GIS Odyssey Journal

2024 • Volume 4 • Number 1

https://doi.org/10.57599/gisoj.2024.4.1.107

Piotr Cichociński[1]

A PROPOSAL OF A 3D SPATIAL DATABASE TO SUPPORT SUSTAINABLE LAND USE ANALYSES

Abstract: Despite the continuous development of geographic information systems (GIS) technology, most spatial datasets used in Poland for spatial planning purposes are still two-dimensional, like the traditional paper maps used previously. This approach makes it difficult, if not impossible, to conduct complex analyses that require three-dimensional (3D) data, for instance the identification of land use elements that could potentially have a positive or negative impact on the perception of an area as developing sustainably.
Meanwhile, since 2018, the Polish Head Office of Geodesy and Cartography provides 3D models of buildings, which are a 3D representation of a significant proportion of buildings from the BDOT10k database of topographic objects. Since 2023, this resource has been successively enriched with 3D models of trees over 4 m in height.
Therefore, an attempt has been made to investigate whether the available tools (especially those belonging to the free and open source software group) allow the creation of a spatial database to collect and share 3D data on at least buildings and trees. Assuming that the proposed solution is based on a relational database management system (RDBMS), it should allow for adequate efficiency in processing of large data sets. Thanks to built-in mechanisms, typical for spatial databases, it should also be possible to perform advanced spatial analyses in this environment. The applied system should allow data export in common formats and their visualisation. The studies carried out indicate that there are appropriate tools which, if put together in the right way, will make it possible, in particular, to analyse the manifestations of sustainable land use.

Keywords: 3D models, CityGML, geo-visualisation, spatial analyses, sustainable development

Received: 3 June 2024; accepted: 2 August 2024

[1] AGH University of Krakow, Faculty of Geo-Data Science, Geodesy, and Environmental Engineering, Department of Photogrammetry, Remote Sensing, and Spatial Engineering, Krakow, Poland, ORCID ID: http://orcid.org/0000-0002-8633-1235, email: Piotr.Cichocinski@agh.edu.pl

Introduction

Three-dimensional (3D) city models are becoming increasingly popular for larger cities. Already in 2012 there were more than a thousand 3D city models worldwide (Morton et al., 2012). They are being adopted more widely across various locations globally, going far beyond just serving as a visual representation. Instead, these 3D city models are now used to facilitate numerous tasks and functions that benefit society. For instance, research conducted by Biljecki et al. (2015) showed that 3D city models were being utilised in at least 29 distinct use cases, which collectively represented over 100 different applications. More recent projects were described by, among others, Biljecki et al. (2016), Eller et al. (2022), Gholami et al. (2022), Harter et al. (2023), Johansson et al. (2016), Katal et al. (2022), Rodríguez et al. (2017), Virtanen et al. (2021), and Willenborg et al. (2018).

To facilitate various simulations and analyses, it's crucial that 3D city models adhere to common standards. City Geography Markup Language (CityGML) is a widely used open standard for representing objects and structures in a city model (Gröger et al., 2012), providing a framework for organizing and describing the most important urban elements, such as buildings, roads, and terrain features, based on their physical properties (shape, size), spatial relationships, meaning, and visual appearance. This includes hierarchical structures of related objects, connections between them, and descriptions of their spatial positions. CityGML also allows for varying levels of detail (LOD). By going beyond simple graphical formats, CityGML enables the creation of detailed 3D city models that can be used for advanced analyses and decision-making.

Despite the growing popularity of CityGML-based city models, poor interoperability between CityGML data and software was reported by Noardo et al. (2020). They conducted an extensive and objective study to assess the software support for CityGML. The researchers invited a diverse group of practitioners with varying skills and expertise in CityGML and software usage. A range of software was tested, including specialized tools for handling 3D data and traditional geographic information systems (GIS) data. Only few applications were able to read and visualize the datasets without issues. Even fewer tools could export the data without losing information. Several factors were identified as contributing to these results, including the complexity of CityGML's geometrical and thematic data, as well as the variety of ways geometries can be represented.

