Generation and Dissemination of a National Virtual 3D City and Landscape Model for the Netherlands

Sander Oude Elberink, Jantien Stoter, Hugo Ledoux, and Tom Commandeur

Abstract

This paper describes the generation and dissemination of a national three-dimensional (3D) dataset representing the virtual and landscape model. The 3D model is produced automatically by fusing a two-dimensional (2D) national object-oriented database describing the physical landscape and the national high-resolution height model of the Netherlands. Semantic constraints are introduced to correctly model 3D objects. Three areas from different regions in the Netherlands have been processed in order to develop, improve, and test the automatic generation of a national 3D city and landscape model. Specific attention has been paid to exceptional cases that may occur in a nationwide dataset. Based on the test results, the Kadaster, the national agency in the Netherlands responsible for the production of nation wide geo-information, decided that it is feasible to produce a national 3D city and landscape model that fulfills the specifications that were defined as part of this study. Future research is identified to make the results further ready for practice.

Introduction

Over the past two years, a uniform approach for acquiring, storing, and visualizing 3D geo-information has been explored in a pilot in The Netherlands. In this pilot, over 65 private, public, and scientific organizations have collaborated to push the use of 3D information. The pilot project established the groundwork for a comprehensive national 3D geo-information program. A major result was a proof of concept for a 3D spatial data infrastructure (SDI) covering issues on the acquisition, standardization, storage, and use of 3D data (Stoter *et al.*, 2011).

Besides the need for a national 3D standard realized as City Geography Markup Language (CITYGML) implementation (OGC, 2008 and 2012) (see Van den Brink *et al.*, 2012), the pilot showed the need for a nationwide 3D city and landscape model. This model can serve as a reference for (new) 3D information in a 3D virtual world, and as a basis for 3D planning and management of public space. The 3D base model can be further refined when a project develops. Many large municipalities have 3D data sets, but these are specifically required for the territory of the city and in various formats and resolutions. The pilot has shown promising results for generating a national 3D city and landscape model as a combination of 2D topography with high-resolution airborne Light Detection and Ranging (lidar) data. This paper describes how those results are further developed to generate a national 3D topographic dataset covering the whole of the Netherlands.

The term "topography" may be multi-interpretable, especially in combination of different types of dimensionalities: for some readers the term "topography" implies the relief description of the terrain, for others it means an object-based representation of elements from a cartographic map. The latter definition is the one that we use in this paper, and it refers to both natural features (rivers and forests) and man-made features (buildings and roads). In addition, three types of dimensionality of topographic objects are used: 2D, 2.5D, and 3D. In our context, topographic 2D data are represented as a cartographic vector map describing the objects such as roads, water, buildings, terrain, vegetation, engineering objects, etc. that we use as source data. 2.5D objects are part of the generated 3D data describing the surfaces in the landscape model (e.g., roads, terrain, water polygons; these can be projected to the 2D plane, but are geometrically represented by 2.5D height surfaces), whereas 3D objects refer to volumetric objects in the generated 3D data (mainly buildings). In the generated 3D data road junctions are built up from two 2.5D surfaces representing both the lower and upper road surface. In the remainder of this paper, the term "3D data" refers to the output of our algorithm (containing both 2.5D and 3D objects), whereas "2D data" refers to the input map data source.

This work is collaboration between the University of Twente, the Delft University of Technology, and the Kadaster, the national agency in the Netherlands responsible for the production and provision of nation wide geo-information. Three areas from different regions in the Netherlands have been processed in order to develop, improve and test the automatic generation of a 3D dataset. The test areas include complex interchanges, buildings above roads, bridges, urban areas, and forested areas.

Sander Oude Elberink is with the Faculty of Geo-Information Science and Earth Observation, University of Twente, Hengelosestraat 99, 7514 AE Enschede, The Netherlands (oudeelberink@itc.nl).

Jantien Stoter is with OTB, Delft University of Technology, Jaffalaan 9, 2600 AA, Delft, The Netherlands; Kadaster, P.O. Box 9046, 7300 GH Apeldoorn, The Netherlands; and Geonovum, Barchman Wuytierslaan 10, 3818 LH, Amersfoort, The Netherlands.

Hugo Ledoux is with OTB, Delft University of Technology, Jaffalaan 9, 2600 AA, Delft, The Netherlands.

Tom Commandeur is with HydroLogic Research BV, Elektronicaweg 2, 2628 XG, Delft, The Netherlands.

Photogrammetric Engineering & Remote Sensing
Vol. 79, No. 2, February 2013, pp. 147–158.

0099-1112/13/7902–147/$3.00/0

This paper describes the applied methodology (including the choices that were made), the results, findings, and remaining challenges to obtain a first version of the national 3D dataset in 2012. As our goal is to generate a nationwide 3D dataset, our procedure has to pay specific attention to manage large datasets and to handle any situation. It means that the design should fulfill five main conditions, i.e., our procedure should:

1. Be fully automatic;
2. Deal with inaccuracies in both datasets;
3. Create valid 3D geometries;
4. Completely cover the country; and
5. Handle all -including special- situations.

In addition, since the dataset is intended as a foundation for the 3D Spatial Data Infrastructure (SDI), the interest is not only in the generation of the 3D data but also in the maintenance and dissemination of the 3D dataset.

The structure of the paper is as follows. After a description of related work, the next Section presents the specifications of the 3D dataset that represents the virtual city and landscape model. The proposed 3D reconstruction methodology covering the five conditions is then presented followed by results and a description of how to maintain and disseminate a national 3D dataset. The final section closes the paper with the main findings and conclusions and identifies future research and development.

Related Work

Generation of 3D Geo-information

The research attention on automated generation of 3D information from a combination of 2D topographic vector data and high-resolution lidar data is not new. Two main reasons justify this combinatorial approach. First, existing 2D datasets contain information about the types of object to generate (i.e., building, road, water, etc.) that increases the possibilities to automatically generate 3D information. Second, the 2D topographic datasets often contain rich semantics, such as functionality of objects, which is difficult to obtain from automated acquisition techniques. Previous research on this topic is reported in Haala *et al.* (1998). They describe the combination of detailed Digital Surface Models (DSM) and 2D topographic objects to reconstruct urban objects such as buildings and trees. To enhance the boundary representation of objects, Koch and Heipke (2006) introduce semantic rules when fusing a DSM with 2D map data. These rules ensure a logical and consistent 2.5D representation (i.e., height surface representation) of the topographic objects. Oude Elberink and Vosselman (2009a) propose an extension of these semantic rules to real 3D, to ensure that actual 3D objects, such as bridges and interchanges can be represented realistically.

In recent years the density of point clouds, either generated from lidar data or aerial imagery, has increased up to several points per square meter. The increasing point density is considered important in 3D reconstruction algorithms because it allows for a more detailed and reliable description of objects. Even more important for national mapping agencies is the national coverage of such datasets.

National 3D Datasets

The generation of 3D datasets of national coverage is relatively new in the research domain for two main reasons. First, the methodologies for 3D reconstruction usually focus on how 3D data can be generated from a combination of lidar data and 2D data, but they do not address the question of how the enormous amount of lidar data can be processed if larger areas are being considered. Specifically, with the recent developments of very high resolution lidar data, this has become a serious problem, which also requires other expertise than 3D mapping. Second, uniform national 3D datasets can only be generated in an automated manner from source (2D and 3D) datasets that are uniformly available for the whole country. This is rarely the case.

