

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего
образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»
(ННГУ)

СЛЯДНЕВ СЕРГЕЙ ЕВГЕНЬЕВИЧ

Модели и методы реинжиниринга сложных объектов индустриального
проектирования

Направление подготовки 09.06.01 «Информатика и вычислительная техника»
Направленность подготовки 05.13.18 «Математическое моделирование, численные
методы и комплексы программ»

Научно-квалификационная работа
(диссертация)

Научный руководитель:

доктор технических наук, доцент,
профессор ННГУ им. Н.И. Лобачевского
Турлапов Вадим Евгеньевич

Нижегород – 2019

Оглавление

Введение. Общая характеристика работы	3
Глава 1. Обзор методов реинжиниринга САD-моделей	7
1.1 Геометрические формы представления объекта.....	7
1.2 Открытые и коммерческие системы реконструкции	7
1.3 Реконструкция моделей в параметризованном виде	9
1.4 Интерактивная и автоматическая реконструкция	10
1.5 Выбор форматов хранения и передачи данных	11
Глава 2. Метод реконструкции на сети кривых	13
2.1 Математическая модель процесса реконструкции	13
2.2 Структура алгоритма реконструкции	14
2.3 Проблема чтения исходных данных	14
2.4 Разнообразие дефектов входных данных и подготовка триангуляции	15
2.5 Выбор топологии в виде сети кривых	16
2.5.1 Полуреберное представление	16
2.5.2 Направленный граф топологии	17
2.5.3 Автоматический выбор топологии.....	17
2.6 Выбор топологии сечением модели	19
2.7 Построение поверхностей по сети кривых.....	19
2.7.1 Построение поверхности Кунса и связанные проблемы	20
2.7.2 Извлечение узлов триангуляции	21
2.7.3 Параметризация облака точек	22
2.7.4 Аппроксимация В-поверхности со сглаживанием	26
2.7.5 Локальное сглаживание регулярной сети кривых	30
2.8 Построение поверхности скиннинга.....	31
2.9 Проверка качества реконструкции.....	32
2.10 Граф зависимостей для реконструкции на регулярной сети кривых	33
Глава 3. Программный комплекс	39
3.1 Архитектура программного комплекса	39
3.2 Объектная модель данных	39
3.3 Подсистема визуализации.....	40
3.4 Геометрическое ядро	42
3.5 Примеры	43
3.5.1 Реконструкция параметрической модели лопатки турбины	43
3.5.2 Реконструкция объектов свободной формы.....	45
Заключение	47
Литература.....	48

Введение. Общая характеристика работы

Актуальность темы диссертационной работы

Воссоздание точной цифровой модели объекта из неполного описания его геометрической формы остается актуальной инженерно-конструкторской проблемой [0]. Задачами реконструкции занимается дисциплина, известная как обратный инжиниринг (реинжиниринг). Ключевым этапом промышленного реинжиниринга является моделирование геометрической формы объекта в виде его граничного представления (boundary representation, B-Rep) [2], исходя из заданного облака точек, либо полигональной сетки. Криволинейное граничное представление является, на сегодняшний день, стандартом де-факто для систем автоматизированного проектирования (САПР). Как отмечают авторы обзора [0], САД-модель (в граничном представлении) является своеобразным «языком», на котором общаются различные средства автоматизации проектирования и производства. Существенной проблемой, поэтому, является обеспечение совместимости различных инженерных программных сред, и реинжиниринг здесь может рассматриваться как одна из типовых подзадач.

В инженерной практике задачи реконструкции имеют различные истоки. Например, если объект создавался без использования САПР, то его цифровое представление отсутствует изначально. Иногда геометрическая модель оказывается недоступна в силу иных причин, таких как устаревание или исчезновение оригинальной САД-системы. В таких случаях целью реинжиниринга является получение ранее не существовавшей или утраченной цифровой модели. Восстановленная геометрия может использоваться для дальнейшего редактирования САД-модели, автоматизированного изготовления детали на станке с ЧПУ, а также для решения частных расчетных задач (исследование прочности, термический анализ, аэродинамический анализ и т.д.) или параметрической оптимизации формы. Если параметрическая оптимизация осуществляется на полигональной модели, то проблема реинжиниринга может возникать при обратном переходе от деформированной сетки [3] к САД-представлению. Подходы обратного инжиниринга находят применение и в качестве основного средства моделирования еще не существующих объектов [4]. Так, в автомобильной промышленности часто используется техника оцифровки физического макета, выполненного скульптором для демонстрации внешних обводов изделия. Иногда геометрическое моделирование выполняется изначально в полигональном виде, после чего реинжиниринг обеспечивает переход к точному граничному представлению [5]. В последнем случае, техники реконструкции играют, в сущности, ту же роль, что и традиционные техники моделирования «снизу-вверх» с той разницей, что вместо работы «с

чистого листа» используется вспомогательная полигональная «подложка». Так или иначе, в задачах реконструкции инженер имеет дело с некоторым опорным геометрическим представлением в виде неупорядоченного или структурированного облака точек, либо поверхностной триангуляции. Опорная геометрия направляет процесс моделирования и служит для сопоставления целевой и воссозданной эталонной формы объекта.

В зависимости от целей обратного инжиниринга, можно рассмотреть различные критерии успешности процесса реконструкции. Например, при создании цифровой копии физического макета, требуется обеспечить минимальную геометрическую невязку между CAD-моделью и ее полигональным представлением. В автомобильной промышленности также предъявляются строгие требования к эстетическим качествам изделия, поэтому в процессе моделирования должны использоваться т.н. «поверхности класса А». С другой стороны, при воссоздании цифровой модели поврежденной детали, либо детали с большим износом, высокая точность реконструкции не только не требуется, но является нежелательной в тех пространственных зонах, где реальная форма существенно расходится с эталонной. Минимизация исключительно геометрической невязки приводит к тому, что результирующая модель оказывается лишенной таких ключевых качеств, как, например, осевая симметрия, параллельность или ортогональность граней и т.п. Таким образом, обратный инжиниринг есть многокритериальный процесс, в котором набор и строгость привлекаемых критериев регулируется в зависимости от конкретного индустриального приложения и планируемой технологии изготовления.

Цели и задачи исследования. Целью настоящего диссертационного исследования является разработка и программная реализация точных, надежных и эффективных методов реконструкции граничного представления объекта из данных оцифровки, представленных неупорядоченным облаком точек, либо поверхностной триангуляцией.

Поставленная цель требует решения следующих задач:

- Исследовать существующие алгоритмы реконструкции поверхностных и твердотельных объектов произвольной формы.
- Развить существующие и разработать новые методы реконструкции, а также необходимые структуры хранения данных.
- Разработать программный комплекс, реализующий предложенные методы.
- Экспериментально проверить корректность реконструкции путём сравнения с реальной формой объекта.
- Оценить (и сравнить с аналогами) эффективность и точность работы программного комплекса, реализованных моделей и методов в задачах реконструкции.

Предметом исследования являются модели, методы и программные архитектуры средств автоматизированной реконструкции граничного представления промышленных объектов; алгоритмизация методов реконструкции поверхностей свободной формы; способы оценки качества реконструкции.

Методы исследования. Решение задач диссертационной работы базируется на теоретических основах информатики, методах и моделях компьютерной графики, прикладной геометрии, численных методах, теоретических основах и моделях аналитической геометрии, а также теории алгоритмов и структур данных.

Научная новизна работы. Получены следующие новые результаты в области реконструкции сложных объектов промышленного назначения:

- 1) Предложен универсальный подход к реинжинирингу объектов свободной формы, основанный на построении параметрической модели управления данными и алгоритмами. Задействованные операторы геометрического моделирования связываются в граф зависимостей. В отличие от аналогов, данный подход впервые задействует параметрическую модель организации вычислений применительно к реконструкции объектов свободной формы. Параметрическое связывание алгоритмов позволяет редактировать реконструируемую модель локально, т.е. не прибегая к ее полному перестроению, что также является одним из достоинств подхода по сравнению с существующими (коммерческими) аналогами.
- 2) Реализована расширяемая система с открытой архитектурой для решения задач реконструкции CAD-моделей. В отличие от аналогов, данная система является единственным специализированным программным продуктом, доступным на некоммерческой основе с исходными кодами.

Достоверность результатов подтверждена проведенными вычислительными экспериментами по реконструкции аэродинамических поверхностей лопаток турбины и некоторых других объектов свободной формы.

Апробация работы. Основные результаты доложены и обсуждены на международных конференциях по компьютерной графике и зрению GraphiCon (Нижний Новгород, 2016; Пермь, 2017; Томск, 2018), а также на семинарах кафедры МОСТ ИИТММ ННГУ.

Публикации. Основные результаты исследования опубликованы в 7 работах, из них 1 – публикация в издании из перечня ВАК, индексируемом Web of Science и Scopus.

Основные положения, выносимые на защиту:

- 1) Интерактивный и автоматический подходы к реконструкции CAD-моделей в граничном представлении с задействованием специализированных операторов геометрического моделирования, объединенных в граф зависимостей для формирования параметрической модели.
- 2) Методы гладкой реконструкции порции параметрической поверхности, включая метод скиннинга со сглаживанием промежуточных кривых, метод аппроксимации со сглаживанием и набор методов сглаживания.
- 3) Программная реализация методов оценки качества реконструкции, таких как анализ распределения кривизн параметрических кривых и поверхностей, анализ аналитических свойств геометрии, анализ геометрической невязки.
- 4) Программная реализация открытой системы геометрического моделирования, специализированной для решения задач реконструкции.

Глава 1. Обзор методов реинжиниринга САД-моделей

Сложности, возникающие в процессе реконструкции, привели к необходимости всестороннего научного исследования для выработки точных, надежных и эффективных алгоритмов реинжиниринга. За прошедшие десятилетия был накоплен большой объем знаний о методах реконструкции САД-моделей в различных постановках задачи. Исследования в этой области прибегают к разнообразным формам представления цифрового объекта, причем выбор «нестандартных» техник моделирования, как правило, оказывается существенным препятствием для индустриализации подхода. Сопутствующее программное обеспечение может быть представлено самостоятельными инструментами, либо быть интегрированным в состав коммерческих САПР. Методы реконструкции могут быть полностью автоматическими или привлекать пользователя. Кроме того, существуют методы, позволяющие реконструировать не одиночный объект, а целое параметрическое семейство объектов. Ниже дается представительный обзор работ, опубликованных по поводу реконструкции САД-моделей. Введенная классификация позволяет рассмотреть проблемную область с нескольких точек зрения, имеющих практическое значение. Кроме того, обсуждается проблема выбора формата для чтения исходных данных и записи результатов реконструкции.

1.1 Геометрические формы представления объекта

Универсальной формой, позволяющей моделировать как каноническую геометрию, так и «скульптурные» поверхности, является NURBS. В то же время, в процессе реконструкции могут использоваться иные схемы представления, например, поверхности подразделения [16, 17] или Т-сплайны [18], с той оговоркой, что в конце моделирования они будут преобразованы в стандартный вид. Так, популярными средствами моделирования с использованием поверхностей подразделения, совместимыми с технологией NURBS, являются дополнения Clayoo и Xirus для САД-системы Rhinoceros, а также, например, плагин Power Surfacing для ПО SolidWorks. Заметим, что решение задач реконструкции нередко осуществляется непосредственно в технологии NURBS на базе существующих коммерческих систем и в качестве их дополнений [33].

1.2 Открытые и коммерческие системы реконструкции

Для решения задач реинжиниринга используют специализированное программное обеспечение или расширения традиционных САПР. К сожалению, в современной литературе уделяется мало внимания архитектурам программных пакетов, предназначенных для реконструкции. Так, авторы обзора [6] отмечают, что исследования в

области обратного инжиниринга часто обходят стороной вопрос о программном обеспечении, реализующем тот или иной подход к реконструкции. По всей видимости, недостаток информации связан с тем, что практически все продукты автоматизированного реинжиниринга в области САПР являются коммерческими, и производители данных систем не заинтересованы в раскрытии деталей их реализации. Вслед за авторами [7], мы считаем, что публикация систем с открытыми исходными кодами способствует снятию барьеров, препятствующих независимому воспроизведению научных результатов. Кроме того, использование открытого программного обеспечения способствует обмену идеями и переносу технологий от академических исследователей к индустрии, как это указано в [8].

В обзорной статье [6] приведено сравнение современного программного обеспечения в смысле полноты реализуемых операторов обратного инжиниринга. В то же время, все рассматриваемые системы являются коммерческими, что оказывается существенным препятствием для их вовлечения в научную деятельность. Автор обзора [9] описывает подходы к реконструкции CAD-моделей, реализованные в широко используемом коммерческом программном обеспечении. В частности, указано, что интерактивные методы, привлекающие пользователя для ручной сегментации сеток, остаются востребованными в индустрии, несмотря на их трудоемкость.

Открытое программное обеспечение широко используется для решения разнообразных научных и инженерных задач [36]. В области трехмерного моделирования популярными системами являются, например, OpenFlipper [23] и MeshLab [24]. К сожалению, обе системы не имеют полноценной интеграции с геометрическими ядрами, реализующими криволинейное граничное представление, и используются, в основном, для работы с полигональными моделями. Система, представленная в рамках настоящего исследования, интегрирована с открытым геометрическим ядром OpenCascade [29], предоставляющим многоцелевые операторы поверхностного и твердотельного моделирования, а также сервисы обмена геометрическими данными (форматы STEP, IGES). CAD-операторы общего назначения (скиннинг, вытягивание, вращение образующей, скругление и т.д.) являются центральными компонентами систем реконструкции, поскольку многие этапы обратного инжиниринга основаны на использовании прямых методов моделирования. В работе [25] описана процедура реинжиниринга с сопутствующим восстановлением истории построения, где операторы OpenCascade используются для создания конструктивных элементов. Растущую роль открытого программного обеспечения в решении задач проектирования отмечают авторы специализированного обзора [26].

1.3 Реконструкция моделей в параметризованном виде

Большинство методов, публикуемых по поводу реконструкции, ограничивается воссозданием «немой» формы изделия (dumb model), которая лишена конструктивных элементов и потому не допускает полноценного редактирования. В частности, для моделей без истории построения оказывается затруднено решение задач параметрической оптимизации. В этой связи, в обзоре [15] указаны два способа параметризации функциональных (сообразующихся с аэродинамическим потоком) моделей: частичная и полная. Отсутствие декомпозиции модели на конструктивные элементы оставляет возможность лишь частичной параметризации. Предпочтительными, поэтому, часто оказываются методы реинжиниринга, допускающие дальнейшее редактирование модели в терминах ее конструктивных элементов. Такие подходы позволяют реконструировать не единственный экземпляр геометрической модели, но параметрическое семейство $S(\mathbf{a})$, где $\mathbf{a} \in \mathbb{R}^N$ – вектор переменных проектирования, а N – размерность пространства параметров. Параметрическая модель является не просто описанием формы, но своеобразным генератором моделей, сохраняющим в их структуре т.н. «замысел проектировщика» (design intent) для различных значений вектора \mathbf{a} . Подходы к реконструкции, восстанавливающие не только «геометрический артефакт», но инженерную концепцию (замысел проектировщика), доминируют, например, в области турбостроения [10, 11].

Развиваются методы реинжиниринга параметрических моделей машиностроительных деталей. Так, авторы современной работы [12] предлагают эффективный метод синтеза CSG-дерева в виде программы, выполняющей булевы операции на предопределенном множестве примитивов, распознанных из полигональных сеток. Метод декомпозиции граничных моделей на элементы объемов изъятия был предложен нами в [13]. В работе [14] представлена серия подходов к реконструкции параметрических моделей на базе коммерческой системы SolidWorks. Авторы [35] занимаются реинжинирингом примитивных конструктивных элементов машиностроительных деталей, подвергшихся значительному износу в ходе эксплуатации.

