iterator` на очередной объект модели. Режим позиционирования (тип объектов, участвующих в поиске) определяются при создании итератора.

При позиционировании на именованную группу она автоматически становится текущей (рабочей). К ней будут относиться операции добавления и исключения объектов (при исключении всех объектов из именованной группы она автоматически удаляется).

Блок-схема процесса перемещения по объектам внутри вида представлена на рисунке 2.5

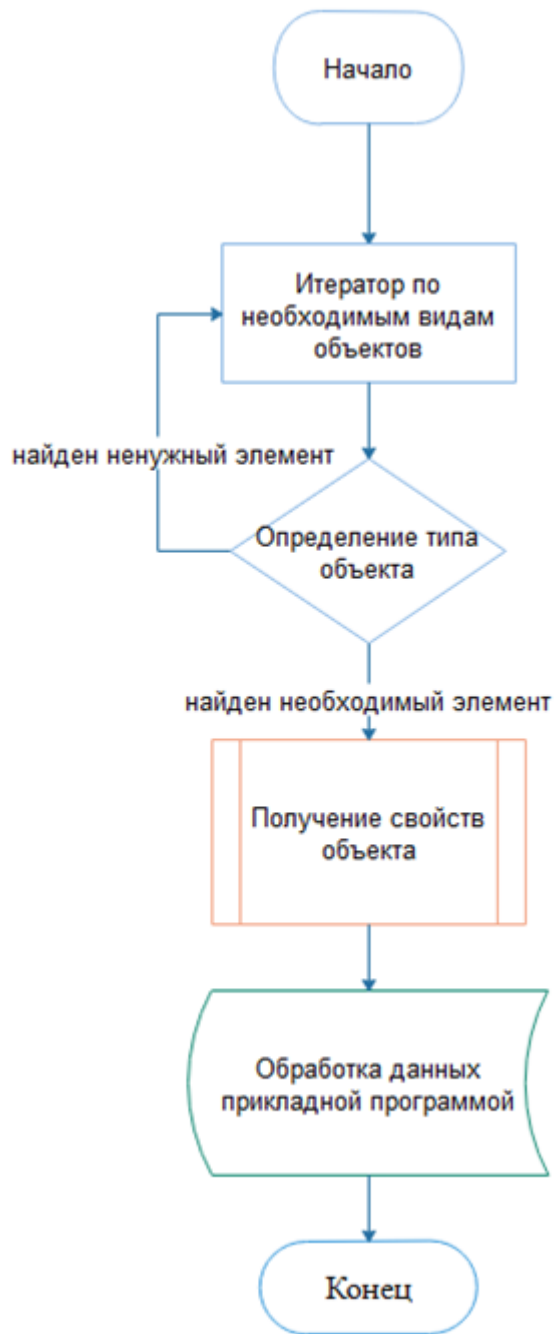


Рисунок 2.5 - Работа итератора.

2.2.6 Чтение свойств объекта

Из объекта, на который указывает итератор, можно получить список свойств текущего объекта. В таблице 2.2 представлены основные объекты, встречающиеся в графических документах. Полный список можно прочитать в [3]

Таблица 2.2 - Примеры типов объектов в Компас-3D.

Идентификатор объекта	Цифровой код	Название объекта	Тип параметров	Структура параметров	Интерфейс параметров
ALL_OBJ	0	все объекты, кроме вспомогательных, которые могут входить в вид			
LINESEG_OBJ	1	отрезок	ALLPARAM	LineSegParam	ksLineSegParam
CIRCLE_OBJ	2	окружность	ALLPARAM	CircleParam	ksCircleParam
TEXT_OBJ	4	текст	ALLPARAM	TextParam	ksTextParam
			Индекс строки текста (начиная с 0)	TextLineParam	ksTextLineParam
POINT_OBJ	5	точка	ALLPARAM	PointParam	ksPointParam
BEZIER_OBJ	8	кривая Безье	ALLPARAM	BezierParam	ksBezierParam
			Индекс узла кривой (начиная с 0)	BezierPointParam	ksBezierPointParam
CUT_OBJ	19	линия разреза	ALLPARAM	CutLineParam	ksCutLineParam
LEADER_OBJ	20	линия-выноска	ALLPARAM	LeaderParam	ksLeaderParam
LINE_OBJ	28	прямая	ALLPARAM	LineParam	ksLineParam
LAYER_OBJ	29	слой	ALLPARAM	LayerParam	ksLayerParam
			VIEW_LAYER_STATE	int (состояние слоя)	ksLtVariant (состояние слоя)
ELLIPSE_OBJ	32	эллипс	ALLPARAM	EllipseParam	ksEllipseParam
NURBS_OBJ	33	кривая NURBS	ALLPARAM,	NurbsParam	ksNurbsParam

			NURBS_CLA MPED_PARA M		
RECTANGLE_ OBJ	35	прямоугольник	ALLPARAM	RectanglePara m	ksRectanglePara m
EQUID_OBJ	37	эквидистанта	ALLPARAM	EquidistantPar am	ksEquidistantPar am
CENTREMARK ER_OBJ	42	Обозначение центра	ALLPARAM	CentreParam	ksCentreParam
RASTER_OBJ	45	Растровый объект	ALLPARAM	RasterParam	ksRasterParam
AXISLINE_OBJ	48	Осевая линия	ALLPARAM	AxisLinePara m	ksAxisLinePara m

2.2.7 Запись во внутренний формат системы управления данными

После получения и обработки всей геометро-графической информации прикладной программой, производится запись в систему управления данными.

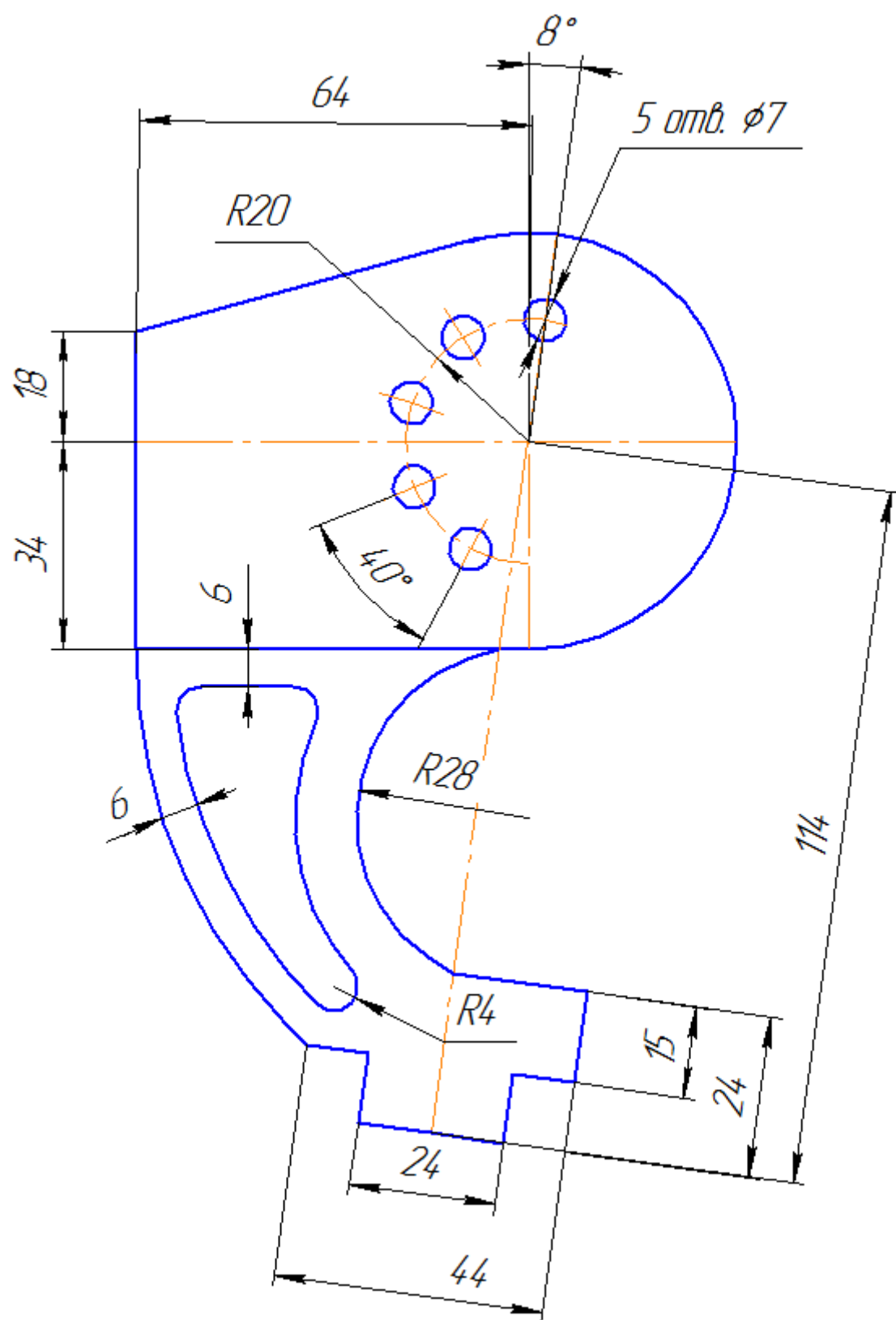
2.2.8 Заккрытие документа

По окончании работы с документом вызывается функция `int ksDocument2D::CloseDocument (reference sheet)`, где `sheet` – указатель на закрываемый документ.

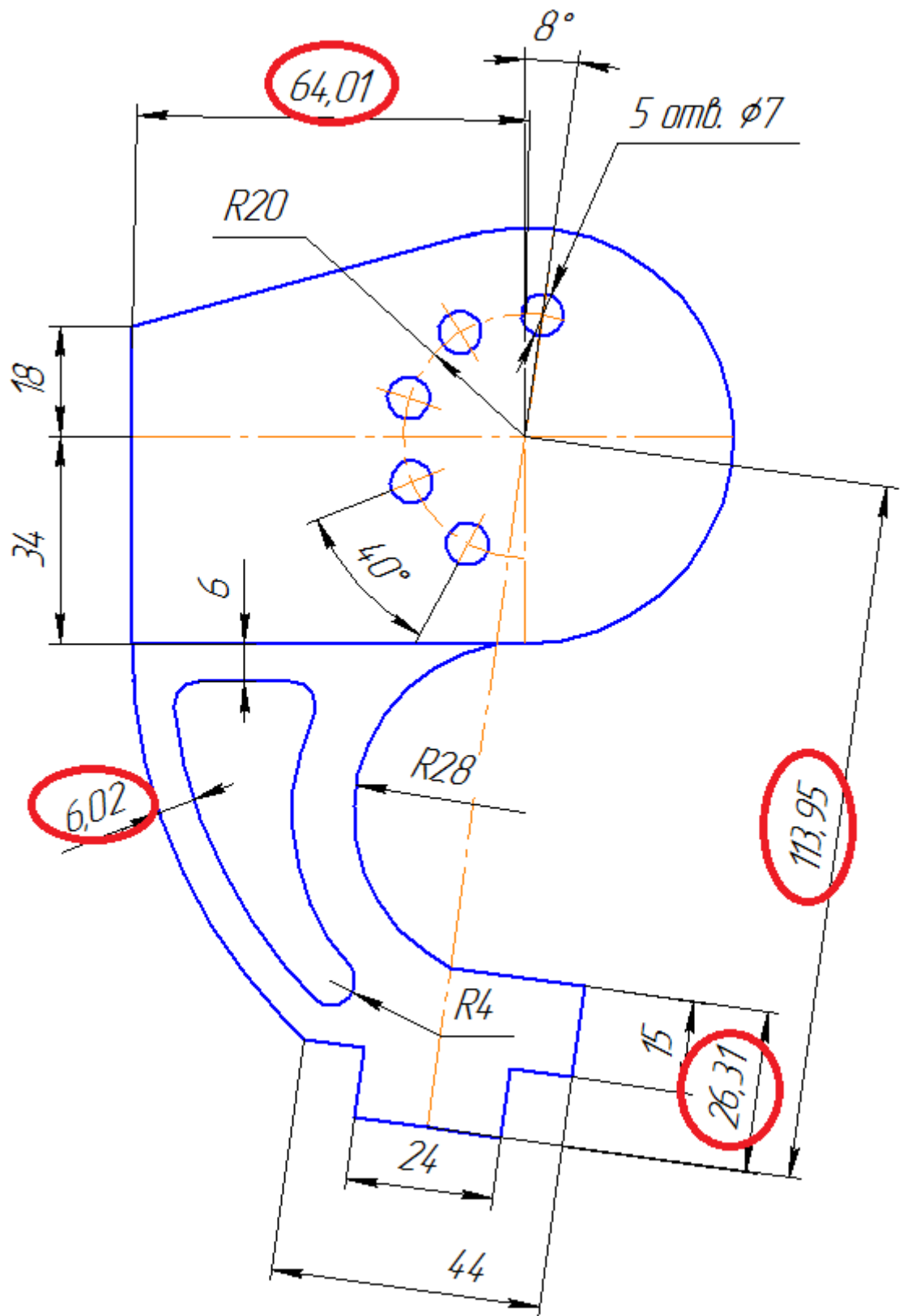
По окончании работы с САПР Компас-График вызывается функция `void KompasObject::Quit()`, закрывающая COM-соединение и саму САПР.

2.3 Поиск и контроль ошибок проектирования/распознавания/векторизации и алгоритмы их исправления

На данный момент не будем рассматривать ошибки проектирования, относящиеся к процессу производства объекта класса «деталь», такие как: несовпадение чертежей со спецификациями; различное положение одного объекта, находящегося на разных листах проектного решения (например, вентиляционной шахты) и т.д.; а остановимся лишь на ошибках, лежащих в одном поле чертежа. В основном такие ошибки возникают из-за невнимательности или же некомпетентности исполнителя (например, подобных ошибок полно в чертежах студентов, рисунок 2.7).



a)



б)

Рисунок 2.7 - Одна из студенческих работ. а) вариант студента с правильными, на его взгляд, размерными числами. б) тот же чертеж, только размерные числа зависят от длины отрезка в САПР.

Исходя из ГОСТ 2.307-2011 «Нанесение размеров и предельных

отклонений»[14], пункт 4.1:

«Основанием для определения величины изображенного изделия и его элементов служат размерные числа, указанные в графическом документе.

Исключение составляют случаи, предусмотренные ГОСТ 2.414(«ПРАВИЛА ВЫПОЛНЕНИЯ ЧЕРТЕЖЕЙ ЖГУТОВ, КАБЕЛЕЙ И ПРОВОДОВ»), ГОСТ 2.417(«ПЛАТЫ ПЕЧАТНЫЕ»), ГОСТ 2.419(«ПРАВИЛА ВЫПОЛНЕНИЯ ДОКУМЕНТАЦИИ ПРИ ПЛАЗОВОМ МЕТОДЕ ПРОИЗВОДСТВА»), когда величину изделия или его элементов определяют по изображениям, выполненным с достаточной степенью точности.

Основанием для определения требуемой точности изделия при изготовлении являются указанные предельные отклонения размеров, а также предельные отклонения формы и расположения поверхностей.»

при изготовлении детали нам следует полагаться на числа, стоящие над размерной линией.

Тем не менее, при автоматическом получении проекционных изображений с трехмерной модели, например, может возникнуть ситуация, когда величина объекта имеет размер, не реализуемый на практике (например, размерное число с точностью до тысячных долей миллиметра при строительстве панельного дома, рисунок 2.8).

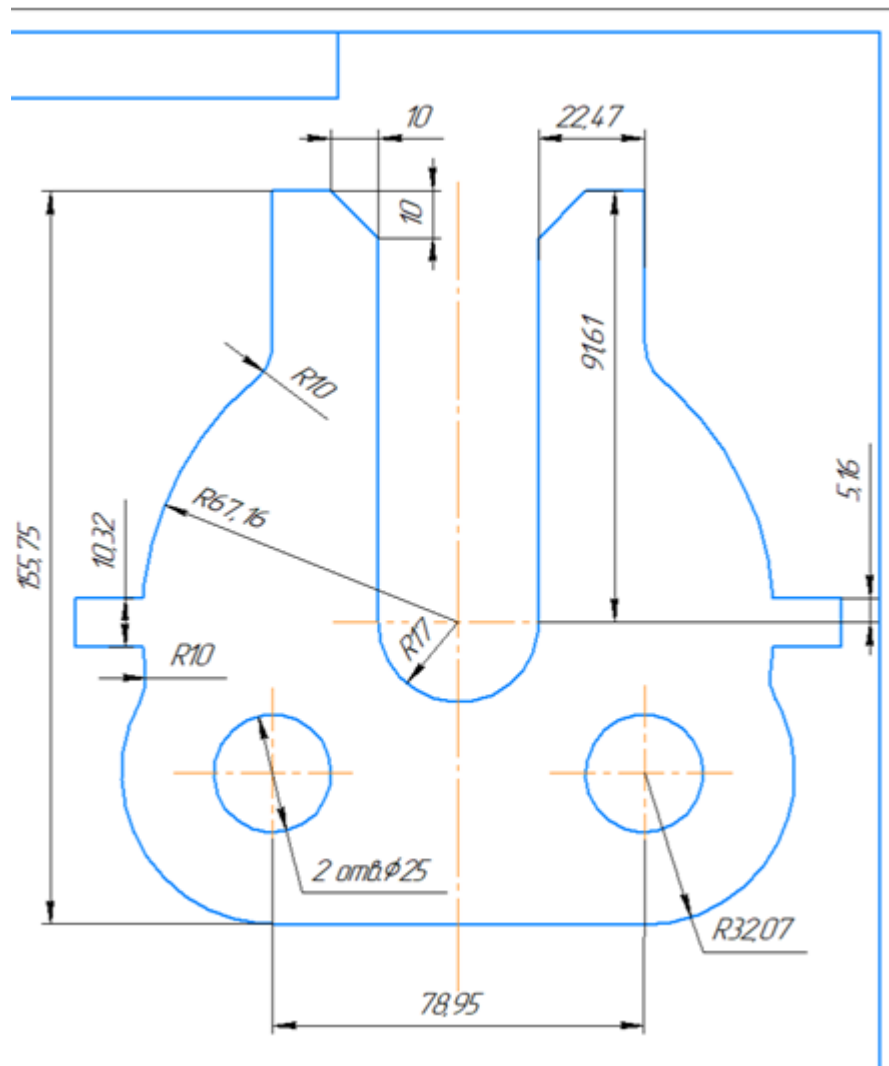


Рисунок 2.8 - Так же чертеж выполненный студентом. Невооруженным взглядом заметно несколько размеров, труднореализуемых на практике в связи с очень высокой требуемой точностью исполнения.

Опять же, исходя из ГОСТ 2.052-2021 «Единая система конструкторской документации. Электронная модель изделия. Общие положения» [12], пункт 4.7:

«Электронный конструкторский документ, выполненный в виде модели, должен соответствовать следующим основным требованиям:

а) атрибуты (модели), обозначения и указания, приведенные в модели, должны быть необходимыми и достаточными для указанной цели выпуска (например, изготовления изделия или построения чертежа в бумажной и/или электронной форме);

б) все значения размеров должны получаться из модели».

Таким образом, при автоматическом чтении чертежа станком с числовым программным управлением или же прикладной программой, лишь использующей

имеющуюся геометро-графическую информацию, размерные числа можно не учитывать, так как при соблюдении элементарной проектной культуры и, собственно, цифровом проектировании, размерное число по умолчанию должно быть равным величине объекта, для которого этот размер нанесен. Исходя из этого, можно оперировать лишь геометрическими примитивами, обращаясь к условным обозначениям только в спорных ситуациях.

Но, при возможности наличия на чертеже вышеописанных ошибок проектирования, возникает необходимость в предобработке двумерной геометро-графической информации перед её непосредственным использованием путем выявления и обнаружения подобных недоразумений. Вывод: алгоритм синтеза трехмерной детали критичен к входной информации, как и вся ЭМИ.

Прежде чем начать обработку полученных данных, необходимо проверить их на соответствие спецификациям задачи, выявить возможные ошибки или неточности. Согласно информационной модели, на вход алгоритма должны поступать координаты точек-вершин на плоскости, параметры несущих эллипсов и множество ребер графа, описываемых номерами инцидентных вершин и несущей линии.

В исходных данных могут встретиться следующие ошибки:

1. ошибки задания параметров несущих эллипсов: отрицательные или близкие к нулю значения для длин полуосей, неопределенность матрицы квадратичной формы (которая для эллипса должна быть строго положительно определена) и др, в зависимости от способа описания геометрии эллипса;
2. ошибки задания топологических связей на чертеже: не определены вершины, которые для ребра указаны как инцидентные, не определена несущая линия с указанным ребром;
3. несоответствия между геометрической и топологической информацией: указанные для ребра вершины не принадлежат эллипсу.

Элементы чертежа, содержащие ложные данные, должны быть отобраны и удалены.

Кроме ошибок, допущенных человеком, в исходных данных также могут

встречаться ошибки или неточности, связанные непосредственно с типом исходного документа - с бумажным чертежом. После сканирования бумажный чертеж передается на следующий этап - векторизацию.

Ключевые ошибки векторизации, возникающие на всех этапах обработки[5]: при сканировании появляются искажения геометрии из-за деформации бумаги и потери данных (исчезновение тонких линий при монохромном сканировании), при бинаризации метод Оцу ошибочно объединяет объекты с близкими цветами и генерирует артефакты, а фильтрация не всегда устраняет помехи («снег» и разрывы линий). Векторизация сопровождается топологическими ошибками (ложные пересечения, некорректное распознавание дуг) и метрическими искажениями (сдвиги элементов, ошибки масштаба). Отдельная проблема – ограничения алгоритмов для сложных случаев: они не обрабатывают сборки и многослойные чертежи, а исходное качество документа напрямую влияет на результат. Для минимизации ошибок авторы советуют дополнять автоматизированную обработку ручным контролем, особенно для повреждённых чертежей.

2.4 Выделение компонент связности на чертежах

Любую проекцию трехмерной модели на плоскость можно представить в виде графа. Если на чертеже одновременно размещены несколько проекционных видов, то их можно выделить в компоненты связности этого графа. Но применение стандартных подходов к поиску компонент связности (таких как поиск в ширину или глубину, то есть обход всех вершин и ребер графа с пометанием пройденных вершин) не дадут конечный результат в виде проекционных видов. В почти каждой трехмерной модели присутствуют отверстия, не связанные с внешней геометрией и они выделятся в отдельные компоненты связности, если использовать, например, поиск в ширину.

В связи с вышеизложенным, мы хотим предложить следующий подход к выделению нескольких проекционных видов на чертеже.

При любом способе получения геометро-графической информации, будь то отсканированный чертеж или начерченный в одной из систем автоматизированного проектирования, для всех видов плоских кривых можно выделить область, внутрь которой эта кривая помещается. Пусть такой областью у нас будет прямоугольник, назовем его ограничивающим. Для вертикальных и горизонтальных отрезков это прямоугольник выродится в сам отрезок.

Для каждой компоненты связности тоже запомним ограничивающий прямоугольник, расширяя его при добавлении новых объектов (при необходимости). Проверку же принадлежности объекта будем проводить по принципу: если ограничивающий прямоугольник объекта и ограничивающим прямоугольник компоненты связности имеют общую площадь или вершину или пересекаются их стороны, то объект входит в компоненты связности. Более подробная блок-схема данного процесса представлена на рисунке 2.9

Немного неправильно употреблять термин «компонента связности», так как, строго говоря, объекты внутри выделяемого проекционного вида, могут быть не связаны явно между собой.



Рисунок 2.9 - Блок-схема выделения проекционных видов.

При данном подходе для чертежа, содержащего несколько проекционных видов (эпюр Монжа, разрезы, сечения и т.д.) будут выделены в компоненты связности и обработаны все пришедшие в виде входной геометро-графической информации виды.

Описанный выше способ выделения проекционных видов на чертеже имеет ряд положительных моментов при работе с геометро-графической информацией.

Первое – удобный способ проверки и восстановления проекционных связей. Сравнив в определенной окрестности ограничивающие прямоугольники выделенных видов, можно выяснить, какие из них имеют (или подразумевают, в случае неточностей входной информации) проекционные связи, а какие являются

дополнительными (вынесенные разрезы, сечения и так далее).

Второе – исходя из ГОСТ 2.305-2008 «Единая система конструкторской документации (ЕСКД). Изображения - виды, разрезы, сечения (с Поправкой)» [13] проекционные виды на чертеже располагаются в определенном порядке. Зная ограничивающие прямоугольники для каждого вида, выяснив проекционные связи между ними, можно утверждать, какое из проекционных изображений на техническом чертеже соответствует входной информации.

Третье – при работе с геометро-графической информацией, поступающей напрямую из системы автоматизированного проектирования [35] можно получить точки привязки секущих плоскостей для вынесенных разрезов и сечений, считав условные обозначения на чертеже.

2.5 Выводы по второй главе

1. Рассмотрены различные варианты хранения ГГИ в памяти компьютера и способы передачи её между прикладными программами, на основании чего был выбран способ получения исходных данных напрямую из памяти САПР, описан алгоритм получения ГГИ через СОМ-объекты на примере САПР Компас.

2. Рассмотрены возможные ошибки, которые могут встретиться в исходных данных. При этом не затрагиваются ошибки, допущенные проектировщиками. Принято решение на основании информации из ГОСТ 2.052-2021. “Единая система конструкторской документации (ЕСКД). Электронная модель изделия. Общие положения.”[12] о чтении размеров из длин элементов внутри САПР, а не из размерных чисел.

ГЛАВА 3. АЛГОРИТМЫ ФОРМИРОВАНИЯ ТРЕХМЕРНОЙ КАРКАСНОЙ МОДЕЛИ

3.1 Выделение проекционных видов и восстановление проекционных связей

В соответствии с ГОСТ 2.305-2008 “Единая система конструкторской документации (ЕСКД). Изображения - виды, разрезы, сечения (с Поправкой)”[13] чертеж объекта может состоять из произвольного числа видов, расположенных как на одном, так и на нескольких листах. И в том, и другом случае эти виды должны быть приведены к единой системе координат, на основе которой формируется каркасная модель объекта. Одним из блоков программной реализации алгоритма синтеза является установление и проверка проекционных связей видов между собой [45]. Именно после установления проекционных связей формируется массив 3D координат модели.

Любую проекцию трехмерной модели на плоскость можно представить в виде графа $G(V,R)$, где V - множество вершин, заданных своими координатами, R - множество ребер – линий первого и второго порядка, соединяющих вершины. Структурно сам чертеж также может быть представлен в виде графа $D(F)$, где F – виды или фрагменты изображений объекта, каждый из которых задан в своей системе координат, связанной с системой координат чертежа.

Если на чертеже одновременно размещены несколько проекционных видов, то их можно выделить в компоненты связности этого графа. Применение стандартных подходов к поиску компонент связности (таких как поиск в ширину или глубину, то есть обход всех вершин и ребер графа с фиксацией пройденных вершин) не дадут конечный результат в виде проекционно связанных видов. В каждой трехмерной модели могут присутствовать отверстия, не связанные с внешней геометрией, и они выделятся в отдельные компоненты связности, если использовать, например, поиск в ширину.

При любом способе получения геометро-графической информации, будь то

отсканированный чертеж или начерченный в одной из систем автоматизированного проектирования, для всех видов плоских кривых можно выделить область, внутрь которой эта кривая помещается. Пусть такой областью у нас будет прямоугольник, назовем его ограничивающим. Для вертикальных и горизонтальных отрезков этот прямоугольник выродится в сам отрезок.

Для каждой компоненты связности тоже запомним ограничивающий прямоугольник, расширяя его при добавлении новых объектов (при необходимости). Проверку же принадлежности объекта будем проводить по принципу: если ограничивающий прямоугольник объекта и ограничивающий прямоугольник компоненты связности имеют общую площадь или вершину или пересекаются их стороны, то объект входит в компоненты связности.

При добавлении объекта в компоненту связности её ограничивающий прямоугольник должен быть переопределен. Примеры различных случаев расположения ограничивающих прямоугольников компоненты и объекта приведены на рисунке 3.1. В случаях а) и б) объект принадлежит компоненте связности и для них определяется новый ограничивающий прямоугольник, показанный красным цветом. В случае в) объект не относится к проверяемой компоненте связности.

Отметим, что немного неправильно употреблять термин «компонента связности», так как, строго говоря, объекты внутри выделяемого проекционного вида, могут быть не связаны явно между собой.

