Automated 3D Reconstruction of LoD2 and LoD1 Models for All 10 Million Buildings of the Netherlands

Ravi Peters, Balázs Dukai, Stelios Vitalis, Jordi van Liempt, and Jantien Stoter

Abstract

In this paper, we present our workflow to automatically reconstruct three-dimensional (3D) building models based on two-dimensional building polygons and a lidar point cloud. The workflow generates models at different levels of detail (LoDs) to support data requirements of different applications from one consistent source. Specific attention has been paid to make the workflow robust to quickly run a new iteration in case of improvements in an algorithm or in case new input data become available. The quality of the reconstructed data highly depends on the quality of the input data and is monitored in several steps of the process. A 3D viewer has been developed to view and download the openly available 3D data at different LoDs in different formats. The workflow has been applied to all 10 million buildings of the Netherlands. The 3D service will be updated after new input data becomes available.

Introduction

Three-dimensional (3D) city models are widely used in urban applications. The outcomes of such applications serve as input for planning and decision-making processes that aim at making cities cooler, sustainable, more accessible, greener, carbon dioxide-neutral, etc. (Biljecki *et al.* 2016; Deren *et al.* 2021). Models of buildings are prominent objects in these models. The building models can be generated at different levels of detail (LoDs). Taking the terminology of CityGML, a building can be modeled at four main levels of detail for the outer shell of the building, i.e., LoD0, LoD1, LoD2, and LoD3, and at LoD4 for the interior of the building (OGC 2012; Kutzner et al 2020). Each of these four CityGML LoDs can be further refined (Biljecki *et al.* 2016; Sun *et al.* 2019).

A higher level of detail is often preferred over a lower one, since building models at higher LoDs look closer to reality. However, higher levels of detail are more complicated (and therefore more expensive) to acquire because it is harder to reconstruct them in an automated manner from available source data. In addition, using models at higher levels of detail in spatial analysis does not automatically lead to better results (Biljecki *et al.* 2018), while at the same time too much detail may have a negative impact on performance. Therefore, for some applications it is better to avoid too much irrelevant detail.

The LoD of a 3D city model is therefore driven by the specific data requirements of the urban application for which it is built (see also the section "LoD in Relation to Urban Applications"). However, the highest achievable LoD is also restricted by the available source data and the reconstruction method used (see also the section "LoD in Relation to Reconstruction Method").

While many 3D city models exist for various parts of the Netherlands, they are often generated for relatively small areas, are using different reconstruction methods, and are based on different source

data. Furthermore, the update cycles are different, and the level of detail is also different because it is collected for different applications.

This can result in inconsistencies between 3D city models of the same area. There may be discrepancies between the geometries of building models like the geometry or height of the footprint. Also, the reference heights for the same building might differ over data sets since the heights may represent different references (e.g., gutter, ridge, maximum) or the reference heights are based on different statistical calculations. In addition, buildings (or building parts) available in one data set might be missing in another data set. There may also be temporal differences because the input data sets that were used for the reconstruction come from another date. Typically, there is no plan to maintain and update the once generated 3D data. This may be another source for indiscrepancies.

All these differences have profound influences in practice, such as affecting the applications for which an existing 3D model can be used, the processing that is necessary to use it, and the likely errors that will be present in the end result.

In this research, we demonstrate how to create a consistent countrywide 3D city model in LoD1.2, LoD1.3, and LoD2.2. In order to achieve this, we look at three main aspects.

First, to ensure consistency between 3D city models of the same area and different LoDs, to improve efficiency, and serve the 3D data needs of different urban applications, we investigate how to reconstruct building models for large areas at different LoDs in one reconstruction process, based on the same reconstruction principles and based on the same source data. For the block models, we provide the user with several reference heights, so that the user can select the appropriate reference height to extrude building blocks for the specific application.

Second, our objective is to develop a fully automated reconstruction method. Our focus is on 3D city models covering large areas to support countrywide urban applications. This requires a fully automated reconstruction method. Automated reconstruction also enables standardization of the output data resulting in consistent geometries, semantics, and temporal aspects of the data. This consistency is also ensured when new models are reconstructed in the future with the same automated procedure based on updated source data.

Third, we investigate how to monitor and assess the quality of the building models that are automatically generated on such a large scale. This is essential for users to assess if models are fit for a specific use.

Finally, we also investigate the visualization and dissemination of such a big data set so that the city model is accessible and usable in an efficient manner.

Structure of this Paper

In this paper, we present our methodology to reconstruct LoD1.2, LoD1.3, and LoD2.2 models of all buildings in the Netherlands in one process. The section "Scope of the Research and Previous

---

3D Geoinformation group, Delft University of Technology, Faculty of Architecture and the Built Environment, Department of Urbanism, Julianalaan 134, Delft 2628BL, The Netherlands (j.e.stoter@tudelft.nl).

Work" further outlines the scope of this research by elaborating on the different LoDs of 3D city models and related work. We present our reconstruction methodology in the section "Methodology for Automated 3D Reconstruction" and the implementation in the section "Implementation". We close with conclusions in the last section.

