2021 European Conference on Computing in Construction
Online eConference
July 26-28, 2021

CITYBIT: CITYGML BUILDING INTERPOLATION TOOL FOR ENERGY PERFORMANCE SIMULATIONS

Avichal Malhotra, Simon Raming, Jérôme Frisch and Christoph van Treeck
Institute of Energy Efficiency and Sustainable Building (e3D)
RWTH Aachen University, Germany

ABSTRACT

This paper describes a CityGML Building Interpolation tool (CityBIT) for the creation of user-defined and interpolated building models for planned and/or existing buildings. A brief explanation of the tool's functionalities and the methodologies adapted to develop the tool are explained in this paper. The CityBIT aims to help urban planners and simulation scientists to facilitate CityGML model developments for energy performance simulations.

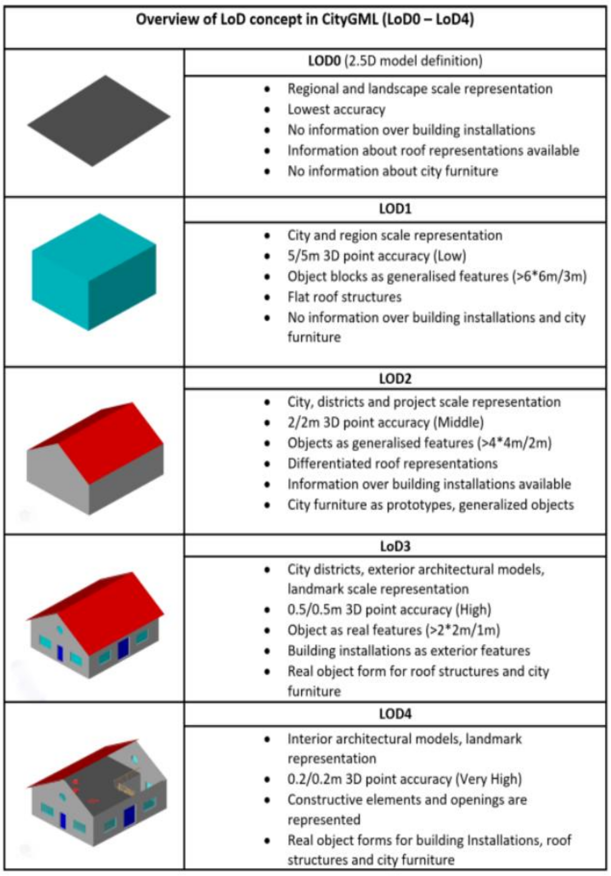
INTRODUCTION

Urban areas have proved to be one of the largest energy consumers and emitters of greenhouse gases in the world (Masanet, et al. 2016). However, these areas also offer a large potential for energy efficiency improvement and greenhouse gases mitigation (Sola, et al. 2018). Facilitating the flexibility in the domain of building simulation, CityBIT will enable urban planners, architects, energy companies and simulation scientists to create CityGML building models for urban scale energy simulations.

Energy Performance Simulations can be complex, intensive and require high computational resources and efforts. For the past 50 years, a wide variety of building energy simulation programs have been developed, enhanced and are in use throughout the building and urban energy community (Crawley, et al. 2008). Energy simulation tools are increasingly being used for analysis of energy performance of buildings or districts and for the thermal comfort of their occupants (Malle, et al. 2007). Moreover, simulation-based methods are being applied by the urban planners, architects as well as the building simulation research community. However, in order to compute the energetic performance of the buildings, 3D geometrical data models do serve as an important entity in the simulation domain (Tian, 2018). Therefore, keeping in consideration the geometrical aspects of buildings, the City Geographical Markup Language (CityGML) (Gröger, et al. 2012), an open XML based modeling language, is the basis for the development of the CityGML Building Interpolation Tool (CityBIT).

CityGML was developed to model virtual 3D buildings at an urban scale. Based on the amount of information present in a model, CityGML datasets are available in five Levels of Detail (LoD) (in Table 1).

Table 1: The five Levels of Detail (LoD) in CityGML. Information retrieved from (Malhotra, et al. 2019)



Overview of LoD concept in CityGML (LoD0 – LoD4)	
	LOD0 (2.5D model definition)
	• Regional and landscape scale representation • Lowest accuracy • No information over building installations • Information about roof representations available • No information about city furniture
	LOD1
	• City and region scale representation • 5/5m 3D point accuracy (Low) • Object blocks as generalised features (>6*6m/3m) • Flat roof structures • No information over building installations and city furniture
	LOD2
	• City, districts and project scale representation • 2/2m 3D point accuracy (Middle) • Objects as generalised features (>4*4m/2m) • Differentiated roof representations • Information over building installations available • City furniture as prototypes, generalized objects
	LoD3
	• City districts, exterior architectural models, landmark scale representation • 0.5/0.5m 3D point accuracy (High) • Object as real features (>2*2m/1m) • Building installations as exterior features • Real object form for roof structures and city furniture
	LOD4
	• Interior architectural models, landmark representation • 0.2/0.2m 3D point accuracy (Very High) • Constructive elements and openings are represented • Real object forms for building Installations, roof structures and city furniture