The 3D Dataset

This section presents the two source datasets and explains the specifications of the 3D object-oriented topographic dataset that were defined as part of our study.

Source Data: The 2D Topographic Dataset

In the Netherlands, two 2D topographic vector data sets are candidates for extension into 3D: the large-scale base dataset, modeled according to Information Model Geography (IMGEO) and the national topographic vector map at scale 1:10 000. The first data set is the object-oriented version of the large-scale base map of the Netherlands, with a scale of approximately 1:500 (1:1000 in rural areas). The new version of the model IMGEO has recently been established and nationwide coverage of IMGEO data (in 2D) is expected to be ready in 2015. Providers of the IMGEO data are municipalities, provinces, agencies responsible for the railway, and high way network infrastructure (Prorail and Rijkswaterstaat), and Kadaster. The second candidate source dataset for extension in 3D is an object oriented topographic dataset at scale 1:10 000 (called TOP10NL) available for the whole country since 2006 and maintained by the Dutch Kadaster. Although IMGEO has been fully prepared for extension into 3D by integration of the information model with the OGC CITYGML standard (Stoter *et al.*, 2011; Van den Brink *et al.*, 2012), we selected TOP10NL as the most suitable candidate for a nationwide 3D base dataset forseveral reasons. First, TOP10NL data is available nationwide while 2D IMGEO data will only be generated in the coming years. Second, TOP10NL is less detailed than IMGEO and therefore better suitable for fully automatic 3D object reconstruction since fewer details are required (therefore less special cases). Despite fewer details, we are of the opinion that the resulting dataset generated from TOP10NL is appropriate for a nationwide dataset (i.e., acceptable performance for nationwide applications), and that it can be further refined when applied in future projects. Finally for an automated workflow, it is essential that the data have been acquired in a standardized manner. Since TOP10NL has only one data provider, the data are homogenous for the whole country, and the beneficial side effect is that the 3D result will be a consistent national 3D dataset. Consequently, it was decided that for the moment, 3D TOP10NL is the best option to generate and disseminate a nationwide 3D base dataset.

The main classes in TOP10NL (see Figure 1) are: Road, Land Use, Water, Railway, Layout Element, Registration Area, Building, Geographical Area, Functional Area, and Relief.

The land use, road and water objects that can be seen from above, form a complete partition of the country without any gaps or overlap. Consequently, buildings, and also functional and geographical areas, overlap with other objects. In addition infrastructure objects can cross (i.e., overlap in 2D). This is modeled using two attributes assigned to infrastructural classes with polygon geometry (Water and Road): a "type of infrastructure" attribute, which models whether the infrastructure object is a connection or a crossing and the "height level" attribute. This last attribute models the relative order of objects where a value of '0' indicates that the object is on top of a stack of two or more objects (i.e., visible from above). All objects at height level '0'constitute the planar partition.

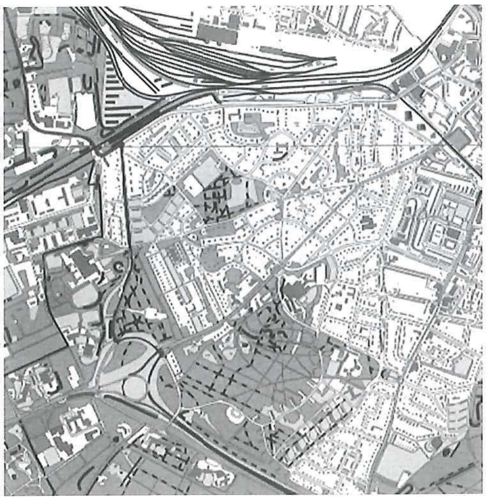


Figure 1. TOP10NL data of the test area in Amersfoort.

Source Data: The Lidar Dataset
3D TOP10NL is generated by combining the TOP10NL vector data with the national height model of the Netherlands (called AHN; see AHN (2012)), which is acquired by airborne lidar systems. 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 is acquired with an average point density of 10 points per square meter. Currently a third version, possibly enriched with pulse count information, is being considered.

The AHN data set has been acquired, assessed, filtered, and stored in a standardized manner in terms of specifications on point densities, overlapping areas, and quality reports. A useful property of the height model is the filtering step that separates points on ground and points on elevated objects such as trees and buildings, as shown in Figure 2. The companies that acquire the data also filtered the data, and therefore the AHN point data is available as two products for any customer. In our procedure, we use the second version of the AHN (i.e., AHN-2) data, which includes both the ground and non-ground points. Notice that the dataset is also available as a grid (different resolutions are available), but we use the original samples.

Specifications of the 3D TOP10NL Dataset
The aim of 3D TOPNL is to have a 3D version of the 2D vector map covering the whole country. Consequently, the map objects should still be available in the 3D model but now with the 2.5D (i.e., height-surface) and 3D (i.e., volumetric) representations. Not all TOP10NL classes are relevant or suitable for an extension to 3D, for instance, functional land use (e.g., zoo, cemetery) and point-based features (e.g., striking objects) are not. We therefore selected the most appropriate classes, which are road, water, geographic land-use, and buildings.

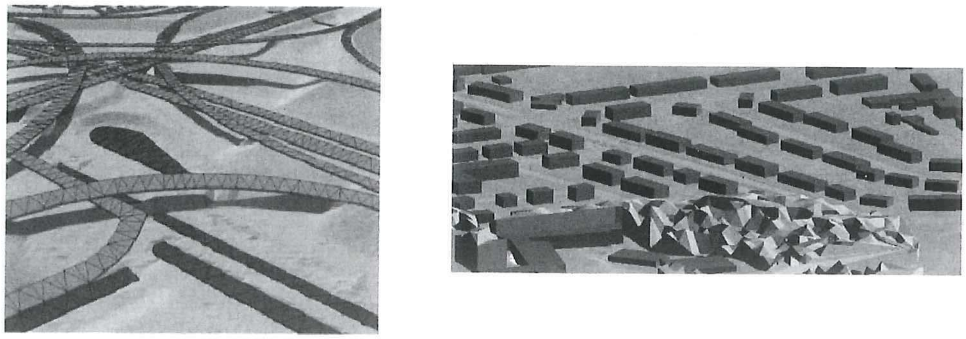
The type of 3D representation that is generated (i.e., 2.5D or 3D) depends on the class. Following the characteristics of the TOP10NL objects, we decided to obtain volumetric geometries (3D) for buildings and height surface geometries (2.5D) for the other objects (road, water, and land-use). Multi-level infrastructural crossings are represented through surface geometries that connect in space (Figure 3a).

"Forest" (which is a type of geographic land-use) is a specific case, since the AHN data contains two products at those locations: the point data at the terrain level and point data representing the heights of the trees. To represent both types of information in the resulting dataset, we create a height surface from the tree heights that we extrude downwards to the terrain level, see Figure 3b. Both types of height (tree heights and terrain heights) are relevant information at those locations.

