

ІТМО

Разработка архитектуры программы вокселизации полигональных моделей для задач трехмерной печати

Залялутдинова Карина Радиковна, R34812

Научный руководитель: Афанасьев Максим Яковлевич, к.т.н., факультет СУиР,
доцент (квалификационная категория "ординарный доцент")

Проблема: не все современные CAD-системы позволяют генерировать и редактировать решетчатую структуру для трехмерных моделей.



Решетчатые структуры позволяют:

- Уменьшить вес и **сохранить** удовлетворительные механические и физические **свойства изделия**;
- Сэкономить материал;
- Управлять температурой изделия и др.



Охлаждающая пластина для графического процессора, разработанная компанией TEMISTh

Цель и задачи работы

Цель работы – создать алгоритм, позволяющий:

- Представлять полигональные модели в виде вокселей;
- Заполнять трехмерные модели решетчатой структурой с заданными параметрами.

Задачи работы:

- Анализ общей проблематики представления объемных компьютерных моделей;
- Анализ существующих алгоритмов вокселизации;
- Общий анализ применения решетчатых структур;
- Разработка алгоритма вокселизации и заполнения модели решетчатой структурой;
- Разработка графического интерфейса алгоритма.

Почему воксели?

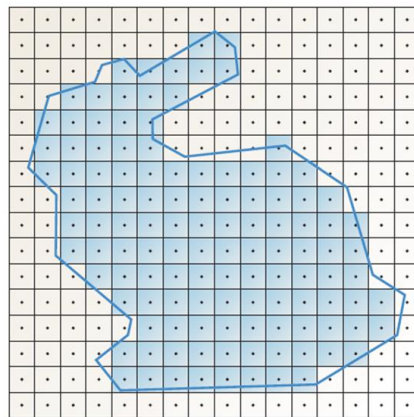
Воксель (от англ. *volute* и *pixel*) – это трехмерное представление пикселя.



Воксельная модель (бинарная) – это **трехмерная матрица**, содержащая **нули** (воксель снаружи объекта) и **единицы** (воксель внутри объекта или на его границе)

Преимущества:

- Отсутствие разрывов на поверхности модели;
- Легко применимы булевы операции.



Сплошная вокселизация модели, двумерный срез
На изображении – модель Стенфордского кролика,
классическая тестовая модель

Функциональные требования:



- Чтение **STL-файла**, содержащего трехмерную модель изделия;
- Преобразование полигональной сетки в **воксельное представление**;
- Реализация **сплошной вокселизации** модели;
- Реализация заполнения воксельной модели **решетчатой структурой**;
- **Обратное преобразование** воксельного представления в STL-файл для слайсинга и генерации G-кода для 3D-принтера

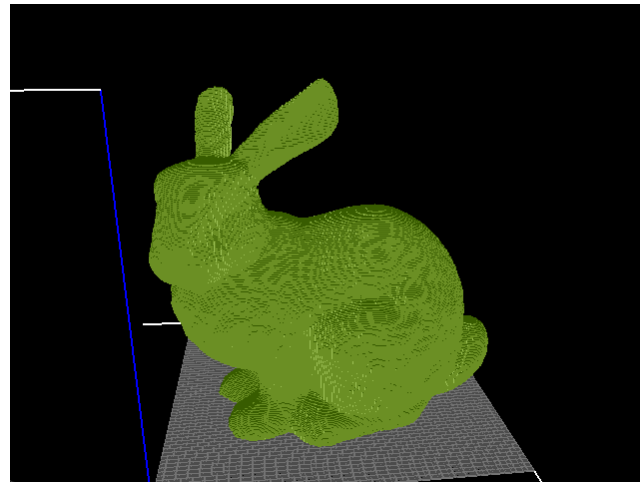
Нефункциональные требования:

- Вокселизация **не более 1 минуты** для воксельной сетки размером 256^3 ;
- Генерация решетчатой структуры с помощью **математических функций**;
- Методология программирования – объектно-ориентированное программирование.