Существуют практики реинжиниринга, отталкивающиеся от уже существующих параметрических моделей. Такие методы называют «реконструкцией по шаблону», где в качестве «шаблона» выступает исходная модель. Задача реинжиниринга здесь сводится к параметрической оптимизации, а в качестве целевой функции выбирается геометрическая невязка между оптимизируемой формой изделия и исходными данными оцифровки. Реконструкция по шаблону, таким образом, отталкивается от заранее известного «замысла проектировщика», выраженного, например, в виде дерева построения некоторой типовой детали. Существенным ограничением подобных методов следует считать саму

необходимость моделирования начального шаблона. С другой стороны, если шаблон известен (например, взят из каталога), оптимизация его параметров позволяет достичь минимальной геометрической невязки с гарантией качества геометрической модели в смысле соблюдения всех конструктивных ограничений (соосность, параллельность, ортогональность граней и т.п.). Опыт реализации метода реинжиниринга по шаблону на базе систем Siemens NX и MATLAB приведен в статье [34]. Помимо описания подхода и результатов вычислительных экспериментов, работа [34] дает сведения о приложениях метода, а также убедительный анализ трудностей его внедрения (низкая производительность, активное вовлечение квалифицированного пользователя).

1.4 Интерактивная и автоматическая реконструкция

Другим аспектом рассмотрения систем реконструкции является степень автоматизации процесса. В статье [19] предлагается автоматический подход к реконструкции граничного представления формы, набранной из NURBS-поверхностей со стыками порядка гладкости G^1 . С другой стороны, автоматически восстановленная топология модели не отражает замысла проектировщика, поскольку не опирается на инженерное знание о конструктивных линиях модели, если таковые могут быть выделены. Согласно [9], интерактивная реконструкция с пользовательской сегментацией позволяет добиться наилучшего качества результирующих поверхностей. Авторы [20] описывают известную коммерческую систему Tebis, в которой топология модели задается пользователем интерактивно (технология RSC, Rapid Surface Creation). Похожий метод реализован в системе PolyWorks и, например, в дополнении Mesh2Surface для широко известной САПР Rhinoceros (разработчики Mesh2Surface также выпустили комплекс реинжиниринга QuickSurface, основанный на геометрическом ядре OpenCascade). Авторы [21] занимаются построением интерактивной платформы инверсного моделирования, позволяющей реконструировать объекты свободной формы, включая кинематические поверхности. Интерактивные системы не имеют целью заменить человека в процессе реконструкции, но стремятся дать эффективные средства для автоматизации рутинных операций, обнаруживая, тем самым, соответствие с концепцией «intelligence amplification» (IA), данной Ф. Бруксом в [22].

Автоматические системы получили широкое распространение для реконструкции полигональных тел, набранных из канонических примитивов (плоскостей, цилиндров, сфер, конусов). В современной работе [37] авторы приводят результаты экспериментов, направленных на выявление надежного способа распознавания канонических поверхностей из фасетов CAD-моделей.

1.5 Выбор форматов хранения и передачи данных

Открытость есть ключевое требование к формату долговременного хранения информации о промышленном объекте, поскольку использование открытого формата позволяет снизить зависимость проектировщиков от конкретного программного обеспечения. В то же время, эффективная интеграция инженерного ПО без потери и искажения данных остается открытой проблемой [27]. На сегодняшний день, формат STEP (ISO 10303) является наиболее распространенным средством передачи данных между различными САПР. Продолжается развитие протоколов STEP и выработка рекомендаций о способах их применения и интерпретации [28]. Кроме того, широко используются другие открытые форматы, такие как JT, а также сеточные форматы, например, OBJ, STL, VRML, FBX и т.д. В таблице 1 приведено сравнение основных характеристик наиболее востребованных форматов хранения и передачи инженерных данных. Несмотря на то, что формат JT в целом не уступает STEP, а где-то его превосходит (в частности, JT поддерживает уровни детализации геометрии), зрелость STEP и его глубокое проникновение в индустрию, согласно отчету [43], делают данный формат наиболее подходящим для обмена данными.

	STEP	JT	OBJ	VRML	STL
Структура объекта	Да	Да	Нет	Да	Нет
Сеточная геометрия	Да	Да	Да	Да	Да
Точное представление	Да	Да	Нет	Нет	Нет
Цвета, имена, материалы	Да	Да	Да	Да	Нет
PMI	Да	Да	Нет	Нет	Нет
Уровни детализации	Нет	Да	Нет	Нет	Нет

Таблица 1 – Сравнение открытых форматов хранения и передачи инженерных данных в смысле полноты содержащейся информации.

Открытые форматы, как правило, не содержат информации о конструктивных элементах модели и истории ее построения. Тем не менее, во многих задачах автоматизированного проектирования выразительных средств этих форматов оказывается достаточно. Кроме того, появление и развитие систем т.н. «прямого моделирования» снижает требования к полноте передаваемых данных. Такие системы, по замыслу, не ограничены в способности редактировать модели с утраченным деревом построения.

Процедура реинжиниринга, описываемая в настоящем исследовании, завершается

записью итоговой геометрии в формате STEP. В качестве транслятора мы используем открытые компоненты библиотеки OpenCascade [29]. Для передачи сеточных данных используется формат STL.

Глава 2. Метод реконструкции на сети кривых

2.1 Математическая модель процесса реконструкции

Математическая модель процесса реконструкции, данная ниже, основана на принципах, изложенных в фундаментальной работе [2]. Рассмотрим пространство всех абстрактных твердотельных объектов (регулярных множеств, r-sets) M и пространство его представлений R (Рис. 1).

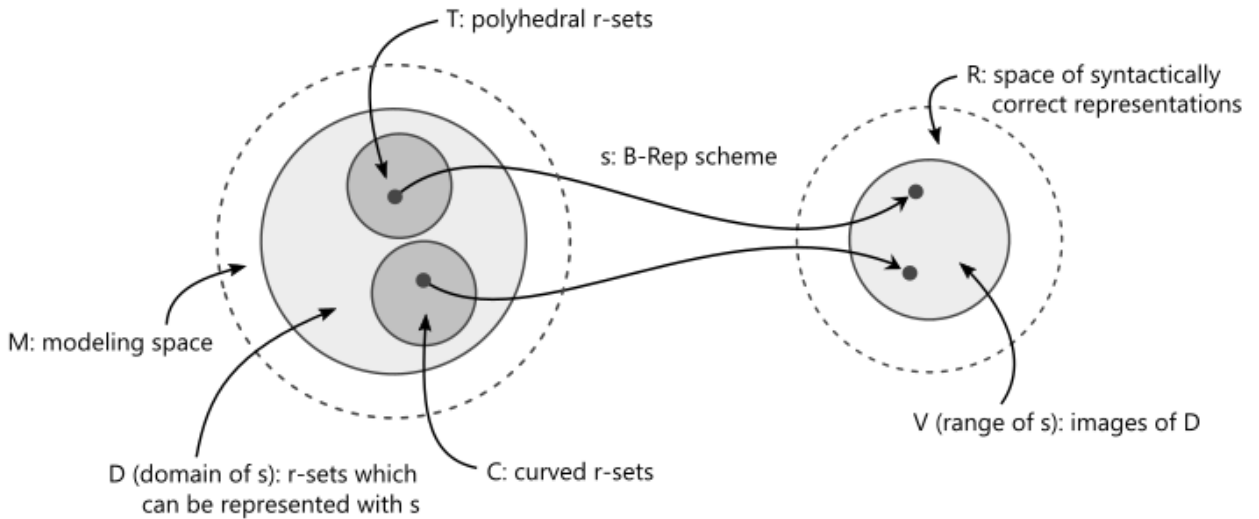


Рисунок 1 – пространство моделирования M с заданной схемой представления s .

Допустим, что на пространстве M определена схема граничного представления s с областью определения D и множеством значений V . Область определения D содержит все твердотельные объекты, описываемые схемой граничного представления. В это множество входят не только криволинейные объекты (подмножество C), но и полигональные тела (подмножество T), являющиеся исходными данными реконструкции (не уменьшая общности, будем полагать, что исходная триангуляция представляет твердотельный объект; для поверхностных объектов формализм аналогичен). Требуется некоторым образом (алгоритмически, при соблюдении заданных критериев) сопоставить объекты области T объектам области C .

Рассмотрим полигональное тело $\tau \in T$ и поставим задачу о нахождении соответствующего ему криволинейного тела $\beta \in C$. Как показывает инженерная практика, для полноценного решения хотя бы узкого класса задач реконструкции, одного алгоритма недостаточно. Возникает потребность в согласованном применении множества геометрических операторов. Процесс реконструкции состоит, поэтому, в поиске такого упорядоченного набора операторов геометрического моделирования $F = \langle F_1, F_2, \dots, F_N \rangle$, что $\beta = F_N F_{N-1} \dots F_1 \tau$, где $\beta \in C$ и соблюдается набор формальных критериев успешности реконструкции $\bar{E}(\tau, \beta) = \bar{1}$; $E : M^2 \rightarrow B^K$, где K есть количество критериев, $B = \{0,1\}$. В

качестве компонент векторного критерия \bar{E} могут выступать функции проверки геометрической невязки, дифференциальных свойств итоговых поверхностей, соблюдения функциональных свойств модели изделия и т.п.

Таким образом, для решения задачи реконструкции необходимо выбрать компоненты вектора F (операторы геометрического моделирования), выполнить их в определенном порядке над полигональным объектом τ и проверить соблюдение заранее зафиксированных критериев успешности \bar{E} .

2.2 Структура алгоритма реконструкции

Математическая модель процесса реконструкции, изложенная выше, допускает алгоритмизацию вычислений, пример которой показан на Рис. 2. Данная схема, как и всякий процесс проектирования, является цикличной, то есть пользователь возвращается к предыдущим этапам в случае, если качество реконструкции оказывается неудовлетворительным в смысле критериев \bar{E} .

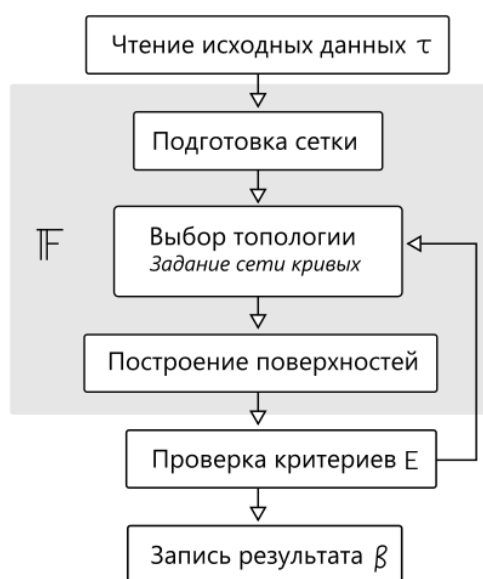


Рисунок 2 – предлагаемый алгоритм реконструкции в виде блок-схемы.

2.3 Проблема чтения исходных данных

Наиболее популярным форматом хранения поверхностной триангуляции на сегодняшний день является STL (stereolithography). Этот формат не содержит данных о топологии модели и, как правило, не передает никакой иной информации, кроме полигональной формы объекта. Благодаря своей простоте, формат STL получил широкое распространение в различных инженерных практиках, таких как 3D-печать (изначально STL создавался для передачи данных на 3D-принтер), конечно-элементный и аэродинамический анализ, визуализация, реинжиниринг.

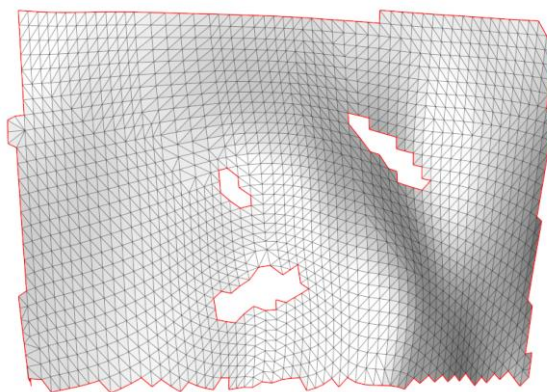


Рисунок 3 – исходная триангуляция; красным цветом отмечены «открытые» ребра.

Большинство программных пакетов для реинжиниринга CAD-моделей предоставляет средства чтения исходных данных в формате STL. Поскольку данные оцифровки, как правило, содержат погрешности и ошибки, представляется целесообразным акцентирование типовых аномалий с целью предостережения пользователя (Рис. 3). Так, реконструкция в зонах, содержащих «открытые» (принадлежащие только одной грани) или немногочисленные (принадлежащие более чем двум граням) ребра, может привести к снижению точности и ошибкам в операторах моделирования.

2.4 Разнообразие дефектов входных данных и подготовка триангуляции

Системы обратного инжиниринга, как правило, содержат набор операторов, позволяющих наилучшим образом подготовить исходную сетку к дальнейшей реконструкции. Среди ключевых операторов отметим децимацию (снижение количества треугольников), устранение зазоров (*gap filling*), подавление незначительных конструктивных элементов (*defeaturing*), повторную триангуляцию для повышения качества сетки, экстраполяцию за пределами границ сетки, инверсию поля нормалей и т.п. Как показывает практика, без подготовительного этапа точная реконструкция модели оказывается существенно затруднена, поскольку данные оцифровки содержат шумы и погрешность, свойственную измерительным приборам. Кроме того, фрагменты поверхности изделия могут оказаться недоступными для измерений, что приводит к «выпадению» некоторых зон из исходного набора данных.

Разнообразие дефектов во входных данных не позволяет в общем случае автоматизировать процесс подготовки триангуляции. Конкретный набор операторов, поэтому, выбирается, исходя из условий, в которых осуществляется реинжиниринг. Выбор зависит от таких факторов, как технология измерения (лазерное сканирование, контактные подходы, фотограмметрия и т.п.), точность измерительных приборов, полнота покрытия измеряемого объекта, желаемая степень автоматизации и проч.

2.5 Выбор топологии в виде сети кривых

Задание топологии на исходной сетке есть процесс ее сегментации. Данный этап является ключевым в задачах реконструкции, поскольку неудачно выбранная топология, как правило, негативно сказывается на точности моделирования.

2.5.1 Полуреберное представление

Топология модели задается сетью кривых на исходной триангуляции. Поскольку реконструируемая модель, в силу постановки задачи, является 2D-многообразием, то для представления границ объекта адекватным выбором является полуреберная структура данных. Всякое ребро (т.н. «первичное» ребро) в данной структуре имеет пару *вхождений*, отвечающих «проходу ребра» на содержащих его гранях (Рис. 4).

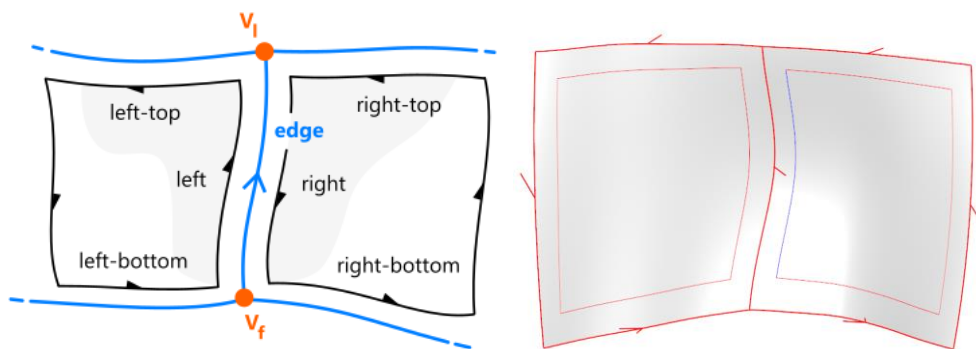


Рисунок 4 – полуреберная структура данных. Слева: принцип организации ребер и полуребер. Справа: визуализация ребер и полуребер на реконструированной модели; цветом показана ориентация (красный – ориентация, согласованная с направлением параметризации кривой; синий – обращенная ориентация).

Первичное ребро представлено параметрической кривой $\mathbf{c}(u) : \mathbb{R}^1 \rightarrow \mathbb{R}^3$, $u \in [0,1]$, имеющей натуральную (геометрическую) ориентацию $\mathbf{c}_u = d\mathbf{c}/du$ в каждой точке. Вхождения ребер представлены указателем на первичное ребро (Рис. 5, 6) с булевым флагом согласования ориентации. Данный флаг (топологическая ориентация) позволяет сгруппировать полуребра в ориентированные замкнутые контуры, описывающие натуральные границы соответствующих поверхностей. Проход контура осуществляется против часовой стрелки со стороны нормали к поверхности.

