3D Visualization of Urban Area Using Lidar Technology and CityGML

Dragana Popovic¹, Miro Govedarica¹, Dusan Jovanovic¹, Aleksandra Radulovic¹, Vlado Simeunovic²
¹University of Novi Sad, Faculty of Technical Sciences, Serbia
²University of East Sarajevo, Faculty of Education, Bosnia and Herzegovina
draganapop@uns.ac.rs

Abstract. 3D models of urban areas have found use in modern world such as navigation, cartography, urban planning visualization, construction, tourism and even in new applications of mobile navigations. With the advancement of technology there are much better solutions for mapping earth's surface and spatial objects. 3D city model enables exploration, analysis, management tasks and presentation of a city. Urban areas consist of terrain surfaces, buildings, vegetation and other parts of city infrastructure such as city furniture. Nowadays there are a lot of different methods for collecting, processing and publishing 3D models of area of interest. LIDAR technology is one of the most effective methods for collecting data due the large amount data that can be obtained with high density and geometrical accuracy. CityGML is open standard data model for storing alphanumeric and geometry attributes of city. There are 5 levels of display (LoD0, LoD1, LoD2, LoD3, LoD4). In this study, main aim is to represent part of urban area of Novi Sad using LIDAR technology, for data collecting, and different methods for extraction of information's using CityGML as a standard for 3D representation. By using series of programs, it is possible to process collected data, transform it to CityGML and store it in spatial database. Final product is CityGML 3D model which can display textures and colours in order to give a better insight of the cities. This paper shows results of the first three levels of display. They consist of digital terrain model and buildings with differentiated rooftops and differentiated boundary surfaces. Complete model gives us a realistic view of 3D objects.

1. Introduction
With the advancement of today's technology there is need to improve current methods in order to enable fast and efficient management and distribution of geospatial information, and greater availability of spatial data. Currently majority of topics in area of geoinformatics is dealing with 3D cadastre and how to represent 3D objects. 2D maps are outdated and do not provide sufficient information that is now more than necessary. 3D cities are becoming more and more important in different areas of the modern world.

There are several data models for the creation of 3D city models including GML, KML, X3D, CityGML, VRML, STL etc. The CityGML defines five levels of details for multi-scale modelling: LOD0 – Regional model contains 2.5D Digital Terrain Model, LOD1 – Building block model without roof structures, LOD2 – Building model including roof structures, LOD3 – Building model including detailed architecture, LOD4 – Building model including interior model. Figure 1 contains LOD representations.
The main aim of this paper is to generate realistic 3D city models in LOD2 as it is defined in the official OGC standard CityGML. At this LOD, buildings have distinctive roof structures and flat facades. LIDAR technology was used for collecting data, several different programs for processing data and final product was 3D model of Petrovaradin Fortress in CityGML.

2. Background
Big data has its advantages and disadvantages. Some of most frequent problems are long time processing, inability to process certain types of problem, lack of automation processes. There are lots of variations of different methods of how to collect data, how to process data and how to publish received information. Main goal is to accomplish real 3D model of city. There are more case studies that have similar approaches to this work.

One of those approaches is Malambo who used fusion of various data such as LiDAR (Light Detection and Ranging) data and aerial imagery to derive the necessary 3D information for CityGML creation. 3D model was extracted by using LandXplorer Studio Professional [1]. Aringer used LiDAR, image matching and cadastre information in order to extract 3D Building Model with standardized roofs and textures. This case study also gave solution to how to solve problem with gaps between buildings [2]. Another variation on this topic is approach of H. Arefi where the automatic approach for reconstructing models in three levels of details is proposed, based on LIDAR data. The modelling approach deals with buildings which are formed by combination of flat roof, gabled roof as well as hipped roof segments [3]. Preka used an approach similar to the approach of this paper, difference is only data type of collected data [4].

3. Study area and data
Study area consists of location that represents a part of Novi Sad Petrovaradin Fortress, Serbia. Today's Petrovaradin Fortress is one of the best-preserved castles in Europe. Its area of 120 hectares is divided into three levels: the upper fortress, bastion Dvorožni or Hornwerk, and Lower fort. The complex located at the highest altitude in the Novi Sad, and it is of great tourist importance in Serbia. Two datasets were used for this project, LIDAR data and orthophoto with oblique images. LIDAR system is a complex, multi-sensor system consisting of at least three sensors, the GPS and INS navigation sensors, and the laser scanner system [5]. Figure 3 represents the example of collected data, point cloud (left) and obliques (right).
LiDAR dataset was given in Universal Transverse Mercator (UTM) WGS 84 / UTM zone 34N - horizontal coordinate system and in Ellipsoid WGS 84, vertical coordinate system.