The 3D models can be used to store and exchange semantic and topological information of individual buildings at a city scale. The CityGML core is intended to be an application-independent geographical information model (Agugiaro, et al. 2018). Based on individual applications, however, CityGML data models can be extended using the Application Domain Extension (ADE) mechanism (Gröger, et al. 2012). For energy related applications, the Energy ADE (Agugiaro, et al. 2018) was developed for simulations as well as dissemination of the energy-enriched CityGML data models. Similar to the modular structure of the CityGML, the Energy ADE extends the building models with information related to building physics, occupancy, energy systems, construction materials, and time series data. Moreover, along with the application of extending CityGML data classes with energy related properties of the buildings, the Energy ADE also makes it possible to divide the buildings internally into multiple thermal zones.

The availability of 3D building models has been increasing in the last few years (Coors, et al. 2020). CityGML LoD1-2 datasets are also available for some countries, states and cities (Malhotra, et al. 2020). However, there is still a lack of open source information for many city districts which would be beneficial for the industrial and research community. In order to bridge the gap between the availability of data models and its application, the authors present the “CityBIT” in this research paper.

It is structured as follows: The “Methodology” section highlights the algorithms that are developed to create new CityGML models; the “Implementation” chapter gives a short overview of a test case that has been developed for demonstration along with user friendly GUI of the tool. This is followed by the “Conclusion” section.

METHODOLOGY

Virtual 3D data models can be generated by different methods such as aerial photography, airborne laser scanning or by using the digital cadastral data combined with building information (Nouvel, et al. 2015). Using these methods, CityGML data models for districts can be generated automatically. However, for the research community, some of these techniques are expensive and depending on the country might also require a lot of governmental approvals. Moreover, for the simulation community and architectural companies, it is also important to analyse the energetic performance of a district during its planning phases. This implies that an analysis should be carried out long before the buildings are actually built. Data enrichment environments such as TEASER (Remmen, et al. 2017) and SimStadt (Coors, et al. 2014) and general simulation environments such as EnergyPlus (NREL, 2019) and TRNSYS (Palme & Salvati, 2018) do facilitate energy simulation with the input functionality of CityGML data models. However, before a building is constructed, in many cases, the CityGML building models are not available. Therefore, it is important to bridge the gap between the modelling of the individual districts and their energy performance simulations by developing open methods to create building and district models.

Furthermore, for retrofitting scenarios, energy demands are calculated to achieve an efficient retrofit. The demand based simulations can be carried out using different simulation environments making the input building model to serve as an important entity. Detailed building models do also exist in formats such as IFC (IFC Web Server, 2018), Shape files (ESRI 1998), etc. Some tools/techniques such as FME (Safe Software, 2020) or CityGML Builder (Virtual City Systems 2019) do facilitate the transformation of one format to another, however, the transformation software, environments or system configurations are not openly available to the community, and, thereby, restricting the usage for many research and development-based studies.

Although, openly accessible city-wide 3D geometrical data is updated on a yearly basis, there exists discrepancies within the city models for each year. One such example can be found in Figure 1, where a city quarter in a German city is shown based on the data modeled in the years 2016 (top), 2018 (middle) and 2019 (bottom). As the dataset from 2017 was not available, it is not included in this analysis. Even though the datasets in Figure 1 look quite similar on a 2D visualization, the number of buildings and building parts do differ for each year.

