

**Министерство образования и науки Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Институт кибернетики

Направление подготовки - 09.04.01 Информатика и вычислительная техника  
Кафедра информатики и проектирования систем

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

Тема работы
<b>Разработка методов визуализации сложных 3D моделей</b>

УДК 004.925.84

Студент

Группа	ФИО	Подпись	Дата
8ВМ4Б	Хасаева Диана Зуфаровна		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
и.о. руководителя каф. ИПС, доцент каф. ИПС	Демин Антон Юрьевич	канд. техн. наук		

**КОНСУЛЬТАНТЫ:**

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент каф. менеджмента	Конотопский Владимир Юрьевич	канд. экон. наук		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
ассистент каф. ЭБЖ	Акулов Петр Анатольевич	-		

**ДОПУСТИТЬ К ЗАЩИТЕ:**

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
и.о. руководителя каф. ИПС	Демин Антон Юрьевич	канд. техн. наук		

Томск – 2016 г.

## ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

Код результата	Результат обучения (выпускник должен быть готов)
<b>Профессиональные компетенции</b>	
P1	Использовать знания в области естественных и инженерных наук для разработки новых программных продуктов
P2	Применять глубокие знания в области современных технологий разработки программного обеспечения для решения междисциплинарных инженерных задач и проектировании программного обеспечения
P3	Применять глубокие знания в области проектирования, разработки и поддержки современных распределенных автоматизированных, микропроцессорных, облачных и информационно-телекоммуникационных систем
P4	Осуществлять постановку и решение инженерных инновационных задач, связанных с разработкой программных продуктов, с применением системного анализа, моделирования объектов и численных методов оптимизации процессов программного обеспечения
P5	Применять знания в области разработки и функционирования вычислительных и интеллектуальных систем при разработке программного обеспечения, конкурентоспособного на мировом рынке
P6	Применять знания в области защиты информации для сохранения информационной безопасности при разработке, эксплуатации и поддержке программных продуктов и систем
P7	Профессионально владеть иностранным языком на уровне, необходимом для осуществления профессиональной деятельности в иноязычной среде, разрабатывать программное сопровождение, представлять и объяснять результаты профессиональной инновационной работы.
P8	Применять знания в области решения философских и методологических проблем науки и техники, современных проблем информатики и вычислительной техники при исследовательской деятельности и разработке программного обеспечения
P9	Внедрять, эксплуатировать и поддерживать современные программные продукты, обладать навыками сохранения и повышения их эффективности, выполнять правила и нормы социальной ответственности на рабочем месте
<b>Универсальные компетенции</b>	
P11	Применять знания и умения в области проектирования, менеджмента инновационных инженерных проектов, с использованием и соблюдением правил защиты интеллектуальной собственности
P12	Показывать высокий уровень квалификации в работе как, так и в составе или руководстве группой, уметь работать в команде со специалистами разных профессий, нести ответственность за профессиональную деятельность и соблюдать корпоративный имидж компании
P13	Уметь применять социальные и культурно-нравственные аспекты в разработке инженерных инновационных проектов
P14	Осуществлять периодически повышение квалификации в области профессиональных компетенций, следить за новшествами и нововведениями в области профессиональных интересов

**Министерство образования и науки Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Институт кибернетики

Направление подготовки (специальность) - 09.04.01 Информатика и вычислительная техника  
Кафедра информатики и проектирования систем

УТВЕРЖДАЮ:  
Зав. кафедрой

\_\_\_\_\_  
(Подпись)      (Дата)      (Ф.И.О.)

**ЗАДАНИЕ**

**на выполнение выпускной квалификационной работы**

В форме:

магистерской диссертации

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8ВМ4Б	Хасаевой Диане Зуфаровне

Тема работы:

Разработка методов визуализации сложных 3D моделей

Утверждена приказом директора (дата, номер)

№ 780/с, от 05.02.2016 г.

Срок сдачи студентом выполненной работы:

11.06.2016 г.

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

<b>Исходные данные к работе</b>	Объект исследования – методы визуализации сложных 3D моделей. Объект проектирования – метод марширующих кубов. Периодический режим работы. Вид сырья - программное обеспечение. Требования к продукту: визуализация сложных 3d моделей с помощью метода марширующих кубов. Среда разработки программного обеспечения: Microsoft Visual Studio 2013, платформа Windows Presentation Foundation. Язык программирования: объектно-ориентированный C#, расширяемый язык разметки для приложений XAML.
---------------------------------	---

<p><b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b></p>	<p>Анализ литературы с целью изучения достижений компьютерной графики в области разработки методов визуализации 3d объектов. Исследование ячеечных методов визуализации 3D объектов: марширующих кубов, Канейро, Скалы. Реализация метода марширующих кубов на платформе WPF.</p> <p>Анализ результатов работы метода марширующих кубов. Сравнительный анализ исследованных алгоритмов.</p> <p>Проведение экономического анализа результатов исследования и эффективности работы алгоритма.</p> <p>Выявление особенностей социальной ответственности функционирования программного продукта.</p>
<p><b>Перечень графического материала</b></p>	<p>Схемы работы алгоритмов, иллюстрации этапов выполнения методов, результаты работы метода марширующих кубов.</p>
<p><b>Консультанты по разделам выпускной квалификационной работы</b></p>	
<p><b>Раздел</b></p>	<p><b>Консультант</b></p>
<p><b>Обзор литературы</b></p> <p><b>Исследование алгоритмов визуализации трехмерных объектов</b></p> <p><b>Реализация алгоритма марширующих кубов на платформе WPF</b></p> <p><b>Анализ результатов</b></p>	<p>Демин Антон Юрьевич</p>
<p><b>Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</b></p>	<p>Конотопский Владимир Юрьевич</p>
<p><b>Социальная ответственность</b></p>	<p>Акулов Петр Анатольевич</p>
<p><b>Раздел ВКР, выполненный на иностранном языке</b></p>	<p>Демин Антон Юрьевич Сидоренко Татьяна Валерьевна</p>
<p><b>Названия разделов, которые должны быть написаны на русском и иностранном языках:</b></p>	
<p><b>1. Обзор литературы</b></p>	
<p><b>2. Исследование алгоритмов визуализации трехмерных объектов</b></p>	
<p><b>3. Реализация алгоритма марширующих кубов на платформе WPF</b></p>	
<p><b>4. Анализ результатов</b></p>	
<p><b>5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</b></p>	
<p><b>6. Социальная ответственность</b></p>	
<p><b>2.1. Visualization of three-dimensional objects with the help of triangulation algorithms for</b></p>	

**defined surface**  
**2.2. Methods for solving a triangulation problem**

<b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b>	19.02.2016
---	------------

**Задание выдал руководитель:**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
и.о. руководителя каф. ИПС, доцент каф. ИПС	Демин Антон Юрьевич	канд. техн. наук		

**Задание принял к исполнению студент:**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8ВМ4Б	Хасаева Диана Зуфаровна		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И  
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

<b>Группа</b>	<b>ФИО</b>
8ВМ4Б	Хасаевой Диане Зуфаровне

<b>Институт</b>	кибернетики	<b>Кафедра</b>	Информатики и проектирования систем
<b>Уровень образования</b>	магистр	<b>Направление/специальность</b>	09.04.01 Информатика и вычислительная техника

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	
2. Нормы и нормативы расходования ресурсов	
3. Ставки НДС и социального налога	

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

1. Формирование плана и графика разработки и внедрения ИР	
2. Обоснование необходимых инвестиций для разработки и внедрения ИР	
3. Оценка ресурсной, финансовой, социальной, бюджетной эффективности ИР и потенциальных рисков	

**Перечень графического материала** (с точным указанием обязательных чертежей)

1. График разработки и внедрения ИР
2. Инвестиционный план. Бюджет ИП

<b>Дата выдачи задания для раздела по линейному графику</b>	01.05.2016
---	------------

**Задание выдал консультант:**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
доцент каф. менеджмента	Конотопский Владимир Юрьевич	канд. экон. наук		

**Задание принял к исполнению студент:**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8ВМ4Б	Хасаева Диана Зуфаровна		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»**

Студенту:

<b>Группа</b>	<b>ФИО</b>
8ВМ4Б	Хасаевой Диане Зуфаровне

<b>Институт</b>	кибернетики	<b>Кафедра</b>	Информатики и проектирования систем
<b>Уровень образования</b>	магистр	<b>Направление/специальность</b>	09.04.01 Информатика и вычислительная техника

**Исходные данные к разделу «Социальная ответственность»:**

1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения

*1. Алгоритм маршрутирующих кубов.  
Область применения: компьютерная графика.  
Сферы применения:  
- научные исследования;  
- медицина.*

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

**1. Производственная безопасность**

1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения  
1.2. Анализ выявленных опасных факторов при эксплуатации проектируемого решения

*1. Анализ вредных и опасных факторов труда инженера-программиста, и разработка мер защиты от них, оценка условий труда и микроклимата рабочей среды, создание оптимальных условий труда, техники безопасности и пожарной профилактики. В качестве исходных данных использованы параметры рабочего помещения, в котором производилась разработка и условия труда при работе с персональным компьютером.*

- 1.1 Анализ вредных факторов:*
- Микроклимат;
  - Производственный шум и вибрации;
  - Исследование освещенности;
  - Электромагнитные волны;
  - Ионизирующее излучение;
  - Психофизиологическое воздействие.
- 1.2 Анализ опасных факторов:*
- Механическая опасность;
  - Термическая опасность;
  - Электробезопасность;
  - Пожарная безопасность.

**2. Экологическая безопасность:**

- защита селитебной зоны
- анализ воздействия объекта на атмосферу (выбросы);
- анализ воздействия объекта на гидросферу (сбросы);

*2. Выявление видов загрязнений и энергетических выбросов (шумы, вибрации, излучения) наносимых окружающей среде в процессе разработки программного обеспечения в помещении офиса. Утилизация*

<ul style="list-style-type: none"> <li>– анализ воздействия объекта на литосферу (отходы);</li> <li>– разработать решения по обеспечению экологической безопасности со ссылками на НТД по охране окружающей среды.</li> </ul>	<p><i>аккумуляторных батарей.</i></p>
<p><b>3. Безопасность в чрезвычайных ситуациях:</b></p> <ul style="list-style-type: none"> <li>– перечень возможных ЧС при разработке и эксплуатации проектируемого решения;</li> <li>– выбор наиболее типичной ЧС;</li> <li>– разработка превентивных мер по предупреждению ЧС;</li> <li>– разработка действий в результате возникшей ЧС и мер по ликвидации её последствий.</li> </ul>	<p><i>3. Анализ возможных чрезвычайных ситуаций, которые могут возникнуть в рабочем помещении офиса и установка общих правил поведения и рекомендаций.</i></p>
<p><b>4. Правовые и организационные вопросы обеспечения безопасности:</b></p> <p>4.1. Специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны).</p> <p>4.2. Правовые нормы трудового законодательства.</p>	<p><i>4. Основные проводимые правовые и организационные мероприятия по обеспечению безопасности, трудящихся в офисном помещении. Ответственность за ошибки.</i></p>

<b>Дата выдачи задания для раздела по линейному графику</b>	<b>15.05.2016</b>
---	-------------------

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
ассистент каф. ЭБЖ	Акулов Петр Анатольевич	-		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8ВМ4Б	Хасаева Диана Зуфаровна		

**Министерство образования и науки Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Институт кибернетики

Направление подготовки (специальность) - 09.04.01 Информатика и вычислительная техника

Уровень образования - магистр

Кафедра информатики и проектирования систем

Период выполнения - осенний / весенний семестр 2015/2016 учебного года

Форма представления работы:

магистерская диссертация

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН  
выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы:

11.06.2016

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
14.03.2016	<i>Обзор литературы</i>	
26.03.2016	<i>Исследование алгоритмов визуализации трехмерных объектов</i>	
16.04.2016	<i>Реализация алгоритма маршрутирующих кубов на платформе WPF</i>	
30.04.2016	<i>Анализ результатов</i>	
14.05.2016	<i>Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</i>	
28.05.2016	<i>Социальная ответственность</i>	
11.06.2016	<i>Visualization of three-dimensional objects with help of triangulation algorithms for defined surface Methods for solving the problem of triangulation</i>	

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
и.о. руководителя, доцент каф. ИПС	Демин Антон Юрьевич	канд. техн. наук		

**СОГЛАСОВАНО:**

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
и.о. руководителя каф. ИПС	Демин Антон Юрьевич	канд. техн. наук		

## РЕФЕРАТ

Выпускная квалификационная работа включает 180 с., 45 рис., 19 табл., 35 источников, 2 прил.

Ключевые слова: ячеечные методы визуализации трехмерных объектов, метод марширующих кубов, метод Канейро, метод Скалы, WPF

Объектом исследования являются ячеечные методы визуализации трехмерных объектов: метод марширующих кубов, метод Канейро, метод Скалы.