Scope of the Research and Previous Work

In this section, we first explain how the reconstructed level of detail of 3D building models depends on the one hand on the data requirements for which the data is collected (section "LoD in Relation to Urban Applications") and on the other hand on the reconstruction method (section "LoD in Relation to Reconstruction Method"). We then describe the motivation for the LoDs that we reconstruct in our research in the section "The LoDs in Our Research". Finally, the section "3D City Models of Large Areas" presents other work on the reconstruction of 3D data for large areas.

LoD in Relation to Urban Applications

As in two dimensions (2D), a one-fits-all approach does not exist for a 3D city model. Instead, specific applications require specific 3D data as is analyzed in Biljecki *et al.* (2015). For example, block models (LoD1) are sufficient for shadow-, wind-, and noise-simulations. Roof structures (LoD2) with information on the roof materials are needed for solar potential estimation or in accurate energy demand estimation. Although LoD2 models are often also preferred in visualizations since they provide a realistic experience, realistic looking LoD2 models could still be ambiguous (Biljecki *et al.* 2018).

LoD in Relation to Reconstruction Method

LoD1 models for every building can be automatically generated rather easily from point clouds and 2D building polygons, i.e., footprints (Ledoux *et al.* 2021). Therefore, LoD1 models are frequently generated by various organizations, as such source data are increasingly available as open data. However, automatically generated LoD1 models for the same area can still differ in, for example, their reference heights (e.g., the rooftop, the gutter height, one third of the roof-height) and the underlying statistical calculations. Many users are not aware of those multiple options to reconstruct a 3D block model, while these options do influence the outcome of analyses for which the LoD1 models are used (Biljecki *et al.* 2018).

With respect to LoD2 models, many roof shapes can be generated fully automatically, although LoD2 reconstruction is still a current topic of research, as both the quality of available surveyed data and new 3D reconstruction algorithms still steadily improve (Rottensteiner *et al.* 2014; Lafarge 2015; Pârvu *et al.* 2018).

The additional elements for LoD3 models are hard to reconstruct in an automated manner. Therefore, they are generated manually or are the result of converted Industry Foundation Classes (IFC) models from the BIM domain (Donkers *et al.* 2016; Colucci *et al.* 2020).

The LoDs in Our Research

In our research, we focus on the reconstruction of LoD1.2, LoD1.3, and LoD2.2 building models using the terminology of the refined LoD framework of Biljecki *et al.* (2016). We distinguish between two types of LoD1 models: LoD1 models that are a result of extruding a complete building footprint to one height, i.e., LoD1.2 models in Biljecki *et al.* (2016) and models that are extruded to one or more heights in case significant height jumps occur within the footprint, like a church with a tower or a house with a shed attached, i.e., LoD1.3 models. Both models are relatively simple and are therefore appropriate for applications that need simplified models. But LoD1.3 models enable more realistic visualizations. In addition, LoD1.3 models are also more accurate data for simulations that take block-shaped models of buildings as input, such as noise simulation where buildings act as noise barriers. This is why the automated reconstruction of LoD1.3 models is included in our research. LoD1.3 models are more difficult to automatically generate than LoD1.2 because it requires the detection of height discontinuities within the building footprint.

The LoD1.0/LoD2.0 and LoD1.1/LoD2.1 models are based on generalized footprints and therefore outside the scope of our research. The LoD3 representations are outside our scope since they require manual work.

3D City Models of Large Areas

There are many other examples of data sets containing building models of large cities or even nations as shown by an inventory by Santhanavanich (2020). Examples are the whole United States, containing 125 million building models at LoD1, the city of New York (with 100 LoD2 models of iconic buildings), as well as the LoD2 models of Montreal, Helsinki, Singapore, cities in North Rhine-Westphalia State (in LoD 1 and LoD2), and many other cities in Germany. An example of an LoD2 building data set covering a whole nation is the swissBUILDINGS3D 2.0 data set (Swisstopo, 2021). It is a vector-based data set which describes (among other topographic objects) buildings as 3D models with roof geometries and roof overhangs. The building models were extracted in a semi-automated manner from aerial images using a photogrammetric method of digital image (stereo) correlation, enhanced with additional information as attributes. Other building elements (footprint, facades, roof overhangs) are created with automated procedures.

Several of these initiatives highlight a problem of existing 3D models: often they are the result of a one-time capture, with a few mostly manual updating exceptions. Updates and extensions may be considered in the future but were not foreseen at the moment they were captured. In addition, existing models resulting from the same workflow (and thus consistent) are limited to one or at the most to two different levels of detail for the same area and therefore the 3D data are limited to specific applications. More often, different LoDs of the same area are a result of different workflows and are therefore nonconsistent with respect to geometry, temporal aspects, and semantics. Finally, detailed metadata about how the buildings were reconstructed, including quality information, are often not generated and thus missing.

