
**ХІ Республіканская навучна-практычная канферэнцыя-конкурс
навучна-даследавальскіх работ учащихся средних,
средних специальных учебных заведений и студентов вузов
«От Альфа к Омеге...» (с международным участием)
Секция 3. Компьютерные науки и программирование
НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЕ РАБОТЫ ШКОЛЬНИКОВ**

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Государственное учреждение образования «Средняя школа № 23 г. Гродно»

СОЗДАНИЕ ПРОГРАММЫ ПО ВИЗУАЛИЗАЦИИ ОБЪЕМНЫХ ДАННЫХ

Чичкан Алексей Михайлович,

учащийся 8 «П» класса

Лещук Сергей Станиславович,

учитель информатики

ГУО «Средняя школа № 23 г. Гродно»,

первая кв. категория учителя

Гродно, 2021

**XI Республиканская научно-практическая конференция-конкурс
научно-исследовательских работ учащихся средних,
средних специальных учебных заведений и студентов вузов
«От Альфа к Омеге...» (с международным участием)
Секция 3. Компьютерные науки и программирование
РЕФЕРАТЫ НАУЧНО-ИССЛЕДОВАТЕЛЬСКИХ РАБОТ ШКОЛЬНИКОВ**

СОЗДАНИЕ ПРОГРАММЫ ПО ВИЗУАЛИЗАЦИИ ОБЪЕМНЫХ ДАННЫХ

А. М. Чичкан

*ГУО «Средняя школа № 23 г. Гродно», 8 «П» класс,
Гродно, Беларусь*

Научный руководитель – С. С. Лещук, учитель информатики ГУО «Средняя школа № 23 г. Гродно», первая кв. категория учителя.

Работа 14 с., 2 ч., 10 рис., 6 источников, 1 прил.

«Ключевые слова:» - компьютерная томография, сканирование, программа «DicomViewer», срезы, полигон, объемное изображение, библиотека DirectX.

Объект исследования: алгоритмы и библиотеки визуализации объемных данных.

Цель работы: создание программы для визуализации данных, полученных с томографа с использованием алгоритма "Marching cubes" и графической библиотеки OpenGL.

Методы проведения работы: анализ информационных источников, сравнение, компьютерный эксперимент.

Результаты работы: создана программы для визуализации данных, полученных с томографа с использованием алгоритма "Marching cubes" и графической библиотеки OpenGL.

Область применения: образование, компьютерная графика, медицина.

Практическая значимость работы: разработанная программа может быстро провести визуализацию данных

Прогнозные предположения о развитии объекта исследования: включить изучение алгоритма "Marching cubes" и графической библиотеки OpenGL в программу занятий с одаренными учащимися школы; использование созданной программы для визуализации данных для обучения учащихся на уроках биологии.

ОГЛАВЛЕНИЕ

Введение.....	4
Основная часть.....	5
1 Обзор литературы	5
1.1. Что такое томография?.....	5
1.2. Компьютерная томография.....	5
2 Практическая работа по созданию программы по визуализации объемных данных.....	6
Заключение.....	13
Список использованных источников	14
Приложения.....	15

ВВЕДЕНИЕ

В различных областях медицины все более широкое распространение получают сегодня методы визуализации внутренних структур человека на основе данных томографических исследований. Эти методы компьютерной графики называются визуализацией объема.

На протяжении двух последних десятилетий технология получения изображений внутренних органов человека (интроскопия) пережила ряд принципиальных изменений. Ранее в распоряжении врачей были лишь рентгеновские снимки, которые давали некоторое представление об исследуемых органах в виде наложения теней на изображениях. Эти изображения отличались плохой контрастностью и отсутствием какой-либо информации о глубине объектов. Использование компьютеров дало возможность развиваться новым направлениям томографической интроскопии, таким как компьютерная томография (CT-computed tomography), магнитная резонансная томография (MRI-magnetic resonance imaging) и позитронная эмиссионная томография (PET-positron emission tomography). С помощью томографической аппаратуры можно получить снимки множества сечений тела пациента, которые характеризуют особенности его анатомии и физиологии. Эти снимки с чрезвычайной четкостью показывают различные органы, причем изображения органов не налагаются друг на друга. Математические методы позволяют реконструировать трехмерную структуру органов по множеству параллельных сечений.

Цель использования метода визуализации объема в медицине - создание точных и реалистичных визуальных представлений объектов по медицинским данным. Получаемые в результате изображения, даже если они по-существу двухмерные, часто называют 3D изображениями или 3D реконструкциями для того, чтобы отличать их от 2D сечений или обычных рентгеновских снимков.

Цель работы: создание программы для визуализации данных, полученных с томографа с использованием алгоритма "Marching cubes" и графической библиотеки OpenGL

Объект исследования – алгоритмы и библиотеки визуализации объемных данных.

Задачи:

- Изучить алгоритм "Marching cubes" и графическую библиотеку OpenGL.
- Использовать алгоритм "Marching cubes" для генерации полигонов изображения по трехмерной модели.
- Использовать библиотеку OpenGL для визуализации полученных полигонов.