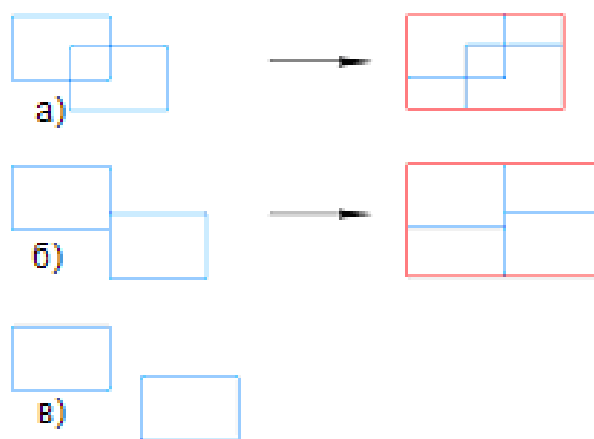


Рисунок 3.1. - Различные случаи расположения ограничивающих прямоугольников при определении принадлежности объекта компоненте связности

При данном подходе для чертежа, содержащего несколько проекционных видов (виды, разрезы, сечения и т.д.) будут выделены в компоненты связности и обработаны все пришедшие в виде входной геометро-графической информации изображения.

Используя ограничивающие прямоугольники выделенных проекционных видов и руководствуясь ГОСТ 2.305-2008[13], описывающим их взаимное расположение на одном листе, можно достаточно легко восстановить проекционные связи при условии их нарушения. Подробно данный алгоритм описан в [33].

В случае же когда геометро-графическая информация изначально располагается на нескольких листах, а ограничивающие прямоугольники видов имеют форму квадрата, данный способ не удастся применить.

Для решения данной задачи предлагается следующий подход. При выделении проекционных видов формируются 2 вектора координат точек – с проекциями элементов вида на оси Ox и Oy чертежа. При добавлении очередного элемента к виду проецируем на две стороны ограничивающего прямоугольника две или более его характерных точек (для отрезка это будут его концы). Назовем такие векторы проекционными (рисунки 3.2 и 3.3).

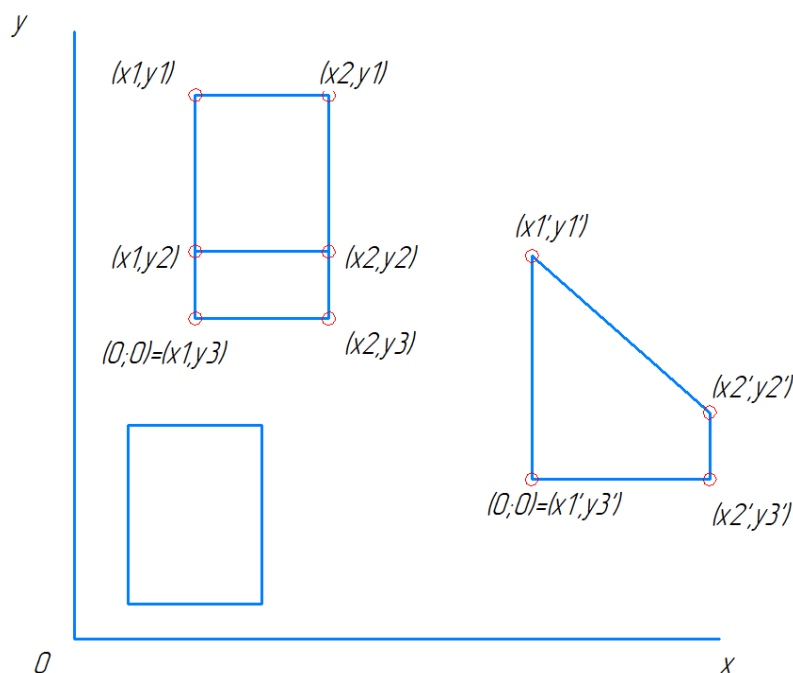


Рисунок 3.2 - Выделяем все точки на виде спереди и на виде слева.

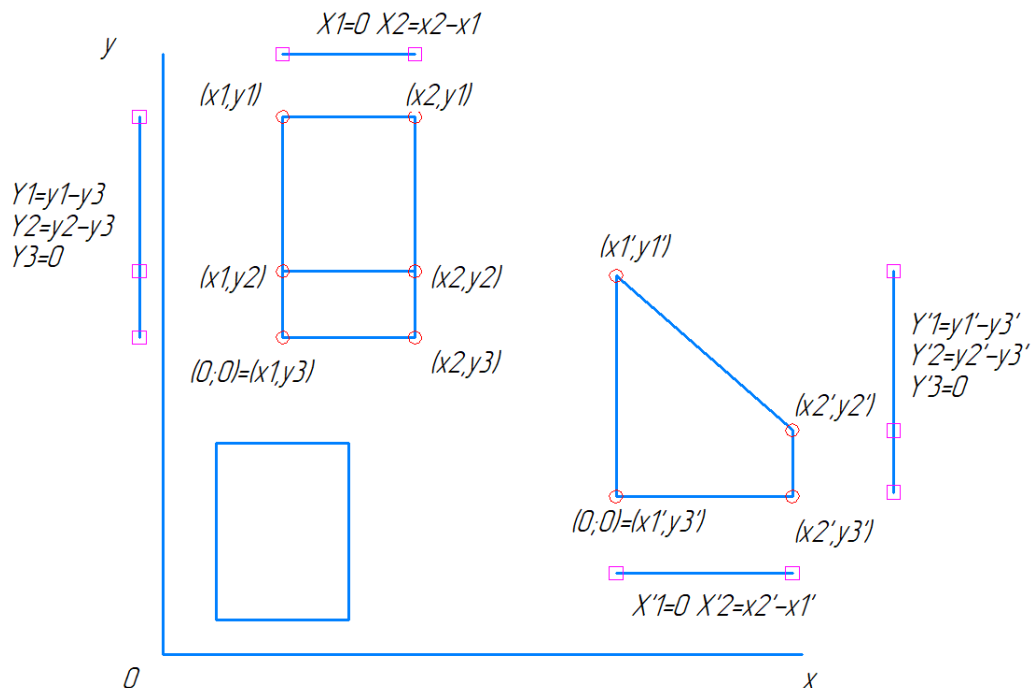


Рисунок 3.3 - Проекционные векторы для видов спереди и слева.

При невозможности восстановления проекционных связей между видами только при помощи ограничивающего прямоугольника полученные для каждого выделенного вида оба проекционных вектора сортируются и приводятся к общей точке отсчета в левом нижнем углу вида. После выполнения сортировки начинаем сравнение данных векторов у различных проекционных видов. Если проекционные векторы совпадают, то есть разница всех их компонент меньше заранее определенного ε , то вычисляем вектор сдвига (рисунки 3.4, 3.5 и 3.6). Для восстановления проекционных связей принимаем за главный вид левое верхнее проекционное изображение. Далее ищем виды, у которых совпадают проекционные векторы с его векторами и, в соответствии с ГОСТ 2.305-2008[13], располагаем их на одном листе. Далее находим координаты точки начала отсчета эппюра Монжа: x координата как середина между видом слева и главным видом, y координата как середина между видом спереди и видом сверху (рисунок 3.7).

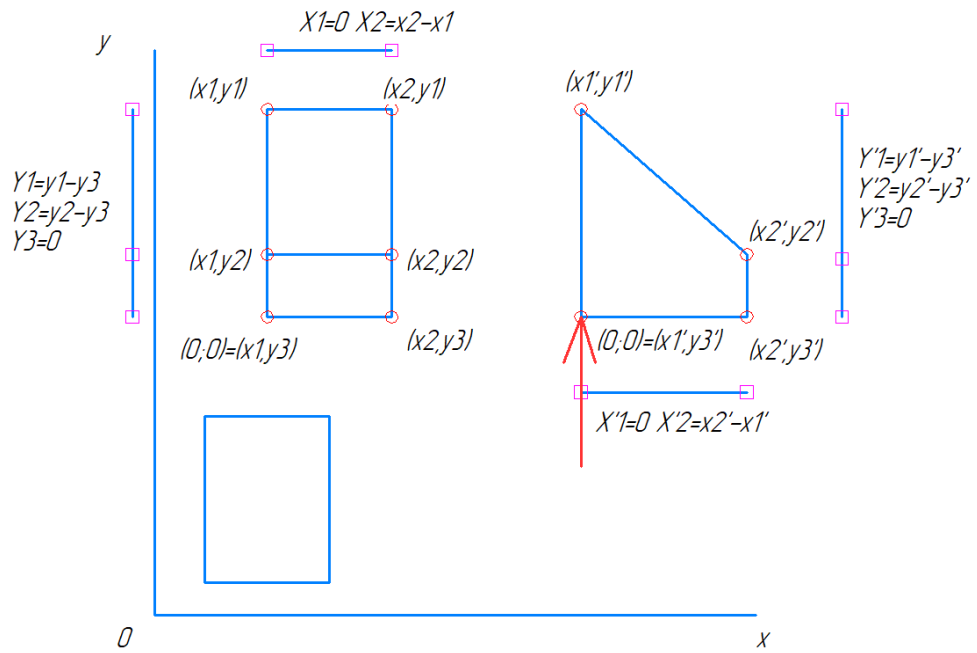


Рисунок 3.4 - Вектор сдвига вида слева для формирования линий проекционной связи с видом спереди.

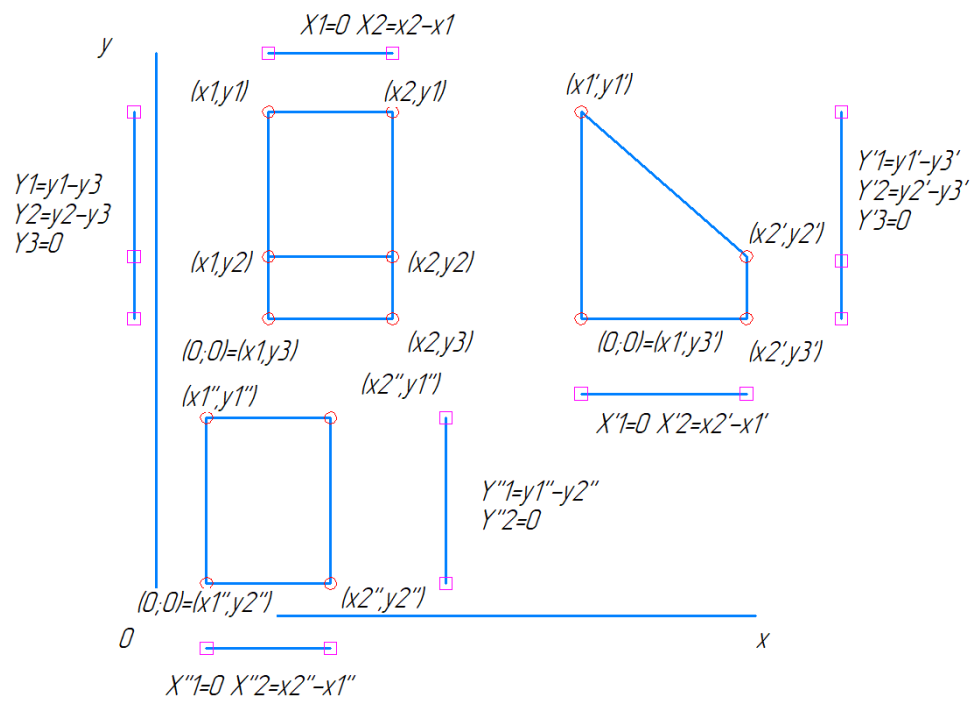


Рисунок 3.5 - Проекционные векторы для вида сверху.

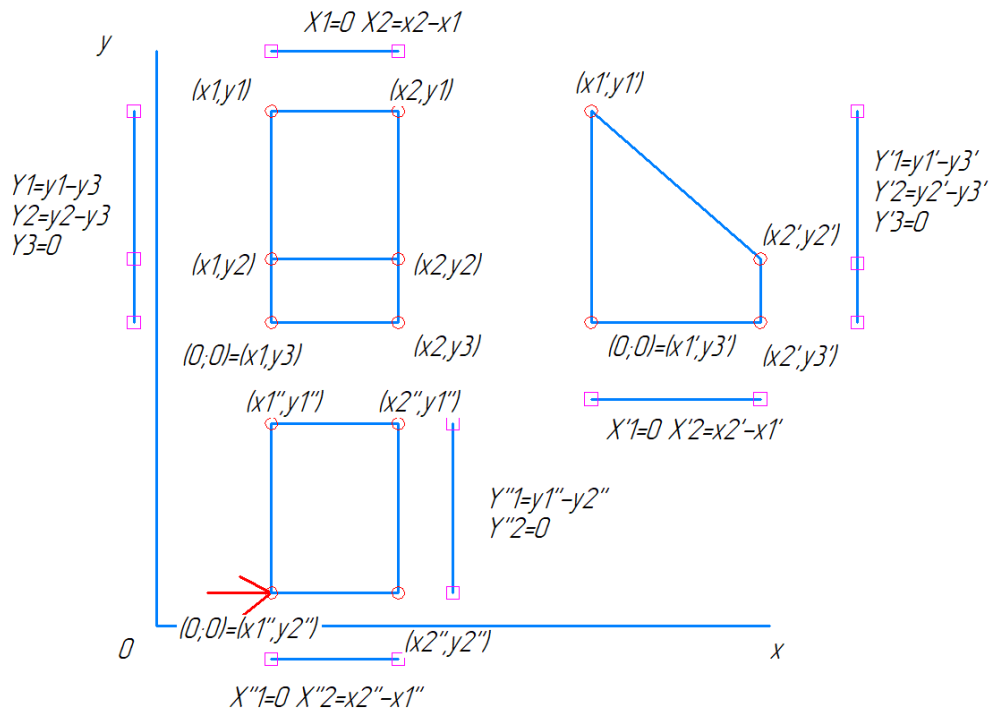


Рисунок 3.6 - Вектор сдвига вида сверху для формирования линий проекционной связи с видом спереди.

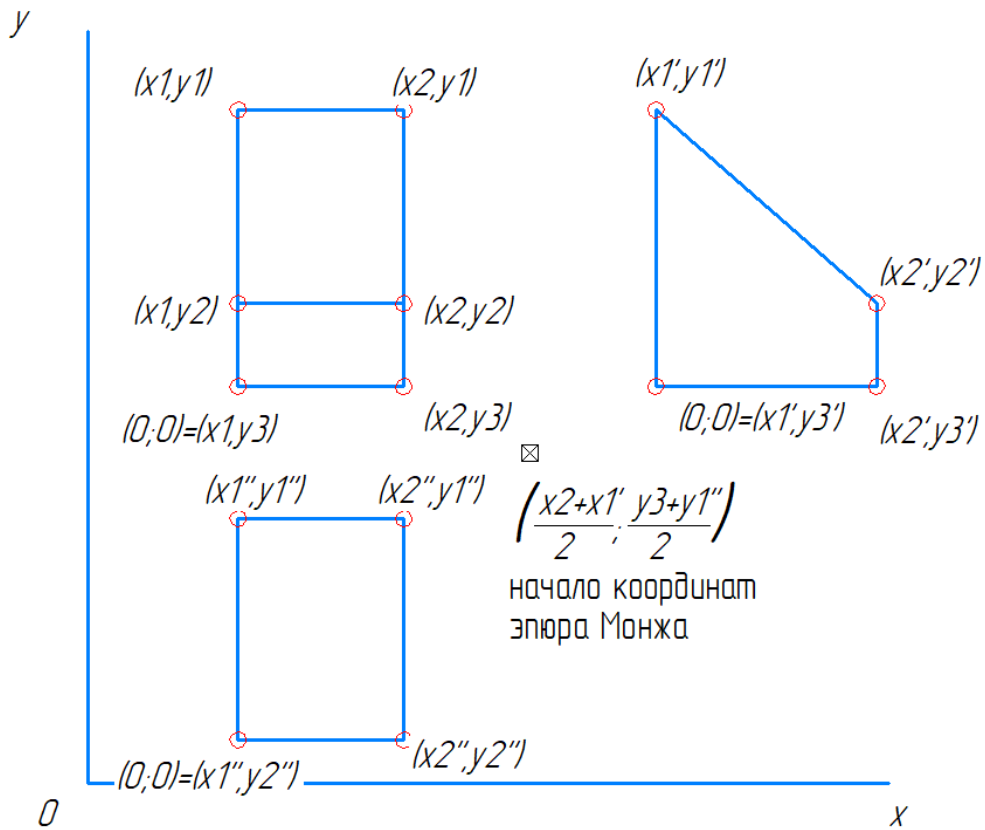


Рисунок 3.7 - Найденная точка начала координат эпюра Монжа.

Для получения координат точек в новой системе координат эпюра Монжа необходимо выполнить следующие преобразования:

- для точек вида спереди

$$x_3 = x_{\text{Monge}} - x_{\text{LCS}} - x;$$

$$z_3 = y_{\text{LCS}} - y_{\text{Monge}} + y;$$

- для точек вида слева

$$y_3 = x_{\text{LCS}} - x_{\text{Monge}} + x;$$

$$z_3 = y_{\text{LCS}} - y_{\text{Monge}} + y;$$

- для точек вида сверху

$$x_3 = x_{\text{Monge}} - x_{\text{LCS}} - x;$$

$$y_3 = y_{\text{Monge}} - y_{\text{LCS}} - y;$$

где x_3, y_3, z_3 - предполагаемые координаты точки в трехмерном пространстве,

x, y - координаты точки в проекционном виде,

$x_{\text{LCS}}, y_{\text{LCS}}$ - координаты точки начала отсчета проекционного вида на чертеже,

$x_{\text{Monge}}, y_{\text{Monge}}$ - точка начала координат эпюра Монжа, полученная на предыдущем шаге.

3.2 Анализ проекционных изображений

Анализ проекционных изображений играет ключевую роль в процессе восстановления трехмерной модели, особенно когда работа ведется с изображениями, полученными после алгоритма векторизации сканированного чертежа[22]. Векторизация преобразует растровые данные чертежа в векторные, выделяя линии, контуры и другие геометрические примитивы. На этом этапе уже выделены компоненты связности, представляющие собой отдельные проекционные изображения, такие как вид спереди, сбоку и сверху. Основная задача анализа заключается в определении соответствий между точками и линиями на разных проекционных изображениях и последующем восстановлении трехмерных координат.

Сопоставление элементов между проекционными изображениями является первым шагом в процессе восстановления трехмерной модели. Для этого используются методы поиска соответствий, такие как сопоставление контуров или анализ пересечений линий [2]. Важным аспектом является учет эpsilon-окрестности, которая позволяет находить соответствия даже при наличии небольших отклонений в положении точек или линий. Например, если точка на одном изображении и точка на другом изображении находятся в пределах заданной окрестности, они считаются соответствующими друг другу. Этот подход особенно важен при работе с чертежами, созданными в ручной технике, где возможны погрешности, такие как неточности в проведении линий, смещения или искажения. Учет эpsilon-окрестности повышает устойчивость алгоритма к таким ошибкам, позволяя корректно восстанавливать трехмерную структуру даже при наличии неточностей в исходных данных [100]. Блок-схема данного процесса представлена на рисунке 3.8.

Таким образом, анализ проекционных изображений, полученных после векторизации, и учет погрешностей, связанных с ручной техникой создания чертежей, являются важными этапами в процессе восстановления трехмерной модели. Использование методов сопоставления и эpsilon-окрестности позволяет

эффективно решать задачу даже при наличии неточностей в данных.

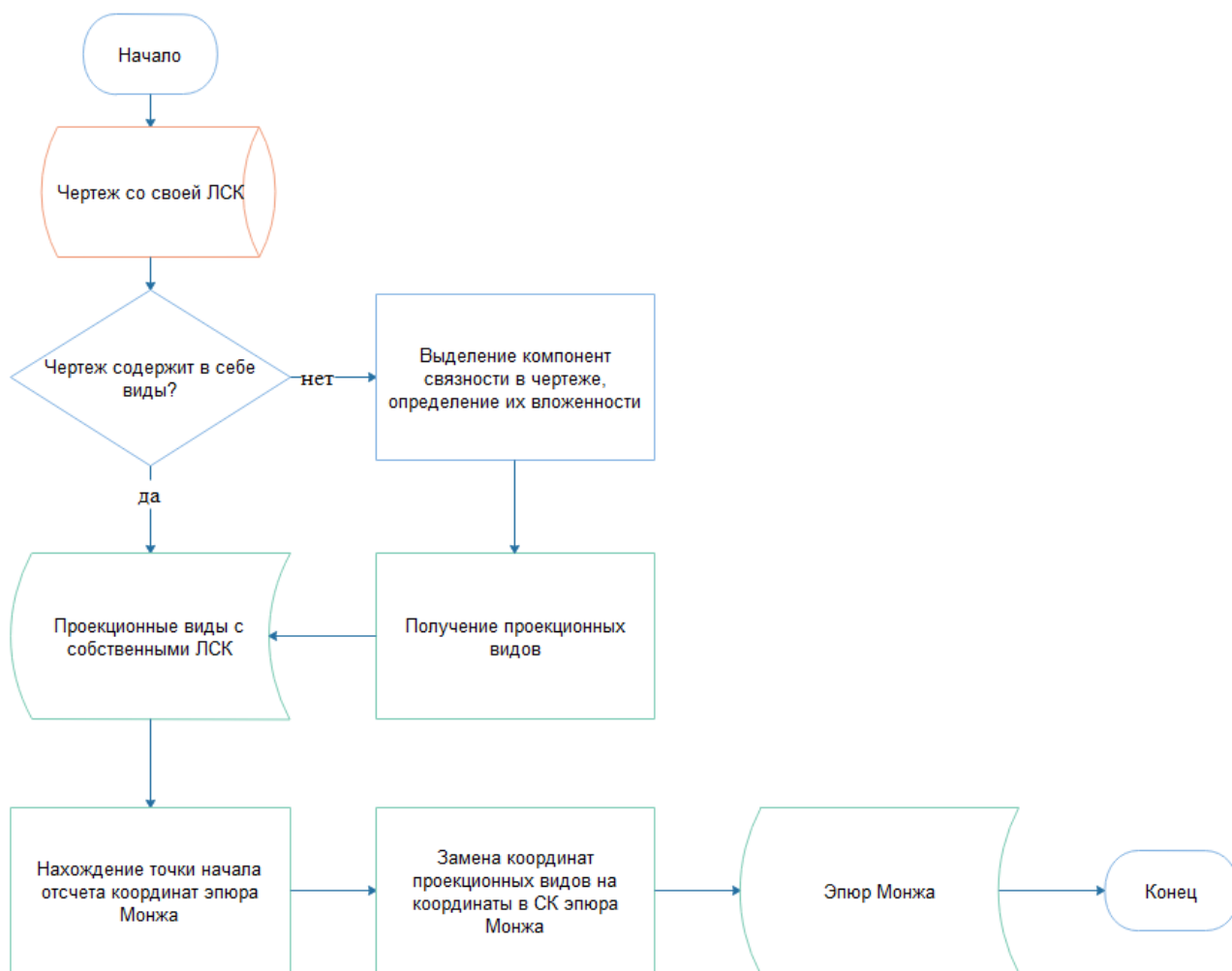


Рисунок 3.8 - Преобразование координат выделенных проекционных видов в единую систему координат эпюра Монжа

Для иллюстрации процесса восстановления трехмерной каркасной модели на основе проекционных изображений была выбрана модель, полученная в ходе взаимодействия с ЦНИИ “Буревестник”. На рисунке 3.9 содержится пример чертежа, необходимого для работы алгоритма. На рисунке 3.10 показаны все элементы чертежа, которые были прочитаны и восприняты реализованным программным комплексом.

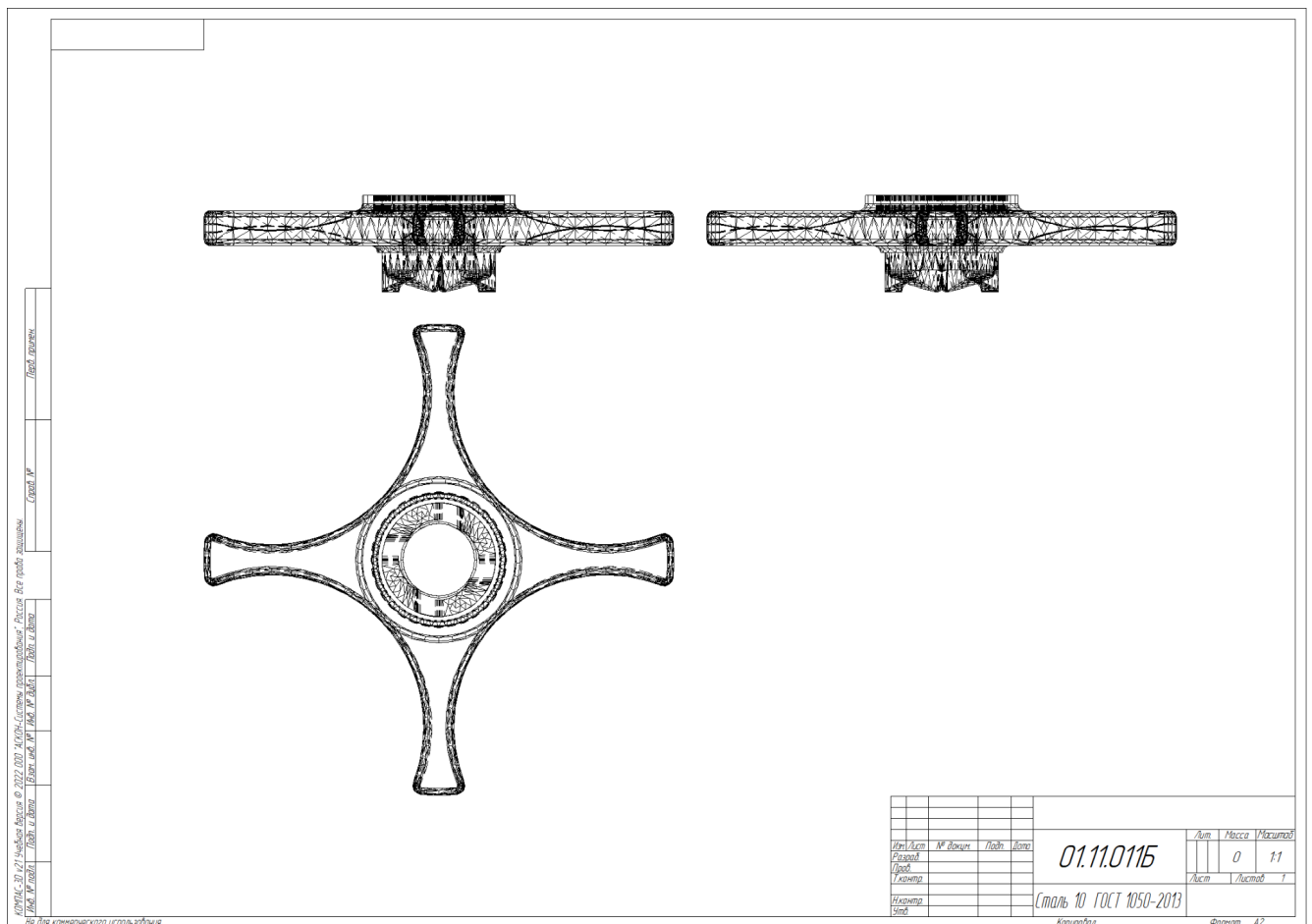


Рисунок 3.9 - Пример чертежа для восстановления каркасной модели

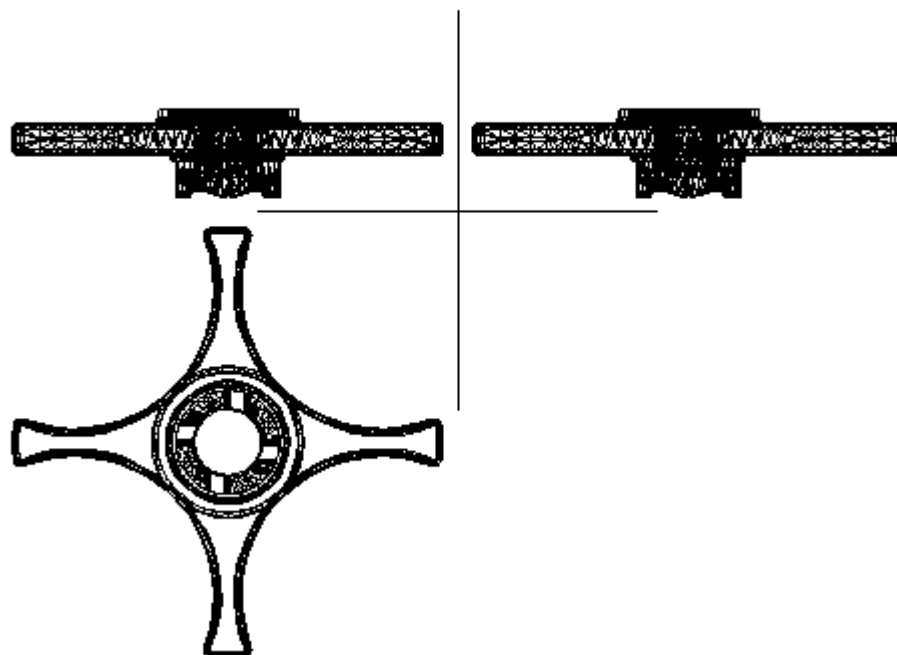


Рисунок 3.10 - Пример прочитанного реализованным программным комплексом чертежа

3.3 Получение точечной модели

Триангуляция является ключевым методом для восстановления трехмерных координат точек на основе их проекций на нескольких изображениях. Этот процесс основан на принципе пересечения лучей, проходящих через соответствующие точки на разных изображениях. Рассмотрим основные этапы данного процесса[42]:

1. Определение соответствующих точек: для каждой точки на одном изображении находится соответствующая точка на другом изображении. Для этого от точки берутся те две нужные координаты, представляющие собой её проекцию на интересующую плоскость.