4. Methodology

4.1. Point Classification

Point classification is done in Microstation V8 and modules TerraScan, TerraPhoto and TerraModel. TerraScan is the main application in the Terrasolid Software family for managing and processing LiDAR point clouds. It offers import and project structuring tools for handling the large amount of points of a laser scanning campaign as well as the corresponding trajectory information. Point cloud management, processing and visualization is only one part of TerraScan. In addition, the software provides tools for creating 3D vector data based on the laser points. TerraModeler creates surface models (TINs) from various sources, such as LiDAR points stored in binary files or loaded in TerraScan, XYZ ascii files and graphical design elements. TerraPhoto is specifically developed for processing images captured together with laser data during a survey mission. The software enables the production of rectified images and ortho mosaics based on ground model that has been extracted from the laser data [6].

Classification methods can be automatic, semi-automatic and manual [7]. It is possible to apply each method separately over some data, but for best results all of these methods are commonly used in combination with one another. After the data pre-processing, automatic classification was applied with thirty-two classes and appropriate attributes for this project. For the purposes of automatic classification macro files were created with parameters which define density of points and altitude. Automatic classification resulted with the points being classified but only for 3 classes: vegetation, buildings and ground. The reason why semi-automatic classification was used is that automatic classification could not extract classes. In this part of the classification, it was necessary to manually define the area of interest and change classified point value from one to another class. Complete classification includes 32 classes with appropriate 2D/3D layers (Figure 4).

4.2. 2D and 3D vectorization

Vectorization converts point cloud, raster or some other formats to vector data type with certain geometry type such as lines, points, polygons, curves, etc. Compared to the input data, output data provide higher order of control and more attributes which are used in geographic information systems (GIS). For the purposes of this project, 32 different vector layers were used.

Figure 3. LiDAR data and obliques
The entire complex of buildings can be seen as a collection of objects. As already mentioned in the chapter that describes the functions of TerraScan module, automatic vectorization is used to obtain the model multiple objects at the same time. In order to obtain better results, as observed in several attempts, a part of the building is vectorised separately from the rest of the buildings. Buildings can get irregular shape if vectorization parameters are not good.

Results of vectorization are given in Figure 5. Buildings are represented with black colour, parcels with green; grey colour was used for the roads, red colour for traffic area.

![Figure 4. Classified point cloud](image1)

![Figure 5. Vectorised elements of Petrovaradin Fortress](image2)
4.3. 3DCityDatabase

3D City Database is a free geo database to store, represent, and manage virtual 3D city models on top of a standard spatial relational database. The database schema implements the CityGML standard with semantically rich and multi-scale urban objects facilitating complex analysis tasks, far beyond visualization. The 3D City Database is implemented as a relational database schema using the spatial datatypes provided by a spatially-enhanced relational database management system [8].

Figure 6. 3DCityDatabase Schemas

Database was implemented in Postgis. PostGIS is an extension to the PostgreSQL object-relational database system which allows GIS (Geographic Information Systems) objects to be stored in the database. This database was used to store converted CityGML data [9]. Figure 6 and figure 7 contain basics 3DCityDatabases concepts.

Table 1. FME Transformations.

| Feature               | Used transformations                                                                 |
|-----------------------|---------------------------------------------------------------------------------------|
| Building, Roof Surface| FeatureMerger, 3DAffiner, Orientor, FaceReplacer, Aggregator, GeometryCoercer, AttributeCreator, GeometryPropertySetter |
| Plant cover           | 3DAffiner, Orientator, FaceReplacer, UUIDGenerator, AttributeCreator, GeometryPropertySetter |
| Water Body            | 3DAffiner, GeometryValidator, UUIDGenerator, AttributeCreator, GeometryPropertySetter |
| Traffic Area          | 3DAffiner, Orientor, FaceReplacer, Aggregator, UUIDGenerator, AttributeCreator, GeometryPropertySetter |
| Road                  | 3DAffiner, Orientor, FaceReplacer, Aggregator, UUIDGenerator, AttributeCreator, GeometryPropertySetter |
| City Furniture        | 3DAffiner, AzimuthCalculator, VertexExtractor, AttributeFilter, FeatureMerger, SharedItemAdder, SharedItemIDSetter, Rotator |
4.4. Conversion to CityGML

After extraction of layers and 2D/3D structure it is necessary to convert data to new format. By utilization of FME (Feature Manipulation Engine) data was transformed and stored in 3DCityGML Database. FME is a platform that streamlines the translation of spatial data between geometric and digital formats. It is intended especially for use with geographic information system, computer-aided design (CAD) and raster graphic software [10]. Table 1 contains used transformations for each feature.