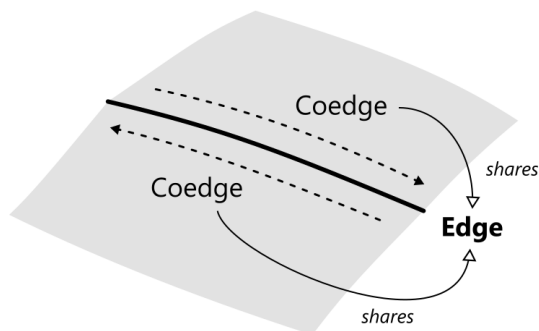


Рисунок 5 – представление полуребер как указателей на первичное ребро.

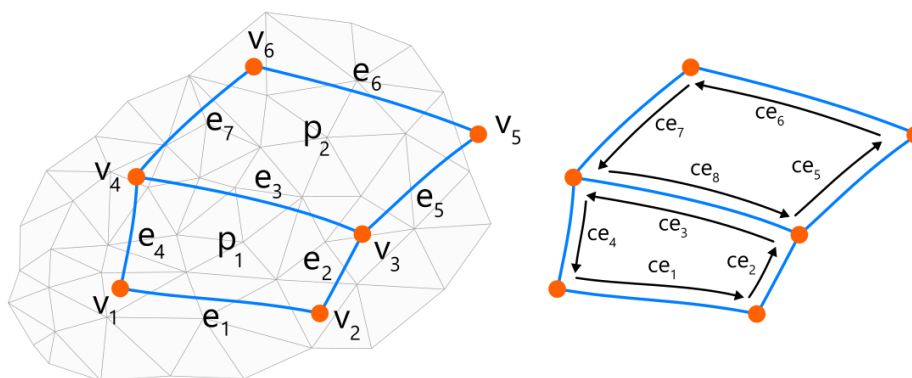


Рисунок 6 – ребра и полуредра топологической структуры данных.

2.5.2 Направленный граф топологии

Сеть кривых определяет направленный граф топологии модели (Рис. 7). Данный граф выражает два основных отношения, существующие в граничном представлении объекта: отношение вложенности («ребро e содержится в грани p ») и отношение смежности («грани p_1 и p_2 являются смежными через ребро e_3 »). Носителями геометрической информации являются узлы типа «грань», «ребро» и «вершина», тогда как узлы типа «полуредра» непосредственно геометрической информации не содержат.

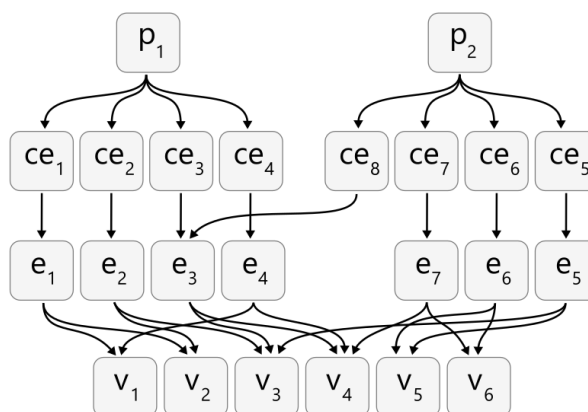


Рисунок 7 – граф топологии модели (стрелками показаны принципиальные зависимости между топологическими элементами).

Граф топологии, формируемый сетью кривых, является также носителем дополнительных параметров реконструкции. Так, для всякого ребра e можно определить порядок гладкости (G^0 , G^1 , G^2), с которым в данном ребре стыкуются соседние поверхности.

2.5.3 Автоматический выбор топологии

Для автоматического преобразования полигональной модели в криволинейное граничное представление в рамках настоящего исследования принят следующий подход. Исходная триангуляция приближается сетью четырехугольных элементов с

контролируемой точностью. Узлы четырехугольников проецируются на исходную триангуляцию вместе со всеми ребрами, в результате чего на полигональном объекте задается топология как сеть параметрических кривых. Таким образом, топология четырехугольной сети в точности переносится на полигональный объект.

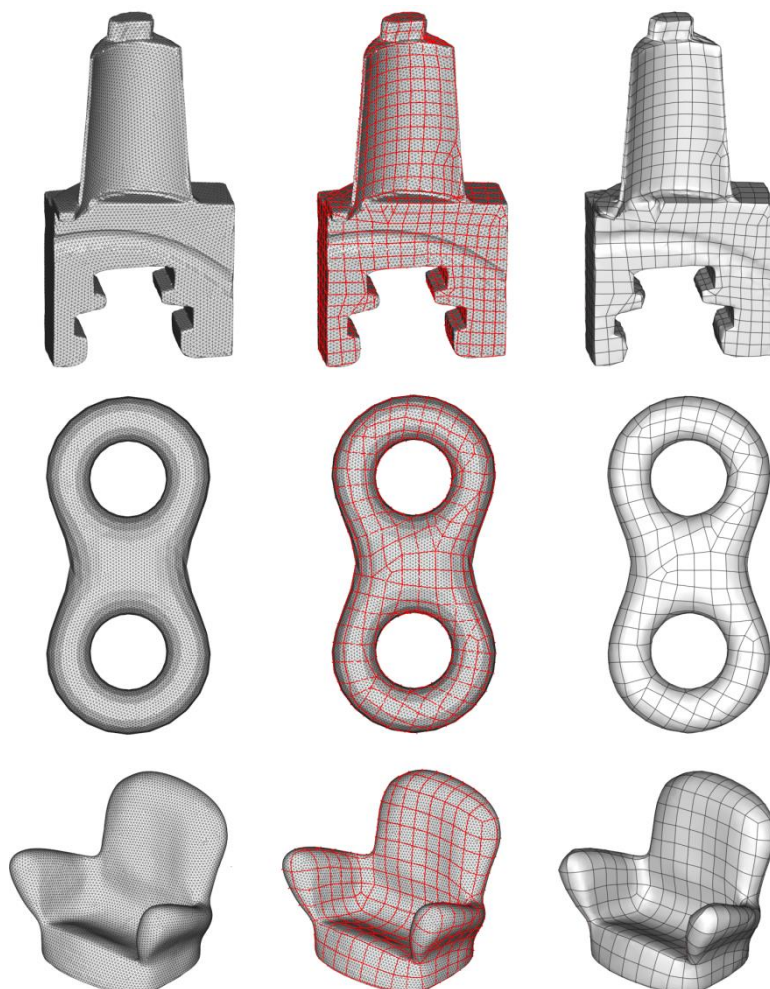


Рисунок 8 – некоторые результаты автоматической реконструкции. Исходная триангуляция (слева) покрывается сетью кривых (посередине), полученной проекцией четырехугольных элементов. Для построения итоговой граничной модели ячейки сети заполняются порциями Кунса (справа).

Существуют эффективные методы преобразования треугольных сеток в четырехугольные, например, с использованием унифицированного локального оператора сглаживания [49]. Данный оператор позволяет сконструировать равномерную сеть четырехугольных элементов, не вполне подходящую для точной реконструкции за отсутствием адаптивности по полю кривизн. С другой стороны, использование данного оператора позволяет полностью автоматизировать ретопологию модели, и, следовательно, процесс реинжиниринга в целом. Отметим, что данный подход к ретопологии нашел применение в решении разнообразных инженерных задач, например, в подготовке конечно-элементных сеток [53]. Некоторые результаты применения оператора равномерной

ретопологии показаны на Рис. 8. Четырехугольная топологическая сеть может быть задана извне, например, при помощи файлов формата OBJ.

2.6 Выбор топологии сечением модели

Объекты, имеющие протяженную форму, создаются, как правило, операторами кинематического моделирования [48]. Те же операторы (вытягивание, вращение, построение по сечениям) оказываются применимы и в задачах реконструкции. Поверхности, форма которых сообразна аэро- или гидродинамическому потоку (такие поверхности называют «функциональными»), часто конструируются техникой скиннинга [47]. Для скиннинга необходимо подготовить упорядоченный набор опорных кривых, образующих изопараметрические линии итоговой поверхности. Интерполяция контрольных точек этих кривых дает искомую протяженную форму. Несмотря на принципиальную простоту метода, надежная и точная реализация скиннинга требует специальной подготовки исходных данных, и, прежде всего, унификации опорных кривых (т.е. их задания на едином базисе). Кроме того, кривым следует обеспечить синхронную параметризацию во избежание появления таких дефектов формы как «скручивание» (Рис. 9).

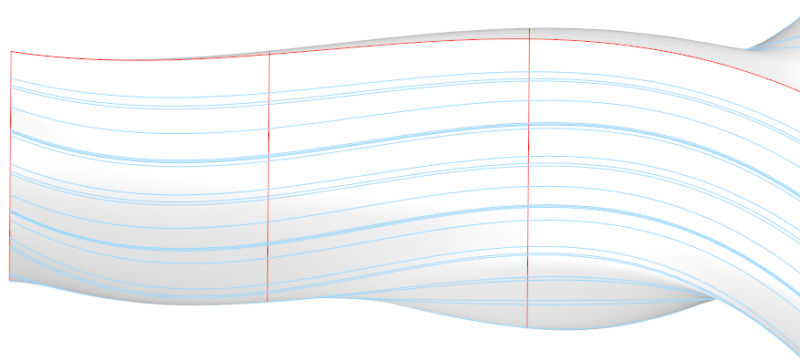


Рисунок 9 – эффект «скручивания», возникающий из-за асинхронной параметризации опорных кривых (замкнутые сечения, направленные вертикально). Видны также сгущения изопараметрических линий, вызванные процессом унификации опорных кривых.

2.7 Построение поверхностей по сети кривых

В рамках диссертационного исследования предложен метод построения поверхности на четырех опорных кривых, состоящий в следующем:

1. Выбирается начальная поверхность в виде порции Кунса. Известно, что поверхность Кунса имеет эквивалентную ей В-поверхность [31].
2. Извлекаются узлы триангуляции, очерченные контуром, представляющим границы порции поверхности. Данные узлы принимаются в качестве неструктурированного облака точек для аппроксимации В-поверхностью.

3. Полученные точки проецируются на поверхность Кунса для получения параметризации.
4. Поверхность Кунса используется в качестве начальной в алгоритме аппроксимации методом наименьших квадратов со сглаживанием.

2.7.1 Построение поверхности Кунса и связанные проблемы

В 1983 У. Гордон дал общий математический аппарат [39] для построения схем т.н. трансфинитной интерполяции, частным случаем применения которого является моделирование поверхности Кунса. Поверхность Кунса [38] интерполирует границы ячейки сети кривых, позволяя, тем самым, построить начальное приближение итоговой формы. Эквивалентное представление поверхностей Гордона-Кунса с использованием В-сплайнов дано в работе [40]. Идея метода состоит в построении булевой суммы $P\mathbf{s} = P_1\mathbf{s} + P_2\mathbf{s} - P_{12}\mathbf{s}$, компонентами $P_1\mathbf{s}$, $P_2\mathbf{s}$ и $P_{12}\mathbf{s}$ которой являются В-поверхности, определенные на едином базисе. Операторы P_1 , P_2 и P_{12} обладают частичными интерполяционными свойствами относительно исходной сети кривых. Оператор P_1 интерполирует кривые $\mathbf{b}_0 = \mathbf{s}(0, v)$ и $\mathbf{b}_1 = \mathbf{s}(1, v)$; оператор P_2 интерполирует кривые $\mathbf{c}_0 = \mathbf{s}(u, 0)$ и $\mathbf{c}_1 = \mathbf{s}(u, 1)$; оператор P_{12} интерполирует угловые точки \mathbf{P}_{00} , \mathbf{P}_{10} , \mathbf{P}_{01} и \mathbf{P}_{11} (Рис. 10).

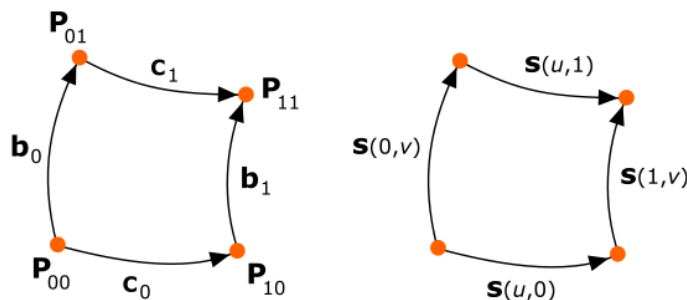


Рисунок 10 – граничные кривые поверхности Кунса в принятых обозначениях.

Пусть заданы В-кривые \mathbf{b}_0 , \mathbf{b}_1 , \mathbf{c}_0 и \mathbf{c}_1 , имеющие общие точки \mathbf{P}_{00} , \mathbf{P}_{10} , \mathbf{P}_{01} и \mathbf{P}_{11} . Не уменьшая общности, допустим, что степень данных кривых не превосходит 3. Построение В-поверхности, эквивалентной поверхности Кунса, состоит в следующей последовательности действий:

1. Приведение кривых \mathbf{b}_0 , \mathbf{b}_1 и \mathbf{c}_0 , \mathbf{c}_1 попарно в совместимый вид путем повышения степени и вставки недостающих узлов [41].
2. Построение В-поверхности $P_1\mathbf{s}$ степени 1 (линейчатой поверхности) путем скиннинга кривых \mathbf{b}_0 , \mathbf{b}_1 .
3. Построение В-поверхности $P_2\mathbf{s}$ степени 1 (линейчатой поверхности) путем скиннинга кривых \mathbf{c}_0 , \mathbf{c}_1 .

4. Построение «косой плоскости» $P_{12}\mathbf{s}$ в виде В-поверхности с параметрическими степенями 1.
5. Приведение поверхностей $P_1\mathbf{s}$, $P_2\mathbf{s}$ и $P_{12}\mathbf{s}$ к единому базису путем поднятия степени 3 (максимальная степень кривых) и перестановкой криволинейных координат (u, v) так, чтобы начало и направления параметризации всех поверхностей совпадали.
6. Сложение координат контрольных точек поверхностей $P_1\mathbf{s}$, $P_2\mathbf{s}$ и $P_{12}\mathbf{s}$.

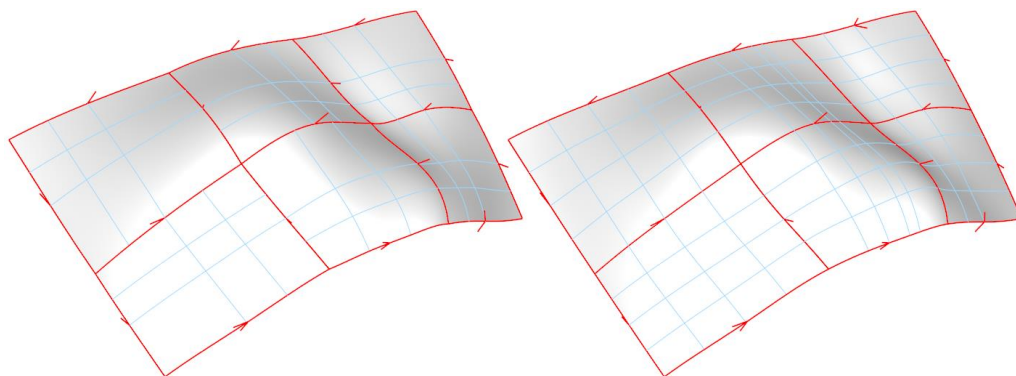


Рисунок 11 – Поверхности, построенные по сети кривых методом Кунса (слева) и уточненные методом аппроксимации со сглаживанием (справа). Методы уточнения рассматриваются далее.

Заполнение ячеек сети кривых порциями Кунса дает начальное приближение для решения задачи реконструкции объектов свободной формы (Рис. 11). Несмотря на то, что иногда изложенный метод оказывается достаточным для практического применения, он имеет следующие недостатки. Во-первых, порция Кунса обладает интерполяционными свойствами только по отношению к граничным кривым, не приближая исходную триангуляцию с непосредственно контролируемой точностью. Во-вторых, билинейные порции Кунса обеспечивают гладкость порядка G^0 (позиционная непрерывность) на стыках, что оказывается существенным ограничением для широкого круга задач реконструкции. Обеспечение гладкости порядка G^1 (тангенциальная непрерывность) возможно путем привлечения бикубических порций Кунса. Однако применение этого расширения на практике затруднено, прежде всего, тем фактом, что изначальная сеть кривых требует соответствующей адаптации для обеспечения гладкости стыков. Следующие разделы описывают предлагаемые способы решения указанных проблем.