ОСНОВНАЯ ЧАСТЬ

1 ОБЗОР ЛИТЕРАТУРЫ

1.1. Что такое томография?

Томография (др.-греч. *τομή* — сечение) — получение послойного изображения внутренней структуры объекта.

Анатомическая, или разрушающая томография основана на физическом выполнении срезов исследуемого организма с их последующей фиксацией с помощью химических веществ.

Реконструктивная, или неразрушающая томография — получение тем или иным способом информации о распределении интересующего параметра в объекте большей размерности по его проекциям меньшей размерности без разрушения объекта; антоним анатомической томографии.

1.2. Компьютерная томография

Компьютерная томография — метод неразрушающего послойного исследования внутреннего строения предмета, был предложен в 1972 году Годфри Хаунсфилдом и Алланом Кормаком, удостоенными за эту разработку Нобелевской премии. Метод основан на измерении и сложной компьютерной обработке разности ослабления рентгеновского излучения различными по плотности тканями. В настоящее время рентгеновская компьютерная томография является основным томографическим методом исследования внутренних органов человека с использованием рентгеновского излучения.

Изображения, полученные методом рентгеновской компьютерной томографии, имеют свои аналоги в истории изучения анатомии. В частности, Николай Иванович Пирогов разработал новый метод изучения взаиморасположения органов оперирующими хирургами, получивший название топографической анатомии. Сутью метода было изучение замороженных трупов, послойно разрезанных в различных анатомических плоскостях («анатомическая томография»). Пироговым был издан атлас под названием «Топографическая анатомия, иллюстрированная разрезами, проведёнными через замороженное тело человека в трёх направлениях». Фактически, изображения в атласе предвосхищали появление подобных изображений, полученных лучевыми томографическими методами исследования.

Разумеется, современные способы получения послойных изображений имеют несравнимые преимущества: нетравматичность, позволяющая проводить прижизненную диагностику заболеваний; возможность аппаратного представления в различных анатомических плоскостях (проекциях) однократно полученных «сырых» КТ-данных, а также трёхмерной реконструкции; возможность не только оценивать размеры и взаиморасположение органов, но и детально изучать их структурные особенности и даже некоторые физиологические характеристики, основываясь на показателях рентгеновской плотности и их изменении при внутривенном контрастном усилении.

Современный компьютерный томограф представляет собой сложный программно-технический комплекс. Механические узлы и детали выполнены с высочайшей точностью. Для регистрации прошедшего через среду рентгеновского излучения используются сверхчувствительные детекторы. Конструкция и материалы, применяемые при их изготовлении, постоянно совершенствуются. При изготовлении компьютерного томографа предъявляются самые жесткие требования к рентгеновским излучателям. Неотъемлемой частью аппарата является обширный пакет программного обеспечения, позволяющий проводить весь спектр компьютерно-томографических исследований (КТ-исследований) с оптимальными параметрами, проводить последующую обработку и анализ КТ-изображений.

2 ПРАКТИЧЕСКАЯ РАБОТА ПО СОЗДАНИЮ ПРОГРАММЫ ПО ВИЗУАЛИЗАЦИИ ОБЪЕМНЫХ ДАННЫХ

Результаты сканирования современными томографами представляются в виде набора отдельных файлов, в которых записаны отсканированные "срезы" в формате DICOM - специальный медицинский формат. Если в обычной фотографии характеристиками точки изображения являются три цвета - rgb - т.е. три цифры, представляющие интенсивность цветов данной точки, то в DICOM "срезах", полученном с томографа, характеристикой точки является только одна цифра - некая "плотность" вещества определяемая томографом с высокой точностью - 64000 градаций в отличие от 256 значений цвета в обычной фотографии. Все ткани человека, или материалы предмета, имеют свою плотность, определяемую томографом. Чтобы увидеть "срез" наглядно, надо "плотность" преобразовать к какому-нибудь цвету по некоторому алгоритму. Тогда можно будет увидеть как обычную разноцветную фотографию среза, в которой разным тканям будут соответствовать разные цвета. Так же можно задать чтобы отображались только ткани одного, или нескольких видов. DICOM является сложным форматом и его разбор выходит за рамки этой работы.

Мы воспользовались свободной программой "DicomViewer", чтобы преобразовать образцовый набор срезов отсканированной томографом бутылки в изображения формата BMP (см. рисунок 2.1). Но гораздо нагляднее сделать из срезов объёмное изображение. Это и есть наша цель.

Таким образом, мы имеем набор изображений в формате BMP, соответствующий срезам предмета через несколько миллиметров:

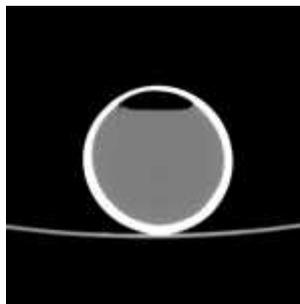


Рисунок 2.1 – Снимок бутылки томографом.

