ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume X-4/W6-2025
20th 3D GeoInfo Conference 2025, 2–5 September 2025, Kashiwa, Japan

From point clouds to CityGML 3.0: An approach to multi-granular urban road modelling

Elisavet Tsiranidou[1], Giorgio Agugiaro[2], Antonio Fernández[1], Lucía Díaz Vilariño[1]

[1]CINTECX, Universidade de Vigo, GeoTECH group, 36310 Vigo, Spain – {elisavet.tsiranidou, antfdez, lucia}@uvigo.gal
[2]3D Geoinformation group, Department of Urbanism, Faculty of Architecture and Built Environment, Delft University of Technology, Delft, The Netherlands – g.agugiaro@tudelft.nl

Keywords: Point cloud processing, 3D city models, Mobile laser scanning, Semantic integration, Urban space

Abstract

Accurate semantic modelling of urban road infrastructure is critical for digital twins, traffic simulations, and smart city planning. This study presents a structured methodology to transform road elements segmented from urban point clouds into CityGML 3.0-compliant representations. Leveraging CityGML's hierarchical Transportation module, the approach introduces a multi-level granularity framework—*area*, *way*, and *lane*—for representing road components like sidewalks, driving lanes, and parking areas. Following geometric pre-processing, segmented surfaces are semantically mapped into appropriate CityGML classes using a rule-based mapping strategy, enriched with descriptive attributes and hierarchical identifiers. The resulting XML-based datasets were validated and visualized using industry-standard tools such as FME, QGIS, and 3DCityDB, demonstrating successful integration into city-scale digital environments.

1. Introduction

Accurate and semantically rich representations of road infrastructure are fundamental for a wide range of applications, including smart mobility, transportation planning, digital twin development, and autonomous vehicle navigation. As cities continue to embrace digitalization, there is an increasing demand for structured, semantically enriched and high-resolution 3D models of the urban environment. In this context, the integration of road elements into 3D city models is a crucial step in facilitating intelligent systems to interact with the urban space in meaningful ways. Such integration supports not only static visualization and mapping but also dynamic analysis, simulation, and monitoring of traffic flow, accessibility, and infrastructure condition. Thus, enhancing the semantic and spatial quality of road representations is essential for building more efficient urban systems.

Recent advancements in mobile laser scanning (MLS) and point cloud processing have enabled the automated segmentation of urban road networks into detailed functional spaces. These include sidewalks, driving lanes, and parking areas captured with high accuracy and classified based on geometric and semantic properties. The high spatial resolution and coverage of MLS systems make them particularly well-suited for capturing the complexity of urban street environments, including variations in surface material, and elevation changes. However, while advances have been made in the detection and segmentation of urban features from MLS data, fewer efforts have focused on integrating this information into standardized urban data models. Consequently, the full potential of point cloud data is not yet much explored in applications that rely on semantic interoperability, such as traffic simulations, urban planning tools, or city-scale digital twins.

To ensure the effective use of 3D urban data, semantic modelling standards such as CityGML have been introduced. CityGML is an open standard developed by the Open Geospatial Consortium (OGC) for the representation and exchange of 3D virtual city models. It provides a consistent framework for encoding geometric, topological, and semantic information about the built environment, enabling the structured representation of urban features such as buildings, terrain, vegetation, and transportation infrastructure. The currently most widely used version is CityGML 2.0, released in 2012 (Gröger & Plümer, 2012), whereas significant revisions and extensions to the standard have been made with the release of CityGML 3.0 (Kolbe et al., 2021), particularly in its Transportation module, as the new version introduces a revised structure for representing transportation infrastructure (Beil et al., 2020).

Despite these improvements, the transformation of segmented point clouds into valid CityGML objects remains a challenging task. While point cloud data provides highly accurate geometric information, it lacks inherent structure and semantic meaning. Therefore, a major challenge lies in converting this raw geometric data into a semantically rich format that aligns with the hierarchical structure and class definitions of CityGML 3.0. This process requires to automate not only the accurate delineation of individual road components, but also the assignment of appropriate semantic attributes and spatial hierarchies between objects. To this end, it is essential to translate segmented road elements from point clouds into a structured, rule-based format that respects the structure and constraints of the CityGML schema. This includes the correct mapping to CityGML classes and attributes, and the modelling of the same feature according to different levels of detail.

