Semantic Surface Integration in CityJSON-Based Urban Digital Twins:

A Rooftop Photovoltaic Potential Use Case

Yassine Oudadda [1, 3], Imane Jeddoub [1], Mazarine Roquet [2], Rafika Hajji [3], Pierre Dewallef [2], Roland Billen [1]

[1] GeoScITY, Spheres Research Unit, University of Liège, 4000 Liège, Belgium – (i.jeddoub, rbillen)@uliege.be, yassine.oudadda@student.uliege.be

[2] University of Liège, Thermodynamics Laboratory, Allée de la Découverte 17 - 4000 Liège, Belgium – (mazarine.roquet, p.dewallef)@uliege.be

[3] College of Geomatic Sciences and Surveying Engineering, Hassan II Institute of Agronomy and Veterinary Medicine, Rabat 10101, Morocco – r.hajji@iav.ac.ma

Keywords: Urban Digital Twin, Rooftop Photovoltaic Potential, CityJSON, Semantic Surfaces, Data Integration, Web Visualization.

Abstract

Urban digital twins provide an integrated framework for combining semantic 3D city models, simulation results, and interactive visualization in support of urban energy analyses. While existing approaches effectively represent results at the building scale, extending this capability to individual semantic surfaces remains challenging. Applications such as rooftop solar assessment require simulation results to be associated with each roof surface and visualized at the same level of detail, yet current CityJSON visualization tools do not support attribute-driven rendering of surfaces belonging to the same semantic category. This paper addresses this limitation through two complementary contributions. First, we propose a client-side workflow that integrates solar potential simulations, computed with pvlib and provided as lightweight JSON files, into individual roof surfaces of a CityJSON model. The workflow enriches semantic surfaces with verified energy attributes and enables interactive analysis from the district scale down to individual roof surfaces. Second, we extend the open-source CityJSON-ThreeJS loader with a reusable class-based visualization mechanism that renders semantic surfaces according to user-defined classification attributes. Although demonstrated using solar potential classes, the approach is generic and can support other surface-level applications, including thermal performance, material characterization, and structural assessment. The approach is demonstrated on 1,667 buildings in the Sart-Tilman district (Liège, Belgium) within an in-house urban digital twin platform. The results demonstrate that detailed surface-level analysis can be achieved entirely in the browser without server-side processing or format conversion, highlighting the potential of lightweight, standards-compliant urban digital twins.

1. Introduction

Buildings account for approximately one third of global final energy consumption and remain a major contributor to greenhouse gas emissions, making their decarbonisation a central objective of urban climate strategies. Supporting this transition requires digital tools capable of assessing both building energy demand and renewable energy opportunities across large urban areas while presenting the results in a form that supports planners and decision-makers rather than simulation specialists (Hong et al., 2020; Reinhart and Cerezo Davila, 2016). Urban Digital Twins (UDTs) have emerged as a promising framework for this purpose by combining semantic 3D city models with simulation, visualization, and analytical capabilities within an interactive environment.

Recent advances in semantic 3D city models have further strengthened the role of UDTs in urban energy applications. Among the available standards, CityJSON provides a lightweight and web-oriented encoding of the CityGML data model that enables semantic city models to be parsed, enriched, and visualized directly in modern web browsers (Ledoux et al., 2019). This lightweight architecture has facilitated the development of client-side UDT platforms capable of integrating heterogeneous urban datasets without relying on complex server-side infrastructures (Rafamatanantsoa et al., 2024).

Building on this approach, Jeddoub et al. (2025) demonstrated that Building Energy Simulation (BES) results can be integrated into a CityJSON-based UDT through a lightweight client-side workflow, where JSON-formatted heating demand results are directly associated with building geometries and visualized entirely within the browser. While this work established the feasibility of browser-based integration for building-level energy analyses, it also highlighted the need to extend this capability to applications requiring finer semantic granularity, such as rooftop photovoltaic potential assessment.

Unlike heating demand, solar potential is inherently a surface-level phenomenon. The photovoltaic yield of a roof depends on the orientation, inclination, area, and local shading conditions of each individual roof surface; consequently, meaningful district-scale analyses require simulation results to be associated with semantic roof surfaces rather than with buildings as a whole (Coors and Padsala, 2024; Freitas et al., 2015). Extending lightweight UDT workflows to this level introduces two closely related challenges. First, simulation outputs must be accurately mapped to the corresponding semantic surfaces of the 3D city model. Second, these enriched surfaces must be visualized according to their computed attributes, allowing users to explore performance variations directly within the urban model. Although CityJSON already supports attributes on semantic surfaces, existing web visualization tools do not readily provide attribute-driven rendering at this level, limiting the practical use of surface-based analyses within lightweight UDTs.