Чёрный цвет соответствует воздуху, белый - стеклу, серый - жидкости в бутылке.

Наша задача - составить из этих изображений трехмерный массив-модель. Далее мы должны получить поверхность, соответствующую какому-либо веществу. Мы выбрали для визуализации стекло (хотя можно выбрать и жидкость, а стекло сделать "невидимым").

Из этих изображений составляем трехмерный массив, где элемент массива - это цвет данной точки из набора изображений. Напомню, что цвет соответствует определённой "плотности" вещества из DICOM-файла. Далее, из этого массива мы формируем новый массив, где каждой точке соответствует 1 или 0. Точки с 1 соответствуют части, которая отображается, а точки с 0 соответствуют невидимой части. В нашем случае, когда отображаться должно только стекло, точки белого цвета переходят в единицы, а все остальные (чёрные и серые) в нули. Получается, что мы имеем объёмный массив, который состоит из кубиков в вершинах которых находятся нули и единицы. Если в одной вершине "кубика" ноль (воздух), а в другой вершине 1 (стекло), то значит между этими вершинами должна проходить поверхность стекла. Всего у "кубика" восемь вершин и возможны 256 комбинаций нулей и единиц в вершинах "кубиков" от 00000000 до 11111111 - где каждая позиция соответствует определённой вершине. Каждой комбинации соответствует определённая отображаемая поверхность - полигон. Мы плавно подошли к алгоритму "Marching cubes", который как раз и решает такую задачу (см. рисунок 2.2):

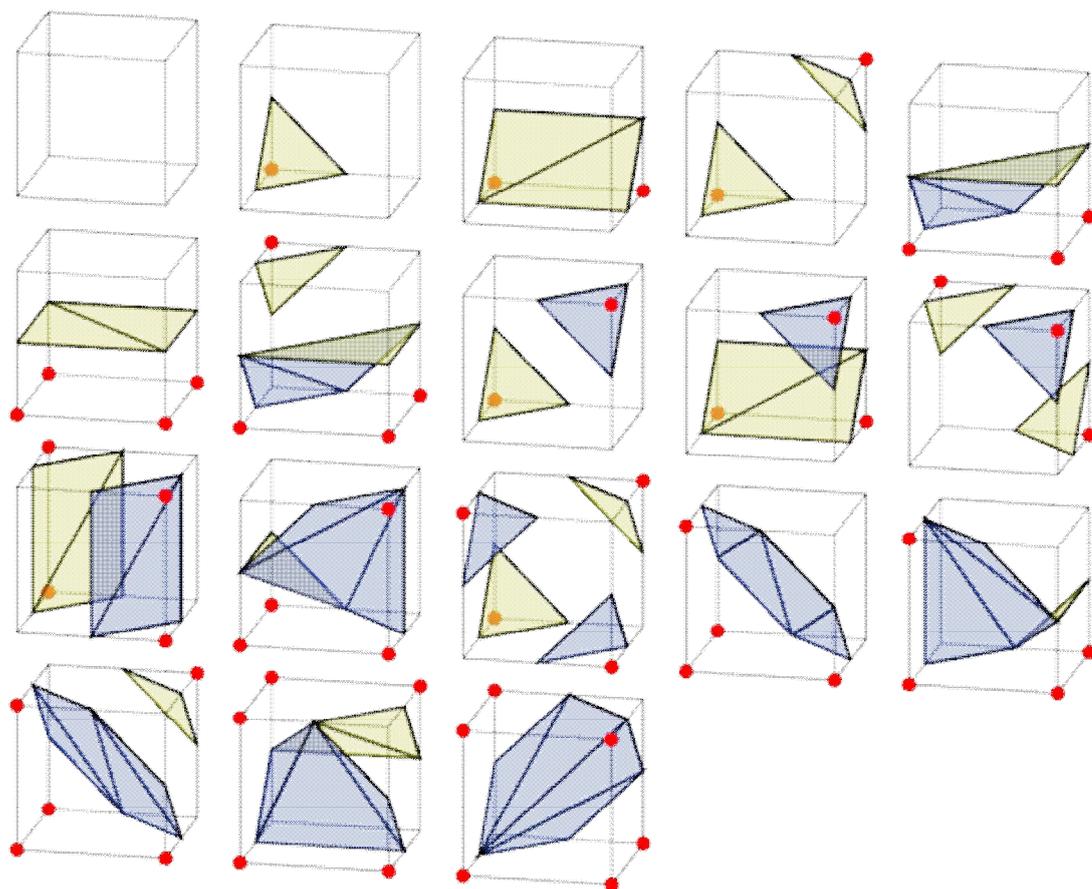


Рисунок 2.2 - Отображение полигонов в "Marching cubes".

(<https://habr.com/ru/post/358658/>)

Обычно этот алгоритм используют для рисования поверхностей объёмных функций $f(x,y,z)$ определённого значения (эквипотенциальных поверхностей).