Methodology for Automated 3D Reconstruction

In this section, we describe the reconstruction methodology of LoD1.2, LoD1.3, and LoD2.2 that we have developed and implemented for large areas, which we deliver as both 3D models and 2D+heights data (see Figure 1).
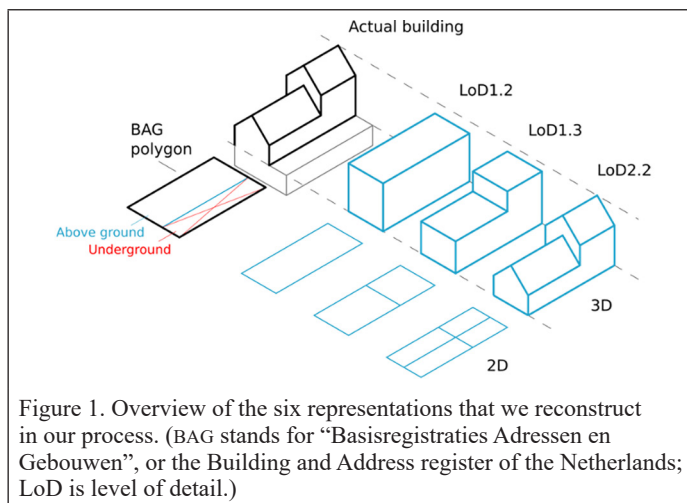

Figure 1. Overview of the six representations that we reconstruct in our process. (BAG stands for "Basisregistraties Adressen en Gebouwen", or the Building and Address register of the Netherlands; LoD is level of detail.)

First, we describe the input data that we use in the reconstruction. Then, we describe the process itself. Since the LoD2.2 reconstruction generates information that is used in the reconstructions of the other LoDs (e.g., distinguishing between points that fall on walls and on roofs; generating a planar partition of the original footprint based on the identified roof planes), we start with the LoD2.2 reconstruction.

Input Data

For our reconstruction process we use building polygons ("Basisregistraties Adressen en Gebouwen", or the Building and Address register of the Netherlands (BAG)) and lidar point data ("Actueel Hoogtebestand Nederland", the national height model of the Netherlands (AHN)) as input.

BAG: The polygons come from the Building and Address register of the Netherlands (BAG). This data set contains all buildings and addresses in the Netherlands. The geometry of addresses is collected as points and those of buildings as polygons (i.e., outline as seen from above). Municipalities are responsible for collecting the BAG data and keeping the data up-to-date.

The geometry for BAG buildings is acquired from aerial photos and terrestrial measurements and the data positional data accuracy is 30 cm. The data is provided via the national geo-portal PDOK (2021) both in a viewer and as download service. BAG also contains the history of buildings, i.e., buildings that are planned and buildings that have existed in the past but now have been removed.

For our reconstruction, we make a selection of the BAG buildings that have been realized and have not (yet) been demolished, i.e., the input building polygon data set represents the current situation.

AHN: The national height model of the Netherlands (AHN, 2019) is a point cloud acquired by airborne lidar. The first version of AHN (with a density of at least one point per 16 square meters, and in forests one point per 36 square meters) was completed in 2003. In the period of 2009 to 2012, the second version of the data set was acquired with an average point density of 10 points per square meter. The third version was collected between 2014 and 2019. The resolution of AHN3 that we use for our reconstruction process is similar to the one of AHN2. In addition, it contains a classification of the point cloud. For the AHN2 and AHN3 point clouds, it is specified that an object of 2×2 m can be mapped with an accuracy of at least 50 cm. The height accuracy is 10 cm. We use the classes "building" and "ground points" to determine building heights respectively heights at ground level. The fourth version of AHN will become available in the next two years. AHN4 will have a point density of about 10–14 points per square meter, and in some locations even higher.

LoD2.2

Our reconstruction method improves upon our earlier research as described in Dukai, Ledoux, and Stoter (2019) Dukai *et al.* (2020), and Stoter *et al.* (2020). The main improvement in this work is the addition of LoD2.2 output.

Our method uses footprints and height points that are well aligned as input and consists of two steps. In the first part the input footprint is partitioned into roof parts. And in the second part this 2D roof partition is extruded into a 3D model.

Footprint Partitioning

In this first step, the input footprint is partitioned by breaklines detected in roof planes (see Figure 2).

These roof planes are detected if sufficient points can be found for that plane using a region-growing algorithm (see Figure 2.2). For the AHN that we use, with a point density of ~8 points/m2, we set the minimum number of points to 15, which is equal to a roof element of about 2 square meters. Points that are on a wall plane (facade), or not part of any plane are removed.

We derive two types of lines from the planes: boundary lines and intersection lines (see Figure 2.3). The boundary lines of roof planes are detected using the α-shape of each detected roof plane. The intersection lines are generated at the location where adjacent planes intersect, e.g., on top of a gable roof.

Before the boundary and intersection lines are used to subdivide the footprint, they are regularized and duplicate lines are removed. For example, the line on top of the gable roof in Figure 2.3 is detected three times: once as an intersection line and twice as a boundary line, i.e., once for each incident roof plane.

