The Journal of Open Source Software

DTCC Builder: A mesh generator for automatic, efficient, and robust mesh generation for large-scale city modeling and simulation

Anders Logg [1,4*], **Vasilis Naserentin** [1,2,4*], **and Dag Wästberg** [3,4*]

1 Chalmers University of Technology **2** Aristotle University of Thessaloniki **3** Chalmers Industriteknik **4** Digital Twin Cities Centre ***** These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 08 June 2022
Published: unpublished

Summary

DTCC Builder is a mesh generator for automatic, efficient, and robust mesh generation for large-scale city modeling and simulation. Using standard and widely available raw data sources in the form of point clouds and cadastral data, DTCC Builder generates high-quality 3D surface and volume meshes, suitable for both visualization and simulation. In particular, DTCC Builder is capable of generating large-scale, conforming tetrahedral volume meshes of cities suitable for finite element (FEM) simulation.

Statement of need

The interest in creating digital twins, i.e., models that mirror physical systems in real-time and enable analysis and prediction, has been rapidly increasing in recent years. In particular, there has been a surge in the interest for creating digital twins of cities (Ketzler et al., 2020). The creation of a digital twin of a city often involves the creation of a 3D model. Such 3D models may either be created manually, semi-automatically, or in a fully automatic way from available raw data, often in the form of point clouds obtained from aerial scanning and cadastral data (property maps).

3D mesh generation is a very challenging process, especially in the face of bad quality and low resolution data, which is often the case for publicly available data for cities. Furthermore, if the 3D meshes are to be used for modeling and simulation, certain requirements are posed on the quality of the meshes. DTCC Builder aims to solve these challenges by automating the mesh generation process in a both robust and efficient way. Related work on 3D mesh generation for city modeling include (Ledoux et al., 2019), (Ledoux et al., 2021), and (Paen et al., 2022).

DTCC Builder is part of the open-source digital twin platform DTCC Platform developed at the Digital Twin Cities Centre.

Functionality

DTCC Builder provides two main programs: `dtcc-generate-citymodel` and `dtcc-generate-mesh`. The two programs are run in sequence. First, `dtcc-generate-citymodel` generates a city model from input data in the form of one or more point clouds and cadastral data. Then, `dtcc-generate-mesh` reads the generated city model and generates output data in the form of both triangular surface meshes and tetrahedral volume meshes. Figure 1 and Figure 2 show a surface mesh generated for an area in Gothenburg.

The Journal of Open Source Software

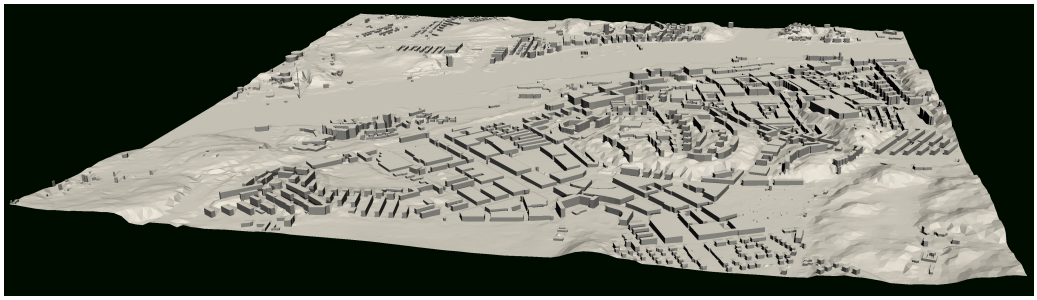

Figure 1: Surface mesh of an area (Majorna) in Gothenburg, generated with DTCC Buider.