Суть этого алгоритма в том, что мы просто имеем массив с индексом, который пробегает значения от 0 до 255 в десятичной системе счисления (или от 00000000 до 11111111 в двоичной системе счисления) и соответствует одной из возможных комбинаций нулей и единиц в вершинах куба. А значения этого массива - это вершины полигона (многоугольника), вписанного в заданный куб, и являющегося отображаемой поверхностью при данной комбинации нулей и единиц.

Таким образом, из изображений мы составляем массив цветов точек, далее преобразуем этот массив в массив единиц и нулей, соответствующих видимости и невидимости соответственно. Затем, из этого массива при помощи алгоритма "Marching cubes" получаем набор полигонов, которые нужно отобразить. Вот мы и подошли к использованию библиотеки OpenGL - одной из самых известных графических библиотек, используемых при создании игр и пакетов трёхмерной графики и анимации. Мы просто инициализируем "сцену" OpenGL, задаём источник света, при помощи функций OpenGL загружаем туда наши полигоны и отображаем их. Всё остальное делает библиотека OpenGL.

Была написана программа в среде разработки Delphi 7 (см. рисунок 2.3):



Рисунок 2.3 – Программа Delphi 7.

При помощи кнопки "Load Bottle" загружается набор изображений из папки scan, соответствующих сканам бутылки, и формируется трёхмерный массив из точек этих изображений (см. рисунок 2.4).

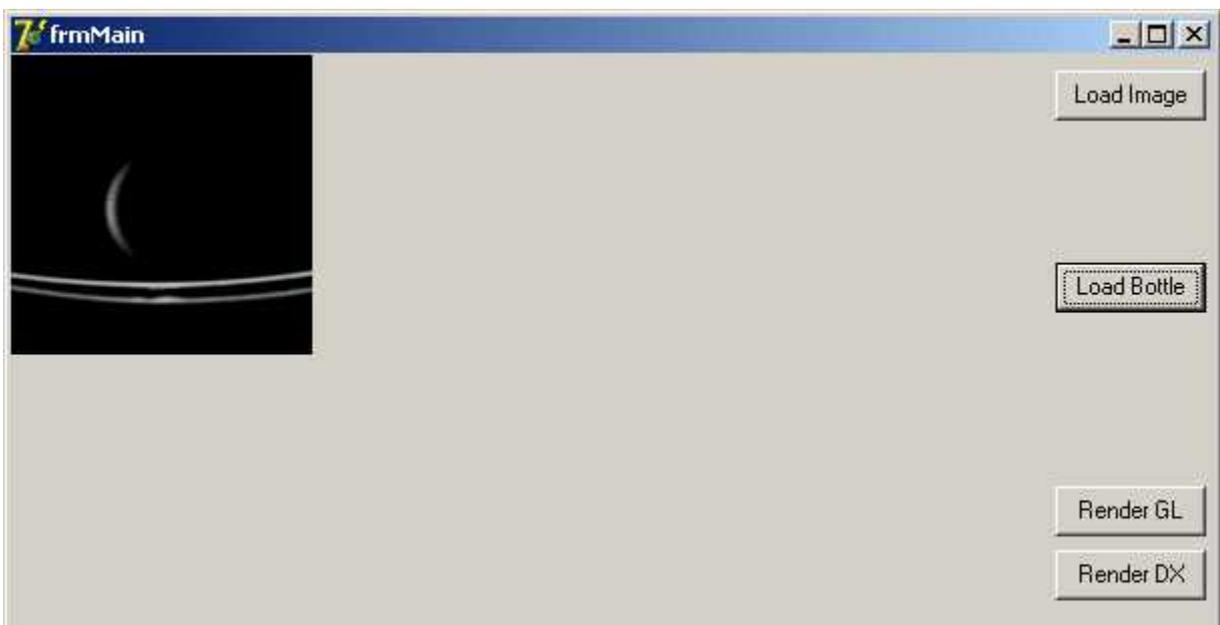


Рисунок 2.4 – Формирование трехмерного массива в Delphi 7.

При нажатии кнопки "Render GL" откроется "сцена" OpenGL, в которой будет визуализирована бутылка (см. рисунок 2.5):