2. Вычисление лучей: для каждой пары соответствующих точек вычисляются лучи, проходящие через эти точки и центр проекции камеры. Эти лучи представляют собой линии в трехмерном пространстве, которые должны пересекаться в точке, соответствующей реальной трехмерной координате.

3. Пересечение лучей: в идеальном случае лучи, соответствующие одной и той же точке в пространстве, должны пересекаться в одной точке. Однако из-за ошибок в данных и неточностей в определении соответствующих точек лучи могут не пересекаться точно. В этом случае используется метод наименьших квадратов для нахождения точки, которая минимизирует расстояние до всех лучей.

4. Восстановление трехмерных координат: на основе пересечения лучей вычисляются трехмерные координаты точек (рисунок 3.11). Эти координаты затем используются для построения трехмерной модели объекта.

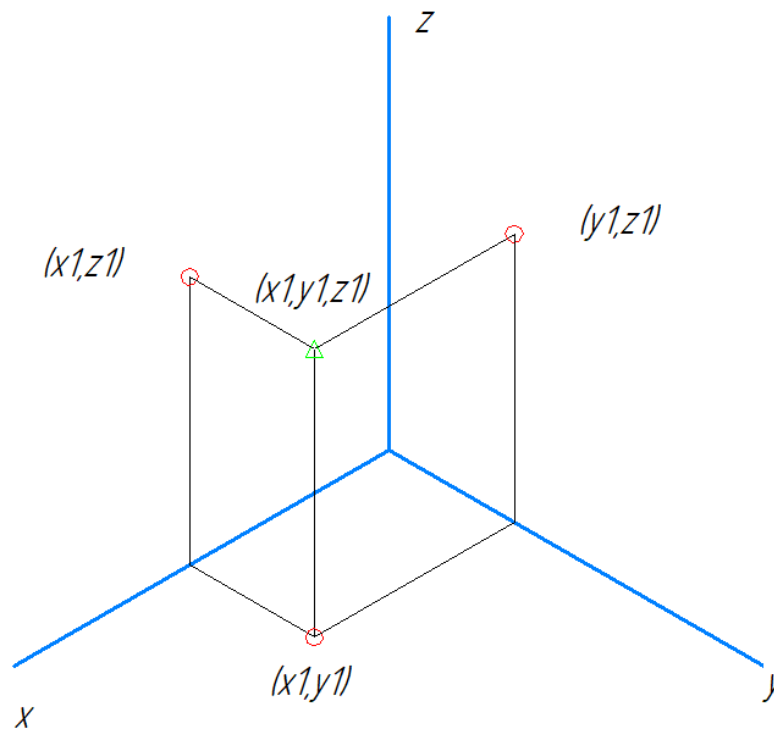


Рисунок 3.11 - Формирование точки из облака на основании трех её проекций
Формирование облака точек

После восстановления трехмерных координат точек формируется облако точек, которое представляет собой набор точек в трехмерном пространстве. Это облако точек уже может быть использовано для построения каркасной модели объекта.

На рисунке 3.12 показана точечная модель, сформированная на основе описанного алгоритма по чертежу с рисунка 3.10

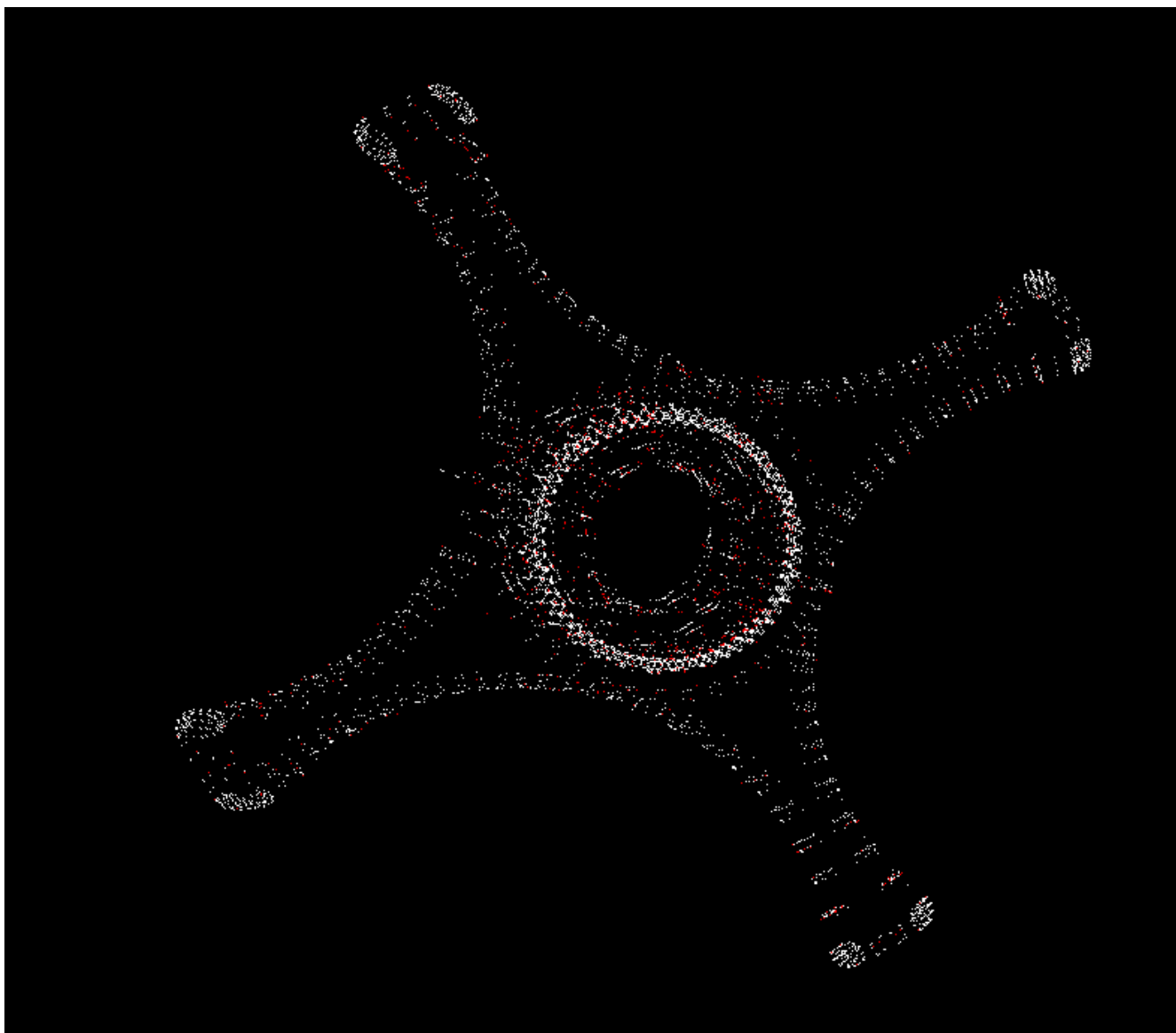


Рисунок 3.12 - Восстановленная на основании чертежа с рисунка 3.10 точечная модель, содержащая 7580 точек

3.4 Формирование каркасной модели

Для каждой точки облака точек необходимо проверить её связь со всеми другими точками. Этот процесс включает следующие шаги[42]:

1. Проекция точки на изображения: для каждой точки облака точек вычисляются её проекции на все проекционные изображения (рисунок 3.11).
2. Анализ матриц смежности: для каждого проекционного изображения имеется матрица смежности, которая описывает связи между проекциями точек. Если две точки соединены ребром на изображении, соответствующий элемент матрицы смежности равен 1, иначе — 0.
3. Определение ребер:
 - Для каждой пары точек ($V1$ и $V2$) проверяется наличие ребра между их проекциями на всех проекционных изображениях ($VF1$ и $VF2$ на фронтальной, $VH1$ и $VH2$ на горизонтальной, $VP1$ и $VP2$ на профильной) (рисунок 3.13).

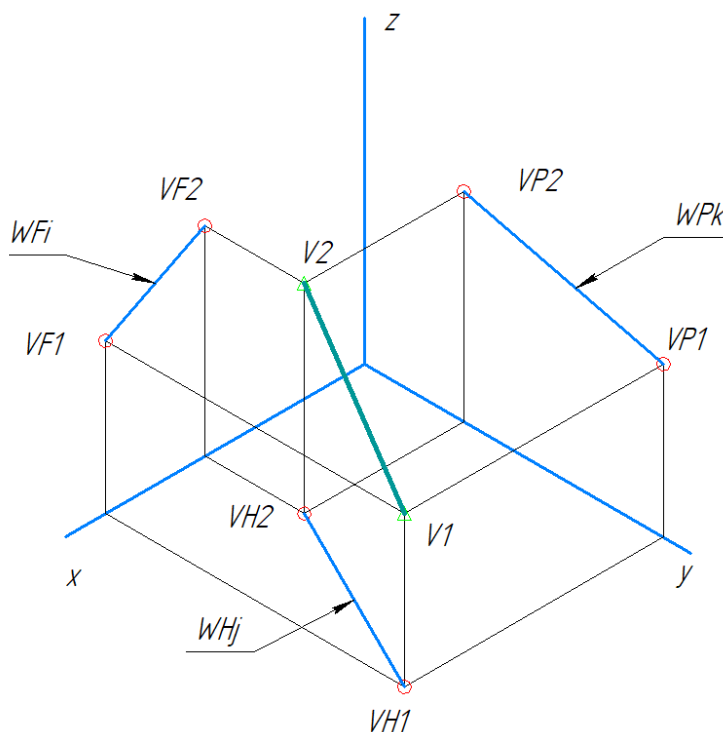


Рисунок 3.13 - Формирование ребра между вершинами $V1$ и $V2$.

- Ребро в каркасной модели между двумя точками считается существующим, если на всех изображениях существует ребро между

проекциями этих точек или проекции точек совпадают (то есть точки находятся на одном проецирующем луче). Если на какой-либо из проекций точки совпали, то может быть восстановлено лишь ребро-отрезок.

- Если проекции не совпали, то выделяем все ребра, по которым смежны данные вершины: WF_i , $i=1\dots N_f$ для $VF1$ и $VF2$, WH_j , $j=1\dots N_h$ для $VH1$ и $VH2$, WP_k , $k=1\dots N_p$ для $VP1$ и $VP2$.

- Рассматриваем всевозможные тройки ребер WF_i , WH_j , WP_k . Если эти три ребра являются образами некоторого ребра каркасной модели, то несущие линии этих ребер - это проекции пространственной прямой или эллипса (по соглашению о типах несущих линий для ребер каркаса). Возможны четыре принципиально различных конфигурации типов несущих линий для выбранных ребер (ниже приведен анализ их соответствия линии в R^3). Если анализ показывает, что WF_i , WH_j , WP_k могут являться проекциями некоторого ребра каркаса, то это ребро добавляется к модели и перебор троек продолжается.

На основе проверенных связей формируется каркасная модель, которая состоит из вершин (точек облака) и ребер (связей между точками).

Анализ ребер на соответствие линии в R^3 .

1. Все три ребра – отрезки прямых. В этом случае отрезок пространственной прямой между вершинами $V1$ и $V2$ может, согласно необходимому условию, являться ребром каркаса.

2. Одно из трёх ребер – дуга эллипса, а два других – отрезки прямых. Для каждого из отрезков определяем плоскость в R^3 , проходящую через него перпендикулярно его картинной плоскости (по терминологии НГ проецирующую плоскость). Ребро каркаса, проецирующееся на эти отрезки, принадлежит в R^3 обеим построенным проецирующим плоскостям. Если эти плоскости пересекаются, то ребро каркаса лежит на линии их пересечения и не может проецироваться в дугу на третьей картинной плоскости. Если плоскости совпадают (то есть параллельны картинной плоскости третьего ребра-дуги), то в их пересечении с эллиптическим цилиндром, ось которого им перпендикулярна, а

направляющей является несущий эллипс третьего ребра, лежит возможное ребро каркаса. Случай параллельности этих плоскостей возникнуть не может, так как уже известны две их общие точки — вершины $V1$ и $V2$.

3. Среди трех ребер два представляют собой дуги, а одно - отрезок. Тогда возможно формирование ребра в каркасной модели, если:

а. несущие эллипсы дуг имеют одинаковую координату центра и одинаковый габаритный размер вдоль оси, по которой проекции смежны (например, X_F и X_H , Y_F и Y_P , Y_H и X_P). В этом случае два эллиптических цилиндра (с осями, перпендикулярными соответствующим картинным плоскостям) пересекаются в пространстве по двум эллипсам.

б. плоскость одного эллипсов, о которых говорилось выше, совпадает с проецирующей плоскостью для ребра-отрезка на третьей проекции.

4. Все три ребра, соединяющие образы вершин $V1$ и $V2$, являются дугами. Формирование 3D-дуги возможно, если:

а. для любых двух из трёх несущих эллипсов выполняется условие совпадения координат центров и габаритных размеров вдоль соответствующих осей (как в пункте 3а).

б. все три эллиптических цилиндра пересекаются в пространстве по одному общему эллипсу, который и становится линией ребра каркаса.

Предложенный алгоритм позволяет корректно формировать каркасную модель на основе облака точек и проекционных изображений. Основные преимущества алгоритма включают:

- Точность: учет связей между точками на всех проекционных изображениях обеспечивает высокую точность формирования ребер.

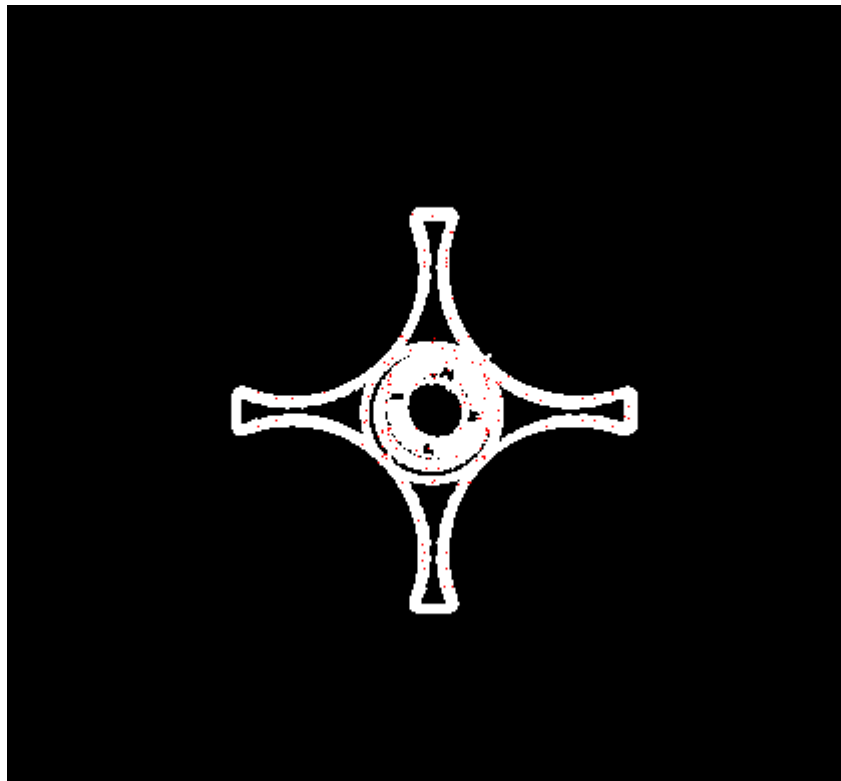
- Устойчивость к ошибкам: использование матриц смежности позволяет минимизировать влияние ошибок в данных.

- Гибкость: алгоритм может быть адаптирован для различных типов проекционных изображений и объектов.

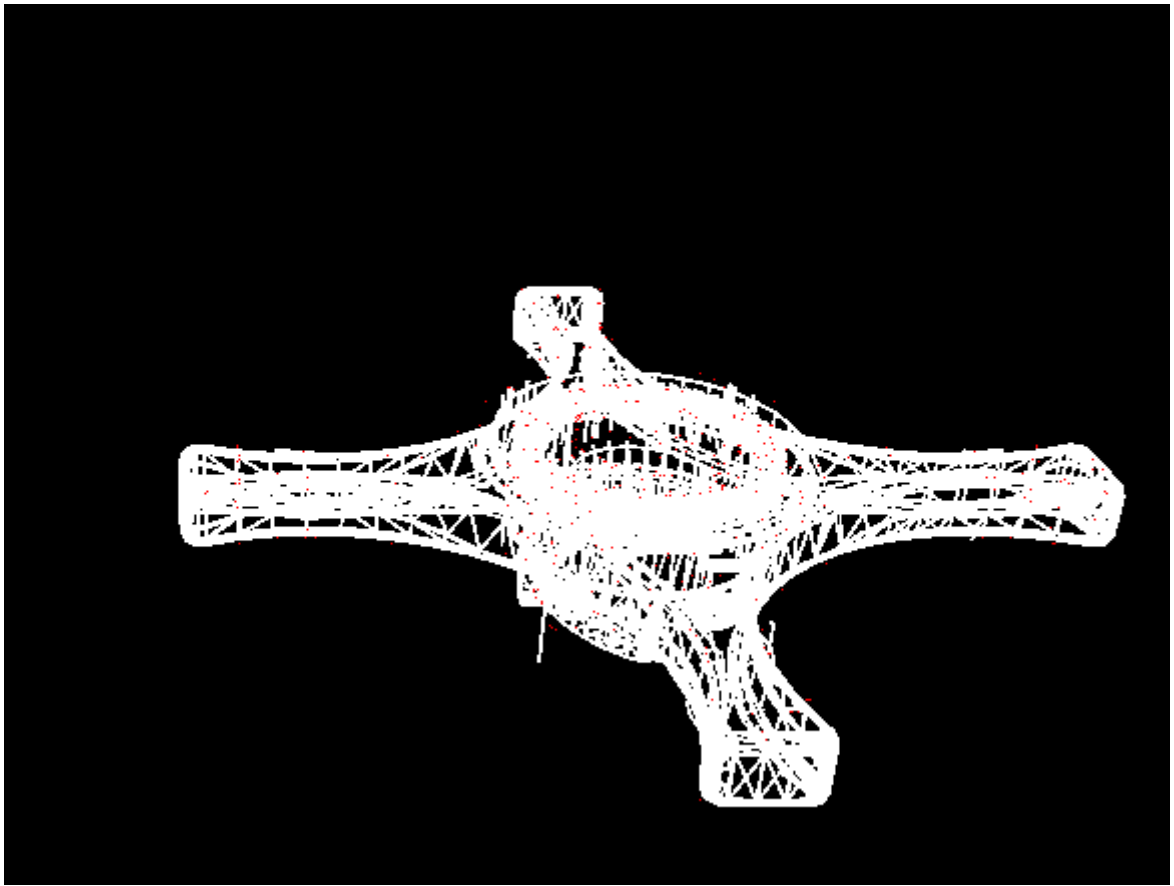
Однако вычислительная сложность алгоритма возрастает с увеличением

количества точек и проекционных изображений[18], что может потребовать применения методов оптимизации, таких как параллельные вычисления или использование GPU. Данный метод может быть адаптирован для различных типов проекционных изображений и объектов, что делает его применимым в задачах компьютерного зрения, робототехники и промышленного дизайна.

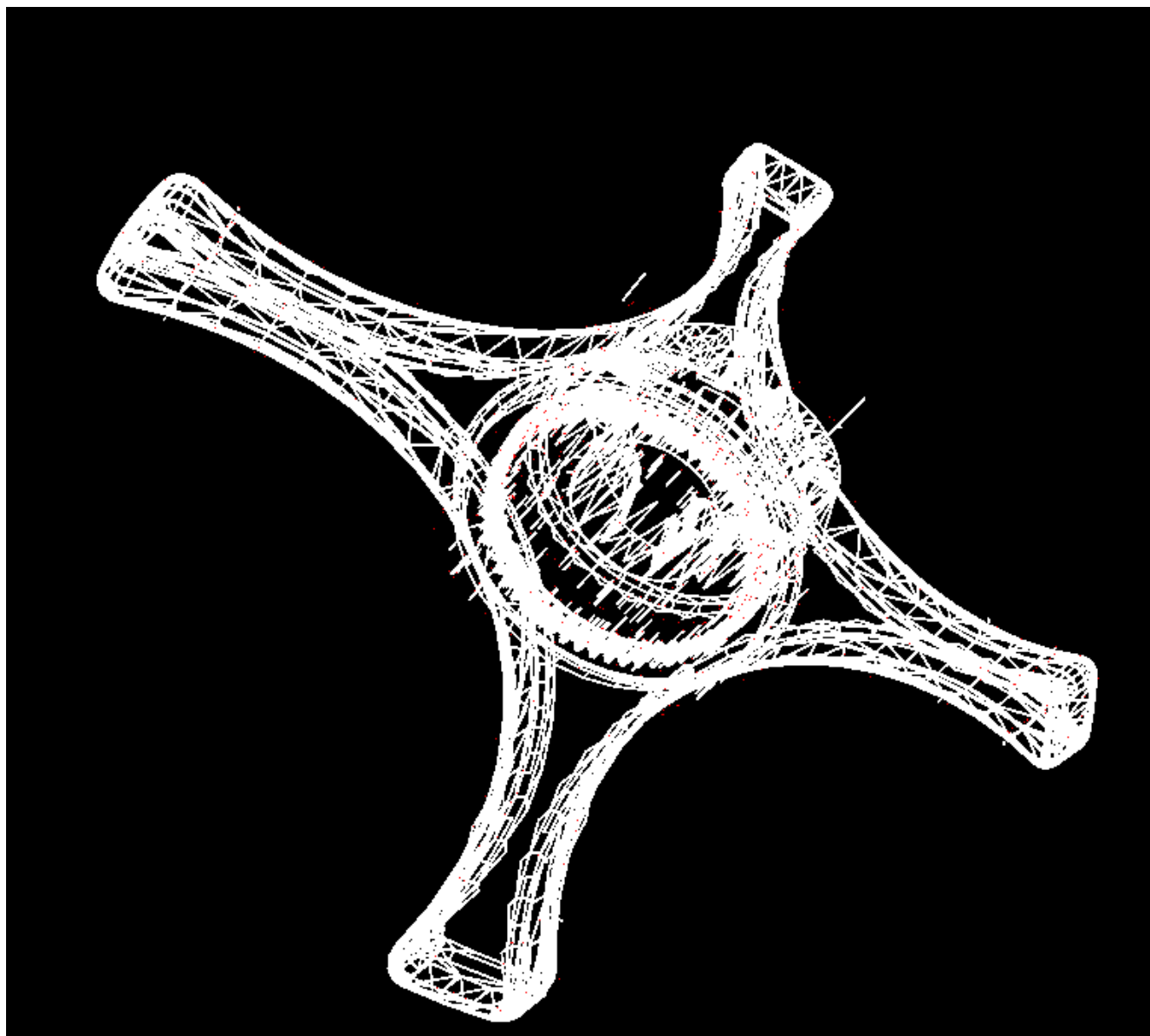
На рисунке 3.14 показана трехмерная каркасная модель, сформированная по описанному алгоритму на основании геометро-графической информации, считанной с чертежа, показанного на рисунка 3.9



a)



6)



в)

Рисунок 3.14 - Восстановленная на основании чертежа с рисунка 3.9 трехмерная каркасная модель, содержащая 21670 ребер, части а,б,в демонстрируют модель под разными углами наклона.

3.5 Выводы по третьей главе

1. Разработаны алгоритмы предварительной обработки двумерных геометрических данных, ориентированных на устранение типовых погрешностей, возникающих при вводе чертежной информации. Отличительными особенностями являются:

а. прямая интеграция с системами автоматизированного проектирования (САПР) на этапе получения входных данных, минуя использование промежуточных форматов, что обеспечивает более высокое качество и полноту исходной геометрической информации для последующей обработки;

б. анализ проекционных изображений, результатом которого является восстановление топологических и метрических связей, убирающих противоречия между входным предметно-манипулятивным чертежом (использующимся на производстве) и выходным эскизом Монжа.

2. Предложенный алгоритм формирования каркасной модели на основе проекционных изображений позволяет корректно формировать каркасную модель на основе облака точек и проекционных изображений. Основные преимущества алгоритма включают:

- Точность: учет связей между точками на всех проекционных изображениях обеспечивает высокую точность формирования ребер.

- Устойчивость к ошибкам: использование матриц смежности позволяет минимизировать влияние ошибок в данных.

- Гибкость: алгоритм может быть адаптирован для различных типов проекционных изображений и объектов.

ГЛАВА 4. ИССЛЕДОВАНИЕ ВЫЧИСЛИТЕЛЬНЫХ ХАРАКТЕРИСТИК АЛГОРИТМА ВОССТАНОВЛЕНИЯ КАРКАСНЫХ МОДЕЛЕЙ ПО ЧЕРТЕЖНО-КОНСТРУКТОРСКОЙ ДОКУМЕНТАЦИИ

4.1 Исследование временных характеристик алгоритма восстановления каркасной модели по проекционным изображениям

Первый этап формирования трехмерной каркасной модели – получение двумерной геометро-графической информации прикладной программой. Учитывая, что способ получения двумерных данных в каждой конкретной программной реализации может отличаться, при определении временных характеристик не будем рассматривать данный этап. В основном получаемая двумерная информация имеет свою локальную систему координат (ЛСК), не связанную с будущей трехмерной моделью. Чтобы иметь возможность построения каркасной модели, необходимо преобразовать имеющуюся систему координат проекционных изображений в эпюр Монжа, содержащий информацию уже о трех (x, y, z) координатах. Укрупненный алгоритм данного преобразования координат был представлен на рисунке 3.3

Вычислительная сложность данного этапа напрямую зависит от способа хранения геометро-графической информации внутри прикладной программы. В экспериментальной программной реализации все вершины на проекционных изображениях хранятся в виде односвязного списка. В среднем, на каждом проекционном изображении находится равное количество точек, то есть при обработке только одного проекционного изображения используется треть от общего числа вершин.

В реализованном алгоритме этап получения геометро-графической информации из внешнего источника представляет собой чтение координат всех вершин на проекционных изображениях и их преобразование в систему координат эпюра Монжа, что соответствует вычислительной сложности $O(m)$, где m – количество вершин на проекционных изображениях.

Вторым этапом получения трехмерной каркасной модели является построение точечной трехмерной модели. Точечная трехмерная модель представляет собой облако точек, никак не связанных между собой. Каждая вершина данной модели получается следующим образом:

1. На одном из проекционных изображений возьмем какую-либо точку. Например, возьмем эту точку на виде спереди, получая таким образом сведения о координатах $(x1, z1)$ будущей трехмерной вершины;
2. Найдем все пары подходящих по одной из координат ($x1$ или $z1$ в нашем примере) вершин на следующем проекционном изображении. Например, на виде сверху есть две подходящие вершины – $(x1, y1)$ и $(x1, y2)$;
3. На третьем проекционном изображении попытаемся найти точки с координатами из двух предыдущих шагов. В нашем примере это вершины с координатами $(y1, z1)$ и $(y2, z1)$;
4. Если данные вершины существуют на проекционном изображении, то данные трехмерные точки есть и в трехмерной точечной модели. В примере это точки с координатами $(x1, y1, z1)$ и $(x1, y2, z1)$.