One might get the impression that the vast majority of the efforts concerning CityGML have been spent on developing the concepts and the data model, and it appears that very little attention has been paid to deriving a usable exchange format. Indeed, the eXtensible Markup Language (XML) encoding is verbose, hierarchical, complex, and not adapted for the web (Bydłosz et al., 2010). These drawbacks hinder the use of CityGML in practice, which can be observed by the low number of software packages supporting full read/write/edit capabilities for CityGML files; and the relatively low number of datasets stored in CityGML files. CityGML files are notoriously known to be very difficult to parse and to extract information from. This has to do with the fact that XML itself requires special libraries to handle the data, that GML has several different ways to store the same geometry, and that CityGML files have deep hierarchies with multiple XLinks (a method for creating internal and external links within XML documents), which can be problematic for DBMS implementations, which tend to be "flat". These challenges make it difficult to develop software that can efficiently read, visualize, analyse, and transform CityGML data (Ledoux et al., 2019).

Somewhat in spite of these pessimistic conclusions, in this paper an attempt is made to investigate whether the available tools (especially those belonging to the free and open source software group) allow the creation of a spatial database to collect and share 3D data on at least buildings and trees. The author puts forward the following thesis: There are software tools which, if properly assembled, make it possible to analyse, in particular, the manifestations of sustainable land use. They should allow for adequate efficiency in processing of large data sets. It should also be possible to perform advanced spatial analyses in such environment. The applied system should allow data export in common formats and their visualisation.

Analysis of the state of the problems

The need for such a database stems from the availability of data that could potentially be used to conduct this type of analyses. From 2018, the Polish Head Office of Geodesy and Cartography provides 3D building models, which are a three-dimensional representation of a significant proportion of buildings from the BDOT10k database of topographic objects. Data at the LOD2 level of detail are available for 236 counties, covering the area of 10 voivodeships, while data at the LOD1 level of detail are provided for the entire country. Since 2023, this resource has been successively enriched with 3D models of trees over 4 m in height. All models are available free of charge and can be used freely, but so far have not found wider adoption. One of the reasons may be the inconvenient and inefficient way of accessing this data. Although they are presented in the 3D Geoportal, its functionality is reduced to viewing these data and elementary analyses such as shading visualisation. To make wider use of these data, they have to be downloaded via the map portal in a tedious manual process. Using the identification tool, users click on the map within the boundaries of the county (poviat) for which they wish to obtain data. This brings up a list of available datasets, which have to be downloaded individually. Data packages obtained this way are further subdivided into files corresponding to topographic map sheets at a scale of 1:5000. Users are advised to employ the free QGIS software (QGIS.org, 2024) to view and process the data. But it should also be borne in mind that at the LOD2 level of detail, these datasets can already be classified as Big Data.

Loading and saving datasets in a wide variety of formats is implemented in QGIS through the GDAL/OGR Geospatial Data Abstraction software Library (GDAL/OGR contributors, 2024). It is released by the Open Source Geospatial Foundation under an MIT-style open source license, making it a versatile translator library for both raster and vector geospatial data formats. This library provides a unified abstract data model for raster data and another for vector data, allowing applications to access all supported formats seamlessly (internally GDAL part is responsible for raster data, while OGR provides access to simple features vector data). Furthermore, it includes a range of command-line utilities that facilitate data translation and processing tasks.

Importing files in Geography Markup Language (GML) format is possible in cases where the structures of the data to be loaded are not too complex and can be translated into a simple feature model. The OGR GML reader used for this purpose does not rely on having an XML Schema definition (XSD, usually stored in file with .xsd extension) to read a GML file. If the schema file is missing or cannot be parsed, the driver takes a different approach: it attempts to automatically identify feature classes and their properties by scanning the GML file for known "gml" objects in the namespace. While this method can be error-prone, it has the benefit of allowing OGR to read GML files even if the associated schema file is lost or unavailable.

Material and methods

However, data structures stored in GML format may be more complex. Fortunately, in addition to the simple import option mentioned above, the GDAL library also offers a solution called GMLAS. It allows for reading and writing XML files of any structure, including those with complex features, as long as they are accompanied by an XML schema (XSD) that outlines their content. Additionally, it can efficiently process large files without consuming excessive RAM due to its streaming mode operation.