Methodology for the Generation of a Nationwide 3D Dataset
This section presents the methodology and engineering choices to generate a nationwide 3D dataset. It is organized according to the five main conditions that we should fulfill previously introduced:

1. Fully automatic reconstruction;
2. Dealing with inaccuracies in both datasets;
3. Creation of valid 3D geometries;
4. Covering a large area, i.e. a complete coverage of the country; and
5. Handling all -including special- situations.

Figure 2. Lidar (AHN) data: (a) ground points ranging between 5 m (dark grey) and 22 m (white) above means sea level (MSL), and (b) non-ground points ranging from 5 m to 48 m above MSL. Black areas do not contain laser points because points are either removed during the filtering or because these are areas with water.



(a) (b)

Figure 3. (a) Multi-level infrastructural crossings, and (b) a urban scene including forest areas.

Automatic 3D Generation Process

The procedure to obtain the 2.5D and 3D representations from TOP10NL object builds on the work of Oude Elberink and Vosselman (2009a). The 3D generation of polygon boundaries is a two-step procedure: first heights are assigned to polygons following specific rules; second, information on neighboring polygons is used to refine the height values.

Polygon-based Processing

The 2.5D and 3D representations are generated using the 2D geometry, the accurate 3D point-based surface description of the airborne lidar data that fall within the object, and specific characteristics of the TOP10NL object class identified by us, such as "roads are continuous surfaces," "water is flat," and "terrain can have height variance within each polygon."

Per-object polygon, lidar points are selected and processed using the 3D modeling constraints of every object class; see Table 1. Ground points have been used for the generation of the classes "Road," "Terrain," and "Water," whereas non-ground points are used to assign lidar data to "Building" and "Forest" polygons.

The surface description within the polygon depends on the object class. Water and road objects are constructed by producing a constrained TIN using the 3D coordinates of the boundary. In terrain polygons, lidar points are added to allow a more detailed height description within the surface.

In general, the basis of our approach is a series of point-in-polygon operation, meaning that all lidar points within a polygon are assigned to that polygon. However, for multiple reasons, some points may not be appropriate for further processing. One reason is a possible planimetric offset between map and lidar data: the offset causes that object boundaries in TOP10NL do not correspond to the object structure of the lidar points. In Figure 4 an example shows an offset of more than two meters between a bridge boundary in the topographic vector data and the corresponding lidar points. For a correct reconstruction of the bridge, only the lidar points on top of the bridge should be selected, thus excluding the points at the water surface level.

Theoretically, it could be decided to tolerate a maximum offset between the datasets. However, our experience show that offsets occur of up to two meters, specifically since the 2D data set is acquired (and therefore generalized) at mid-scale (i.e., 1:10 000). Our automatic modeling approach is designed such that it can deal with these inaccuracies. The lidar data is segmented into locally planar surface patches. Subsequently, the program checks the most frequent segment number in a radius of five meters. Those points are assigned to the polygon for further processing and others are excluded. In the example of Figure 4, only the points at the bridge are selected and the points at the water surface are excluded in the 3D generation of the bridge.

The unique ID number of the original topographic object is kept during the reconstruction process. It means that per topographic object, it is known which original lidar points have been selected and how the further processing was performed.

Including Neighborhood Relations

The next step in the 3D reconstruction procedure refines the initial heights that are assigned to the polygon boundaries

Table 1. Class-based Constraining of Processing Lidar Points within Polygons

Class	Lidar data taken from	3D Representation type / Semantic constraint	Initial height of object points on boundary	Surface description
Water	Ground	Horizontal plane	All object points are set to average height	Determined by triangulation of boundary object points
Roads	Ground	Locally planar	Each object point is determined by height of local fitted plane	Determined by triangulation of boundary object points
Terrain	Ground	May vary locally	Each object point is determined by height of local fitted plane	Lidar points are inserted inside polygon, followed by constrained triangulation
Buildings	Non-ground	Horizontal plane, LoD 1	All object points are set to average height	Determined by triangulation of boundary points
Forest	Non-ground	May vary locally	Each object point is determined by height of local fitted plane	Lidar points are inserted inside polygon, followed by constrained triangulation

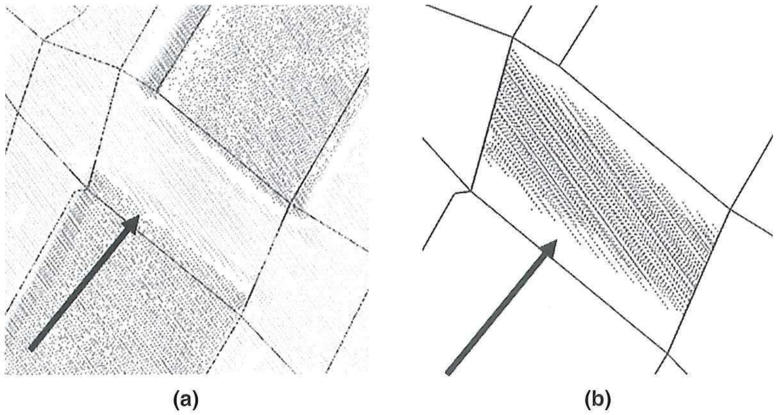


Figure 4. (a) Discrepancy between bridge boundary in the topographic dataset and Lidar data, indicated by black arrow, and (b) Lidar points for further processing of the bridge polygon are selected for further processing.

using the 2D neighbor relationships. The final height of a boundary point depends on which height was determined for the neighboring polygon at the same location. Figure 5 shows that the planar partition in the 2D vector map assures that each object point connects to two or more polygons. Depending on the classes of both neighboring object polygons, specific constraints (see Table 2) are applied and the final 3D boundaries are created accordingly; see Figure 6.

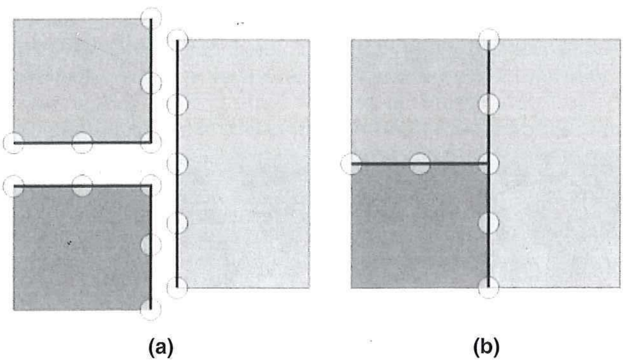


Figure 5. (a) Initial height calculation based on single polygon processing, and (b) Final height calculation by considering neighbor relationships.

Table 2 lists the possible relationships and their constraints. Per object point the classes of the adjacent polygons are identified and heights are adjusted accordingly. For example, object points that connect to water polygons will always keep the height of the water polygon. If terrain polygons are adjacent to this water polygon the corresponding boundary of the terrain will be "pulled" to the water height. Vertical "walls" are added to objects from the classes "building" and "forest" (which is a specific type of land-use). These "walls" extrude the polygons downwards to the height at the terrain, i.e., at the height of the Digital Terrain Model (DTM). The generated "walls" get the class label of the above ground objects, e.g., building or forest. The lowest DTM height that is located below a building is taken as a constant height for the ground height of the building. Consequently, the floors of buildings built on a hillside are approximated. An extra floor polygon is added to the building, to make it a solid block object. These additional polygons are shown as dotted lines in Figure 6.