Цель работы – исследование методов: марширующих кубов, Канейро, Скалы, реализация метода марширующих кубов на платформе WPF, сравнительный анализ данных алгоритмов.

В процессе исследования проводились эксперименты по визуализации трехмерных объектов методом марширующих кубов на платформе WPF.

В результате исследования было изучено три ячеечных алгоритма визуализации объектов, реализован метод марширующих кубов на платформе WPF, приведен сравнительный анализ алгоритмов.

Основные технико-эксплуатационные характеристики: визуализация метода марширующих кубов может производиться на компьютере, работающем под операционной системой семейства Windows с платформой .NET Framework не ниже 4.0.

Степень внедрения: по мере необходимости для заинтересованной стороны.

Области применения: компьютерная графика, визуализация экспериментальных данных, функциональное представление, медицина.

Экономическая эффективность/значимость работы: данное исследование носит исследовательский характер и не несет за собой прямой экономической выгоды. Но результаты данной работы можно применять в дальнейших исследованиях в рамках данной тематики и областей применения.

Основными перспективами развития исследования на будущее являются: разработка методов Канейро и Скалы на платформе WPF; модификация алгоритмов марширующих кубов, Канейро и Скалы для оптимизации процесса визуализации трехмерных объектов.

Выпускная квалификационная работа выполнена в соответствии с требованиями стандарта ТПУ (СТО ТПУ 2.5.01-2006) в текстовом редакторе Microsoft Word 2010.

## Определения, обозначения, сокращения, нормативные ссылки

Данная работа содержит термины и определения:

**Алгоритм** — набор инструкций, описывающих порядок действий исполнителя для достижения некоторого результата.

**Алгоритм «Марширующие кубы»** (англ. **Marching cubes Algorithm**, в пер. с англ. — «алгоритм шагающих кубиков») — ячеечный алгоритм в компьютерной графике, созданный для обработки полигональной сетки изоповерхности трехмерного скалярного поля (чаще называемой сеткой вокселей).

**Алгоритм Канейро** – ячеечный алгоритм компьютерной графики, в основе которого лежит разбиение пространства на определенное множество ячеек - треугольных пирамид, определение ячеек, которые пересекаются с заданным объектом и аппроксимация поверхности в выявленных ячейках.

**Алгоритм Скалы** - ячеечный метод, визуализирующий скалярные 3D поля, явно заданные функцией.

**Воксел** (в разговорной речи *воксель*, англ. *Voxel* — образовано из слов: объёмный (англ. *volumetric*) и пиксел (англ. *pixel*)) — элемент объёмного изображения, содержащий значение элемента растра в трёхмерном пространстве, является аналогом двумерного пиксела для трёхмерного пространства.

**Полигон** - многоугольник, минимальная поверхность для визуализации в трёхмерной графике.

**Полигональная сетка** - (англ. *polygon mesh* или сокр. *mesh*) — это набор вершин, ребер и граней, определяющих поверхность объемной фигуры в компьютерной 3D графике и пространственно-объемном моделировании.

**Триангуляция** – это разбиение геометрического объекта на симплексы. Например, на плоскости это разбиение на треугольники.

**Кэш** или **кеш** (англ. *cache*, от фр. *cacher* — «прятать»); произносится [кæʃ] — «кэш») — промежуточный буфер с быстрым доступом, содержащий информацию, которая может быть запрошена с наибольшей вероятностью.

**Сверхбольшая поверхность** — это поверхность, которая не может уместить свои исходные данные в памяти компьютера.

**Скалярное поле** — это функция, отображающая в пространство

В данной работе использованы ссылки на документы:

Федеральный закон от 22.07.2008 N 123-ФЗ Технический регламент о требованиях пожарной безопасности.

В текущей работе использованы следующие обозначения и сокращения:

**WPF (Widows Presentation Foundation)** — платформа для создания настольных и веб-браузерных приложений с широкими возможностями для двумерной и трехмерной компьютерной графики, входящая в состав платформы .NET Framework и поддерживающая язык C# и XAML.

**C#** (произносится «си шарп») — объектно-ориентированный язык программирования, разработанный компанией Microsoft как язык разработки приложений для платформы Microsoft .NET Framework и впоследствии стандартизированный как ECMA-334 и ISO/IEC 23270.

**XAML - (eXtensible Application Markup Language)** — расширяемый язык разметки для приложений (произносится [замл] или [зэмл]), основанный на XML язык разметки для декларативного программирования приложений, разработанный компанией Microsoft.

**2D (2-D)** — аббревиатура (от англ. *2-dimensional*), которая может означать: что-либо, имеющее два измерения, двумерное пространство, двумерная графика (2D).

**3D (3-D)** (от англ. *3-dimensional*) — аббревиатура, которая может означать: что-либо, имеющее три измерения, трёхмерное пространство, трёхмерная графика.

**ЭВМ** — электронно-вычислительная машина.

**ПК** – персональный компьютер

**НР** – научный руководитель.

**И** – инженер.

**ЕСН** - единый социальный налог.

**НДС** – налог на добавленную стоимость.

**ОУ** – огнетушитель углекислый.

**ЧС** – чрезвычайные ситуации.

## Оглавление

Определения, обозначения, сокращения, нормативные ссылки.....	11
Оглавление.....	14
Введение .....	17
1. Обзор литературы.....	20
2. Исследование алгоритмов визуализации трехмерных объектов	<b>Ошибка! Закладка не определена.</b>
2.1. Визуализация трехмерных объектов с использованием алгоритмов триангуляции заданной поверхности.....	<b>Ошибка! Закладка не определена.</b>
2.2. Методы решения задачи триангуляции	<b>Ошибка! Закладка не определена.</b>
2.2.1. Алгоритмы построения триангуляции Делоне	<b>Ошибка! Закладка не определена.</b>
2.2.1.1. Итеративные алгоритмы .....	<b>Ошибка! Закладка не определена.</b>
2.2.1.2. Простой итеративный алгоритм ....	<b>Ошибка! Закладка не определена.</b>
2.2.1.3. Алгоритмы с кэшированием поиска треугольников	<b>Ошибка! Закладка не определена.</b>
2.2.1.4. Итеративный алгоритм со статическим кэшированием поиска	<b>Ошибка! Закладка не определена.</b>
2.2.1.5. Итеративный алгоритм с динамическим кэшированием поиска	<b>Ошибка! Закладка не определена.</b>
2.2.1.6. Итеративный алгоритм с послойным сгущением	<b>Ошибка! Закладка не определена.</b>
2.2.2. Ячеечные алгоритмы триангуляции	<b>Ошибка! Закладка не определена.</b>
2.2.2.1. Алгоритмы: «марширующие кубы», Канейро и Скалы	<b>Ошибка! Закладка не определена.</b>
2.2.2.2. Алгоритм «Марширующие тетраэдры 6»	<b>Ошибка! Закладка не определена.</b>
2.2.2.3. Алгоритм «марширующие призмы»	<b>Ошибка! Закладка не определена.</b>
2.2.2.4. Триангуляция пятивершинными пирамидами	<b>Ошибка! Закладка не определена.</b>
2.2.2.5. Алгоритм «Марширующие октаэдры»	<b>Ошибка! Закладка не определена.</b>
2.2.3. Триангуляция сверхбольших поверхностей	<b>Ошибка! Закладка не определена.</b>
2.2.4. Адаптивная триангуляция.....	<b>Ошибка! Закладка не определена.</b>
2.2.5. Алгоритм Финча и Бишопа .....	<b>Ошибка! Закладка не определена.</b>
2.2.6. Современные направления исследований и разработок	<b>Ошибка! Закладка не определена.</b>
2.3. Метод марширующих кубов.....	<b>Ошибка! Закладка не определена.</b>
2.4. Метод Канейро .....	<b>Ошибка! Закладка не определена.</b>
2.5. Метод Скалы.....	<b>Ошибка! Закладка не определена.</b>
3. Реализация алгоритма марширующих кубов на платформе WPF	<b>Ошибка! Закладка не определена.</b>

4.	Анализ результатов.....	<b>Ошибка! Закладка не определена.</b>
4.1.	Результаты разработки .....	<b>Ошибка! Закладка не определена.</b>
4.2.	Сравнительный анализ методов .....	<b>Ошибка! Закладка не определена.</b>
5.	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.	31
5.1	Организация и планирование работ .....	31
5.1.1	Продолжительность этапов работ .....	33
5.1.2	Расчет накопления готовности проекта.....	38
5.2	Расчет сметы затрат на выполнение проекта .....	39
5.2.1	Расчет затрат на материалы.....	40
5.2.2	Расчет заработной платы.....	41
5.2.3	Расчет затрат на социальный налог .....	42
5.2.4	Расчет затрат на электроэнергию .....	42
5.2.5	Расчет амортизационных расходов .....	44
5.2.6	Расчет прочих расходов.....	45
5.2.7	Расчет общей себестоимости разработки.....	45
5.2.8	Расчет прибыли .....	46
5.2.9	Расчет НДС.....	46
5.2.10	Цена разработки НИР .....	46
5.3	Оценка экономической эффективности проекта.....	46
5.3.1	Оценка научно-технического уровня НИР .....	47
6.	Социальная ответственность .....	<b>Ошибка! Закладка не определена.</b>
6.1.	Производственная безопасность .....	<b>Ошибка! Закладка не определена.</b>
6.1.1	Анализ вредных факторов проектируемой производственной среды	<b>Ошибка!</b>
6.1.2	Анализ опасных факторов производственной среды	<b>Ошибка! Закладка не о</b>
6.2.	Экологическая безопасность .....	<b>Ошибка! Закладка не определена.</b>
6.3.	Безопасность в чрезвычайных ситуациях	<b>Ошибка! Закладка не определена.</b>

- 6.4. Правовые и организационные вопросы обеспечения безопасности **Ошибка! З**
- 6.4.1 Требования к организации рабочих мест пользователей ПЭВМ **Ошибка! За**
- 6.4.2 Режим труда и отдыха при работе с компьютером **Ошибка! Закладка не оп**
- 6.4.3 Медико-профилактические и оздоровительные мероприятия **Ошибка! Закл**
- 6.4.4 Организация труда на предприятии **Ошибка! Закладка не определена.**

Заключение .....	<b>Ошибка! Закладка не определена.</b>
Список публикаций .....	51
Список использованных источников.....	<b>Ошибка! Закладка не определена.</b>
Приложение А .....	52
Приложение Б.....	<b>Ошибка! Закладка не определена.</b>

## Введение

Моделирование и реалистичная визуализация поверхностей является ключевой составляющей многих приложений разной направленности. И основным требованием к подсистеме визуализации подобных приложений является достижение реалистичности виртуальной поверхности (или рельефа) при сохранении интерактивной частоты отображения.

Основные **проблемы**, лежащие в основе данного исследования:

1. Отсутствие готовых объемных исследовательских работ по изучению основных алгоритмов визуализации трехмерных объектов.
2. Отсутствие реализации алгоритма марширующих кубов на платформе Windows Presentation Foundation.

**Актуальность** данной работы обуславливает не только решение данных проблем, но и важность реализации и применения данных алгоритмов в компьютерной графике. В настоящее время практически ни одна сфера научных исследований, техники, да и просто ни одна сфера жизни не обходятся без визуализации данных. Достижения компьютерной графики применяются повсюду и внедряются во все сферы жизнедеятельности. Программные продукты используются как в прямом виде, так и в качестве средств для создания других программ, систем, комплексов. Особо велика значимость качественной, точной и быстрой визуализации данных в таких областях, как:

1. Научные исследования: физика (визуализация экспериментальных данных и др.), математика (функциональное представление, визуализация трехмерных скалярных полей, поверхностей и т.д.) и другие.
2. Медицина. Эта область просто нуждается в применении, развитии существующих и создании новых алгоритмов визуализации трехмерных данных, объемов, фигур и объектов. Построение трехмерных объектов, в особенности визуализация по исходным двумерным данным помогает медицине обследовать состояние здоровья человека и других живых существ, предотвращать и лечить

многие болезни, заболевания, проблемы здоровья. С помощью алгоритмов компьютерной графики можно визуализировать органы человека: легкие, мозг, нервы и т.д.

В соответствии с этим, целью данного исследования является:

1. Изучение основных существующих алгоритмов визуализации трехмерных объектов.
2. Реализация алгоритма марширующих кубов на платформе Windows Presentation Foundation.

Для достижения данной цели необходимо выполнить следующие задачи:

1. изучить предметную область, исследовать литературу по данной тематике;
2. провести обзор существующих основных алгоритмов визуализации неявно заданной поверхности;
3. реализовать алгоритм марширующих кубов на платформе Windows Presentation Foundation;
4. визуализировать с помощью реализованного алгоритма пример части МРТ, изначально заданной неявно двумерным изображением, и другие примеры;
5. визуализировать с помощью реализованного алгоритма и другие примеры;
6. изучить, протестировать и представить результаты разработки проекта;
7. привести сравнительный анализ некоторых существующих алгоритмов, а именно: «Марширующие кубы», Канейро, Скалы.
8. дать оценку экономической эффективности проекта;
9. изучить нормы и особенности социальной ответственности при разработке проекта;
10. сделать выводы и заключение.