OGR GMLAS driver analyses XSDs to extract the underlying object model and identify the type of each element and attribute, and relationships (links) between elements. On this basis it creates the database model corresponding to the schema (in particular converting links between elements into relations between database tables). It results in multiple tables and relationships between them. To fully integrate this relational data within GIS software, users have to load all tables (including those without geometry) and establish connections between them. This requires adequate geoinformatics knowledge to understand this complex data structure and a good knowledge of the software used. In particular within QGIS, the concept of "relations" enables users to declare 1:N relationships between vector layers. These relations can then be leveraged in layer form views for navigating the model.

This task can be facilitated by the "QGIS GML Application Schema Toolbox" plug-in. Its purpose is to manipulate Complex Feature streams through a user-friendly graphical user interface (GUI). The plugin offers various capabilities, including downloading GML data from Web Feature Service (WFS) sources, reading GML files in native XML format, converting these files to PostGIS and SpatiaLite spatial databases, and exporting the converted databases back to GML files. It automatically generates a QGIS project with all the layers loaded and all the known relations declared.

The advantage of this approach (whether with or without the help of a plug-in) is that the loaded data can be stored in a relational database. Relational databases are more popular than other types of databases. They offer a range of features such as support for different spatial data types, query languages, indexing structures, and capabilities for performing geometrical and topological analyses. Furthermore, they can integrate the geometric modeling of both man-made and natural geographic objects, making them well-suited for applications that require spatial analyses and visualisation. Relational databases provide several key benefits due to their ability to handle transactions in a robust manner, manage concurrent access to data, and optimize queries effectively.

This was confirmed by, among others, Schmid (2021), who proposed a solution that transforms the 3D data into a property graph and stores this graph in the database system. He successfully imported datasets as large as nearly 1 TB without issues, and even larger 3D models (up to 300 MB) within just five minutes. Furthermore, his queries achieved response times of under 5 milliseconds in an impressive 99% of executions.

In contrast, the relational database (PostGIS) is only partially used by the deegree3D software, which also offers file systems storage option. This combination is intended to allow flexible object modeling and fast data access. It adheres to industry standards and offers data management, access, and portrayal capabilities for 3D models of buildings, cities, terrains, and their textures. Existing 3D models can be imported from and exported as CityGML files.

Additional data access options are provided by Open Geospatial Consortium (OGC) Web Services allowing for seamless integration with spatial data infrastructures. In particular deegree3D comprises several key components that enable geospatial data processing and visualisation:

- deegree featureService provides an OGC Web Feature Service (WFS) interface to interact with the CityGML database, including transactions (insert/update/delete);
- deegree coverageService offers an OGC Web Coverage Service (WCS) interface for accessing raster-based terrain models;
- deegree mapService provides an OGC Web Mapping Service (WMS) interface for generating maps and retrieving map object information, specifically used within deegree3D for georeferenced terrain model textures;
- deegree terrainService offers a lightweight web-based interface for rendering 3D still images presenting perspective views (based on combinations of building, city, and terrain models, along with their textures) and an interactive desktop viewer for “fly-through” experiences. The web-based interface facilitates easy integration of 3D perspective views into web pages and geospatial portal applications.

Unfortunately, the project appears to have been abandoned. The last mentions of it that can be found online are from 2018.

Results

Meanwhile, another worthwhile project is still going well: the 3D City Database (3DCityDB) – a free, open-source package that includes a database schema and software tools for importing, managing, analysing, visualizing, and exporting virtual 3D city models based on the CityGML standard (Yao et al., 2018). Being open-source, the entire software is available to anyone interested, and licensed under Apache License Version 2.0, which permits integration into commercial projects. Its stated goal is to provide a high-performance, scalable data storage solution for 3D digital city models, also known as Digital Twins. The database schema is derived from mapping the object-oriented data model of CityGML to the relational structure of a spatially-enhanced relational database management system (SRDBMS), leveraging built-in spatial types. The 3DCityDB supports both open-source and commercial SRDBMS solutions, including PostgreSQL with PostGIS extension and Oracle with Spatial capabilities. By utilizing the specific representation and processing capabilities of the SRDBMS for spatial data elements, the 3DCityDB is expected to efficiently handle large models with millions of 3D objects, hundreds of millions of geometries, and texture images. This approach avoids custom solutions and allows various geospatial systems to directly read and write geometry objects stored in the database.