This paper addresses these challenges while preserving the lightweight, browser-based philosophy of previous work. It makes two primary contributions. First, it proposes a client-side workflow that integrates rooftop photovoltaic potential simulations computed with pvlib into individual CityJSON roof surfaces through sequential surface mapping, complemented by a verification procedure based on the geometric signature (orientation, inclination, and area) of each surface to ensure reliable correspondence between simulation outputs and

semantic geometries. Second, it extends the open-source CityJSON-ThreeJS loader[1] with a reusable class-based visualization mechanism that enables attribute-driven rendering of semantic surfaces while maintaining backward compatibility with existing applications. Although demonstrated through a solar potential use case, the proposed visualization mechanism is generic and can support any semantic surface classification, including thermal performance, material characterization, and structural assessment.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 presents the proposed methodology. Section 4 reports the experimental results and evaluates the proposed workflow. Section 5 discusses the findings, limitations, and future research directions. Finally, Section 6 concludes the paper.

2. Related Work

2.1 Solar Potential Assessment using Semantic 3D City Models

Assessing the rooftop photovoltaic potential of buildings has become a key component of urban energy planning, supporting renewable energy deployment and local energy transition strategies (Freitas et al., 2015). Early urban-scale approaches were predominantly raster-based, relying on Digital Surface Models (DSMs) to estimate solar irradiation through GIS tools such as ArcGIS Solar Analyst (Fu and Rich, 1999) and r.sun in GRASS GIS. These methods remain widely used because they are robust, relatively simple to implement, and computationally efficient. However, raster representations provide limited geometric detail for complex roof structures and do not explicitly preserve building semantics, which restricts their ability to support fine-grained urban solar analysis.

The increasing availability of semantic 3D city models has shifted research towards vector-based approaches capable of exploiting detailed building geometries, computing solar radiation directly on CityGML building data across different levels of detail (Wieland et al., 2015). More recent developments have focused on improving computational efficiency through acceleration structures such as bounding volume hierarchies, enabling city-scale analyses at high temporal resolution (Xu et al., 2024). Comparative evaluations nevertheless show that solar simulation tools can diverge considerably on identical inputs, underlining that the choice of tool and its calibration remain application-dependent (Giannelli et al., 2022; León-Sánchez et al., 2025).

Although these studies have considerably advanced the computation of urban rooftop photovoltaic potential, their primary focus remains on the estimation process itself. Comparatively little attention has been paid to how the resulting surface-level information can be integrated into interactive Urban Digital Twins and effectively communicated to end users.

2.2 Integration and Visualization of Solar Potential Results

Beyond the computation of solar potential, an important part of the workflow is how the resulting values are integrated back into semantic 3D city models and visualized for analysis and decision support. In CityGML, four integration mechanisms are reported in the literature. The most common one stores the results as generic attributes on buildings or boundary surfaces (Biljecki et al., 2015). This approach is simple and supported by most tools, but generic attributes cannot be stored in a systematic and semantically standardized way (Agugiaro and Padsala, 2025). The second mechanism uses Application Domain Extensions (ADEs), which extend the standard with energy-related concepts for representing simulation outputs and linking them to the original city model (Agugiaro et al., 2018). Recent work proposed solar-specific classes that link panel installation, shading, and analysis results to building components (Casisirano and Claridades, 2026), and the new version of the Energy ADE, released as Energy ADE 3.0, introduces a Resources module that represents actual and potential energy production of a city object and defines zone boundaries through thematic surfaces (Agugiaro and Padsala, 2025). The third mechanism, Dynamizers, links time-dependent simulation results to city objects (Chaturvedi et al., 2017). The fourth maps the results onto textures for display on building surfaces. These developments confirm the relevance of representing solar potential not only as numerical output, but as model-integrated information that remains associated with roofs, facades, and other semantic building elements.

In practice, several of these mechanisms are combined. The Munich solar potential analysis published textured CityGML models, per-point CSV samples keyed to the gml_id of each thematic surface, and glTF visualization models, showing that a single simulation can be expressed through complementary result formats (Willenborg et al., 2026). The urban digital twin of Sofia integrates street and tree models through CityGML and 3DCityDB, with solar radiation maps served alongside 3D Tiles (Shirinyan et al., 2025). In CityJSON, similar enrichment is possible: an energy Extension was tested for space heating demand calculation (Tufan et al., 2022), and Jeddoub et al. (2025) showed that simulation outputs can be integrated client-side into a CityJSON-based urban digital twin at the building level. However, no standardized extension exists yet for solar indicators, and simulation outputs must be linked to semantic surfaces through ordering conventions, because these surfaces do not carry persistent identifiers of their own: unlike city objects, semantic surfaces in CityJSON are addressed only by their position in the geometry arrays.