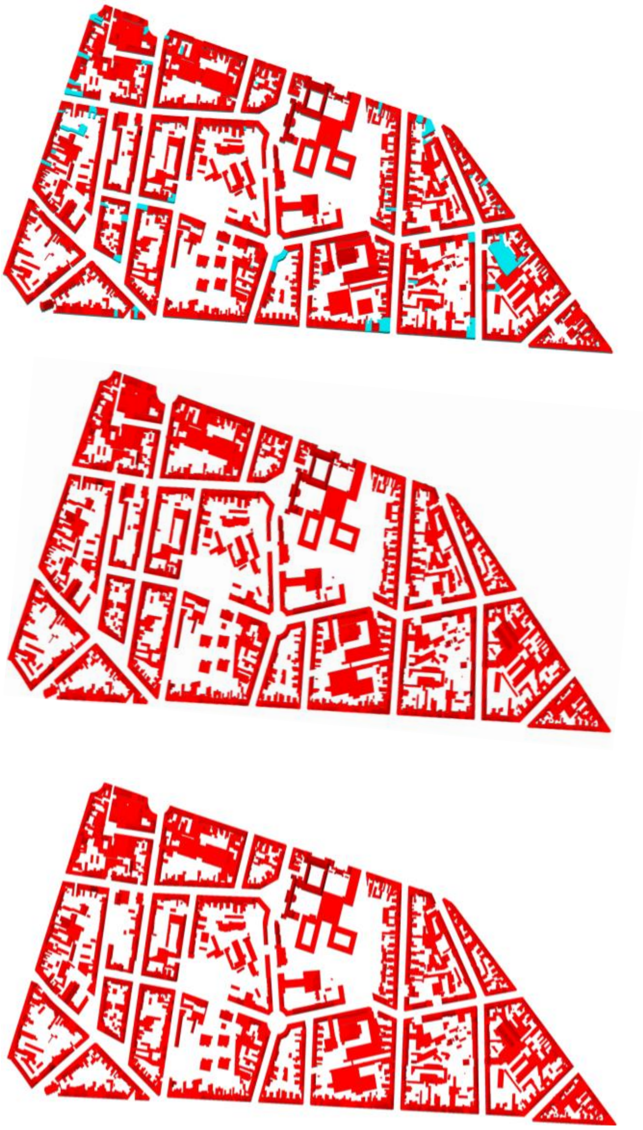


Figure 1: The urban area visualization for the years 2016 (Top), 2018 (Middle), and 2019 (Bottom). The visualization of the different CityGML datasets was performed using the FZK Viewer (IAI/KIT, 2019)

Analysing the previously considered dataset, the authors also examined the city quarter physically comparing to the building models in the CityGML dataset. Based on the investigations, the authors concluded that though some buildings were modeled incorrectly, some others did not even exist in the dataset. Due to data security policies in different countries, the exact number of previously mentioned incorrectly modeled buildings, their addresses and coordinates are not included in this paper.

Table 2 gives an overview of the analysis of CityGML models modeled in the year 2016, 2018 and 2019. For the visualization and analysis, individual datasets have been selected using the searching functionality of CityGML Analysis Toolbox (CityATB) (Malhotra, et al. 2020).

Table 2: Overview of data analyzed for a city quarter

Year of Modeling	2016	2018	2019
No. of Buildings	1489	1524	1521
No. of Building Parts	1190	1834	1825
CRS (EPSG)	25832	25832	25832
Level of Detail	1 and 2	2	2
CityGML version	1.0	1.0	1.0

The challenges in data acquisition and availability for urban scale simulations increase considering the building(s), which are currently in planning phases. As the datasets are not commonly available and/or updated, the flexibility of analysing the energetic performance of the planned district/building(s) decreases drastically. Therefore, using the CityBIT, the users will be able to create/model new CityGML datasets using two different techniques. The next subsection describes the required input data that the user needs to enter for model creation. This is followed by the two approaches considered by the authors for creating new building models.

Required Information for CityGML Model Creation

As the main focus of CityBIT is to facilitate the CityGML model creation, building models based on basic user inputs can be created using the tool. Some of the required user inputs are:

- **Information about the desired output coordinate(s)/surface**: The output coordinate(s) refers to the desired target point/surface for the building model that the user wants to create using the tool. The input functionality of the coordinates is similar to the approach taken by the developers of the CityATB (Malhotra, et al. 2020). The target point in this case is the centre of the ground surface polygon for the individual buildings. Whereas, the surface input refers to the ground surface polygon coordinates of the desired building.
- **Input and Output Coordinate Reference System (CRS)**: As CityGML datasets are geometrically and geographically modeled, the input CRS has to be entered by the user. The input CRS refers to the coordinate system in which the user would input the desired output coordinate(s). For example, if a user wants to input the latitude and longitude of the desired target point, using Google Maps (Google, 2020), the user needs to select the input CRS according to the EPSG code. In this case it would be 4326. Furthermore, the user also needs to select the desired output CRS within the tool. The user entered target coordinate(s) is/are then checked based on the selected output CRS. The output CRS refers to the coordinate system of the newly created building models.

Furthermore, once the user inputs the above mentioned required data into the tool, two approaches can be used for creating the model.

Model creation using basic user input

Generally, CityGML datasets do contain the information about building geometry, usage, roof type, roof height and more. For creating new geometrical models, CityBIT gives the user a flexibility to input the following parameters:

- **Building Height:** The building height is similar to the ‘measured height’ as defined in the Modeling Guide for 3D Objects (SIG3D, 2018). It refers to the difference in [m] between the highest point of the roof structure and the defined footprint of the building.
- **Ground Surface Height:** The surface height refers to the height of the ground slab of the building in [m] above sea level. In this case, the storeys below ground (if provided by the users) are modeled above the ground surface polygon.
- **Ground Surface Area:** As described previously, one of the essential user inputs for the tool is the information about the desired output coordinate(s)/surface. If the user inputs only a single coordinate for the building model, the building is considered to have a rectangular geometry and the entered point is considered to be the center point of the building. This point is then used to calculate the ground surface area. Moreover, with the additional flexibility to enter area [m^2], the user can input the desired area of the building ground surface polygon.
- **Side Ratio:** For rectangular building geometries, if only the desired output surface area is entered by the user, the side ratio is calculated for a ground surface length of the longer side of the building divided by the length of the shorter side. However, if precise coordinates are provided by the user, the side ratio can be calculated using the given coordinates.