2.7.2 Извлечение узлов триангуляции

Перенумеруем узлы триангуляции T , составив множество N индексов всех узлов. Допустим, что r есть множество точек, лежащих на триангуляции T и задающих контур ячейки кривых. Требуется найти область $T_r \subset T$, обведенную контуром r , как множество

треугольников, узлы которых будут использованы для уточнения поверхности Кунса. Дополнительно обозначим через $\partial T_r \subset T_r$ подмножество всех треугольников, включающих точки контура r (Рис. 12). Заметим, что область ∂T_r должна являться двусвязной, так как в противном случае $N_r = N$.

Допустим, что существует треугольник $\tau_0 \in T_r \setminus \partial T_r$, заведомо обведенный контуром r . Требуется найти подмножество индексов $N_r = \{n \in N: \tau_n \in T_r \setminus \partial T_r, \tau_n \leftarrow \tau_0\}$ всех узлов триангуляции, обведенных контуром r (исключая граничный регион ∂T_r), и достижимых из начального треугольника τ_0 путем выборки по смежности. Здесь символ τ_n обозначает любой треугольник, содержащий вершину с индексом n .

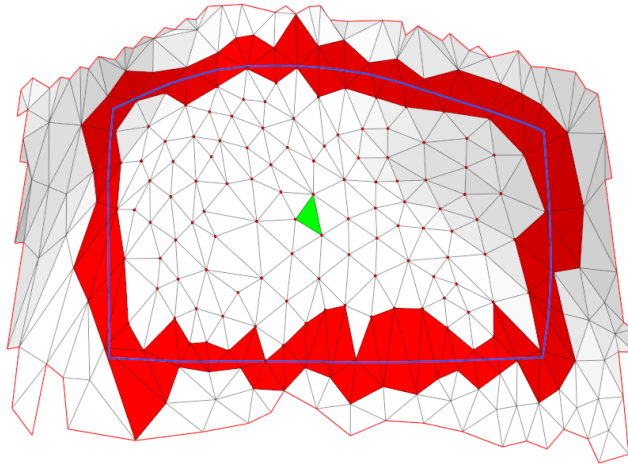


Рисунок 12 – Извлечение области T_r . Красным цветом показана область ∂T_r , содержащая контур r (темная линия). Начальный треугольник τ_0 отмечен зеленым цветом.

Для решения поставленной задачи необходимо выполнить следующие шаги:

1. Вычислить область ∂T_r по заданному контуру и триангуляции. На данном этапе требуется также удостовериться в том, что полученная область является двусвязной (формирует кольцо).
2. Задать треугольник τ_0 , из которого начинается выборка по смежности. На практике, этот треугольник можно выбрать путем проецирования барицентра точек контура r на триангуляцию.
3. Начиная с треугольника τ_0 , осуществить выборку всех достижимых из него треугольников, исключая те, что принадлежат области ∂T_r . Узлы полученных треугольников дают искомое множество N_r .

2.7.3 Параметризация облака точек

Параметризация облака точек с целью последующей аппроксимации поверхности осуществляется путем проецирования каждой точки на опорную поверхность Кунса.

Данная операция известна также как «инверсия точки» [41] (Рис. 13).

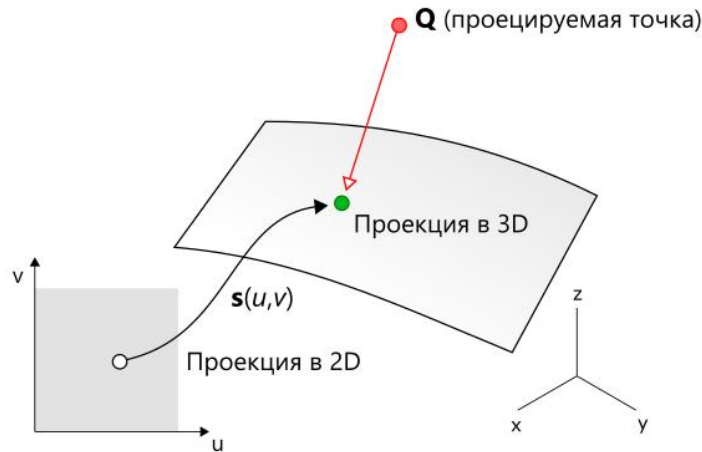


Рисунок 13 – Схематическое изображение процесса инверсии точки Q на поверхность $s(u, v)$.

Инверсия точки задействует следующие вычислительные этапы. Для данной точки Q выбирается ближайшая контрольная точка В-поверхности P_{ij} , индексы i, j которой определяют область $D_{ij} = [u_i, u_{i+p+1}] \times [v_j, v_{j+q+1}] \subset D$, где p и q – степени базисных сплайнов по направлениям u и v соответственно, $D = [u_{min}, u_{max}] \times [v_{min}, v_{max}]$ – область определения поверхности $s(u, v)$. В области D_{ij} выбирается средняя точка (u_0, v_0) , после чего точка $Q_0 = s(u_0, v_0)$ выбирается в качестве начальной для итерационной схемы Ньютона.

Для построения итерационной схемы введем вектор $r(u, v) = s(u, v) - Q$, определяющий отклонение проекции $s(u, v)$ в данной точке параметрического пространства от целевой точки Q . Задачей итераций Ньютона является минимизация функций $f = \langle r, s'_u \rangle$ и $g = \langle r, s'_v \rangle$ как скалярных произведений вектора r на частные производные поверхности по направлениям u и v соответственно. Критерием успешного останова является выполнение условия (1) достижения заданной точности.

$$|f| < \varepsilon, |g| < \varepsilon \quad (1)$$

Здесь ε – точность метода, задаваемая извне. Поскольку метод Ньютона локален, выбор начального приближения (u_0, v_0) позволяет избежать сходимости к нежелательным, но формально корректным решениям (Рис. 14).

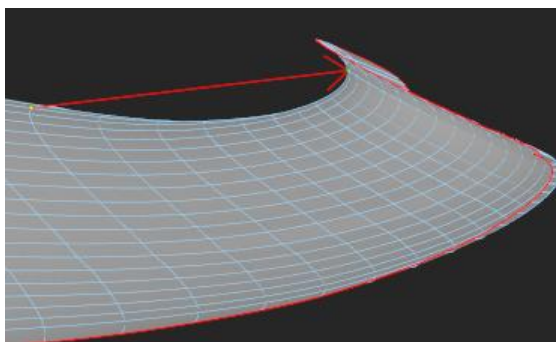


Рисунок 14 – Некорректная инверсия точки, полученная при выборе начального приближения вне области D_{ij} .

В случае, если точка \mathbf{Q} оказывается вблизи границы поверхности, условие $(f, g) = \mathbf{0}$ невыполнимо. Проведенные вычислительные эксперименты позволили модифицировать метод инверсии для корректной работы в этом случае следующим образом. На каждой итерации метода Ньютона проверяется условие $(u_k, v_k) \in D$, где k – номер итерации. Если следующее приближение оказывается вне области определения поверхности, то данное приближение модифицируется так, чтобы точка оказалась на границе ∂D области D (соответствующая координата приближения принимает граничное значение). Поскольку данная модификация метода Ньютона является эвристической и не гарантирует его сходимости, вводится дополнительный критерий останова (2).

$$|(u_{k+1}, v_{k+1}) - (u_k, v_k)| < \delta \quad (2)$$

Здесь величина δ принята равной $1.e-8$. Кроме того, ограничивается количество итераций метода во избежание его закливания (Рис. 15).

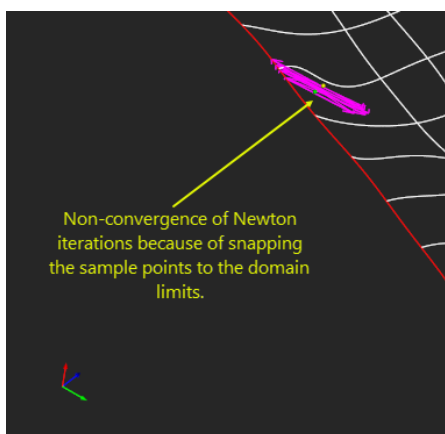


Рисунок 15 – Закливание метода Ньютона при запуске с условием $(u_k, v_k) \in D$.

Если наложение требования $(u_k, v_k) \in D$ нарушает сходимость метода, приводя к его останову по истечении максимального числа итераций, метод запускается еще один раз без ограничений (Рис. 16). Заметим, что в общем случае выполнение итераций Ньютона вне области определения D допустимо, поскольку базисные сплайны определены на всей числовой оси (вычисление сплайнов вне области D нередко используется на практике для

т.н. «натуральной экстраполяции» В-кривых и В-поверхностей [45]).

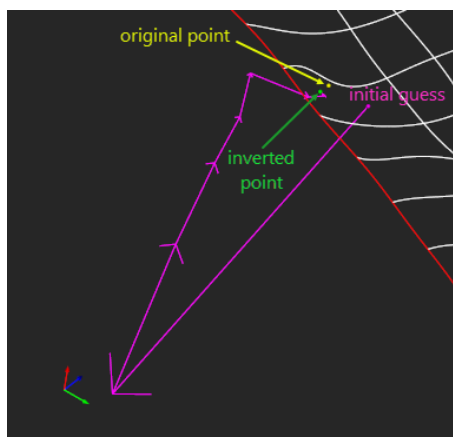


Рисунок 16 – Сходимость вне области D .

Заметим, что поиск ближайшей точки Q_0 , возможно, не самый эффективный способ выбора начального приближения, поскольку на этом этапе выполняется полный перебор контрольных точек поверхности. Кроме того, данный способ может привести к выбору неэффективного, либо неподходящего начального приближения в случае сильно осциллирующей контрольной сетки поверхности. Поэтому для поиска «эффективной» области D_{ij} интерес представляют другие подходы, в частности, связанные с подразбиением В-поверхности. Так, работа [44] описывает эффективный метод инверсии точки путем подразбиения кривых и поверхностей с гарантированным нахождением области D_{ij} , содержащей искомое решение.

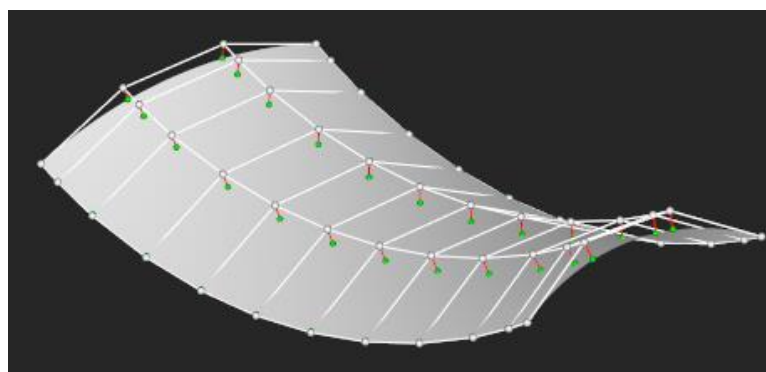


Рисунок 17 – Инверсия контрольных точек поверхности.

Метод, изложенный выше, был протестирован на обширном наборе тестовых данных. С целью верификации метода осуществлялась инверсия контрольных точек В-поверхностей (Рис. 17). Продемонстрированные путем вычислительных экспериментов точность, надежность и достаточная эффективность метода позволили сделать вывод о его пригодности в качестве средства параметризации облака точек. Целью параметризации является назначение каждой точке Q_i пары параметров (u_i, v_i) для формулирования задачи аппроксимации В-поверхности, рассматриваемой ниже.

2.7.4 Аппроксимация В-поверхности со сглаживанием

Задача аппроксимации В-поверхностью неструктурированного облака точек $R = \{\mathbf{R}_k, k = \overline{0, K}\}$ получила широкое освещение в научной литературе. Простейший способ решения этой задачи состоит в применении метода наименьших квадратов (МНК) с использованием начальной В-поверхности, имеющей некоторый, заранее выбранный, базис. Рассмотрим В-поверхность $\mathbf{s}(u, v)$, заданную сетью контрольных точек $\{\mathbf{p}_{ij}\}$ и базисными функциями $N_i^p(u)$, $N_j^q(v)$ в соответствии с формулой (3).

$$\mathbf{s}(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_i^p(u) N_j^q(v) \mathbf{p}_{ij} \quad (3)$$

Для выбора базиса достаточно зафиксировать степени p и q поверхности по параметрам u и v соответственно, а также выбрать узловые векторы $U = (u_0, u_1, \dots, u_r)$ и $V = (v_0, v_1, \dots, v_s)$. Неопределенными остаются координаты \mathbf{p}_{ij} контрольных точек поверхности, которые выбираются алгоритмом так, чтобы минимизировать квадратичную ошибку аппроксимации, представленную функционалом (4).

$$F_{lsq}[\mathbf{s}] = \sum_k (\mathbf{s}(u_k, v_k) - \mathbf{R}_k)^2 \rightarrow \min \quad (4)$$

Поскольку базисные функции известны, а целевая функция является квадратичной, задача сводится к решению системы линейных уравнений. Действительно, единственными переменными задачи (4) являются координаты контрольных точек поверхности. Из необходимого условия экстремума имеем для каждой координаты контрольной точки p_v ($v \in \{x, y, z\}$):

$$\frac{\partial F_{lsq}}{\partial p_v} = 2 \sum_k (\mathbf{s}(u_k, v_k) - \mathbf{R}_k) \frac{\partial \mathbf{s}}{\partial p_v}(u_k, v_k) = 0 \quad (5)$$

Как показывает практика, МНК редко дает удовлетворительный результат. Отметим основные недостатки данного подхода.

Во-первых, остается нерешенной проблема выбора узловых векторов U и V . Если в большинстве случаев выбор степеней p и q аппроксиманта не вызывает затруднений (выбор $p = 3$ и $q = 3$ часто дает удовлетворительные результаты), то компоненты узловых векторов оказывают существенное влияние на поведение итоговой поверхности. Так, вставка дополнительных узлов позволяет достичь большей «податливости» аппроксиманта, но слишком большое их количество сопряжено с существенным снижением производительности операторов моделирования (как самого метода аппроксимации, так и последующих действий) и получением неоправданно сложного представления даже элементарных форм. В то же время, отсутствие, либо недостаточное количество внутренних узлов в направлениях параметризации u и v , делает базовую поверхность слишком

«жесткой», не позволяя МНК преобразовать ее в аппроксимант с необходимой точностью (степеней свободы, даваемых координатами контрольных точек, оказывается слишком мало).

Во-вторых, формулировка задачи аппроксимации в виде МНК с использованием В-сплайнов в качестве базисных функций, имеет существенный недостаток в связи со свойством локальности базиса. Для получения удовлетворительного аппроксиманта необходимо, чтобы «покрытие» базовой поверхности исходным облаком точек было достаточно полным. В противном случае, некоторые базисные функции, не связанные параметризацией с исходными точками, принимают нулевое значение, и матрица СЛАУ, соответствующая МНК, оказывается сингулярной (определитель матрицы равен или близок к нулю). Другими словами, в зонах, где базисные функции оказываются несвязанными с облаком точек, допустимы любые решения, и, в частности, произвольно большие значения координат контрольных точек. В результате, аппроксимант, оставаясь математически корректным, приобретает осциллирующую форму, редко приемлемую для нужд геометрического моделирования.

В-третьих, формулировка МНК не предполагает ограничений на гладкость аппроксиманта. Если исходное облако точек содержит шумы (что характерно в задачах реинжиниринга), то без внесения поправок на гладкость, результирующая поверхность может демонстрировать значительные дефекты формы, недопустимые, например, при моделировании «поверхностей класса А» и в прочих сценариях, предъявляющих высокие требования к качеству геометрического представления объекта.

