

DEVELOPMENT OF QGIS PLUGIN FOR URBAN ENERGY SIMULATION USING 3D CITY MODEL AT THE CITY DISTRICT LEVEL

Mohammad Hosseingholizadeh^{1,2}, Volker Coors², Hamidreza Ostadabbas¹, Frank Friesecke¹

1: die STEG Stadtentwicklung GmbH, Germany; 2: Stuttgart University of Applied Sciences

KEY WORDS: Energy demand estimation, reducing carbon dioxide, ALKIS and CityGML data, QGIS plugin, 3D visualization

ABSTRACT:

In the context of climate change, the increasing demand for energy-efficient buildings and sustainable urban development has become a pressing issue due to the significant proportion of global energy consumption and carbon dioxide (CO_2) emissions attributable to the building sector. This requires a concerted effort to reduce its environmental impact, and Geographic Information System (GIS) applications are vital tools for achieving this by optimizing heat supply, calculating costs, analyzing profitability, and balancing CO_2 emissions. This study aims to address the challenge of achieving energy efficiency and reducing CO_2 emissions in the building sector, specifically at the district level. To this end, the research objective is to develop a QGIS plugin that can simulate urban energy demand at the district level by integrating 2D data with CityGML files and connecting QGIS to SimStadt software via API to visualize the simulated urban energy results in 3D on the Web Globe. The proposed plugin leverages the open-source QGIS tool QField to capture building conditions and connect 2D and 3D data on urban energy simulation. Supplementary to this, this plugin provides up-to-date information on energy demand, consumption, CO_2 emissions, building component conditions via updating related tables in the database. Decision-makers can use this comprehensive and user-friendly tool to understand and act on the results, ultimately leading to a CO_2 -neutral district by 2045. The development of the QGIS plugin represents a significant step towards sustainable urban development and climate change mitigation by utilizing GIS applications for optimizing energy demand and reducing CO_2 emissions in the built environment.

1. INTRODUCTION

One of the most overwhelming worldwide topics for several governments in the coming years is how to protect the environment from the drastic climate changes to have climate neutrality (Singla and Padia, 2021). Moreover, declining CO_2 emissions and boosting energy efficiency have already become paramount tasks to tackle climate change and reduce district heating dependency to fossil fuels (Dermentzis et al., 2021). Approximately a third of definitive energy consumption and greenhouse gas emission belong to the residential and non-residential building sectors (Bundesregierung, 2021). Therefore, a combination of reducing energy consumption and utilizing renewable energy leads to reaching the German plan that is CO_2 neutral by 2045 for building sectors (Bundesministerium fuer Umwelt Naturschutz nukleare Sicherheit, 2019).

The utilization of renewable energy has witnessed remarkable progress in the electric energy sector, but its adoption in other sectors, particularly heating systems, has been comparatively slower. According to the latest data from the German Association of Energy and Water Industries, approximately half of residential buildings in Germany rely on gas for heating, while a quarter utilize oil. District heating, wood, heating pumps, and electricity are also used for heating purposes (BDEW, 2022). Furthermore, it is worth noting that gas remains the predominant heating source for nearly 30 percent of newly constructed buildings. Despite renewable energy accounting for nearly 20 percent of total final energy consumption for heating buildings in 2020 (BDEW, 2022), the heating demand continues to represent a significant proportion of the overall energy demand in the German building sector (Weiler and Eicker, 2021). These findings emphasize the need to prioritize energy-efficient solutions and further integration of renewable energy sources

in heating systems to achieve sustainability and reduce greenhouse gas emissions.

According to (Wate and Coors, 2015), there are various energy-efficient development strategies that can be adopted at the district level to combat climate change. To reduce greenhouse gas emissions, these strategies must integrate the use of renewable energy sources and focus on improving energy efficiency, as emphasized by (Strzalka et al., 2011). The German federal government and KfW banking group have been providing funds to support the implementation of these strategies through programs such as Energy-efficient Urban Redevelopment (Energetische Stadtsanierung), which is grant program 432 (KfW, 2022). Collaboration among various stakeholders, including municipalities, property owners, city planners, and the energy industry, is essential for the success of urban energy-efficient redevelopment projects that focus on building stock, heat supply, climate-friendly mobility, electricity use, and the use of renewable energies, as highlighted by (Strzalka et al., 2012). Hence, an adaptable application that can estimate and simulate building energy consumption, even with minimal data availability, is needed to help municipalities achieve their goal of becoming CO_2 -neutral districts by 2045.

In accordance with several studies, including (Weiler and Eicker, 2021), utilizing a high allocation of heat pumps and fossil-free district heating can effectively reduce greenhouse gas emissions. Different studies have attempted to evaluate the best approaches for achieving energy efficiency through diverse measurement and pathway methods. For instance, (Swan and Ugursal, 2009) used composite indicators such as GDP, energy prices, and climate change to apply up-down methods to assess the energy demand of a district area. On the other hand, (Fumo and Biswas, 2015) focused on how bottom-up methods could be used to evaluate energy demand at the city district

level. The latter approach employs statistical models to extend the evaluation of energy demand from individual buildings to the entire city, based on parameters extracted from statistical models. This method is suitable for municipalities with similar structures, ranging from building geometry data to building-specific data. In contrast, the physical models in the bottom-up method are based on building interaction with the environment and building physical components, as shown in the work of (Reinhart and Davila, 2016). This approach requires more detailed information, such as BIM (Building Information Model) and weather data, to estimate energy demand accurately. However, providing such information can be challenging, costly, and time-consuming for most municipalities.

The initial step in estimating energy demand involves calculating the heating and domestic hot water requirements of each building. This is accomplished by using a 3D city model that represents the building's geometry. SimStadt can be used to carry out these calculations, which include determining the volume, gross area, heated area, outer walls, roof area, and windows area. In addition, the construction year of the building is a crucial parameter that must be included in CityGML files. This information is used to determine the typical materials of the building elements, which are subsequently used in further calculations regarding building physics. However, the lack of data documentation, outdated data, incomplete information, and data protection may result in the unavailability of critical information related to building geometry and construction year, resulting in problems in estimating energy demand accurately (Biljecki et al., 2016).