We have considered generating buildings with roof shapes instead of building blocks, based on large-scale 2D building geometries available in the building register (BAG, 2012) as proposed in Oude Elberink and Vosselman (2009b). But experience has shown that this is not an appropriate solution. First, because the 2D large-scale building objects of the building register do not fit with TOP10NL data, the

Table 2. Modeling Constraints based on Neighborhood Relationships

	Water	Road	Terrain	Building	Forest
Water	Both keep own height	Both own height, create additional polygon below road	Take water height	Both keep own height, create wall in-between	Both keep own height, create wall in-between
Road		Average if close in height	Take road height	Both keep own height, create wall in-between	Both keep own height, create wall in-between
Terrain			Take average of both heights	Both keep own height, create wall in-between	Both keep own height, create wall in-between
Building				Both keep own height	Both keep own height
Forest					Both keep own height

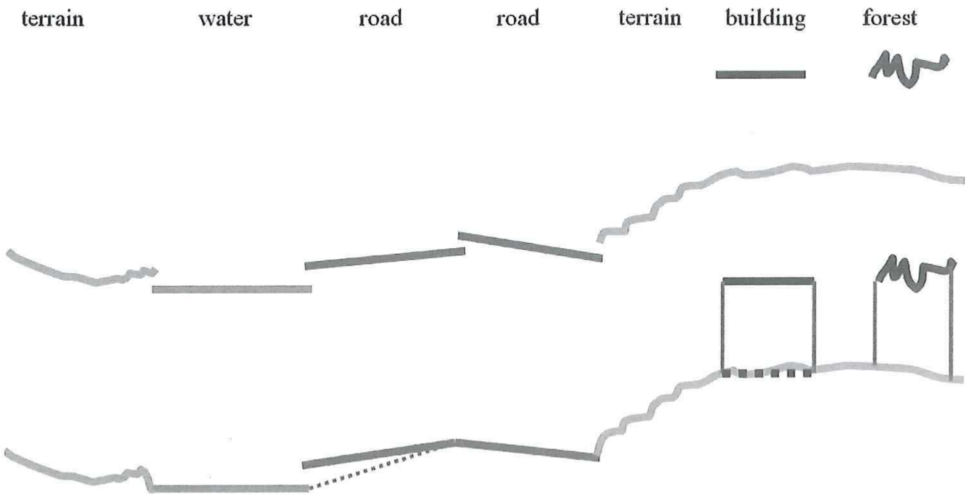


Figure 6. Schematic profile visualizing the initial height (top) and final height (bottom) depending on neighbor relationships from Table 2. Additional polygons are shown as dotted lines.

geometrical differences between the two data sets is mainly caused by different acquisition methods: TOP10NL is obtained from aerial photographs and applies (little) generalization, while the building objects in the building register are obtained form terrestrial measurements. Another problem for generating the roof shapes is that this process cannot be done fully automatically in all cases, which is an important prerequisite for a nationwide dataset. Finally, the complex details of the roofs do not fit with a 1:10 000 representations of 3D data.

Revealing the Inaccuracies in the Result

To be able to use the 3D data in a correct manner, it is important to provide accuracy measures of the constructed data. Therefore, at the end of the 3D generation procedure, the program calculates and analyzes the differences between heights of the original lidar points and the 3D model (Plate 1), based on Oude Elberink and Vosselman (2011). Green lidar points in Plate 1f are within 20 cm to the 3D model; the yellow points are between 20 cm and 50 cm (these hardly exist in

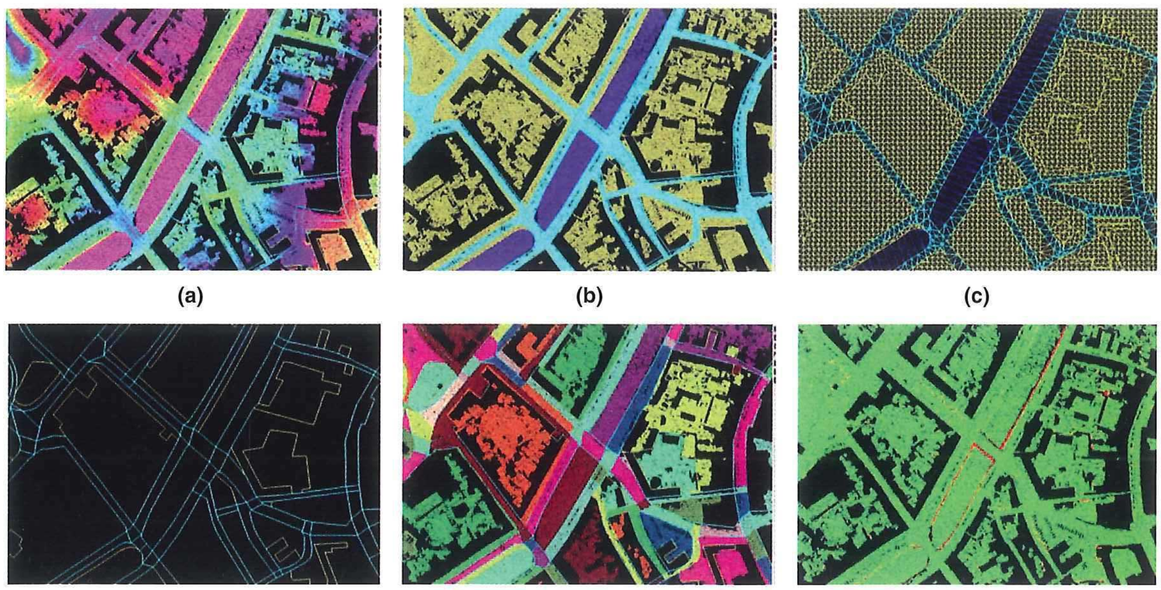


Plate 1. Overview of the complete workflow: input data, (a) lidar data, and (d) topographic data; Intermediate processing results: (b) lidar points colored by class label, and (e) object; Results: (c) triangulated 3D model colored by class label, and (f) quality indicator. Lidar points within 20 cm to the 3D model are colored green; red points are further than 50 cm from the 3D model.

this area); points that are further away are red. Any systematic behavior in residuals between the two is an indication that either the two data sets, AHN and TOP10NL, do not match or the assumptions do not hold for that particular object. The main reason for discrepancies is the time difference between production of the TOP10NL vector data and AHN data. To increase the accuracy, the user can select polygons with a certain percentage of lidar points that show large residual values. If necessary, these polygons can be adjusted manually to obtain higher accurate data.

Reconstructing Valid 3D Geometries

Our methodology to reconstruct the 3D TOP10NL dataset ensures that the 3D volumetric geometries generated for buildings and forests are *geometrically* valid. While different definitions of a valid 3D object are used in different disciplines, we focus on the definition given in the ISO 19107 Standards (ISO, 2003) and implemented with Geography Markup Language (GML; OGC, 2007). A GML Solid is defined as follows: "The extent of a solid is defined by the boundary surfaces as specified in ISO 19107:2003. gml:exterior specifies the outer boundary, gml:interior the inner boundary of the solid" (OGC, 2007). In the case of the 3D TOP10NL dataset, no inner boundaries will be created (since extrusion is used), but a solid will be represented by its boundary surfaces. The criteria for a valid 3D geometry are the following: (a) a solid should be "watertight," i.e., there should be no gaps in its boundary; (b) it must be simple: no self-intersection of the surfaces is allowed; (c) each surface should be planar; (d) the normal of each boundary surface should point "outwards." These rules have been taken into account when developing the reconstruction algorithm, and the resulting solids are valid (see Figure 7).