Недостатки метода наименьших квадратов, изложенные выше, могут быть в достаточной мере компенсированы путем включения в целевой функционал члена, отвечающего за гладкость [42]. Комбинированный функционал (6) агрегирует энергию деформации поверхности F_{smooth} , которая минимизируется вместе с геометрической невязкой F_{lsq} . Коэффициент $\lambda > 0$, балансирующий вклад члена F_{smooth} , выбирается эмпирически, либо путем построения дополнительного цикла оптимизации.

$$F[\mathbf{s}; \lambda] = F_{lsq}[\mathbf{s}] + \lambda F_{smooth}[\mathbf{s}] \rightarrow \min \quad (6)$$

$$F_{smooth}[\mathbf{s}] = \iint (\mathbf{s}''_u)^2 + 2(\mathbf{s}''_{uv})^2 + (\mathbf{s}''_v)^2 dudv \quad (7)$$

Выбор функционала F_{smooth} в квадратичном виде (7), приближающем энергию деформации поверхности, позволяет сохранить линейность задачи при устранении недостатков 2 и 3, заявленных выше. Действительно, агрегированный функционал гладкости дополнительно «связывает» контрольные точки аппроксиманта, значения

координат которых теперь не могут выбираться произвольно из-за ограничений гладкости. Кроме того, сбалансированный коэффициентом $\lambda > 0$ энергетический функционал, позволяет скомпенсировать шум, содержащийся в исходном облаке точек. Неопределенность же выбора узловых векторов U и V решается путем построения итерационной процедуры, задача которой состоит в добавлении новых узлов и повторении аппроксимации с сопутствующим контролем геометрической невязки.

Перенумеруем контрольные точки поверхности (Рис. 18), перейдя к базису $N_l(u, v) = N_{i(l)}^p(u)N_{j(l)}^q(v)$. Уравнение (3) теперь записывается в виде $\mathbf{s}(u, v) = \sum_l N_l(u, v)\mathbf{p}_l$, где $l = \overline{0, L}$.

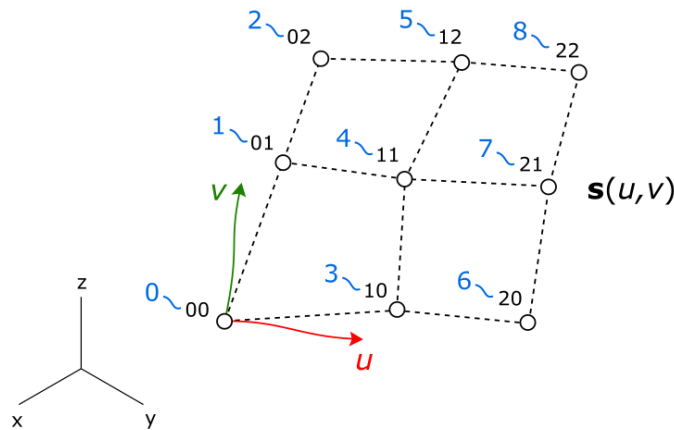


Рисунок 18 – Последовательная нумерация контрольных точек B -поверхности.

Из необходимого условия минимума для задачи (6) имеем матричное уравнение следующего вида:

$$M_1 + M_2 = B \quad (8)$$

Здесь M_1 – матрица коэффициентов, отвечающих функционалу F_{lsq} , M_2 – матрица коэффициентов, отвечающих члену λF_{smooth} , B – матрица правых частей. Матрицы M_1 и M_2 имеют размерности $[(L + 1) \times (L + 1)]$. Матрица B имеет размерность $[(L + 1) \times 3]$, поскольку содержит три столбца для координат контрольных точек x, y, z . Компоненты m_{ij}^1 , m_{ij}^2 , b_i матриц M_1 , M_2 и B соответственно имеют следующий вид:

$$m_{ij}^1 = \sum_k N_i(u_k, v_k)N_j(u_k, v_k) \quad (9)$$

$$m_{ij}^2 = \lambda \iint \frac{\partial^2 N_i}{\partial u^2} \frac{\partial^2 N_j}{\partial u^2} + 2 \frac{\partial^2 N_i}{\partial u \partial v} \frac{\partial^2 N_j}{\partial u \partial v} + \frac{\partial^2 N_i}{\partial v^2} \frac{\partial^2 N_j}{\partial v^2} dudv \quad (10)$$

$$b_i^v = \sum_k N_i(u_k, v_k)r_k^v, \quad v \in \{x, y, z\} \quad (11)$$

Коэффициенты m_{ij}^2 рассчитываются по формулам, данным в работе [51], применительно к следующей задаче. Допустим, имеется начальная В-поверхность $\mathbf{s}_0(u, v)$ и требуется построить сглаживающую В-поверхность $\mathbf{s}(u, v) = \mathbf{s}_0(u, v) + \sum_{l \in L} \mathbf{D}_l N_l(u, v)$, где \mathbf{D}_l – векторы возмущений контрольных точек поверхности, доставляющие минимум функционалу $J = \lambda F_{smooth} + \iint (\mathbf{s} - \mathbf{s}_0)(\mathbf{s} - \mathbf{s}_0) dudv$. Второе слагаемое функционала J не позволяет итоговой поверхности вырождаться в плоскость, чья энергия деформации есть в точности ноль. Замена данного члена функционалом F_{lsq} дает постановку задачи (6). С точки зрения организации вычислений, имеет смысл реализовать подсчет коэффициентов m_{ij}^1 и m_{ij}^2 в независимых программных модулях. Благодаря модульности вычислений, удастся повторно использовать коэффициенты оператора сглаживания для решения задачи аппроксимации В-поверхности. Заметим, что в работе [51] приведены также формулы для построения оператора сглаживания кривых. Соответствующая вычислительная процедура была использована нами для построения оператора скиннинга со сглаживанием [11] для реконструкции лопаток турбин.

В реализованных операторах аппроксимации и сглаживания (для кривых и поверхностей) всюду требуется вычисление значений базисных функций $N_l(u, v)$, а также их производных. Заметим, что коэффициенты функционала сглаживания m_{ij}^2 предполагают двойное интегрирование в области определения В-поверхности, и, следовательно, интенсивное вычисление подынтегральной функции. Для ускорения вычислений используются две техники. Во-первых, для минимизации динамического выделения памяти подготавливается «арена», то есть единый блок памяти, повторно и поочередно используемый функциями. Во-вторых, вся область определения (u, v) В-поверхности разбивается на ячейки для кэширования значений N_l и их производных до второго порядка включительно. Запрос на вычисление базисных функций в некоторой точке (u_i, v_j) предваряется проверкой наличия в кэше ячейки параметрического пространства, включающей данную точку с уже подсчитанными значениями базиса. Если такой ячейки нет, значения базисных функций и их производных вычисляются однократно, и полученный результат поступает в кэш.

Использование методов аппроксимации неструктурированного облака точек дополнительно упрощается, если выполняются следующие условия:

1. Исходное облако точек сегментировано.
2. Аппроксимирующая поверхность не обрезается внешним контуром, то есть сохраняются ее натуральные границы.

Указанные требования соблюдаются автоматически при использовании сети кривых,

формирующей топологическое разбиение реконструируемого объекта на четырехугольные ячейки. Кроме того, подобное разбиение диктует выбор базовых поверхностей аппроксимации в виде порций Кунса. Система линейных уравнений (8) с коэффициентами (9-11) дает координаты контрольных точек, доставляющие минимум функционалу (6). При этом, координаты граничных контрольных точек фиксируются (новые значения им не присваиваются), чтобы обеспечить гладкость порядка G^0 на стыках сети кривых.

2.7.5 Локальное сглаживание регулярной сети кривых

Для регулярной топологии гладкость порядка G^1 между порциями поверхностей может быть обеспечена соответствующей подготовкой сети кривых. Рассмотрим пару кривых $\mathbf{c}(u)$ и $\mathbf{g}(t)$, имеющих общую точку \mathbf{p} (Рис. 19).

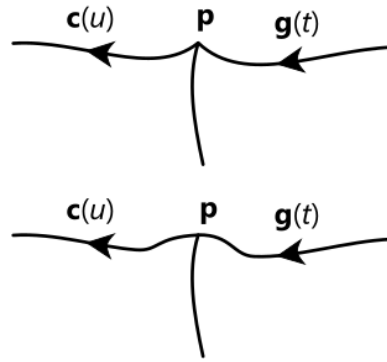


Рисунок 19 – Кривые до и после адаптации.

Требование гладкости порядка G^1 записывается в следующем виде:

$$\mathbf{c}'(u_0) = \delta \mathbf{g}'(t_0) \quad (12)$$

Здесь $\mathbf{c}(u_0) = \mathbf{g}(t_0) = \mathbf{p}$, а δ есть некоторый отличный от нуля коэффициент. В случае, если $\mathbf{c}(u)$ и $\mathbf{g}(t)$ являются В-кривыми, для соблюдения условия гладкости достаточно адаптировать контрольные точки, соседние с точкой \mathbf{p} так, чтобы они лежали на одной прямой. Аналогичным образом адаптируются контрольные точки соседних В-поверхностей.

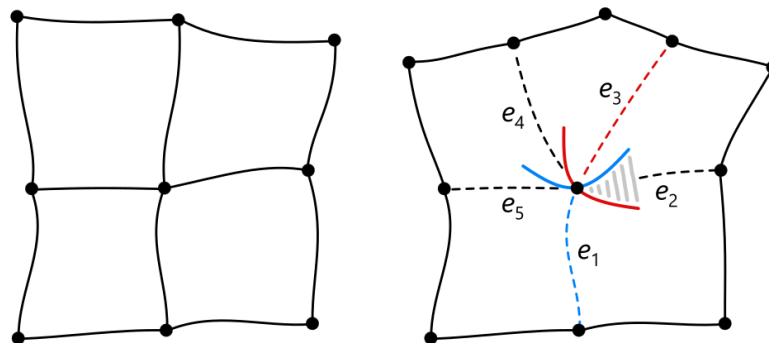


Рисунок 20 – Регулярная (слева) и нерегулярная (справа) сеть кривых. Синим и красным цветами показаны независимые ограничения, накладываемые на гладкость в ребрах e_2 , e_4 и e_5 . Для ребра e_2 ограничения оказываются несовместны.

Подход к адаптации кривых, изложенный выше, может применяться только к регулярной сети кривых, т.е. такой сети, каждая вершина которой принадлежит не более, чем четырем ребрам. В случае, если свойство регулярности не соблюдается, требование (12) приводит к противоречивым ограничениям на ребра, смежные в сингулярной вершине. Так, на Рис. 20 ребра e_1 и e_3 , вдоль которых назначена тангенциальная непрерывность, продуцируют несовместную систему уравнений типа (12) для ребра e_3 .

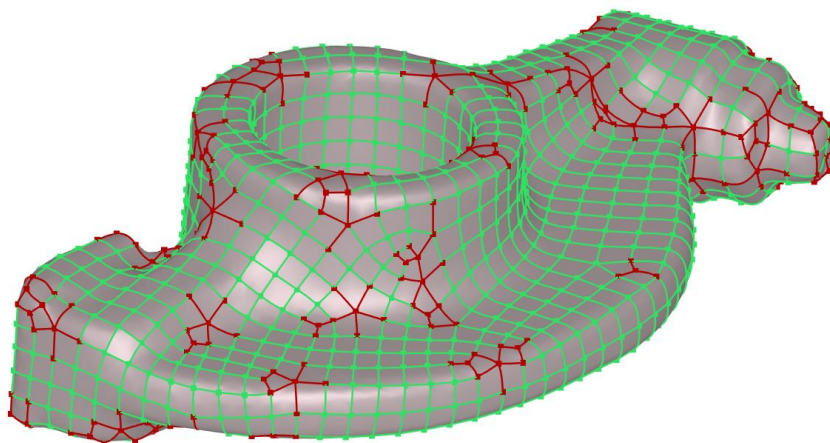


Рисунок XXX – Дефекты гладкости G^1 (показаны красным цветом) в окрестности нерегулярных вершин на автоматически реконструированной модели.

2.8 Построение поверхности скиннинга

Построение В-поверхности, проходящей через k кривых, имеющих n контрольных точек, дает в общем случае $O(k^2n)$ контрольных точек [47]. Например, при скиннинге 10 кривых, имеющих 4 контрольные точки каждая, результирующая В-поверхность будет содержать 400 контрольных точек вместо 40 ожидаемых интуитивно (то есть в k раз больше). Взрывной рост сложности связан с процессом унификации кривых, а конкретно, с процедурой вставки узловых значений (Рис. 21).

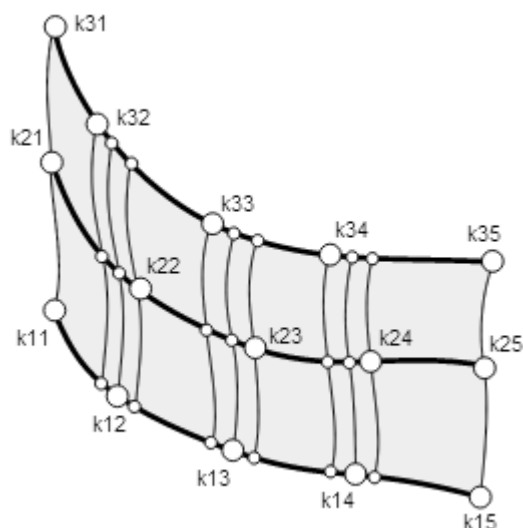


Рисунок 21 – Каждый узел каждой кривой вставляется во все остальные кривые-направляющие скиннинга (k раз).

Оценка сложности итоговой геометрии требует диагностических средств, не всегда доступных в программных пакетах, используемых для моделирования. В качестве косвенных признаков неэффективности геометрического представления могут выступать такие факторы, как низкая производительность алгоритмов, неоправданно большие размеры обменных файлов, повышенное потребление памяти, снижение общей надежности системы и т.п. Визуальную оценку сложности дает картина распределения изопараметрических линий поверхности для узловых значений параметров. Так, характерные сгущения в продольном направлении кинематической поверхности свидетельствуют о неэффективной реализации оператора скиннинга.

2.9 Проверка качества реконструкции

Широко применяемым средством контроля качества реконструкции является измерение геометрической невязки между результирующим телом β и исходной сеткой τ . Наглядное представление о точности реконструкции дает скалярное поле знаковых дистанций. Поскольку точное прохождение восстановленной поверхности через узлы триангуляции, как правило, не требуется, при инспектировании поля невязок следует учитывать величину допустимых отклонений, определяемую постановкой задачи. Так, если расстояние от поверхности до полигональной подложки всюду не превосходит допустимого значения, результат реконструкции полагается точным. Наличие же отклонений вне заданного допуска может быть скомпенсировано соответствующим подбором значений параметров реконструкции и дополнительным подразбиением сети кривых.

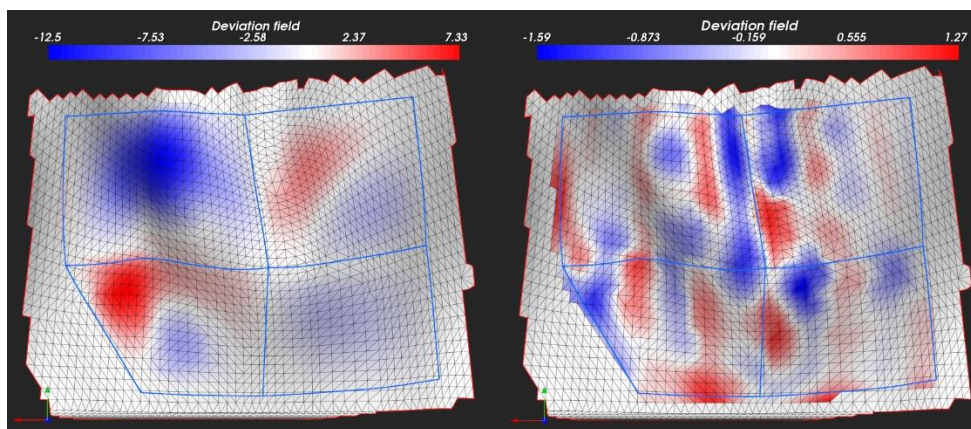


Рисунок 22 – Визуализация поля знаковой дистанции для результатов реконструкции с разной степенью точности: слева – поверхности Кунса; справа – аппроксимирующие B -поверхности со сглаживанием. В обоих случаях величина допуска выбрана равной 0.1 (в единицах измерения модели).