Учитывая, что поиск по списку имеет сложность $O(l)$, где l – длина списка, получаем следующее выражения для вычислительной сложности этапа формирования трехмерной точечной модели:

$$O(m/3) * (O(m/3) + O(m/3)) = 2 * O(m/3)^2 = O(m^2/9) = O(m^2), (11)$$

где m - количество вершин на проекционных изображениях.

На третьем этапе путем добавления ребер в точечную модель получается искомая каркасная трехмерная модель. Ребра же добавляются исходя из следующего утверждения: если для точек $(x1, y1, z1)$, $(x2, y2, z2)$ верно что на соответствующих проекционных изображениях существуют ребра $(x1, z1)-(x2, z2)$; $(x1, y1)-(x2, y2)$; $(y1, z1)-(y2, z2)$ или соответствующие точки совпадают, то, скорее всего, существует ребро $(x1, y1, z1)-(x2, y2, z2)$.

Здесь возможно выполнение задачи двумя способами.

Первый способ формирования списка ребер трехмерной каркасной модели состоит в поиске ребер между всеми вершинами:

1. для каждой вершины взять все последующие за ней в списке трехмерной точечной модели;
2. для каждой пары вершин проверить связь на всех трех проекциях, то есть выполнить поиск на всех трех проекционных изображениях проекций обеих точек и сверить списки их ребер.

Вычислительная сложность первого этапа – $O(n^2)$, сложность второго этапа – $3 * O(m)$, таким образом сложность данного способа формирования списка ребер каркасной модели:

$$O(n^2) * 3 * O(m) = O(n^2 * m), (12)$$

где n – количество вершин в трехмерной точечной модели,

m – количество вершин на проекционных изображениях.

Второй способ формирования списка ребер трехмерной каркасной модели, использованный в реализованном алгоритме, состоит в:

1. для каждой вершины из списка точечной модели найти её проекцию на две координатные плоскости – сложность этапа $2 * O(m)$;
2. для каждой проекции взять список связанных вершин и найти их восстановленные в точечной модели вершины – сложность этапа $O(k * n)$ (k – среднее количество ребер, исходящих из вершины);
3. для каждой полученной на втором этапе вершины и для начальной проверить связь на всех трех проекциях, то есть выполнить поиск на всех трех проекционных изображениях проекций обеих точек и сверить списки их ребер – сложность этапа $3 * O(m)$.

Таким образом получаем, что вычислительная сложность данного способа формирования списка ребер трехмерной каркасной модели равна:

$$2 * O(m) * O(k * n) * 3 * O(m) = 6 * O(m^2 * k * n) = O(m^2 * k * n), (13)$$

где m – количество точек на проекционных изображениях;

k – среднее количество ребер, исходящих из одной вершины;

n – количество вершин в трехмерной точечной модели.

Исходя из вышеописанных этапов формирования каркасной трёхмерной модели на основе проекционных изображения, получаем следующую

вычислительную сложность реализованного алгоритма:

$$O(m) + O(m^2) + O(m^2 * k * n) = O(m^2 * k * n), (14)$$

где m – количество точек на проекционных изображениях;

k – среднее количество ребер, исходящих из одной вершины;

n – количество вершин в трехмерной точечной модели.

4.2 Программная реализация представленного алгоритма

Данный программный комплекс предназначен для визуализации результатов работы алгоритма, описанного в 3 главе. Внешний вид реализованного программного комплекса представлен на рисунке 4.1.

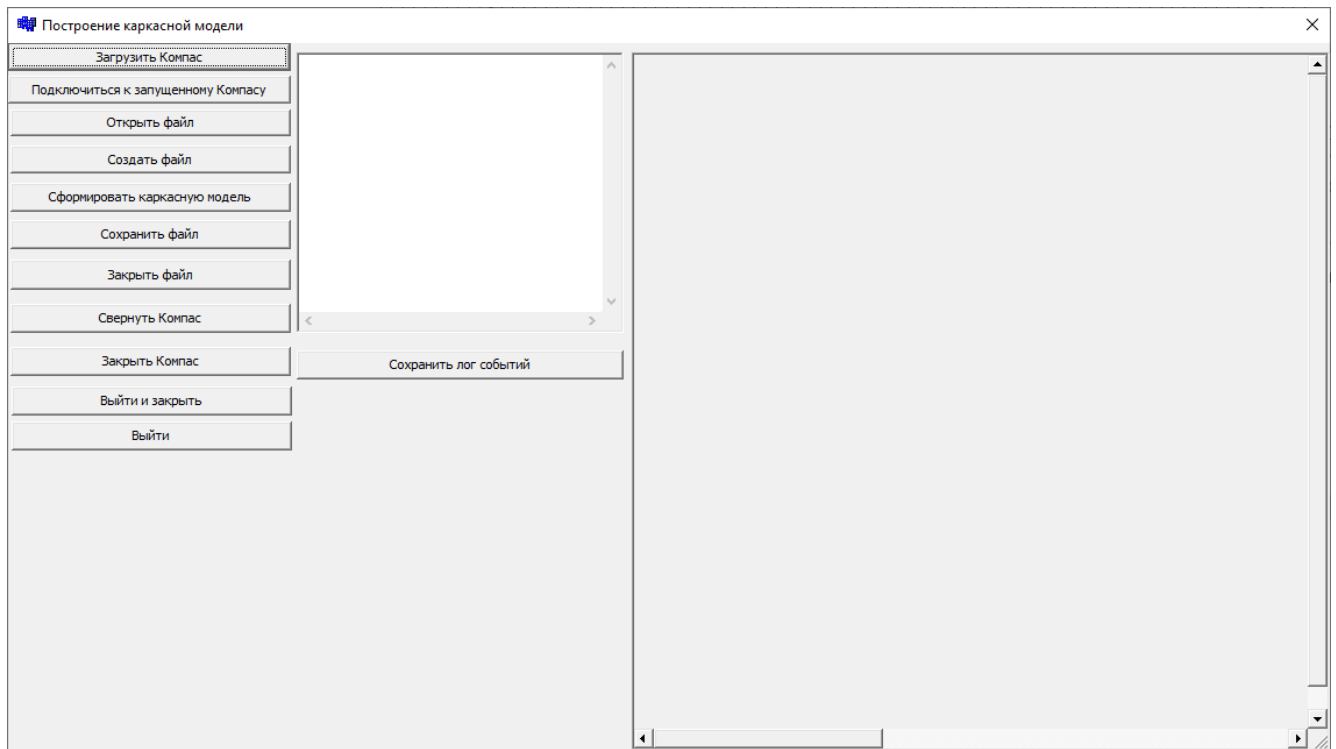


Рисунок 4.1 - Внешний вид демонстрационного комплекса.

Возможности программного комплекса:

- Кнопка “Загрузить Компас” позволяет запустить имеющийся в системе экземпляр программы Компас 3D и после его загрузки подключиться к данному экземпляру при помощи COM-объекта
- Кнопка “Подключиться к запущенному Компасу” позволяет создать COM-объект для взаимодействия с заранее запущенным экземпляром программы Компас 3D.
- Кнопка “Открыть файл” вызывает системный диалог открытия файла и дает команду подключенному экземпляру программы Компас 3D на открытие указанного файла.
- Кнопка “Создать файл” дает команду подключенному экземпляру программы Компас 3D на создание нового документа вида ”чертеж”.

- Кнопка “Сформировать каркасную модель” запускает алгоритм создания каркасной модели на основе проекционных изображений, описанный в 3 главе данной диссертации.
- Кнопка “Сохранить файл” дает команду подключенному экземпляру программы Компас 3D на сохранение открытого документа.
- Кнопка “Закрыть файл” дает команду подключенному экземпляру программы Компас 3D на закрытие текущего открытого документа без сохранения данных.
- Кнопка “Свернуть Компас” дает команду подключенному экземпляру программы Компас 3D на сворачивание в системный трей.
- Кнопка “Закрыть Компас” дает команду подключенному экземпляру программы Компас 3D на закрытие. При наличии несохраненных открытых документов пользователю будет предложено сохранить внесенные изменения.
- Кнопка “Выйти и закрыть” дает команду подключенному экземпляру программы Компас 3D на закрытие и после успешного выполнения закрывает демонстрационный комплекс.
- Кнопка “Выйти” закрывает демонстрационный комплекс без завершения работы подключенного экземпляра программы Компас 3D.

Белая область правее вышеперечисленных кнопок нужна для вывода промежуточной информации о стадиях работы алгоритма по формированию каркасной модели: количество вершин на всех считываемых проекционных изображениях, количество сгенерированных вершин в точечной модели, количество ребер, получившихся в каркасной модели. После завершения работы алгоритма в это же окошко будет выведена информация о времени, затраченном на каждый этап работы программного комплекса. Кнопка “Сохранить лог событий” позволяет сохранить все вышеприведенные текстовые данные в отдельный файл.

4.3 Сравнение ручного построения трехмерной детали и автоматического

Первый вид эксперимента – сравнение скорости построения трехмерной модели путем ручного ввода ГГИ в САПР и автоматизированного построения трехмерной каркасной модели. В среднем на создание простой трехмерной модели тратится от 2 до 5 минут[48]. Экспериментальная программная реализация показывает результат уже через 0.5-3 секунды после запуска процесса чтения двумерной информации на простых трехмерных моделях. Чертеж такой детали и её каркасная модель, восстановленная разработанным программным комплексом, представлены на рисунке 4.2

Вышеизложенные результаты эксперимента позволяют говорить об ускорении процесса получения трехмерной модели по проекционным изображениям в 60-200 раз в зависимости от сложности восстанавливаемой модели и мощности центрального процессора.

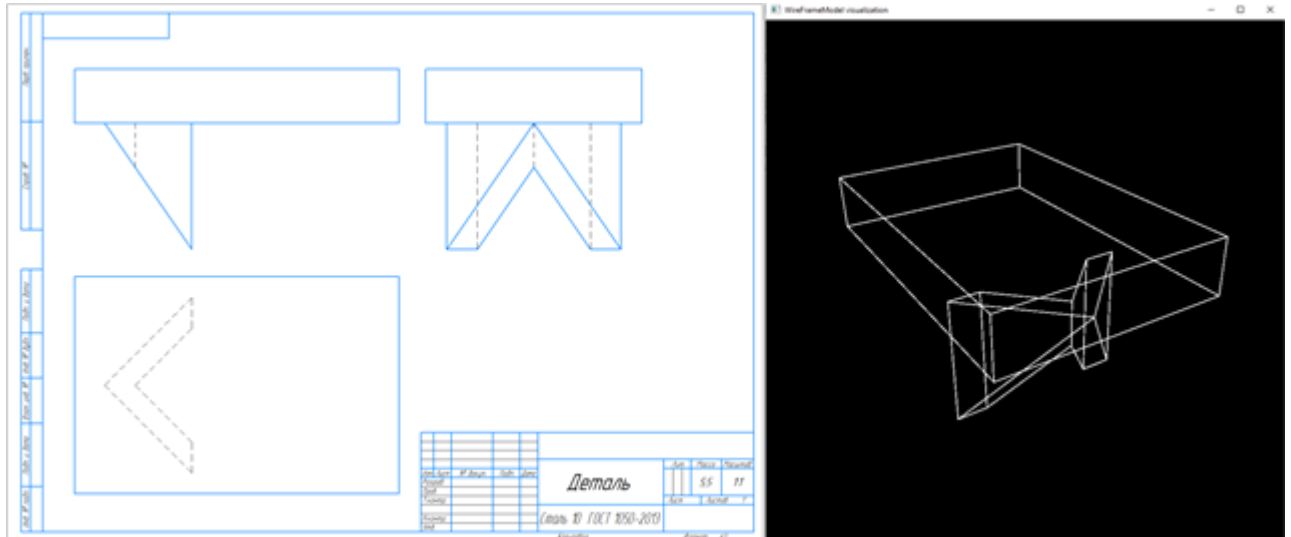


Рисунок 4.2 - Простая трехмерная модель, в ручном режиме строится около 5 минут, восстановленная автоматически за примерно 3 секунды.

4.4 Сравнение времени чтения геометрической информации из внешнего источника и времени восстановления модели

Второй вид эксперимента – сравнение времени, которое тратится алгоритмом на выполнение своих различных составных частей: реализацию операций чтения чертежа, получения точечной трехмерной модели и её преобразования в каркасную.

Первая часть исследования проводилась на несложных трехмерных моделях – до 100 вершин. Эксперимент показал, что самый большой процент времени тратится на получение ГГИ из внешнего источника – до 99%. После выполнения операции чтения вся двумерная геометрическая информация помещается в виртуальное адресное пространство разработанного программного комплекса и соответственно на доступ к ней тратится существенно меньше времени. На экспериментальных чертежах время восстановления точечной и каркасной моделей не превышало 1 мс.

Вторая часть исследования включала в себя более сложные модели – более 100 вершин. Для тестирования были взяты трехмерные модели с сайта thingiverse.com. Данные, полученные в ходе исследования[18], приведены в таблице 4.1, проиллюстрированы на рисунках 4.3, 4.4, 4.5.

Таблица 4.1 - Временные характеристики разработанного алгоритма

Количество вершин	Общее время восстановления, сек.	Время чтения двумерных данных, с	Время восстановления точечной модели, с	Время получения каркасной модели, с
844	29,859	28,125	0,5	1,234
2440	206,547	185,140	1,282	20,125
15616	424,969	95,156	14,78	315,735

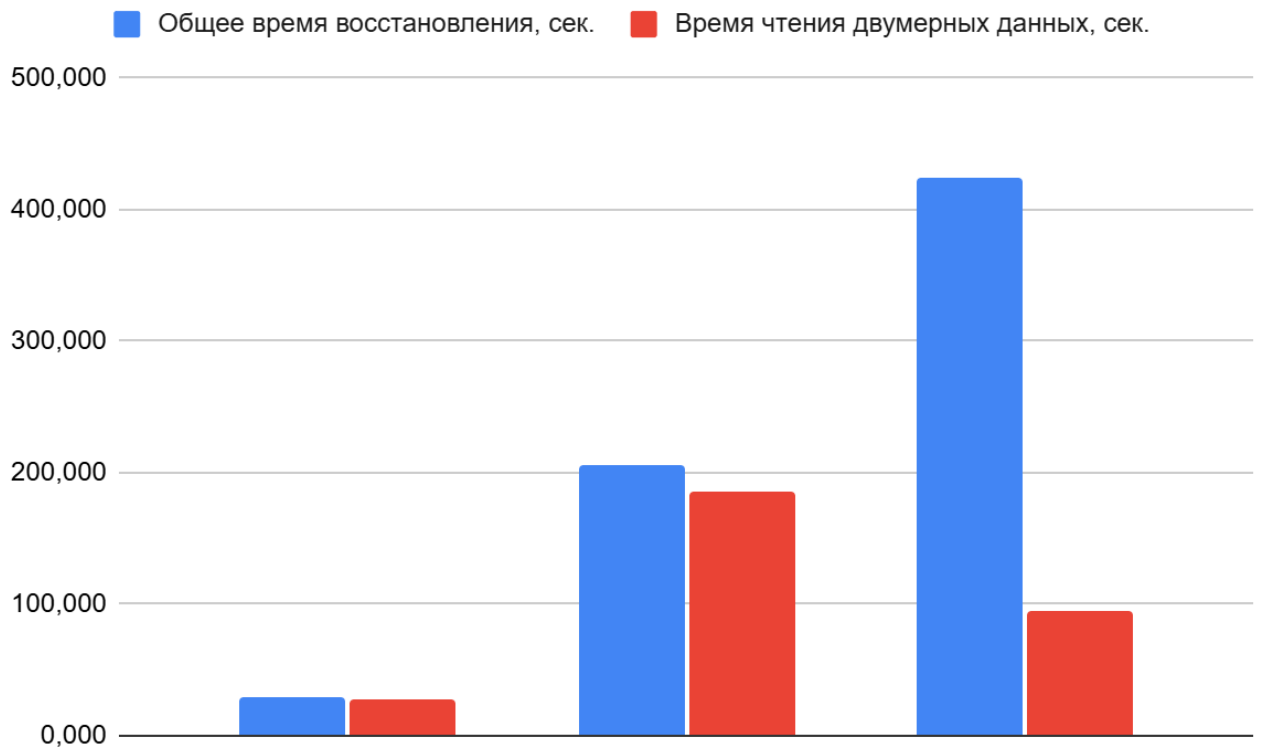


Рисунок 4.3 - Диаграмма, построенная на данных из таблицы 4.1. Показывает сравнение времени, затраченного на чтение ГГИ и времени, затраченного на весь процесс реконструкции модели.

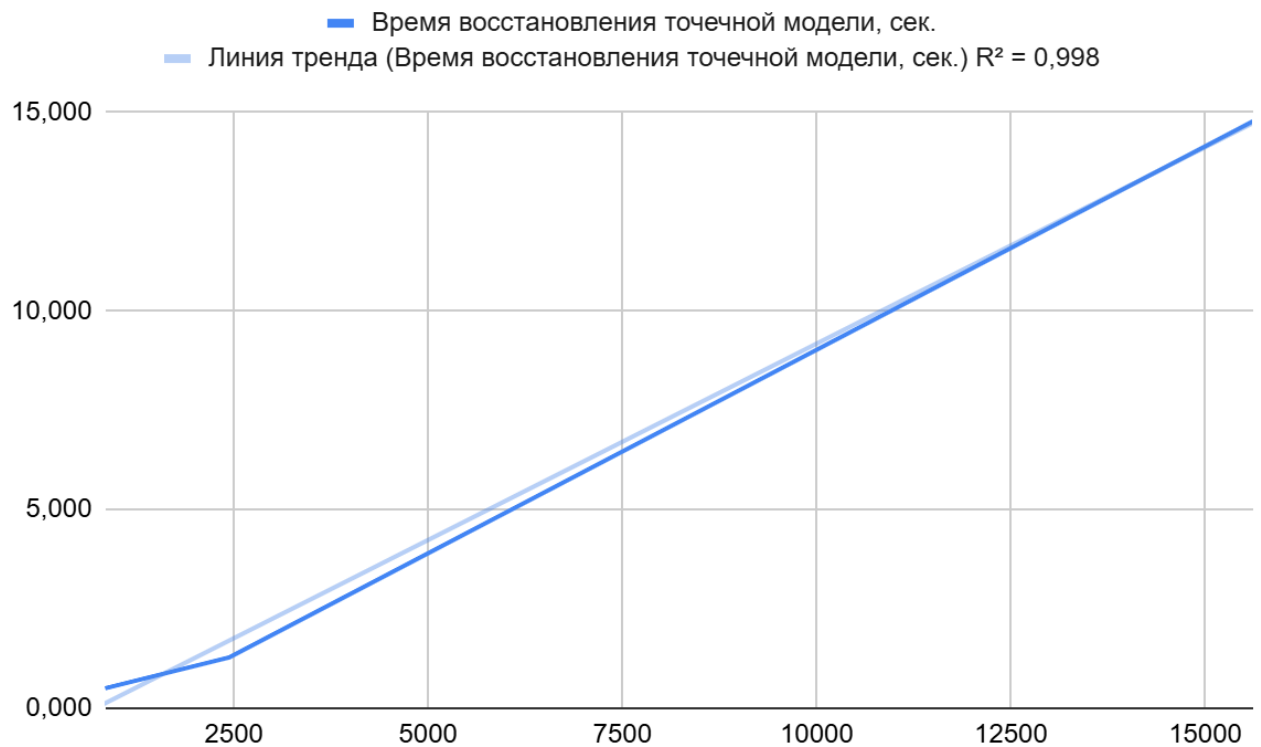


Рисунок 4.4 - Диаграмма, построенная на данных из таблицы 4.1. Показывает зависимости времени восстановления точечных моделей от количества получаемых трехмерных вершин.

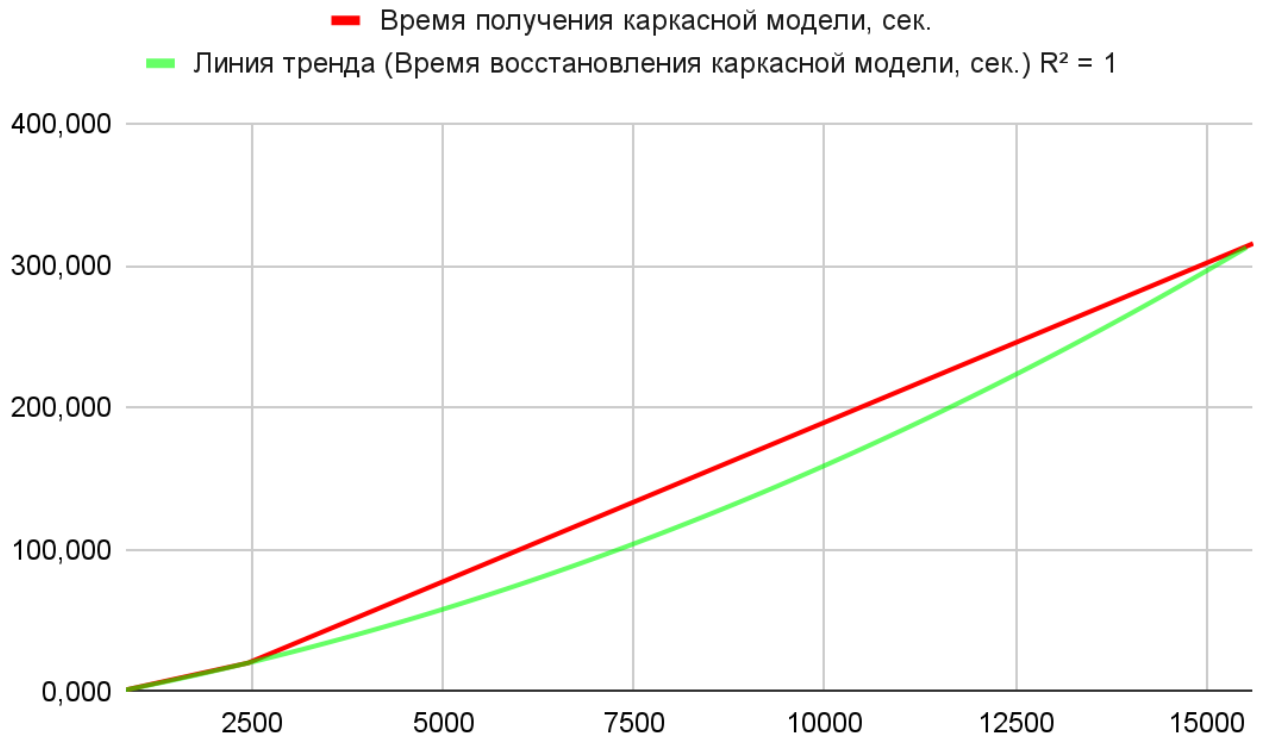
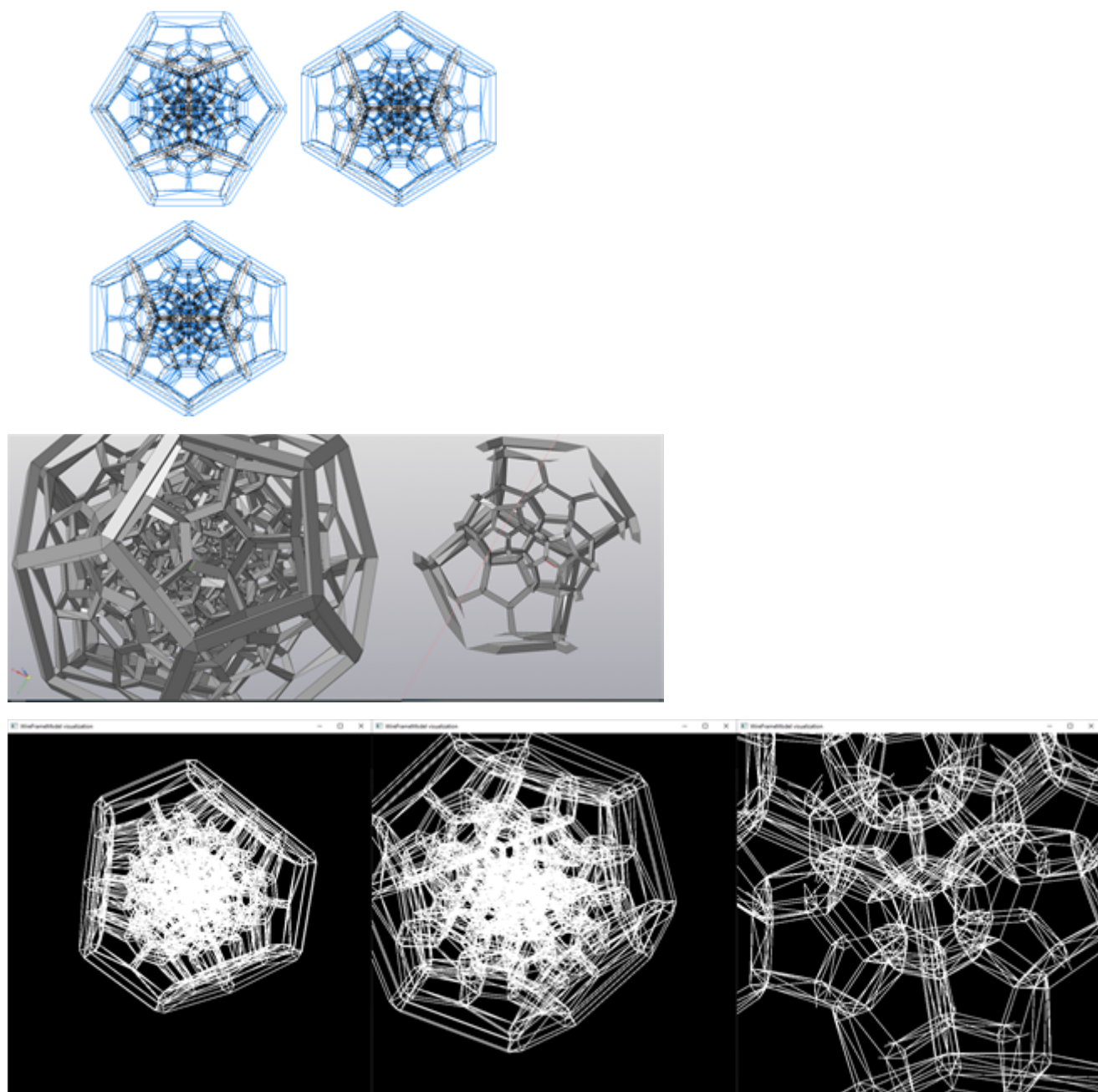


Рисунок 4.5 - Диаграмма, построенная на данных из таблицы 4.1. Показывает зависимости времени восстановления каркасных моделей от количества получаемых трехмерных вершин.