- **Building Orientation:** The building orientation refers to the heading of the longer side of the building surface. If the desired surface coordinates are not entered by the user, the building orientation can be inputted by the users to have a precise orientation of the desired building.
- **Roof Height:** The roof height refers to the minimum eaves height. The eaves of a building are the point where the external walls, if projected upwards, meet the lowest point of the upper surface of the roof (Welsh Government, 2013). Figure 2 gives a visual representation of the building- and roof height.

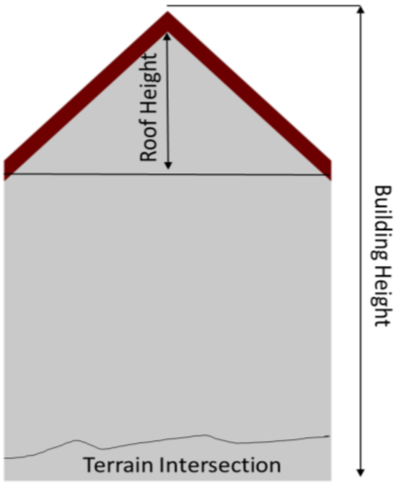


Figure 2: Visual representation of building height, roof height and terrain intersection

- **Roof Type:** Currently, within the tool, the user can select one out of six different roof types. These refer to a flat roof, a monopitch roof, a dual pent roof, a gabled roof, a hipped roof and a pavilion roof (as shown in Figure 3). In future, the addition of more roof types is planned.

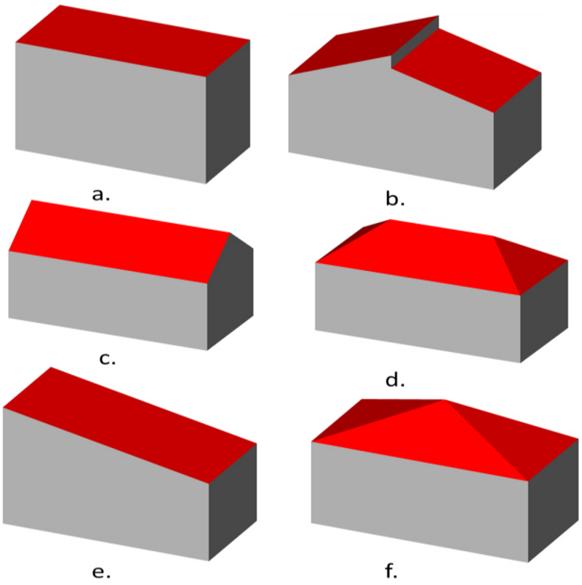


Figure 3: Visualization of the different roof types modeled using CityBIT. (a) Flat roof (b)Dual pent roof (c)Gabled roof (d)Hipped roof (e)Mono-pitch roof (f) Pavilion roof

- **Roof Heading:** Similar to the building heading, the user can select the orientation of the roof as well.
- **Building Function:** As the building function plays an important role in analysing the energetic performance of a building, the function/usage for individual building(s) can also be entered by the user.
- **Storeys Above Ground:** These refer to the number of above grounded floors of the building.
- **Storeys Below Ground:** These refer to the number of underground floors for the buildings.
- **Terrain Intersection:** It is an attribute of a building or building part and is generated by the intersection of the building/building part with the terrain.

Another approach for the model creation using the interpolation techniques is described below.

Model creation using interpolation

Starting from building level to an urban scale simulation, the amount of required relevant data for energy analysis can differ with respect to the considered data model and/or simulation environment (Malhotra, et al. 2020). Therefore, considering the application use case(s) of urban energy performance simulation, CityGML data models must be geometrically detailed and correctly modeled. As mentioned in the previously described technique of model creation using basic user inputs, some important parameters (if missing) are interpolated in the tool with respect to the user selected dataset (explained later).

Moreover, based on the characterizations in Wate and Coors (2015) and Kolbe (2009), certain building parameters do play an important role in the energetic analysis of an urban area. Studies such as Strzalka, et al. (2015) and Jaeger, et al. (2018) related to the sensitivity analysis of building geometries also highlight the importance of attributes such as building orientation, etc. for energy simulation.

During the planning phases of a new construction, different building parameters have to be checked, investigated upon and thereafter finalized. The CityBIT, therefore, facilitates the model creation not only using the user inputted values but can also interpolate different attributes for the creation of new CityGML models based on the buildings in the surrounding area. These interpolation techniques can be selected by the user within the tool. The main techniques and methods for implementing CityBIT are based on previous studies (Meijering, 2002) and python libraries such as ‘Math’ (Van Rossum, 2020) or ‘Scipy’ (Virtanen, et al. 2020), and will be explained below:

- **Linear Interpolation**: This is a method of curve fitting using linear polynomials to construct new data points within the range of a discrete set of known data points (Arlinghaus, 1994).