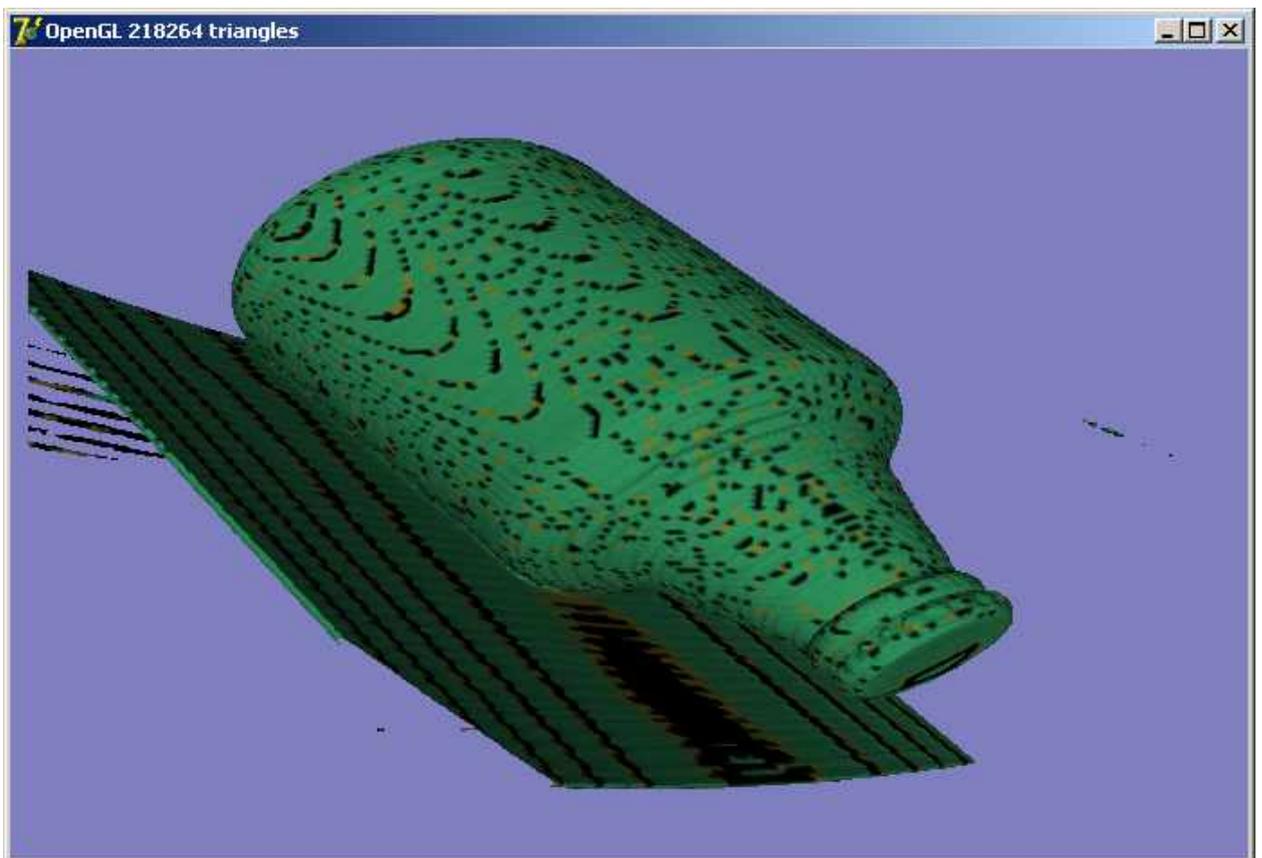
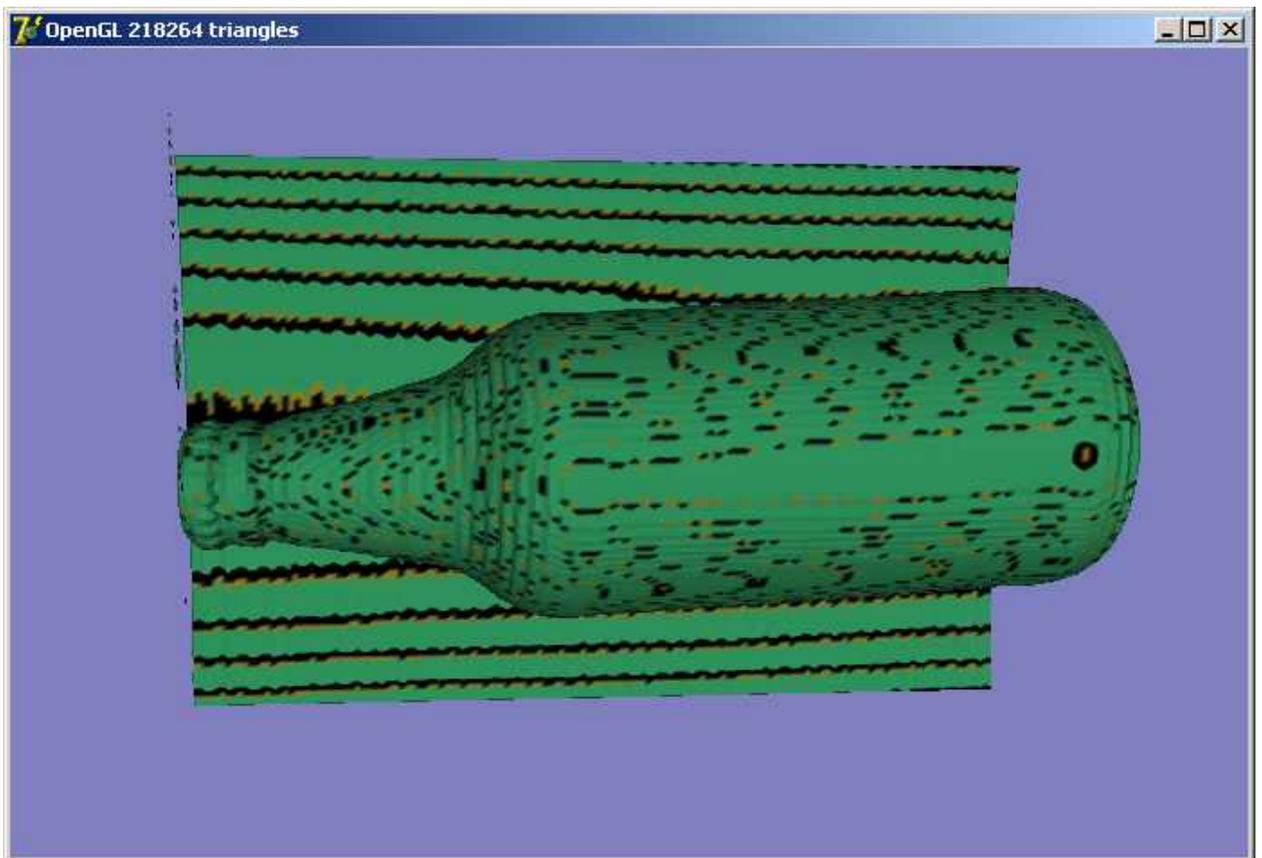