The 3DCityDB allows for interoperable data access in at least three ways:

1. By using the included CityGML Importer/Exporter to exchange data in CityGML format. This tool is designed to handle large files (> 4 GB) with ease. It can process complex XML structures efficiently thanks to multithreaded programming. This allows it to take advantage of multiprocessor systems or multicore CPUs for faster processing. A tiling strategy is implemented to efficiently export massive datasets by distributing texture images into multiple subdirectories, preventing overwhelming directories with millions of files that could crash Microsoft Windows operating systems. Additionally, the Importer checks CityGML files for errors and can be set up to only import valid features. The tool also offers advanced filtering options during import/export, including spatial regions (bounding box), object IDs, feature types, names, and levels of detail. Additionally, users can interactively select bounding boxes on a map using OpenStreetMap (OSM) as a background. Such operations are possible thanks to graphical user interface (GUI), designed for easy use on desktop computers by end-users, while the command-line interface (CLI) can be used in headless systems or embedded in batch processing workflows and third-party applications to automate tasks.
2. Through a WFS interface that permits web-based access to stored 3D city objects. The OGC Web Feature Service Interface Standard (WFS) provides a standardized way for requesting geographic features across the web using platform-agnostic calls (Vretanos, 2014). Instead of sharing data at the file level, WFS enables direct access to feature and property-level information. This allows clients to retrieve or modify specific data without downloading unnecessary files. The WFS is platform-independent and database-independent, making it easy to develop CityGML-aware applications.
3. By directly accessing the database tables, which enables users to enrich a 3D city model by adding information to the database tables within their own

applications. The enriched dataset can then be exported to CityGML without losing any information. Similarly, the 3DCityDB can be used to import a CityGML dataset and access and work with the city model by directly accessing the database tables from within GIS software. Additionally, PostGIS offers 3D spatial functions which can be used for (Chenaux, 2019): selecting and extracting data over one or several tables (spatial predicates, which define spatial relationships between objects, such as ST_3DIntersects, ST_3DDWithin, etc.); computing new geometries from existing ones (spatial methods, for example ST_3DBuffer, 3DClosestPoint, ST_3DLongestLine, ST_3DIntersection, etc.); taking measurements (ST_3DDistance, ST3DArea, etc.).

The 3DCityDB software bundle also includes a web-based front-end called the "3DCityDB-Web-Map-Client" (Chaturvedi et al., 2015), which is designed for high-performance, interactive visualisation and exploration of large semantic 3D city models over the Internet within web browsers on desktop and mobile devices. This client has been built upon CesiumJS (an open-source JavaScript library for creating 3D globes and 2D maps in a web browser without a plugin), adding features like caching, prefetching, and dynamic loading/unloading of these massive data sets. It leverages HTML5 and Web Graphics Library (WebGL) as its core technologies (Sellers et al., 2013), providing hardware acceleration and cross-platform functionality to display 3D graphics on the web without requiring additional plugins. The interface allows users to interact with the 3D models in an intuitive way through mouse over and mouse click highlighting, hiding/showing, or shadowing objects. Additionally, 3D models can be linked to online spreadsheets, enabling querying of thematic city data based on GMLID.

This way features of the 3DCityDB span from efficiently storing, managing, analysing, and sharing digital 3D city models in accordance with the CityGML format all the way to delivering high-performance visualisation and exploration on the web – making it perfect and complete software solution, that is able to support all required functionalities.

Discussion and conclusions

Analysing the resources of free software available on the Internet and their documentation, the author can conclude that despite the doubts expressed in the literature, there are software tools that make it possible to handle data stored in CityGML format. In particular, despite their complex structure, it is possible and beneficial to store them in a relational database, which is expected to perform well even with large amounts of data, regardless of size or complexity, allowing for adequate efficiency in processing of large data sets. Thanks to built-in mechanisms, typical for spatial databases, it should also be possible to perform advanced spatial analyses in this environment.