- **Nearest-neighbour Interpolation**: It is often used to derive estimates of the surface height at the vertices of a regular grid from irregularly spaced samples (Bourke, 1999). It is a method of sampling multivariate interpolation in one or more dimensions.
- **Cubic Interpolation**: It is the simplest method that offers true continuity between the segments. It requires more than just the two endpoints of the segment but also the two points on either side of them (Bourke, 1999). However, for CityBIT the 'Cubic Interpolation' is currently not preferred by the authors due to the method's higher computational requirements and irregularity in results.

Moreover, along with the previously mentioned attributes such as 'Information about the desired output coordinate(s)/surface' and 'Input and Output Coordinate Reference System (CRS)', other inputs/methods which are required to be entered/selected by the user for interpolation are explained below:

- **Input CityGML Dataset**: To interpolate the parameters for model creation, the user needs to provide an input dataset. This refers to the CityGML data models on the basis of which the attributes can be imported by the tool. Once the data is inserted into the tool, the user can limit the interpolation using two different methods:
    - **Using specific attributes**: If the roof type and/or the building function have been inputted by the user, they can limit/filter the interpolation only to buildings with similar attributes in the input CityGML dataset.
    - **Limit by coordinates:** If this option is selected by the user, they can:
      i) select all the buildings inside the datasets.
      ii) input an enclosed polygon to select the buildings inside the inputted area.
      iii) by the number of buildings (e.g. the 500 closest buildings to be interpolated upon).
      iv) by a radius from the building for which the parameters have to be interpolated.

Furthermore, the parameters that can be interpolated within the tool with the help of previously mentioned techniques are:

- **Building Height**: The height of the new building is interpolated with respect to the surrounding buildings.
- **Ground Surface Height**: The surface height is also interpolated based on the considered buildings in the dataset.
- **Ground Surface Area**: The area of the ground surface polygon is calculated based on the considered buildings as well as checking the intersection of the polygon with the polygon surfaces of the other buildings.
- **Side ratio**: This is also calculated for the buildings with similar construction side ratios.
- **Building heading**: The orientation of the building is interpolated based on the orientation of other buildings in the dataset.
- **Building function**: The interpolation of the building usage is currently under development and will be available in the future versions of the tool.
- **Roof type**: Similar to the other parameters, the interpolation techniques outputs the suitable roof type based on the other buildings.
- **Roof Heading**: The interpolation of the orientation of the roof is currently under development and will be available in the future versions of the tool.
- **Storeys above/below ground**: These parameters are calculated in two different ways:
    - Interpolated based on the user selected dataset. The average number of storeys are calculated and therefore assigned to the building.
    - Once the height of the building is interpolated, the average height of the floors is calculated. Furthermore, the number of storeys below or above ground level are also calculated and assigned to the individual building(s).
- **Terrain Intersection**: To interpolate the terrain intersection of a building, the terrain height is calculated for an area of a maximum of 2 km^2 within the input dataset. Furthermore, using the terrain height, the intersection points for the desired building(s) are calculated and considered in the building model creation.

The next chapter demonstrates the implementation of creating new CityGML building model(s) using CityBIT.

IMPLEMENTATION

To demonstrate the functionalities of CityBIT, the authors considered open source CityGML data (Landesbetrieb Information und Technik Nordrhein-Westfalen, Ref 324 Geoinformationszentrum, 2020) for a city in Germany. Furthermore, in this paper, the authors

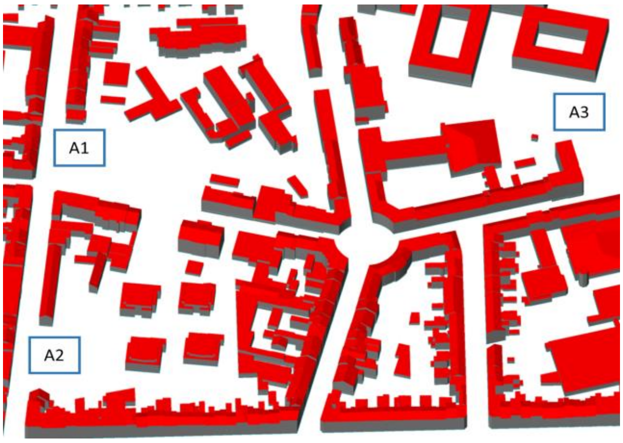


Figure 4: Section of dataset D1 and the selected areas A1, A2 and A3

used the FZK Viewer (KIT/IAI, 2019) for the visualisation of CityGML data models.