Рисунок 2.5 - Визуализированная бутылка.

В заголовке выводится число полигонов. В данном случае модель бутылки состоит из 218264 полигонов. При помощи кнопок WASD изображение бутылки можно вращать в разных плоскостях:

При помощи кнопки "Load Image" можно загрузить произвольное изображение, из которого потом сформируется трёхмерный "цилиндр" (см. рисунок 2.6):

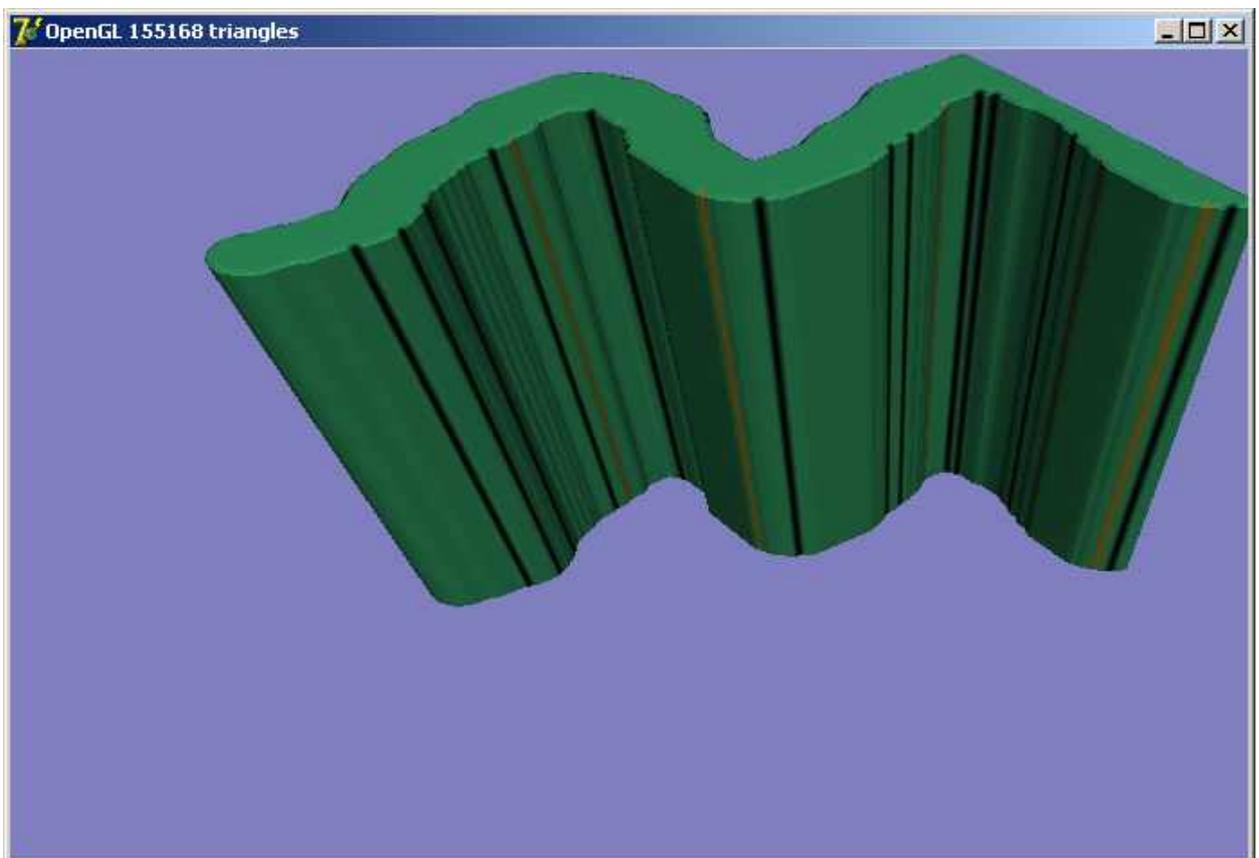


Рисунок 2.6 - Визуализация произвольного изображения.