Ensuring valid solids has many advantages. First, it facilitates their use by others and their processing with other tools (such as the DataBase Management System Oracle Spatial 11g) as users will not face problems when importing them and converting them to other formats and representations. A tool to convert CITYGML datasets to Keyhole Markup Language (KML) could for instance not function properly if the surfaces of the buildings are not planar. Second, it permits users to use the 3D model for spatial analysis. While at this moment 3D models are mostly used only for visualization, we believe that in the near future they will be used for analysis in different domains. CITYGML already has Application Domain Extensions for noise modeling, flood simulation and for utility networks (OGC, 2012b). Other examples of 3D analyses are disaster management (Kolbe *et al.*, 2008), urban planning (Yasumoto *et al.*, 2011) and 3D cadastre (Stoter and van Oosterom, 2005). That is also why we aim at an object-based 3D model and not at a relief description of the terrain possibly overlaid with a topographic map.

It should be noticed that in our current implementation each extruded building-footprint becomes a valid solid, but the buildings as a whole are not *topologically consistent*. Indeed, two adjacent buildings will have duplicated surfaces that might not have their vertices at the same location. The solution to that problem is to consider topological relations in 2D when extruding (see Figure 8; Ledoux and Meijers, 2011), but we have not included that in the implementation yet.

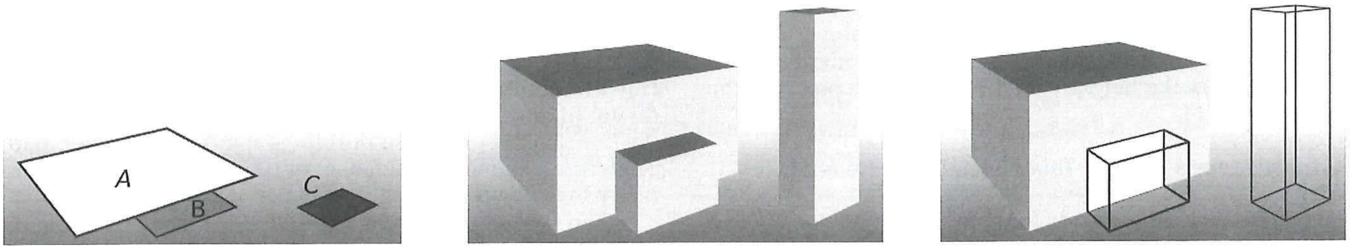
Nationwide Processing

The 2D version of TOP10NL is a seamless product, and that is also the goal for the 3D version. The developed procedure assigns height values to each polygon, also taking into account the information from the neighboring polygons. To make the procedure that uses the very large AHN point dataset possible and efficient, at this moment we resort to a simple solution: we split the input datasets into tiles of about 1 km × 1 km, and each tile is processed separately. The AHN dataset is already tiled in that way (it is stored this way), and we perform the tiling on the vector map of TOP10NL (notice that we are not splitting polygons, but we simply perform a selection of the polygons intersecting the 1 km × 1 km tile).

Because we need the knowledge of the neighboring polygons, we exclude the polygons intersecting the boundary of a tile (those not completely contained inside the tile) and process these after the ones from a neighboring tile have been processed. This procedure yields exactly one 3D version for each of the 2D TOP10NL objects, assuring a seamless 3D product. It is however rather slow as several selections of the polygons and AHN samples have to be performed. We are currently investigating alternatives where the space is not partitioned according to an arbitrary tiling, but where features are used (Briat *et al.*, 2011; Stoter *et al.*, 2011) have already performed such a partitioning of TOP10NL for carrying out cartographic generalization. They used highways and made sure that features do not overlap the partitioning. If we use the same partitioning, we can process each tile almost independently.

Special Cases

Applying the procedure previously described to the different test areas identified specific situations that needed to be solved. These situations including the way it was solved are described in the remainder of this section. All problems listed here are detected and solved automatically, without any manual selection or interpretation.



(a) (b) (c)

Figure 7. (a) Polygons in the plane, (b) Polyhedra obtained by extrusion of the polygons as shown in (c). To be topologically consistent, the front polygon of the polyhedron as extrusion of A should be modeled with two polygons (Figure from Ledoux and Meijers (2011)).

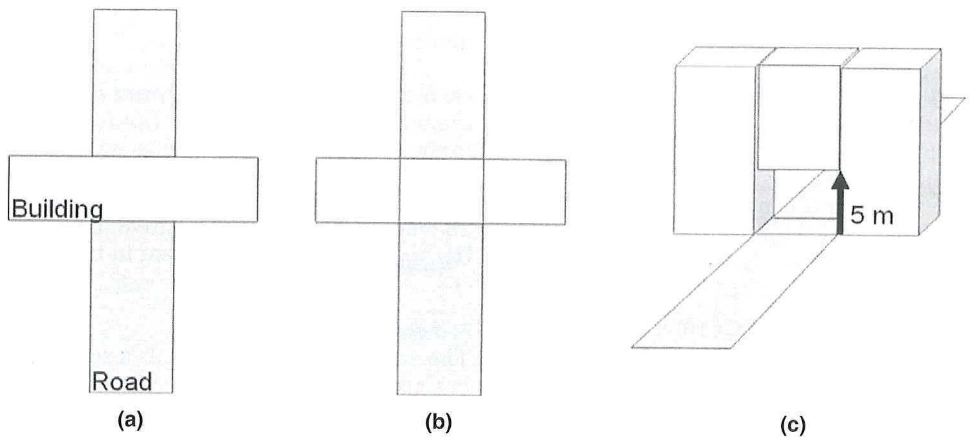


Figure 8. (a) Bridge building above a road gets the ground floor 5 m above (c) DTM (road) level. The building polygon is intersected by the road polygon (b).

Bridge Buildings over Roads

Problem

Because all buildings are extruded downwards to the terrain, buildings over roads are placed on top of the roads.

Solution

The geometry of buildings over roads is split into the parts that are located on the terrain and the part that is located above the roads. For the parts that are located on the terrain the "normal" procedure is applied, i.e., buildings are extruded downwards to the terrain. For the parts located above the road the geometries are extruded downwards to the level of DTM +5 meters (see Figure 8). Although this is an arbitrary value, this solution assures that it is possible to navigate over the road below the building in the dataset. Bridge buildings can be detected automatically because they have a specific attribute in the topographic dataset. Therefore, the generation of the buildings can be done automatically as well.

Interchanges

Problem

Road objects below viaducts do not continue because at those locations 2D TOP10NL contains a specific geometry (indicated as crossing at height level <0) and AHN does not contain height points at the terrain level (because they are occluded by the viaduct).

Solution

2D road polygons are generated from TOP10NL road objects that continue under the viaducts and are used as input for the reconstruction procedure. At the location below the viaduct, height values for the road polygon boundaries are interpolated from the height values on the road objects in the neighborhood.