The remaining lines are used to subdivide the footprint into an initial planar partition (Figure 2.4). This is referred to as the initial roof partition. The initial roof partition may still have a high complexity, i.e., it may contain many small faces. To further reduce the complexity of the roof-partition, an optimization step is performed using a

graph-cut optimization (Zebedin, 2008). In this step, a roof plane is assigned to each face in the roof-partition (see Figure 2.5). This is done in such a way that the total error with the input point cloud is minimized and the total length of the edges between faces of a different roof plane is minimized. The latter assures a minimum number of edges and vertices, i.e., a low model complexity. After this step, the edges for which the two incident faces are assigned to the same roof plane are removed from the partition. The faces in the resulting final roof partition are referred to as roof parts.

Extrusion

In the LoD2.2 reconstruction, the identified roof parts are extruded from ground level to a 3D mesh (Figure 2.6). The mesh consists of three types of surfaces: the ground plane, the roof surfaces, and the wall surfaces. The height of the ground plane is based on the lowest point around the building and is calculated as the fifth percentile of all ground points that are within a 4 m buffer of the building. An intersection curve of the terrain could also be used for this. The construction is done in such a way that no internal walls are created and the mesh is topologically correct.

LoD1.3

The LoD1.3 reconstruction uses the same footprint partitioning as is generated for the LoD2.2 reconstruction. But for LoD1.3, the footprint partitions are further simplified by merging neighboring parts that have no significant height difference. We use 3 m as a threshold in this process, which is more or less equal to a floor-height, no matter the area. As the height reference we use the 70th percentile height for each roof part. The merging starts with merging from small to large height gaps and is an iterative process, i.e., if merging two roof parts leads to an elevation difference of <3 m with another part, they are merged again. The iteration stops when there are no more height differences smaller than 3 m between adjacent roof parts.

In the next step, reference heights are calculated for each remaining part and used to extrude the part.

As explained in the section "LoD in Relation to Reconstruction Method", these reference heights can represent different extrusion heights for one building, i.e., the roof edge, the ridge height, or the maximum height (like a chimney). Furthermore, the underlying statistical calculations used to calculate the extrusion height can differ, e.g., average, median, or maximum.

To standardize possible extrusion variations and to let the user choose which one to use, our method calculates four different reference heights from the points that fall on a roof part (excluding the points on walls) and assigns these to the 2D roof parts, i.e., minimum, maximum, 50th percentile, and 70th percentile.

The LoD1.3 models are provided in two representations: as 2D roof-parts with the different reference heights as attributes, and as reconstructed (i.e., extruded) 3D models based on the 70th percentile of the roof points the specific part contains. The 2D roof parts also
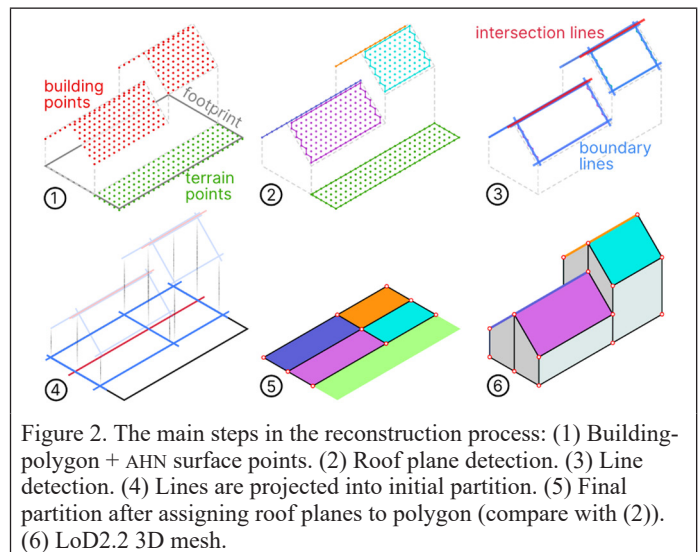

Figure 2. The main steps in the reconstruction process: (1) Building-polygon + AHN surface points. (2) Roof plane detection. (3) Line detection. (4) Lines are projected into initial partition. (5) Final partition after assigning roof planes to polygon (compare with (2)). (6) LoD2.2 3D mesh.

contain the identification of the original building to link the individual roof parts to the original building. The 3D models are reconstructed in such a way that there are no inner walls.

The reconstructed 3D models are represented as solids since rules for solids are stricter than for MultiSurfaces, e.g., using the solid geometry enforces a 2-manifold (i.e., watertight) 3D object.

For the ground surface, the same height is used as for the LoD2.2 models, i.e., fifth percentile of all surface points that fall within 4 m radius of footprint.

LoD1.2

For the LoD1.2 models, we calculate the same four reference heights for the extrusion as we do for LoD1.3 and assign these to the original footprints. Also for LoD1.2, our method both reconstructs the 3D block models (based on the 70th percentile of height points that fall on all roof parts of the building) and assigns the four reference heights to the original 2D footprints. The same surface height is used as for the other LoDs.

