3dfier: automatic reconstruction of 3D city models

Ledoux, H.; Biljecki, Filip; Dukai, B.; Kumar, Kavisha; Peters, R.Y.; Stoter, J.E.; Commandeur, T.J.F.

DOI
10.21105/joss.02866

Publication date
2021

Document Version
Final published version

Published in
Journal of Open Source Software

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

The Journal of Open Source Software

3dfier: automatic reconstruction of 3D city models

Hugo Ledoux[*1], **Filip Biljecki**[2], **Balázs Dukai**[1], **Kavisha Kumar**[1], **Ravi Peters**[1], **Jantien Stoter**[1], **and Tom Commandeur**[1]

1 Delft University of Technology, the Netherlands **2** National University of Singapore, Singapore

DOI: 10.21105/joss.02866

Software

- Review
- Repository
- Archive

Editor: Arfon Smith

Reviewers:

- @GANys
- @chenkianwee

Submitted: 19 November 2020
Published: 26 January 2021

Summary

Three-dimensional city models are essential to assess the impact that environmental factors will have on citizens, because they are the input to several simulation and prediction software. Examples of such environmental factors are noise (Stoter et al., 2008), wind (García-Sánchez et al., 2014), air pollution (Ujang et al., 2013), and temperature (Hsieh et al., 2011; Lee et al., 2013).

However, those 3D models, which typically contain buildings and other man-made objects such as roads, overpasses, bridges, and trees, are in practice complex to obtain, and it is very time-consuming and tedious to reconstruct them manually.

The software *3dfier* addresses this issue by automating the 3D reconstruction process. It takes 2D geographical datasets (e.g., topographic datasets) that consist of polygons and "3dfies" them (as in "making them three-dimensional"). The elevation is obtained from an aerial point cloud dataset, and the semantics of the polygons is used to perform the lifting to the third dimension, so that it is realistic. The resulting 3D dataset is semantically decomposed/labelled based on the input polygons, and together they form one(many) surface(s) that aim(s) to be error-free: no self-intersections, no gaps, etc. Several output formats are supported (including the international standards), and the 3D city models are optimised for use in different software.

Statement of need

The 3D city models needed as input in environmental simulations have specific requirements that go beyond the typical 3D models used for visualisation: they require semantic information (i.e., an object, modelled with one or more surfaces, "knows" what it is, for instance a window or a roof surface) and they should be free of geometric errors. It is known that practitioners and researchers wanting to perform some simulations or analysis can spend a significant part of their time constructing and repairing the input 3D models; McKenney (1998) estimates this to as much as 70% of their time. Furthermore, the formats required by the different software and/or the agencies (for instance the international standard CityGML), are complex to generate (Ledoux et al., 2019).

The software *3dfier* automates the reconstruction step, it enriches the data with semantics, and it supports several output formats (used in different fields).

It builds upon previous work done for reconstructing the whole country of the Netherlands (with 10M+ buildings) (Oude Elberink et al., 2013), and provides the following improvements: use of recent and maintained libraries (e.g., CGAL, GDAL and Boost), a clear open-source license, recent formats and international standards are supported, no geometric errors in output.

*Corresponding author

There exist different commercial software that can perform extrusion (e.g., Safe FME or ArcGIS), but these extrude each classes of objects separately (usually only buildings), without ensuring that adjacent objects should be "stitched" together. As a result, the resulting 3D dataset is often unsuitable as input for other spatial analysis software.

Overview of the reconstruction steps

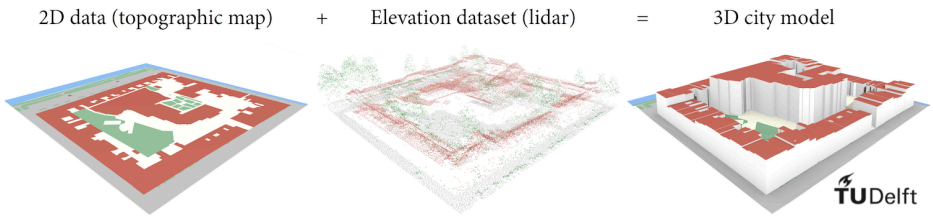


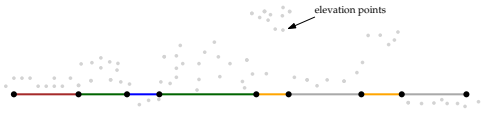
Figure 1: Overview of 3dfier.