The visualization of these results follows a similar pattern. Common approaches include color-coded roof and facade textures, surface-based thematic rendering, and browser-based 3D viewers that display solar information directly on city models. In CityGML-based workflows, results are often visualized through textured city models or web map clients (Yao et al., 2018), while CityJSON-based environments rely on browser-oriented viewers such as ninja and the CityJSON-ThreeJS loader. The CityJSON-ThreeJS loader supports thematic coloring at the level of city object types and semantic surface categories, for example rendering all roofs with one color and all walls with another, but it does not differentiate surfaces of the same semantic class based on attributes stored in their definition, such as solar potential classes. The same limitation was reported for 3D Tiles and glTF, where coloring individual roof polygons required direct modification of KML files (Kolk, 2023). As a result, solar values attached to individual semantic surfaces are not yet fully exploited in current lightweight web viewers, which limits the direct visualization of fine-grained solar potential results in Urban Digital Twins.

This literature shows that solar potential integration is no longer limited to computing irradiation values, but also includes the challenge of storing, structuring, and visualizing these results

[1] https://github.com/cityjson/cityjson-threejs-loader

within semantic 3D city models. While CityGML provides formal mechanisms through ADEs, current implementations still tend to separate simulation, enrichment, and visualization. This motivates client-side workflows that combine simulation, data integration, and attribute-driven rendering without format conversion, integrating solar potential values directly into semantic surfaces and making them accessible in browser-based environments while preserving interoperability and analytical

detail (Jeddoub et al., 2026).

2.3 Web Visualization of CityJSON Semantic Surfaces

The CityJSON-ThreeJS loader parses CityJSON data into Three.js scenes through a five-stage pipeline of parsing, geometry processing, material configuration, shader compilation, and rendering, and is the basis of the reference web viewer ninja (Vitalis et al., 2020). Its coloring model operates at two levels. Object-level coloring distinguishes city object types, while semantic-level coloring distinguishes surface categories, assigning for example one color to all roofs. Both levels are categorical: neither can consult an attribute stored inside a semantic surface definition. Consequently, a CityJSON file in which every roof surface carries a photovoltaic potential class can be loaded and displayed but not rendered according to that class. To fill this gap, we propose an attribute-driven coloring mechanism. Such a coloring level is a prerequisite for surface-level energy visualization and is addressed in Section 3.4.

3. Methodology

The proposed methodology, illustrated in Figure 1, comprises four stages: preparation and enrichment of the semantic CityJSON model of the study area; per-surface photovoltaic simulation with pvlib, provided as plain JSON files; client-side integration of the results into individual roof surfaces, including geometric verification of the surface correspondence; and surface-level visualization and analysis in City2Twin (Rafamatanantsoa et al., 2024), enabled by the class-based extension of the CityJSON-ThreeJS loader.

3.1 Study Area and Data Preparation

The study area covers the Sart-Tilman campus and scientific park near Liège, Belgium, and comprises 1,667 buildings whose roofs span flat, sloped, and mixed configurations with diverse orientations and areas, making the district a demanding test case for per-surface analysis. The 3D city model is reconstructed with **Roofer**[2] from 2D building footprints and classified LiDAR point clouds, producing LoD2.2 geometries with semantic surfaces. An FME pipeline then attaches to every semantic surface the geometric attributes on which the photovoltaic simulation depends: azimuth, tilt, and surface area. Each building carries a unique identifier that acts as the primary key between the simulation outputs and the city model, and the final file is checked against the CityJSON 2.0 specifications with the official cjval[3] validator.

3.2 Rooftop Photovoltaic Potential Simulation

For every building, pvlib[4] computes the annual photovoltaic electricity production potential (kWh/year) of each individual roof surface from the total solar irradiance incident on the roof, accounting for its orientation, tilt, and surface area (Anderson et al., 2023). The total irradiance is computed using pvlib.irradiance.get_total_irradiance, which combines the direct, diffuse, and ground-reflected components of solar radiation. It is then converted into annual electricity generation by applying the effectively available roof area and a fixed photovoltaic module efficiency. The results are delivered as a JSON file with one entry per building, identified by its unique identifier. Each entry contains two elements (refer to Figure 2): a solar_potential object whose keys roof_1, roof_2, and so on hold the annual yield of each roof surface, and a roof_surfaces array that replicates, for each of those surfaces, the azimuth, tilt, and surface area used by the simulation. The surfaces are listed in the same order as the RoofSurface order of the corresponding building in the CityJSON model. This ordering, together with the replicated geometric properties, forms the contract on which the integration of Section 3.3 relies (e.g., the sequence establishes the correspondence, and the geometry makes that correspondence verifiable).