Given the simple structure of the shared CityGML files, it seems that an easier but still effective solution would be to import the data into a PostGIS database, using QGIS software as an intermediary or directly utilizing the GDAL tools. However, given the still rather limited capabilities of GIS software for 3D visualisation, the availability of a "3D web client" offered by 3dCityDB is tempting. Its additional advantage is that it functions in a web browser and thus relieves users of the need to own and be able to operate specific GIS software. The WFS service cannot be underestimated either, which should significantly improve data availability.

In the next stages of the work, it is envisaged that both options will be thoroughly tested in order to select a more suitable one. Firstly, this will involve the acquisition and import of existing and self-generated data (based mainly on remote sensing sources – images and point clouds from various ground-based and airborne sensors) representing 3D objects and complexes of such objects with different levels of development (e.g. highly urbanised, suburbanised, rural, undeveloped, revitalised, etc.) and complexity (simple, complex, with various colours and textures), on which planning and development activities leading to sustainable development have been or potentially will be carried out. In this way, the performance of a given software can be verified. Subsequently, the tools of the tested database should make it possible to identify and, in the next step, visualise the land-use elements that influence the perception of an area as being developed in a sustainable way. Finally, it is required that the images generated in this way are of sufficient quality to be shown to users of the analysed spaces as a part of eye-tracking surveys.

Acknowledgements

Research project supported by program "Excellence Initiative – Research University" for the AGH University of Krakow (Action 4, Application number: 6325).

References

Biljecki F., Arroyo Ohori K., Ledoux H., Peters R., Stoter J. (2016). Population estimation using a 3D city model: A multi-scale country-wide study in the Netherlands. PloS one, 11(6), e0156808. https://journals.plos.org/plosone/article/file?id=10.1371/journal.pone.0156808&type=printable [access: 31.05.2024].

Biljecki F., Stoter J., Ledoux H., Zlatanova S., Çöltekin A. (2015). Applications of 3D city models: State of the art review. ISPRS International Journal of Geo-Information, 4(4), 2842–2889. https://www.mdpi.com/2220-9964/4/4/2842/pdf [access: 31.05.2024].

Bydłosz J., Cichociński P., Basista I. (2010). The possibilities of geoinformation resources recorded in GML accessing with chosen GIS software. Geomatics and Environmental Engineering, 4(1), 33–44. https://bibliotekanauki.pl/articles/386112.pdf [access: 31.05.2024].

Chaturvedi K., Yao Z., Kolbe T.H. (2015). Web-based Exploration of and interaction with large and deeply structured semantic 3D city models using HTML5 and WebGL. In: Bridging Scales-Skalenübergreifende Nah-und Fernerkundungsmethoden, 35.

Wissenschaftlich-Technische Jahrestagung der DGPF. https://mediatum.ub.tum.de/doc/1245285/document.pdf [access: 31.05.2024].

Chenaux A., Murphy M., Pavia S., Fai S., Molnar T., Cahill J., ... & Corns A. (2019). A review of 3D GIS for use in creating virtual historic Dublin. https://arrow.tudublin.ie/cgi/viewcontent.cgi?article=1155&context=beschreccon [access: 31.05.2024].

Eller L., Svoboda P., Rupp M. (2022). A deep learning network planner: Propagation modeling using real-world measurements and a 3D city model. IEEE Access, 10, 122182-122196. https://ieeexplore.ieee.org/iel7/6287639/6514899/09954403.pdf [access: 31.05.2024].

Gholami M., Torreggiani D., Tassinari P., Barbaresi A. (2022). Developing a 3D city digital twin: Enhancing walkability through a green pedestrian network (GPN) in the City of Imola, Italy. Land, 11(11), 1917. https://www.mdpi.com/2073-445X/11/11/1917/pdf [access: 31.05.2024].

Gröger G., Kolbe T.H., Nagel C., Häfele K.H. (2012). OGC City Geography Markup Language (CityGML) Encoding Standard, Version 2.0. Open Geospatial Consortium, Doc. No. 12-019. https://portal.ogc.org/files/?artifact_id=47842 [access: 31.05.2024].

Harter H., Willenborg B., Lang W., Kolbe T. H. (2023). Life Cycle Assessment of building energy systems on neighbourhood level based on semantic 3D city models. Journal of Cleaner Production, 407, 137164. https://www.sciencedirect.com/science/article/pii/S0959652623013227 [access: 31.05.2024].