Two-dimensional Polygons Touch in 2D, but Not in 3D

Problem

Gaps may occur where objects touch in 2D but do not touch in 3D because of a height difference. An example is a road that runs along the water at a higher level than the water.

Solution

For the road-near-water situation extra polygons are created from the road at DTM level which connect to the water; see red polygons in Figure 9. If those gaps occur between two land-use objects, the objects are "glued" together by connecting the vertices.

Too Few Lidar Data Available to Reconstruct the Object in 3D

Problem

Because of the way height data is collected (i.e., using airborne laser altimetry), in some situations too few lidar points are available to reconstruct 3D objects. If the point density of a polygon is less than 10 percent of the average point density, this polygon is considered to have too few Lidar points.

Solution

The polygons without a sufficient number of lidar points, are connected to the neighboring polygons at ground level.

Buildings Blocks that are treated as One Object but have Varying Heights in Reality

Problem

TOP10NL building polygons are too coarse to reflect the height variance that occurs within buildings. One example is shown in Figure 10 for one building of the TU Delft Campus, one can clearly see that resulting 3D model is far from the reality.

Solution

We have not completely solved the problem yet, but there are two potential solutions.

The first one is to use geometries from the large-scale building register (*Basisregistratie Adressen en Gebouwen*; BAG, 2012). Our experiments have shown that those 2D geometries can better reflect the true height variance over buildings (i.e., building parts are usually distinct polygons). However, the 2D building objects from the building register do not fit within the TOP10NL dataset (as mentioned earlier), which may result in buildings standing on top of roads. Therefore, the reconstructed 3D buildings also may be incorrectly located on the roads.

The second potential solution looks more promising and is a method we are currently developing. As shown in Figure 11a, the basic idea is to decompose the footprints of TOP10NL into polygons, and then to perform the steps previously described for each part of the decomposed building (i.e., find their average elevation). Parts having similar heights (we use for instance a threshold of 20 cm) are merged in a postprocessing step to create the result shown in Figure 11b.

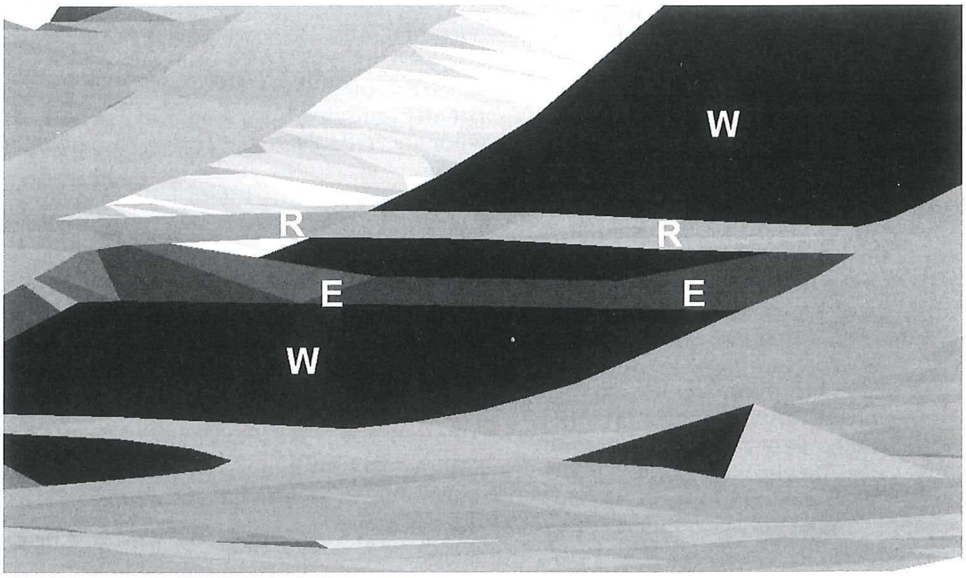
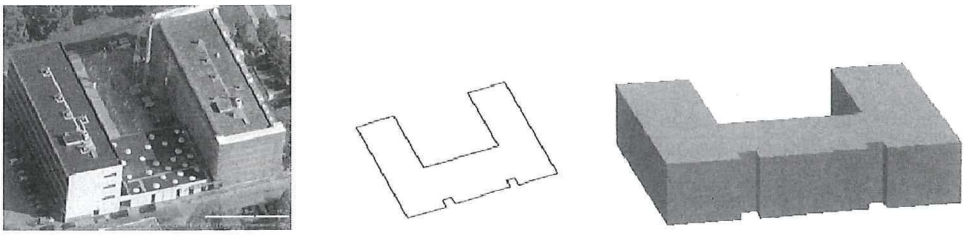
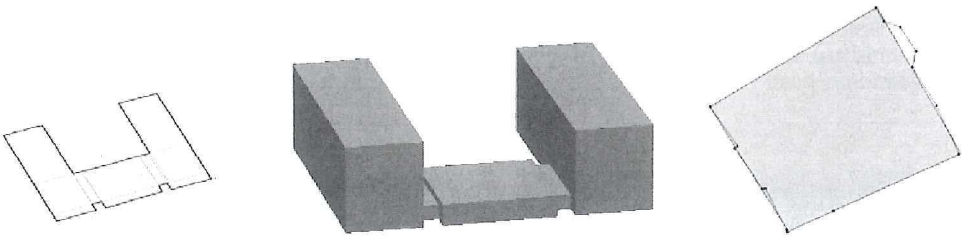


Figure 9. Extra polygons (indicated by 'E') below the road surface ('R') at water level ('W').



(a) (b) (c)

Figure 10. (a) Building at TU Delft campus, (b) TOP10NL footprint, and (c) 3D building obtained by extrusion.



(a) (b) (c)

Figure 11. (a) Footprint from Figure 11 decomposed into parts, (b) Resulting 3D geometry created, and (c) A building footprint (black line with vertices) that has been generalized and where parallelism has been preserved (grey surface).

The rationale behind the method is that buildings are usually symmetrical, and if we decompose the footprints by "prolonging" the line segments that it is most likely, then the locations of the height differences will be created. We implemented the algorithm as described in Kada and McKinley (2009). To ensure that the decomposition does not create long and skinny polygons, we also generalize the footprints during the decomposition (also according to Kada and McKinley (2009)). Figure 11c shows (for another building) that we modified the generalization algorithm so that generalized angles are at 90 degrees (when possible) and that parallel walls are preserved.

Experiments and Results

Three areas from different regions have been processed to refine and test the automatic generation of a 3D dataset of that area. In the test, datasets were copied to disk and the regions of interest were cropped. In Figure 12 part of an urban scene is presented to show the ability of the program to correctly reconstruct bridges, even in the case of a discrepancy between topographic data and lidar data (as shown in Figure 4).

An overview of part of the cities of Middelburg and Amersfoort are shown in Figure 13. As the processing time depends on the number of polygons, number of neighbor relationships, and complexity of the scene, it is of interest to analyze the differences in processing time for each step per test area.

Processing has been done on a standard desktop computer, containing an Intel Core™ Duo CPU, running at 3 GHz and 3GB of RAM memory. Table 3 shows that most time is spent on fusing the datasets, i.e., assigning polygon information to lidar points and vice versa, and on the 3D boundary generation of ground objects. The latter step includes the checking of the height of the neighboring polygons, which is a time consuming step.