Научно-практическая значимость работы в особенности подтверждается значимостью той области исследований, в которой она не только применима, но и попросту необходимо (медицина).

Для реализации метода были изучены различные версии и модификации алгоритма марширующих кубов, выполненные на различных платформах различными разработчиками и в разное время. Прототипом для нынешней разработки стал проект Паула Борке, который в 1994 году выложил свою реализацию метода в свою электронную библиотеку. Данная программа была реализована на языке C++ как консольное приложение с поддержкой Unix-лексики. Паул визуализировал часть данных MPT, изначально заданных в качестве двумерного изображения диаграммы MPT.

Несмотря на то, что алгоритм марширующих кубов впервые был предложен в 1987 году на конференции SIGGRAPH Вильямом Лоренсенем и Харви Клайном, наличие версий его реализации и модификаций, разработанных в самое разное время, подтверждает значимость и важность существования алгоритма. Сейчас алгоритм находится в свободном доступе и не носит на себе никаких ограничений по возможности его использования, реализации и модификации.

Данная разработка алгоритма марширующих кубов не только имеет положительный эффект от применения, но и дает широкие перспективы для своего дальнейшего развития, усовершенствования и возможности применения в качестве средства для создания новых программных продуктов.

Основные недостатки, выявленные в ходе работы:

1. нехватка времени на реализацию и других алгоритмов визуализации неявно заданных поверхностей;
2. нехватка времени на модификацию алгоритма марширующих кубов с целью повышения его производительности и улучшения критериев оптимизации.

## 1. Обзор литературы

Компьютерная графика - это особая область информатики, отвечающая за представление 2D, 3D – изображений, видео и анимации средствами ПК. Зародилась компьютерная графика в 1950-х годах. Сначала с помощью нее можно было выводить лишь несколько десятков отрезков на экран. Современные средства и инструменты компьютерной графики дают возможность создания реалистичных изображений, по качеству не уступающим качественным фотографиям, и не только. Существуют различные аппаратные и программные средства, обеспечивающие создание самых различных видов изображений, начиная от несложных конструкторских чертежей и заканчивая высоко реалистичными моделями реальных объектов.

Области применения компьютерной графики бесконечно широки, можно сказать, она используется практически во всех областях науки и техники, также является необходимой составляющей подавляющего большинства научных и инженерных дисциплин при визуализации воспринимаемых образов и придания облику получаемых-передаваемых информационных знаний. В любой сфере деятельности и профессиональной среде существует необходимость представлять, презентовать свои и не только информационные данные, итоги работы, отчеты на командных заседаниях. Особую значимость возможности компьютерной графики имеют в таких областях, как: медицина (компьютерная томография), геодезия, картографии, геофизике, ядерной физике. Самой прибыльной сферой применения компьютерной графики является сфера рекламных услуг, телевидения и других отраслей индустрии развлечений, использующих анимационные возможности компьютерной графики (фильмы, компьютерные игры). Особую значимость возможности компьютерной графики несут для разработки и использования компьютерного моделирования, с помощью которого обучаются пилоты, летчики, космонавты и представителей других профессий (тренажеры). Без

знаний основ компьютерной графики не могут осуществлять свои исследования и профессиональную деятельность ни инженер, ни ученый.

Итоговый результат применения средств компьютерной графики - это изображение, используемое для самых различных целей. Схемы, карты, графические обозначения возникли еще в глубокие времена, т.к. лучше всего большинство людей усваивают информацию через органы зрения.

На данный момент компьютерная графика является достаточно сложной, основательно проработанной и многоаспектной научно-технической дисциплиной. Самые ее востребованные модули, например, связанные с геометрическими преобразованиями, способами описания кривых и поверхностей, к настоящему времени изучены довольно обширно. Перечень областей и сфер постоянно растет и развивается: это и методы растрового сканирования, и способы удаления невидимых линий и поверхностей, задание цвета, освещенности, эффектов теней, прозрачности и полупрозрачности, текстурирование и др.

Область использования применения компьютерной графики можно разделить на 4 основные сферы.

### **1. Представление информации**

Компьютерная графика решает проблему представления различных объемов самых различных данных (например, спектроскопических данных, данных результатов исследований, данных широкомасштабных социологических опросов).

Практически ни одна область современной науки не обходится без графического представления информации. Кроме научного применения компьютерной графики для решения исследовательских задач, компьютерная графика используется и для математического моделирования процессов и явлений, которые не могут обходиться без графического отображения. Например, процессы, протекающие в атмосфере или океане, наглядно изображаются с помощью визуализации различных картин течений или полей температуры. 3D геометрия пластов, расположенных на значительной глубине

литосферы в геологии может быть представлена с помощью компьютерной графики.

Самая значимая область применения компьютерной графики – медицина, позволяет применять методы диагностики, основанные на представлении 3D визуализации внутренних органов человека. Магнитно-резонансная томография (в частности, например, ультразвуковое исследование) дает возможность получить трехмерную информацию, которая затем подвергается математической обработке и выводится на экран. Кроме того широко используется и 2D графика: энцефалограммы, миограммы, кардиограммы, осуществляющие вывод на графопостроитель или экран компьютера.

## **2. Проектирование**

Критически важно применение компьютерной графики в строительстве, отображении чертежей архитектурных предметов, объектов и сооружений, а также изделий. Особую сложность и важность представляет то, что процесс проектирования вынужденно носит итеративный характер, т.к. конструктор испробует множество вариантов и выбирает самый оптимальный по определенному перечню параметров. Вносит свой вклад в данную сторону проектирования частое отсутствие первоначальных четких требований заказчика, который не всегда сразу и конкретно представляет итоговый результат и его технические характеристики. Конструирование предварительных макетов, особенно на многих этапах работы, - достаточно долгое и дорогое дело. В настоящее время существуют развитые мощные программные средства автоматизации проектно-конструкторских работ (САПР), с помощью которых возможно быстро создавать чертежи объектов, осуществлять различные расчеты и т.д. Используя данные возможности, можно детально представить макеты, проекции изделия, продукта средствами ПК, также рассмотреть его в объемном виде с различных сторон. Данные инструменты просто необходимы и дизайнерам интерьера, ландшафтным дизайнерам.

### **3. Моделирование**

Моделирование - это имитация различных ситуаций, которые возникают, к примеру, при полете вертолета или ракеты, передвижении автомобиля и т.д. На английском языке лучше всего характеризует данный процесс термин *simulation*. Сейчас моделирование применяется не только для проектирования и создания различных тренажеров. В маркетинге, телевидении, телевизионной рекламе, в научно-популярных и других жанров фильмах в настоящее время визуализируются движущиеся объекты, которые по визуальному представлению незначительно уступают полученным с помощью видеокамеры. Особое достижение компьютерная графика подарила киноиндустрии возможностью создания спецэффектов, немислимых даже в человеческом воображении в прежнее время. В последнее время набрала обороты популярность такая сфера применения компьютерной графики как создание виртуальной реальности.

### **4. Графический пользовательский интерфейс**

Графический интерфейс не всегда сопровождал системы графического вывода информации. Раньше информация, даже графическая, выводилась в аналого-цифровом виде или по отдельности. Сейчас практически все платформы программирования поддерживают возможность создания графического интерфейса, которая активно используется разработчиками. Лидером в данном аспекте являются разработки в области сети Internet. В настоящее время заняли передовое место в области разработки веб-технологий, различные веб-браузерные приложения, реализующие разнообразные средства общения в сети. Несмотря на то, что эти программы работают под управлением разных операционных сред, они реализуют, одни и те же функции, окна, баннеры, анимацию и т.п.

Современная компьютерная графика выделяет следующие основные направления: изобразительная компьютерная графика, обработка и анализ изображений, анализ сцен, компьютерная графика для научных абстракций.

Предметом **изобразительной компьютерной графики** являются синтезированные изображения. Базовые задачи, которые она решает:

- построение модели объекта и формирование изображения;
- преобразование модели и изображения;
- идентификация объекта и получение требуемой информации.

**Обработка и анализ изображений** преимущественно затрагивают цифровое представление изображений. Инструменты компьютерной графики используются для:

- увеличения качества изображения;
- оценки изображения;
- распознавания образов.

**Анализ сцен** определяет абстрактные модели графических объектов и взаимосвязи между ними. Объекты используются как синтезированные, так и выделенными на снимке. Одними из таких задач являются, к примеру, моделирование "машинного зрения" (роботы), анализ рентгеновских снимков с выделением и изучением интересующего объекта (внутреннего органа), разработка систем видеонаблюдения.

**Когнитивная компьютерная графика** – относительно новое направление, пока еще только развивающееся. В основе ее лежат идеи визуализации научных абстракций, которые могут поспособствовать рождению нового научного знания. Технической базой для нее являются мощные ЭВМ и высокопроизводительные средства визуализации [19].

**Трёхмерная графика** — раздел компьютерной графики, посвящённый методам создания изображений или видео путём моделирования объёмных объектов в трёхмерном пространстве.

**3D-моделирование** — это процесс создания 3D модели объекта. Задачей 3D-моделирования является разработка визуального объёмного образа необходимого объекта. Модель может являться прототипом реального объекта окружающего естественного мира (автомобили, комета, планеты и т.д.) или быть абсолютно абстрактной.

Графическое изображение трёхмерных объектов характеризуется построением геометрической проекции трёхмерной модели *сцены* на плоскость с помощью специальных программ. С появлением 3D-дисплеев и 3D-принтеров, трёхмерная графика не обязательно должна проецироваться на плоскость [19].

**Алгоритм** — набор инструкций, описывающих порядок действий исполнителя для достижения некоторого результата.

Алгоритмы создания реалистических изображений должны обеспечивать передачу свойств моделируемого объекта: объемность, расположение, передача полутонов, тени, освещение, текстуры поверхности и т.д. Генерация таких объемных изображений представляет собой очень сложную вычислительную задачу. В связи с этим на практике выполняют её декомпозицию: объекты разбиваются на составные части, и сложные изображения формируются из получаемых фрагментов. На практике, наиболее часто применяется разбиение изображений на треугольники. Это связано с тем, что треугольник - простейший полигон, вершины которого однозначно задают грань. Алгоритмы разбиения на треугольники существенно проще, чем при использовании других полигонов. Разбиение на треугольники существенно упрощает реализацию процедур рендеринга. Ну и наконец, на треугольники можно гарантированно разбить любую область, чего не скажешь о других полигонах. Процесс разбиения полигональной области со сложной конфигурацией в набор треугольников называется триангуляцией [15].

В данной работе приводится обзор существующих основных алгоритмов визуализации неявно заданных поверхностей на основе готового [19]. Также для обзора в числе других ссылок использован текст англоязычной статьи **Matthew Ward** [24]. Особую роль в данном исследовании отвели трем алгоритмам: «Марширующие кубы», Канейро и Скалы. Для их обзора были использованы не только существующие многочисленные литературные источники на русском языке, но и англоязычные статьи авторов-разработчиков данных методов [25], [31], [26].

Основная задача реконструкции поверхности формулируется следующим образом – необходимо найти оптимальную аппроксимацию для неизвестной поверхности так, чтобы минимизировать погрешность измерения, возникающую из-за ограничения по точности в измерительных приборах или из-за недостаточного качества поверхности физической модели.

Отличительной чертой всех алгоритмов восстановления поверхностей является их исключительная ресурсоемкость, обусловленная необходимостью обработки большого числа однотипных данных, поэтому повышение производительности реализации таких алгоритмов является актуальной задачей [18].

Согласно [18] Более ресурсоёмким и менее оптимизированным является алгоритм марширующих кубов, который, тем не менее, интересен для задач интерактивного и автоматического восстановления поверхностей объектов, представленных облаком точек, а также для построения выпуклых оболочек. Поэтому в данной работе будет реализован именно этот алгоритм.

Алгоритм состоит из двух основных стадий. Первая стадия: построение неявной функции  $F: D \rightarrow R$ , где  $D$  принадлежит  $R^3$  -окрестность поверхности. Функция  $F$  представляет собой оценку расстояния от любой точки из  $D$  до поверхности. Вторая стадия: триангуляция поверхности нулевого уровня функции  $F$  при помощи алгоритма марширующих кубов.

Алгоритм марширующих кубов можно разделить на 2 этапа:

1. Разбиение области  $G$  пространства  $R^3$  на заданное множество ячеек, определение ячеек, пересекаемых искомой поверхностью.
2. Аппроксимация поверхности в найденных ячейках.

Алгоритм марширующих кубов содержит много этапов, на которых выполняются независимые задачи. Наличие независимых задач является достаточным условием для распараллеливания алгоритма. К ним относятся:

- вычисление идентификаторов точек;
- переупорядочение точек;
- выделение вокселей сетки;

- вычисление шаблонов ячеек;
- генерация треугольников.