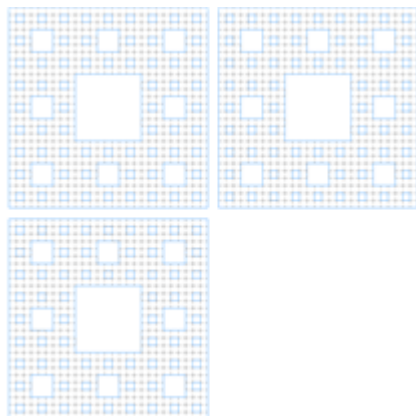
Из проведенного исследования [18] можно сделать следующие выводы:

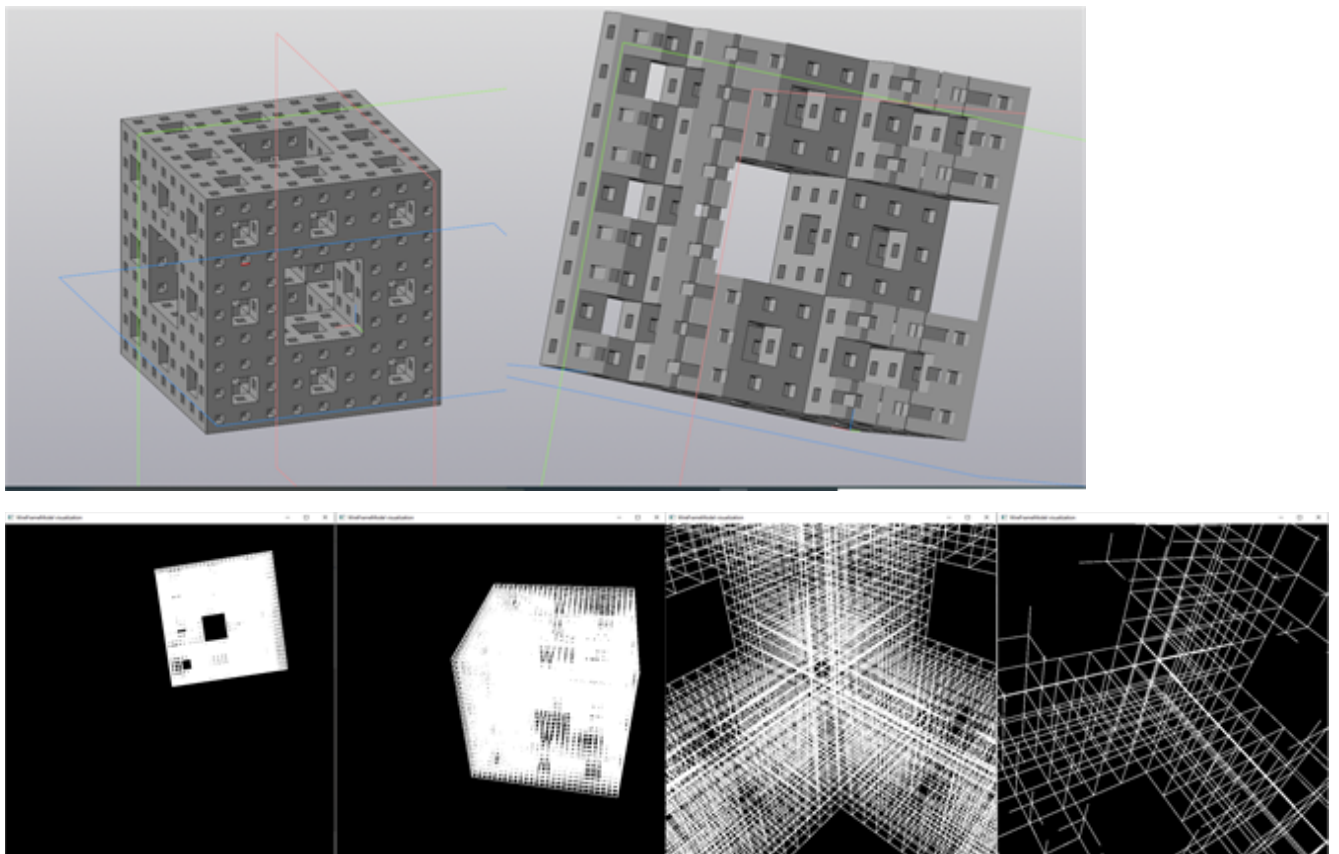
1. На моделях до 1000 вершин около 90% времени занимает чтение ГГИ из внешнего источника и лишь 10% - собственно формирование точечной и каркасной моделей;
2. Экспериментальные данные показывают сложность формирования точечной трехмерной модели $O(n)$, где n – количество вершин в обрабатываемой модели;
3. Экспериментальные данные показывают сложность алгоритма формирования каркасной трехмерной модели $O(k*n)$, где n – количество вершин в обрабатываемой модели, k -среднее количество ребер, исходящих из одной вершины.

Примеры восстановленных трехмерных каркасных моделей и их чертежи на рисунке 4.6.



а) модель с 2440 вершинами





б) модель, содержащая 15616 вершин

Рисунок 4.6 - Модели, полученные при помощи разработанного программного комплекса.

4.5 Восстановление трехмерных каркасных моделей на примере чертежей, использующихся на производстве

Пять примеров чертежей изделий вида «деталь», которые использовались при выполнении тестирования предложенной технологии, приведены на рисунках 4.7-4.16. В среднем, на одну восстановленную трехмерную модель при ручном проектировании необходимо сделать порядка 10-100 формообразующих операций. Конкретное число операций определяется непосредственно исполнителем и его стилем проектирования в каждом случае. Данные чертежи получены в результате взаимодействия с ЦНИИ “Буревестник”[7].

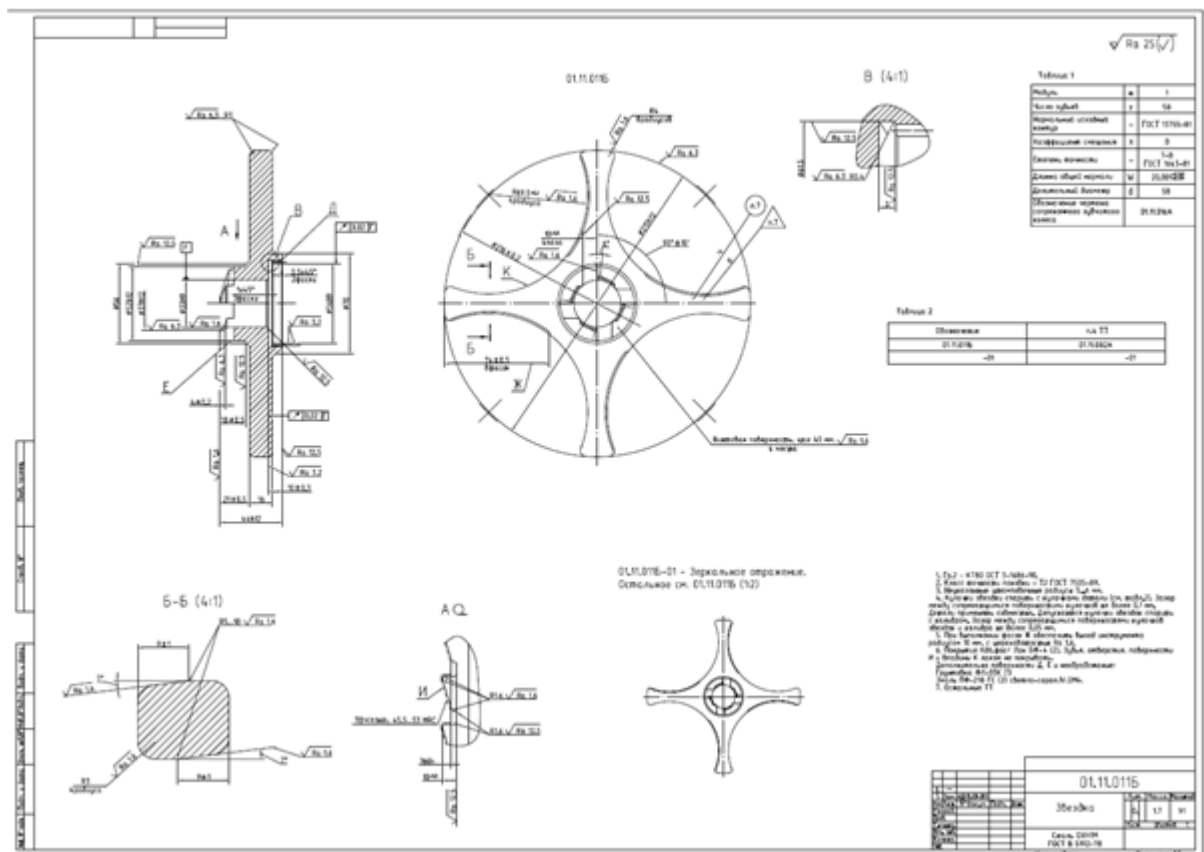


Рисунок 4.7. - Технический многовидовый чертёж детали 1 «Звездка»

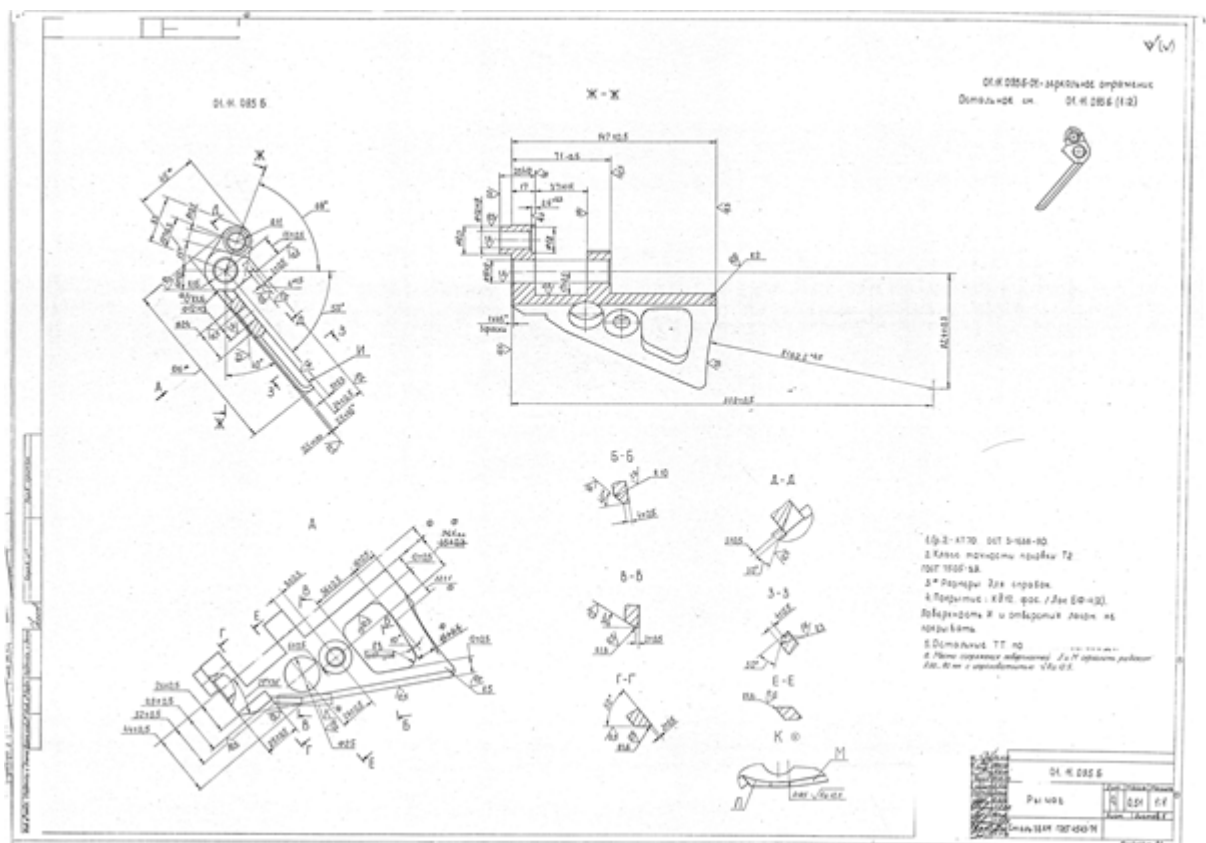


Рисунок 4.8. - Технический многовидовый чертёж детали 2 «Рычаг»

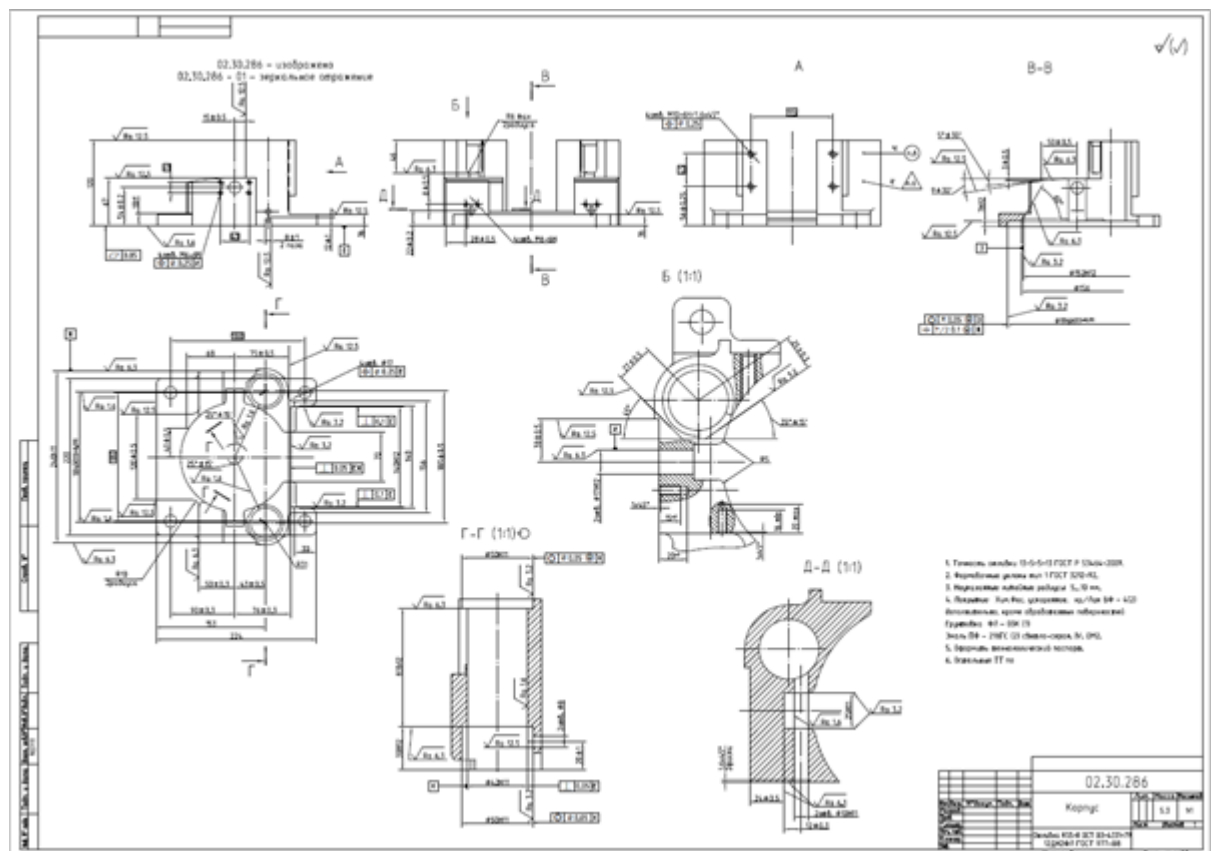


Рисунок 4.9. - Технический многовидовый чертёж детали 3 «Корпус»

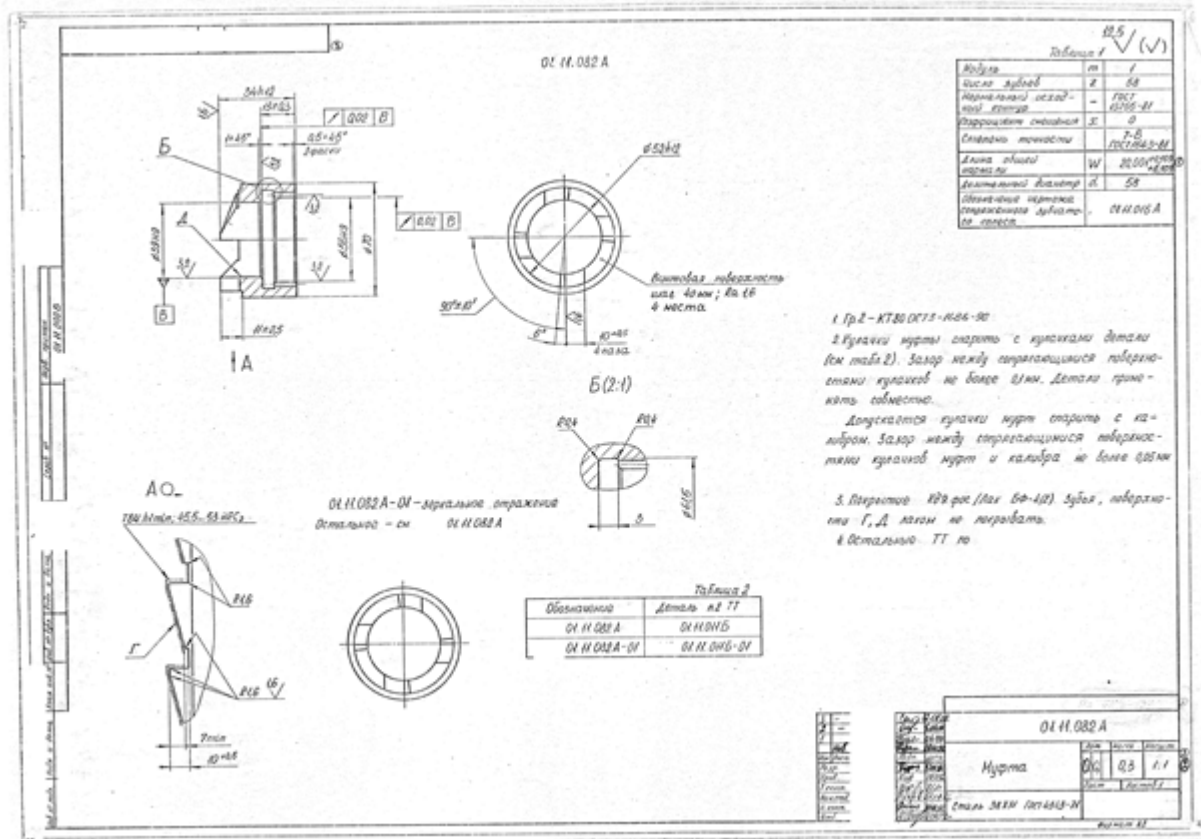


Рисунок 4.10. - Технический многовидовый чертеж детали 4 «Муфта»

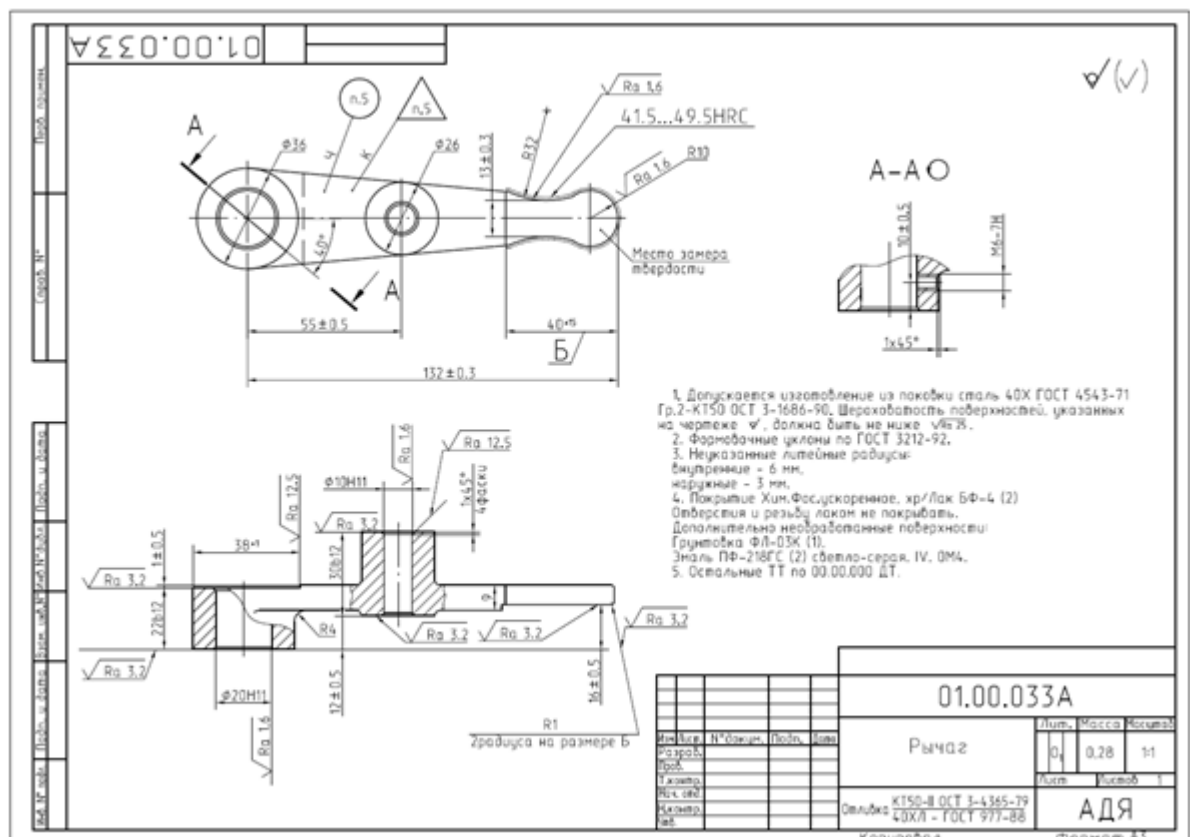


Рисунок 4.11. - Технический многовидовый чертеж детали 5 «Рычаг»

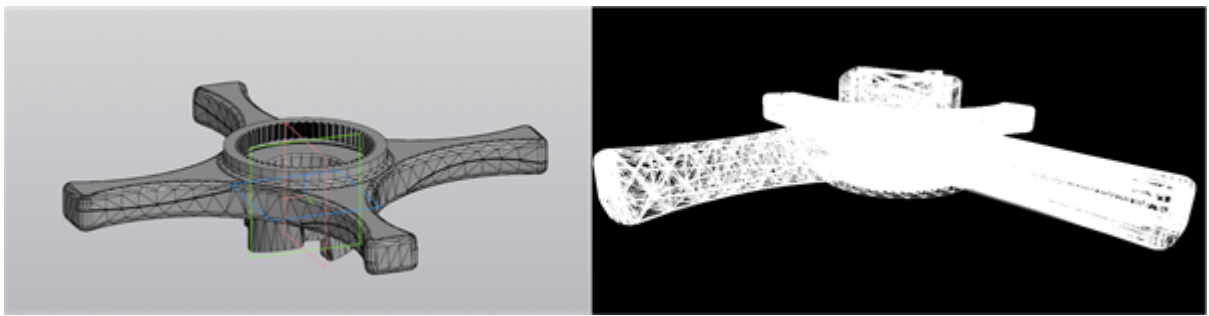


Рисунок 4.12. - Визуализация полученной каркасной модели детали 1 «Звездка»

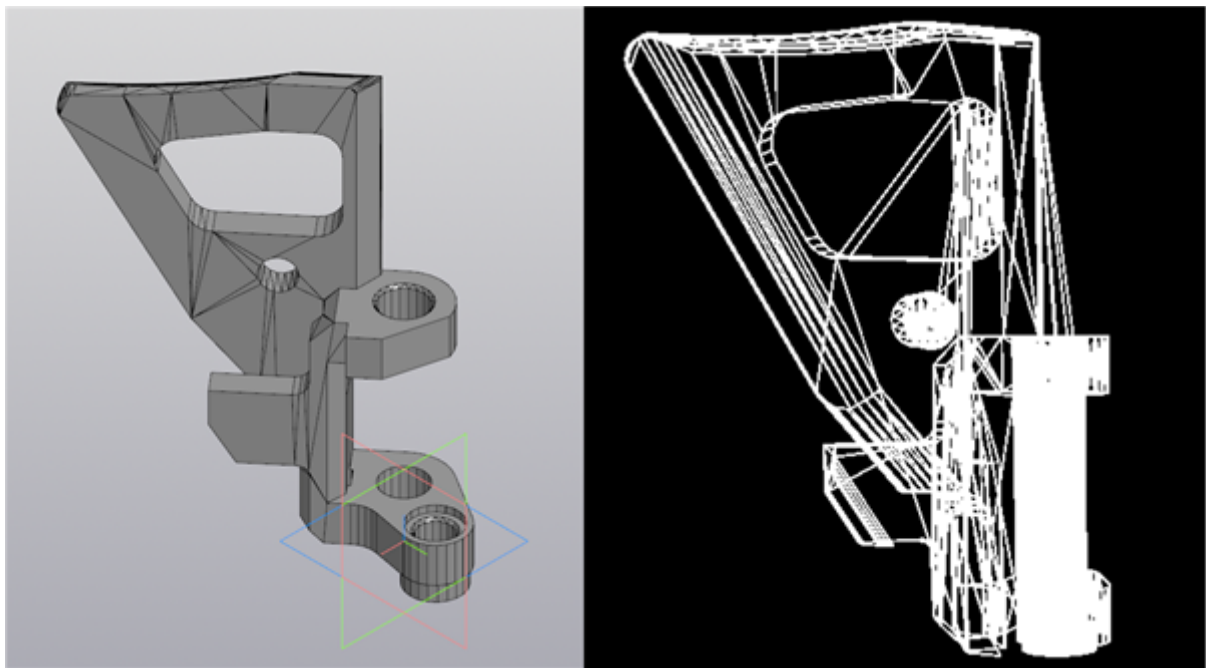


Рисунок 4.13. - Визуализация полученной каркасной модели детали 2 «Рычаг»

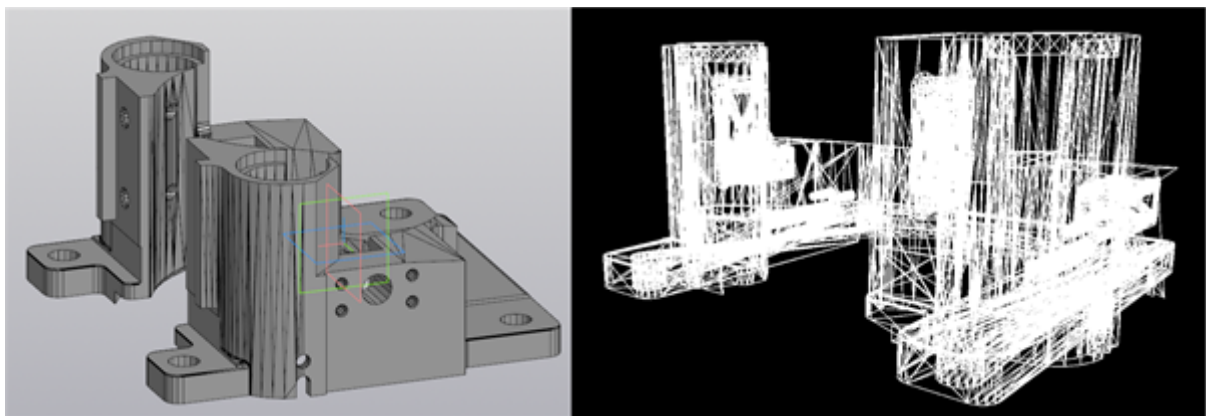


Рисунок 4.14. - Визуализация полученной каркасной модели детали 3 «Корпус»

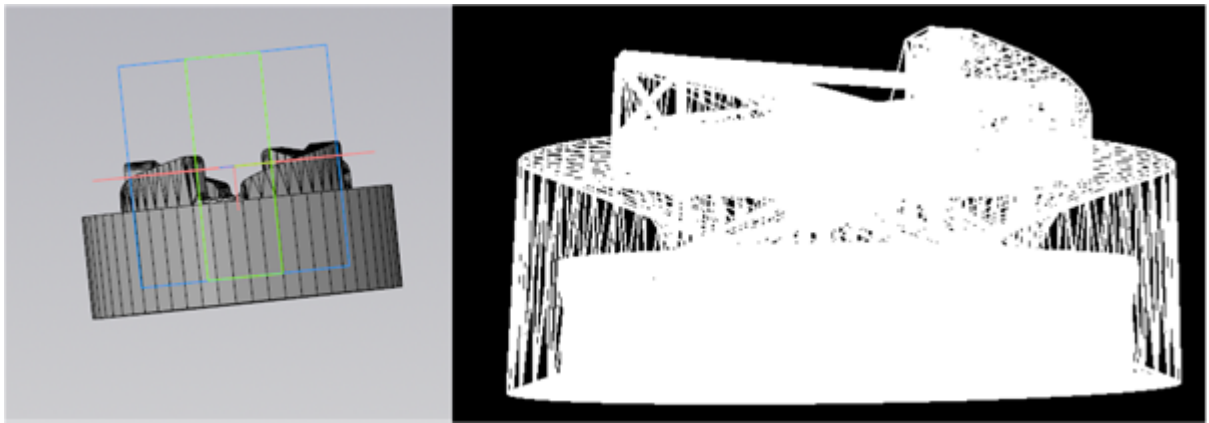


Рисунок 4.15. - Визуализация полученной каркасной модели детали 4 «Муфта»

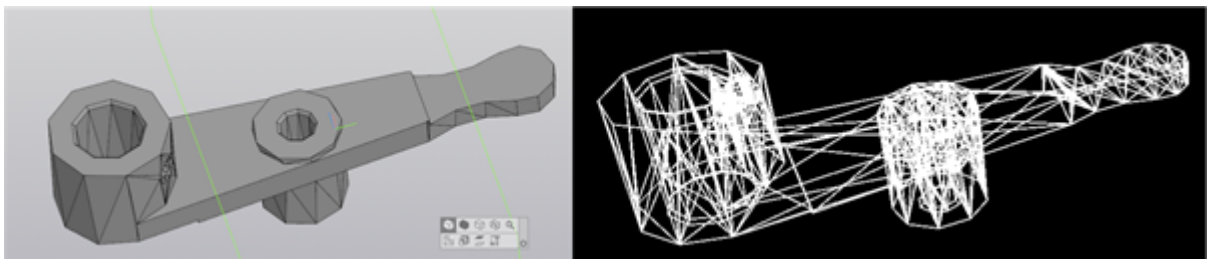


Рисунок 4.16. - Визуализация полученной каркасной модели детали 5 «Рычаг»

В процессе тестирования алгоритма проводилось параллельное фиксирование и сравнение параметров математических моделей объектов, а также временных характеристик восстановления объектов.