Implementation

Implementation of the Reconstruction Process

The implementation of the whole process as described above is visualized in Figure 3. The input data is tiled to make the reconstruction and dissemination of the data more manageable. After reconstruction, the building models are stored in a PostgreSQL database from which the data is exported or directly consumed in various formats.

There are two considerations for optimally tiling the building footprints. First, the objective is to limit the number of buildings in each tile so that the workload is as balanced as possible between the processes. Second, the buildings should be spatially clustered so that the corresponding point cloud can be read efficiently. To meet both conditions, we use a quadtree with a maximum cell size of 3500 for subdividing the buildings (see Figure 4).

Thus, the building tiles are the leaves of the quadtree, where each tile contains a maximum of 3500 buildings. This assures that the reconstruction-time per tile is more or less the same and that the tiles available for download are similar in file size. The reconstruction of all ten million buildings in the Netherlands takes about 40 hours, with 30 concurrent processes on a single machine (two Xeon E5-2650 Central Processing Units, 128 GB RAM). The computation cost scales linearly with the number of buildings, since each building is processed independently. The reconstruction process is highly automated, which allows us to quickly run a new iteration in case of improvements in an algorithm or in case new input data becomes available. Figure 5 shows an example of reconstructed 3D models at different LoDs for one building.

Visualization and Dissemination

To view and query as well as to download the reconstructed building models, we built a website with a 3D viewer (Figure 6). The 3D viewer was developed with two main goals: network performance (i.e., fast fetching of the data) and client performance, i.e., to minimize the resource needed on the device being used (including mobile devices). We developed our own solution since we could not find a suitable off-the-shelf solution.

To satisfy the network and client performance, we use a web graphics-friendly format with minimal size requirements, i.e., 3D Tiles[1]. This is based on the glTF format[2]. We export the data set to 3D Tiles using the same tiling scheme as for the reconstruction of the buildings, which ensures that tiles have a relatively equal distribution of objects. We use 3DTilesRendererJS[3] to render these tiles in the viewer. Our solution to make the user interface easy for users and mobile friendly uses VueJS[4] for the website's logic and Bulma[5] for the styling of the user interface elements.

The viewer provides several functionalities that enable users to investigate the whole data set, as well as share it with others. A user lands at an initial point and can move around the country. A simple flat terrain using Web Map Tile Service (WMTS)[6] is used to provide proper orientation context to the user. Each location in the viewer corresponds to a unique address so that the user can bookmark or share the current
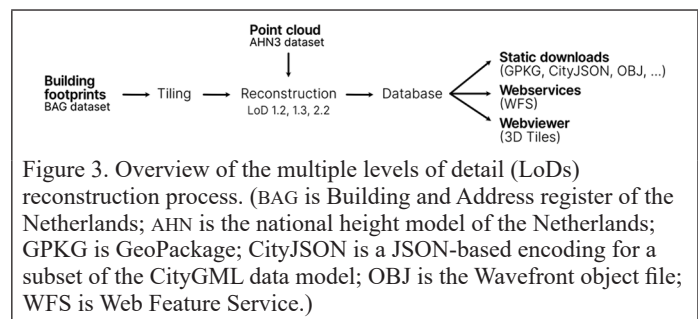

Figure 3. Overview of the multiple levels of detail (LoDs) reconstruction process. (BAG is Building and Address register of the Netherlands; AHN is the national height model of the Netherlands; GPKG is GeoPackage; CityJSON is a JSON-based encoding for a subset of the CityGML data model; OBJ is the Wavefront object file; WFS is Web Feature Service.)
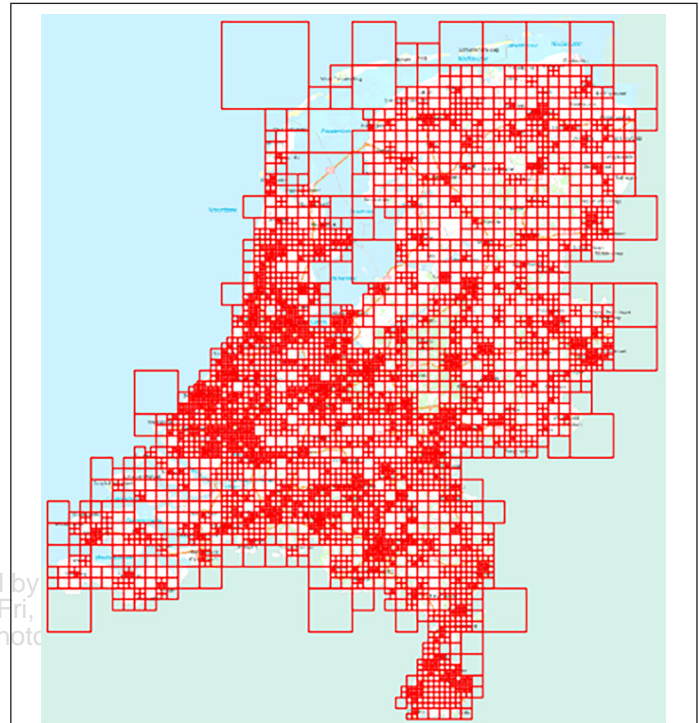