The objective of this study is to determine the energy demand, energy consumption, and CO_2 emissions by utilizing 2D data enriched with field data capture, using CityGML data, or the combination of both in case of having lack of data in CityGML file. To achieve this goal, we will develop an open-source QGIS plugin that can be adapted to different countries by modifying local information such as weather data. The plugin will provide an initial estimation for energy agencies and urban planning companies to gain insight into the community's current energy usage. The plugin automates all calculations and generates maps required for all energy projects. Furthermore, the results obtained will be integrated into CityGML files and displayed on a web globe for better visualization. This plugin will also contribute to the overall goal of reducing greenhouse gas emissions and combating climate change. By providing accurate estimations of energy demand and consumption, urban planners and energy agencies can make informed decisions on how to reduce energy usage and promote the implementation of renewable energy sources. This, in turn, can contribute to the creation of more sustainable and resilient communities.

2. MATERIAL AND METHODS

2.1 Study area

The reason for selecting Burgrieden in Baden-Württemberg, Germany as the study area is due to specific factors that make it suitable for this research. While we cannot disclose the exact location for confidentiality reasons, Burgrieden offers certain advantages that contribute to the research objectives. Firstly, the state of Baden-Württemberg and Germany as a whole have robust data security measures, ensuring the protection of sensitive information. This aligns with the need to maintain data privacy

and security throughout the research process. Secondly, Burgrieden provides a representative sample of the region, with a diverse building stock and varied roof structures that are essential for accurately estimating energy demand. The presence of different building types, including residential, schools, bakeries, restaurants, pharmacies, and small shops, further enhances the richness of the data collected. These factors collectively make Burgrieden a suitable location to conduct this research and generate meaningful insights into energy demand estimation and building refurbishment. By selecting Burgrieden as our study area, we can ensure data confidentiality while obtaining comprehensive and reliable data for analysis, contributing to the robustness and validity of our research findings.

2.2 Dataset

2.2.1 2D Data: Since 2015, the Authoritative Real Estate Cadastre Information System (ALKIS) has been implemented in all states of Germany, providing official and cadastral data that includes both spatial and non-spatial information in a NAS format based on Extensible Markup Language (XML). This study will utilize ALKIS as the primary source of data to obtain essential information such as street names, house numbers, and ALKIS IDs. In addition, ALKIS will serve as an input layer for data capturing in QField, a mobile application for collecting data in the field. The use of ALKIS data in this study will ensure the accuracy and reliability of the data collected and analyzed, which is essential for achieving the study's objectives.

2.2.2 Field Data: This data contains extensive information on the building's construction and energy characteristics, which is crucial for estimating its energy demand and carbon emissions. This information will serve as input data, in combination with 2D data from ALKIS, for developing an open-source QGIS plugin that can automatically calculate these energy-related parameters. By integrating this data into the plugin, it will allow for a more accurate and comprehensive estimation of energy demand and carbon emissions at the district level, providing a useful tool for urban planners and energy agencies.

2.2.3 CityGML Data: CityGML is an open data model for the representation of 3D urban objects, which is used to provide a standardized way of encoding and exchanging information about the city's features. The Level of Detail (LoD) is a parameter in CityGML that indicates the level of geometric detail and completeness of a 3D model. LoD2 in CityGML includes the external shape of buildings and their basic internal structures, such as floors and walls, as well as the basic surface properties of walls and roofs. This level of detail is appropriate for various applications, such as urban planning, energy simulation, and disaster management.

2.3 Methodology

The figure 1 shows the proposed procedure for the entire methodology. The overall procedure is as follows:

- Import CityGML (LoD2) and/or ALKIS data into the PostgreSQL database
- Importing the cleaned data into the QGIS and prepare the tablets for field data capturing
- Using attribute form of QGIS and QField application for this purpose

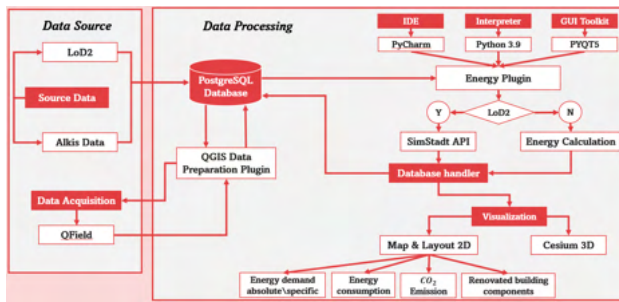


Figure 1. Methodology

- After field data acquisition, captured data (energetic and construction building parameters) should be stored or updated in the database via the developed plugin in the QGIS
- According to the availability of users data (in this research we use both to compare the result), users decide which energy calculation method should be selected
- If the CityGML data exist, a post request should be sent to the SimStadt API to get the result of calculated energy demand. Also with this request users can get some extra information such as information in the CityGML file like time of building refurbishment date, constructed parameter like height, area of roofs, outer walls, cellar ceiling, and the volume of building
- If the CityGML file does not exist, 2D data like ALKIS data enriched by field data capturing imported to calculate energy demand, energy consumption, and CO_2 emission
- Update database with the calculated energy results
- Creating different maps and layouts in case of using 2D data
- Visualizing the result in the Cesium

2.3.1 Database Structure The development of the database in this study was carried out using the Django framework. The figure shown in (2) represents the database structure, which consists of four categories: ALKIS data, CityGML data, energy data, and project data. The ALKIS tables are linked to the project table through a one-to-many relationship, where each building and land parcel belong to a specific project. Similarly, the project table is linked to the CityGML tables. The CityGML category contains ground surface, roof surface, and wall surface, all of which have a one-to-many relationship. The CityGML building table stores all values in 3D format, while the other CityGML tables store coordinates in 2D format. This feature allows users to import data directly into GIS software such as QGIS, where both 2D and 3D models can be created using CityGML data. Thus, the combination of ALKIS and CityGML data in this database enables the creation of detailed building models that accurately represent the building and its surroundings. This data is useful in various applications, such as urban planning and energy analysis. The ALKIS data provides information about the cadastral boundaries, while the CityGML data provides a 3D model of the building and its components, allowing for a detailed representation of the building and its environment.

