

Формирование трехмерной каркасной модели в проблеме преобразования чертежно- конструкторской документации на бумажном носителе в электронную модель объекта *

С.А. Роменский¹, С.И. Ротков²

¹ Нижегородский государственный архитектурно-строительный университет
romensky@nngasu.ru

² Нижегородский государственный архитектурно-строительный университет
rotkov@nngasu.ru

Аннотация. Данная работа посвящена различным этапам формирования трехмерной каркасной модели при решении проблемы преобразования бумажных чертежей в чертежно-конструкторскую документацию. Подробно рассматриваются этапы получения геометро-графической информации прикладной программой и формирования трехмерной каркасной модели. Так же приводится исследование временных характеристик разработанного алгоритма.

Ключевые слова: каркасная модель, геометро-графическая информация, проекционные изображения.

1 Введение

Любой объект производства в течение долгого времени должен иметь информационную поддержку жизненного цикла изделия (ЖЦИ). Данная поддержка осуществляется при помощи совокупности различных систем, объединенных единой базой данных. CALS (англ. 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. Возможность автоматически восстанавливать

* Работа выполнена при поддержке гранта РФФИ №19-07-01024.

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.

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.

Первые исследования в области восстановления 3D моделей по проекционным изображениям были проведены Х.И. Тани (1964) [1], В.А. Щеколдиным (1966) [2], и В.С. Полозовым в его докторской диссертации (1983) [3]. 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» [4]. Ими была поставлена проблема автоматизации процесса преобразования двумерной геометрической модели в трехмерную конструктивную модель.

Балабан О.М. [5] в своей кандидатской диссертации описал метод синтеза из двумерных проекционных изображений каркасных и граничных трехмерных моделей в рамках двух классов трехмерных объектов: тела вращения и многогранники.

Тюрина В.А. [6] предложила методы и алгоритмы автоматизации процесса преобразования трехмерной каркасной модели в граничную для объектов класса «деталь», причем проблема удаления ложных ребер каркасной модели решалась путем применения теории графов и фундаментальных циклов.

Существует система автоматизированного проектирования (САПР) «Китеж», созданная в 1985 году НИИ механики ННГУ им. Лобачевского группой разработчиков (Аристов Е.В., Зудин А.А., Лабутин С.Е., Митин С.В., Шубин В.П., Дергунов В.И., Лапшин М.Л., Полозов В.С., Ротков С.И.) [7], включающая в себя реализованный алгоритм преобразования технического чертежа в трехмерную каркасную модель, но данная САПР имеет существенные ограничения по количеству вершин в каркасной модели (256), осуществление строгого контроля входных данных, необходимость удаления всех элементов оформления чертежа перед восстановлением, отсутствие поддержки современных операционных систем.

2 Проблема передачи геометро-графической информации в прикладную программу

Для решения проблемы передачи данных между различными системами хранения и обработки геометро-графической информации (ГГИ) в рамках поддержки жизненного цикла изделия существуют международные стандарты передачи данных IGES и STEP (STandard for the Exchange of Product model data – стандарт для обмена данными о модели изделия), но для оперативной работы с геометро-графической информацией данный способ не эффективен из-за больших объемов побочных данных и необходимости двойной перекодировки, ведущей, в свою очередь, к возникновению ошибок. Для чтения же внутреннего формата хранения данных любой системы автоматизированного проектирования необходимо либо изучение открытой спецификации (если таковая предоставляется фирмой-разработчиком данной САПР) либо незаконное вскрытие внутренней структуры хранения данных при закрытой для сторонних разработчиков спецификации.

Электронная модель любого изделия, используемая в CALS-технологиях, на 80-90% состоит из ГГИ, создающейся конструктором или технологом при помощи соответствующих программных средств, входящих в состав систем автоматизированного проектирования и технологической подготовки производства. Каждая из таких систем имеет свой, только ей присущий, экраный интерфейс, свою внутреннюю структуру данных и свои алгоритмы решения элементарных геометро-графических задач. Это как раз и приводит к проблемам передачи данных между составными частями систем поддержки жизненного цикла изделия, 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 (за редкими исключениями).