В таблице 4.2 приведены данные, полученные при восстановлении трехмерной каркасной модели на основе проекционных изображений, считанных с реальных производственных технических чертежей. Замеры времени были проведены на основе 10 циклов восстановления.

Сравнение данных, представленных в таблице, с временными характеристиками формирования 3D моделей деталей, полученных традиционным способом проектирования (в ходе интерактивного моделирования), позволяет сделать однозначный вывод о преимуществе и перспективности автоматического восстановления моделей объектов по архивным (бумажным и электронным) техническим чертежам.

Таблица 4.2 - Экспериментальные данные

Номер детали	Количество вершин на проекционных изображениях			Количество восстановленных трехмерных вершин	Общее время работы алгоритма, с	Время создания точечной модели, с	Время создания каркасной модели, с
	Фронтальная проекция	Горизонтальная проекция	Профильная проекция				
1	3116	3211	3116	5177	32,969	4,1624	28,806
2	641	472	641	705	0,6125	0,1109	0,5016
3	1509	1257	1509	2016	4,5718	0,7905	3,7813
4	1768	1721	1768	3061	9,1172	1,2909	7,8263
5	182	188	182	225	0,0921	0,0125	0,0796

Также данные из таблицы подтверждают доказанную в пункте 4.1 вычислительную сложность алгоритма

$$O(m^2 * k * n), (15)$$

где m – количество точек на проекционных изображениях;

k – среднее количество ребер, исходящих из одной вершины;

n – количество вершин в трехмерной точечной модели.

По приведенной на рисунке 4.17 диаграмме видно, что с ростом количества вершин на проекционных изображениях увеличивается время формирования точечной модели растет линейно, а время формирования каркасной модели и общее время работы алгоритма увеличивается в квадратичной зависимости.

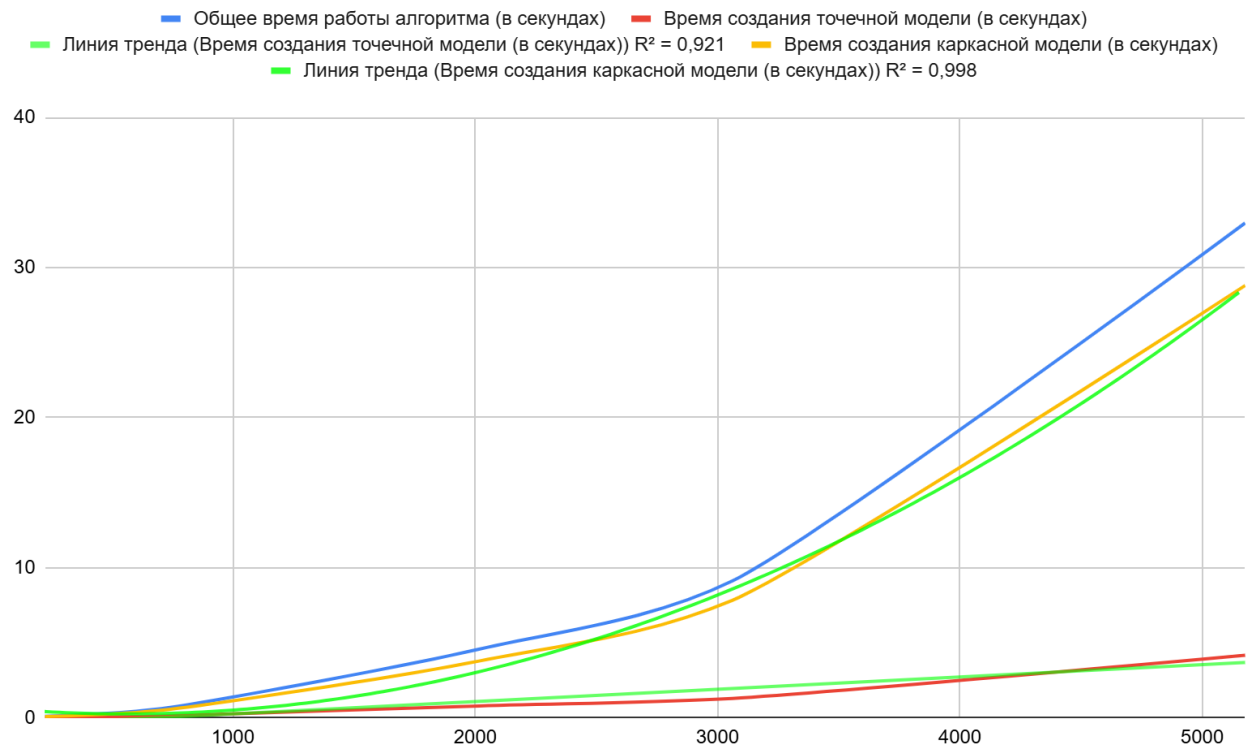


Рисунок 4.17 - Диаграмма, построенная на данных из таблицы 4.2. Показывает соотношение между количеством вершин в точечной модели и временем работы всего алгоритма, а также его составных частей - формирования точечной и каркасной моделей.

4.6 Выводы по четвертой главе

1. Было проведено фиксирование и сравнение параметров математических моделей объектов, а также временных характеристик восстановления объектов. Приведены данные, полученные при восстановлении трехмерной каркасной модели на основе проекционных изображений, считанных с реальных производственных технических чертежей. Сравнение полученных данных, представленных в таблицах, с временными характеристиками формирования 3D моделей деталей, полученных традиционным способом проектирования (в ходе интерактивного моделирования), позволяет сделать однозначный вывод о преимуществе и перспективности автоматического восстановления моделей объектов по архивным (бумажным и электронным) техническим чертежам. Проведена апробация алгоритма на реальных производственных моделях из ЦНИИ “Буревестник”

2. Доказана теоретически и подтверждена экспериментально вычислительная сложность реализованного алгоритма как $O(m^2 * k * n)$, где m – количество точек на проекционных изображениях; k – среднее количество ребер, исходящих из одной вершины; n – количество вершин в трехмерной точечной модели.

ГЛАВА 5. ОРГАНИЗАЦИЯ СИСТЕМЫ УПРАВЛЕНИЯ ДАННЫМИ ДЛЯ ОПЕРАТИВНОЙ РАБОТЫ С ГЕОМЕТРО-ГРАФИЧЕСКОЙ ИНФОРМАЦИЕЙ

5.1 Постановка задачи

В предыдущих главах было показано, что все программные реализации алгоритма синтеза 3D объекта по его проекциям созданы без учета и использования операторов обмена различными данными, что приводит к необходимости создания специализированной дополнительной программной разработки, эквивалентной по своей сути любому текстовому редактору. Набор операторов этой разработки для блока синтеза 3d модели должен обеспечивать функции поиска, чтения-записи конкретных типов данных, удаления данных, например, ложных и «мусора», сжатия данных и других необходимых операций. В противном случае резко возрастают трудоемкость алгоритма синтеза и его эксплуатационные показатели.

Геометро-графическая информация может храниться в различных форматах и системах. Например, для хранения и обмена геоданными используются форматы, такие как Shapefile, GeoJSON, KML, а также более современные форматы, такие как GeoPackage и Web Feature Service (WFS).

Для хранения географической информации также используются географические информационные системы (ГИС), которые могут хранить, обрабатывать и анализировать географические данные. Некоторые из наиболее распространенных ГИС включают в себя ArcGIS, QGIS и GRASS.

Важно отметить, что хранение геометро-графической информации может быть сложным, особенно когда речь идет о больших объемах данных. Поэтому важно правильно организовывать данные и выбирать соответствующие системы и форматы для хранения и обработки геометрической и графической информации.

Для хранения и управления большими объемами данных САПР можно использовать специализированные базы данных, такие как Oracle Spatial или

PostgreSQL с расширением PostGIS, которые поддерживают геометрические типы данных и операции над ними.

Кроме того, существует множество различных инструментов для обработки данных САПР, таких как AutoCAD, SolidWorks, CATIA и другие, которые позволяют редактировать, анализировать и визуализировать данные САПР.

Существуют базы данных, предназначенные специально для хранения геометро-графической информации из САПР (систем автоматизированного проектирования). Такие базы данных обычно называются геометрическими или пространственными базами данных и предназначены для хранения и управления пространственными данными, такими как геометрические объекты и координаты.

Примерами таких баз данных являются Oracle Spatial, PostGIS и Microsoft SQL Server Spatial. Эти базы данных имеют специальные функции и индексы для работы с пространственными данными и обычно поддерживают многие геометрические и географические типы данных, а также различные системы координат и проекции.

Такие базы данных могут использоваться для хранения и управления пространственными данными, такими как картографические данные, геодезические данные, снимки спутников и другие данные, полученные с помощью систем автоматизированного проектирования. Они также могут использоваться для различных приложений, связанных с геометрическими и географическими данными, таких как навигация, геоинформационные системы и т.д.

5.2 Обзор существующих решений

5.2.1 Oracle Spatial

Oracle Spatial - это расширение для базы данных Oracle, которое позволяет работать с географическими данными в базе данных. Оно включает в себя набор функций, типов данных и индексов, которые позволяют хранить, обрабатывать и анализировать географические данные.[76]

Oracle Spatial поддерживает многие геометрические и географические типы данных, включая точки, линии, полигоны, многогранники, поверхности и объемы. Оно также поддерживает различные системы координат и проекций.

С помощью Oracle Spatial можно выполнять различные операции над географическими данными, такие как поиск ближайших объектов, измерение расстояний и площадей, определение пересечений и наложений объектов, а также выполнение пространственного анализа данных.

Кроме того, Oracle Spatial интегрируется с другими инструментами и технологиями Oracle, такими как Oracle Business Intelligence и Oracle Fusion Middleware, что позволяет использовать географические данные в бизнес-анализе и принятии решений.

Oracle Spatial является мощным инструментом для работы с географическими данными в базе данных Oracle и может быть полезным для организаций, работающих с большими объемами географических данных. Однако, для работы с Oracle Spatial необходимы специальные навыки и знания в области баз данных и географической информации.

Основные достоинства Oracle Spatial:

1. позволяет хранить и обрабатывать различные типы геометрических данных, включая точки, линии, полигоны, многогранники и т.д.
2. легко интегрируется с другими инструментами и приложениями, в том числе с системами автоматизированного проектирования (САПР) и геоинформационными системами (ГИС).
3. предоставляет более 200 функций пространственного анализа,

которые позволяют выполнять различные задачи, такие как нахождение ближайших объектов, определение пересечений и наложений объектов, расчет расстояний и площадей, а также анализ сетей и маршрутов.

4. соответствует международным стандартам, таким как OGC Simple Features for SQL и ISO 19125.

5. обеспечивает высокую производительность и масштабируемость при работе с большими объемами геометрических данных.

6. предоставляет высокий уровень безопасности данных, включая авторизацию и аутентификацию пользователей, шифрование данных и аудит действий пользователей.

Основные недостатки Oracle Spatial:

1. Сложность: может быть достаточно сложным в использовании, особенно для новичков. Для работы с ним требуется глубокое знание SQL и понимание геометрических принципов.

2. Цена: является коммерческим продуктом, и его использование может быть дорогим для малых и средних предприятий.

3. Сложность установки: Установка и настройка Oracle Spatial может быть сложной задачей, требующей опыта и знаний в области администрирования баз данных.

4. Зависимость от аппаратной части: Oracle Spatial требует высокой производительности аппаратной части, поэтому для его эффективной работы может потребоваться дорогостоящее оборудование.

5.2.2 PostgreSQL с расширением PostGIS

PostGIS - это расширение для базы данных PostgreSQL, которое позволяет работать с географическими данными в базе данных. Оно включает в себя набор функций, типов данных и индексов, которые позволяют хранить, обрабатывать и анализировать географические данные.[84]

PostGIS поддерживает многие геометрические и географические типы данных, включая точки, линии, полигоны, многогранники, поверхности и объемы. Оно также поддерживает различные системы координат и проекции.

С помощью PostGIS можно выполнять различные операции над географическими данными, такие как поиск ближайших объектов, измерение расстояний и площадей, определение пересечений и наложений объектов, а также выполнение пространственного анализа данных.

PostGIS также интегрируется с другими инструментами и технологиями PostgreSQL, такими как pgAdmin и PostgreSQL JDBC Driver, что позволяет использовать географические данные в приложениях и системах, работающих с базами данных PostgreSQL.

PostGIS является мощным инструментом для работы с географическими данными в базе данных PostgreSQL и может быть полезным для организаций, работающих с большими объемами географических данных. Однако, для работы с PostGIS так же необходимы специальные навыки и знания в области баз данных и географической информации.

Основные достоинства расширения PostGIS для PostgreSQL:

1. является свободным программным обеспечением с открытым исходным кодом, что делает его доступным для всех.
2. соответствует международным стандартам, таким как OGC Simple Features for SQL и ISO 19125.
3. позволяет добавлять пользовательские функции и типы данных, что делает его гибким и адаптивным к различным задачам.
4. предоставляет более 300 функций пространственного анализа, которые позволяют выполнять различные задачи, такие как нахождение ближайших объектов, определение пересечений и наложений объектов, расчет расстояний и площадей, а также анализ сетей и маршрутов.
5. обеспечивает высокую производительность и масштабируемость при работе с большими объемами геометрических данных.

Основные недостатки расширения PostGIS для PostgreSQL:

1. Сложность: PostGIS может быть достаточно сложным в использовании, особенно для новичков. Для работы с ним требуется глубокое знание SQL и понимание геометрических принципов.

2. Необходимость определенного уровня обученности: PostGIS требует от пользователей определенного уровня знаний и опыта в области баз данных и геоинформатики.

3. Ограниченная поддержка визуализации: PostGIS не предоставляет собственных инструментов визуализации данных, поэтому для этой задачи необходимо использовать дополнительное программное обеспечение.

4. Ограниченная поддержка специализированных форматов: PostGIS может иметь ограниченную поддержку специализированных геометрических форматов, используемых в некоторых САПР и ГИС.

5.2.3 Microsoft SQL Server Spatial

Microsoft SQL Server Spatial - это расширение SQL Server, предназначенное для работы с географическими и геометрическими данными в базе данных. Это расширение позволяет пользователям сохранять, обрабатывать и анализировать географические данные в базе данных SQL Server.[59]

SQL Server Spatial имеет множество возможностей, которые делают его полезным для широкого круга приложений, таких как:

1. Анализ территорий: SQL Server Spatial позволяет анализировать геоданные, чтобы выявить территориальные тренды, анализировать использование земель, определять местоположение ресурсов и т.д.

2. Геолокация: SQL Server Spatial позволяет определять местоположение клиентов, магазинов, объектов и т.д. с помощью геоданных.

3. Маршрутизация: SQL Server Spatial может использоваться для определения оптимальных маршрутов для транспортировки грузов или для перемещения людей в городских условиях.

4. Геокодирование: SQL Server Spatial может использоваться для геокодирования адресов, что позволяет определять местоположение объектов на карте.

5. Использование в GIS: SQL Server Spatial может быть использован в геоинформационных системах (GIS) для хранения и обработки геоданных.

SQL Server Spatial поддерживает широкий набор функций для работы с геоданными, включая определение расстояния между точками, пересечение геометрических фигур, определение площади и т.д. SQL Server Spatial также поддерживает использование специальных индексов для оптимизации производительности при выполнении запросов на геоданные.

SQL Server Spatial является коммерческим продуктом, что может быть недоступно для небольших организаций и индивидуальных пользователей. Однако, SQL Server Express включает SQL Server Spatial, что позволяет использовать его бесплатно, но с некоторыми ограничениями по объему данных.

Кроме того, Microsoft SQL Server Spatial имеет хорошую документацию и сообщество, которое активно обсуждает различные аспекты использования этого расширения, что делает его более доступным для пользователей.

Достоинства Microsoft SQL Server Spatial:

1. интеграция с другими инструментами Microsoft: SQL Server Spatial легко интегрируется с другими продуктами Microsoft, такими как Bing Maps и Excel, что облегчает обмен геоданными между приложениями.
2. имеет оптимизированный движок пространственных запросов, что позволяет выполнять запросы на геоданные очень быстро.
3. предоставляет широкий набор функций для работы с геоданными, таких как определение расстояния между точками, пересечение геометрических фигур и т.д.
4. поддерживает стандарты OGC (Open Geospatial Consortium) и ISO (International Organization for Standardization), что обеспечивает совместимость с другими системами обработки геоданных.

Недостатки Microsoft SQL Server Spatial:

1. Высокая стоимость: Microsoft SQL Server Spatial является коммерческим продуктом, и его использование может быть дорогим для небольших организаций или индивидуальных пользователей.
2. Сложность в использовании: SQL Server Spatial может быть сложным для использования для пользователей, не знакомых с SQL-запросами и

геоданными.

3. Ограниченный набор геометрических типов: SQL Server Spatial поддерживает только ограниченный набор геометрических типов, что может быть недостаточным для некоторых приложений.

4. Требования к аппаратному обеспечению: SQL Server Spatial требует высокопроизводительных серверов для обработки больших объемов геоданных, что может быть дорогим для малых организаций.

Все эти базы данных имеют поддержку пространственных типов данных, таких как точки, линии, полигоны и многоугольники, которые используются для хранения и обработки геометрических объектов. Эти системы управления данными также предоставляют набор функций и операторов для работы с геометрическими данными, таких как расчет расстояний между объектами, определение пересечений и объединений объектов.

Кроме того, все эти СУБД являются реляционными, то есть они хранят данные в таблицах, которые могут быть связаны между собой по ключевым полям. Это позволяет эффективно организовывать и обрабатывать большие объемы геометрических данных.

Наконец, все эти системы являются коммерческими продуктами, за исключением PostgreSQL, который является бесплатным и с открытым исходным кодом. Тем не менее, для PostgreSQL доступны коммерческие расширения и инструменты, такие как PostGIS Professional, которые расширяют возможности базы данных для работы с геометрическими данными.

В целом, PostgreSQL с расширением PostGIS, Oracle Spatial и Microsoft SQL Server являются хорошими выборами для хранения геометрической информации, так как они предоставляют поддержку специальных типов данных и функций для работы с геометрическими объектами.

Однако, существуют причины, по которым эти системы не подходят для применения в рамках рассматриваемой технологии преобразования двумерного чертежа в трехмерную модель:

1. Сложность настройки и управления: хранение и обработка

геометрической информации может потребовать сложной настройки и управления базой данных, особенно при работе с большими объемами данных. Это может потребовать опытных администраторов баз данных.

2. Высокая стоимость: Oracle Spatial и Microsoft SQL Server являются коммерческими продуктами, что может привести к высоким затратам на лицензирование и поддержку.

3. Ограничения по производительности: обработка больших объемов геометрических данных может привести к ограничениям по производительности базы данных, особенно при работе с сложными запросами.

4. Низкая масштабируемость: в некоторых случаях может быть сложно масштабировать базу данных для обработки больших объемов геометрических данных, что может привести к ограничениям по производительности.

5. Необходимость использования специальных инструментов: для работы с геометрическими данными может потребоваться использование специальных инструментов, которые не входят в стандартный набор функций и операторов базы данных. В этом случае может потребоваться дополнительная настройка и установка этих инструментов.

6. Избыточность: вышеупомянутые базы данных работают так же с географическими данными, что в решаемой задаче излишне.

Таким образом, при выборе системы управления данными для хранения ГГИ необходимо учитывать конкретные требования проекта и выбирать СУБД, которая лучше всего соответствует этим требованиям.

Для реализации в полной мере технологии по преобразованию бумажного чертежа в электронную модель изделия необходимо создать новый вариант системы управления данными для оперативной работы с двумерной и трехмерной геометро-графической информацией с целью создания новых функциональных операторов различных программных продуктов ЖЦИ.

Двумерные модели представляют из себя:

- чертежи, созданные в системе автоматизированного проектирования, оформленные согласно ЕСКД, в виде файлов либо структурированных

геометрических данных,

- отсканированные и затем векторизированные (автоматически или вручную) чертежи из архивов чертежно-конструкторской документации [92].

Трехмерные модели представлены в виде [43]:

- Точечной модели, представляющей собой совокупность $TM = \{V, A\}$, где

- V - вектор координат X_i, Y_i, Z_i вершин объекта, где $i = 1, \dots, n$;

- A - информационный массив атрибутов вершин V_i .

- Каркасной, представляющей собой совокупность $KM = \{V, R, A\}$, где

- V - вектор координат X_i, Y_i, Z_i вершин объекта, где $i = 1, \dots, n$;

- R - информационный массив, содержащий данные о ребрах R_{ij} , соединяющих вершины V_i и V_j объекта. Такой информационный массив называется матрицей смежности ориентированного или неориентированного графа и может быть представлен в памяти ЭВМ самыми разными способами, в том числе двумерным массивом данных размерностью $n \times n$, либо списковыми структурами [16].

- A - информационный массив атрибутов вершин V_i и ребер R_{ij} .

- Граничной, представляющей собой совокупность $GM = \{G, V, R, A\}$, где

- V, R - информационные массивы, используемые в KM ,

- G - информационный массив граней (отсеков поверхностей), где i -тый элемент массива (i - ая грань) $G_i = \{(V_{1i}, V_{2i}, V_{3i}, \dots, V_{ni}) \in F_i\}$, где $F_i \in F$, а $V_{1i}, \dots, V_{ni} \in V$ задается своими вершинами, перечисленными в порядке обхода грани против часовой стрелки, начиная с произвольно выбранной вершины. При этом контур, охватывающий грань G_i , может быть как односвязным, так и неодносвязным, без самопересечений.

- A - информационный массив атрибутов вершин V_i , ребер R_{ij} и граней G_i , куда, в свою очередь входят следующие массивы:

- F - информационный массив, содержащий коэффициенты уравнений $F_i(x, y, z) = 0$ поверхностей, которые являются носителями

граней G_i [34, 38, 52];

- L – информационный массив, содержащий коэффициенты уравнений $L_{ij}(x, y, z) = 0$ кривых 1 или 2 порядка, которые являются носителями ребер R_{ij} .
- Твердотельной, представляющей собой совокупность $CM = \{G(R), B(M), A\}$, где
 - $G(R)$ - граф сборки конструкции, либо бинарная древовидная структура, определяющие порядок и характер взаимодействия составляющих объектов[57],
 - R - ребра графа, атрибутами которого могут быть параметры формы и положения i -ой модели относительно j -ой модели и знаки регуляризованных теоретико-множественных операций объединения, вычитания и пересечения.
 - Параметрами формы могут быть числовые значения масштабных коэффициентов по каждой из осей в отдельности либо по всем трем осям сразу.
 - Параметрами положения могут быть вектор переноса T и вектор поворота E , задающий углы поворота i -го объекта на заданные углы относительно координатных осей.
 - $B(M)$ - банк моделей пространственных объектов, каждый из которых может быть как непроизводной, так и составной фигурой. Модели объектов, составляющих банк, могут быть сформированы различными средствами и способами и заданы в своей собственной системе координат, не связанной с системой координат проектируемого объекта. Каждой модели поставлено в соответствие символическое имя, по которому идет поиск информации, и имя файла, в котором хранится вся информация, относящаяся к модели.
 - A - атрибуты моделей (материал, плотность, цвет и т.д.)

Отметим также, что одна и та же трехмерная модель может быть представлена сразу во всех вышеперечисленных видах: в ходе применения

информационной технологии преобразования архивной чертежно-конструкторской и технологической документации в электронную модель изделия для одной и той же трехмерной модели сначала заполняется массив точечного представления, на основе её и чертежа образуется каркасная модель, к которой добавляется граничная модель, и данный способ хранения трехмерной ГТИ позволяет избежать ненужного дублирования.

5.3 Макет системы управления для хранения геометрической информации

Для представления макета системы управления и хранения ГГИ авторами предлагается следующая схема [41]:

1. Центральную роль играет хранилище графических данных, содержащее:
 - a. Таблицы для хранения двумерных и трехмерных моделей.
 - b. Отдельные таблицы для хранения топологии моделей - списки ребер, матрицы смежности, данные о гранях и их контурах.
 - c. Таблицы иерархических связей компонентов составных объектов.
 - d. Таблицы пространственных преобразований.
2. Вокруг основного хранилища графических данных располагаются вспомогательные модули:
 - a. Модуль импорта/экспорта, обеспечивающий загрузку и выгрузку данных в разных форматах.
 - b. Модули обработки данных - сжатие, расчет параметров и т.д.
 - c. Поисковый модуль с индексами по именам, пространственным и другим атрибутам.
 - d. Модуль визуализации данных.
 - e. Модуль администрирования и настройки прав доступа.
3. В качестве СУБД можно использовать реляционную, объектно-ориентированную или гибридную СУБД.
4. Доступ к данным может осуществляться через клиентское ПО - CAD/CAM/CAE системы, специальные приложения и т.д.

Такая архитектура позволит гибко и эффективно хранить и использовать разнообразную ГГИ.

5.3.1 Выбор типа базы данных

Выбор типа базы данных для хранения ГГИ является критическим

решением при проектировании информационных систем. В современном мире, где данные пространственной природы становятся все более важными, необходимость эффективного хранения и обработки геометрической информации становится все более значимой.

Одним из наиболее популярных типов баз данных для ГГИ являются графовые базы данных. Графовая модель представления данных позволяет эффективно описывать и анализировать сложные связи между геометрическими объектами, такими как точки, линии и полигоны [85]. Благодаря своей способности описывать сетевую структуру данных, графовые базы данных отлично подходят для моделирования пространственных отношений и поиска путей между геометрическими объектами.

Другим распространенным выбором являются пространственные реляционные базы данных. Реляционная модель хранения данных предоставляет мощные инструменты для организации сложных структур данных и выполнения широкого спектра операций, включая запросы и анализ пространственных данных [54]. Пространственные расширения реляционных баз данных, такие как PostGIS, предоставляют дополнительные возможности для работы с геометрическими данными, такие как индексы пространственной информации и функции анализа.

Однако, при выборе типа базы данных для хранения ГГИ следует учитывать не только функциональность и производительность, но и особенности конкретного применения. Например, если требуется обработка больших объемов данных в реальном времени, то NoSQL базы данных, такие как MongoDB или Apache Cassandra, могут быть более подходящим выбором [53]. Они обладают горизонтальным масштабированием и способностью обрабатывать большие объемы данных без потери производительности.

В целом, выбор типа базы данных для хранения ГГИ зависит от специфических требований проекта, объема и структуры данных, желаемой функциональности и ожидаемой производительности. Подробный анализ этих факторов поможет определить наиболее подходящий тип базы данных для конкретного применения ГГИ.

5.3.2 Обоснование выбора реляционной базы данных

Выбор реляционной базы данных для хранения ГГИ обоснован следующими причинами:

1. Этот вид баз данных обеспечивает структурированное хранение ГГИ с использованием таблиц, строк и столбцов. Это позволяет организовать ГГИ в виде таблиц с определенными полями для координат, размеров, форм и других свойств геометрических объектов.

2. Реляционные базы данных предоставляют мощный язык запросов (например, SQL), который позволяет легко выполнять сложные запросы и аналитические операции над геометрическими данными. Например, вы можете выполнить поиск объектов по определенным критериям, производить агрегирование данных или осуществлять пространственные запросы, такие как поиск объектов в заданном радиусе или пересечение объектов.

3. Поддержка механизма транзакций гарантирует согласованность данных при одновременном доступе нескольких пользователей или при выполнении сложных операций, включающих несколько таблиц. Это особенно важно при работе с геометрическими данными, где точность и соответствие между различными объектами могут быть критически важными.

4. Реляционные базы данных широко используются и поддерживаются большинством современных систем и приложений. Это облегчает интеграцию геометрических данных с другими компонентами вашей информационной системы, такими как веб-приложения, аналитические инструменты или системы управления данными.

5. У выбранного типа базы данных существует широкий выбор коммерческих и открытых решений, которые обладают богатым функционалом и имеют активное сообщество пользователей и разработчиков. Это обеспечивает поддержку, расширяемость и возможность получения помощи при необходимости.