Измерение поля знаковых дистанций осуществляется следующим образом. Результирующее тело β покрывается облаком точек χ путем выборки координат (u_i, v_j) с некоторым шагом в параметрических пространствах каждой порции поверхности. Точки $\mathbf{s}_{ij} = \mathbf{s}(u_i, v_j) \in \chi$ полагаются пробными для замера дистанции до исходной сетки. Проекция \mathbf{t}_{ij} точки \mathbf{s}_{ij} на сетку τ вычисляется с использованием ускоряющей структуры данных BVH (Bounding Volume Hierarchy) [50], широко используемой в алгоритмах трассировки лучей. Знак дистанции принимается равным знаку скалярного произведения $\langle \mathbf{n}_k, \mathbf{v}_{ij} \rangle$, где \mathbf{v}_{ij} есть вектор из точки \mathbf{t}_{ij} в точку \mathbf{s}_{ij} , а \mathbf{n}_k есть вектор нормали к треугольнику с индексом k , содержащему точку проекции \mathbf{t}_{ij} . В результате формируется множество K индексов треугольников, содержащих все точки проекции. При выборе достаточно плотного облака точек χ , множество K насыщается, обеспечивая, тем самым, непрерывное поле дистанций. Максимальное по модулю значение дистанции, достигнутое в данном треугольнике, ассоциируется с его вершинами для визуализации скалярного поля (Рис. 22).

2.10 Граф зависимостей для реконструкции на регулярной сети кривых

Граф алгоритмических зависимостей есть структура данных, связывающая между собой алгоритмы и позволяющая автоматизировать их исполнение при каждом изменении пользователем входных параметров. Граф зависимостей находит широкое применение в инженерном ПО, начиная с задач автоматизированного проектирования [30] и дизайна [32], и заканчивая численным моделированием (в качестве примера можно привести известный программный пакет ANSYS Workbench).

Для построения графа зависимостей выход одного алгоритма связывается (каким-либо способом) с одним из входов другого. Введем бинарное отношение зависимости

между алгоритмами A_1 и A_2 как $A_1 \rightarrow A_2$, где стрелка выражает зависимость входных параметров алгоритма A_2 от выходных параметров алгоритма A_1 . Данное отношение обладает следующими свойствами:

1. Нереклексивность («алгоритм не зависит сам от себя»): $\forall A \in \mathbb{F}, \neg(A \rightarrow A)$.
2. Асимметричность («если алгоритм B зависит от алгоритма A , то алгоритм A не может зависеть от алгоритма B »): $\forall A, B \in \mathbb{F}, A \rightarrow B \Rightarrow \neg(B \rightarrow A)$
3. Транзитивность («если алгоритм B зависит от алгоритма A , а алгоритм C зависит от алгоритма B , то алгоритм C зависит от алгоритма A »): $\forall A, B, C \in \mathbb{F}, A \rightarrow B, B \rightarrow C \Rightarrow A \rightarrow C$.

Запись попарных отношений вида $A_i \rightarrow A_j$ для всех алгоритмов $A_i, A_j \in \mathbb{F}$ (здесь \mathbb{F} – множество операторов реинжиниринга) позволяет сконструировать ориентированный граф, в котором могут оказаться циклы. Появление циклических зависимостей, как правило, нежелательно, поэтому необходимы средства обнаружения соответствующего подграфа (наличие цикла, вообще говоря, не лишает алгоритм возможности выполнить остальные – нециклические – зависимости).

Построение ребра	E
Построение поверхности	P
Сглаживание углов	C
Сглаживание поверхностей	S

Таблица 2 – Функции, используемые для построения графа зависимостей, и их имена.

В Таблице 2 приведены основные функции, используемые для построения графа зависимостей, автоматизирующего процесс реконструкции. Функция построения ребра E (от англ. «Edge») выполняет аппроксимацию упорядоченного набора точек, лежащих на сеточной поверхности, в результате чего конструируется одиночная B -кривая. Функция построения поверхности P (от англ. «Patch») аппроксимирует ячейку сети кривых порцией Кунса с последующим уточнением до B -поверхности общего вида, приближающей узлы триангуляции в данной ячейке. Функция сглаживания углов C (от англ. «Corner») выполняет локальную адаптацию сети кривых для обеспечения возможности гладкого сопряжения между поверхностями. В связке с ней находится функция S (от англ. «Smoothing»), выполняющая локальную модификацию формы смежных поверхностей для обеспечения стыка с порядком гладкости G^1 .

Все данные, используемые программным комплексом в процессе реконструкции,

организованы иерархически. Так, объект типа «Триангуляция» содержит в качестве дочернего объект типа «Топология», организующий структуры данных для представления сети кривых. Объект «Топология» содержит дочерние объекты для представления граней, ребер, полуребер и вершин. Полуребра являются ссылочным типом данных для обеспечения смежности граней на уровне структуры данных. Аналогичным образом ребро ссылается на пару вершин. Каждый объект модели данных набран из некоторого числа т.н. «параметров», имеющих один из 21 возможных типов: целочисленный, вещественный, булев, строковый, B-Rep, триангуляция и т.д. В частности, определены следующие ссылочные типы параметров: узел дерева, ссылка на объект/параметр, список ссылок на объекты/параметры, функция. Алгоритмы, связываемые в граф зависимостей, представлены параметрами типа «Функция», принадлежащими тому или иному объекту.

Шаблон графа зависимостей назовем такой ориентированный граф $G = (A, E)$, который содержит в качестве узлов A абстрактные алгоритмы, а в качестве дуг E принципиальные зависимости между ними. Шаблон графа зависимостей отражает логические взаимосвязи между алгоритмами системы в минимальном виде, без привязки к конкретным данным. Привязку алгоритма к конкретной иерархической структуре данных назовем параметризацией. Допустим, что все объекты модели данных пронумерованы. Параметризацию алгоритма A данными, доступными из объекта K , будем обозначать символом A_K . Параметризованные алгоритмы связываются между собой согласно шаблону графа зависимостей. При этом формируется *экземпляр* графа зависимостей, имеющий, вообще говоря, сложную (для непосредственного восприятия человеком) структуру.

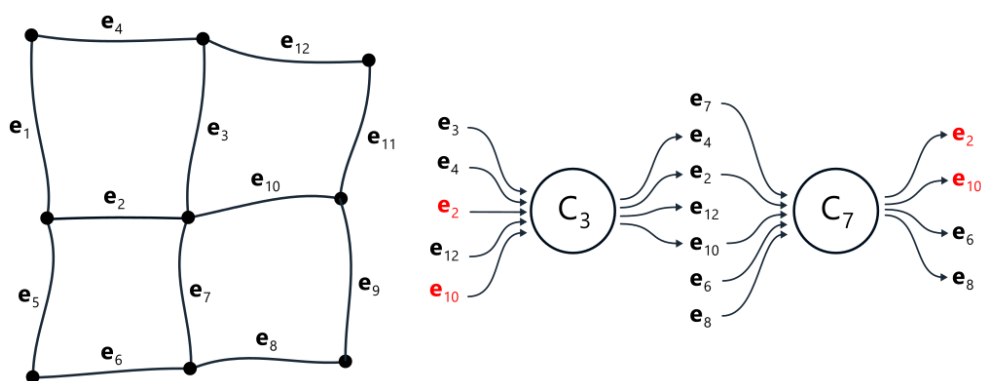


Рисунок 23 – Циклические зависимости при сглаживании ребер e_3 и e_7 . Красным отмечены входы функции C_3 , являющиеся выходами функции C_7 .

Появление циклических зависимостей для функций, заданных на сети кривых, закономерно из-за наличия смежности ребер. В качестве примера рассмотрим функции сглаживания углов C_3 и C_7 для ребер e_3 и e_7 соответственно (Рис. 23). Поскольку обе

функции принимают ребра e_2 и e_{10} , возвращая их же в качестве результата, возникновение циклической зависимости в данном случае оказывается закономерным. В рамках настоящего исследования для организации графа зависимостей диссертантом разработана открытая библиотека Active Data [46], не допускающая выполнение циклически зависящих алгоритмов (за исключением итерационных алгоритмов как таковых). Отметим, однако, что в рассматриваемом примере такие зависимости не являются противоречивыми, поскольку результат сглаживания ребер e_2 и e_{10} одинаков при любом порядке выполнения функций C_3 и C_7 .

Составим шаблон графа зависимостей из алгоритмов (функций), приведенных в Таблице 2. Функция E принимает на входе дискретное представление ребра и выдает кривую, аппроксимирующую заданные точки с некоторой точностью (Рис. 24).

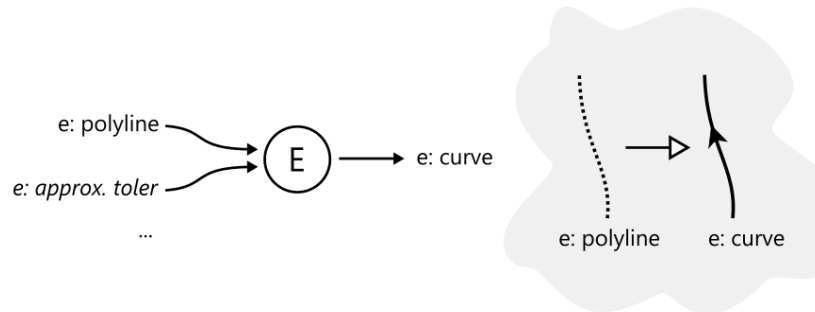


Рисунок 24 – Функция построения ребра E . Параметры, не влияющие на последовательность вычислений, обозначены курсивом.

Функция P выполняет построение поверхности по данному контуру. В качестве входных параметров используются все ребра и полуребра соответствующей ячейки сети кривых. Выходом функции является единственный параметр, представляющий построенную поверхность (Рис. 25).

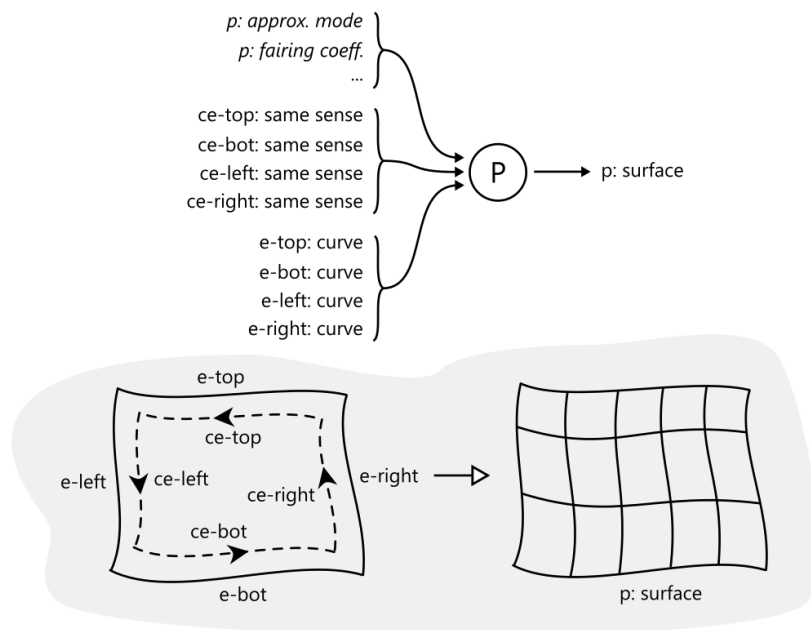


Рисунок 25 – Функция построения поверхности P.

Функция C адаптирует сеть кривых для обеспечения гладкого сопряжения соседних поверхностей. Во избежание появления циклических зависимостей, данная функция параметризована неявно (Рис. 26).

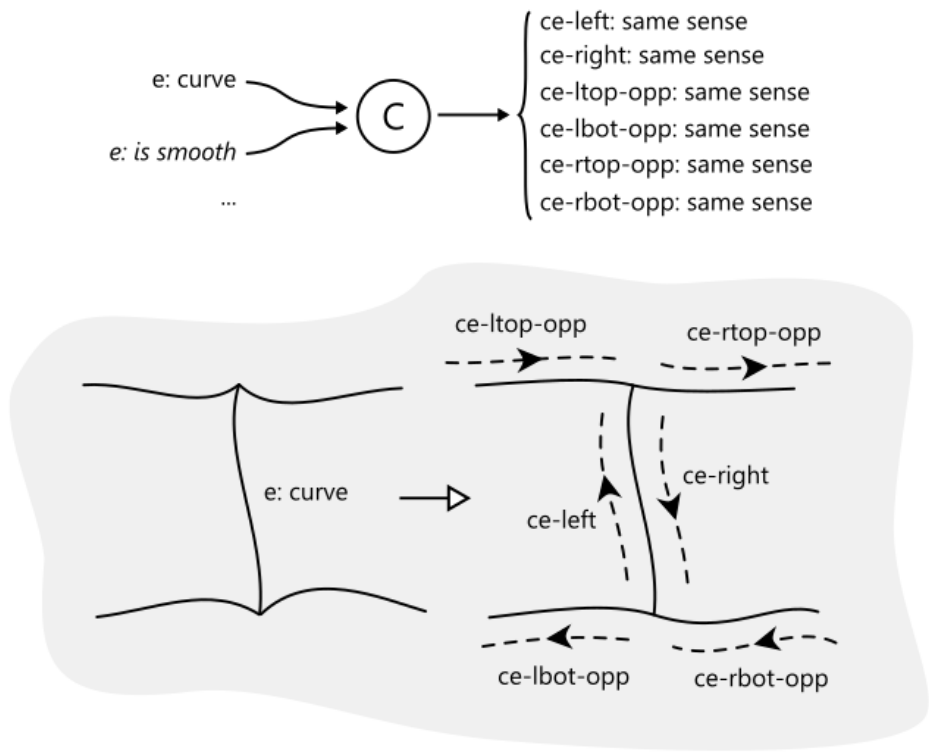


Рисунок 26 – Функция адаптации сети кривых C.

Функция S адаптирует смежные В-поверхности для стыковки с гладкостью G^1 (Рис. 27).

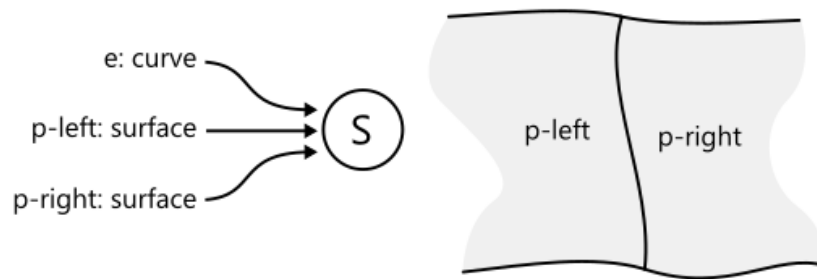


Рисунок 27 – Функция адаптации смежных поверхностей S .

Параметризация алгоритмов настраивается таким образом, чтобы обеспечить желаемый порядок вычислений, не допуская возникновения циклических зависимостей. Функции исполняются в порядке следования ярусов графа зависимостей (Рис. 28). Порядок исполнения функций, принадлежащих одному ярусу, является произвольным.

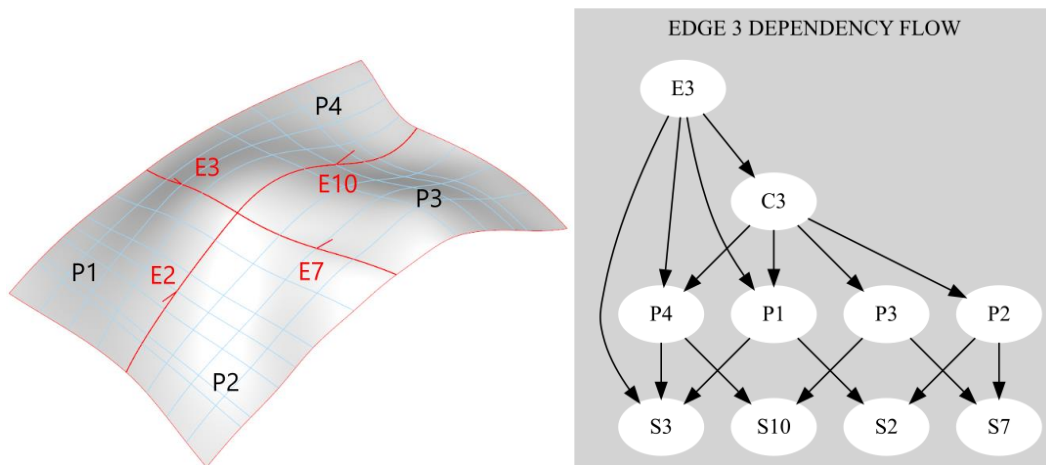


Рисунок 28 – Реконструированная оболочка (слева) и соответствующий граф зависимостей (справа), начиная от ребра с номером 3.

