Advanced CityGML Visualization in 3D Environment

Mateusz Ilba*

* AGH University of Science and Technology in Krakow, Poland

Extended Abstract

The CityGML (Kolbe et al. 2005) is uniform standard for the exchange of three-dimensional data about urban space. The data can be used to create advanced visualization for various purposes. Unfortunately, the standard CityGML is not a popular source of input data for visualization software. Currently, are available open source solutions dedicated to the CityGML data, that can display geometry and texture with simple shading (Aristoteles, FZKViewer, tridicon(R) CityDiscoverer light (Rothe & Janne 2009). Some of the available applications allow better visualization of data, generating shadows (LandXplorer CityGML Viewer (Döllner 2006), CityGML SpiderViewer). Available solutions are enough to review the data, but if we want to present CityGML data in better image quality we need to use CAD programs, that use algorithms allowing advanced visualization of 3D data. There are also gaps in the capabilities of CityGML presentation in the form of symbolization based on available object attributes.

Lack of high-quality visualization and data symbolization led the author to verify the CityGML visualization capabilities in an advanced open source application named Blender. The application has the ability to easily add-ons programming in Python language. Software also allows you to perform advanced 3D visualization in the form of rendering. The main effects of increasing the quality visualization by the program Blender is drop shadows, ambient occlusion, reflection and refraction of light and environmental effects (clouds, fog, air pollution).

The author decided to create an original plugin to support CityGML data in the Blender application. Data, which was visualised, were loaded from the CityGML database, created in the PostGIS environment using available tools 3DCityDB Importer Exporter (Kolbe et al. 2013). Through a psycopg2 library (Hajji 2008) offering communication with the PostGIS database in python, was possible to reference to individual tables with the Blender

Published in "Proceedings of the 1st ICA European Symposium on Cartography", edited by Georg Gartner and Haosheng Huang, EuroCarto 2015, 10-12 November 2015, Vienna, Austria

application environment. Piece of code responsible for importing data (*name and building_root_id* attribute) from table building in PostGIS database:

```
import psycopg2
import bpy
try:
   conn = psycopg2.connect("dbname='Citygml_wizualizacja' user='postgres'
host='localhost' password='password'")
except:
   print ("error")
cur = conn.cursor()
cur.execute("SELECT name, building_root_id FROM building")
tabela_budynki=cur.fetchall()
cur.close()
conn.close()
```

Performing SQL queries to the respective tables is possible to generate objects semantic, their corresponding names and their hierarchy. Using the geometry fields in the table surface_geometry is possible to generate the object mesh. An exemplary flowchart to generate data for buildings is shown in the *Figure 1* below.

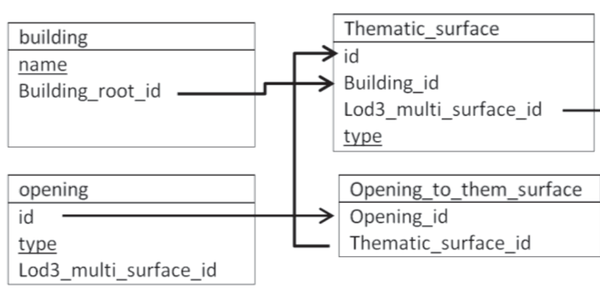

Figure 1. Generating scheme for the building data. Underlined attributes are used to define the name of 3D objects in Blender software.

Imported data, through enhanced visualization tool, may be presented in many different ways. A simple visualization of data in real time is done using OpenGL API (Rost et al. 2009). Additionally, we have access to significantly expanded data rendering options. Rendering allows for advanced three-dimensional data processing, adding to objects advanced textures, analysis of reflections, shading and light scattering, antialiasing the edges of objects, add effects in the form of fog, air pollution. The result of rendering is high quality image showing 3D data. In the figures below (*Figure 2 and 3*), we can see a sample CityGML database visualization.