Для обеспечения информационной совместимости между различными системами поддержки жизненного цикла изделия существуют два основных способа передачи данных:

- через программы – конвертеры данных. Данное решение обеспечивает прямую передачу ГГИ из одной системы в другую, что приводит к необходимости создавать дополнительную программу связи (рис. 1);

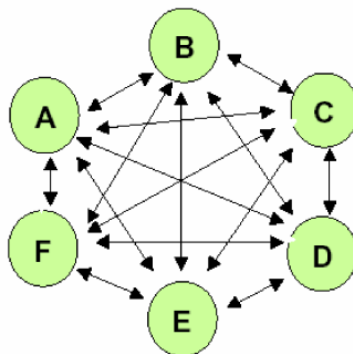


Рис. 13. Диаграмма связи форматов хранения различных систем

- через международные стандарты передачи ГГИ типа IGES или STEP (рис. 2). Недостатком данного способа является право фирмы-разработчика систем поддержки жизненного цикла изделия реализовывать лишь подмножество нейтрального формата, что неминуемо ведет к разночтениям при передаче и анализе данных в системе-приемнике.

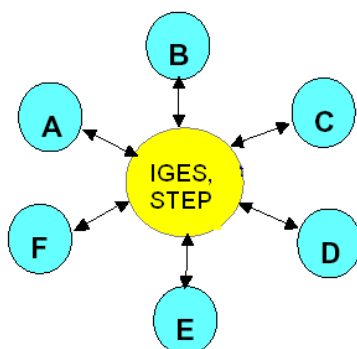


Рис. 14. Диаграмма связи форматов хранения различных систем при использовании международных стандартов передачи

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

Таким образом, появляется проблема создания прикладной программы, использующей внутренние форматы хранения данных систем автоматизированного проектирования. 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.

Далее рассмотрим перечень два основных наиболее значимых формата в контексте конвертации их в систему поддержки ЖЦИ.

2.1 Формат хранения .cdw

Используется в российской САПР «Компас», разработан компанией «Аскон». Предназначен для хранения и использования внутри данной САПР, содержит информацию о чертеже в бинарном виде. Содержит двухмерные данные, представляющие собой упорядоченный набор геометрических параметров: 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 и отсутствие дублирования данных.

Недостатками формата является то, что все данные о чертеже хранятся в бинарном виде, что неудобно для расшифровки без спецификации.

2.2 Формат хранения .dxf

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.

К числу достоинств данного формата можно отнести хранение ГГИ 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.

Таблица 1. Числовые соотношения средних длин файлов систем

Система	КОМПАС	AUTOCAD	CATIA2D	CATIA3D	КОМПАС
КОМПАС	⊕	0.45	0.31	0.125	⊕
AUTOCAD	2.22	⊕	0.684	0.278	2.22
CATIA2D	3.226	1.46	⊕	0.407	3.226
CATIA3D	8	3.6	2.457	⊕	8