Алгоритм должен содержать следующие шаги:

1. Реализация ускоряющей структуры, для входного набора данных.
2. Восстановление поля нормалей.
3. Вычисление оценки расстояний.
4. Нахождение поверхности нулевого уровня оценки расстояния (алгоритм «марширующих кубов»).

На первом шаге в качестве ускоряющей структуры предлагается использовать регулярную сетку. Параллельное построение регулярной сетки состоит из пяти этапов: построение ограничивающего параллелепипеда, вычисление идентификаторов точек, сортировка идентификаторов точек, переупорядочивание точек, выделение вокселей сетки. В итоге, в памяти будут сохраняться только те воксели, которые содержат точки данных, что позволяет значительно снизить использование видеопамати.

На втором шаге, необходимо найти касательную плоскости для каждой из точек входных данных.

Реализовать можно в такой последовательности:

1. Получить из глобальной памяти обрабатываемую точку.
2. Определить целочисленные координаты вокселя  $(x, y, z)$ .
3. Выполнить выборку индексов точек вокселя  $(x, y, z)$  и его соседей.
4. Скопировать точки с полученными индексами в локальную память и отсортировать точки по возрастанию расстояния.
5. Выбрать из локальной памяти координаты ближайших точек.
6. Вычислить центроид и поместить его в выходной буфер.
7. Вычислить матрицу ковариации для выбранных точек, найти минимальное собственное число и поместить соответствующий ему собственный вектор в выходной буфер.

Для вычисления оценки расстояния предлагается помещать центроиды в ускоряющую структуру, что позволяет использовать следующую параллельную схему:

1. Получить из глобальной памяти обрабатываемую точку.
2. Определить целочисленные координаты вокселя  $(x, y, z)$ .
3. Найти ближайший центроид и соответствующую ему нормаль.
4. Вычислить оценку расстояния.

Следует отметить, что, несмотря на свою популярность и широкое применение, алгоритм марширующих кубов имеет существенный недостаток: на достаточно простых участках поверхности он выполняет слишком большое количество разделений. Например, на поверхности, которая может быть аппроксимирована с помощью 440 полигонов, классический алгоритм может сгенерировать порядка 67000. Это подчеркивает выигрыш в производительности от распараллеливания, но также говорит о необходимости улучшения базового алгоритма.

В данной работе показано, что задача повышения производительности методов, реконструирующих и визуализирующих поверхность, является актуальной [18].

Целью данной работы является реализация метода марширующих кубов на платформе Windows Presentation Foundation.

Для достижения данной цели необходимо выполнить следующие задачи:

1. исследовать предметную область и литературу по данной тематике;
11. проанализировать существующие основные алгоритмы визуализации неявно заданной поверхности;
12. визуализировать с помощью реализованного алгоритма марширующих кубов на платформе Windows Presentation Foundation пример части МРТ, изначально заданной неявно двумерным изображением, и другие примеры;
13. визуализировать с помощью реализованного алгоритма и другие примеры;

14. выявить, оценить, проанализировать результаты разработки проекта;
15. сравнить некоторые существующие алгоритмы, а именно: «Марширующие кубы», Канейро, Скалы.
16. оценить экономическую эффективность проекта;
17. проанализировать нормы и особенности социальной ответственности при разработке проекта;
18. проанализировать выводы и заключение.

Объектом исследования являются методы визуализации сложных 3D моделей. Предметом исследования является метод марширующих кубов.

**Научная или практическая новизна** – реализация метода марширующих кубов на платформе WPF. На данной платформе метод реализован впервые.

**Практическая значимость результатов ВКР:** реализованный метод может применяться в различных областях математики, физики, медицины, например:

1. Визуализация экспериментальных данных. Например, информации, собранной с различных датчиков, сенсорных сетей, результатов измерений, результатов имитационного моделирования.
2. Визуализация функционального представления. Визуализация сложных трёхмерных объектов в математике, визуализация аналитически заданных поверхностей и т.п.
3. Визуализация медицинских наборов данных. Одним из широко используемых методов диагностики является томография. Томографы, работающие на рентгеновском принципе (компьютерная томография, далее КТ) и на магнитно-резонансном принципе (далее МРТ) позволяют получить изображения множества сечений тела пациента вдоль какой-либо оси, которые дают много информации об особенностях его анатомии и физиологии. Методы визуализации позволяют реконструировать трехмерную структуру органов по множеству параллельных сечений. Визуализация всего набора данных в

настоящее время реализована в программном обеспечении большинстве современных томографов, но задачи выделения трехмерных областей, визуализации области интереса, заданной набором контуров на срезах, визуализация области при радиологическом лечении пациентов, как правило, в этом программном обеспечении не реализованы.

4. Визуализация экспериментальных и модельных данных, из других областей, таких как геология, метеорология, молекулярный анализ, динамика жидкостей [19].

В результате выполнения работы с помощью алгоритма была визуализирована МРТ, а также другие примеры трехмерных объектов.

**Реализация и апробация работы:** по данной тематике был представлен доклад на конференции «Гагаринские чтения», г. Москва в апреле 2016 г.

#### **Методы исследования**

Данное исследование находится на эмпирическом уровне научного познания. В основе исследования применен один из общих методов научного познания – метод эмпирического исследования. Из методов эмпирического исследования применен метод: материальное моделирование [21].

Основное средства разработки – язык программирования C# (XAML), среда разработки программного обеспечения Microsoft Visual Studio 2013.

## **2. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение**

Данная работа посвящена разработке алгоритма компьютерной графики «марширующие кубы» на платформе Windows Presentation Foundation. Соответственно, данная работа имеет экономическую составляющую, которую необходимо оценить с целью определения значимости выполненной работы и ее вклада в заданную область исследования.

Целью данного раздела является анализ финансово-экономической стороны выполненной работы. Оцениваются финансовые затраты на исследование, дается экономическая оценка результатов ее внедрения и оценивается экономическая целесообразность реализации разработки, приводится комплексная оценка научно-технического уровня работы, основанная на экспертных данных.

### **5.1 Организация и планирование работ**

Процесс реализации работы должен быть рационально спланирован, рассчитана занятость каждого из участников и сроки выполнения отдельных работ.

В данном подразделе составляется перечень проводимых работ, определяются их исполнители и рациональная продолжительность. Планирование работ наглядно представлено на линейном графике выполнения работы. Для выполнения перечисленных в таблице 5.1 работ требуются специалисты: инженер (в его роли действует исполнитель НИР) и научный руководитель.

Линейный график отражает хронологически упорядоченные данные, которые сведены в таблицу 5.1.

Таблица 5.1 - Состав работ и продолжительность их выполнения

Этапы работы	Исполнители	Загрузка исполнителей
<b>Постановка задачи</b>		
1. Постановка целей и задач, получение исходных данных	НР	НР – 100%
2. Составление и утверждение требований к программе	НР, И	НР – 100% И – 10%
3. Составление календарного плана	НР, И	НР – 100% И – 10%
<b>Исследование темы и изучение теории</b>		
4. Подбор и обзор литературы по теме	НР, И	НР – 30% И – 100%
5. Исследование алгоритмов визуализации трехмерных объектов	НР, И	НР – 10% И – 100%
<b>Реализация программной части</b>		
6. Реализация алгоритма маршрутирующих кубов на платформе WPF	НР, И	НР – 10% И – 100%
7. Анализ результатов	НР, И	НР – 10% И – 100%
<b>Изучение дополнительных составляющих и документирование</b>		
8. Расчет экономической составляющей исследования	И	И – 100%
9. Оценка социальной ответственности исследования	И	И – 100%
10. Оформление пояснительной записки	И	И – 100%

11. Подведение итогов	НР, И	НР – 60% И – 100%
-----------------------	-------	----------------------

### 5.1.1 Продолжительность этапов работ

В данной работе расчет продолжительности этапов работ осуществляется опытно-статистическим методом, который реализуется экспертным способом.

Экспертный способ используется при отсутствии проекта, который по всем значимым параметрам идентичен выполняемой НИР, и предполагает генерацию необходимых количественных оценок специалистами определенной предметной области. Для определения вероятных (ожидаемых) значений продолжительности работ  $t_{ож}$  применяется формула:

$$t_{ож} = \frac{3 \cdot t_{min} + 2 \cdot t_{max}}{5}, \quad (1.1-a)$$

где  $t_{min}$  – минимальная продолжительность работы, дн.;

$t_{max}$  – максимальная продолжительность работы, дн.

Для построения линейного графика рассчитывается длительность этапов в рабочих днях, а затем переводится в календарные дни. Расчет продолжительности выполнения каждого этапа в рабочих днях ( $T_{РД}$ ) ведется по формуле:

$$T_{РД} = \frac{t_{ож}}{K_{ВН}} \cdot K_{Д} \quad (1.2)$$

где  $t_{ож}$  – продолжительность работы, дн.;

$K_{ВН}$  – коэффициент выполнения работ, учитывающий влияние внешних факторов на соблюдение предварительно определенных длительностей, в частности, возможно  $K_{ВН} = 1$ ;

$K_d$  – коэффициент, учитывающий дополнительное время на компенсацию непредвиденных задержек и согласование работ ( $K_d = 1-1,2$ ; в этих границах конкретное значение принимает сам исполнитель).

Расчет продолжительности этапа в календарных днях ведется по формуле:

$$T_{KD} = T_{RD} \cdot T_K, \quad (1.3)$$

где  $T_{KD}$  – продолжительность выполнения этапа в календарных днях;

$T_K$  – коэффициент календарности, позволяющий перейти от длительности работ в рабочих днях к их аналогам в календарных днях, и рассчитываемый по формуле\*

$$T_K = \frac{T_{КАЛ}}{T_{КАЛ} - T_{ВД} - T_{ПД}} \quad (1.4)$$

где  $T_{КАЛ}$  – календарные дни ( $T_{КАЛ} = 365$ );

$T_{ВД}$  – выходные дни ( $T_{ВД} = 52$ );

$T_{ПД}$  – праздничные дни ( $T_{ПД} = 10$ ).

$$T_K = \frac{365}{365 - 52 - 10} = 1,205$$

Таблица 5.2 отражает продолжительность всех этапов работ и их трудоемкость по двум исполнителям на каждом этапе. Столбцы (3–5) демонстрируют экспертный способ по формуле (1.1-а). Столбцы 6 и 7 содержат величины трудоемкости этапа для каждого из двух участников проекта при коэффициенте  $K_d = 1,2$ . Каждое из них в отдельности не превышает соответствующее значение  $t_{ож} * K_d$ . Столбцы 8 и 9 содержат те же самые трудоемкости, выраженные в календарных днях при  $T_K = 1,212$ . Итог по столбцу 5 отражает общую ожидаемую продолжительность работы над проектом в рабочих днях, итоги по столбцам 8 и 9 – общие трудоемкости для каждого из двух участников работы. Две последних величины используются в дальнейшем для определения затрат на оплату труда участников и прочие

затраты. Величины трудоемкости этапов по исполнителям  $T_{КД}$  (данные столбцов 8 и 9 кроме итогов) позволяют построить линейный график осуществления проекта (см. табл. 5.3).