To decrease processing time, we implemented a lidar data reduction by factor 3 for non-ground objects, e.g., for producing the building blocks. Our argument is that for producing building blocks (i.e., one height for each building) it is not necessary to keep a point density of ten points per square meter. Our experiences confirm this assumption. Further increase of processing performance is work in progress.

Maintenance and Dissemination of 3D TOP10NL

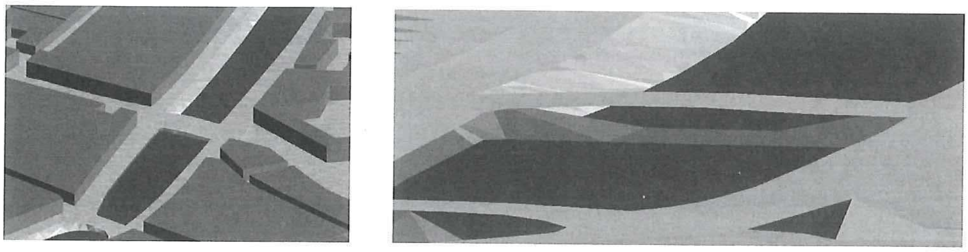
After the generation of 3D TOP10NL, the next question is how to maintain and disseminate the data. For the maintenance of the product, it is relevant to question how to guarantee that 3D TOP10NL remains synchronized with the source TOP10NL. And whether the 3D data is managed separately from 2D TOP10NL or should it always be generated on the fly?

For the dissemination of 3D TOP10NL data we are currently studying how users can access and use the information. For 3D TOP10NL we distinguish two types of users: users who want to access the 3D data itself for spatial operations and users who are only interested in visualizing 3D data. For both types, we acknowledge that 3D TOP10NL may reach a new and wider public than the "traditional" users of TOP10NL. This is one of the motivations to make the first version of 3D TOP10NL available soon, and to be able to receive feedback early in the process.

To disseminate the data, we are studying KML and (City) GML as exchange formats in which the original TOP10NL objects are still available and queryable. We plan to provide these datasets per areas of our partitioning of the country (as previously explained), so that each object is present in only one tile.

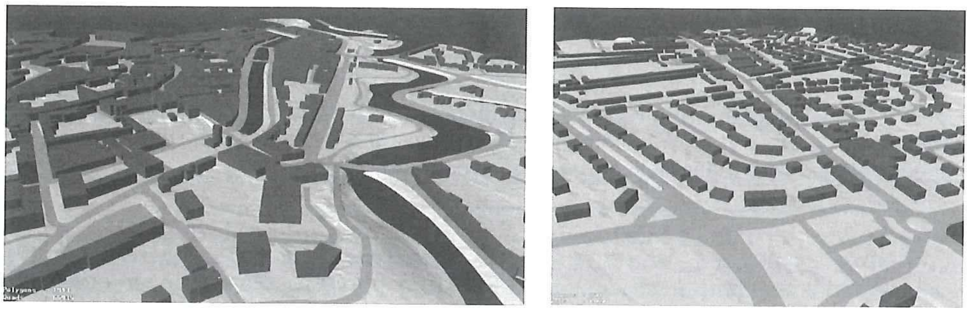
To disseminate the datasets, it is possible to make use of the Web Feature Service (WFS), which supports 3D features such as solids; surfaces can also have 3D coordinates. However, at this moment there are very few clients supporting the standard in 3D.

For simply visualizing the data, we are evaluating KML, WEBGL (Web-based Graphics Library) and the WVS (Web View Service) as possible alternatives. WEBGL is a software library that extends the capability of the JavaScript programming language to be able to view 3D graphics within any compatible web browser, i.e., no plugins need to be installed (which is perfect for non-expert users). Our preliminary results have



(a) (b)

Figure 12. (a) 3D Model of an urban scene, and (b) including real 3D objects such as bridges.



(a) (b)

Figure 13. (a) 3DTOP10NL of Middelburg, and (b) Amersfoort.

Table 3. Performance in Terms of Processing Time per Step

	Size of datasets (number of polygons, number of Lidar points)	Fusing topographic data and Lidar (minutes)	3D boundary generation (minutes)	3D surface reconstruction (minutes)	Total processing time (in minutes), and minutes per object
Amersfoort (ground)	659, 5.3 Million	50	131	7	188, 0.29 per object
Amersfoort (non- ground)	2707, 5.5 Million	70	11	1	82, 0.03 per object
Middelburg (ground)	814, 4.8 Million	79	132	4	215, 0.26 per object
Middelburg (non- ground)	362, 2.9 Million (reduced by factor 3)	59	3	1	63, 0.17 per object
The Hague (ground)	670, 6.9 Million	263	117	6	386, 0.57 per object
The Hague (non- ground)	941, 2.5 Million (reduced by factor 3)	34	2	1	37, (0.04 per object)

shown that WebGL is a viable alternative, although in the meantime KML is used since for our study area it has shown good results; see Figure 14. WebGL and KML are not appropriate when computing resources are limited, for instance on a smartphone. We are therefore currently investigating the use of a WVS: images from a 3D model are generated on a server and served to the clients, allowing them also to use symbology and to have some level of interaction with the 3D model (Hagedorn *et al.*, 2009).

The visualization of 3D TOP10NL should ideally resemble that of TOP10NL (i.e., similar symbols should be used), which requires further investigation. In addition, further research is needed into the visualization of forest. The reconstruction software was adjusted to obtain heights of individual trees that are located in TOP10NL forest areas. This information can be used to place tree models in the 3D model so that the forests look "real."

Conclusions and Future Work

This paper presents the work carried out by the University of Twente, the Netherlands Kadaster, and the Delft University of Technology to produce and disseminate a nationwide 3D city and landscape model at midscale (approximately 1:10 000) based on TOP10NL vector data and high resolution lidar data. The methodology is described as well as the choices that were made.

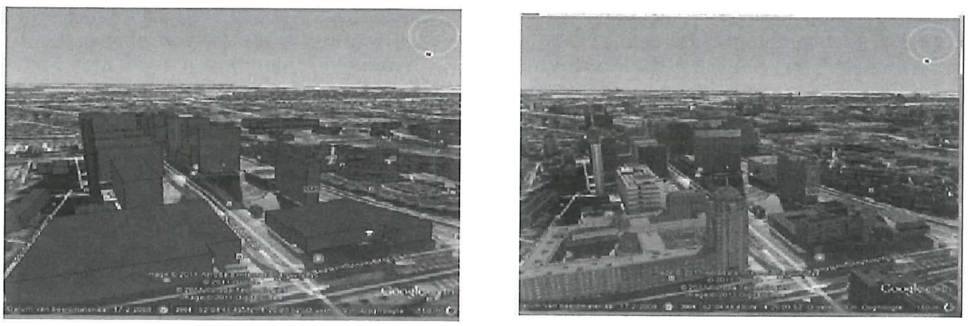
Regarding the specifications of the resulting 3D dataset (i.e., what should be its content?), it was decided that the 3D dataset contains a 2.5D representation of the terrain-, the water-, and the road-objects, and a block-representation for the buildings (volumetric objects). Complex interchanges such as viaducts and bridges are represented by a combination of several 2.5D representations of the particular infrastructure objects.