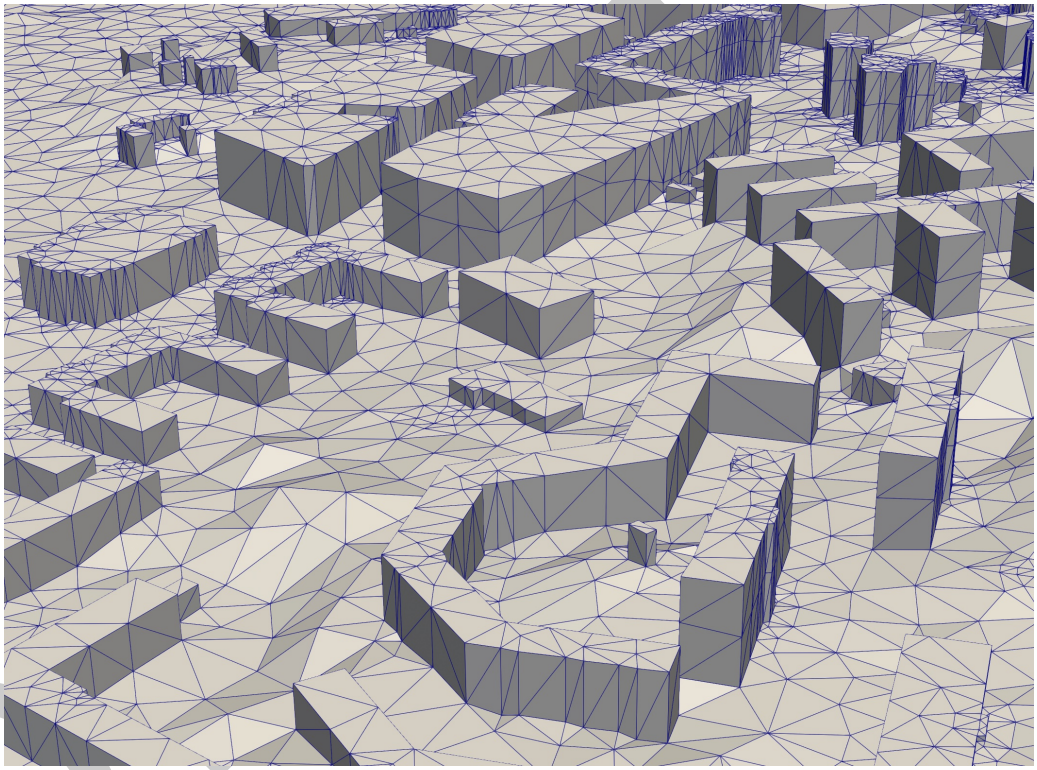

Figure 2: Detail of surface mesh of an area (Majorna) in Gothenburg, generated with DTCC Builder.

Method and implementation

DTCC Builder uses a novel algorithm for mesh generation. The key idea is to utilize the special geometry of city models to reduce the 3D mesh generation problem to a 2D problem. A 2D mesh respecting the polygonal footprints of buildings is generated and then layered to create a 3D mesh. Building heights and ground height are incorporated through a PDE-based smoothing process. The method and algorihtms are described in detail in the paper (Naserentin & Logg, 2022).

DTCC Builder is implemented in C++ and makes use of several open-source packages, notably FEniCS (Logg et al., 2012) for solving PDEs, Triangle (Shewchuk, 1996) for 2D mesh generation, and GEOS (GEOS contributors, 2021) for geometric operations.

Documentation

The documentation for DTCC Builder is published on the DTCC Builder GitLab pages as well as on the documentation pages for DTCC Platform.

Limitations and future work

DTCC Builder currently only provides a C++ and command-line interface. Future versions will provide a Python interface and also an online interface as part of DTCC Platform.

DTCC Builder currently generates city models in Level of Detail (LoD) 1.2 but ongoing work seeks to extend DTCC Builder to LoD1.3 and LoD2.x.

DTCC Builder currently runs on a single thread. Future versions versions will provide means of parallelization across shared or distributed memory.

Acknowledgements

This work is part of the Digital Twin Cities Centre supported by Sweden's Innovation Agency Vinnova under Grant No. 2019-00041.

References

GEOS contributors. (2021). *GEOS coordinate transformation software library*. Open Source Geospatial Foundation.

Ketzler, B., Naserentin, V., Latino, F., Zangelidis, C., Thuvander, L., & Logg, A. (2020). Digital Twins for Cities: A State of the Art Review. *Built Environment*, *46*(4), 547–573. https://doi.org/10.2148/benv.46.4.547

Ledoux, H., Arroyo Ohori, K., Kumar, K., Dukai, B., Labetski, A., & Vitalis, S. (2019). CityJSON: A compact and easy-to-use encoding of the CityGML data model. *Open Geospatial Data, Software and Standards*. https://doi.org/10.1186/s40965-019-0064-0

Ledoux, H., Biljecki, F., Dukai, B., Kumar, K., Peters, R., Stoter, J., & Commandeur, T. (2021). 3dfier: Automatic reconstruction of 3D city models. *Journal of Open Source Software*, *6*(57), 2866. https://doi.org/10.21105/joss.02866

Logg, A., Mardal, K.-A., & Wells, G. (2012). *Automated solution of differential equations by the finite element method: The FEniCS book* (Vol. 84). Springer Science & Business Media.

Naserentin, V., & Logg, A. (2022). Digital twins for city simulation: Automatic, efficient, and robust mesh generation for large-scale city modeling and simulation. *arXiv:2210.05250*. https://arxiv.org/abs/2210.05250

Paen, I., García-Sánchez, C., & Ledoux, H. (2022). Towards automatic reconstruction of 3D city models tailored for urban flow simulations. *Frontiers in Built Environment*, *8*.

Shewchuk, J. R. (1996). Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. *Workshop on Applied Computational Geometry*, 203–222.