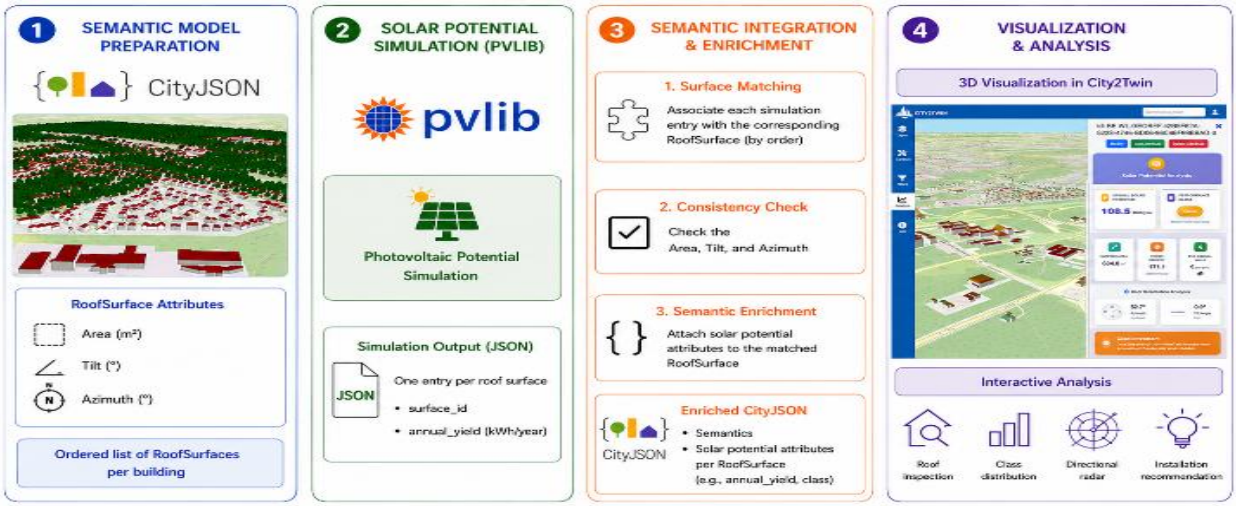

Figure 1.Overview of the proposed framework for surface-level rooftop photovoltaic potential integration, semantic enrichment, and visualization in CityJSON-based Urban Digital Twins.

[2] https://github.com/3DBAG/roofer
[3] https://github.com/cityjson/cjval
[4] https://github.com/pvlib/pvlib-python

```
{
  "BE.WL.GEOREF.006196EF-660A-4125-B497-E51992AEB711-0": {
    "roof_surfaces": [
      {
        "azimuth": 358.8055419921875,
        "tilt": 29.62717628479004,
        "surface area": 19.53037954087466
      },
      {
        "azimuth": 177.4617919921875,
        "tilt": 28.65818214416504,
        "surface_area": 0.06822179282501001
      },
      {
        "azimuth": 177.4644317626953,
        "tilt": 28.65947151184082,
        "surface area": 22.79459033760257
      },
      {
        "azimuth": 294.4830322265625,
        "tilt": 0.4201137721538544,
        "surface_area": 54.88206869786136
      },
      {
        "azimuth": 274.9624938964844,
        "tilt": 21.34937858581543,
        "surface area": 21.887731255416206
      }
    ],
    "solar_potential (kWh/an)": {
      "roof_1": 1960.3957830754496,
      "roof_2": 10.140884370851374,
      "roof_3": 3388.322492126325,
      "roof_4": 7293.811932361332,
      "roof_5": 2743.2507830664595
    }
  },
```


Figure 2. Extract of the simulation output JSON file: per-building entries identified by unique identifier, with sequentially keyed photovoltaic potential values and the replicated geometric properties of each roof surface for roof-surface verification.

To assess the consistency of the photovoltaic production estimates obtained with pvlib, the simulated annual electricity yields are compared with those provided by PVGIS (Huld et al., 2012), a freely accessible web application developed by the European Commission's Joint Research Centre (JRC) for photovoltaic performance assessment. The comparison is performed for four representative roof surfaces with different orientations, approximately facing north, south, east, and west. Both models use the same geographic location, roof surface area, tilt, azimuth, and photovoltaic module efficiency; the irradiance data, however, differ by construction, as PVGIS relies on its own satellite-based radiation database. The resulting annual production estimates are presented in Figure 3. Despite the different irradiance sources, the two approaches agree within 10% for all tested orientations, indicating that the pvlib-derived per-surface estimates are consistent with those obtained from an established reference model. This comparison provides confidence in the simulation results but does not constitute formal validation, which would require comparison against measured photovoltaic production and lies outside the scope of this study (refer to Section 5).

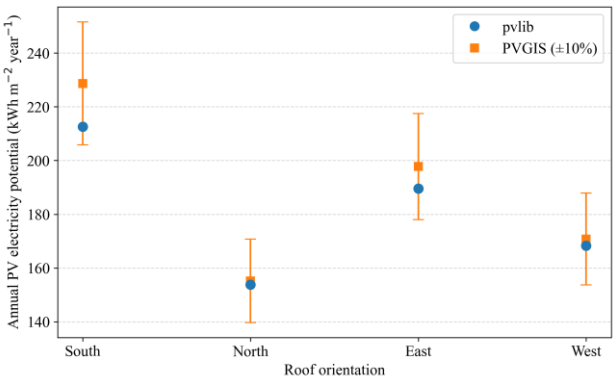

Figure 3. Comparison of annual photovoltaic electricity production estimated by pvlib and PVGIS for four main roof orientations.