Таблица 5.2 - Трудозатраты на выполнение этапов проекта

Этап	Исполнители	Продолжительность работ, дни			Трудоемкость работ по исполнителям чел.- дн.			
					$T_{РД}$		$T_{КД}$	
		$t_{min}$	$t_{max}$	$t_{ож}$	НР	И	НР	И
1	2	3	4	5	6	7	8	9
1. Постановка целей и задач, получение исходных данных	НР	2	4	2,8	3,36	–	4,07	–
2. Составление и утверждение требований к программе	НР, И	2	3	2,4	2,88	0,29	3,5	0,35
3. Составление календарного плана	НР, И	2	4	2,8	3,36	0,34	4,07	0,41
4. Подбор и обзор литературы по теме	НР, И	10	14	11,6	4,18	13,92	5,07	16,87
5. Исследование алгоритмов визуализации трехмерных объектов	НР, И	10	12	10,8	1,3	12,96	1,58	15,71

6. Реализация алгоритма марширующих кубов на платформе WPF	НР, И	15	21	17,4	2,1	20,88	2,55	25,31
7. Анализ результатов	НР, И	10	14	11,6	1,4	13,92	1,7	16,87
8. Расчет экономической составляющей исследования	И	10	14	11,6	–	13,92	–	16,87
9. Оценка социальной ответственности исследования	И	10	14	11,6	–	13,92	–	16,87
10. Оформление пояснительной записки	И	8	10	8,8	–	10,56	–	12,8
11. Подведение итогов	НР, И	2	4	2,8	2,02	3,36	2,45	4,07
<b>Итого:</b>				<b>94,2</b>	<b>20,6</b>	<b>104,07</b>	<b>25</b>	<b>126,13</b>

Таблица 5.3 - Линейный график работ

Этап	Исполнитель	Продолжительность работ, дни			Трудоемкость работ по исполнителям чел.- дн.			
					$T_{рд}$		$T_{кд}$	
		$t_{min}$	$t_{max}$	$t_{ож}$	НР	И	НР	И
1	2	3	4	5	6	7	8	9

12. Постановка целей и задач, получение исходных данных	НР	2	4	2,8	3,36	–	4,07	–
13. Составление и утверждение требований к программе	НР, И	2	3	2,4	2,88	0,29	3,5	0,35
14. Составление календарного плана	НР, И	2	4	2,8	3,36	0,34	4,07	0,41
15. Подбор и обзор литературы по теме	НР, И	10	14	11,6	4,18	13,92	5,07	16,87
16. Исследование алгоритмов визуализации трехмерных объектов	НР, И	10	12	10,8	1,3	12,96	1,58	15,71
17. Реализация алгоритма маршрутирующих кубов на платформе WPF	НР, И	15	21	17,4	2,1	20,88	2,55	25,31
18. Анализ результатов	НР, И	10	14	11,6	1,4	13,92	1,7	16,87
19. Расчет экономической составляющей исследования	И	10	14	11,6	–	13,92	–	16,87
20. Оценка социальной ответственности исследования	И	10	14	11,6	–	13,92	–	16,87
21. Оформлены	И	8	10	8,8	–	10,56	–	12,8

е пояснительной записки								
22. Подведение итогов	НР, И	2	4	2,8	2,02	3,36	2,45	4.07
<b>Итого:</b>				<b>94,2</b>	<b>20,6</b>	<b>104,0</b> <b>7</b>	<b>25</b>	<b>126,1</b> <b>3</b>

НР –  ; И – 

### 5.1.2 Расчет накопления готовности проекта

Целью данного пункта является оценка текущих состояний работы над проектом. Величина накопления готовности работы отражает, на сколько процентов по окончании текущего (*i*-го) этапа выполнен общий объем работ по проекту в целом.

Обозначения:

- $TP_{\text{общ.}}$  – общая трудоемкость проекта;
- $TP_i$  ( $TP_k$ ) – трудоемкость *i*-го (*k*-го) этапа проекта,  $i = \overline{1, I}$ ;
- $TP_i^H$  – накопленная трудоемкость *i*-го этапа проекта по его завершении;
- $TP_{ij}$  ( $TP_{kj}$ ) – трудоемкость работ, выполняемых *j*-м участником на *i*-м этапе, здесь  $j = \overline{1, m}$  – индекс исполнителя, в нашем случае  $m = 2$ .

Степень готовности определяется формулой (5.5)

$$CG_i = \frac{TP_i^H}{TP_{\text{общ.}}} = \frac{\sum_{k=1}^i TP_k}{TP_{\text{общ.}}} = \frac{\sum_{k=1}^i \sum_{j=1}^m TP_{km}}{\sum_{k=1}^I \sum_{j=1}^m TP_{km}} \quad (5.5)$$

Применительно к таблице (5.2) величины  $TP_{ij}$  ( $TP_{kj}$ ) находятся в столбцах (6,  $j = 1$ ) и (7,  $j = 2$ ).  $TP_{\text{общ.}}$  равна сумме чисел из итоговых клеток этих столбцов. Пример расчета  $TP_i$  (%) и  $CG_i$  (%) на основе этих данных содержится в таблице (5.4).

Таблица **Ошибка! Текст указанного стиля в документе отсутствует.**4

- Нарастание технической готовности работы и удельный вес каждого этапа

Этап	ТР <sub>i</sub> , %	СГ <sub>i</sub> , %
23. Постановка целей и задач, получение исходных данных	2.7	2.7
24. Составление и утверждение требований к программе	2.5	5.2
25. Составление календарного плана	3	8.2
26. Подбор и обзор литературы по теме	14.52	22.72
27. Исследование алгоритмов визуализации трехмерных объектов	11.44	34.16
28. Реализация алгоритма маршрутирующих кубов на платформе WPF	18.43	52.59
29. Анализ результатов	13	65.59
30. Расчет экономической составляющей исследования	11.17	76.76
31. Оценка социальной ответственности исследования	11.17	87.93
32. Оформление пояснительной записки	8.47	96.4
33. Подведение итогов	4.32	100.00

## 5.2 Расчет сметы затрат на выполнение проекта

В состав затрат на создание проекта включаются все расходы, необходимые для реализации перечня работ, входящих в состав данной разработки. Сметная стоимость рассчитывается по следующим статьям затрат:

- материалы и покупные изделия;
- заработная плата;
- социальный налог;

- расходы на электроэнергию (без освещения);
- амортизационные отчисления;
- командировочные расходы – для данного проекта нет;
- оплата услуг связи – для данного проекта нет;
- арендная плата за пользование имуществом – для данного проекта нет;
- прочие услуги (сторонних организаций) – для данного проекта нет;
- прочие (накладные расходы) расходы.

### 5.2.1 Расчет затрат на материалы

К текущей статье расходов относится стоимость материалов и других материальных ценностей, которые расходуются в процессе работы. Также сюда относятся специально приобретенное оборудование и прочие объекты, относимые к основным средствам, стоимостью до 40 000 руб. включительно. Цена материальных ресурсов определяется на основе ценников или договоров на поставку. В данную статью включены и транспортно-заготовительные расходы - для данного проекта их нет. Сюда же включаются расходы на совершение сделки купли-продажи – для данного проекта их нет. Приблизительно они оцениваются в процентах к отпускной цене закупаемых материалов, обычно это 5 ÷ 20 %. Исполнитель работы самостоятельно выбирает их величину в указанных границах. Для данной работы выбрано значение 5%. Расчет представлен в табл. 5.5.

Таблица 5.5 - Расчет затрат на материалы

Наименование материалов	Цена за ед., руб.	Кол-во	Сумма, руб.
Бумага для принтера формата А4	190	1 уп.	190
Картридж для принтера	1550	1 шт.	1550

<b>Итого:</b>			<b>1740</b>
---------------	--	--	-------------

Расходы на материалы с учетом ТЗР равны  $C_{\text{мат}} = 1740 * 1,05 = 1827$  руб.

### 5.2.2 Расчет заработной платы

Данная статья расходов включает заработную плату двух участников проекта: научного руководителя и инженера. Основная заработная плата рассчитывается на основе трудоемкости выполнения каждого этапа, а также величины месячного оклада руководителя и исполнителя. В данном случае рассматриваются следующие оклады:

- 1) научный руководитель: должность – зав. кафедрой, степень – кандидат технических наук;
- 2) исполнитель: программист 1-го квалификационного уровня, в соответствии с окладами ТПУ.

Среднедневная тарифная заработная плата ( $ЗП_{\text{дн-т}}$ ) рассчитывается по формуле:

$$ЗП_{\text{дн-т}} = \text{МО} / 24,83 \quad (5.6)$$

где МО – месячный оклад исполнителя,

учитывающей, что в году 298 рабочих дней и, следовательно, в месяце в среднем 24,83 рабочих дня (при шестидневной рабочей неделе)<sup>†</sup>.

Расчеты затрат на полную заработную плату приведены в таблице 5.6. Затраты времени по каждому исполнителю в рабочих днях с округлением до целого взяты из таблицы 5.2. Для учета в ее составе премий, дополнительной зарплаты и районной надбавки используется следующий ряд коэффициентов:  $K_{\text{ПР}} = 1,1$ ;  $K_{\text{доп.ЗП}} = 1,188$ ;  $K_{\text{р}} = 1,3$ . Таким образом, для перехода от тарифной (базовой) суммы заработка исполнителя, связанной с участием в проекте, к соответствующему полному заработку (зарплатной части сметы) необходимо первую умножить на интегральный коэффициент  $K_{\text{и}} = 1,1 * 1,188 * 1,3 = 1,699$ .

Вышеуказанное значение  $K_{\text{доп.ЗП}}$  применяется при шестидневной рабочей неделе.

Таблица **Ошибка! Текст указанного стиля в документе отсутствует..3**

- Затраты на заработную плату

<b>Исполнитель</b>	<b>Оклад, руб./мес.</b>	<b>Среднедневная ставка, руб./раб.день</b>	<b>Затраты времени, раб.дни</b>	<b>Коэффициент</b>	<b>Фонд з/платы, руб.</b>
НР	<b>29 744,67</b>	1 197,93	21	1,699	<b>42 740,94</b>
И	<b>7 864,11</b>	316,72	104	1,699	<b>55 963,16</b>
<b>Итого:</b>					<b>98 704,1</b>

### 5.2.3 Расчет затрат на социальный налог

Затраты на единый социальный налог (ЕСН), составляют 30 % от полной заработной платы по проекту. Данные затраты включают в себя отчисления в пенсионный фонд, фонды социального и медицинского страхования.

$C_{\text{соц.}} = C_{\text{зп}} * 0,3$ . В нашем случае  $C_{\text{соц.}} = 98\,704,1 * 0,3 = 29\,611,23$  руб.

### 5.2.4 Расчет затрат на электроэнергию

Затраты на электроэнергию включает в себя расходы на

электроэнергию, потраченную в ходе выполнения проекта, на работу задействованного оборудования, и рассчитываются по формуле:

$$C_{\text{эл.об.}} = P_{\text{об}} * t_{\text{об}} * Ц_{\text{э}} \quad (5.7)$$

где  $P_{\text{об}}$  – мощность, потребляемая оборудованием, кВт;

$Ц_{\text{э}}$  – тариф на 1 кВт·час;

$t_{\text{об}}$  – время работы оборудования, час.

В ТПУ  $Ц_{\text{э}} = 5,257$  руб./кВт·час (с НДС).

Время работы оборудования вычисляется на основе итоговых данных таблицы 5.2 для инженера ( $T_{\text{рд}}$ ) из расчета, что продолжительность рабочего дня равна 8 часов.

$$t_{\text{об}} = T_{\text{рд}} * K_t, \quad (5.8)$$

где  $K_t \leq 1$  – коэффициент использования оборудования по времени, равный отношению времени его работы в процессе выполнения проекта к  $T_{\text{рд}}$ , приняли самостоятельно  $K_C = 0.9$ .

Мощность, потребляемая оборудованием, определяется по формуле:

$$P_{\text{об}} = P_{\text{ном.}} * K_C \quad (5.9)$$

где  $P_{\text{ном.}}$  – номинальная мощность оборудования, кВт;

$K_C \leq 1$  – коэффициент загрузки, зависящий от средней степени использования номинальной мощности. Для технологического оборудования малой мощности  $K_C = 1$ .

Результаты расчета затрат на электроэнергию представлены в таблице 5.7.

Таблица **Ошибка! Текст указанного стиля в документе отсутствует.**4

- Затраты на электроэнергию технологическую

Наименование оборудования	Время работы оборудования $t_{\text{об}}$ , час	Потребляемая мощность $P_{\text{об}}$ , кВт	Тариф $Ц_{\text{э}}$ , руб/кВт·час	Затраты $Э_{\text{об}}$ , руб.
Персональный	$8*104*0,9=748,8$	0,3	5,257	1 180,93

компьютер				
Струйный принтер	30	0,1	5,257	15,77
<b>Итого:</b>				<b>1 196,7</b>

### 5.2.5 Расчет амортизационных расходов

В данной статье определяется амортизация задействованного оборудования за время выполнения проекта по формуле:

$$C_{AM} = \frac{N_A * C_{OB} * t_{pф} * n}{F_D}, \quad (5.10)$$

где  $N_A$  – годовая норма амортизации единицы оборудования;

$C_{OB}$  – балансовая стоимость единицы оборудования с учетом ТЗР.

$F_D$  – действительный годовой фонд времени работы соответствующего оборудования, берется из специальных справочников или фактического режима его использования в текущем календарном году.

$$F_D = 8 * 104 = 832 \text{ часа.}$$

$t_{pф}$  – фактическое время работы оборудования в ходе выполнения проекта, учитывается исполнителем проекта;

$n$  – число задействованных однотипных единиц оборудования.

$N_A$  определяется по данным постановления правительства РФ «О классификации основных средств, включенных в амортизационные группы». Дает рамочные значения сроков амортизации оборудования  $\equiv$  СА. Например, для ПК это 2 ÷ 3 года.

Величина  $t_{pф}$  берется на основании фактического режима использования оборудования в течение текущего календарного года.