Figure 4. Quadtree-based tiling scheme for data processing and dissemination.
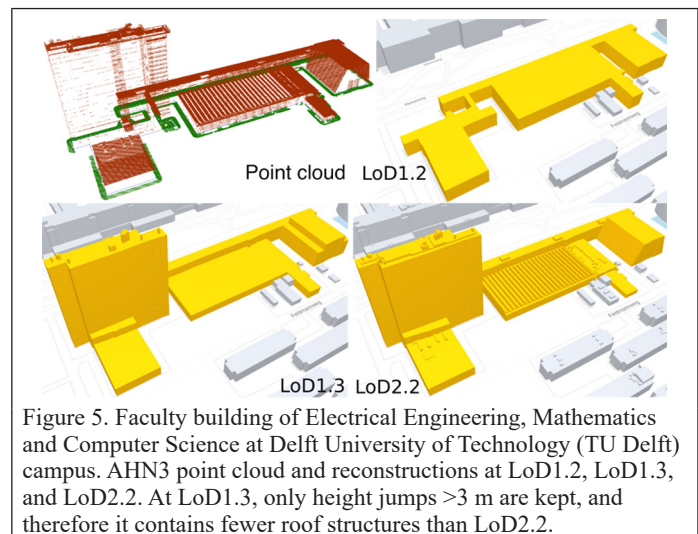

Figure 5. Faculty building of Electrical Engineering, Mathematics and Computer Science at Delft University of Technology (TU Delft) campus. AHN3 point cloud and reconstructions at LoD1.2, LoD1.3, and LoD2.2. At LoD1.3, only height jumps >3 m are kept, and therefore it contains fewer roof structures than LoD2.2.

1. https://github.com/CesiumGS/3d-tiles
2. https://www.khronos.org/gltf/
3. https://github.com/NASA-AMMOS/3DTilesRendererJS/
4. https://vuejs.org/
5. https://bulma.io/
6. http://opengeospatial.github.io/e-learning/wmts/text/index.html

view with others. Finally, the user can click on a building and get its information, as well as some derived properties: the height of the building at the specified point, and the slope of the surface that the user clicked on.

Through the website www.3DBAG.nl, the data is downloadable in different formats: GeoPackage, PostGIS backup, Wavefront OBJ, and CityJSON. The viewer also contains a function to report errors by users to help us improve our process, which can concern any part of the process, i.e., from preprocessing input data to reconstruction, viewing, and use.

Quality Information

Quality information regarding the resulting models is needed to identify a badly reconstructed model or an exceptional situation for which the 3D reconstruction process had not yet accounted for. With this information, the reconstruction process can be improved. In addition, it provides the user with information on how good a specific model is so that the user can act upon this.

We calculate two types of quality parameters and assign these as attributes to the individual models. First, we calculate parameters that assess the quality of the source data for the specific building. For example, the number of points that were available for the 3D reconstruction, the no-data area, and the timeliness of the source data. Second, we calculate parameters that measure the success of the automatic reconstruction, e.g., the root-mean-square error (RMSE) between the reconstructed model and the input points, the maximum error between reconstructed mesh and the point clouds and eventual invalidity codes both in 2D (which means the input data contained an error) and in 3D (which means that the reconstruction failed). For the LoD2.2 building models of the Netherlands, the RMSE is less than 31 cm for 95% of the models and less than 9 cm for 75% of the models (see Figure 7). A more extensive evaluation is in progress. This evaluation is based on the quality parameters that we calculate in order to identify opportunities to further improve our workflow (Dukai *et al.* 2021).

Conclusions

In this paper, we describe the process that we have developed to automatically reconstruct LoD1.2, LoD1.3, and LoD2.2 building models (supporting different reference heights for the block models) for large areas in one reconstruction process and based on the same source data. This provides the user with consistent 3D data of the same area meeting the data-needs of different applications. We monitor quality information throughout the entire process in order to continuously improve the process from input data, preprocessing, and 3D reconstruction to download and use the data in urban applications. In addition, the user can use the quality information to decide on the fit-for-purpose of the data for their own application.

The 3D data that we generate has been a good source to experiment and test all kinds of urban applications that need 3D data, and it is being used in, for example, noise simulations (Stoter *et al.* 2020), wind flow simulations (García-Sánchez *et al.* 2021), and energy consumption calculations (Wang *et al.* 2020; León-Sánchez *et al.* 2021).