3.3 Roof-Surface Data Integration and Consistency Check

The integration runs entirely in the browser and consists of two steps. In the first step, the system scans the semantic surfaces of each building geometry, collects the indices of all surfaces typed RoofSurface, and sorts them to obtain a deterministic ordering. The sequential entries of the simulation output are then bound one-to-one to this ordering: the first sorted RoofSurface receives the value of roof_1, the second that of roof_2, and so forth. Each surface is enriched with its annual photovoltaic potential as an additional semantic attribute, alongside the azimuth, tilt, and area supplied by the simulation.

Positional mapping alone is not robust: any divergence between the surface order in the simulation and in the model would assign values to incorrect surfaces without producing a detectable error. To prevent this, the azimuth, tilt, and area reported for each simulation entry are verified against the attributes of the assigned surface, and the value is assigned only when these geometric properties are consistent.

Once mapped, each roof surface receives a class attribute derived from its photovoltaic potential value. Class thresholds were selected to provide a representative distribution of photovoltaic potential values across the study area and to facilitate thematic visualization. The enriched model remains a valid CityJSON file and can be exported for reuse in third-party software.

3.4 Visualization of Semantic Surface Attributes: Extending the CityJSON-ThreeJS Loader

Rendering each roof surface by its performance class requires the attribute-based coloring level identified in Section 2.3. In the loader, the color of the surface is not decided at a single point. It is progressively determined along with the rendering pipeline, from the data structures built during parsing to the program executed on the graphics card. Introducing a new coloring criterion therefore requires the class information to propagate through the entire pipeline. The proposed extension, referred to as the showClass feature, achieves this through four coordinated modifications (refer to Figure 4), each answering a specific need.

The first need is storage: since the graphics card colors geometry vertex by vertex, the class of a surface must be available at the vertex level. The data structures that accumulate vertex information during parsing are therefore extended so that every

vertex records, next to its building identifier and surface type, the class of the surface it belongs to.

The second need is extraction: the class initially exists only as an attribute inside the CityJSON semantic surface definition. The parser is extended to read this attribute while constructing the geometry and to translate it into a color, assigning new colors automatically when previously unseen classes are encountered. This makes the mechanism independent of any predefined classification (e.g., six performance classes, ten material types, or any other scheme are handled without configuration).

The third need is arbitration: further consideration concerns the visualization of roof surfaces that possess both a semantic type and a photovoltaic potential class. While the original CityJSON-ThreeJS renderer assigns colors according to semantic type (e.g., all roof surfaces are displayed in red), the proposed extension allows class-based colors to override the default semantic colors when the visualization mode is enabled. Consequently, roof surfaces are rendered according to their photovoltaic potential class, whereas all other surfaces retain the original semantic coloring. When the class-based visualization is disabled, the rendering behavior remains identical to that of the original CityJSON-ThreeJS implementation.

The fourth need is control: to support interactive exploration, the visualization can be switched dynamically between the original semantic rendering and the proposed class-based rendering. This functionality allows users to alternate between viewing semantic surface types and photovoltaic potential classes without reloading the CityJSON model, facilitating comparative analysis during an interactive session (see Section 4).

The proposed functionality is integrated into the existing visualization workflow with minimal changes to the original library. When photovoltaic class attributes are available, they can be visualized through the class-based rendering mode; otherwise, the standard semantic rendering is preserved, ensuring full backward compatibility.

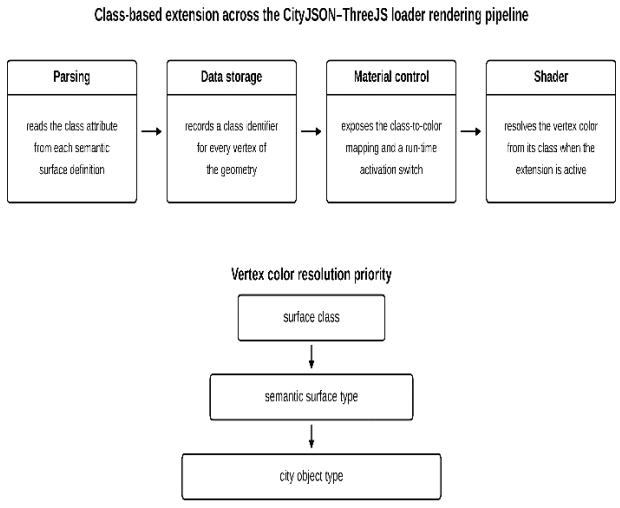

Figure 4. Architecture of the showClass extension.