As shown in Figure 1, as input we use geographical datasets that are readily available for an area (often as open data too):

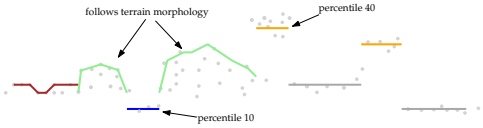
1. 2D polygons representing buildings, lakes, roads, parts, etc. (OpenStreetMap is one option);
2. elevation points, usually acquired with a laser-scanner and available in LAS format, or derived from aerial images.

Each of the classes in the input 2D polygons is mapped to a specific class: *Terrain*, *Forest*, *Water*, *Road*, *Building*, *Bridge/Overpass*, and *Separation* (walls and fences).

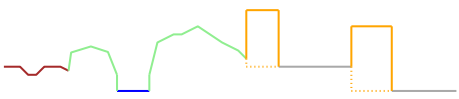
(a) Input polygons are mapped to a class.



(b) Input elevation points are overlayed with polygons.



(c) Each polygon is lifted separately, based on its class.



(d) Adjacent polygons are stitched together to form an error-free surface.

Figure 2: 1D visualisation of the reconstruction process.

The semantics of every input 2D polygon is used to perform the lifting to the third dimension. For example, water polygons are extruded to horizontal polygons, buildings to prismatic blocks, roads as smooth surfaces, etc. Every polygon is triangulated and in a next step the lifted polygons are "stitched" together so that one surface is reconstructed. In this step, priority is given to "hard" objects such as roads, i.e., vegetation polygons are moved to be aligned with the road polygons. The output of the software is one watertight surface with no intersecting

triangles and no holes, and features such as buildings and trees can be added or omitted. Triangles are grouped and labelled with the class and the attributes that were in the input 2D polygons they decompose. This surface can be used directly as input in several urban applications, such as simulations of noise or wind.

Use of the software

The software is a command-line interface (CLI), and uses a configuration file as input (a YAML file). This file allows the user to control the mapping between the input and the *3dfier* classes, to specify which LAS classes to use/ignore, to control how the lifting is performed for the different classes, etc.

The software, being modular, is also extensible for other use cases or for use in different countries. As an example, new topographic classes (for instance trees) could be added by simply creating a new C++ class that inherits from the parent class, and the output for the different formats supported must be added.

Great care was taken to keep the software as efficient as possible and make it suitable for reconstructing very large areas. For instance *3dfier* is the software that enables the Dutch national mapping agency (Kadaster) to create the 3D base registration for the Netherlands.