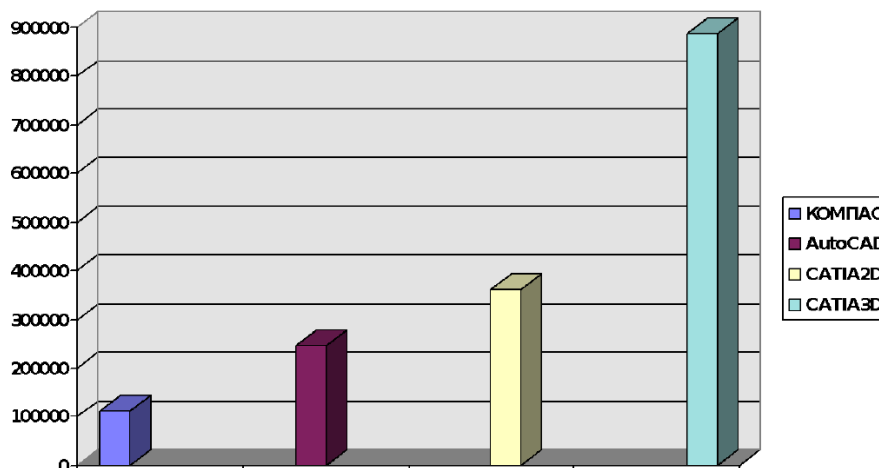


Рис. 15. Диаграмма соотношения средних длин файлов систем

В кандидатской диссертации Г.Б.Широкого «Средства архивизации геометро-графической информации в процессе автоматизированного проектирования на крупном предприятии» [8] описаны некоторые существующие форматы хранения ГГИ, а так же проведен анализ объема выходного файла у различных систем для одинакового чертежа. Часть данных приведена на рис. 3 и в таблице 1, из которых следует вывод о разреженности внутренней структуры хранения данных у систем Autocad, Catia2d и Catia3d по сравнению с системой Компас, что увеличивает время доступа к данным и их обработку.

Рассмотрим подробнее процедуры чтения и записи данных формата .cdw.

2.3 Чтение и запись формата .cdw

Чтение бинарного формата .cdw осуществляется из прикладного приложения путем взаимодействия с САПР «Компас» через предлагаемое компанией «Ас-кон» API и СОМ-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 СОМ-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.

Перейдем к исследованию возможности предобработки ГГИ и исправлению численных ошибок проектирования.

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). На рис. 4 приведен пример подобной работы: слева – вариант необходимыми размерными числами; справа – тот же чертеж, только размерные числа проставлены самой САПР.

Исходя из ГОСТ 2.307-2011 «Нанесение размеров и предельных отклонений», пункт 4.1: «Основанием для определения величины изображенного изделия и его элементов служат размерные числа, указанные в графическом документе.» при изготовлении детали нам следует полагаться на числа, стоящие над размерной линией.

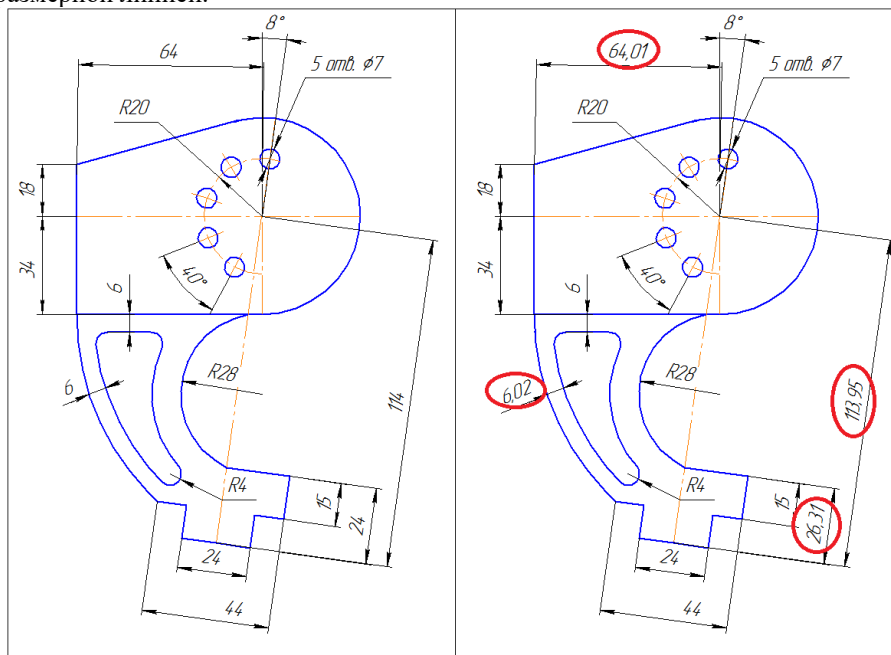


Рис. 16. Пример чертежа с различающимися длинами отрезков и размерным числом

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).

На рис. 5 можно заметить несколько размеров, труднореализуемых на практике в связи с очень высокой требуемой точностью исполнения.