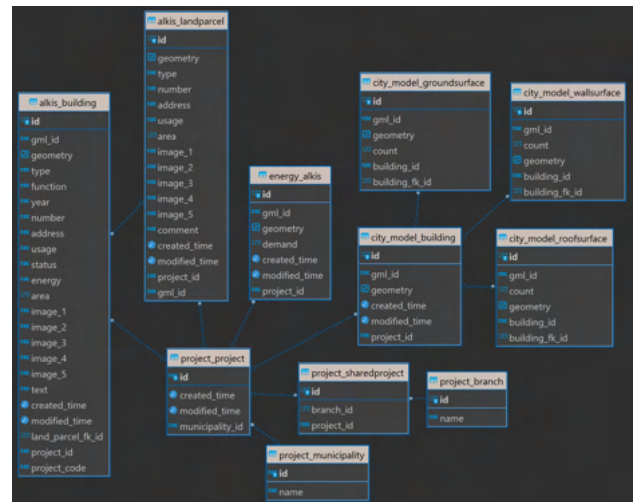


Figure 2. Database Structure

2.3.2 Energy demand calculation with ALKIS data (2D data) Ensuring accurate and reliable data is crucial for determining the energy requirements of buildings and performing associated calculations. However, purchased external data is often outdated, lacking detail, and fails to provide an accurate reflection of reality. To tackle this issue, several measures have been implemented. One of these measures involves using ALKIS data, which provides only the field plan geometry specific to our needs. Despite this apparent limitation, this data is sufficient for calculating all the necessary areas and volumes. However, capturing field data is necessary for achieving this. Besides computing the relevant areas and volumes, other parameters must be surveyed to guarantee the data foundation's reliability and precision for subsequent computation.

- Verification of building construction year
- Number of stories, referring to the number of floors in a building
- Height of the floors, plinth height of the flap tile [m], referring to the measurement of floor heights and the base height of the outer wall
- Roof shape, roof angle [°], referring to the shape and angle of the building's roof
- number of attics/levels, referring to the number of additional levels or spaces that may be present in the building's roof area

The focus of this section of the research study is on developing an automated approach for determining a building's energy demand and CO_2 emissions. This is achieved by computing and establishing the parameters outlined in (David et al., 2022) for the building stock. Equation 1 is utilized to calculate the heating energy requirement for each building (Q_{ht}). This is done by taking into account various factors such as the total demand for hot water (Q_w), heat losses due to transmission (Q_{tr}) and ventilation (Q_{ve}), heat gains due to solar radiation (Q_{sol}), as well as heat gains from internal sources (Q_{int}).

$$Q_{ht} = Q_w + Q_{tr} + Q_{ve} + Q_{sol} + Q_{int} \quad (1)$$

The amount of heat needed to supply domestic hot water (Q_w) can be determined using equation 2, which requires information about the building's floor area and number of levels.

$$Q_w = area \times 0.84 \times (level + 1) \times 12.5. \quad (2)$$

Equation 3 can be used to approximate Q_{tr} , which represents the amount of heat lost through the building envelope due to transmission. This equation incorporates several factors, including the total area of the building envelope (including roofs, walls, windows, and cellar ceilings), the average U-Value of the building envelope, a conversion factor of 0.024 from Watt to kW per hour per day, and the long-term average temperature of the weather station (David et al., 2022).

$$Q_{tr} = area_{total} \times U_{value-average} \times 0.024 \times Gt_{15-20}. \quad (3)$$

The formula for calculating heat losses through ventilation (Q_{ve}) is presented in Equation 4. This equation takes into account several parameters, including the volume-specific heat capacity of air ($c_{(p,air)}$), which is equivalent to $0.34 \frac{Wh}{m^3K}$. The average air change rate during the heating season for an old, non-refurbished building ($n_{(air,use)}$) is assumed to be 3 1/h, while the air change rate due to infiltration ($n_{(air,infiltration)}$) is estimated to be 0.2 1/h. The reference area of the building is represented by $area_r$, and h_r refers to, the reference height of the ventilation room.

$$Q_{ve} = 0.024 \times c_{(p,air)} \times (n_{(air,use)} + n_{(air,infiltration)}) \times area_r \times h_r \times Gt. \quad (4)$$

Equation 5 provides a formula for determining the amount of heat gained through solar radiation (Q_{sol}). This equation incorporates several factors, including the reduction factor for external shading (F_{sh}), which is assumed to be 0.6. The frame fraction of the windows (F_F) is equal to 0.3, and there is a reduction factor (F_W) that considers the radiation of non-perpendicular angles to the glazing. F_{EA} represents the total area of the windows, and $I_{(sol,j)} (\frac{kWh}{a})$ is the average global irradiation received by surfaces oriented in direction j during the heating season.

$$Q_{sol} = F_{sh} \times (1 - F_F) \times F_W \times g_{(gl,n)} \times \sum_j (0.25 \times F_{EA} \times I_{(sol,j)}), \quad (5)$$

where $j \in \{E, W, S, N\}$ are equal to $271 \frac{kWh}{a}$, $271 \frac{kWh}{a}$, $392 \frac{kWh}{a}$, $271 \frac{kWh}{a}$, $160 \frac{kWh}{a}$.

The formula for calculating the amount of heat gained through internal heat sources (Q_{int}) is given by Equation 6. This equation takes into account the average thermal output of internal heat sources, which is assumed to be $3 \frac{W}{m^2}$. $area_{total}$ represents the reference area of the building in square meters. The number of heating days in Germany, which is the duration of the heating period, is assumed to be 222.

$$Q_{int} = 0.024 \times 3 \frac{W}{m^2} \times 222 \times area_{total} \quad (6)$$

Equation 7 can be used to estimate the amount of CO_2 emissions based on the type of energy source. The value of hs_{const} represents the parameter that varies depending on the specific type of energy source being used. The constant energy sources value such as Natural Gas, Liquid Gas, Heating oil, Electricity and Wood are equal to 0.2450, 0.2670, 0.3190, 0.5650, and 0.0190.

$$CO_2 = Q_{tr} \times hs_{const} \quad (7)$$

2.3.3 Energy demand calculation with CityGML data (3D data)

The initial step in estimating the energy demand for buildings involves determining their heating and domestic hot water requirements. This is typically accomplished by utilizing a 3D city model, which provides a representation of the building's geometry, allowing for the calculation of various parameters such as volume, gross area, heated area, outer walls, roof area, and window area. SimStadt, developed by Stuttgart university of applied sciences, is a powerful software tool that enables all of these calculations to be performed efficiently. Additionally, the construction year of a building is another essential factor that should be available in CityGML files. By using the construction year, the typical materials of building elements can be determined based on building physics, which is then used for further calculations. However, there are several reasons why critical pieces of data such as building geometry and construction year may not always be available. These reasons include lack of data documentation, outdated data, incomplete information, and data protection concerns. In such cases, the integration 2D and 3D data can solve the problem.