Обзор используемых библиотек, модулей и open-source решений

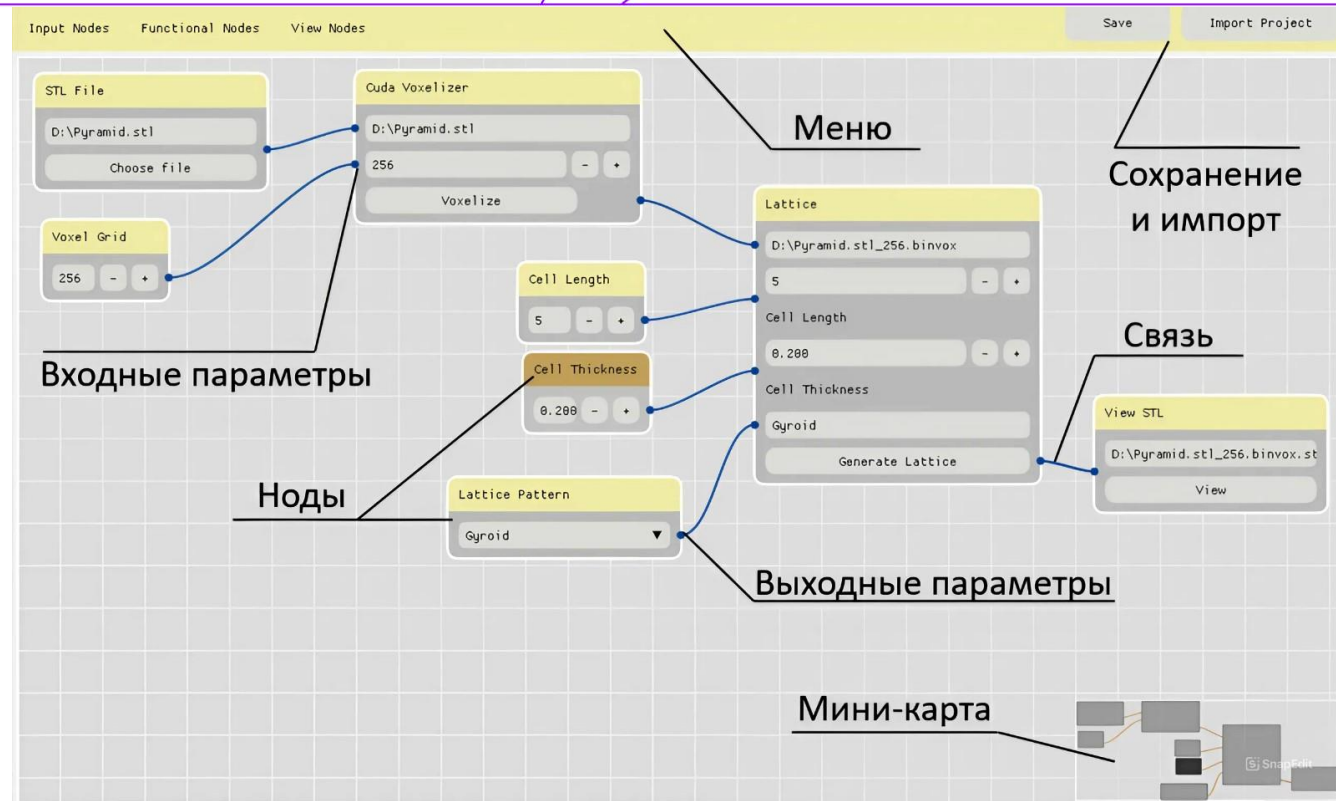
В разрабатываемом алгоритме используются:

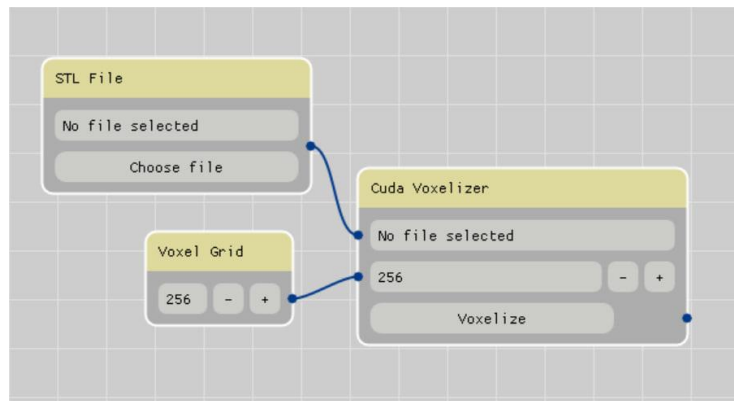
- **Cuda Voxelizer** – инструмент для вокселизации полигональных моделей;
- **DearPyGUI** – библиотека для создания графических интерфейсов;
- **Trimesh** – библиотека на Python, используется для записи STL-файлов;
- **Binvox_rw** – модуль на Python, используется для чтения BINVOX-файлов;
- **Os** – модуль на Python, используется для вызова терминала;
- **Open 3D** – библиотека на Python, визуализирует полученный результат.



