

## Software Lab:

Modeling:	<input type="checkbox"/>
Mathematics:	<input type="checkbox"/>
Programming:	<input type="checkbox"/>
Science:	<input type="checkbox"/>

# Implementation of a High Performance Marching Cubes Library

## Description

Marching cubes is a well-known computer graphics algorithm that extracts surfaces from volumetric datasets (1). Although it was first published 1987, the algorithm and its variations are still being studied and used in many fields, such as medical and scientific image computing. The challenge is to have an efficient cross-platform implementation, with a modern API, that can make use of multi-core CPUs as well as GPUs. Such a library can be embedded into various projects.

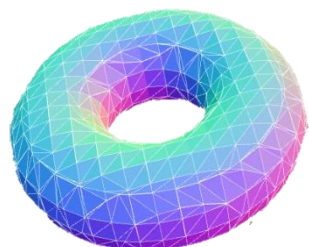


Figure 1: A triangulated torus (2)

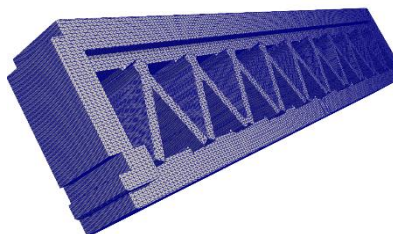


Figure 2: Triangulation of a structural element

## Task

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

- 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.

## Supervisor

Oguz Oztoprak | Chair of Computational Modelling and Simulation

## References

- (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 )