Таблица 5.8 - Амортизационные расходы

	$N_A$	Ц <sub>об</sub> , руб.	$t_{рф}$ , часов	n, шт	$F_d$ , час	$C_{ам}$ , руб
ПК	0,4	45 000	832	1	832	22 500
Принтер	0,5	12 000	30	1	500	360
<b>Итого</b>						<b>22 860</b>

### 5.2.6 Расчет прочих расходов

В статье «Прочие расходы» отражены расходы на выполнение проекта, которые не учтены в предыдущих статьях, их следует принять равными 10% от суммы всех предыдущих расходов, т.е.

$$C_{\text{проч.}} = (C_{\text{мат}} + C_{\text{зп}} + C_{\text{соц}} + C_{\text{эл.об.}} + C_{\text{ам}}) \cdot 0,1$$

Для нашего примера это

$$C_{\text{проч.}} = (1\,740 + 98\,704,1 + 29\,611,23 + 1\,196,7 + 22\,860) \cdot 0,1 = 15\,410 \text{ руб.}$$

### 5.2.7 Расчет общей себестоимости разработки

Теперь определяем общую себестоимость проекта.

Таблица **Ошибка! Текст указанного стиля в документе отсутствует..9**

- Смета затрат на разработку проекта

Статья затрат	Условное обозначение	Сумма, руб.
Материалы и покупные изделия	$C_{\text{мат}}$	1 740
Основная заработная плата	$C_{\text{зп}}$	<b>98 704,1</b>
Отчисления в социальные фонды	$C_{\text{соц}}$	29 611,23
Расходы на электроэнергию	$C_{\text{эл.}}$	<b>1 196,7</b>
Амортизационные отчисления	$C_{\text{ам}}$	<b>22 860</b>
Прочие расходы	$C_{\text{проч}}$	15 410
<b>Итого:</b>		<b>169 500</b>

Итого затраты на разработку составили  $C = 169\,500$  руб.

### 5.2.8 Расчет прибыли

Прибыль от реализации проекта может рассчитываться различными способами, в зависимости от конкретной ситуации. Если у исполнитель нет данных для использования специальных методов, прибыль следует принять в размере 5 ÷ 20 % от полной себестоимости проекта.

Данная работа имеет широкую многоотраслевую сферу применения и не используется в прямом виде, как продукт, а лишь является средством для визуализации трехмерных объектов и данных в компьютерной графике. Чтобы продуктом мог пользоваться исследователь или работник, например, математик или врач, необходимо на основе данного продукта-средства разрабатывать другие программные продукты. В связи с этим программный продукт не несет за собой прямой экономической выгоды. В нашем примере примем прибыль условно равной 8 475 руб. (5 %) от расходов на разработку проекта.

### 5.2.9 Расчет НДС

НДС составляет 18% от суммы затрат на разработку и прибыли. В нашем случае это  $(169\,500 + 8\,475) * 0,18 = 177\,975 * 0,18 = 32\,035,5$  руб.

### 5.2.10 Цена разработки НИР

Цена равна сумме полной себестоимости, прибыли и НДС, в нашем случае

$$C_{\text{НИР(КР)}} = 169\,500 + 8\,475 + 32\,035,5 = 210\,010,5 \text{ руб.}$$

## 5.3 Оценка экономической эффективности проекта

В данной работе реализуется алгоритм компьютерной графики «марширующие кубы» на платформе WPF. Область применения данного

алгоритма очень широка, алгоритм может применяться при программировании компьютерной графики в различных сферах применения. Но чтобы использовать возможности данного алгоритма для визуализации сложных трехмерных объектов, необходимо, используя данную разработку как средство, разрабатывать новые системы с понятным и доступным работникам определенных областей интерфейсом взаимодействия с пользователем. Другими словами, данный проект не является самостоятельным.

Поскольку разработка не носит коммерческий характер, оценка экономической эффективности для данного проекта не является целесообразной.

Данный продукт может быть использован в дальнейших научных исследованиях и, возможно, в будущем, при его развитии и интеграции к конкретной области, системы, куда он будет встраиваться смогут носить коммерческий характер и приносить экономическую прибыль.

Из достоинств данной разработки можно отметить необходимость в небольших ресурсах персонального компьютера.

### **5.3.1 Оценка научно-технического уровня НИР**

Научно-технический уровень определяет влияние проекта на уровень и динамику обеспечения научно-технического прогресса в текущей области. Оценка научной ценности, технической значимости и эффективности, планируемых и выполняемых НИР рассчитывается методом балльных оценок. Суть балльной оценки состоит в том, что каждому фактору по установленной шкале присваивается определенное количество баллов. Суммарная оценка рассчитывается по сумме баллов по всем показателям. И на основе данной оценки заключается вывод о целесообразности НИР.

Интегральный показатель научно-технического уровня рассчитывается по формуле:

$$K_{\text{НТУ}} = \sum_{i=1}^3 R_i \cdot n_i, \quad (5.14)$$

где  $I_{\text{НТУ}}$  – интегральный индекс научно-технического уровня;

$R_i$  – весовой коэффициент  $i$ -го признака научно-технического эффекта;

$n_i$  – количественная оценка  $i$ -го признака научно-технического эффекта, в баллах.

Таблица **Ошибка! Текст указанного стиля в документе отсутствует.** 10- Весовые коэффициенты признаков НТУ

Признаки научно-технического эффекта НИР	Характеристика признака НИР	Ri
Уровень новизны	Систематизируются и обобщаются сведения, определяются пути дальнейших исследований	00,4
Теоретический уровень	Разработка способа (алгоритм, программа мероприятий, устройство, вещество и т.п.)	00,1
Возможность реализации	Время реализации в течение первых лет	00,5

Величина интегрального показателя принадлежит интервалу  $[0, 10]$ .

Таблица 5.11 отражает соответствие качественных уровней НИР значениям показателя, рассчитываемого по формуле (5.14).

Таблица 5.11 - Соответствие

Уровень НТЭ	Показатель НТЭ
Низкий	1-4
Средний	4-7
Высокий	8-10

Для данного проекта частные оценки уровня  $p_i$  и их краткое обоснование приведены в таблице (5.12).

Таблица **Ошибка! Текст указанного стиля в документе отсутствует.**2

- Оценки научно-технического уровня НИР

Значимость	Фактор НТУ	Уровень фактора	Выбранный балл	Обоснование выбранного балла
0,4	Уровень новизны	Новая	5	Разработка алгоритма маршрутирующих кубов на современной перспективной платформе WPF с возможностью модификации и оптимизации данного метода в будущем
0,1	Теоретический уровень	Разработка способа (программа, реализующая алгоритм)	6	Разработан программный продукт, реализующий алгоритм маршрутирующих кубов для визуализации трехмерных объектов

0,5	Возможность реализации	В течение первых лет	10	Визуализация трехмерной объектов разработчиками компьютерной графики.
-----	------------------------	----------------------	----	---

Отсюда интегральный показатель НТУ для нашего проекта составляет:

$$I_{\text{НТУ}} = 0,4 * 5 + 0,1 * 6 + 0,5 * 10 = 2 + 0,6 + 5 = 7,6$$

Таким образом, опираясь на данные таблицы 5.15, видно, что данный проект имеет средний уровень научно-технического эффекта.

**Вывод:** В данном разделе была произведена оценка различных экономических аспектов разработанного проекта. На основании полученных результатов проекту можно дать следующую характеристику:

- проект относится к краткосрочному виду проектов по времени разработки;
- проект можно отнести к категории малых проектов по масштабности;
- актуальность решаемой проблемы повышают конкурентоспособность проекта в текущей предметной области.

## Список публикаций

Хасаева Д.З. Создание 3D моделей сложных объектов методом марширующих кубов с использованием технологии WPF // **Гагаринские чтения – 2016**: XLII Международная молодёжная научная конференция: Сборник тезисов докладов: В 4 т., Москва, 12-14 апреля 2016. - Москва: МАИ (НИУ), 2016 - с. 456-457.

Хасаева Д.З., Демин А.Ю. Визуализация 3D моделей сложных объектов с помощью метода марширующих кубов на платформе Windows Presentation Foundation // Технологии Microsoft в теории и практике программирования: Сборник трудов XIII Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых, Томск, 22-23 марта 2016. – Томск: Изд-во ТПУ.

Хасаева Д.З., Демин А.Ю. Разработка графической библиотеки для визуализации объектов робототехники на основе технологии Windows Presentation Foundation // **XXVII Международная инновационно-ориентированная конференция молодых ученых и студентов (МИКМУС - 2015)**: Труды конференции, Москва, 2-4 декабря, 2015. - Москва: Изд-во ИМАШ РАН, 2015 – с. 536-539

Sorokin V. A. , Demin A. Y. , Khasaeva D. Z. Three-dimensional visualization of the underwater environment using graphical library "3Dbodies" // Mechanical Engineering, Automation and Control Systems: Proceedings of International Conference, Tomsk, October 16-18, 2014. - Tomsk: TPU Publishing House, 2014 - p. 1-5.

Хасаева Д.З., Демин А.Ю. **Разработка графической библиотеки для построения 3D тел на основе технологии WPF** // Технологии Microsoft в теории и практике программирования: Сборник трудов XI Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых, Томск, 23-24 апреля 2014. – Томск: Изд-во ТПУ.

# Приложение А (обязательное)

## Раздел 2.1

Visualization of three-dimensional objects with the help of triangulation algorithms for defined surface

## Раздел 2.2

Methods for solving a triangulation problem

Студент:

Группа	ФИО	Подпись	Дата
8ВМ4Б	Хасаева Диана Зуфаровна		

Консультант кафедры ИПС:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
и.о. руководителя каф. ИПС, доцент каф. ИПС	Демин Антон Юрьевич	канд. техн. наук		

Консультант–лингвист кафедры ИЯИК:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
зав. каф. ИЯИК, доцент каф. ИЯИК	Сидоренко Татьяна Валерьевна	канд. пед. наук		

## 2.1. Visualization of three-dimensional objects with the help of triangulation algorithms for defined surface

Visualization of an implicit surface it is visualization of the surface defined by a three-argument function and a fixed value of this function, called the level.

$$(x, y, z)/f(x, y, z) = c \quad (2.1)$$

where  $f(x, y, z)$  – the defined function, and  $c$  – the defined level.

The Algorithms of surface rendering build the image of a surface in three-dimensional space. However, it is more convenient to make restoring not of this surface but the surface approximating it with the help of triangles. The first original surface can be approximated by polygons and then the polygons are visualized with graphic libraries.

The problem of visualization appears in many areas of mathematics, physics, and medicine:

- *Visualization of experimental data.* During the experiments in physics, often it is necessary to display the information immediately from all sensors.
- *Functional representation.* Some mathematical tasks or calculations need to visualize a geometric object, which is defined by a single continuous describing function of several variables, which look like  $F(X) > 0$ .
- *Medicine.* Computers use makes possible the development in new areas of such technologies as computed tomography, magnetic resonance imaging (MRI) and positron emission tomography. With the help of tomography equipment, we can get many images of some sections of a patient's body. It allows to give characteristics of his features of anatomy and physiology. These images with a big clarity can show us the different organs. Visualization techniques give possibility to reconstruct the three-dimensional structure of the organs using a variety of parallel sections. Often for the right diagnosis, a doctor just visually analyzes the individual images of the object sections obtained from the tomography examination.

Similar approaches are used for experimental and simulated data, which are obtained in other areas, such as fluid dynamics, geology, meteorology, molecular analysis.

In the process of visualization, an important role is in determining the function that describes the desired surface. Most of application tasks use the function in a tabular form

However, sometimes there are some problems in which there is not a clear imaging, or a table of values has the irregular grid.

These problems can be in many applications, for example: range measurement using irradiation surface or reconstruction of three-dimensional structure with the help of plurality of sections contour in the medical research.

These problems use the following sequence of actions: the surface  $S$ , a set of the sampling  $X$  approximated by the tangent planes passing through each sampling point  $X$  [1].

Then the desired function, which defines the surface, is calculated as follows: for each point of the space  $R$  the function at this point is equal to the nearest distance of the tangent plane with the + sign, if the point there is inside the volume, bounded with the built planes, or with the sign – if the point there is outside this volume. After that, the surface is exposed triangulation defined by a receiving function.

It is clear that the task of constructing a triangulation on the original set of points is ambiguous, so the question is: which of the possible triangulations is better.

The triangulation is called optimal if the sum of the lengths of all the edges is minimal among all possible triangulations of the original set of points.

However, the task of constructing a triangulation is NP-complete. Its complexity is  $O(e^N)$ , however, in practice, approximate algorithms are used. .

There are so-called a greedy triangulation and triangulation which they build also is called greedy.

The complexity of the work of the greedy algorithm for some of its improvements is  $O(N^2 \log N)$ . It is rarely used in practice for the reason of a high complexity. .

## 2.2. Methods for solving a triangulation problem

In addition to optimal and greedy algorithms, there are other triangulations and the methods to construct them. All the methods applied in solving the triangulation problem of are divided into the following three groups.

- *Cell-based methods.* The methods of this type make partition of the triangulation on the cells consist of parallelepipeds, triangular pyramids, tetrahedrons, octahedrons, etc. Then, each of surface triangulation performed in individual cell. Moreover, each cell is triangulated one of the previously defined methods, it means that coordinate values for triangles simply "substituted" from a predetermined table. To use this type of methods is necessary to specify the allowable error of approximation, from which to select the cell size. After that, with the help of the known triangulation tables to get the required set of triangles. In this procedure the triangulation of each cell is reduced to the analysis of the values at the vertices of the cell - in other words, it is determined which vertices are "inside" the surface, and which - "outside." Based on this we can conclude on the adequacy of the definition of the function only in the cells of the vertices.