Johansson T., Segerstedt E., Olofsson T., Jakobsson M. (2016). Revealing social values by 3D city visualization in city transformations. Sustainability, 8(2), 195. https://www.mdpi.com/2071-1050/8/2/195/pdf S0959652623013227 [access: 31.05.2024].

Katal A., Mortezazadeh M., Wang L.L., Yu H. (2022). Urban building energy and microclimate modeling–From 3D city generation to dynamic simulations. Energy, 251, 123817. https://www.sciencedirect.com/science/article/pii/S0360544222007204 [access: 31.05.2024].

Ledoux H., Arroyo Ohori K., Kumar K., Dukai B., Labetski A., Vitalis S. (2019). CityJSON: A compact and easy-to-use encoding of the CityGML data model. Open Geospatial Data, Software and Standards, 4(1), 1–12. https://link.springer.com/content/pdf/10.1186/s40965-019-0064-0.pdf [access: 31.05.2024].

Morton P.J., Horne M., Dalton R.C., Thompson E.M. (2012). Virtual city models: Avoidance of obsolescence. In Education and Research in Computer Aided Architectural Design in Europe–30th eCAADe Conference (pp. 213–224). https://papers.cumincad.org/data/works/att/eCAADe_2012-vol-1-lowres.pdf#page=214 [access: 31.05.2024].

Noardo F., Arroyo Ohori K., Biljecki F., Ellul C., Harrie L., Krijnen T., ... & Stoter J. (2021). Reference study of CityGML software support: The GeoBIM benchmark 2019 – Part II. Transactions in GIS, 25(2), 842–868. https://onlinelibrary.wiley.com/doi/pdf/10.1111/tgis.12710 [access: 31.05.2024].

QGIS.org, (2024). QGIS Geographic Information System. QGIS Association. http://www.qgis.org [access: 31.05.2024].

Rodríguez L.R., Duminil E., Ramos J.S., Eicker U. (2017). Assessment of the photovoltaic potential at urban level based on 3D city models: A case study and new methodological approach. Solar Energy, 146, 264–275. https://core.ac.uk/download/pdf/288003205.pdf [access: 31.05.2024].

Schmid M. (2021). Towards Storing 3D Model Graphs in Relational Databases. Dissertation (unpublished). University of Passau. https://opus4.kobv.de/opus4-uni-passau/files/1035/Matthias_Schmid_Dissertation_print.pdf [access: 31.05.2024].

Sellers G., Obert J., Cozzi P., Ring K., Persson E., de Vahl J., van Waveren J. M.P. (2013). Rendering massive virtual worlds. In ACM SIGGRAPH 2013 Courses (pp. 1–88). https://dl.acm.org/doi/abs/10.1145/2504435.2504458 [access: 31.05.2024].

Virtanen J.P., Jaalama K., Puustinen T., Julin A., Hyyppä J., Hyyppä H. (2021). Near real-time semantic view analysis of 3D city models in web browser. ISPRS International Journal of Geo-Information, 10(3), 138. https://www.mdpi.com/2220-9964/10/3/138/pdf [access: 31.05.2024].

Vretanos P.A. (2014). OGC Web Feature Service 2.0 Interface Standard – With Corrigendum, Version 2.0.2. Open Geospatial Consortium, Doc. No. 09-025r2. https://docs.ogc.org/is/09-025r2/09-025r2.html [access: 31.05.2024].

Willenborg B., Sindram M., Kolbe T.H. (2018). Applications of 3D city models for a better understanding of the built environment. In: M. Behnisch, G. Meinel (ed.), Trends in Spatial Analysis and Modelling – Decision-Support and Planning Strategies, Springer Verlag. URL https://mediatum.ub.tum.de/doc/1348882/document.pdf [access: 31.05.2024].

Yao Z., Nagel C., Kunde F., Hudra G., Willkomm P., Donaubauer A., ... & Kolbe T.H. (2018). 3DCityDB-a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML. Open Geospatial Data, Software and Standards, 3(1), 1–26. https://link.springer.com/content/pdf/10.1186/s40965-018-0046-7.pdf [access: 31.05.2024].