The transferability of the research findings presented in this paper is supported by the utilization of SimStadt, a comprehensive energy simulation tool widely recognized and implemented internationally. SimStadt has been successfully applied in various projects and collaborations, including joint publications with esteemed organizations such as the Joint Research Center of the European Union. Furthermore, the data model utilized in this study is based on an international standard, ensuring compatibility and interoperability with data infrastructures in different countries. These aspects highlight the broader applicability of our research beyond the specific context of Germany. While recognizing that each country has its unique characteristics, our methodology and findings provide a robust framework that can be adapted and utilized in diverse international settings. As future research endeavors continue to explore and refine our approach, international collaborations and exchange of knowledge will facilitate the development of country-specific guidelines and frameworks, fostering sustainable and energy-efficient built environments worldwide (Martirano et al., 2022).

The following procedure is needed to get the energy demand from SimStadt.

1. Before using the CityGML file for energy demand calculations, it is essential to verify that the file includes key information such as the GML ID, building construction year, and building functionality. This can be accomplished through a careful review of the metadata and attributes associated with the file. Only once this information has been confirmed can accurate and reliable energy demand estimations be made using the file.

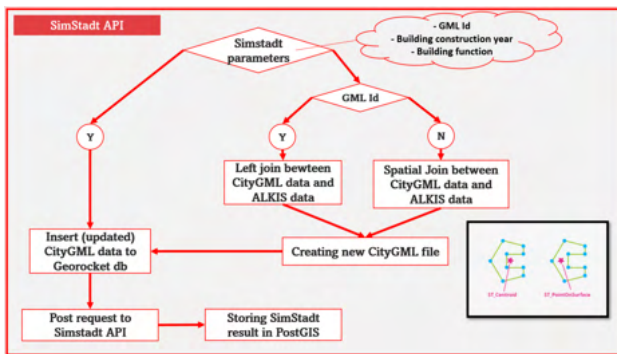


Figure 3. Calculate energy demand by using SimStadt API

- If all three parameters (GML ID, building construction year, and building functionality) are available, then the relevant data should be inserted into the Georocket database. This database provides a centralized location for storing and accessing important building information, including geometry, CityGML tags, and other relevant metrics.
- If either the building construction year or building functionality parameters are missing, it is necessary to check the GML ID. If the GML ID exists in the database, a left join can be made with the ALKIS data to fill in the missing information. However, if the GML ID is not found, a spatial join (figure 4) must be performed between the center point of the CityGML ground surface. To ensure that the point is located on the polygon, the ST_PointOnSurface function should be used instead of ST_Centroid (see Figure 3). After this step, a new CityGML file should be created, and the same process as the previous step should be followed to insert the updated CityGML file to Georocket.
- Once the CityGML data containing the GML ID, building construction year, and building functionality has been inserted into the Georocket database, a POST request should be sent to the SimStadt API to retrieve the calculated values in a JSON format.
- Finally, the requested result from SimStadt will be read by Python and stored in the PostGIS database. Once the data has been stored, it can be used for further analysis, such as identifying buildings with high energy demands or optimizing energy efficiency. The stored data can also be visualized to provide a better understanding of the energy demand patterns across the city.

```

1 create or replace view join_alkis_city_model as
2
3 select
4
5     alkis_building.gml_id as gml_id,
6
7     alkis_building.function as function,
8
9     alkis_building.year as year,
10
11     st_pointonsurface(city_model_groundsurface.geometry)::geometry(POINT, 25832) as geom,
12
13     city_model_groundsurface.gml_id as my_gml
14
15 from alkis_building
16
17 join city_model_groundsurface
18 on st_intersects(alkis_building.geometry, st_pointonsurface(city_model_groundsurface.geometry))

```

Figure 4. SQL code of spatial join implementation between CityGML ground surface and ALKIS data

2.3.4 Creating and finding the best location of PV-panel in open-area

In the following steps, you will find all the necessary procedures to create the PV panel model and identify the optimal location for it inside a selected area (Figure. ??).

- First, utilize the regular point tool of QGIS to identify the center locations of the PV panels.
- Next, create a 3-meter buffer around the PV panel points to ensure that all the points are within the polygon.
- Select only the points that are located inside the buffer (Figure. 6).
- Extract the latitude and longitude coordinates of the selected points as a GeoJSON file.
- If a DTM is available, use it to obtain the altitude information for each point. Unfortunately, the DTM was not available for this research study area.
- Finally, use Blender to generate the 3D model for the PV panels (Figure. 7 and 8).



Figure 5. The boundary of agrivoltaic pv-panels

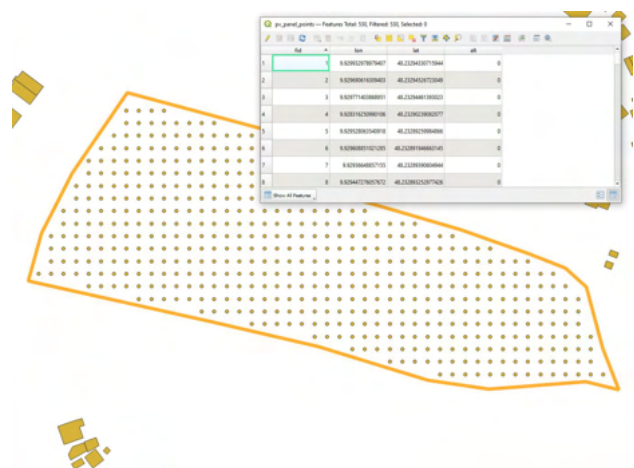


Figure 6. The exact location of well-distributed PV-Panels generated in QGIS

2.3.5 Visualizing the results on the Cesium platform

To visualize 3D objects and 3D tiles in Cesium, the following general steps can be followed:

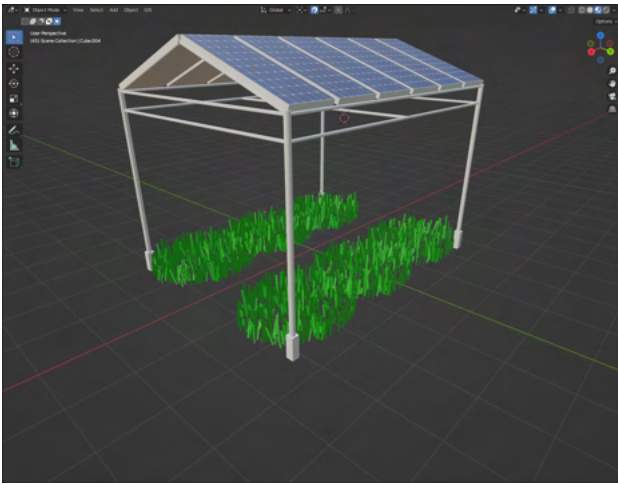


Figure 7. 3D model of agri-pv-panels



Figure 8. 3D model of parking place with e-charging

- Prepare your 3D data: This can be done using software such as Blender, SketchUp, or CityEngine to create or export 3D models or 3D tiles datasets.
- Convert data to Cesium format: To visualize 3D data in Cesium, it needs to be converted to the appropriate format. The Cesium 3D Model Converter can be used to convert models to the *.glTF format, while the 3D Tiles Tools can be used to create tiles and convert them to the *.b3dm or *.i3dm formats.
- Set up a Cesium viewer: A Cesium viewer can be created on a webpage or application using the Cesium library. The library and its dependencies can be downloaded or a CDN can be used.
- Add 3D data to the viewer: Once the 3D data is in the appropriate format, it can be added to the Cesium viewer using the viewer.entities API to add models or the viewer.scene.primitives API to add tilesets.
- Customize the visualization: Various Cesium properties and APIs can be used to customize the visualization of 3D data. Examples include changing the appearance of 3D objects, adding labels, and adjusting the lighting.
- Publish and share: Once the Cesium viewer is set up and 3D data is added, the visualization can be published and

shared with others by hosting it on a server or publishing it to a platform like Cesium ion.

So far, the complete process for 3D visualization has been outlined. To apply this process to the research project, the following steps illustrated in Figure 9 need to be followed to prepare the data for visualization in Cesium. By following these steps, it can be ensured that the data is properly formatted and optimized for use in Cesium, allowing for the creation of an accurate and visually compelling 3D representation of the study area. Proper data preparation is crucial for achieving accurate and effective 3D visualization results, and these steps will help to achieve the desired outcome.

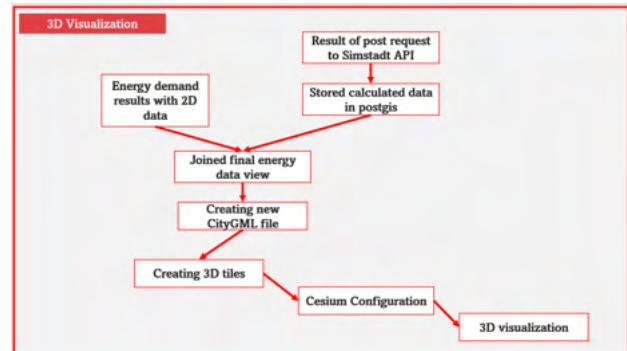


Figure 9. 3D visualization on Cesium platform

3. IMPLEMENTATION

3.1 Developing database using Django framework

The energy plugin's connected database is developed using Django models, which is a high-level Python web framework. This framework offers several advantages, such as using Python classes, which allows users to avoid repetitive updates or modifications to SQL code. The plugin database is categorized into four Django apps: ALKIS, CityModel, Energy, and Project. Figure 10 shows a sample class for this database, and each type is defined by the relative tables. Additionally, a separate table stores the two-dimensional information of building components, such as ground surface, roof surface, and wall surface, with a one-to-many relation to the CityModel building table. This enables users to import the 2D information of roof, wall, and ground directly into QGIS or other GIS software. Moreover, having separate tables for building components allows for more efficient storage and retrieval of information. For example, it enables users to quickly query and visualize all buildings with a certain type of roof or wall material. This can be particularly useful for energy analysis, as different building components may have different energy performance characteristics. With this structures users can access to both 3D and 2D coordinates of features.

3.2 Energy Plugin (ALKIS Data)

In Figure 11.1, the first step in configuring the plugin is to import all relevant information related to the city or municipality name in the config tab. This information is critical for generating accurate maps and layouts. Once this is completed, the user should import the PostGIS database credentials to allow the plugin to establish a connection and access the data. Any errors made during this process will be highlighted with a red warning and logged in the logger section for easy reference.

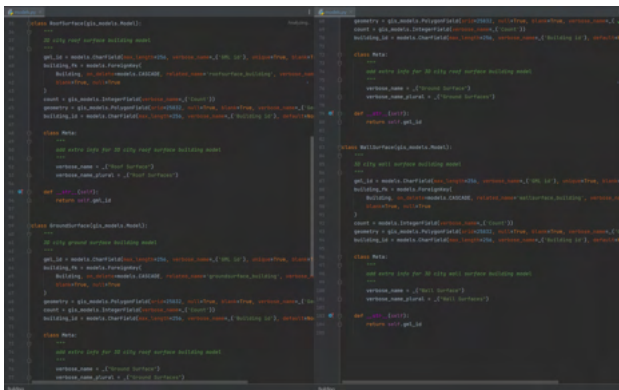


Figure 10. 3D visualization on Cesium platform

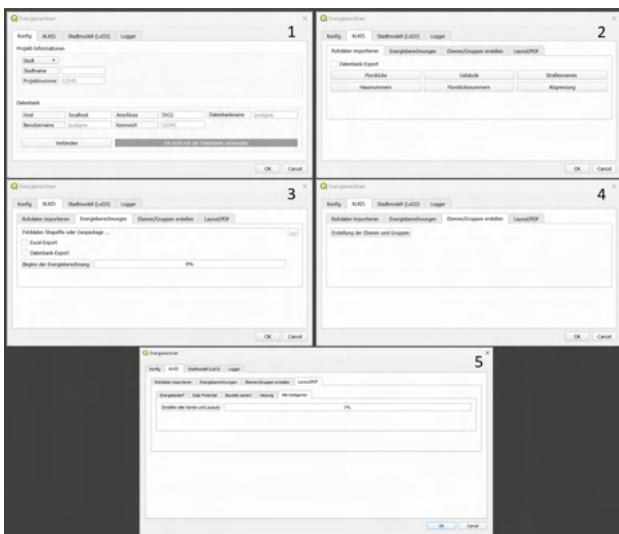


Figure 11. Plugin 2D part. 1. Database configuration. 2. Importing raw data with specific symbology. 3. Calculating energy demand using enriched ALKIS data (Field data). 4. Creating groups and layers with their specific symbology from the calculated energy demand layer 5. Creating different map and layout based on the company format