Или вот так (см. рисунок 2.7):

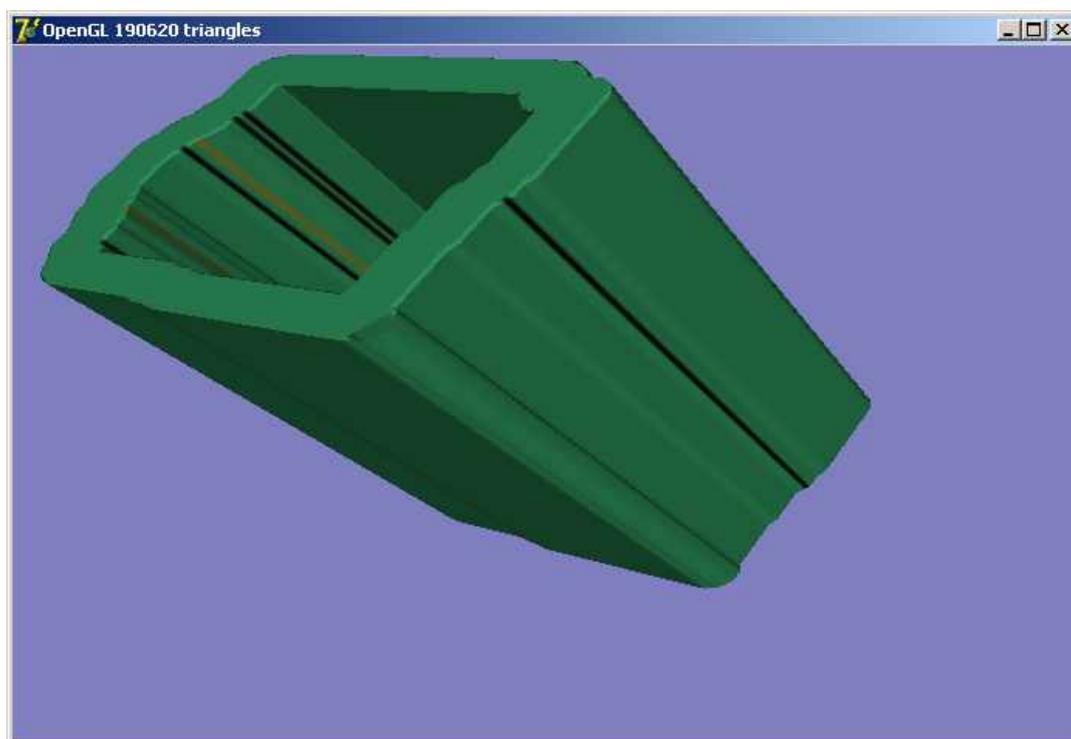
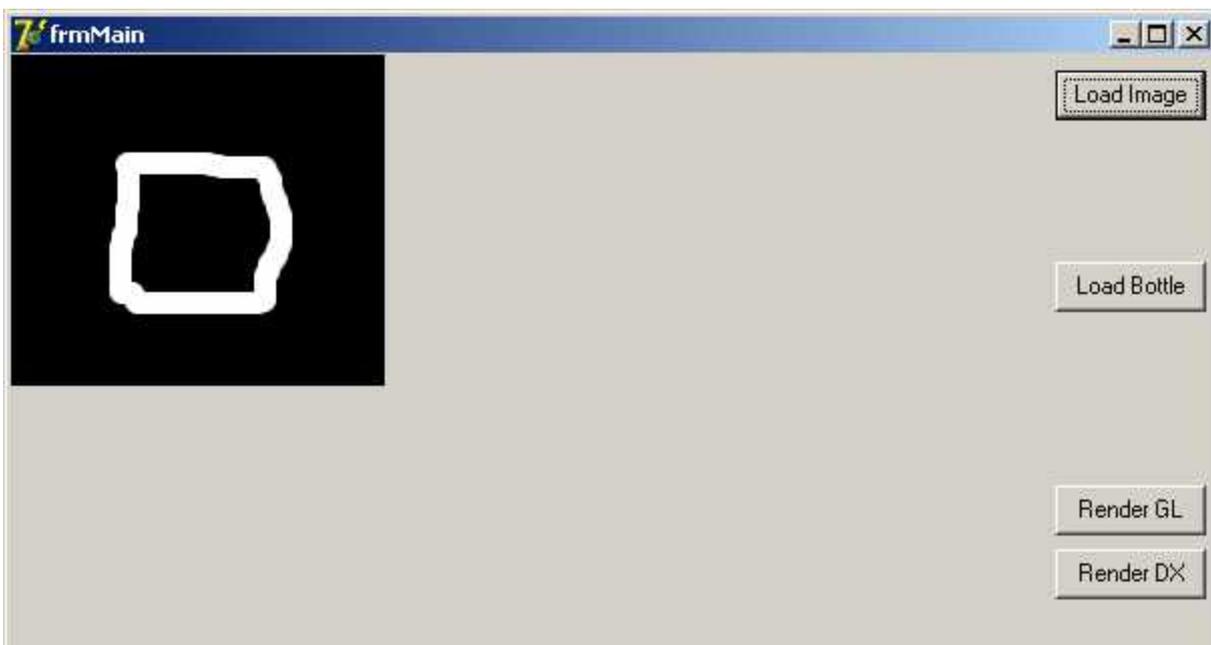


