Chair of Computational Modeling and Simulation
TUM Deparment of Civil, Geo and Environmental Engineering
Technical University of Munich

Implementation of a Modern, High-Performance Marching Cube Library

Project Characteristics

Modeling:	▰▰▰▰▰▱▱▱
Mathematics:	▰▰▰▰▰▰▱▱
Programming:	▰▰▰▰▰▰▰▰
Science:	▰▰▰▰▱▱▱▱

Task

Develop a library using modern C++ that implements the marching cubes algorithm[1]. The focus is on:

- A robust, cross-platform and high-performance implementation that makes use of multi-core CPUs (e.g. with OpenMP) and GPUs (e.g. with OpenCL).
- Modern API, ease of usability and integration to other projects.

The tasks are:

- Literature research on marching cubes algorithm and implementations.
- Get familiar with modern C++ and CMake
- Use and understand a reference implementation with example datasets
- Select a scalable and efficient implementation and implement for multi-core CPUs.
- Select a scalable and efficient implementation and implement for multi-core CPUs.
- Benchmark your library with various other open-source libraries.

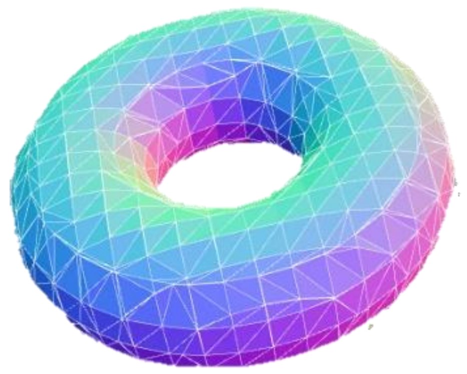

Figure 1: A triangulated torus [2]

[1] William E. Lorensen and Harvey E. Cline. 1987. Marching cubes: A high resolution 3D surface construction algorithm. SIGGRAPH Comput. Graph. 21, 4 (July 1987), 163–169. DOI:https://doi.org/10.1145/37402.37422

[2] “Polygonising a scalar field” by Paul Bourke http://paulbourke.net/geometry/polygonise/ (Acc. 27 January 2022)