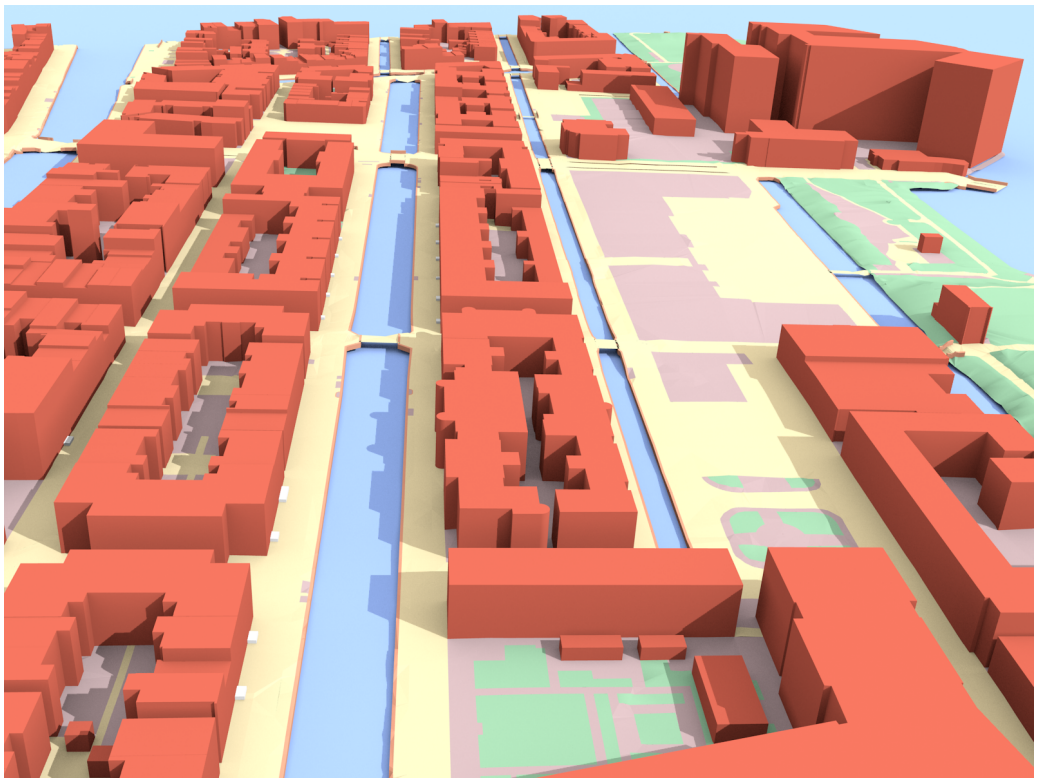


Figure 3: An example of the output of 3dfier, for the city of Leiden in the Netherlands.

Acknowlegements

This work was funded by the Netherlands Kadaster and received funding from the European Research Council (ERC) under the European Unions Horizon2020 Research & Innovation Programme (grant agreement no. 677312 UMnD: Urban modelling in higher dimensions).

References

García-Sánchez, C., Philips, D. A., & Gorlé, C. (2014). Quantifying inflow uncertainties for CFD simulations of the flow in downtown oklahoma city. *Building and Environment*, *78*, 118–129. https://doi.org/10.1016/j.buildenv.2014.04.013

Hsieh, C.-M., Aramaki, T., & Hanaki, K. (2011). Managing heat rejected from air conditioning systems to save energy and improve the microclimates of residential buildings. *Computers, Environment and Urban Systems*, *35*(5), 358–367. https://doi.org/10.1016/j.compenvurbsys.2011.02.001

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*, *4*(4). https://doi.org/10.1186/s40965-019-0064-0

Lee, D., Pietrzyk, P., Donkers, S., Liem, V., Oostveen, J. van, Montazeri, S., Boeters, R., Colin, J., Kastendeuch, Pi., Nerry, F., Menenti, M., Gorte, B., & Verbree, E. (2013). Modeling and observation of heat losses from buildings: The impact of geometric detail on 3D heat flux modeling. *Proceedings 33rd European Association of Remote Sensing Laboratories (EARSeL) Symposium*, 1–20.

McKenney, D. (1998). *Model quality: The key to CAD/CAM/CAE interoperability*. International TechneGroup Incorporated, Milford, OH.

Oude Elberink, S., Stoter, J., Ledoux, H., & Commandeur, T. (2013). Generation and dissemination of a national virtual 3D city and landscape model for the Netherlands. *Photogrammetric Engineering & Remote Sensing*, *79*(2), 147–158. https://doi.org/10.14358/PERS.79.2.147

Stoter, J., Kluijver, H. de, & Kurakula, V. (2008). 3D noise mapping in urban areas. *International Journal of Geographical Information Science*, *22*(8), 907–924. https://doi.org/10.1080/13658810701739039

Ujang, U., Anton, F., & Abdul Rahman, A. (2013). Unified data model of urban air pollution dispersion and 3D spatial city model: Groundwork assessment towards sustainable urban development for Malaysia. *Journal of Environmental Protection*, *4*(7), 701–712. https://doi.org/10.4236/jep.2013.47081