The previously mentioned dataset in the bottom of Figure 1, which will be named D1 in further scope of this paper, consisted of 1521 buildings and 1825 building parts. Though, this dataset is modeled in LoD2, it is still available in CityGML 1.0. In order to convert between versions of CityGML, the authors used the CityATB (Malhotra et al., 2020). Figure 4, demonstrates a section of the dataset D1 and areas A1, A2, A3 where the authors intend to create new building models.

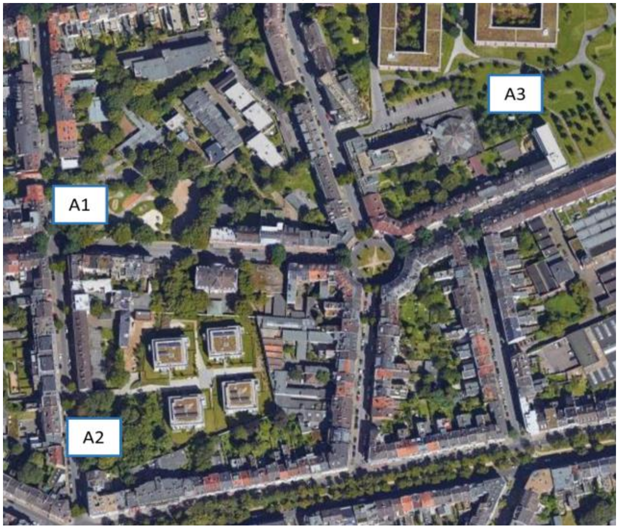


Figure 5: Visual representation of the selected areas A1, A2 and A3 using web mapping service of Google Maps

Currently, only a single building model can be created using CityBIT at once, however, simultaneous model creation is also envisioned by the authors in future. Figure 5 gives an overview of the selected areas using the web mapping service of Google Maps (Google, 2020). Though the selected areas are characterized as green areas in the city, the authors selected these areas only for demonstration purposes.

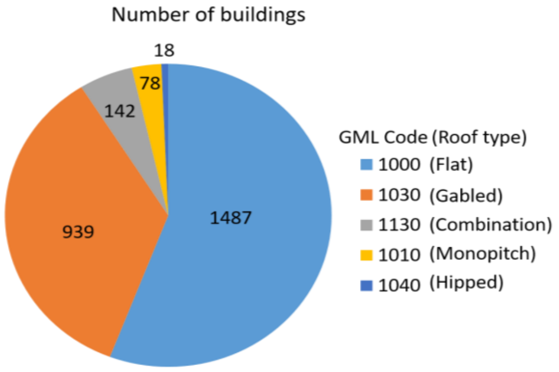


Figure 6: Distribution of the buildings and building parts according to the CityGML roof types codes in dataset D1

For creating the intended building models B1 and B2 in areas A1 and A2 respectively, interpolation techniques were used. Parameters such as roof type, building height, etc. were interpolated based on the surrounding buildings in the dataset. Figure 6 shows a distribution of roof types within the dataset D1. Here, 55.7% of the buildings (1487 out of 2667) were modelled with flat roofs (CityGML code: 1000).Due to the higher percentage of flat roofs in D1, the buildings B1 and B2 were also interpolated with flat roofs. Furthermore, for B1 and B2, as the building heights were also interpolated from the dataset, for each of them, the calculated height was 9.79 m and 13.04 m, respectively.

The building B3 in area A3 was created using the user inputs of 12 m for building height with pavilion roof and it is considered to be entirely above terrain. Figure 7 shows the buildings B1, B2 and B3 that were created for the areas A1, A2 and A3 respectively using CityBIT. Due to data protection laws, the authors do not include the exact coordinates of the created buildings in this paper.

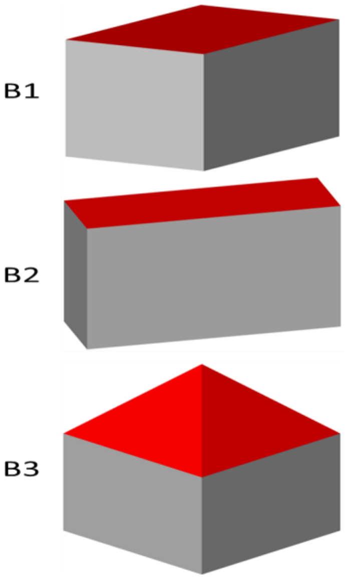


Figure 7: Buildings B1(top), B2 (middle) and B3 (bottom) created using CityBIT in areas A1, A2 and A3, respectively

Figure 8 represents the visualization of the newly created building models combined with the large dataset D1. The process of combining multiple CityGML files together was carried out using the CityATB (Malhotra et al., 2020).