Based on experiences and users' feedback, we will improve the process and optimize for different users and applications. These improvements may be generic such as filling the no_data areas in the point cloud with artificial intelligence; better alignment to specific data needs of urban applications, like optimizing the level of detail for specific applications (in relation to processing time); and enriching the data with relevant information, for example distinguish between external and internal walls and their areas for energy-related applications or the estimation of the number of floors per building. In addition, the availability of the next version of AHN (to be expected next year) will provide more reconstruction possibilities, as can be seen in Figure 8. In future research, we will also study more fundamental issues, i.e., how to maintain and manage different temporal and geometric versions of the 3D data and how to better align the different parts in the process to obtain even better results.
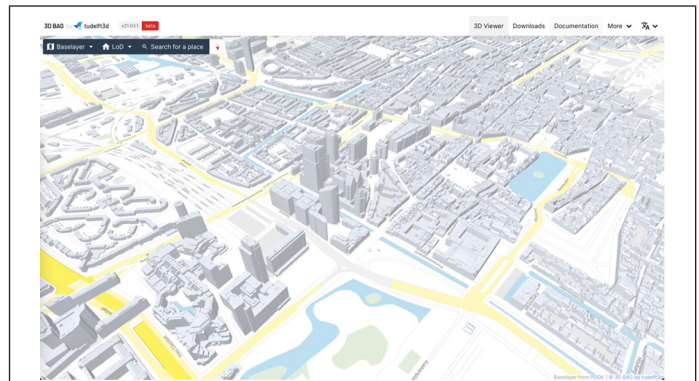

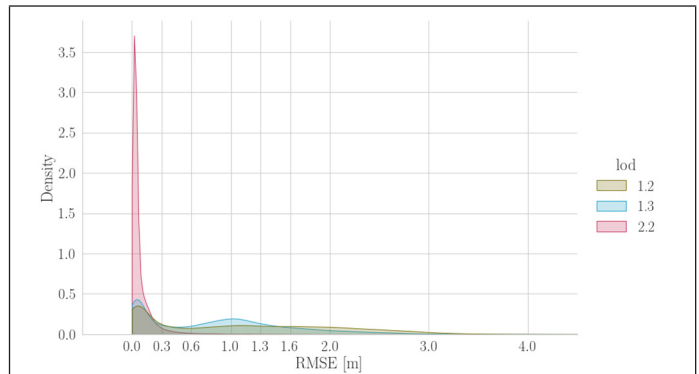Figure 6. Screenshot of the developed 3D viewer available at www.3dbag.nl.

Figure 7. The root-mean-square error (RMSE) between the input point cloud and the LoD2 reconstruction result (Dukai *et al.* 2021).
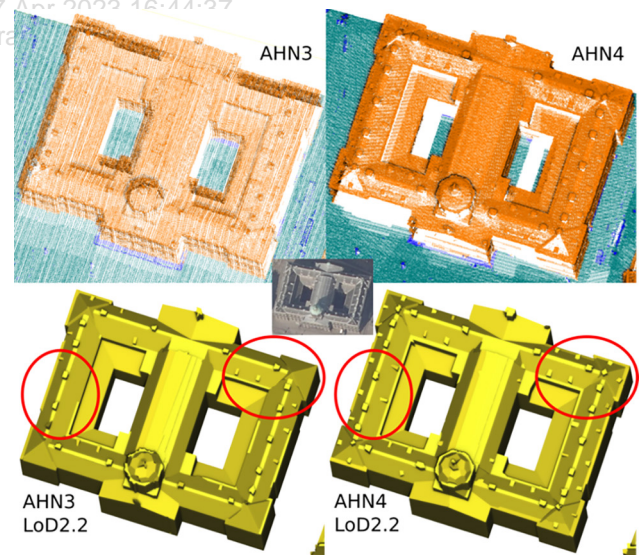

Figure 8. The palace in Amsterdam. AHN3 and AHN4 point clouds and the resulting LoD2.2 reconstruction. More details are reconstructed from the AHN4 point cloud because it has a higher resolution.

References

Biljecki, F., G.B.M. Heuvelink, H. Ledoux and J. Stoter. 2018. The effect of acquisition error and level of detail on the accuracy of spatial analyses. *Cartography and Geographic Information Science* 45(2):156–176.

Biljecki, F., H. Ledoux and J. Stoter. 2016. An improved LOD specification for 3D building models. *Computers, Environment and Urban Systems* 59:25–37.

Biljecki, F., J. Stoter, H. Ledoux, S. Zlatanova and A. Çöltekin. 2015. Applications of 3D city models: State of the art review. *ISPRS International Journal of Geo-Information* 4(4):2842–2889.

Colucci, E.; V. De Ruvo, A. Lingua, F. Matrone and G. Rizzo. 2020. HBIM-GIS integration: From IFC to CityGML standard for damaged cultural heritage in a multiscale 3D GIS. *Applied Sciences* 10:1356. https://doi.org/10.3390/app10041356.

Deren, L., Y. Wenbo and S. Zhenfeng. 2021. Smart city based on digital twins. *Computational Urban Science* 1(4). https://doi.org/10.1007/s43762-021-00005-y.

Donkers, S., H. Ledoux, J. Zhao and J. Stoter. 2016. Automatic conversion of IFC datasets to geometrically and semantically correct CityGML LOD3 buildings. *Transactions in GIS* 20(4):547–569.