Глава 3. Программный комплекс

3.1 Архитектура программного комплекса

Разработанное программное обеспечение состоит из двух частей: модуль взаимодействия с пользователем (frontend) и собственно расчетное ядро (backend) (Рис. 29). Такое разделение способствует интеграции программного комплекса в стороннее инженерное ПО через предоставляемый прикладной интерфейс (API), обеспечивающий полный цикл решения задач реконструкции. При этом модуль взаимодействия с пользователем может быть полностью или частично заменен, либо адаптирован под нужды конкретного приложения. Основным средством взаимодействия с пользователем остается, на данный момент, текстовая командная строка, реализованная на базе интерпретатора языка Tcl. Помимо командной консоли доступны также базовые элементы управления (реализованные средствами библиотеки Qt) и подсистема визуализации (средствами библиотеки VTK). Расчетное ядро, в свою очередь, предоставляет алгоритмы реконструкции и средства иерархической организации данных с возможностью их параметрического связывания в граф зависимостей.

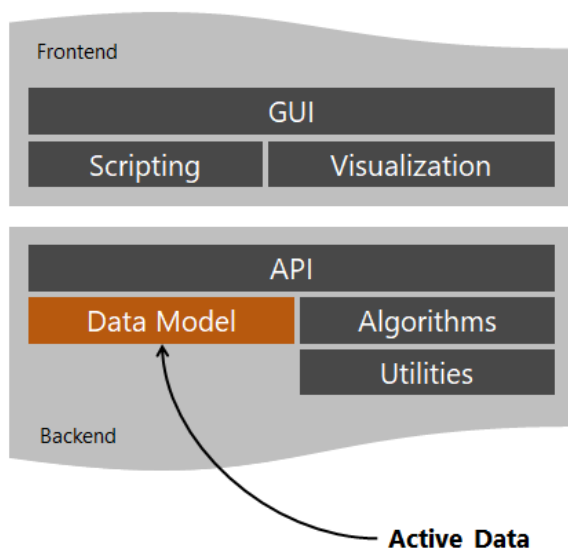


Рисунок 29 – Компоненты архитектуры программного комплекса для реконструкции CAD-моделей.

3.2 Объектная модель данных

Для организации иерархической модели данных используется разработанная диссертантом библиотека Active Data [46], являющаяся надстройкой над компонентами OSAF (Open Cascade Application Framework) библиотеки OpenCascade. Целью разработки данной надстройки было обеспечение объектно-ориентированного подхода к организации и управлению инженерными данными общего вида. Модуль OSAF можно рассматривать

как иерархическую NoSQL базу данных, предоставляющую элементы для конструирования произвольной объектной модели и базовые сервисы CRUD (Create, Read, Update, Delete). Этот взгляд предопределяет архитектурное решение для взаимодействия с хранимыми данными путем введения дополнительных уровней абстракции, таких как DAO (Data Access Object) и сервисный слой, реализующий собственно логику приложения (Рис. 30). Основными объектами библиотеки Active Data являются Узел (Node) и Параметр (Parameter). Узел есть «курсор данных» по отношению к некоторому конкретному объекту. Параметр есть «курсор данных» по отношению к некоторому свойству (внутреннему полю) объекта. Главный принцип «курсора» состоит в том, чтобы не хранить данные непосредственно в объекте, но предоставлять доступ к данным по запросу, путем обращения к хранилищу через внутренний указатель. Этот указатель может представлять собой, например, «соединение», либо, в случае OCAF, «метку» (объект типа TDF_Label) для доступа к иерархически вложенным «меткам», содержащим собственно данные как список «атрибутов» (объектов типа TDF_Attribute).

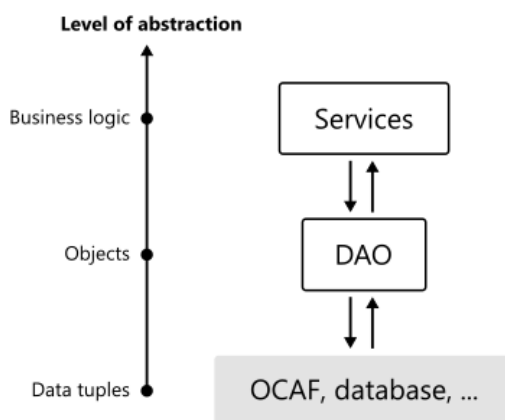


Рисунок 30 – Схема абстрагирования объектной модели «снизу-вверх», от иерархической базы данных (OCAF), через DAO (Active Data) к логике приложения (Service).

Основными объектами программного комплекса являются Триангуляция (Triangulation) и Деталь (Part), представляющие входные и выходные данные для реконструкции соответственно. Ретопология сетки приводит к созданию дополнительных объектов, таких как Грань (Patch), Ребро (Edge), Полуребро (Coedge) и Вершина (Vertex), а также установлению параметрических зависимостей между ними.

3.3 Подсистема визуализации

Модуль визуализации тесно связан с объектной моделью приложения. Для каждого Узла (Node) может быть создан соответствующий ему объект типа Презентация (Presentation), являющийся набором Конвейеров (Pipelines) как алгоритмических цепочек

преобразования данных для их отображения на экране (Рис. 31).

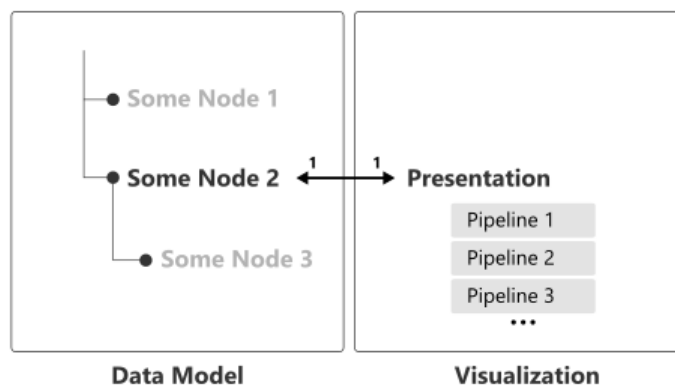


Рисунок 31 – Соответствие «один-к-одному» между Узлами и Презентациями, набранными из Конвейеров.

Соответствие между типом Узла и типом Презентации устанавливается как «один-к-одному». Вариативность же представления объекта на экране достигается путем включения в Презентацию нескольких Конвейеров. Рассмотрим в качестве примера визуализацию В-поверхности. Чтобы представить на экране В-поверхность можно использовать такие типы представлений, как: а) затененная поверхность (shading); б) изопараметрические линии (этот тип представления часто используется в профильной литературе); в) узловые изопараметрические линии (дают представление о качестве параметризации поверхности и дефекте сплайна); г) контрольная сетка; д) начало и направления криволинейных координат (u, v) . Каждый из указанных типов представления соответствует некоторому Конвейеру, включенному в состав единственного объекта-Презентации В-поверхности.

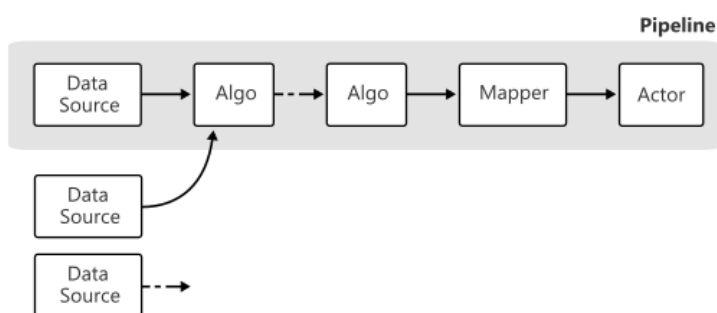


Рисунок 32 – Принцип построения Конвейера визуализации.

Исходные данные для визуализации Конвейером поставляются т.н. Источником Данных (Data Source), не зависящим от ОСАФ или DAO (Рис. 32). Отсутствие прямой связи между Конвейером и хранилищем данных позволяет повысить модульность архитектуры и повторно использовать Конвейер на другом, либо видоизмененном хранилище. Косвенная же связь между Конвейером и хранилищем обеспечивается абстрактным интерфейсом Поставщика Данных (Data Provider), полиморфно используемым для инициализации

Источника Данных (Рис. 33).

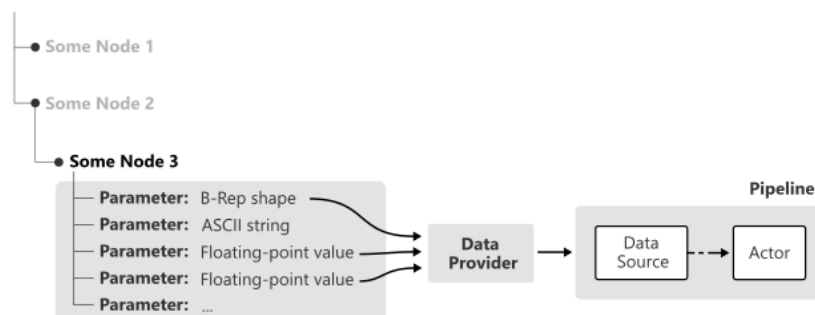


Рисунок 33 – Связь между Конвейером и хранилищем данных через интерфейс Поставщика Данных.

Параметры всех объектов, представленных средствами Active Data, хранят временную метку последней модификации данных. Такую же метку хранит каждый Конвейер. Сравнение временных меток позволяет повысить эффективность визуализации, поскольку повторное преобразование данных осуществляется только для тех Конвейеров, чья временная метка устарела относительно Параметров, зарегистрированных в Поставщике.

В целом, реализованная подсистема визуализации есть адаптация базовой архитектуры VTK для работы с иерархическим хранилищем данных.

3.4 Геометрическое ядро

Основным геометрическим ядром, задействованным в алгоритмах реконструкции, является библиотека OpenCascade, предоставляющая базовые операторы формообразования, такие как выдавливание, построение тел вращения, булевы операторы и проч. Необходимым компонентом этой библиотеки является модуль обмена данными и, прежде всего, транслятор формата STEP, привлекаемый для записи результирующей геометрии.

Кроме того, для обеспечения некоторых этапов реконструкции используется разработанный диссертантом программный комплекс, предоставляющий следующую функциональность:

1. Фундаментальные алгоритмы для работы с базисными сплайнами.
2. Алгоритмы интерполяции упорядоченного набора точек В-кривыми и В-поверхностями.
3. Алгоритмы сглаживания В-кривых и В-поверхностей.
4. Алгоритм аппроксимации неструктурированного облака точек В-поверхностью со сглаживанием.

Обмен данными между двумя геометрическими ядрами осуществляется путем непосредственного преобразования их структур данных, а также при помощи нейтрального

формата JSON.

3.5 Примеры

3.5.1 Реконструкция параметрической модели лопатки турбины

Лопатка турбины является функциональной поверхностью, процесс проектирования которой включает контроль и оптимизацию формы по целому ряду критериев. Данный процесс циклический и состоит из таких этапов, как моделирование начальной формы, численные эксперименты, натурные испытания и внесение апостериорных правок для синтеза результирующей оптимальной геометрии (Рис. 34).

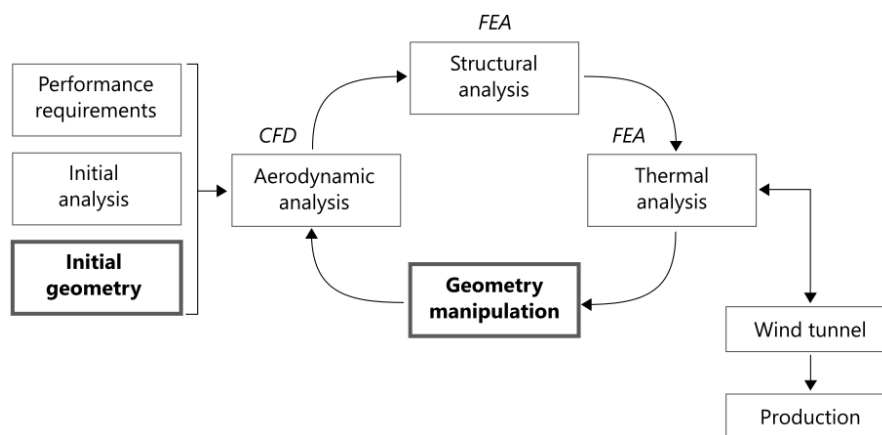


Рисунок 34 – Типовой цикл проектирования лопатки турбины. Выделены этапы, требующие геометрических построений.

Обратный инжиниринг лопатки, как правило, состоит в построении «стопки» ее сечений с последующим протягиванием интерполирующей поверхности (Рис. 35). В случае, если исходные данные упорядочены и сегментированы, процесс реконструкции поддается полной автоматизации. Наличие шума в исходных данных компенсируется, например, использованием техник сглаживания [11]. В результате имеем параметрическую модель лопатки турбины, параметрами которой являются, в простейшем случае, координаты контрольных точек кривых-сечений.

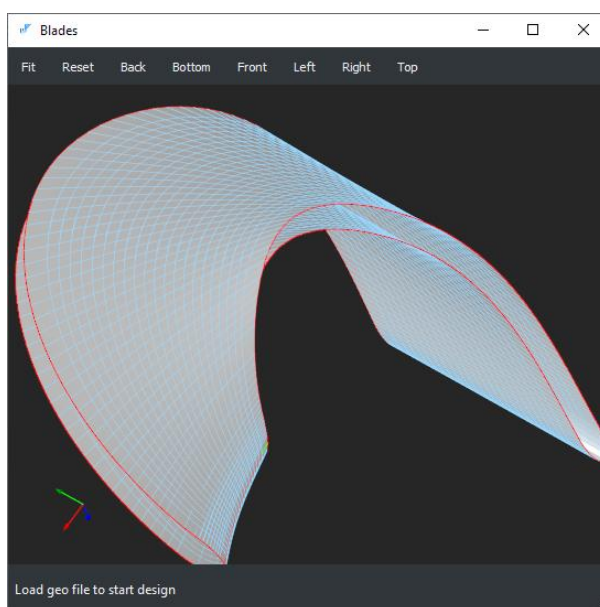


Рисунок 35 – Восстановленная по сечениям поверхность лопатки турбины.

В рамках диссертационного исследования автоматическая реконструкция семейства лопаток осуществлялась из структурированного облака точек, описывающего желаемую конфигурацию линий тока (т.н. задача «инверсного проектирования»). В такой конфигурации были априорно заданы сегменты сечений, представляющие «спинку» и «корытце» лопатки. Кроме того, сами сечения были упорядочены в радиальном направлении рабочего колеса турбины. Задачей алгоритма реконструкции поэтому являлось выполнение следующей последовательности операторов геометрического моделирования: восстановление несущих поверхностей вращения для линий тока вместе с их образующими; аппроксимация точно заданных линий тока нерациональными В-кривыми; скиннинг лопатки со сглаживанием [11].

В случае, если исходные данные представлены неструктурированной поверхностной сеткой, применяется интерактивный подход к реконструкции, состоящий в следующем. Пользователь формирует стопку секущих плоскостей, пересечение которых с полигональной моделью дает точечное представление опорных кривых. Для упорядочивания точек сечений задействуется алгоритм построения огибающей по k ближайшим соседям [52]. Сечения аппроксимируются с заданной точностью одной или несколькими (в зависимости от того, сегментирован ли набор точек) В-кривыми, после чего данные кривые повторно интерполируются для получения синхронной параметризации на едином узловом векторе. Завершающий этап, как и в случае с автоматической реконструкцией, состоит в применении оператора скиннинга со сглаживанием.