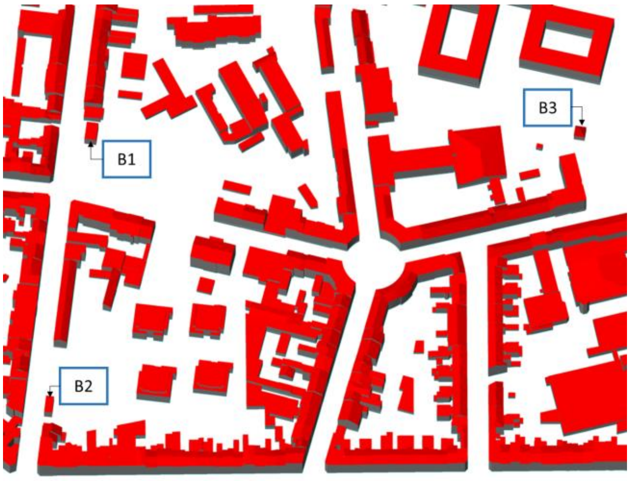


Figure 8: Visualization of the buildings B1, B2 and B3 in the combined dataset D1

As no parameters were interpolated for building B3, the creation of the new CiyGML dataset only took 0.0137 seconds. Table 3 gives an overview of the time analysis for creating the buildings B1, B2 and B3 along with their respective height, ground surface areas, roof types and the attributes that were interpolated.

Table 3: Overview of creation time analysis, height, ground surface areas and volume

Building	B1	B2	B3
Creation Time (s)	8.39	9.39	0.0137
Height (m)	9.79	13.04	12.00
Ground Area (m^2)	108.8	80.06	60.30
Roof Type	Flat	Flat	Pavilion
Interpolated attributes (for B1 and B2)	Ground surface area, side ratio, surface height, building height, roof type, roof height, building function and building heading		None

With a user friendly graphical user interface and a modular architecture, the CityBIT can be integrated with other tool chains and workflows. Figure 9 shows the main window of the CityBIT. The user interface is currently being developed using Pyside 2 Python QT bindings (PyQt, 2012) and the tool is compatible with Python version 3.5+. Furthermore, the tool exports building models in CityGML version 2.0. Presently, the tool is being further developed and rigorously tested. CityBIT is available open-source under the MIT license (https://gitlab.e3d.rwth-aachen.de/e3d-software-tools/citybit/citybit).

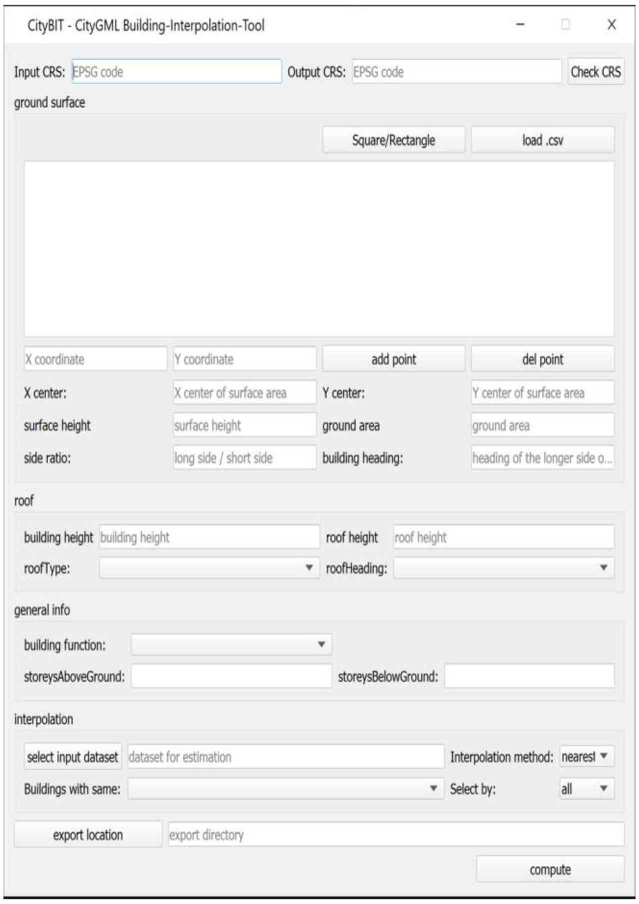


Figure 9: User interface window of CityBIT

CONCLUSIONS

The CityBIT is developed to facilitate model creation for planned and existing buildings. This tool allows dataset creation for areas where no or less data exists. Using the tool, the authors were able to develop new building models for existing city quarters. Moreover, urban areas with individually combined building models were also created and geometrically validated using tools such as GML ToolBox (IAI/KIT, 2020), *val3dcity* (Ledoux, 2018). Some interpolation techniques for attributes such as building heading, building function, etc. are being further improved for greater accuracy. In future the authors would like to qualitatively analyze the interpolation techniques for the generated models. It is envisioned to integrate district level model creation in the tool and further test different use cases for urban building energy modeling. The authors would also like to demonstrate the functionalities of the open-source tool in different urban scale researches and projects.

ACKNOWLEDGEMENT

Part of this work was funded by the German Federal Ministry of Economics and Energy (BMWi), promotional reference 03EWR010B.