В целом, реляционная база данных является хорошим выбором для хранения ГГИ благодаря своей структурированной природе, поддержке сложных

запросов и аналитики, согласованности данных, интеграции с другими системами, а также расширяемости и поддержке со стороны сообщества.

5.4 Описание хранения двумерной ГГИ

Для хранения двумерной ГГИ, представленной в виде чертежей (содержащих точки, отрезки, окружности, дуги, штриховки, условные обозначения, и др.) можно использовать различные подходы к архитектуре базы данных. Авторами предлагается следующий способ организации базы данных[25]:

1. Таблица "Чертежи" (Drawings): таблица содержит информацию о каждом чертеже, включая уникальный идентификатор чертежа, название, описание и другие метаданные чертежа.

2. Таблица "Объекты" (Objects): данная таблица содержит информацию обо всех объектах на чертеже. Каждый объект будет иметь уникальный идентификатор, тип объекта (точка, отрезок, окружность и т.д.), координаты или параметры, связанные с этим объектом (например, радиус окружности или длина отрезка) и ссылку на соответствующий чертеж.

3. Таблица "Штриховка" (Hatching): если чертеж содержит штриховку для заполнения областей, эта таблица может содержать информацию о шаблонах штриховки, используемых на чертеже. Это может включать параметры штриховки, такие как стиль, плотность, цвет и т.д.

4. Таблица "Условные обозначения" (Symbols): если чертеж содержит условные обозначения, эта таблица может содержать информацию о каждом условном обозначении. Каждое условное обозначение будет иметь уникальный идентификатор, тип символа, координаты положения символа на чертеже и ссылку на соответствующий чертеж.

5. Дополнительные таблицы: в зависимости от конкретных требований и особенностей чертежей, могут быть созданы дополнительные таблицы для хранения специфической информации. Например, таблица "Дуги" может использоваться для хранения данных о дуговых объектах, таблица "Отношения" может использоваться для описания связей между объектами и т.д.

Связи между таблицами могут быть установлены с помощью первичных и

внешних ключей. Например, у каждого объекта на чертеже может быть внешний ключ, указывающий на соответствующий чертеж. Это позволяет связать объекты с конкретным чертежом. Данные связи наглядно показаны на рисунке 5.1.

Такая архитектура базы данных позволяет эффективно организовать хранение и управление ГГИ чертежей, обеспечивая удобный доступ к объектам, их свойствам и метаданным.

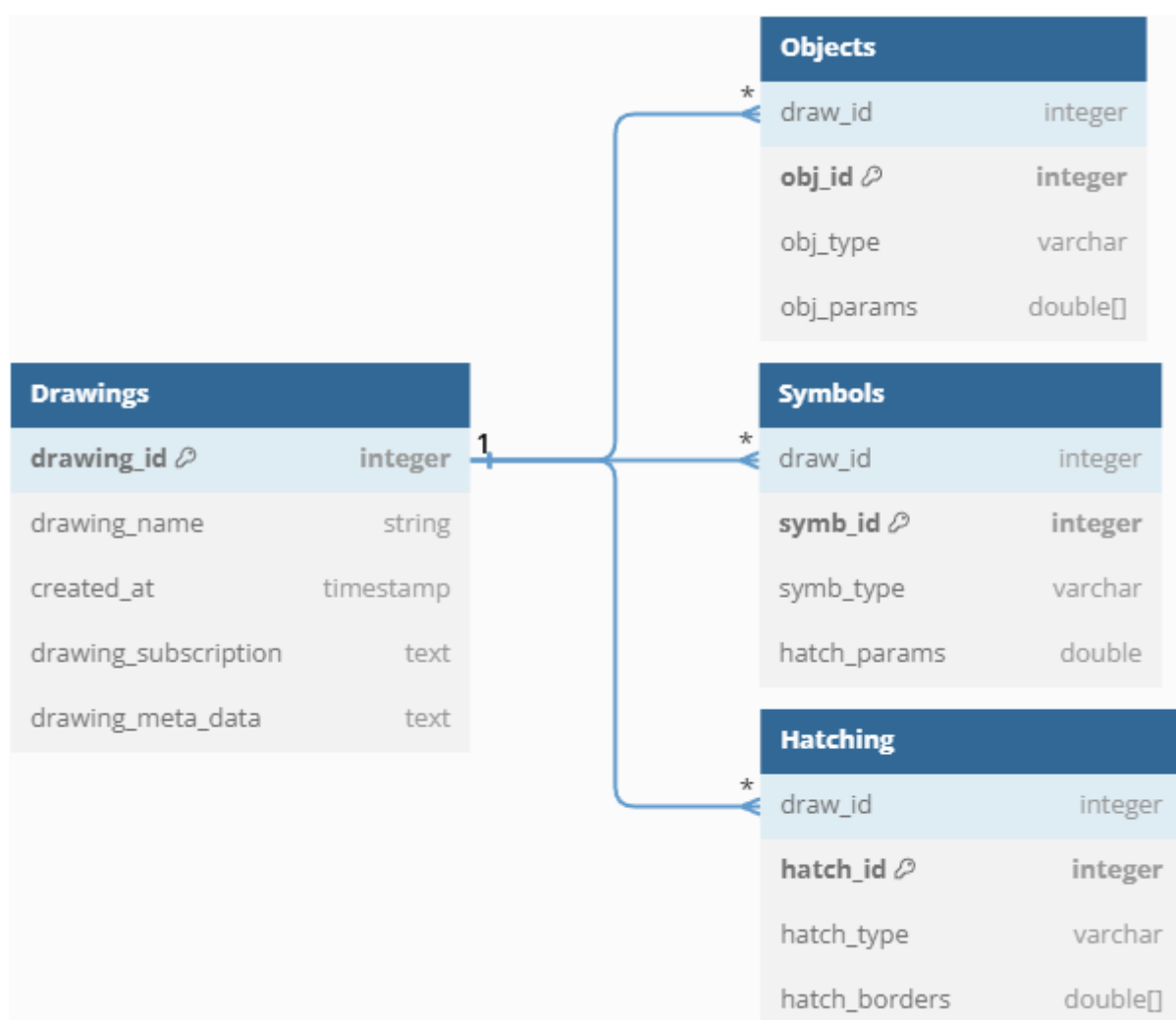


Рисунок 5.1. - Предлагаемая схема хранения двумерной информации

5.5 Описание хранения трехмерной ГГИ

Предлагается следующая архитектура базы данных для хранения трехмерных геометрических моделей:

- Таблица "Модели":
 - Идентификатор модели (уникальный идентификатор)
 - Название модели
 - Задействованные типы модели (точечная, каркасная, граничная, твердотельная)
- Таблица "Точки":
 - Идентификатор точки (уникальный идентификатор)
 - Координаты точки (x, y, z)
 - Идентификатор модели (внешний ключ к таблице "Модели")
- Таблица "Ребра":
 - Идентификатор ребра (уникальный идентификатор)
 - Идентификатор начальной точки (внешний ключ к таблице "Точки")
 - Идентификатор конечной точки (внешний ключ к таблице "Точки")
 - Идентификатор модели (внешний ключ к таблице "Модели")
- Таблица "Грани":
 - Идентификатор грани (уникальный идентификатор)
 - Идентификатор модели (внешний ключ к таблице "Модели")
- Таблица "Поверхности":
 - Идентификатор поверхности (уникальный идентификатор)
 - Идентификатор модели (внешний ключ к таблице "Модели")
- Таблица "Геометрические примитивы":
 - Идентификатор примитива (уникальный идентификатор)
 - Параметры формы геометрического примитива (например, для прямого кругового цилиндра необходимо задать диаметр основания и

высоту цилиндра)

- Таблица "Булевы операции":
 - Идентификатор операции (уникальный идентификатор)
 - Тип операции — объединение, пересечение или вычитание
 - Идентификатор первого примитива (внешний ключ к таблице "Геометрические примитивы") или трехмерной модели (внешний ключ к таблице "Модели")
 - Параметры положения первого операнда
 - Идентификатор второго примитива (внешний ключ к таблице "Геометрические примитивы") или трехмерной модели (внешний ключ к таблице "Модели")
 - Параметры положения второго операнда
 - Идентификатор модели (внешний ключ к таблице "Модели")

Такая структура базы данных позволяет представить различные типы трехмерных геометрических моделей и хранить их компоненты отдельно.

Для точечной модели в таблице "Точки" будут храниться координаты каждой отдельной точки, а в таблице "Модели" будет указан тип модели.

Для каркасной модели в таблице "Ребра" будут храниться связи между вершинами (точками) модели, а также указываться соответствующая модель в таблице "Модели".

Для граничной модели в таблице "Грани" будут храниться грани, составленные из ребер, а также указываться соответствующая модель в таблице "Модели".

Для твердотельной модели используется вариант её хранения через историю построения с использованием булевых операций из примитивов или промежуточных моделей [36].

Таким образом, архитектура базы данных предоставляет возможность организации структуры трехмерных геометрических моделей и их компонентов, а также связей между ними. Все связи наглядно показаны на рисунке 5.2.

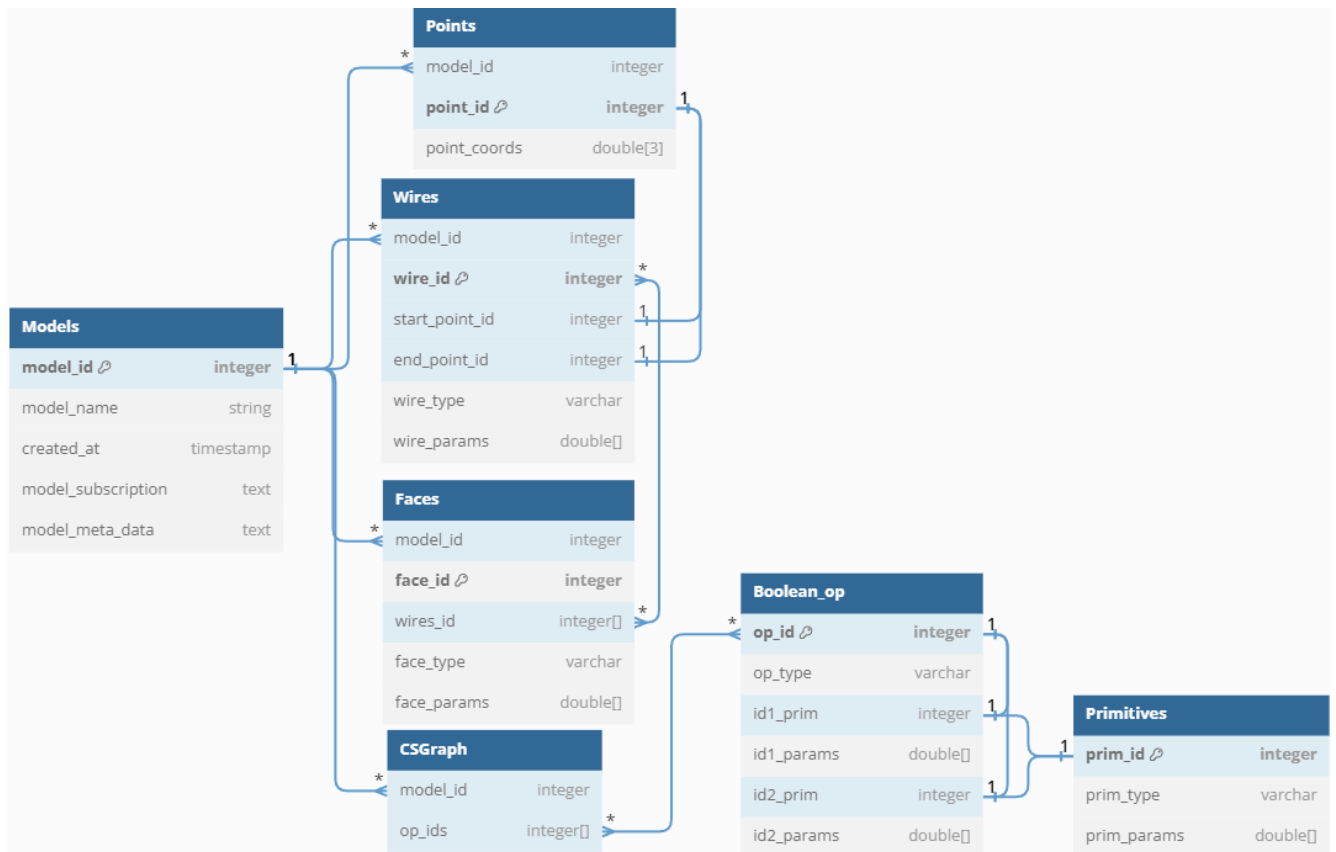


Рисунок 5.2 - Предлагаемая схема хранения трехмерной информации.

Для полноценной работы предлагаемой в данной диссертации технологии преобразования бумажного чертежа в электронную модель изделия необходима организация непрерывной связи между двумерными проекционными изображениями и генерируемой по ним трехмерной моделью. Способ организации такой связи показан на рисунке 5.3.



Рисунок 5.3 - Связь между двумерными и трехмерными данными.

5.5.1 Применимость различных способов хранения и управления ГГИ в поставленной задаче

Способы хранения и управления ГГИ являются ключевыми методами и структурами данных в области компьютерной графики, компьютерного зрения и автоматизированного проектирования. Они служат для представления и обработки сложных геометрических моделей, состоящих из точек, линий, поверхностей и объемных тел. Важной целью таких структур хранения является оптимизация использования памяти и времени выполнения операций над этими моделями.

Один из основных алгоритмов, широко используемых для решения этой задачи, - это триангуляция Делоне. Она разбивает множество точек на непересекающиеся треугольники таким образом, чтобы внутри окружности, описанной вокруг каждого треугольника, не содержалось других точек. В рассматриваемой задаче нет необходимости использовать данный алгоритм, так как вся информация о гранях хранится в аналитическом виде.

Другой распространенный подход - использование квадродеревьев и октаэдрических деревьев.

Квадродеревья являются иерархическими структурами данных, которые разделяют пространство на регулярные или иерархические ячейки для более эффективного поиска объектов по их положению и из каждой вершины квадродерева выходит только четыре ребра.

Октаэдрические деревья аналогичны квадродеревьям, но используются в трехмерном пространстве. Из каждой вершины октодерева выходит только по восемь ребер.

Одним из важных классов алгоритмов являются алгоритмы проверки пересечения, которые определяют, пересекаются ли два или более геометрических объекта. Эти алгоритмы широко применяются в области компьютерных игр и проектирования, где необходимо обнаружить столкновения или пересечения объектов.

В зависимости от конкретных задач и требований различные комбинации

этих алгоритмов и структур данных могут быть использованы для хранения и управления ГГИ.

5.5.2 Способы хранения двумерной ГГИ

Для хранения двумерной ГГИ в базе данных геометрических объектов авторами предлагаются следующие основные способы хранения:

- Векторное представление примитивов. Этот подход будет наиболее эффективен, обеспечивая компактность и точность данных. при хранении двумерных геометро-графических данных, содержащих геометрические примитивы, такие как линии, окружности, дуги и т.д.
- Квадродеревья Эти структуры данных могут быть применены для пространственного индексирования объектов на чертеже по их координатам. Например, в CAD-приложениях для поиска объектов в определенных областях чертежа, выделения и редактирования групп объектов с определенными характеристиками.
- Алгоритмы сжатия данных (RLE, LZW) пригодятся для экономии места при хранении растровых изображений сканированных чертежей.
- Алгоритмы проверки пересечений потребуются для определения коллизий между элементами чертежа.

Подходящая комбинация этих и других способов хранения ГГИ позволит оптимизировать хранение и поиск двумерной геометрической информации.

5.5.3 Способы хранения трехмерной ГГИ

Для хранения трехмерной ГГИ авторами предлагается использование следующих основных способов хранения:

- Граничное представление. Этот подход будет эффективен для хранения всех поддерживаемых видов трехмерных моделей через их границы. Например, в области инженерного проектирования, где требуется представление сложных трехмерных объектов, таких как машины, здания или механизмы, В-гер представление позволит точно описать их геометрию и топологию.
- Графы сборки конструкции - CSG деревья. Данный подход удобен для

представления моделей как комбинации примитивов и выполнения булевых операций над ними. Например, в системах автоматизированного проектирования, где пользователь может комбинировать примитивы (например, объединять, пересекать или вычитать) для создания сложных объектов. По сути, этот вариант хранения ГГИ единственный для описанной в постановке задачи твердотельной трехмерной модели объекта.

- Октодеревья. Эти структуры данных оптимальны для поиска пространственных столкновений между объектами.
- Алгоритмы сжатия данных необходимы для уменьшения объема 3D моделей.

Таким образом, применение перечисленных алгоритмов позволит оптимизировать скорость и объем хранения как двумерной, так и трехмерной ГГИ в разрабатываемой базе данных. Конкретный выбор методов будет зависеть от приоритетов по быстродействию и компактности представления данных.

5.6 Выводы по пятой главе

1. Рассмотрена актуальная задача разработки системы управления данными для эффективного хранения и управления ГГИ, представленной в виде двумерных чертежей и трехмерных моделей.

2. Предложен макет системы управления данными, включающей в себя центральное хранилище графических данных и вспомогательные модули для загрузки, обработки, поиска и визуализации информации. Описаны подходы к структурированному хранению данных об объектах чертежей и компонентах трехмерных моделей с использованием реляционной СУБД. Для эффективного хранения двумерной геометрической информации было предложено использование таких структур данных, как векторное представление геометрических примитивов, квадродеревьев для пространственного индексирования, а также алгоритмов сжатия данных для экономии места при хранении растровых изображений. Комбинация этих методов позволит оптимизировать хранение и управление геометрической информацией, обеспечивая компактность и быстрый доступ к данным. Для трехмерной геометрической информации были рассмотрены такие способы хранения информации, как граничное представление для точного описания сложных 3D моделей, графы сборки конструкции для комбинации примитивов и выполнения булевых операций, октодеревья для быстрого поиска пространственных столкновений.

ЗАКЛЮЧЕНИЕ

1. Разработана структурная схема решения обратной задачи начертательной геометрии, отличительной особенностью которой является наличие этапа автоматизированного синтеза трехмерной модели с применением алгоритма предварительной обработки входной геометро-графической информации.

2. Составлен аналитический обзор существующих решений, выявлены их недостатки: алгоритмические методы либо существенно ограничивают класс моделируемых тел (например, многогранниками), либо имеют ограниченную автоматизацию (с необходимостью ручной корректировки модели в процессе формирования), либо очень сильно зависят от качества входных данных; варианты же реконструкции при помощи нейросетевых алгоритмов обладают ограниченной обобщаемостью (результат целиком зависит от обучающего датасета), проблемами с точностью реконструируемого объекта (часто генерируются приближенные результаты, а также подобные алгоритмы имеют склонность к формированию вариантов решения, не имеющих общего с исходными данными), огромнейшей вычислительной сложностью (чем больше будет входных параметров, тем больше вычислительной мощности будет требовать нейросеть).

3. Выделен класс моделируемых тел и, как следствие, класс данных, содержащих исходную информацию для алгоритма преобразования проекционных изображений в каркасную модель, сформулирована информационная модель задачи.

4. Рассмотрены различные варианты хранения ГГИ в памяти компьютера и способы передачи её между прикладными программами, на основании чего был выбран способ получения исходных данных напрямую из памяти САПР, описан алгоритм получения ГГИ через СОМ-объекты на примере САПР Компас.

5. Рассмотрены возможные ошибки, которые могут встретиться в

исходных данных. При этом не затрагиваются ошибки, допущенные проектировщиками. Принято решение на основании информации из ГОСТ 2.052-2021. “Единая система конструкторской документации (ЕСКД). Электронная модель изделия. Общие положения.”[12] о чтении размеров из длин элементов внутри САПР, а не из размерных чисел.

6. Разработаны алгоритмы предварительной обработки двумерных геометрических данных, ориентированных на устранение типовых погрешностей, возникающих при вводе чертежной информации. Отличительными особенностями являются:

а. прямая интеграция с системами автоматизированного проектирования (САПР) на этапе получения входных данных, минуя использование промежуточных форматов, что обеспечивает более высокое качество и полноту исходной геометрической информации для последующей обработки;

б. анализ проекционных изображений, результатом которого является восстановление топологических и метрических связей, убирающих противоречия между входным предметно-манипулятивным чертежом (использующимся на производстве) и выходным эпюром Монжа.

7. Предложенный алгоритм формирования каркасной модели на основе проекционных изображений позволяет корректно формировать каркасную модель на основе облака точек и проекционных изображений. Основные преимущества алгоритма включают:

а. Точность: учет связей между точками на всех проекционных изображениях обеспечивает высокую точность формирования ребер.

б. Устойчивость к ошибкам: использование матриц смежности позволяет минимизировать влияние ошибок в данных.

с. Гибкость: алгоритм может быть адаптирован для различных типов проекционных изображений и объектов.

8. Было проведено фиксирование и сравнение параметров математических моделей объектов, а также временных характеристик

восстановления объектов. Приведены данные, полученные при восстановлении трехмерной каркасной модели на основе проекционных изображений, считанных с реальных производственных технических чертежей. Сравнение полученных данных, представленных в таблицах, с временными характеристиками формирования 3D моделей деталей, полученных традиционным способом проектирования (в ходе интерактивного моделирования), позволяет сделать однозначный вывод о преимуществе и перспективности автоматического восстановления моделей объектов по архивным (бумажным и электронным) техническим чертежам. Проведена апробация алгоритма на реальных производственных моделях из ЦНИИ “Буревестник”

9. Доказана теоретически и подтверждена экспериментально вычислительная сложность реализованного алгоритма как $O(m^2 * k * n)$, где m – количество точек на проекционных изображениях; k – среднее количество ребер, исходящих из одной вершины; n – количество вершин в трехмерной точечной модели.

10. Рассмотрена актуальная задача разработки системы управления данными для эффективного хранения и управления ГГИ, представленной в виде двумерных чертежей и трехмерных моделей.

11. Предложен макет системы управления данными, включающей в себя центральное хранилище графических данных и вспомогательные модули для загрузки, обработки, поиска и визуализации информации. Описаны подходы к структурированному хранению данных об объектах чертежей и компонентах трехмерных моделей с использованием реляционной СУБД. Для эффективного хранения двумерной геометрической информации было предложено использование таких структур данных, как векторное представление геометрических примитивов, квадродеревьев для пространственного индексирования, а также алгоритмов сжатия данных для экономии места при хранении растровых изображений. Комбинация этих методов позволит оптимизировать хранение и управление геометрической информацией, обеспечивая компактность и быстрый доступ к данным. Для трехмерной

геометрической информации были рассмотрены такие способы хранения информации, как граничное представление для точного описания сложных 3D моделей, графы сборки конструкции для комбинации примитивов и выполнения булевых операций, октодеревья для быстрого поиска пространственных столкновений.

СПИСОК ЛИТЕРАТУРЫ

1. Автоматизированная технология ввода графических документов со слабо формализованным описанием объектов / Д. Ю. Васин, В. П. Громов, Е. А. Кустов [и др.] // Труды Международной конференции по компьютерной графике и зрению "Графикон". – 2018. – № 28. – С. 332-336.
2. Автоматизированный расчет параметрического числа плоского контура по его электронному чертежу / В. А. Тюрина, М. М. Смычек, С. А. Роменский, М. В. Лагунова // Труды Международной конференции по компьютерной графике и зрению "Графикон". – 2022. – № 32. – С. 963-970. – DOI 10.20948/graphicon-2022-963-970.
3. Базовые интерфейсы API системы КОМПАС [Электронный ресурс] URL:– <http://www.litmir.net/br/?b=115812&p=107>. (Дата обращения: 09.10.2015)
4. Барабанов, В.Ф. Разработка программного комплекса восстановления и визуализации 3D-моделей по набору приближенных ортогональных проекций / В.Ф. Барабанов, А.М. Нужный, С.Л. Подвальный, В.В. Сафронов // Научная визуализация, 2017, Т. 9, № 2, С. 82 - 93
5. Васин, Д.Ю. Автоматизированная технология преобразования чертежно-конструкторской документации на бумажных носителях в электронную 3D модель объекта / Д. Ю. Васин, В. П. Громов, С. А. Роменский, С. И. Ротков // Информационные технологии и нанотехнологии (ИТНТ-2020) : Сборник трудов по материалам VI Международной конференции и молодежной школы. В 4-х томах, Самара, 26–29 мая 2020 года / Под редакцией В.А. Фурсова. Т. 4. – Самара: Самарский национальный исследовательский университет имени академика С.П. Королева, 2020. – С. 997-1006.
6. Васин, Д.Ю. Исследование моделей описания, разработка алгоритмического, программного и технологического обеспечения обработки растровых изображений графических документов : диссертация кандидата технических наук : 05.01.01. - Нижний Новгород, 2006. - 199 с. : ил.

7. Восстановление каркасной модели 3D-объекта по бумажному архивному чертежу / С. А. Роменский, С. И. Ротков, Ю. П. Бурцев [и др.] // Вестник Южно-Уральского государственного университета. Серия: Строительство и архитектура. – 2022. – Т. 22, № 4. – С. 66-75. – DOI 10.14529/build220407.
8. Восстановление структурной информации об антропогенных объектах из одиночных аэрокосмических изображений / Н. В. Антипова, О. Г. Гвоздев, В. А. Козуб [и др.] // Известия Академии наук СССР. Техническая кибернетика. – 2023. – Т. 2, № 3. – С. 90-105. – DOI 10.31857/S0002338823030010.
9. Вяткин, С. И. Метод реконструкции 3D-объектов по данным одного ракурса изображения / С. И. Вяткин, Б. С. Долговесов // Автометрия. – 2023. – Т. 59, № 5. – С. 47-55. – DOI 10.15372/AUT20230505.
10. Ганин, Н.Б. Трехмерное проектирование в Компас-3D, Москва, 2012 - 784 с.
11. Глушков, В.М. Основы безбумажной информатики / В.М. Глушков. – М.: Наука, Главная редакция физ.-мат. литературы, 1982. – 551 с.
12. ГОСТ 2.052-2021 «Единая система конструкторской документации. Электронная модель изделия. Общие положения». URL: <https://docs.cntd.ru/document/1200179217> (дата обращения 12.05.2025).
13. ГОСТ 2.305-2008 «Единая система конструкторской документации (ЕСКД). Изображения - виды, разрезы, сечения (с Поправкой)». URL: <https://docs.cntd.ru/document/1200069435> (дата обращения 12.05.2025).
14. ГОСТ 2.307-2011 Единая система конструкторской документации (ЕСКД). Нанесение размеров и предельных отклонений (с Поправками). URL: <http://docs.cntd.ru/document/1200086238> (дата обращения 12.05.2025).
15. Журавлев, А.С. AutoCAD для конструкторов. Стандарты ЕСКД в AutoCAD 2009/2010/2011. Санкт-Петербург, 2011 - С. 24-27

16. Задача преобразования каркасной модели 3D объекта, восстановленной по его техническому чертежу / Т. В. Мошкова, С. И. Ротков, М. М. Смычек, В. А. Тюрина // Научная визуализация. – 2018. – Т. 10, № 5. – С. 13-31. – DOI 10.26583/sv.10.5.02.
17. Исследование возможностей расширения видов входной информации в задаче синтеза 3D-модели / М. В. Лагунова, Т. В. Мошкова, С. А. Роменский [и др.] // Виртуальное моделирование, прототипирование и промышленный дизайн : Материалы IV Международной научно-практической конференции, Тамбов, 15–17 ноября 2017 года. Выпуск 4, Том II. – Тамбов: Тамбовский государственный технический университет, 2017. – С. 219-226.
18. Исследование временных характеристик алгоритма восстановления каркасной модели по проекционным изображениям / Т. В. Мошкова, С. А. Роменский, С. И. Ротков, В. А. Тюрина // Труды Международной конференции по компьютерной графике и зрению "Графикон". – 2018. – № 28. – С. 369-371.
19. КИТЕЖ – система геометрии и графики пространственных объектов / Е.В. Аристова, А.А. Зудин, С.В. Митин [и др.]// Сб. тезисов 5-ой Всесоюзной конференции по проблемам машинной графики. - Новосибирск, 1989.
20. Котов, И.И. Алгоритмы машинной графики / И.И. Котов, В.С.Полозов, Л.В. Широкова. – М.: Машиностроение, 1977. – 232 с.
21. Котов, И.И. Начертательная геометрия. На принципах программированного обучения. Учебник. -М.: Высшая школа, 1970. - 384 с.
22. Мультикодовое представление учебного материала как способ активизации учебно-познавательной деятельности / Е. В. Конопацкий, М. В. Лагунова, А. В. Назаровская [и др.] // Вестник Нижегородского университета им. Н.И. Лобачевского. Серия: Социальные науки. – 2024. – № 3(75). – С. 237-246. – DOI 10.52452/18115942_2024_3_237.
23. Норенков, И. П. Информационная поддержка наукоемких изделий (CALS-технологии) / И.П. Норенков, П.К. Кузьмик. — М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. – 217 с.