The first section of the 2D energy plugin is crucial for the successful implementation of energy projects in the city or municipality. It enables users to import raw data such as land parcels, buildings, and project boundaries, as shown in figure 11.2. The database checkbox allows for the automatic storage of all imported data in the database. This section significantly reduces the time required to import raw data and prepare projects. Once the data is imported, users can view the results in the "Raw Data Result" section, as depicted in figure 12.1. This section displays information regarding the imported data, including the total number of imported features and any potential errors or warnings encountered during the import process. Users can use this information to ensure that all data is correctly imported and to address any potential issues before moving on to the next step of the energy project and other urban planning projects.

The third section (figure 11.3) of plugin 2D part is generating different layers and groups with their specific symbology in couple of second with one click. In case of any changes all the layer symbology will be updated automatically. Figure 12.2 shows the result of layers and groups that created automatically.

Eleven different maps and layouts should be created automat-



Figure 12. Result of 2D part. 1. Imported raw data. 2. Categorized result after energy demand calculation. 3. Map and layout creation

ically for various purposes like energy demand, energy consumption, CO₂ emission, refurbished building components, and photovoltaic potential. With this automation, if any changes happen to the data in the database, all the maps and layouts will be updated and created again with one click (figure 11.4 and 11.5). The result showed in figure 12.3.

3.3 Energy Plugin (CityGML data)

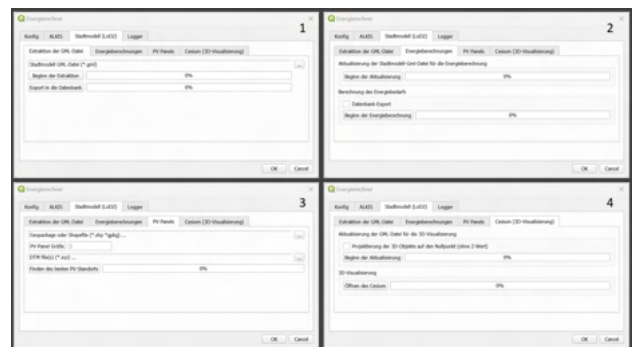


Figure 13. Energy plugin 3D part. 1. Extracting the CityGML file and storing data in the database. 2. Creating new updated CityGML file and calculating building energy demand via sending post request to SimStadt. 3. Finding the best location of 3D panels. 4. 3D visualization on Cesium platform

Figure 13.1 displays the initial step of calculating energy demand using CityGML data by sending a post request to Simstadt. As mentioned in the methodology, one of the first steps is to verify whether the three vital parameters necessary for SimStadt API are available in the data. To do this, CityGML data must be extracted, as demonstrated in Figure 16, which displays a sample of extracted data used for inserting data into the database. During the data extraction process, it is essential to consider the GML ID. If the GML ID does not exist, a new GML ID must be assigned. Therefore a new CityGML file with a new GML ID should be created.

In this section (figure 13.2), if the CityGML file does not have the same GML ID as the ALKIS data, a postgres view connects the ALKIS to CityGML ground surface. In this case, a new CityGML file will be created with all three important parameters (GML ID, building construction year, and building functionality) needed by SimStadt to calculate the energy demand. This is achieved by assigning the GML ID to the CityGML building component. Once the Georocket server is up and running, the CityGML data should be inserted into it, followed by running the SimStadt API and sending a single post request with fixed

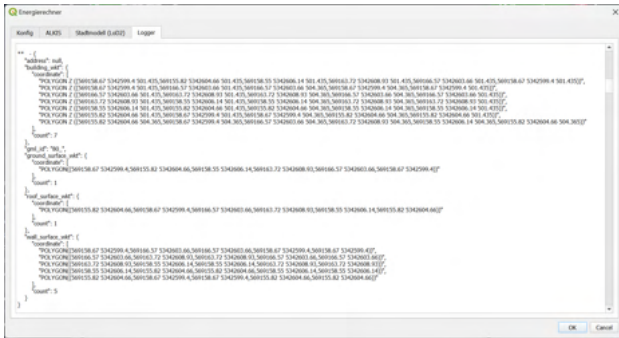


Figure 14. Extracting the CityGML file into the Python dictionary

parameters to it. The result will be returned in a JSON format and stored in the database. But if the CityGML file contains all the three parameter, this step is not needed.

The third part of plugin (figure 13.3), the first step involves importing a Shapefile or Geopackage file that contains the boundary of the 3D objects such as agri-voltaic panels. After importing the file, the size of the 3D panels should also be imported, and this value will be assigned to the JavaScript file automatically. However, it is unfortunate that the Burgrieden municipality does not provide the DTM for the study area. In the case that the DTM is available, it can be imported in this part of the plugin. Alternatively, a faster way to import the DTM is to use the Cesium ion and then add the token to the JavaScript file.

The final step in the QGIS plugin is shown in Figure 13.4. Firstly, the availability of the DTM should be taken into consideration. If it does not exist, the checkbox should be activated to project the entire building component onto a flat surface to prevent the building from flying in the Cesium platform. In this case, a new CityGML file with new z values will be created. Moreover, this updated CityGML file with all the values and information from ALKIS data, field data, CityGML data, and results SimStadt API such as construction year, building usage, energy demand, energy consumption, CO₂ emission and the building refurbishment scenarios. The last part of this section involves Cesium configuration. The first step is to create the 3D tiles. There are many options available for creating 3D tiles, with the most popular one being the commercial software FME. However, in this plugin, the citygml-to-3dtiles Javascript library is used via npm installation. Finally with just one click, Cesium can be launched directly from QGIS and all the HTML, CSS, and JavaScript files run automatically, displaying all the results in Cesium from QGIS. This allows for a seamless transition from QGIS to Cesium and facilitates the exploration and visualization of the 3D city model. By using this plugin, users can easily import and analyze various data sources, including ALKIS data, field data, CityGML data, and results from SimStadt API, to generate a comprehensive and accurate 3D model.