- *Method of predictor-corrector.* The methods of this class are based on the addition to the existing plurality of triangulation points another lying on the plane tangent to the given function (this position predictor (predictor) - predicted), and then moving it to the imaging surface (this position corrector (corrector) - corrected).

By using the methods of this class, you must know the value of the function at all points of space and find at least one point belonging to the desired surface. The method consists in increasing the number of triangles iteratively - each iteration of the method to an existing set of triangles added another, built on the edge of the triangle and the predicted extreme (and then adjusted according to the curvature of the surface) of the point.

- *Pre-tessellation methods & particle-based methods.* The essence of these methods lies in decomposition of the desired surface in the part to further their

triangulation. Splitting apart in pre-tessellation techniques involves breaking the surface on the surface of the primitive - fragments of spheres and planes. Splitting into pieces in the methods of particle-based constellation of at least intellectually - sought only fragments of the planes. In this connection, there is a problem already triangulate parts.

– *Pre-tessellation methods & particle-based methods*). The essence of these methods lies in decomposition of the desired surface in the part to further their triangulation. Splitting apart in pre-tessellation techniques involves breaking the surface on the surface of the primitive - fragments of spheres and planes. Splitting into pieces in the methods of particle-based constellation of at least intellectually - sought only fragments of the planes. In this connection, the problem arises already triangulate parts. Most often, this process reduced to a Delaunay triangulation, that is, the selection by the local Delaunay triangles connecting portion of a surface.

All these methods have both advantages and disadvantages. The basis of the first type of methods - triangulation independent of each cell using triangulation tables - is both their strengths and weaknesses. The high speed of these techniques makes them more attractive in relation to other methods and makes it possible to use them in interactive applications, but the big drawback is their relative indifference to the behavior of the function is selected the set of points. In other words, it is impossible for properly visualize the local curvature - scale triangles is always proportional to the size of the cell. Such methods are ideally suited for the visualization of three-dimensional scalar fields defined on a regular grid.

Methods of the second and third type are useful only when rendered fields defined at every point of the part of space that we are interested. A great advantage of such methods can considered as their dependence on the local curvature of the function - in such methods are not minor details "will be gone." Despite the strong loss in speed compared with the methods of the first group and restrictions on the differentiability of functions and connectivity of the surface, they attract a high "quality" of the resulting surface.

### 2.2.1. Algorithms for constructing a Delaunay triangulation

Firstly, the task of constructing the Delaunay triangulation was set in 1934 by Soviet mathematician Boris Delaunay. A triangulation is called the Delaunay triangulation, if the circle inside circumscribed around any triangle constructed, does not fit one of the given points of triangulation (see Figure 2.1).

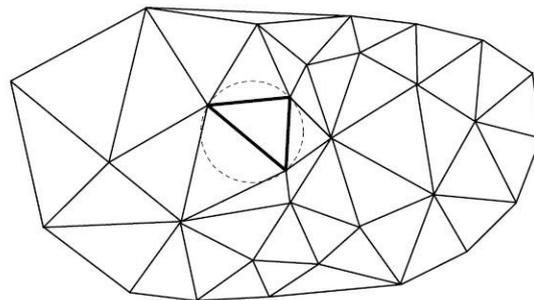


Figure a.1 - Triangulation of Delaunay.

Its complexity is determined by  $O(N \log N)$  arithmetic operations. There are algorithms that reach this estimation in the average and worst cases. Furthermore, it is known that the algorithms are able to achieve in some cases on average  $O(N)$ .

List of the most famous algorithms for constructing the Delaunay triangulation:

1. Iterative Algorithms
  - (a) Simple iterative algorithm
  - (b) Algorithms with search indexing triangles
  - (c) Caching algorithms search triangles
  - (d) Algorithms with the changed order of addition of points
2. Algorithms for constructing a Delaunay triangulation merger
  - (a) Merging Algorithm Divide and Conquer
  - (b) A recursive algorithm for cutting diameter
  - (c) Bandpass fusion algorithms
3. Algorithms for direct construction of the Delaunay triangulation
  - (a) Incremental algorithm

(b) Step by step algorithms with acceleration of searching Delaunay's neighbors

4. The two-pass algorithms for constructing a Delaunay triangulation

(a) Two-pass fusion algorithms

(b) Modified hierarchical algorithm

(c) Linear algorithm (d) Fan algorithm

(d) The algorithm of recursive splitting

(e) Belt algorithm

Now, many people are working on improving and creation of new algorithms. Main reason it is instability of known algorithms and unsatisfactory time of their work on real data sets.

The following there are some of the faster and simple for implementation algorithms for constructing the Delaunay triangulation.

### **2.2.1.1. Iterative algorithms**

All iterative algorithms based on the idea of sequential addition of the points in the partially constructed Delaunay triangulation.

Let there be the Delaunay triangulation on the set of  $(n-1)$  points. Another  $n$ -i point added to the already constructed structure of a triangulation as follows.

*Step 1.* First produced localization point, that is, It is a triangle (built before), which gets the next point. If the point does not fall into the triangulation, it means that triangle is on the border triangulation is closest to the next point.

*Step 2.* If the point came on the previously inserted triangulation node, then this point discarded, otherwise the point inserted into the triangulation as a new node. Thus, if the hit point on an edge, it is splitting into two new and two adjacent triangle edge it is also divided into two smaller ones. If hit point inside any triangle then it is split into three new. If the point came out of triangulation, the builds one or more triangles. Then, occur local checks of new triangles for conformity the Delaunay condition, and performed the necessary constructions.

The complexity of the algorithm consists of the complexity of the search triangle, which is added to the next step point, the complexity of the construction of the new triangles, as well as the complexity of the relevant restructured triangulation that is result of unsatisfactory inspections pairs of neighbors triangles obtained triangulation to conformity the Delaunay condition.

In the construction of the new triangles are two possible situations: the added point falls inside, or outside of triangulation. In the first case, making creation of new triangles, and the number of operations performed by the algorithm is fixed. In the second - necessary to build additional external triangles to current triangulation, and their count may be equal in the worst-case  $n-3$ .

However, for all the steps of the algorithm will be added at most  $3N$  triangles, where  $N$  - the total number of starting points. Therefore, in both cases, the total time spent on the construction of triangles  $O(N)$ .

Any addition of a new point in the triangulation could theoretically compromise the integrity of the Delaunay condition, so after adding points usually immediately made local check on the condition of the Delaunay triangulation. This check should cover all newly constructed triangles and adjacent to them. The number of such evolutions in the worst case can lead to a complete rebuild entire triangulation, so the complexity of rebuilding is  $O(N)$ . However, the average number of these pas rebuilding the actual data is only about three [24].

### **2.2.1.2. Simple iterative algorithm**

In a simple iterative algorithm search of the next triangle is implemented as follows. Take any triangle, already owned by triangulation (eg, randomly selected), and consecutive transitions on connected triangles searched the desired triangle. In the worst case, we have to cross all the triangles of the triangulation, so the complexity of such a search is  $O(N)$ . However, on average to evenly distribute the squared only necessary to make  $O(\sqrt{N})$  transition operations. Thus, the complexity of

the iterative algorithm is the simplest in the worst case  $O(N^2)$ , and an average -  $O(N^{3/2})$  [27].

### 2.2.1.3. Caching algorithms search triangles

When implementing these algorithms is based cache - a special structure, which allows for time (1) to find a triangle, close to the target. In this modified triangles from the cache is not removed (assuming that each remote triangle as a record in the computer memory is converted into a new triangle and therefore permissibility reference to the triangles is not affected by the algorithm), the same triangle can repeatedly be in the cache, and some triangles may be absent at all there.

The basic idea of the cache is to build a more simple planar partition of the plane than the triangulation, in which you can quickly perform localization points. For each element, a simple reference is made to partition the triangle of triangulation. The search process is reduced to a simple localization element-partitioning, click on a link to the triangle and the subsequent localization of the desired triangle algorithm of a simple iterative algorithm of Delaunay triangulation. As such partitioning is easiest to use a regular square network [24].

### 2.2.1.4. The iterative algorithm to the static cache search

The triangulation algorithm to search a static quotation must select a number  $m$  and make cash in the form of a two-dimensional array  $r$  size of  $m \times m$  to refer to the triangles. Initially, the array must complete the link to the very first constructed a triangle. Then, after performing another search, which was found by a triangle, starting the search with the square  $(i, j)$ , you need to update information in the cache:  $r_{ij} := \text{link to } T$ . The size of the static cache should be selected according to the formula  $m = s \times N^{3/8}$ , where  $s$  - coefficient of static cache. In practice, the value of  $s$  should take  $\approx 0,6 - 0,9$  [33].

The first time, the cache is not updated completely, the search can go for a long time, and then speed are increased. This shortage is deprived of the following algorithm.

### 2.2.1.5. The iterative algorithm with dynamic caching search

The triangulation algorithm with dynamic caching search you must have a minimum size of the cache, such as  $2 \times 2$ . As the number of points added to the triangulation necessary to consistently increase its size by half, thus overwriting the information from the old to the new cache. At the same time it is necessary to perform the following shipment to increase the cache

( $r$  - the old cache,  $\overline{r^I}$  - new cache):  $\forall i, j = 0, m - 1$ :

Increase in dynamic cache size must be carried out twice every time when the number of points in the triangulation  $n = r \times m^2$ , where

$r$  - dynamic cache growth factor, and  $m$  - the current size of the cache. In practice, the value of the coefficient of dynamic growth in the cache, select  $\approx 3 - 8$ .

Labor-intensive triangulation algorithms for caching like all iterative algorithms constitute the worst case  $O(N^2)$ , and an average of even distribution of static caching -  $O(N^{9/8})$  and for dynamic caching -  $O(N)$  [32].

For the majority of random distributions starting points, this algorithm is significantly faster than any other algorithms. However, on some real data in which sequential starting points are close to each other (for example, in terms of relief contour maps), dynamic caching algorithm may spend more time than other algorithms.

### 2.2.1.6. Iterative Algorithm with layer thickening

You must break the plane with a point in the iterative algorithm triangulation stratified condensation on  $n = (2^u + 1) \times (2^v + 1)$  Elementary cell-squares of equal

size. Each square is numbered from 0 before  $2^u$  and horizontally from 0 before  $2^v$  vertically. Next, introduce the concept of layers. It is believed that the point belongs to layer  $i$ , if both numbers are multiples of its square  $2^i$  (then all the reference points form a layer 0, layer  $i + 1$  It is a subset of the layer  $i$ , and the maximum number of the layer  $k = \min(u, v)$ ). According to numbers of pairs of values of all the points  $i$  layer are dividing into four subsets:

1. The corner points (both their numbers are multiples  $2^{i+1}$ ) - this layer  $i + 1$ ;
2. interior points (both their number is not a multiple of  $2^{i+1}$ );  
 $rI \quad I \quad I \quad I$   
 $2i, 2j, r2i, 2j+1, r2i+1, 2j, r2i+1, 2j+1 := ri, j.$

This caching algorithm allows working equally effectively on a small and a large number of points in advance without knowing their number.

3. X- boundary points (only the coordinate X number is a multiple of  $2^{i+1}$ );
4. Y- boundary points (only the Y-coordinate is a multiple number  $2^{i+1}$ ).

For arbitrary sets of points of this algorithm (like all other iterative algorithms) has the complexity of  $(N^2)$ . If the reference points have a uniform distribution, the complexity of the algorithm is the average case  $O(N)$ .

The advantage of this algorithm is more and what is uniformly consistent in sticking triangulation nodes, it is possible to get rid of the situations the construction of long narrow triangles which are then rearranged. Due to this, the faster the algorithm on real data than many other algorithms.

## 2.2.2. Cell triangulation algorithms

### 2.2.2.1. Algorithms "marching cubes", Caneira and Rocks

The earliest construction of the triangulation algorithm cell surface algorithm is "marching cubes" Lorensen submitted in 1987.

The algorithm proposed by Caneira (also called "marching tetrahedra 5"), based on the partition of the space into triangular pyramids.

Rocks algorithm developed for visualization three-dimensional scalar fields defined by a function defined at each point of space.

These algorithms discussed in the sections "marching cubes method", "Caneira Method" and "Method of the Rock" [34].

### 2.2.2.2. Marching tetrahedrons 6 algorithm

Marching tetrahedrons 6 is algorithm that proposed as an alternative Guezekom Caneira algorithm. The main difference between these two methods is that for MT6 algorithm eliminates the need to change the templates directly to the mirror and back. This achieved by partitioning a symmetrical tetrahedron box 6 shown in Figure 2.2.