REFERENCES

Agugiaro, G., Benner, J., Cipriano, P. & Nouvel, R., 2018. The Energy Application Domain Extension for CityGML: enhancing interoperability for urban energy simulations. Open Geospatial Data, Software and Standards, 5 March.

Arlinghaus, S. L., 1994. PHB Practical Handbook of Curve Fitting.: CRC Press.

Bourke, P., 1999. Interpolation methods: paulbourke.net.

Crawley, D. B., Hand, J. W., Kummert, M. & Griffith, B. T., 2008. Contrasting the capabilities of building energy performance simulation programs. Building and Environment, 06 December, pp. 661-673.

Google, 2020. GoogleMaps. [Online] Available at: https://www.google.de/maps/ [Accessed 03 October 2020].

Gröger, G., Kolbe, T., Nagel, C. & Häfele, K., 2012. OGC City Geography Markup Language (CityGML) Encoding Standard, s.l.: OGC.

IAI/KIT, 2020. GML-Toolbox, Karlsruhe: KIT.

IFC Web Server, 2018. IFC Webserver. [Online] Available at: ifcwebserver.org

Institut für Automation und angewandte Informatik (KIT), 2019. FZKViewer (5.2). [Online] Available at: https://www.iai.kit.edu/1302.php [Accessed 19 February 2020].

Jaeger, I. D., Reynders, G., Ma, Y. & Saelens, D., 2018. Impact of building geometry description within district energy simulations. Energy, 01 September, pp. 1060-1069.

Kolbe, T. H., 2009. Representing and Exchanging 3D City Models with CityGML. In: 3D Geo-Information Sciences. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 15-31.

Landesbetrieb Information und Technik Nordrhein-Westfalen (Ref 324 * Geoinformationszentrum), 2020. https://www.opengeodata.nrw.de. [Accessed 30 January 2020].

Ledoux, H., 2018. val3dity: validation of 3D GIS primitives according to the international standards.. Open Geospatial Data, Software and Standards, 06 December.

Malhotra, A. et al., 2020. A review on country specific data availability and acquisition techniques for city quarter information modelling for building energy analysis. in Proceedings of BauSIM 2020, Graz.

Malhotra, A., Raming, S., Frisch, J. & Treeck, C. v., 2020. CityATB (CGML ATB) : CityGML Analysis Toolbox for Energy Performance Simulations, in Proceedings of BauSIM 2020 Graz, 194-201.

Malle, T., Fischer, M. & Bazjanac, V., 2007. Building energy performance simulation tools-a life-cycle and interoperable perspective.: Center for Integrated Facility Engineering (CIFE).

Masanet, E. et al., 2016. Energy Technology Perspectives 2016-Towards Sustainable Urban Energy Systems. International Energy Agency.

Meijering, E., 2002. A Chronology of Interpolation: From Ancient Astronomy to Modern Signal and Image Processing. Proceedings of the IEEE, pp. 319-342.

Nouvel, R. et al., 2015. Urban Energy Analysis nased on 3D city model for national scale applications. Aachen, IBPSA, pp. 83-90.

NREL, 2019. EnergyPlus. [Online] Available at: https://energyplus.net/ [Accessed 21 July 2019].

Palme, M. & Salvati, A., 2018. UWG - TRNSYS Simulation Coupling for Urban Building Energy Modelling. IBPSA England, 11-12 September, pp. 635-641.

PyQt, 2012. PyQt Referencing Guide.

Remmen, P. et al., 2017. TEASER: an open tool for urban energy modelling of building stocks.. Journal of Building Performance Simulation, February, pp. 84-98.

SIG3D, 2018. Modeling Guide for 3D Objects: Part 2 Modeling of Buildings (LoD1, LoD2, LoD3). Available at: https://en.wiki.quality.sig3d.org/ [Accessed 20 November 2020].

Software, S., 2020. FME - The Simple Solution for Complex Integration. [Online] Available at: https://www.safe.com/ [Accessed 29 October 2020].

Sola, A., Corchero, C., Salom, J. & Sanmarti, M., 2018. Simulation Tools to Build Urban-Scale Energy Models: A Review. Energies, 13 November, pp. 1-24.

Strzalka, A., Monien, D., Koukofikis, A. & Eicker, U., 2015. Sensitivity analysis for minimization of input data for urban scale heat demand forecasting.. Nottingham, UK,

Tian, W., 2018. A review of uncertainty analysis in building energy assessment. Renewable and Sustainable Energy Reviews, October.

Van Rossum, G., 2020. The Python Library Reference, release 3.8.2. s.l.:Python Software Foundation.

Virtanen, P. e. a., 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods.

Wate, P. & Coors, V., 2015. 3D Data Models for Urban Energy Simulations in 6th International Building Physics Conference, IBPC 2015.

Welsh Government, 2013. Technical Guidance: Permitted development for households: Crown.