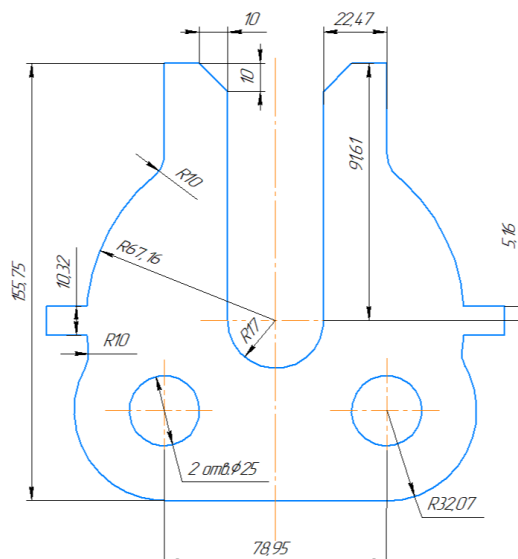


Рис. 17. Пример чертежа с размерами, требующими высокой точности при изготовлении

Исходя из ГОСТ 2.052-2006 «ЭЛЕКТРОННАЯ МОДЕЛЬ ИЗДЕЛИЯ. Общие положения», пункт 4.7: «Электронный конструкторский документ, выполненный в виде модели, должен соответствовать следующим основным требованиям:

- 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);
- все значения размеров должны получаться из модели».

Таким образом, при автоматическом чтении прикладной программой, лишь использующей имеющуюся ГГИ, 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.

Возникает необходимость в предобработке двумерной ГГИ перед её непосредственным использованием путем выявления и обнаружения вышеописанных ошибок проектирования.

3 Алгоритмы и методы получения геометро-графической информации

Для реализации операций чтения и записи ГГИ на примере API САПР Компас-График [9] в прикладной программе необходимо проделать следующие действия (блок-схема на рис. 6):

1. Создать COM-объект для взаимодействия между САПР и прикладной программой.
2. Открыть необходимый графический документ для считывания информации.
3. В активном документе создать итератор для получения доступа ко всем объектам в документе.
4. При помощи итератора обойти все объекты в графическом документе.

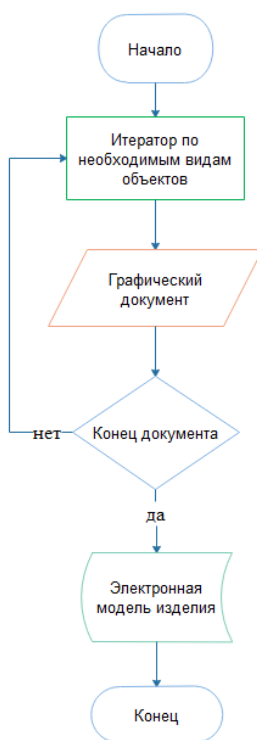


Рис. 18. Принцип работы прикладной программы при чтении ГГИ

3.1 Создание COM-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).

3.2 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.

3.3 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 (блок-схема на рис. 7).

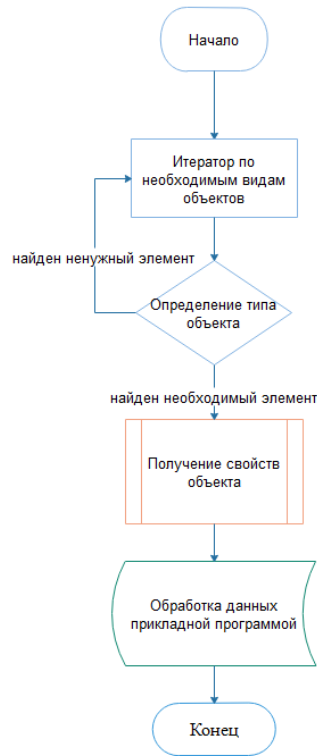


Рис. 19. Алгоритм работы итератора на модели данных

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.

3.4 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 ис-

ключения объектов (при исключении всех объектов из именованной группы она автоматически удаляется).

3.5 Чтение свойств объекта

Из объекта, на который указывает итератор, можно получить список свойств текущего объекта. Полный список видов объектов, встречающихся в графических документах и поддерживаемых Компас-3D, можно прочитать в справке по SDK Компас-3D.

3.6 Запись во внутренний формат базы данных

После получения и обработки всей ГГИ прикладной программой, производится запись в базу данных.

3.7 Закрытие 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.