3.5 Analytical Tools

Three analytical tools accompany the visualization and operate on the enriched model. A distribution chart counts roof surfaces per performance class, giving a portfolio-wide statistical overview. A directional radar chart aggregates performance by orientation: the azimuth of each surface, available from the enrichment, assigns it to one of eight 45-degree compass sectors, and the average photovoltaic potential and surface count are computed per sector. A click-based selection mode, finally, reports the metrics of an individual roof surface, including its annual yield in MWh per year, performance class, area, power density in kWh/m²/year, azimuth and tilt, together with an indicative installation priority based on the same thresholds, intended as a first screening aid rather than an investment recommendation.

4. Results and Analysis

4.1 Surface-Level Photovoltaic Potential Visualization

The workflow processes all 1,667 buildings of the study area. After integration, the geometric verification of Section 3.3 confirms the correspondence: all roof surfaces are matched, with no geometric mismatch and no surface left unmapped. Activating the class-based visualization renders every roof surface according to its performance class, on a six-class scale from very low to excellent potential (refer to Figure 5). High-yield surfaces become identifiable at a glance across the district, and a dedicated checkbox switches between the photovoltaic rendering and the standard semantic rendering interactively, without reloading or reprocessing the model.

Figure 5. Surface-level photovoltaic potential visualization of the Sart-Tilman district in City2Twin: individual roof surfaces colored by performance class and per-surface assessment panel: annual yield, performance class, area, power density, orientation, and automated installation recommendation for a selected roof surface.

4.2 Analytical Assessment

The distribution chart in Figure 6 quantifies how the roof surfaces of the district spread across the six performance classes, and the directional radar chart summarizes orientation-dependent behavior: south-facing surfaces reach the highest average potential, followed by the southeast and southwest sectors, while north-facing surfaces remain lowest. The agreement of this pattern with physical expectation provides a useful sanity check of the end-to-end chain from simulation to rendering. At the level of a single surface, the selection mode assembles the per-surface metrics and the automated installation recommendation into one panel (see Figure 5), supporting site-specific feasibility reasoning. The evaluation here concerns the workflow, not the simulation. The figures show that per-surface results are correctly mapped, classified, rendered, and queried within the platform, while the accuracy of the simulation model itself lies outside the scope of this study and is discussed in Section 5.

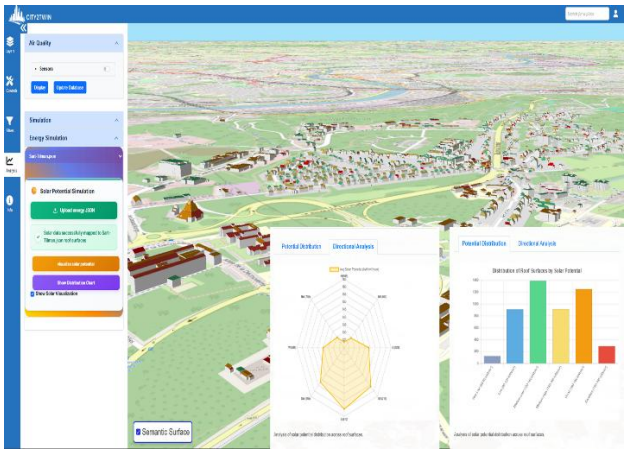

Figure 6. City2Twin interface for roof-surface photovoltaic potential analysis, combining semantic surface visualization with the class distribution and directional radar charts.

4.3 Beyond Photovoltaic Applications

To test the extension outside the energy domain, random class attributes were assigned to all semantic surfaces of the Sart-Tilman dataset, including walls and ground surfaces. Loaded in City2Twin, the model renders each surface by its class, emulating a material characterization scenario in which construction materials drive the classification (see Figure 7). This test confirms two properties. First, the class attribute is domain-agnostic: nothing in the attribute or in the loader refers to photovoltaic energy. Second, the implementation is self-contained: no modification was needed beyond enabling the feature, since the classification logic and the rendering live inside the library. Any application that assigns numeric classes to semantic surfaces, for example for thermal performance or material characterization, can reuse the mechanism directly.

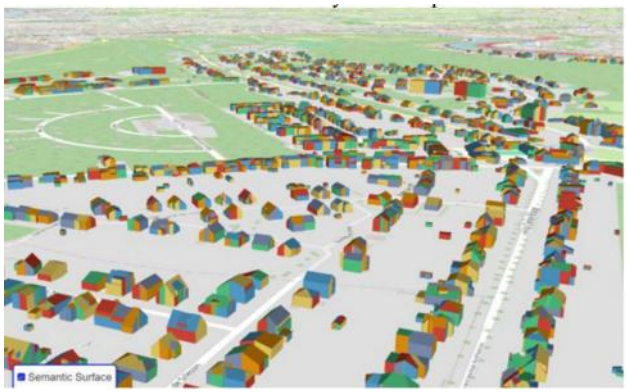

Figure 7. Class-based rendering applied to a building material characterization scenario on the Sart-Tilman dataset, using randomly assigned class attributes on all semantic surfaces.