4. RESULT AND DISCUSSION

The results will be as the following:

- Developing a QGIS plugin to simulate urban energy that is adaptable to 2D and 3D data at the city district level
- Connecting the QGIS to the Cesium platform for visualizing the 3D models

- It has been developed based on object-oriented programming (OOP) facilitating future developments.
- Automating map/layout generation and data preparation will avoid the repetition and time-consuming tasks of urban planners and energy experts

For this study area, the energy demand was calculated using both 2D data from ALKIS and 3D data from CityGML. Field data collected through QField, such as construction and energetic parameters of buildings, were used as input data for 2D energy demand calculations. In case of having CityGML data, SimStadt software can be used for energy demand calculation via API from QGIS. The accuracy of SimStadt is far better than calculating energy demand with 2D data because it takes into account the more precise volume and roof/wall surfaces of buildings. In case that the CityGML data does not exist or some parameters like building construction year does not exist, ALKIS or 2D data can be used for the first estimation and evaluation.

All of the plugin codes, which are based on Python, and the visualization part, which is based on JavaScript, are developed using object-oriented programming principles. This makes it easier to further develop and improve the plugin's functionality. Moreover, the logger part of the plugin is an important feature, as it allows users to monitor and access additional information about the plugin's functionality, warnings, errors, and formulas. The development of this plugin can help streamline and simplify the process of generating a 3D city model and analyzing energy demand, allowing for more efficient urban planning and sustainable development.

4.1 3D Visualization

To have the best performance in loading 3D objects on Cesium the following items should be take into consideration. When working with Cesium, there are some practices that should be avoided to ensure optimal performance and efficient use of resources. Firstly, it is recommended to avoid using animations in Cesium as they can cause lag and reduce the overall performance of the platform. Animations require a lot of processing power and can slow down the rendering of the 3D model. Additionally, using a complete package of frameworks like Bootstrap is not recommended as it can lead to conflicts with other frameworks and libraries. Instead, it is advised to use a Content Delivery Network (CDN) or just the gridding system of Bootstrap to minimize the amount of resources required for the project. Another practice to avoid is loading all the coordinates of each object before rendering the entire project. This can cause the platform to become unresponsive and may lead to long load times. Instead, it is recommended to load the coordinates in smaller batches or on-demand to optimize the performance of the platform. Additionally, it is recommended to use classes and call them once in the rendering time and then call functions from the object. This can improve the efficiency of the platform and reduce the amount of memory required for the project. Overall, following these practices can help to ensure that the Cesium platform runs smoothly and efficiently, providing a seamless 3D visualization experience. Therefore, it is important to be mindful of the resources required when using Cesium and avoid practices that may cause lag or reduce performance. Avoiding animations, using a complete package of frameworks like Bootstrap, loading all coordinates before rendering, and using classes and functions efficiently are some key practices to

keep in mind. By following these best practices, users can ensure that their Cesium projects run smoothly and provide an optimal 3D visualization experience.

4.1.1 Visualization result on Cesium platform Cesium is a powerful JavaScript library that provides an excellent platform for visualizing complex 3D models in a web application. Our energy plugin utilizes Cesium to allow users to seamlessly transition from QGIS to a web browser where they can easily explore the final results. The ability to integrate Cesium into the plugin has greatly enhanced the user experience, providing a more immersive and interactive approach to visualizing and analyzing the data. In the following section, we present some examples of 3D visualizations generated by the plugin. These visualizations allow users to interact with the 3D models in real-time, which can be useful for identifying patterns and trends that might not be immediately apparent in 2D data. Moreover, Cesium allows users to apply various visual styles, such as color gradients, opacity, and textures, to the 3D models. This provides an additional layer of information that can help users gain a deeper understanding of the data. Overall, Cesium has proven to be an invaluable tool for our energy plugin, allowing us to create dynamic and engaging visualizations that help users gain insights into complex datasets.

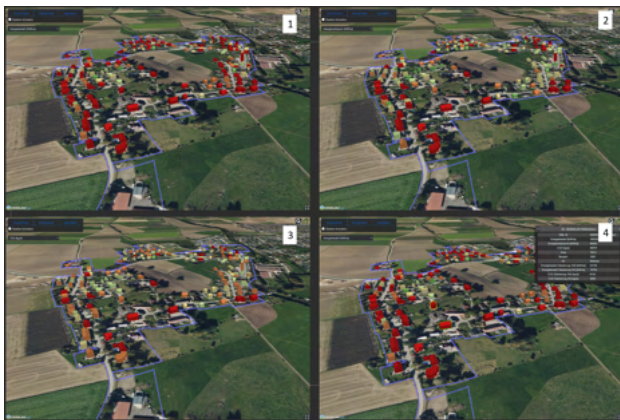


Figure 15. 3D visualization results of energy calculation. 1. Categorizing buildings based on energy demand, 2. energy consumption, 3. CO_2 emission. 4. Hovered buildings showed in yellow and selected building showed in Green. The selected building information appear in the right-hand side.