4 Разработанный алгоритм формирования трехмерной каркасной модели на основе двумерных проекционных изображений

Первый этап формирования трехмерной каркасной модели [10] – получение двумерной ГГИ прикладной программой. В основном получаемая двумерная информация имеет систему координат, 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 представлен на рис. 8.

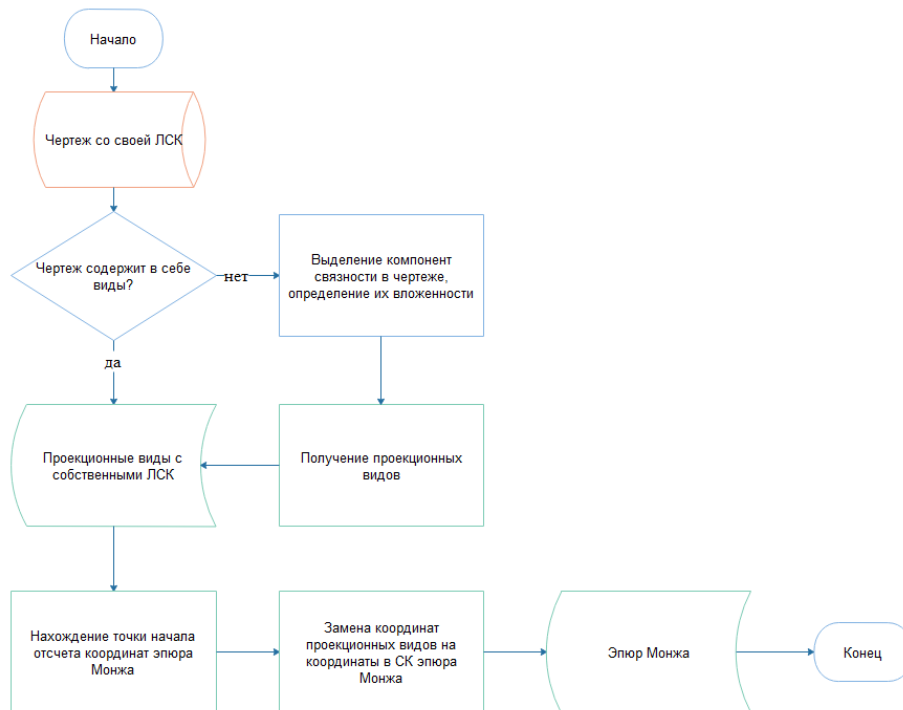


Рис. 8. Блок-схема алгоритма получения из внешней ГГИ эпюра Монжа.

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 точку на виде спереди, получив 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 $(x1, y2, z1)$.

На третьем этапе путем добавления ребер в точечную модель получается искомая каркасная трехмерная модель. Ребра же добавляются исходя из следую-

шего утверждения: если для точек (x_1, y_1, z_1) , (x_2, y_2, z_2) верно, что на соответствующих проекционных изображениях существуют ребра $(x_1, z_1) - (x_2, z_2)$; $(x_1, y_1) - (x_2, y_2)$; $(y_1, z_1) - (y_2, z_2)$ или соответствующие точки совпадают, то, скорее всего, существует ребро $(x_1, y_1, z_1) - (x_2, y_2, z_2)$.

При наличии на входных проекционных изображениях ребер второго порядка алгоритм получения трехмерной каркасной модели фактически не изменяется. Отличия состоят в том, что имеющееся ребро второго порядка дополнительно разбивается несколькими характерными точками. Количество точек зависит от необходимой точности отображения, минимальное количество таких точек для окружности – 4, исходя из количества пересечений двумя секущими плоскостями, каждая из которых параллельна соответствующей координатной плоскости и проходит через центр окружности. Далее находятся ребра, являющиеся отрезками, с вершинами в этих характерных точках, и восстанавливаются трехмерные ребра с аппроксимацией кубическими сплайнами. После восстановления эти точки и ребра не учитываются при восстановлении остальной модели.

5 Временные характеристики разработанного алгоритма

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. Экспериментальная программная реализация показывает результат уже через 0.5-3 секунды после запуска процесса чтения двумерной информации на простых трехмерных моделях.

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.