3.5.2 Реконструкция объектов свободной формы

Ключевым этапом реконструкции объекта свободной формы является сегментация его полигонального представления. Разработанный программный комплекс позволяет выполнить сегментацию в интерактивном и автоматическом режимах. Для построения четырехугольной сетки используется программа Instant Meshes [49].

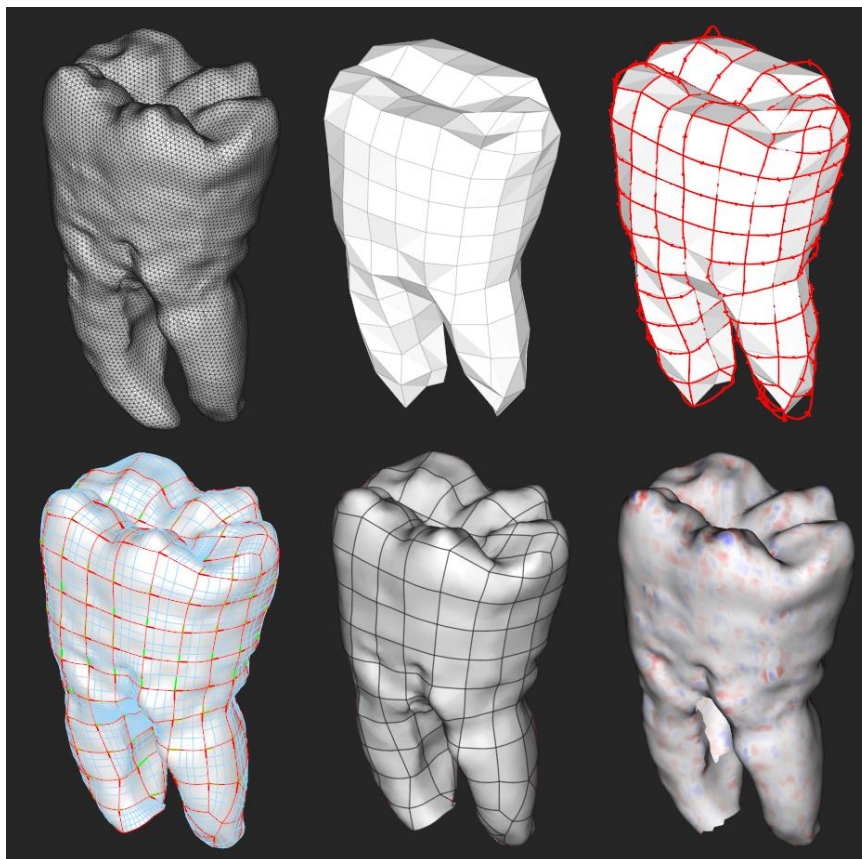


Рисунок 36 – Этапы автоматической реконструкции. Верхний ряд (слева-направо): исходная сетка, ретопология четырехугольными элементами, соответствующая сеть кривых. Нижний ряд (слева-направо): восстановленные порции поверхностей, граничная модель, поле геометрических невязок.

Программный комплекс позволяет строить грани модели в виде порций Кунса, либо как аппроксимирующие В-поверхности общего вида. Итоговая геометрическая невязка зависит от следующих факторов:

- 1) Гранулярность сети кривых.
- 2) Точность аппроксимации ребер.
- 3) Точность аппроксимации В-поверхностей (зависит от количества внутренних узлов).

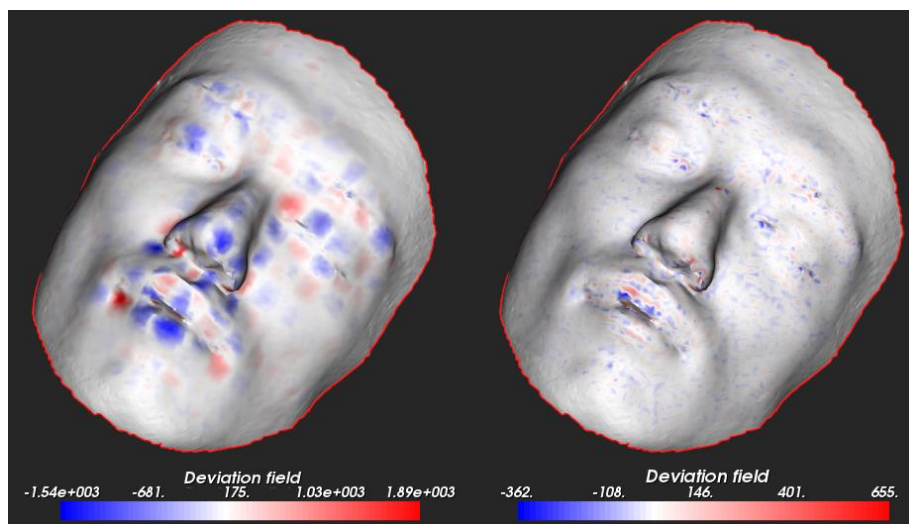


Рисунок 37 – Сравнение точности реконструкции порциями Кунса (слева) и аппроксимирующими В-поверхностями (справа).

Реконструкция порциями Кунса дает «грубый» результат, точность которого полностью определяется начальной сетью кривых. В свою очередь, реконструкция аппроксимирующими В-поверхностями позволяет достичь требуемой точности путем добавления внутренних узлов в базисные сплайны поверхностей и варьированием параметра сглаживания λ (Рис. 37).

Заключение

В результате настоящего исследования разработаны теоретические основы создания программных средств реконструкции САД-моделей в граничном представлении. Представлена программная реализация интерактивного и автоматического реинжиниринга с использованием графа параметрических зависимостей. Узлами графа являются операторы геометрического моделирования, в том числе:

- 1) Автоматическая ретопология модели путем проецирования прямоугольной сетки на исходную триангуляцию. Кроме того, даны средства интерактивной ретопологии.
- 2) Построение поверхности, заполняющей ячейку сети кривых, путем уточнения порции Кунса до аппроксимирующей В-поверхности. Уточнение состоит в выборе внутренних узлов региона триангуляции с последующей оптимизацией контрольных точек поверхности при фиксированном базисе.
- 3) Адаптация сети кривых и В-поверхностей для обеспечения гладкости стыков порядка G^1 .
- 4) Проверка критериев качества реконструкции, включающая построение карты геометрических невязок, а также исследование дифференциальных свойств кривых и поверхностей.

Представленные программные средства доступны в исходных кодах публично.

Литература

1. Geng Z. and Bidanda B. Review of reverse engineering systems – current state of the art. *Virtual and Physical Prototyping*, vol. 12, no. 2, 2017, pp. 161–172.
2. Requicha A.G. Representations for Rigid Solids: Theory, Methods, and Systems. *ACM Computing Surveys* vol. 12, no. 4, 1980, pp. 437–464.
3. Sederberg T.W. and Parry S.R. Free-form deformation of solid geometric models. In *Proc. of the 13th annual conference on Computer graphics and interactive techniques (SIGGRAPH'86)*, 1986, pp. 151–160.
4. Bradley C. and Currie B. Advances in the Field of Reverse Engineering. *Computer-Aided Design and Applications*, vol. 2, no. 5, 2005m pp. 697–706.
5. Gulánová J., Gulán L., Forrai M., and Hirz M. Generative engineering design methodology used for the development of surface-based components. *Computer-Aided Design and Applications* vol. 14, no. 5, 2017, pp. 642–649.
6. Francesco Buonamici, Monica Carfagni, Rocco Furferi, Lapo Governi, Alessandro Lapini, Yary Volpe. Reverse engineering modeling methods and tools : a survey. *Computer-Aided Design and Applications*, vol. 15, issue 3, 2018, pp. 443-464.
7. Hafer L. and Kirkpatrick A.E. Assessing open source software as a scholarly contribution. *Communications of the ACM*, vol. 52, no. 12, 2009, pp. 126-129.
8. Brown C.M. PADL-2: A Technical Summary. *IEEE Computer Graphics and Applications*, vol. 2, issue 2, 1982, pp. 69-84.
9. Varady T. Automatic Procedures to Create CAD Models from Measured Data. *Computer-Aided Design and Applications*, vol. 5, no. 5, 2008, pp. 577–588.
10. Mohaghegh K., Sadeghi M.H., and Abdullah A. Reverse engineering of turbine blades based on design intent. *The International Journal of Advanced Manufacturing Technology*, vol. 32, issue 9–10, 2007, pp. 1009–1020.
11. Сляднев С.Е., Турлапов В.Е. Реконструкция параметрической модели лопатки турбины из набора аэродинамических сечений с использованием техники сглаживания // Труды 28-й Международной конференции по компьютерной графике и машинному зрению (ГрафиКон-2018), 2018, стр. 495–499.
12. Du T., Inala J.P., Pu Y. et al. InverseCSG: automatic conversion of 3D models to CSG trees. *ACM Transactions on Graphics*, vol. 27, no. 4, 2018, pp. 1–16.
13. Сляднев С., Турлапов В. Метод декомпозиции машиностроительных твердотельных моделей на элементы объема изъятия // Труды 26-й Международной конференции по компьютерной графике, обработке

изображений и машинному зрению, системам визуализации и виртуального окружения (ГрафиКон-2016), 2016, стр. 58–63.

14. Ye X., Liu H., Chen L., Chen Z., Pan X., and Zhang S. Reverse innovative design – an integrated product design methodology. *CAD Computer Aided Design*, vol. 40, no. 7, 2008, pp. 812–827.
15. Harries S., Abt C., and Brenner M. Upfront CAD – Parametric Modeling Techniques for Shape Optimization. *Computational Methods in Applied Sciences*, vol. 48, 2019, pp. 191–211.
16. Catmull E. and Clark J. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, vol. 10, no. 6, 1978, pp. 350–355.
17. Antonelli M., Beccari C.V., Casciola G., Ciarloni R., and Morigi S. Subdivision surfaces integrated in a CAD system. *Computer Aided Design*, vol. 45, no. 11, 2013, pp. 1294–1305.
18. Sederberg T.W., Zheng J., Bakenov A., and Nasri A. T-splines and T-NURCCs. *ACM Transactions on Graphics*, vol. 22 issue 3, 2003, pp. 477–484.
19. Eck, M. and Hoppe, H. 1996. Automatic reconstruction of B-spline surfaces of arbitrary topological type. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*, ACM Press, 325–334.
20. Albat F., Müller R. 2005. Free-Form Surface Construction in a Commercial CAD/CAM System. In: Martin R., Bez H., Sabin M. (eds) *Mathematics of Surfaces XI. Lecture Notes in Computer Science*, vol 3604. Springer, Berlin, Heidelberg.
21. Andrews, J., Jin, H., and Séquin, C. 2012. Interactive Inverse 3D Modeling. *Computer-Aided Design and Applications* 9, 6, 881–900.
22. Brooks, F.P. 1996. The computer scientist as toolsmith II. *Communications of the ACM* 39, 3, 61–68.
23. Mobius, J. and Kobbelt, L. 2012. OpenFlipper: An open source geometry processing and rendering framework. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6920 LNCS, 488–500.
24. Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., and Ranzuglia, G. 2008. MeshLab: an Open-Source Mesh Processing Tool. *Sixth Eurographics Italian Chapter Conference*, 129–136.
25. Wang, J., Gu, D., Gao, Z., Yu, Z., Tan, C., and Zhou, L. 2013. Feature-Based Solid Model Reconstruction. *Journal of Computing and Information Science in Engineering* 13, 1, 011004.

26. Hirz, M., Rossbacher, P., and Gulánová, J. 2016. Future Trends in CAD – from the Perspective of Automotive Industry. CAD'16, CAD Solutions LLC, 734–741.
27. Kasik D.J., Buxton W., Ferguson D.R. Ten CAD challenges // IEEE Computer Graphics and Applications. 2005. (25). С. 81–92.
28. CAx Implementor Forum [Электронный ресурс]. Режим доступа: <https://www.cax-if.org/>.
29. Сляднев С. Open CASCADE Technology Overview [Электронный ресурс] 2014. Режим доступа: http://isicad.net/articles.php?article_num=17368.
30. Li M., Zhang Y.F., Fuh J.Y.H. Retrieving reusable 3D CAD models using knowledge-driven dependency graph partitioning // Computer-Aided Design and Applications. 2010. № 3 (7). С. 417–430.
31. Lin, F. and Hewitt, W.T. 1994. Expressing Coons-Gordon surfaces as NURBS. Computer-Aided Design 26, 2, 145–155.
32. Чарман, В. Introduction: The Max Creation Graph (MCG) [Электронный ресурс] 2016. Режим доступа: <https://www.augi.com/articles/detail/introduction-the-max-creation-graph-mcg>
33. Makhlof A. Ben [и др.]. Reconstruction of a CAD model from the deformed mesh using B-spline surfaces // International Journal of Computer Integrated Manufacturing. 2019. № 7 (32). С. 669–681.
34. Buonamici F. [и др.]. Reverse engineering of mechanical parts: A template-based approach // Journal of Computational Design and Engineering. 2018. № 2 (5). С. 145–159.
35. Zheng Y. [и др.]. A primitive-based 3D reconstruction method for remanufacturing // The International Journal of Advanced Manufacturing Technology. 2019. № 9–12 (103). С. 3667–3681.
36. Nowogrodzki A. How to support open-source software and stay sane // Nature. 2019. № 7763 (571). С. 133–134.
37. Šigut P., Vaněček P., Váša L. Analytic Surface Detection in CAD Exported Models SCITEPRESS - Science and Technology Publications, 2019. 278–285 с.
38. Coons S. Surfaces for computer-aided design of space forms. 1967.
39. Gordon W. An Operator Calculus for Surface and Volume Modeling // IEEE Computer Graphics and Applications. 1983. № 7 (3). С. 18–22.
40. Lin F., Hewitt W.T. Expressing Coons-Gordon surfaces as NURBS // Computer-Aided Design. 1994. № 2 (26). С. 145–155.
41. Piegl, L. and Tiller, W., The NURBS Book, New York: Springer, 1997.

42. Weiss V. [и др.]. Advanced surface fitting techniques // Computer Aided Geometric Design. 2002. (19). С. 19–42.
43. ProSTEP AG 3D Formats in the Field of Engineering. [Электронный ресурс] 2011. Режим доступа: http://isicad.ru/uploads/http___www.prostep.pdf
44. Selimovic I. Improved algorithms for the projection of points on NURBS curves and surfaces // Computer Aided Geometric Design. 2006. № 5 (23). С. 439–445.
45. Wolters H.J. Extensions : Extrapolation Methods for CAD. International Conference on Curves and Surfaces [4th], Saint-Malo, France, 1-7 July 1999. Proceedings, Volume 1. Curve and Surface Design.
46. Slyadnev, S., Active Data: SDK for organizing data of CAD/CAM/CAE software (powered by Open CASCADE Technology). Technical report. [Электронный ресурс] 2015. Режим доступа: https://www.researchgate.net/publication/282149692_Active_Data_SDK_for_organizing_data_of_CADCAMCAE_software_powered_by_Open_CASCADE_Technology.
47. Piegl, L. a. and Tiller, W. 2002. Surface skinning revisited. The Visual Computer 18, 4, 273–283.
48. Голованов Н.Н. Геометрическое моделирование, -М. :Изд-во курс Инфра-М, 2016.
49. Jakob, W., Tarini, M., Panozzo, D., and Sorkine-Hornung, O. (2015) Instant field-aligned meshes // ACM Transactions on Graphics 34, 6, 1–15.
50. Bergen G. van den Efficient Collision Detection of Complex Deformable Models using AABB Trees A K Peters/CRC Press, 2005. 131–144 с.
51. Kallay, M., Constrained optimization in surface design, Modeling in Computer Graphics, Berlin: Springer, 1993, pp. 85–93.
52. Moreira, A. and Santos, M.Y., Concave hull: A k-nearest neighbours approach for the computation of the region occupied by a set of points, Proc. 2nd Int. Conf. Computer Graphics Theory and Applications, 2007, pp. 61–68.
53. Rossoni M. [и др.]. Retopology and Simplification of Reality-based Models for Finite Element Analysis // Computer-Aided Design and Applications. 2019. № 3 (17). С. 525–546.