5. CONCLUSION

In conclusion, the development of the Energy QGIS plugin has aimed to simulate urban energy at the city district level using adaptable 2D and 3D data. By comparing the result of energy simulation with ALKIS data and CityGML data, the plugin's accuracy in handling 2D data was demonstrated. The plugin stands out as a completed version compared to others for QGIS, as it can not only retrieve data from the database but also update and modify information obtained from SimStadt or calculated by the plugin. The plugin's ability to connect QGIS to SimStadt software to retrieve the result of energy calculations is an important feature that provides urban planners and energy experts with valuable information for decision-making. By integrating 2D and 3D data on the database side, the plugin has addressed a paramount parameter of 3D data, such as GML id, building construction year, and building usage, even in cases of lack of information. This has allowed for more comprehensive and ac-

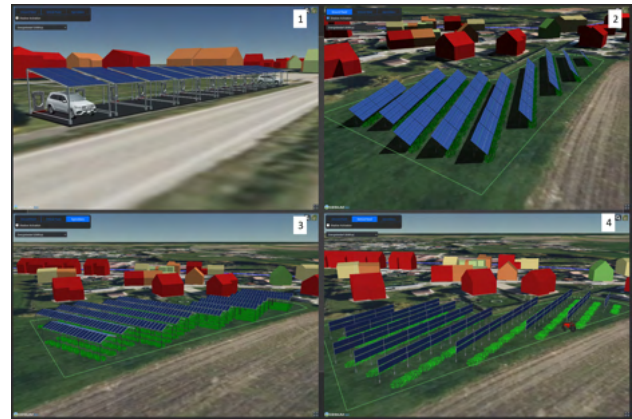


Figure 16. 3D visualization results of PV-Panels. 1. PV panels on the roof of open parking places. 2. Ground PV panels in the agricultural land and effect of shadow. 3. Agrivoltaic PV panels in the agricultural land. 4. Vertical PV panels in the agricultural land with agricultural vehicle

curate energy simulations, which are essential in achieving sustainable urban development.

Furthermore, the plugin's capability to visualize the installation of 3D photovoltaic panels for open areas, such as agri-voltaic panels and garage roofs, is an innovative feature that adds to the value of the plugin. It allows urban planners and energy experts to assess the feasibility of implementing such technologies in a given area and contributes to the development of sustainable and energy-efficient urban environments. The integration of the plugin with the Cesium platform for visualizing the 3D data is a significant development that opens up a wide range of possibilities for further improvements and modifications. Based on object-oriented programming, other developers can change this functionality for their purposes, contributing to the development of an even more comprehensive and versatile urban energy simulation tool. Finally, automating map/layout generation and data preparation through the plugin saves urban planners and energy experts a significant amount of time and effort, allowing them to focus on more critical aspects of their work.

In future work, there are several potential enhancements and expansions to consider for the Energy QGIS plugin. Firstly, automating the connection of 3D data to 2D data would streamline the data integration process and improve the overall efficiency of the plugin. This would enable seamless integration and synchronization between different data sources, ensuring a more comprehensive representation of the urban energy landscape. Additionally, enhancing the visualization capabilities of the plugin could be beneficial. This could involve further development of the integration with Cesium, exploring advanced visualization techniques, and providing more interactive and immersive experiences for users. Considering the implementation of the same application based on other platforms such as Lizmap, Django, and Cesium could broaden the accessibility and usability of the plugin. Lastly, incorporating feedback and suggestions from users and collaborating with experts in the field of urban energy simulation would contribute to the continuous improvement and refinement of the Energy QGIS plugin.

The Energy QGIS plugin represents a significant contribution to urban energy simulation and supports sustainable urban de-

velopment efforts. It provides a powerful tool for estimating energy demand, exploring building refurbishment scenarios, and visualizing results in an intuitive and informative manner. By integrating adaptable 2D and 3D data, automating data processing, and offering innovative visualization capabilities, the plugin empowers urban planners and energy experts to make informed decisions, optimize energy efficiency, and drive the transition towards sustainable urban environments.

REFERENCES

- BDEW, 2022. The usage of renewable energy in germany.
- Biljecki, F., Ledoux, H., Stoter, J., 2016. An improved LOD specification for 3D building models. *Computers, Environment and Urban Systems*, 59, 25–37.
- Bundesministerium fuer Umwelt Naturschutz nukleare Sicherheit, D., 2019. Klimaschutzprogramm 2030 der Bundesregierung zur Umsetzung des Klimaschutzplans 2050. URL: <https://www.bundesregierung.de/resource/blob/975226/1679914/e01d6bd855f09bf05cf7498e06d0a3ff/2019-10-09-klimamassnahmen-data.pdf>.
- Bundesregierung, D., 2021. Entwurf eines ersten gesetzes zur änderung des bundes-klimaschutzgesetzes.
- David, R., de Boer, J., Erhorn, H., 2022. *Heizen, Kühlen, Belüften und Beleuchten.: Bilanzierungsgrundlagen zur DIN V 18599*. Fraunhofer IRB Verlag.
- Dermentzis, G., Ochs, F., Thuer, A., Streicher, W., 2021. Supporting decision-making for heating and distribution systems in a new residential district-An Austrian case study. *Energy*, 224, 120141.
- Fumo, N., Biswas, M. R., 2015. Regression analysis for prediction of residential energy consumption. *Renewable and sustainable energy reviews*, 47, 332–343.
- KFW, 2022. Kfw grants for energy projects.
- Martirano, G., Pignatelli, F., Vinci, F., Struck, C., Coors, V., Fitzky, M., Hernández Moral, G., Serna-González, V., Ramos-Díez, I., Valmaseda, C., 2022. Comparative analysis of different methodologies and datasets for Energy Performance Labelling of buildings. *Luxembourg (Luxembourg): Publications Office of the European Union*.
- Reinhart, C. F., Davila, C. C., 2016. Urban building energy modeling—A review of a nascent field. *Building and Environment*, 97, 196–202.
- Singla, J. G., Padia, K., 2021. A novel approach for generation and visualization of virtual 3D city model using open source libraries. *Journal of the Indian Society of Remote Sensing*, 49(6), 1239–1244.
- Strzalka, A., Alam, N., Duminil, E., Coors, V., Eicker, U., 2012. Large scale integration of photovoltaics in cities. *Applied Energy*, 93, 413–421.
- Strzalka, A., Bogdahn, J., Coors, V., Eicker, U., 2011. 3D City modeling for urban scale heating energy demand forecasting. *HVAC&R Research*, 17(4), 526–539.
- Swan, L. G., Ugursal, V. I., 2009. Modeling of end-use energy consumption in the residential sector: A review of modeling techniques. *Renewable and sustainable energy reviews*, 13(8), 1819–1835.
- Wate, P., Coors, V., 2015. 3D data models for urban energy simulation. *Energy Procedia*, 78, 3372–3377.
- Weiler, V., Eicker, U., 2021. Automatic energy demand and system simulation at district level. *Sustainability Management Forum— NachhaltigkeitsManagementForum*, 29number 2, Springer, 133–141.