Рисунок 2.7 - Визуализация произвольного объекта.

Так же в программе есть кнопка "Render DX", которая реализует визуализацию модели при помощи библиотеки DirectX, но не напрямую, а через прослойку компонент "DelphiX". К сожалению на практике оказалось, что при большом числе полигонов компоненты "DelphiX" работают ужасно медленно и годятся для использования только с моделями до нескольких тысяч полигонов, а для визуализации моделей с сотнями тысяч полигонов совершенно неприменимы.

Для визуализации одного предмета с помощью разработанного приложения используются 148 изображений размерностью 150x150 пикселей, которые вписываются в объёмную решётку размерностью 150x150x150. В режиме загрузки одиночного изображения оно дублируется на 148 слоёв решётки. То есть нужно 148 изображений. Но эту цифру легко изменить - она определяется только числом изображений в бывшем в наличии скане бутылки и вопросами производительности и визуального качества картинки

Практическая значимость.

В перспективе, после усовершенствования и доработки, эту программу можно применять в медицине для визуализации результатов томографического сканирования для отображения отдельных частей организма человека - например скелета, сосудистой системы и т.д.

Для этого необходимо:

1) Добавить возможность считывать непосредственно сканы в формате DICOM с произвольным числом изображений и произвольной размерности. Соответственно понадобится возможность масштабировать до какого-то базового размера решётки для рендеринга.

2) Сделать пресеты сканированной "плотности" для разных веществ и тканей человека. Тогда можно будет выбирать диапазоны для отображения, которые будут соответствовать чему-то одному - кости, сосуды, камни различной природы и т.д.

3) Сделать карты отображений сканированная "плотность" - цвет. Тогда можно будет отображать различные органы цветами близкими к естественным, или при помощи ярких цветов сделать легко контрастными и легко различимыми разные ткани, не сильно отличающиеся по плотности и плохо заметные на "серой" картинке.

4) Добавить возможность масштабирования, чтобы можно было выбрать интересующую подобласть модели и растянуть её до размеров решётки для рендига.

5) Добавить инструменты для измерения размеров.

6) Добавить инструменты "Сечения", чтобы можно было видеть разрез модели при помощи секущей плоскости, или угла из двух плоскостей.

ЗАКЛЮЧЕНИЕ

В результате теоретического и практического анализа данной темы, я пришел к выводу, что моя гипотеза подтвердилась: можно создать программу для визуализации данных, полученных с томографа с использованием алгоритма "Marching cubes" и графической библиотеки OpenGL самостоятельно.

Так же я сделал следующие выводы:

Использование алгоритма "Marching cubes" и графической библиотеки OpenGL дает возможность визуализации объемных данных.

Самостоятельно изучить алгоритм "Marching cubes" и графическую библиотеку OpenGL.

Использовать библиотеку OpenGL для визуализации полученных полигонов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. OSP – Гид по технологиям цифровой трансформации [Электронный ресурс] - Режим доступа: <https://www.osp.ru/os/1996/05/178989/> Дата доступа: 05.05.2020г.
2. Википедия [Электронный ресурс] - Режим доступа: <https://ru.wikipedia.org/wiki/%D0%A2%D0%BE%D0%BC%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F/> Дата доступа: 05.05.2020г.
3. Википедия [Электронный ресурс] - Режим доступа: https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D1%82%D0%BE%D0%BC%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F/ Дата доступа: 05.05.2020г.
4. Краснов, М.В. Графика в проектах Delphi. – СПб.: БХВ-Петербург, 2004. – 352с.
5. Краснов, М.В. DirectX. Графика в проектах Delphi. – СПб.: БХВ-Петербург, 2003. – 416с.
6. Фаронтов, В.В. Delphi 5 Учебный курс. М.: «Нолидж», 2000. – 608 с.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

Ссылка на ресурс с размещёнными материалами в облачном хранилище данных:

https://drive.google.com/drive/folders/1I7MJUA9tNSXBH7_h5qClbX4FMdYUetUh?usp=sharing