Based on the test results, the Kadaster has decided that it is feasible to produce a nationwide 3D city and landscape model that fulfills the specifications that were defined as part of this study. At the same time, the results of different projects have shown that a nationwide 3D dataset would foster different applications in 3D. The fact that TOP10NL data is, since 01 January 2012, open and freely available creates the urge to make it also available in 3D.

Therefore in a follow up research project, the results for the three test areas are being applied to other areas to obtain a data set for the whole country. Specific attention is being paid to make the 3D reconstruction procedure efficient as well as to maintain and disseminate the 3D data. To get feedback on the 3D product early in the process, the current results have been made available for the wide public.

Future research has been proposed and granted to enhance the quality of automatically generated 3D building models with roof shapes. As the 2D large-scale building objects (available for the whole country) are open data, the highly detailed 3D building data could be delivered as an additional product.

Another challenge is the management and the processing of the huge lidar data set, which is estimated to contain around 200 billion elevation points. At this moment, simple



(a) (b)

Figure 14. Examples of models using KML format for viewing: (a) 3D TOP10NL buildings in KML, and (b) Models currently available through Google Earth™.

text files are managed, and that does not help applications such as ours to scale to the whole country. One solution is to store all the data in a database management system: Oracle Spatial 11g has support for point clouds (Finnegan and Smith, 2010) and we plan in the near future to use it (or an open-source alternative, if available).

For further research on automatic interpretation of lidar data, it is of high interest to store and analyze this massive fusion process of 2D data and lidar data. This can be used for obtaining knowledge on behavior of lidar data in relation to specific classes, objects or complete regions, such as the differences between relief structure of terrain polygons in the west (clay area) and the east (sandy areas) of The Netherlands. This knowledge could potentially improve the 3D reconstruction process, as it provides statistical input for optimizing parameter settings during the selection and further processing of lidar data into 3D information.

Acknowledgment

This work is supported by the Dutch Technology Foundation STW, which is part of the Netherlands Organization for Scientific Research (NWO) and partly funded by the Ministry of Economic Affairs, Agriculture and Innovation (Project Code: 11300).

References

AHN, 2012. Homepage of Actueel Hoogtemodel Nederland (AHN), URL: *www.ahn.nl* (last date accessed: 13 November 2012).

BAG, 2012. Viewer of building and address register (BAG), URL: *bagviewer.geodan.nl/index.html* (last date accessed: 13 November 2012).

Briat, M.-O., J.-L. Monnot, and E.M. Punt, 2011. Scalability of contextual generalization processing using partitioning and parallelization, *Proceedings of the 14th Workshop of the ICA commission on Generalisation and Multiple Representation*, Paris, France.

Finnegan, D.C., and M. Smith, 2010. Managing LiDAR topography using Oracle and open source geospatial software, *Proceedings of GeoWeb 2010*, Vancouver, Canada.

Haala, N., C. Brenner, and K.-H. Anders, 1998. 3D urban GIS from laser altimeter and 2D map data, *Proceedings of ISPRS Commission III Symposium on Object Recognition and SceneClassification from Multispectral and Multisensor Pixels*, International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Columbus, Ohio, pp. 339-346.

Hagedorn, B., D. Hildebrandt, and J. Döllner, 2009. Towards advanced and interactive web perspective view services, *Developments in 3D Geo-Information Sciences*, Berlin/Heidelberg, Springer, pp.33–51.

ISO, 2003, ISO(TC211), ISO 19107:2003: Geographic information—Spatial schema, International Organization for Standardization.

Koch, A., and C. Heipke, 2006. Semantically correct 2.5D GIS data - The integration of a DTM and topographic vector data, *ISPRS Journal of Photogrammetry and Remote Sensing*, 61(1):23–32.

Kolbe, T.H., G. Gröger, and L. Plümer, 2008. CityGML-3D city models and their potential for emergency response (S. Zlatanova and J. Li, editors), *Geospatial Information Technology for Emergency Response*, Taylor & Francis.

Kada, M., and L. McKinley, 2009. 3D building reconstruction from LIDAR based on a cell decomposition approach, *Proceedings of CMRT 2009*, International Archives of Photogrammetry, Remote Sensing and Spatial Information, Paris, France, pp. 47–52.

Ledoux, H., and M. Meijers, 2011. Topologically consistent 3D city models obtained by Extrusion, *International Journal of Geographical Information Science*, 25:557–574.

OGC, 2007. *OpenGIS® Geography Markup Language (GML) Encoding Standard*, version 3.2.1, Document no. OGC 07-036, URL: *http://portal.opengeospatial.org/files/?artifact_id=20509* (last date accessed: 13 November 2012).

OGC, 2008. *OpenGIS® City Geography Markup Language (CityGML) Encoding Standard*, version 1.0.0, Document no. 08-007rl, URL: *http://portal.opengeospatial.org/files/?artifact_id=28802* (last date accessed: 13 November 2012).

OGC, 2010. *OpenGIS® Web Feature Service (WFS) Implementation Specification*, version 2.0, URL: *www.opengeospatial.org/standards/wfs* (last date accessed: 13 November 2012).

OGC, 2012a. *OpenGIS® City Geography Markup Language (CityGML) Encoding Standard*, version 2.0, URL: *www.opengeospatial.org/standards/citygml* (last date accessed: 13 November 2012).

OGC, 2012b. CityGML *Application Domain Extensions*, URL: *http://www.ogcnetwork.net/gmlprofiles* (last date accessed: 13 November 2012).

Oude Elberink, S.J., and G. Vosselman, 2009a. 3D information extraction from laser point clouds covering complex road junctions, *The Photogrammetric Record*, 24(125):23–36.

Oude Elberink, S.J., and G. Vosselman, 2009b. Building reconstruction by target based graph matching on incomplete laser data: Analysis and limitations, *Sensors* 9 (8):6101–6118.

Oude Elberink, S.J., and G. Vosselman, 2011. Quality analysis on 3D building models reconstructed from airborne laser scanning data, *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(2):157–165.

Stoter, J.E., and P.J.M. van Oosterom, 2005. Technological aspects of a full 3D cadastral registration, *International Journal of Geographical Information Science*, 19(6):669–696.

Stoter, J.E., J. van Smaalen, R. Nijhuis, A. Dortland, J. Bulder, and B. Bruns, 2011. Fully automated generalisation of topographic data in current geo-information environments, *Urban and Regional Data Management (UDMS)* (S. Zlatanova, H. Ledoux, E.M. Fendel, and M. Rumor, editors),Taylor & Francis.

Stoter, J., G. Vosselman, J. Goos, S. Zlatanova, E. Verbree, R. Klooster, and M. Reuvers, 2011. Towards a national 3D spatial data infrastructure: Case of The Netherlands, *Photogrammetrie, Fernerkundung, Geoinformation,* 2011(6):405–420.

Van den Brink, L., J. Stoter, and S. Zlatanova, 2012. Establishing a national standard for 3D topographic data compliant to CityGML, *International Journal of Geographical Information Science*, DOI:1 0.1080/13658816.2012.667105, in press.

Yasumoto, S., A.P. Jones, T. Nakaya and K. Yano, 2011. The use of a virtual city model for assessing equity in access to views, *Computers, Environment and Urban Systems*, 35:464–473.