Dukai, B., H. Ledoux and J. Stoter. 2019. A multi-height LoD1 model of all buildings in the Netherlands. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* IV-4/W8:51–57.

Dukai, B., R. Peters, J. Van Liempt, S. Vitalis and J. Stoter. 2021. Quality assessment of a large dataset containing automatically reconstructed 3D building models. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLVI-4/W4-2021:17–24. https://doi.org/10.5194/isprs-archives-XLVI-4-W4-2021-17-2021.

Dukai, B., R. Peters, T. Wu, T. Commandeur, H. Ledoux, T. Baving, M. Post, V. van Altena, W. van Hinsbergh and J. Stoter. 2020. Generating, storing, updating and disseminating a countrywide 3D model. Pages 27–32 in *Proceedings ISPRS 15th 3D GeoInfo Conference*, held in London, U.K., 7–11 September 2020.

García-Sánchez, C., S. Vitalis, I. Pađen and J. Stoter. 2021. The impact of level of detail in 3D city models for CFD-based wind flow simulations. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLVI-4/W4-2021:67–72. https://doi.org/10.5194/isprs-archives-XLVI-4-W4-2021-67-2021.

Kutzner, T., K. Chaturvedi and T. H. Kolbe. 2020. CityGML 3.0: New functions open up new applications. *Journal of Photogrammetry, Remote Sensing and Geoinformation Science* 88:43–61.

Lafarge, F. 2015. Some new research directions to explore in urban reconstruction. *Joint Urban Remote Sensing Event (JURSE)*:1–4. https://doi.org/10.1109/JURSE.2015.7120488.

Ledoux, H., K. Arroyo Ohori, K. Kumar, B. Dukai, A. Labetski and S. Vitalis. 2019. CityJSON: A compact and easy-to-use encoding of the CityGML data model. Hugo Open Geospatial Data, Software and Standards 4(4). https://doi.org/10.1186/s40965-019-0064-0.

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

Ledoux, H., T. Commandeur, C. Nagel, S. Trometer and S. Vager. 2016. Automatic reconstruction of simulation-ready 3D city models. https://3d.bk.tudelft.nl/projects/simwind/simwind-final-report.pdf.

León-Sánchez, C., D. Giannelli, G. Agugiaro and J. Stoter. 2021. Testing the new 3D bag dataset for energy demand estimation of residential buildings. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLVI-4/W1-2021:69–76. https://doi.org/10.5194/isprs-archives-XLVI-4-W1-2021-69-2021, 2021.

Open Geospatial Consortium (OGC). 2012. *OGC City Geography Markup Language (CityGML) Encoding Standard 2.0.0 (Document No. 12019)*. Wayland, Mass.: Open Geospatial Consortium.

Pârvu, I. M., F. Remondino and E. Ozdemir. 2018. LOD2 building generation experiences and comparisons. *Journal of Applied Engineering Sciences* 8(2):59–64. https://doi.org/10.2478/jaes-2018-0019.

Public Services on the Map (PDOK). 2021. National Geo-Portal. <https://www.pdok.nl> Last accessed December 2021.

Rottensteiner, F., G. Sohn, M. Gerke, J. D. Wegner, U. Breitkopf and J. Jung. 2014. Results of the ISPRS benchmark on urban object detection and 3D building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing* 93:256–271.

Santhanavanich, J. 2020. Open-Source CityGML 3D Semantical Building Models in 2020. <https://towardsdatascience.com/open-source-3d-semantical-building-models-in-2020-f47c91f6cd97> Accessed DATE.

Stoter, J., R. Peters, T. Commandeur, B. Dukai, K. Kumar and H. Ledoux. 2020. Automated reconstruction of 3D input data for noise simulation. *Computers, Environment and Urban Systems* 80:101424.

Sun, J., P. Olsson, H. Eriksson and L. Harrie. 2019. Evaluating the geometric aspects of integrating BIM data into city models evaluating the geometric aspects of integrating BIM data into city models. *Spatial Science*. 65(2):235–255. https://doi.org/10.1080/14498596.2019.1636722.

Swisstopo. 2021. <https://www.swisstopo.admin.ch/en/knowledge-facts/topographic-landscape-model.html> Last accessed 17 December 2021.

Wang, C., S. Tindemans, C. Miller, G. Agugiaro and J. Stoter. 2020. Bayesian calibration at the urban scale: A case study on a large residential heating demand application in Amsterdam. *Journal of Building Performance Simulation* 13(3):347–361.

Zebedin, L., J. Bauer, K. Karner and H. Bischof. 2008. Fusion of feature- and area-based information for urban buildings modeling from aerial imagery. Pages 873–886 in *Computer Vision–ECCV 2008*. Edited by D. Forsyth, P. Torr and A. Zisserman. Lecture Notes in Computer Science vol. 5305. Berlin: Springer. https://doi.org/10.1007/978-3-540-88693-8_64.