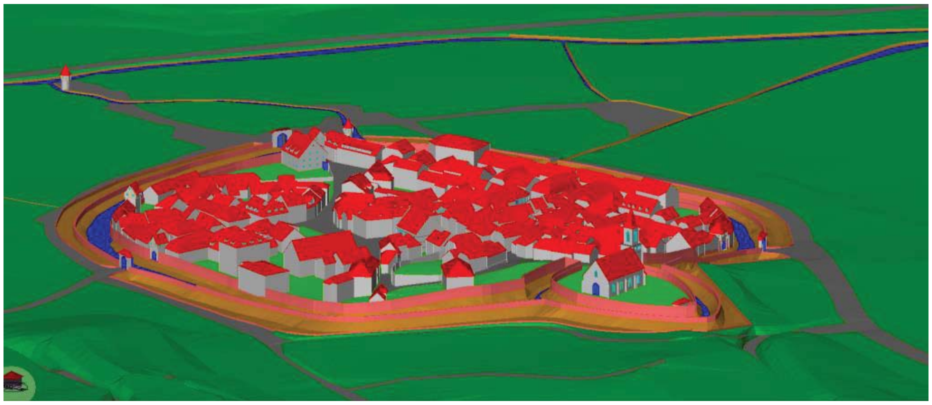

Figure 2. Visualization virtual 3D city model of Ettenheim in Germany. Visualization performed with the FZKViewer program. Data derived from (Research Center Karlsruhe).

In conclusion, the visualization of three-dimensional data is an important process which allows to see the spatial data. Commercially available visualization tools for CityGML data are simple, the effect of visualization is correct, but the quality is not satisfactory. In order to improve the quality of the resulting image, should be carried out data rendering, using specialized applications. High quality images allow a better understanding of three-dimensional data, shadows allow better assess the depth of the image and locate components in 3D space.

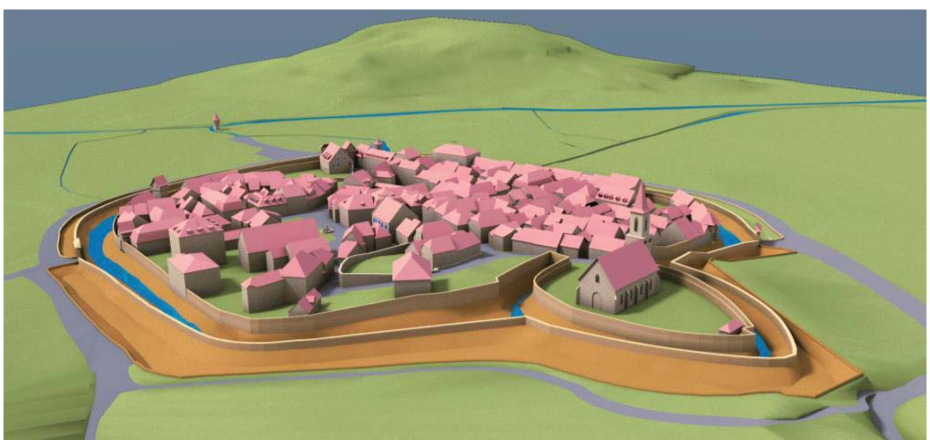

Figure 3. Visualization virtual 3D city model of Ettenheim in Germany. Visualization performed with the Blender and author scripts. Data derived from (Research Center Karlsruhe)

References

Döllner, J. (2006). LandXplorer–ein Werkzeug für komplexe Geoinformationen auf Grundlage virtueller 3D-Stadt-und 3D-Landschaftsmodelle. Vermessung Brandenburg, (1), 32-39.

Hajji, F. (2008). Das Python-Praxisbuch: Der große Profi-Leitfaden für Programmierer. Pearson Deutschland GmbH, 540-566.

Kolbe, T. H., Gröger, G., & Plümer, L. (2005). CityGML: Interoperable access to 3D city models. In Geo-information for disaster management (pp. 883-899). Springer Berlin Heidelberg.

Kolbe, T. H., Nagel, C., & Herreruela, J. (2013). 3D City Database for CityGML. Addendum to the 3D City Database Documentation Version, 2(1).

Research Center Karlsruhe, Institute for Applied Computer Science http://www.citygml.org/fileadmin/count.php?f=fileadmin%2Fcitygml%2Fdocs%2FStadt-Ettenheim-LoD3_edited_v1.0.0.zip
Accessed 5 August 2015

Rost, R. J., Licea-Kane, B. M., Ginsburg, D., Kessenich, J. M., Lichtenbelt, B., Malan, H., & Weiblen, M. (2009). OpenGL shading language. Pearson Education.

Rothe, D. I. R., & Janne, D. I. F. C. (2009) Vergleich des tridicon™ CityDiscoverer und Autodesk® LandXplorer zur Nutzung von 3D Stadtmodellen.