5. Discussion

The design principle running through this work is the minimality of the contract between the simulation and the platform: a shared building identifier, an agreed surface ordering, and a replicated geometric signature. Nothing else is required, no schema extension, no conversion step, no server. This is what makes the workflow transferable: any party holding a CityJSON model with semantic surfaces and per-surface results in plain JSON can reproduce the integration, the visualization, and the analytics in a browser. Taken together with the building-level workflow of Jeddoub et al. (2025), City2Twin covers client-side energy analysis at both granularities of the CityJSON building model, and the same integration logic transfers to any other building-level or surface-level quantity. Formal enrichment mechanisms such as the Energy ADE (Agugiaro et al., 2018) or the prototype CityJSON energy extension (Tufan et al., 2022) offer semantic completeness at the price of tooling and expertise; a single numeric attribute paired with a rendering capability offers immediacy at the price of semantic depth. The two are complementary rather than competing, and a natural evolution is to generate standardized extensions automatically from the lightweight inputs, retaining the accessibility of the current workflow while gaining schema-based validation and interoperability.

The main limitation of this work lies in the separation between the simulation engine and the visualization environment: the simulation results are precomputed externally and loaded into the platform as JSON files. A natural next step would be to run the simulation directly within the platform, on demand, removing the need to import a separate file and turning the tool from a viewer of precomputed scenarios into an interactive analysis environment. The mapping between the JSON output and the CityJSON surfaces, by contrast, proved reliable: across the whole dataset, the verification of Section 3.3 produced no mismatch and left no surface unmapped, confirming that the geometric check is sufficient to secure the correspondence in practice.

Regarding accuracy, this study does not benchmark the photovoltaic estimates against measured production. The contribution under evaluation is the integration and visualization chain, and comparative work has shown that solar simulation tools already diverge among themselves on identical inputs (Giannelli et al., 2022; León-Sánchez et al., 2025), so validating a given engine is a study in its own right. The same position was taken for the heating demand workflow (Jeddoub et al., 2025), and calibration against measured campus data remains a direction for future work.

Finally, the workflow was applied at the district scale. This is a deliberate choice rather than a constraint: surface-level photovoltaic potential is a granular quantity whose visual reading is meaningful precisely at the zoom level of a district, where individual roof surfaces can be distinguished and interpreted. Scalability is therefore not a concern for this use case. The performance classes, however, were derived from the statistical distribution of this dataset, and normative thresholds would be required in a regulatory context.

6. Conclusion and Perspectives

This paper extended a CityJSON-based energy UDT framework from the building level to the semantic surface level. Per-surface photovoltaic potential computed with the pvlib Python library was integrated into the individual roof surfaces of the CityJSON model and rendered through a class-based coloring capability added to the open-source CityJSON-ThreeJS loader. Demonstrated using the 1,667 buildings of the Sart-Tilman district in Belgium within City2Twin, and shown to generalize to a material characterization scenario, the approach establishes that surface-level energy analysis is achievable entirely on the client side, without format conversion or server-side infrastructure. The loader extension and the class attribute convention are reusable by the CityJSON community for any application requiring attribute-driven surface visualization.

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* 3, 2. https://doi.org/10.1186/s40965-018-0042-y

Agugiaro, G., Padsala, R., 2025. A proposal to update and enhance the CityGML Energy Application Domain Extension. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* X-4-W6-2025, 1–8. https://doi.org/10.5194/isprs-annals-X-4-W6-2025-1-2025

Anderson, K.S., Hansen, C.W., Holmgren, W.F., Jensen, A.R., Mikofski, M.A., Driesse, A., 2023. pvlib python: 2023 project update. *Journal of Open Source Software* 8, 5994. https://doi.org/10.21105/joss.05994

Biljecki, F., Heuvelink, G.B.M., Ledoux, H., Stoter, J., 2015. Propagation of positional error in 3D GIS: estimation of the solar irradiation of building roofs. *International Journal of Geographical Information Science* 29, 2269–2294. https://doi.org/10.1080/13658816.2015.1073292

Casisirano, J.D.D., Claridades, A.R.C., 2026. Extending CityGML for urban solar potential estimation: a semantically enriched model informed by UAV-derived analysis. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* X-5-W4-2025, 151–158. https://doi.org/10.5194/isprs-annals-X-5-W4-2025-151-2026

Chaturvedi, K., Willenborg, B., Sindram, M., Kolbe, T.H., 2017. Solar potential analysis and integration of the time-dependent simulation results for semantic 3D city models using Dynamizers. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* IV-4/W5, 25–32. https://doi.org/10.5194/isprs-annals-IV-4-W5-25-2017