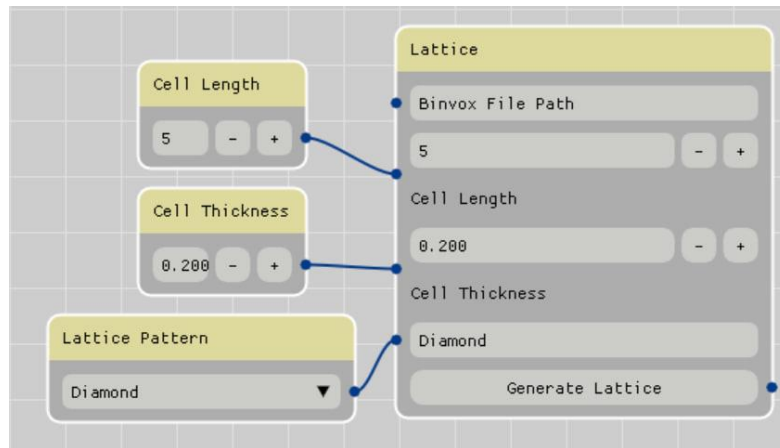
Визуализация результата работы инструмента
Cuda Voxelizer (256^3 вокселей; 2,7 сек.)

Логика и структура алгоритма

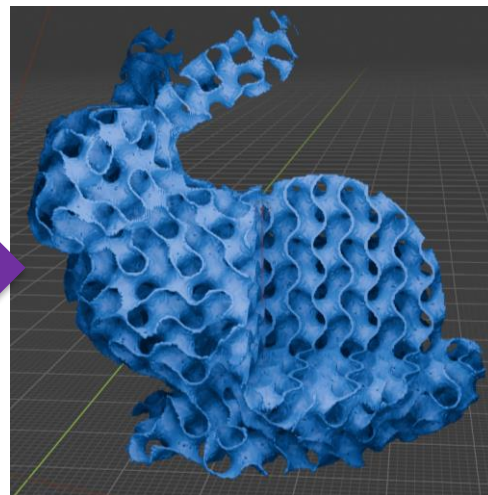
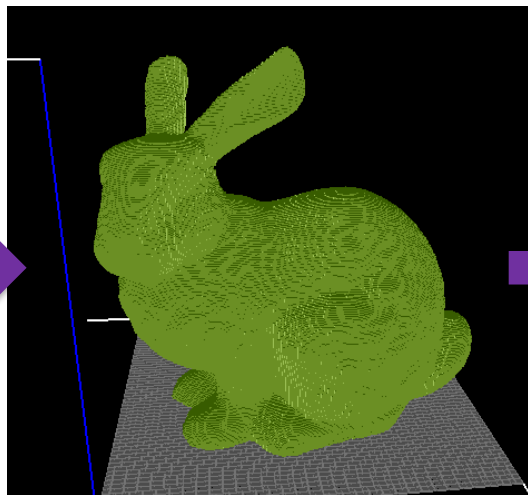




Нода Cuda Voxelizer
вокселизирует STL-файл



Нода Lattice заполняет модель
решетчатой структурой



Исходная модель (STL-файл)

Модель после вокселизации
(BINVOX-файл; 256^3 вокселей; 2,7 сек.)

Выходная модель – STL-файл
(разрез сделан для презентации)

В разрабатываемом алгоритме реализуются:

- Вокселизация полигональной модели;
- Заполнение воксельной модели решетчатой структурой;
- Визуализация полученных результатов.

Дальнейшие планы:

- Адаптация программы для проектирования моделей для медицины;
- Реализация заполнения решетчатой структурой только определенных мест детали;
- Реализация неравномерного заполнения решетчатой структурой



**Спасибо
за внимание!**

it'sMO *re than a*
UNIVERSITY