24. Норсеев, С.А. Разработка приложений под КОМПАС в Delphi - Санкт-Петербург:Изд-во Питер, 2016. - 346 с.
25. Организация переноса геометро-графической информации в базу данных для прикладной программы / Н. Л. Макаров, С. А. Роменский, С. И. Ротков [и др.] // Системы проектирования, технологической подготовки производства и управления этапами жизненного цикла промышленного продукта (CAD/CAM/PDM - 2016) : труды XVI-ой международной молодёжной конференции, Москва, 17–19 октября 2016 года. – Москва: ООО "Аналитик", 2016. – С. 124-128.
26. Орлов П.И. Основы конструирования. Справочно-методическое пособие в 3-х книгах.Кн. 1.Изд. 2-е, перераб. и доп. М., "Машиностроение", 1977. - 623 стр. с ил.
27. Осипов, В.А., Теоретические основы каркасно-кинематического метода направляющей линии / В.А. Осипов, Л.И. Осипова // Изв. ВУЗов. Серия “Авиационная техника”. 1980, № 4. - С. 48-52.
28. Осипов, В.А. Теоретические основы формирования системы машинной геометрии и графики : учеб. пособие. – М. : МАИ, 1983. – 34 с., ил.
29. Полещук Н.Н., Autocad 2009: наиболее полное руководство, Санкт-Петербург, 2009 - 1184с.
30. Полозов, В.С. Автоматизация решения графических инженерных задач с помощью цифровых электронных вычислительных машин : диссертация ... кандидата технических наук : 05.00.00. - Москва, 1966. - 158 с. : ил.
31. Полозов, В.С. Моделирование и синтез операторов геометрического расчета и машинной графики в системах автоматизированного проектирования и автоматизации технологической подготовки производства : диссертация ... доктора технических наук : 05.01.01. - Горький, 1983. - 382 с. : ил.

32. Проблема преобразования бумажных архивов чертежно-конструкторской документации в электронную модель изделия и связанные с ней геометро-графические задачи / С. И. Ротков, Е. В. Попов, Т. В. Мошкова [и др.] // GraphiCon 2017 : Труды 27-й Международной конференции по компьютерной графике и машинному зрению, Пермь, 24–28 сентября 2017 года / Пермский государственный национальный исследовательский университет. – Пермь: Издательский центр Пермского государственного национального исследовательского университета, 2017. – С. 20.

33. Распознавание компонент связности в задаче восстановления трехмерной модели по проекционным изображениям / Н. Д. Жилина, Т. В. Мошкова, С. А. Роменский [и др.] // Труды Международной конференции по компьютерной графике и зрению "Графикон". – 2019. – № 29. – С. 185-187. – DOI 10.30987/graphicon-2019-1-185-187.

34. Разработка тренажера для изучения поверхностей в составе электронного обучающего комплекса / Т. В. Мошкова, С. А. Роменский, О. А. Румянцева, В. А. Тюрина // Виртуальное моделирование, прототипирование и промышленный дизайн : Материалы II Международной научно-практической конференции, Тамбов, 17–19 ноября 2015 года. Том 2. – Тамбов: Тамбовский государственный технический университет, 2016. – С. 147-149.

35. Роменский, С. А. Передача геометро-графической информации из САПР "Компас-График" в прикладную программу / С. А. Роменский // Системы проектирования, технологической подготовки производства и управления этапами жизненного цикла промышленного продукта (CAD/CAM/PDM - 2015) : Труды международной конференции, Москва, 26–28 октября 2015 года / Под ред. А.В. Толока. – Москва: ООО «Аналитик», 2015. – С. 97-100

36. Роменский, С.А. Алгоритмическая и программная реализация электронного конструктора графа сборки 3D объекта / Т.В. Мошкова, С.А. Роменский, С.И. Ротков, В.А. Тюрина// Кибернетика и программирование. — 2017. - № 5. - С.1-13. DOI: 10.25136/2306-4196.2017.5.24342.

37. Роменский, С.А. Выявление ложных элементов каркасной модели 3D объекта при помощи разрезов / С.А. Роменский, М.М. Смычѣк // XXI Нижегородская сессия молодых ученых. Технические науки: материалы докладов / Отв. за вып. Зверева И. А. – Княгинино : НГИЭУ, 2016. – С. 81-82.

38. Роменский, С.А. Динамическое моделирование и визуализация поверхностей для тренажѣра / С.А. Роменский, О.А. Румянцева, М.М. Смычек // XXI Нижегородская сессия молодых ученых. Естественные, математические науки: материалы докладов / Отв. за вып. Зверева И. А. – Княгинино : НГИЭУ, 2016. – С. 94-97.

39. Роменский, С.А. О проблеме передачи данных между пользовательскими программами в информационных технологиях / С. А. Роменский // Сборник трудов аспирантов, магистрантов и соискателей. Технические науки. Науки о Земле. Экология : Сборник статей. – Нижний Новгород : Нижегородский государственный архитектурно-строительный университет, 2015. – С. 119-125.

40. Роменский, С.А. Обеспечение информационной совместимости между различными системами автоматизированного проектирования / С. А. Роменский // Сборник трудов конференции регионального значения XX Нижегородская сессия молодых ученых. Естественные, математические науки.: сб.тр. рег.конф., 19-22 мая 2015г., - г. Арзамас, 2015г.

41. Роменский, С.А. Разработка базы данных для хранения и обмена геометрической информацией / С. А. Роменский, А. В. Назаровская, С. И. Ротков // Вестник компьютерных и информационных технологий. – 2023. – Т. 20, № 12(234). – С. 11-18. – DOI 10.14489/vkit.2023.12.pp.011-018.

42. Роменский, С.А. Формирование трехмерной каркасной модели в проблеме преобразования чертежноконструкторской документации на бумажном носителе в электронную модель объекта / С. А. Роменский, С. И. Ротков // Труды Международной конференции по компьютерной графике и зрению "Графикон". – 2020. – № 30. – С. 67-82. – DOI 10.51130/graphicon-2020-1-67-82.

43. Ротков, С. И. Средства геометрического моделирования и компьютерной графики пространственных объектов для CALS-технологий : диссертация доктора технических наук : 05.01.01. – Нижний Новгород, 1999. – 287 с.

44. Ротков, С.И. Синтез моделей пространственных объектов по многовидовому чертежу. / С.И. Ротков // В сб. материалов 4-й Международной конференции по компьютерной графике и визуализации ГРАФИКОН-94.-Нижний Новгород, 1994, с.37-39

45. Синтез 3D модели объекта по изображениям на поле чертежа / С. А. Роменский, С. И. Ротков, М. М. Смычек, В. А. Тюрина // Графикон'2016 : Труды 26-й Международной научной конференции, Нижний Новгород, 19–23 сентября 2016 года. – Нижний Новгород: Автономная некоммерческая организация "Институт физико-технической информатики", 2016. – С. 272-275.

46. Синтез параметрической конструктивной модели 3D-объекта / Н. Д. Жилина, М. В. Лагунова, Т. В. Мошкова [и др.] // GraphiCon 2017 : Труды 27-й Международной конференции по компьютерной графике и машинному зрению, Пермь, 24–28 сентября 2017 года / Пермский государственный национальный исследовательский университет. – Пермь: Издательский центр Пермского государственного национального исследовательского университета, 2017. – С. 350-354.

47. Система геометрического моделирования пространственных объектов КИТЕЖ / Е.В. Аристова, А.А. Зудин, С.Е. Лабутин [и др.]// Сб. трудов 4-ой Междунар. конф. по комп. графике и визуализации ГРАФИКОН-94. - Н. Новгород. - 1994. - С. 72 – 74.

48. Создание 3D-модели изделия и её многовидового чертежа [Электронный ресурс]: учеб.- метод. пос. / Т. В. Мошкова, С. А. Роменский, М. М. Смычек, В. А. Тюрина; Нижегор. гос. архитектур. - строит. ун - т – Н. Новгород: ННГАСУ, 2017. – 29 с; ил. 1 электрон. опт. диск (CD-RW)

49. Тани, Х.И. Алгоритм построения пространственного описания тела, заданного проекциями / Х.И. Тани // Известия АН СССР. Техническая кибернетика. – 1966. – № 6.
50. Трельсен Э. Модель СОМ и применение ATL 3.0. СПб.: БХВ-Петербург, 2001. - 928с.
51. Тюрина, В. А. Разработка методов преобразований каркасной модели в задаче синтеза образа 3D-объекта по его проекциям : специальность 05.01.01 "Инженерная геометрия и компьютерная графика" : диссертация на соискание ученой степени кандидата технических наук / Тюрина Валерия Александровна. – Нижний Новгород, 2003. – 170 с.
52. Управление параметрами поверхностей в электронном тренажёре / М. В. Лагунова, Т. В. Мошкова, С. А. Роменский [и др.] // Информатизация инженерного образования : Труды Международной научно-практической конференции - ИНФОРИНО-2016, Москва, 12–13 апреля 2016 года. – Москва: Издательский дом МЭИ, 2016. – С. 163-166
53. Фаулер, М. NoSQL: новая методология разработки нереляционных баз данных = NoSQL Distilled. /М. Фаулер, П. Дж. Садаладж — М.: «Вильямс», 2013. — 192 с. — ISBN 978-5-8459-1829-1.
54. Шекхар, Ш. Основы пространственных баз данных. / Ш. Шекхар, Ч. Санжей— Москва: Кудиц-образ, 2004. — 336 с. — 3000 экз. — ISBN 0-13-017480-7, 5-93378-091-X.
55. Широкий, Г. Б. Средства архивизации геометро-графической информации в процессе автоматизированного проектирования на крупном предприятии : диссертация кандидата технических наук : 05.01.01. - Нижний Новгород, 1998. - 126 с.
56. Щеколдин, В.А. Построение на ЭВМ пространственного описания многогранного тела по описаниям его проекций / В.А. Щеколдин // Вычислительная техника в машиностроении: Сб. ст. – Минск. – ИТК АН БССР, 1968.

57. Электронный конструктор графа сборки 3D объекта / М. В. Лагунова, Т. В. Мошкова, С. А. Роменский [и др.] // Ситуационные центры и информационно-аналитические системы класса 4i для задач мониторинга и безопасности (SCVRT2015-16) : Труды Международной научной конференции: в 2-х томах, ЦарьГрад, Московская область, Россия, 21–24 ноября 2016 года. Том 2. – ЦарьГрад, Московская область, Россия: Автономная некоммерческая организация "Институт физико-технической информатики", 2016
58. Электронный учебный иллюстратор по дисциплине "Компьютерная графика" / М. В. Лагунова, А. В. Матренин, Т. В. Мошкова [и др.] // Виртуальное моделирование, прототипирование и промышленный дизайн : Материалы III Международной научно-практической конференции, Тамбов, 15–17 ноября 2016 года. Том 2. – Тамбов: Тамбовский государственный технический университет, 2016. – С. 296-300.
59. Aitchison A Pro Spatial with SQL Server 2012 / A Aitchison. – Berkeley, CA : Apress, 2012. – 588 с. – ISBN 978-1430234913
60. Alwan S. Unsupervised and hybrid vectorization techniques for 3D reconstruction of engineering drawings (Ph.D. thesis), Ecole nationale supérieure Mines-Télécom Atlantique; 2021.
61. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In Advances in Neural Information Processing Systems, Vol. 30. DOI:10.48550/arXiv.1706.03762
62. Çiçek A, Gülesin M. Reconstruction of 3D models from 2D orthographic views using solid extrusion and revolution. Journal of Materials Processing Technology, 2004;152(3):291–8. DOI:10.1016/j.jmatprotec.2004.04.368
63. DWG [Электронный ресурс]: Wikipedia. URL: <https://ru.wikipedia.org/wiki/DWG> (дата обращения: 11.03.2015).
64. DXF [Электронный ресурс]: Wikipedia. URL: <https://ru.wikipedia.org/wiki/DXF> (дата обращения: 11.03.2015).
65. Eisenberg, J.D. SVG Essentials, 2002 - 360с.

66. Furferi R, Governi L, Palai M, Volpe Y. From 2D orthographic views to 3D pseudo-wireframe: An automatic procedure. *International Journal of Computer Applications* 2010;975:8887. DOI:10.5120/918-1296
67. Gong J-H, Zhang H, Zhang Y-W, Sun J-G. Converting hybrid wire-frames to B-rep models. In: *Proceedings of the 2007 ACM symposium on solid and physical modeling*. 2007, p. 283–9.
68. Gong J-H, Zhang G-F, Zhang H, Sun J-G. Reconstruction of 3D curvilinear wire-frame from three orthographic views. *Comput Graph* 2006;30(2):213–24.
69. Gong J-H, Zhang H, Zhang G-F, Sun J-G. Solid reconstruction using recognition of quadric surfaces from orthographic views. *Comput Aided Des* 2006;38(8):821–35.
70. Gorgani, Hamid Haghshenas et al. 3D Model Reconstruction from Two Orthographic Views using Fuzzy Surface Analysis. *European Journal of Sustainable Development Research*, vol. 3, no. 2, 2019, em0081. DOI:10.29333/ejosdr/5726
71. Harish AB, Prasad AR. Automated 3D solid reconstruction from 2D CAD using opencv. 2021, arXiv Preprint: 2101.04248. DOI:10.48550/arXiv.2101.04248
72. Idesawa, M. A System to Generate a Solid Figure from a Three View / M.A. Idesawa, *Bull. JSME* 16. – February, 1973. – P. 216 – 225.
73. ISO 10303-1:1994 Industrial automation systems and integration Product data representation and exchange - Overview and Fundamental Principles, International Standard, ISO TC184/SC4, 1994
74. Jayaraman PK, Lambourne JG, Desai N, Willis KD, Sanghi A, Morris NJ. SolidGen: An autoregressive model for direct B-rep synthesis. 2022, arXiv Preprint: 2203.13944. DOI:10.48550/arXiv.2203.13944
75. Kazaryan, M. Reconstruction of 3-D models of infrastructure objects from satellite images based on typed elements / M. Kazaryan, A. Richter, A. Gvozdev [et al.] // *Proceedings of SPIE - SPIE Remote Sensing*, Berlin, 05–08 сентября 2022 года. Vol. 12269. – SPIE : SPIE, 2022. – P. 122690J. – DOI 10.1117/12.2641134.

76. Kothuri, R. Pro Oracle Spatial for Oracle Database 11g / R Kothuri, A Godfrind, E Beinat. – <https://doi.org/10.1007/978-1-4302-0446-6>. – Berkeley, CA : Apress, 2014. – 787 c. – ISBN 978-1-4302-1187-7
77. Lambourne JG, Willis KD, Jayaraman PK, Sanghi A, Meltzer P, Shayani H. Brepnet: A topological message passing system for solid models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021, p. 12773–82. DOI:10.48550/arXiv.2104.00706
78. Lee H, Han S. Reconstruction of 3D interacting solids of revolution from 2D orthographic views. Computer-Aided Design 2005;37(13):1388–98. DOI:10.1016/j.cad.2005.01.007
79. Li C, Pan H, Bousseau A, Mitra NJ. Free2CAD: parsing freehand drawings into CAD commands. ACM Transactions on Graphics 2022;41(4):1–16. DOI:10.1145/3528223.3530133
80. Li C, Pan H, Bousseau A, Mitra NJ. Sketch2cad: Sequential cad modeling by sketching in context. ACM Transactions on Graphics 2020;39(6):1–14. DOI:10.1145/3414685.3417807
81. Li, W. 3D Virtual Modeling Realizations of Building Construction Scenes via Deep Learning Technique / W. Li // Computational Intelligence and Neuroscience. – 2022. – Vol. 2022. – P. 6286420. – DOI 10.1155/2022/6286420.
82. Lu Z, Guo J, Xiao J, Wang Y, Zhang X, Yan D-M. Extracting cycle-aware feature curve networks from 3D models. Computer Aided Design 2021;131:102949. DOI:10.1016/j.cad.2020.102949
83. Markowsky G, Wesley MA. Fleshing out wire frames. IBM Journal of Research and Development, 1980;24(5):582–97.
84. Obe, R. PostGIS in Action, Third Edition / R Obe, L Hsu. – New York City, United States : Manning, 2021. – 664 c. – ISBN 978-1617296697
85. Robinson, I. and Webber, J. and Eifrem, E. Graph Databases. — O'Reilly Media, 2013. — 178 p. — ISBN 978-1449356262.
86. Russell, J., Cohn, R., IGES, 2013 – 101c.

87. Shi-Xia Liu, Shi-Min Hu, Yu-Jian Chen, Jia-Guang Sun, Reconstruction of curved solids from engineering drawings, *Computer-Aided Design*, Volume 33, Issue 14, 2001, Pages 1059-1072, ISSN 0010-4485, DOI:10.1016/S0010-4485(00)00143-3.
88. SVG [Электронный ресурс]: Wikipedia. URL: <https://ru.wikipedia.org/wiki/SVG> (дата обращения: 11.03.2015).
89. Tanaka M, Kaneeda T. Feature extraction from sketches of objects. *Comput-Aided Design and Applications* 2015;12(3):300–9. DOI: 10.1080/16864360.2014.981459
90. Theia: the revolutionary innovation that will accelerate digitization, relocation and decarbonization of industry. Press release 19.03.2024. Систем.требования: Adobe Acrobat Reader. URL:<https://spare-parts-3d.com/wp-content/uploads/2024/03/ENG-Final-Press-Release-1.pdf> (дата обращения 30.09.2024)
91. Varley PA, Company PP. A new algorithm for finding faces in wireframes. *Computer Aided Design* 2010;42(4):279–309. DOI:10.1016/j.cad.2009.11.008
92. The Automated Technology for Converting Drawing and Design Paper-Based Documentation into Electronic 3D Model of an Object / D. Vasin, V. Gromov, S. Rotkov, S. Romensky // *Proceedings of ITNT 2020 - 6th IEEE International Conference on Information Technology and Nanotechnology : 6, Virtual, Samara, 26–29 мая 2020 года. – Virtual, Samara, 2020. – P. 9253284. – DOI 10.1109/ITNT49337.2020.9253284*
93. Wang K, Zheng J, Zhou Z. Neural face identification in a 2D wireframe projection of a manifold object. In: *Proceedings of the IEEE/CVF conf. on computer vision and pattern recognition*. 2022, p. 1622–31.
94. Weiss-Cohen M. 3D reconstruction of solid models from engineering orthographic views using variational geometry and composite graphs. *Comput-Aided Design and Applications* 2007;4(1–4):159–67. DOI: 10.1080/16864360.2007.10738536
95. Wesley MA, Markowsky G. Fleshing out projections. *IBM Journal of Research and Development*, 1981;25(6):934–54.
96. Willis KD, Jayaraman PK, Lambourne JG, Chu H, Pu Y. Engineering sketch generation for computer-aided design. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, p. 2105–14.

97. Willis KD, Pu Y, Luo J, Chu H, Du T, Lambourne JG, Solar-Lezama A, Matusik W. Fusion 360 gallery: A dataset and environment for programmatic cad construction from human design sequences. *ACM Transactions on Graphics* 2021;40(4):1–24. DOI:10.48550/arXiv.2010.02392
98. Wu R, Xiao C, Zheng C. Deepcad: A deep generative network for computer-aided design models. In: *Proceedings of the IEEE/CVF int. conf. on computer vision. ICCV*, 2021, p. 6772–82. DOI:10.48550/arXiv.2105.09492
99. Xu, B. A 3D reconstruction method for buildings based on monocular vision / B. Xu, Ch. Liu // *Computer-Aided Civil and Infrastructure Engineering*. – 2022. – Vol. 37, No. 3. – P. 354-369. – DOI 10.1111/mice.12715
100. Yan Q-W, Chen CP, Tang Z. Efficient algorithm for the reconstruction of 3D objects from orthographic projections. *Computer Aided Design*, 1994;26(9):699–717.
101. Yifan Qie, Nabil Anwer. Invariance Class based Surface Reconstruction using CNN Transfer Learning | 1204-1220 | doi: 10.14733/cadaps.2023.1204-1220
102. Zakharov A, Zhiznyakov A. Synthesis of three-dimensional models from drawings based on spectral graph theory. In: *Applied mechanics and materials*, Vol. 756. Trans Tech Publ; 2015, p. 598–603. DOI:10.4028/www.scientific.net/AMM.756.598
103. Zhang A, Xue Y, Sun X, Hu Y, Luo Y, Wang Y, Zhong S, Wang J, Tang J, Cai G. Reconstruction of 3D curvilinear wireframe model from 2D orthographic views. In: *Int. conf. on computational science*. Springer; 2004, p. 404–11.

ПРИЛОЖЕНИЕ А. АКТЫ О ВНЕДРЕНИИ.

УТВЕРЖДАЮ



Директор ООО «Лаборатория КОМПАС»

Бахин Е.В.

«29» декабря 2020 г.

АКТ

О внедрении результатов работ.

Составлен комиссией в составе Бахина Е.В., Устиновой Л.Ю. в том, что результаты диссертационной работы Роменского Сергея Александровича, представленной на соискание ученой степени кандидата технических наук по специальности 05.01.01 – Инженерная геометрия и компьютерная графика, использованы при разработке новых алгоритмов и программ формирования 3D геометрии электронной модели изделия в процессе создания новых функциональных операторов системы КОМПАС.

Председатель



Бахин Е.В.

Члены комиссии



Устинова Л.Ю.





АКЦИОНЕРНОЕ ОБЩЕСТВО
«ЦЕНТРАЛЬНЫЙ НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ
ИНСТИТУТ «БУРЕВЕСТНИК»
(АО «ЦНИИ «БУРЕВЕСТНИК»)
Сормовское шоссе, д. 1а
г. Нижний Новгород, Россия, 603950
Тел.: (831) 241-12-42
e-mail: mail@burevestnik-nnov.ru
ОГРН 1085259003664
ИНН/КПП 5259075468/525901001

УТВЕРЖДАЮ

Генеральный директор
АО «ЦНИИ «Буревестник»

П.А. Ковалев

2025 г.



№ _____
На № _____ от _____

АКТ

внедрения результатов работ

Составлен комиссией в составе: председателя – директора инженерного центра Савиных Александра Михайловича и членов комиссии – начальника отдела, д.т.н., доцента Чернова Вадима Валентиновича, ведущего инженера-конструктора Шашкова Романа Олеговича в том, что результаты диссертационной работы Роменского Сергея Александровича, представленной на соискание ученой степени кандидата технических наук по специальности 2.5.1 – «Инженерная геометрия и компьютерная графика. Цифровая поддержка жизненного цикла изделия», были использованы при формировании 3D-геометрии электронных моделей изделий, разрабатываемых в АО «ЦНИИ «Буревестник».

Настоящий акт не является основанием для взаимных финансовых расчетов.

Председатель

А.М. Савиных

Члены комиссии

В.В. Чернов

Р.О. Шашков

ПРИЛОЖЕНИЕ Б. СВИДЕТЕЛЬСТВА О РЕГИСТРАЦИИ ПРОГРАММ ДЛЯ ЭВМ.

РОССИЙСКАЯ ФЕДЕРАЦИЯ



RU

2017662587

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ГОСУДАРСТВЕННАЯ РЕГИСТРАЦИЯ ПРОГРАММЫ ДЛЯ ЭВМ

Номер регистрации (свидетельства): 2017662587	Авторы:
Дата регистрации: 13.11.2017	Мошкова Татьяна Владимировна (RU),
Номер и дата поступления заявки: 2017618578 24.08.2017	Роменский Сергей Александрович (RU),
Дата публикации: 13.11.2017	Ротков Сергей Игоревич (RU),
	Тюрина Валерия Александровна (RU)
	Правообладатель:
	Федеральное государственное бюджетное образовательное учреждение высшего образования «Нижегородский государственный архитектурно-строительный университет» (ННГАСУ) (RU)

Название программы для ЭВМ:

Тренажер для создания графа сборки конструкции «Bit-GRASS v. 1.0»

Реферат:

Разработанная программа позволяет пользователю понять принцип конструирования с помощью теоретико-множественных операций сначала простейших 3D-моделей, а затем и более сложных. Для этого предусмотрены три уровня работы в программе: базовый (с входной информацией в виде аксонометрического чертежа изделия), основной (с входной информацией в виде технического многовидового чертежа), творческий (с внешней входной информацией). В процессе работы с тренажёром пользователь: закрепляет навыки чтения чертежей разных видов; получает навык сборки составных фигур, который реализуется построением графа сборки конструкции (бинарного дерева). Работа выполнена по грантам РФФИ №15-07-05110 и №17-07-00543.

Тип реализующей ЭВМ: IBM PC - совмест. ПК

Язык программирования: C++

Вид и версия операционной системы: Windows

Объем программы для ЭВМ: 800 Кб

РОССИЙСКАЯ ФЕДЕРАЦИЯ



RU

2020612320

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ГОСУДАРСТВЕННАЯ РЕГИСТРАЦИЯ ПРОГРАММЫ ДЛЯ ЭВМ

Номер регистрации (свидетельства): 2020612320	Авторы:
Дата регистрации: 19.02.2020	Роменский Сергей Александрович (RU), Мошкова Татьяна Владимировна (RU), Ротков Сергей Игоревич (RU), Лысенко Илья Александрович (RU)
Номер и дата поступления заявки: 2020611373 10.02.2020	Правообладатель:
Дата публикации и номер бюллетеня: 19.02.2020 Бюл. № 2	Федеральное государственное бюджетное образовательное учреждение высшего образования «Нижегородский государственный архитектурно- строительный университет» (ННГАСУ) (RU)
Контактные реквизиты: нет	

Название программы для ЭВМ:
Конструктор интерактивных технических документов с динамической визуализацией трехмерных геометрических моделей «CoGAN v.1.0»

Реферат:

Разработанный программный комплекс позволяет формировать в интерактивном режиме технический документ (инструкция, препринт, учебное пособие и пр.), в котором, кроме текстовой части и статичных рисунков, могут быть средства визуализации трехмерных моделей проектируемых или изготавливаемых объектов. Визуализация модели осуществляется динамически, т.е. объект может быть сориентирован в пространстве произвольным образом. Также эта возможность позволяет подключить видео файл, отражающий динамику поведения объекта. Эти существенно отличает данную разработку от ранее существующих. Работа выполнена по грантам РФФИ № 19-07-01024.

Язык программирования: html, javascript

Объем программы для ЭВМ: 2130 Кб

РОССИЙСКАЯ ФЕДЕРАЦИЯ



RU

2021663454

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ГОСУДАРСТВЕННАЯ РЕГИСТРАЦИЯ ПРОГРАММЫ ДЛЯ ЭВМ

Номер регистрации (свидетельства): 2021663454	Авторы:
Дата регистрации: 17.08.2021	Роменский Сергей Александрович (RU), Назаровская Агата Владимировна (RU), Ротков Сергей Игоревич (RU), Смышлёк Мария Михайловна (RU), Тюрина Валерия Александровна (RU)
Номер и дата поступления заявки: 2021662563 09.08.2021	Правообладатель:
Дата публикации и номер бюллетеня: 17.08.2021 Бюл. № 8	Федеральное государственное бюджетное образовательное учреждение высшего образования «Нижегородский государственный архитектурно- строительный университет» (ННГАСУ) (RU)
Контактные реквизиты: нет	

Название программы для ЭВМ:
Программный комплекс по формированию каркасной модели на основе проекционных изображений «WireProj v.1.0»

Реферат:

Программный комплекс позволяет сформировать трехмерную каркасную модель по имеющимся проекционным изображениям. Сформированная трехмерная модель динамически визуализируется во встроенном модуле. Программный комплекс поддерживает современные операционные системы семейства Windows. Также программный комплекс является частью информационной технологии преобразования чертежно-конструкторской документации на бумажном носителе в электронную модель изделия. Программный комплекс выполнен по гранту РФФИ № 19-07-01024. Тип ЭВМ: IBM PC - совмест. ПК на базе процессоров Intel/AMD; ОС: Windows NT 4.0 и выше.

Язык программирования: C++**Объем программы для ЭВМ: 582 КБ**