Layers that were able to transform and insert into created database were Building, Roof Surface, Water Body, Road, Traffic Area, and City Furniture. In order to transform listed layers there were used some of the transformers which are embedded in software.

- 3DAffiner - performs 3D affine transformation on the coordinates of the feature. An affine transformation preserves parallelism of lines and planes in geometry.
- FeatureMerger - copies and merges the attributes/geometry from one feature (or multiple features) onto another feature (or multiple features).
- Orientor - adjusts the orientation of a polygonal feature or the direction of a linear feature.
- FaceReplacer - replaces the geometry of a feature from donut, raster or polygon to face.
- Aggregator - combines feature geometries into heterogeneous or homogeneous aggregates.
- GeometryCoercer - resets the geometry type of the feature.
- UUIDGenerator - calculates a Universally Unique IDentifier (UUID) for each incoming feature, and adds it as a new attribute.
- AttributeGenerator - adds one or more attributes to the feature and optionally assigns a value derived from constants, attribute values, and expressions.
- GeometryPropertySetter - sets selected geometry names or traits from feature attributes or constants.

Figure 8 represents a diagram which contains process from buildings.dgn to buildings.CityGML. Readers are defined with 4 layers which together make complete building data. Writers are defined with new gml data and a connection with the database with proper tables is set.
The entire workflow is shown in figure 2.

5. Results and discussion
This paper contains one of the approaches for reconstructing models in 2 levels of detail based on LiDAR data. In this section Petrovaradin Fortress is represented in LOD2 level (Figure 9). The obtained data is accurate up to 15 cm. The obtained model can be used for the digitization and preservation of certain objects of significance. The results were evaluated for the visual and metric qualities of the models. The overall task of 3D models, namely to reproduce objects in a realistic way, was achieved and it was therefore concluded that the visual accuracy was sufficient.

6. Conclusion
CityGML is applicable for large areas and small regions and can represent the terrain and 3D objects in different levels of detail simultaneously. This method is very applicable on big data, and also can be used in 3D Smart Cities. The 3D spatial data are useful in a wide range of applications related to urban environment visualization and representation. The Cadastre, the urban planning and commercial
activity, energy planning, disaster/crisis management, environmental management, navigation and traffic management as well as virtual tourism are fields that can benefit from 3D technology advancements.

The availability of spatial data is increasing steadily as more and more municipalities decide to create virtual 3D city models. The process of generation of LOD 3 models of buildings is still in progress, further research will investigate the formalization of LOD 3, LOD 4 model generation processes.

References
[1] Malamboa, L., and M. Hahnb. "Lidar assisted citygml creation." AGSE 2010 (2010): 13.
[2] Arlinger, K., and R. Roschlaub. "Calculation and update of a 3D building model of Bavaria using LiDAR, image matching and cadastre information." Proceedings of the 8th international 3D GeoInfo conference, Istanbul, Turkey. 2013.
[3] Arefi, H. "Levels of detail in 3d building reconstruction from lidar data." (2008).
[4] Preka, Dimitra, and Anastasios Doulamis. "3D building modelling in LOD2 using the CityGml standard." International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences 42 (2016).
[5] May, N. C., & Toth, C. K. (2007). Point positioning accuracy of airborne LiDAR systems: A rigorous analysis.
[6] https://www.terrasolid.com/home.php
[7] Popovic, Pajic, Jovanovic, Sabo, Radovic "Semi-automatic classification of power lines by using Airborne LiDAR" FIG Working Week (2017)
[8] http://www.3dcitydb.org/3dcitydb/3dcitydbhomepage/
[9] http://www.postgis.net/
[10] https://www.safe.com/
Erratum: 3D Visualization of Urban Area Using Lidar Technology and CityGML

IOP Conf. Ser.: Earth Environ. Sci. 2017 95 042006

Dragana Popovic¹, Miro Govedarica¹, Dusan Jovanovic¹, Aleksandra Radulovic¹ and Vlado Simeunovic²
¹ University of Novi Sad, Faculty of Technical Sciences, Serbia
² University of East Sarajevo, Faculty of Education, Bosnia and Herzegovina

The author has acknowledged that Figure 1 has not been fully attributed to the original author.

The figure in question can be found in:

Biljecki, F., Ledoux, H., & Stoter, J. (2016). An improved LOD specification for 3D building models. Computers, Environment and Urban Systems, 59, 25-37.
https://www.sciencedirect.com/science/article/pii/S0198971516300436?via%3Dihub