Coors, V., Padsala, R., 2024. Urban digital twins empowering energy transition: citizen-driven sustainable urban transformation towards positive energy districts. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLVIII-4-W10-2024, 51–56. https://doi.org/10.5194/isprs-archives-XLVIII-4-W10-2024-51-2024

Freitas, S., Catita, C., Redweik, P., Brito, M.C., 2015. Modelling solar potential in the urban environment: state-of-the-art review. *Renewable and Sustainable Energy Reviews* 41, 915–931. https://doi.org/10.1016/j.rser.2014.08.060

Fu, P., Rich, P., 1999. Design and implementation of the Solar Analyst: an ArcView extension for modeling solar radiation at landscape scales. *Proceedings of the 19th Annual ESRI User Conference*.

Giannelli, D., León-Sánchez, C., Agugiaro, G., 2022. Comparison and evaluation of different GIS software tools to estimate solar irradiation. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* V-4–2022, 275–282. https://doi.org/10.5194/isprs-annals-V-4-2022-275-2022

Hong, T., Chen, Y., Luo, X., Luo, N., Lee, S.H., 2020. Ten questions on urban building energy modeling. *Building and Environment* 168, 106508. https://doi.org/10.1016/j.buildenv.2019.106508

Huld, T., Müller, R., Gambardella, A., 2012. A new solar radiation database for estimating PV performance in Europe and Africa. *Solar Energy* 86, 1803–1815. https://doi.org/10.1016/j.solener.2012.03.006

Jeddoub, I., Kasprzyk, J.-P., Billen, R., 2026. Towards flexible and lightweight solutions for Urban Digital Twins. *Bulletin de la Société Géographique de Liège* 85, 1–23. https://doi.org/10.25518/0770-7576.7657

Jeddoub, I., Roquet, M., Oudadda, Y., Hajji, R., Dewallef, P., Billen, R., 2025. A lightweight framework for seamless integration of building energy simulations into urban digital twins. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XLVIII-4/W15-2025, 77–84. https://doi.org/10.5194/isprs-archives-XLVIII-4-W15-2025-77-2025

Kolk, A., 2023. Visualization of rooftop solar potential in smart cities. Tartu Ülikool.

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, 4. https://doi.org/10.1186/s40965-019-0064-0

León-Sánchez, C., Giannelli, D., Agugiaro, G., Stoter, J., 2025. Comparative analysis of geospatial tools for solar simulation. *Transactions in GIS* 29, e13296. https://doi.org/10.1111/tgis.13296

Rafamatanantsoa, B.P., Jeddoub, I., Yarroudh, A., Hajji, R., Billen, R., 2024. City2Twin: an open urban digital twin from data integration to visualization and analysis. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XLVIII-2/W8-2024, 387–394. https://doi.org/10.5194/isprs-archives-XLVIII-2-W8-2024-387-2024

Reinhart, C.F., Cerezo Davila, C., 2016. Urban building energy modeling – a review of a nascent field. *Building and Environment* 97, 196–202. https://doi.org/10.1016/j.buildenv.2015.12.001

Shirinyan, E., Stefanov, K., Petrova-Antonova, D., 2025. 3D solar analysis of the street network and impact estimation of the shade of trees at the district scale. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLVIII-G-2025, 1339–1346. https://doi.org/10.5194/isprs-archives-XLVIII-G-2025-1339-2025

Tufan, Ö., Arroyo Ohori, K., León-Sánchez, C., Agugiaro, G., Stoter, J., 2022. Development and testing of the CityJSON energy extension for space heating demand calculation. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLVIII-4-W4-2022, 169–176. https://doi.org/10.5194/isprs-archives-XLVIII-4-W4-2022-169-2022

Vitalis, S., Labetski, A., Boersma, F., Dahle, F., Li, X., Arroyo Ohori, K., Ledoux, H., Stoter, J., 2020. CityJSON + Web = Ninja. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* VI-4-W1-2020, 167–173. https://doi.org/10.5194/isprs-annals-VI-4-W1-2020-167-2020

Wieland, M., Nichersu, A., Murshed, S.M., Wendel, J., 2015. Computing solar radiation on CityGML building data.

Willenborg, B., Münzinger, M., Behnisch, M., Kolbe, T.H., 2026. 3D solar potential analysis results for Munich based on CityGML semantic city models. https://doi.org/10.5281/zenodo.19192862

Xu, L., León Sánchez, C., Agugiaro, G., Stoter, J., 2024. High resolution solar potential computation in large scale urban areas by means of semantic 3D city models. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLVIII-4/W11-2024, 167–174. https://doi.org/10.5194/isprs-archives-XLVIII-4-W11-2024-167-2024

Yao, Z., Nagel, C., Kunde, F., Hudra, G., Willkomm, P., Donaubauer, A., Adolphi, T., 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, 5. https://doi.org/10.1186/s40965-018-0046-7