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. Данные, полученные в ходе исследования, приведены в таблице 2.

Таблица 2. Временные характеристики разработанного алгоритма

Количество вершин	Общее время восстановления, сек.	Время чтения двумерных данных, сек.	Время восстановления точечной модели, сек.	Время получения каркасной модели, сек.
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

Из проведенного исследования [11] можно сделать следующие выводы:

1. на моделях до 1000 вершин около 90% времени занимает чтение геометрической информации 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.

6 Заключение

1. Проведен анализ публикаций по существующим методам и алгоритмам преобразования двумерных геометрических моделей в трехмерные геометрические модели. Выявлено отсутствие во всех системах геометрического моделирования и компьютерной графики оператора преобразования проекционных изображений в трехмерную модель.
2. Разработан алгоритм преобразования двумерной геометрической модели в трехмерную каркасную модель, содержащую ребра, описываемые уравнениями до второго порядка включительно, в котором сняты ограничения на количество геометрических элементов, как во входных данных, так и в синтезируемой модели.
3. Выполнена программная реализация разработанного алгоритма преобразования двумерной геометрической модели в трехмерную каркасную модель на языке программирования C++, работающая в современных версиях OS Windows и позволяющая установить временные зависимости составных частей алгоритма.
4. Экспериментальная программная реализация позволяет ускорить процесс создания трехмерной каркасной модели в 60-200 раз (в зависимости от сложности модели) по сравнению с созданием этой модели ручным способом в одной из САПР.

Литература

1. Тани, Х.И. Ввод чертежей и синтез пространственного образа тел в ЦВМ: Автореф. дис. канд. техн. наук / Х.И. Тани. Минск, 1968. -23 с.
2. 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.
3. Полозов В.С. Моделирование и синтез операторов геометрического расчета и машинной графики в системах автоматизированного проектирования и автоматизации технологической подготовки производства. // Автореф. дисс. докт. техн. наук. М., МАИ, 1983.
4. Idesawa M. (1973). A system to generate a solid figure from three views. Bull JSME, 16, 216-225.
5. Балабан, О.М. Разработка метода синтеза 3D моделей изделий по информации, регламентированной ЕСКД. // Автореф. дисс. канд. техн. наук, Н.Новгород. 1995.
6. Тюрина, В.А. Разработка методов преобразований каркасной модели в задаче синтеза образа 3D-объекта по его проекциям // Автореф. дис. . канд. техн. наук / В.А. Тюрина. Н.Новгород, 2003
7. Зудин А.А., Ротков С.И. Графические диалоговые средства построения конечно-элементных сеток в системе «КИТЕЖ» // В сб. Прикладные проблемы прочности и пластичности. Численное моделирование физико-механических процессов. Горький, 1988.
8. Широкий Г.Б. Средства архивизации геометро-графической информации в процессе автоматизированного проектирования на крупном предприятии. // Диссертация кандидата технических наук. Нижний Новгород, 1999, 126 стр.
9. Роменский, С.А. Передача геометро-графической информации из системы автоматизированного проектирования в прикладную программу на примере САПР “Компас-График” // Системы проектирования, технологической подготовки производства и управления этапами жизненного цикла промышленного продукта (CAD/CAM/PDM-2015): сб. тр.
10. Роменский, С.А. Синтез 3D модели объекта по изображениям на поле чертежа / С.А. Роменский, С.И. Ротков, М.М. Смычк, В.А. Тюрина // GraphiCon 2016: труды 26-й Междунар. конф. по компьютерной графике и машинному зрению (Нижний Новгород, 19-23 сентября 2016) / Нижегородский архитектурно-строительный университет. – Нижний Новгород, 2016. с. 272-275.
11. Мошкова, Т.В. Исследование временных характеристик алгоритма восстановления каркасной модели по проекционным изображениям / Т.В. Мошкова, С.А. Роменский, С.И. Ротков, В.А. Тюрина // GraphiCon 2018: труды 28-й Междунар. конф. по компьютерной графике и машинному зрению Томск, 24–27 сент., 2018 г.) / Нац. исслед. Том. политех. ун-т. – Томск, 2018. с. 369-371