Comparisons can be conducted on 4 criteria:

- The speed of work;
- Error of approximation;
- The number of generated triangles;
- The quality of the generated triangles.

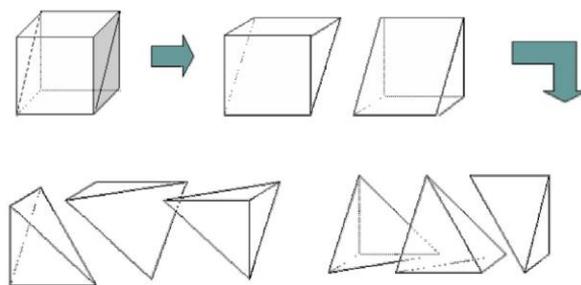


Figure a.2 - Split the box into 6 tetrahedrons

Thus, the honeycomb type discussed algorithms have the same basis. Therefore, the speed and the approximation error they differ insignificantly. The complexity of these algorithms is  $O(N)$ . Therefore, they are enough to compare the two parameters: the number of triangles and the "quality".

### 2.2.2.3. Marching prism algorithm

The region of space filled prisms with regular triangles at their base.

Perhaps the prism 64 and the intersection of the case surface, which is stored in a special table. These 64 cases can be reduced to 9, using turns and symmetry relative to a plane parallel to the base of the prism and passing through the middle of the vertical edges as well as the possibility of applying the sign function. Figure 2.3 shows all the 9 cases. It is noted the top, on which the function has a positive value, a negative value on the other vertices.

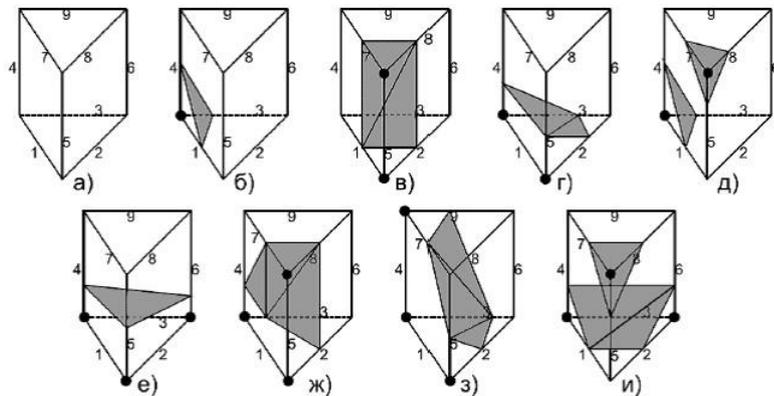


Figure 2.3 - Possible cases of the intersection of the prism and the surface.

### 2.2.2.4. Triangulation five-vertex pyramid

The region of space divided into cubic cells. Each cell is divided into 6 five top of the pyramid: the pyramids are the vertices of the cell center and at the top with 4 of each of the 6 faces of the cube. Next, the triangles constructed at the intersection of the pyramid and the surface. Possible variants of intersection 32 and the surface of

the pyramid, which can be reduced to 6 nonequivalent, as at the pyramid vertex 5. Figure 2.4 presents 6 non-equivalent variants. It is noted the top, on which the original function takes positive values. For a software, implementation of the triangulation method uses the numbering of edges in the method of the Rock and the table of the 32 cases, the intersection of the pyramid and the surface [35].

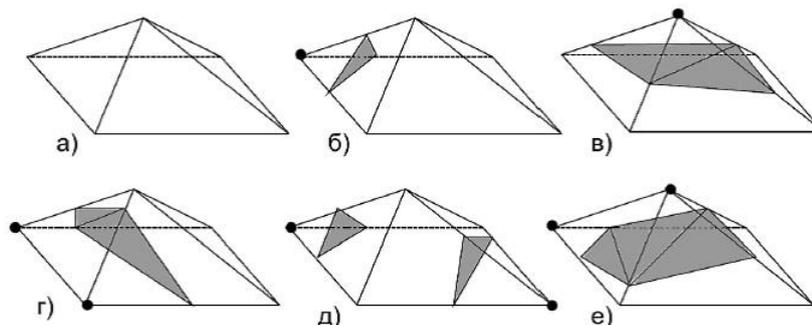


Figure 2.4 - Options intersection of the pyramid and the surface.

#### 2.2.2.5. Marching octahedrons algorithm

The region of space filled octahedrons. Filling carried out in the following manner: a region of space filled with cubes and cubes building rib from the centers of their peaks. The result is a network of octahedrons, in which eight ribs - ribs is constructed for each face of the cube, with its center and adjacent to the center of the cube it. Figure 2.5 presents options for surface intersection and octahedron.

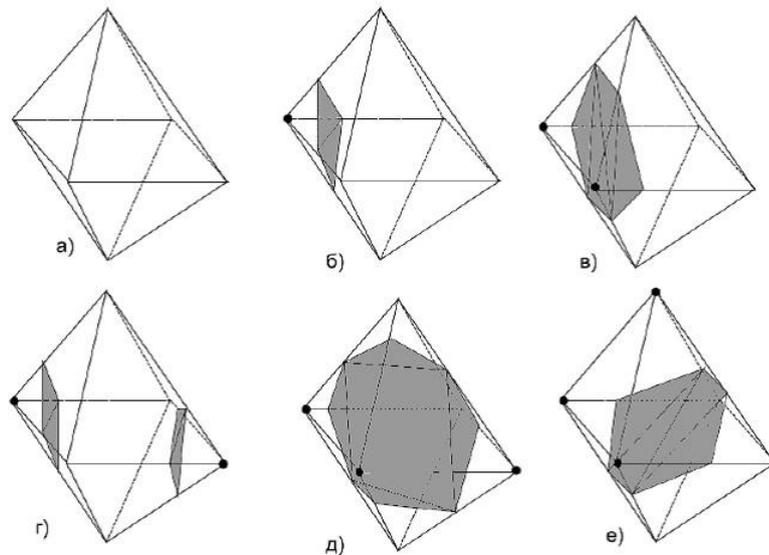


Figure 2.5. Options intersection of the pyramid and the surface.

After filling in the field of space octahedrons the approximation triangles intersection of the surface and each octahedron [23].

### 2.2.3. Triangulation extra-large surfaces

As already mentioned, and shows significant problem cell algorithms is their indifference, that is, the inability to always correctly visualize the behavior of the surface inside the cell. Reducing the size of the cell can solve the problem, but then there is another - lack of memory. The surface, the raw data that do not fit in the computer's memory, called ultra. If the surface does not fit into memory, the video memory cannot fit it even more. Therefore, such a surface visualization by conventional methods takes a very long time.

Conventional algorithms for constructing a Delaunay triangulation as a method of surface modeling and visualization are not suitable for large volumes of raw data.

One of the first attempts to work with large triangulations can be as simplified algorithms for constructing triangulations.

The task of constructing simplified triangulation is also NP-hard. Therefore, in practice it uses approximate algorithms that divided in accordance with the strategy used in two main classes, working from top to bottom and bottom to top.

Top-down strategy starts with a simple approximating model consisting of one or more triangles, covering the original triangulation. Further, the triangulation consistently adding new points to until you reach the desired resolution. One of these algorithms is the algorithm of «selector of Delaunay».

In bottom-up, strategy begins with the initial operation of triangulation, triangulation number of elements and gradually decreases until a desired number of nodes reached or a predetermined tolerance from the initial triangulation simplified.

Algorithms for constructing and visualizing extra-large surfaces allow interactively work with models, significantly exceeding the amount of available RAM.

#### **2.2.4. Adaptive triangulation**

An adaptive triangulation method is to make the level of detail (and pitch obtained mesh) is inversely proportional to the distance to the observer.

On the other hand, because the triangulation is not adapted to the local features of the surface, it is overly excessive. A more significant drawback of the method is that it cannot guarantee the specified accuracy of the approximation, which will be especially noticeable on complex surfaces.

In the methods of M. Gross (M. Gross) et al., And A. Pereberin wavelet coefficients are used only for construction of adaptive triangulation, and no compression is applied. More importantly, the methods described cannot achieve a desired restoration accuracy. As a solution to this problem, we propose a two-step method of coding adaptive triangulation, in which the wavelet coefficients form the basic information, which complemented by the residual information, ensuring that the adaptive triangulation approximates the surface with a given error [34].

In the first stage of building a compressed representation of the original multi-scale grid heights sequentially filtered and decimated, resulting in a multi-level pyramid, each level that is a grid with increasing twice-sampling interval in each direction, approximating the original with decreasing accuracy.

Each level of the pyramid divided into square blocks having the same number of elements that combined in a hierarchical structure quadtree. Built hierarchy is compressed downward recursive algorithm starting from the blocks at the very rough approximation level (level 0). In addition, for each block of encoded information that allows us to refine its net heights and get the elevation grid blocks 4 subsidiaries.

The matching area is composed of several loops. To ensure consistency grids heights only one contour, but to elements of normal maps on the borders calculated the same way, it needs more. Each circuit compressed without using the wavelet transform. Elements of the circuit regarded as one-dimensional sequence of samples of the high places that are quantized and encoded [28].

### **2.2.5. Algorithm of the Finch and Bishop.**

The algorithm can significantly reduce storage costs, as calculates the net for the current position of the camera without the use of a pre-existing data. The effectiveness of the algorithm achieved by using a regular grid and the use of pre-defined simple ways to coarsening when removing the camera from the display surface area. Reduced storage costs is due to the calculation of the grid for the current position of the camera without the use of a pre-existing database.

The algorithm uses the concept of the level of detail (LOD). Each LOD level on the surface is given a uniform square grid. Step partition your for each level, and is a power of 2:  $LOD = 0$  for the move is the minimum for  $LOD = 1$  each cell side length twice bigger than  $LOD = 0$ , etc. Cells for  $LOD = LOD_{max}$  are the most rude and partition called pages. The number of levels is a parameter of the algorithm.

The result is a triangulation algorithm, presented in the form of a sequence *strip*. Each individual *strip* directed along the axis  $Ox$  surface (minor route), and the

sequence *strip* ordered along the axis  $Oy$  (majeure direction). In the process of triangulation occurs processing is not the entire surface but only the area visible at the moment on the screen. The projection of this area onto the plane  $Oxy$  named *footprint* [23].

### **2.2.6. Modern trends in research and development**

Trivial task in the construction of three-dimensional models of range sequence of images presented in the form of individual triangulation grids (shells), is the problem of combining these triangulation grids. Its solution involves the construction of a single coherent shell, corresponding to the original data, without duplicating the surface areas that are visible from several species. An important requirement of the algorithmic solution of the problem is the high processing speed, since saturated scenes with complex objects can take many forms with many describing the triangles (about  $10^5$ ,  $10^6$  and above).

At present, the emphasis in increasing the performance of applications shifted to the active use of parallel systems and coprocessors. Increased productivity is achieved through the extensive use of massive vector parallelism or pipelining and specialization thanks calculators, customized for a specific task.

The superiority of the conventional stream processors in the tens of times the computing power makes it a promising treatment is two-dimensional images on the stream processors.

Modern graphics API (OpenGL, DirectX) allow you to program the stream processors in high-level language (GLSL for OpenGL and HLSL for DirectX), which greatly simplifies the design, reduces the time.

Until recently, relatively few attempts undertaken paralleling recovery algorithms surfaces of which method it is worth noting that use as the octree data structure for storing the processed pixels. There is also a reconstruction algorithm based on Poisson surfaces using adaptive mesh as accelerating structure.

Currently, one of the methods implemented a parallel algorithm for constructing a Delaunay triangulation in the plane using the graphics processor is considered. In this paper, we consider technology NVIDIA CUDA.

There is a parallel algorithm resurfacing for massively parallel architectures. The basis of the developed algorithm marching cubes algorithm is taken.

Promising also is the idea of the use of stream processors for image processing in the browser-based applications due to their limited access to the computing power of the CPU.

For a long time image processing in the browser-based applications can be implemented using two major approaches: using java-applet and using Flash-component. For example, in image processing is widely used java- applets ImageJ library.

With the advent of HTML5 standard made possible image processing directly on javascript language using the canvas element. The advantage lies in the absence of the need to install any plug-ins, but this method is difficult to use multi-core, and an image processing program is running on a virtual machine javascript. These factors have a negative impact on the performance approach: its speed is several times less than the image processing speed by using ImageJ.

Currently, widespread PixelBender technology implemented in Adobe Flash since version 10.0. This technology allows to a limited extent to use the possibility of stream processors, developing programs for them in a special language specific to Flash.

Implementation of OpenGL ES 2.0 for use in browser applications - February 10, 2011 open standard WebGL specification was published. WebGL provides an opportunity to develop OpenGL-programs directly on javascript, including the use of vertex and pixel shaders.

A significant advantage in favor of WebGL is still the fact that its use does not require the installation of additional modules, technology support is provided directly in the browser. All this makes it a promising technology for image processing organization in browser applications [30].