This study focuses on the semantic integration of automatically extracted polygons from point clouds into CityGML 3.0. The proposed methodology involves the integration of already segmented urban road components—such as sections, intersections, sidewalks, parking areas, and driving lanes—across three levels of granularity. These elements are transformed into valid CityGML objects, compliant with the CityGML 3.0 transportation module and ready for integration into city models and digital twin environments.

The main contributions of this work are:

- A semantic integration framework for transforming road polygons, derived from MLS point cloud data, into valid CityGML 3.0 objects. The framework ensures compliance with the Transportation module's structure and supports its hierarchical representation of road spaces.

- Support for multi-level granularity, enabling the representation of transportation objects at the *area*, *way*, and *lane* levels in accordance with CityGML 3.0 specifications and levels of detail.
- The generation of semantically rich, interoperable CityGML datasets that can be eventually applied in urban planning tools, traffic simulations, and digital twin environments.

This paper is organized as follows: Section 2 provides a comprehensive review of point cloud processing methods and the integration of road features into standardized 3D city models. Section 3 describes the proposed methodology, while Section 4 shows the experiments and results obtained from applying the method to a real case study. Finally, Section 5 is devoted to concluding this work.

2. Related work

In recent years, the growing importance of digital twins and smart cities has driven significant advances in the modelling of 3D urban environments. The semantic and geometric modelling of road infrastructure has seen substantial advancements through the integration of point cloud data and standardized city models.

Various standards have been introduced for the 3D modelling of road networks, from linear-based representations such as OpenDrive and RoadXML, to surface-based approaches like LandInfra and CityGML. The introduction of CityGML 3.0 brought major conceptual and structural updates—including, among others, the space concept, improved LoD representation, and a revised Transportation module (Kutzner et al., 2020).

Tan et al., 2023 conducted a systematic literature review to evaluate the role of CityGML in BIM/GIS integration. Their analysis found that IFC-to-CityGML conversion remains challenging, mainly due to mismatches in geometry representations and semantic models. However, the space concept introduced in CityGML 3.0 offers a promising solution to maintain semantic consistency, while its new LoD structure improves the adaptability of features across levels.

While the above standards provide a robust semantic framework, the actual creation of geometric digital twins—particularly for roads— still demands significant manual effort. Addressing this, Davletshina et al., 2024 proposed a novel fully automated method that leverages LiDAR point clouds for segmenting and meshing road features. Their approach combines context-aware and location-aware segmentation using PointNet++ to classify road elements (e.g., lamps, guardrails, traffic signs) and builds semantically labelled polygonal mesh models. These 3D models are geometry-focused and platform-neutral, lacking structured semantic hierarchies like those in CityGML. A key advance lies in the integration of asset-specific GPS data, allowing for enhanced segmentation precision through localized bounding box extraction. Evaluated on large-scale datasets such as KITTI360 and the Digital Roads dataset, the method achieved up to 91.7% mean Intersection over Union (mIoU) for road furniture, significantly outperforming existing benchmarks.

Among recent applications of the CityGML 3.0 Transportation module, Beil & Kolbe, 2020 presented a framework for modelling road space elements. Their approach supports the semantic decomposition of road spaced into TrafficSpace and AuxiliaryTrafficSpace entities. A significant advantage of the model is its hierarchical structure, which enables the integration of traffic semantics and spatial analysis into simulation and decision-making processes. In their study, Schwab et al., 2020, proposed a spatio-semantic modelling approach targeted at vehicle-pedestrian simulations. Utilising georeferenced point clouds, they generated a CityGML dataset, which represents both physical and functional characteristics of urban environment.

Building on recent efforts to generate detailed semantic models from point clouds, González-Collazo et al., 2025 introduced a method that transforms MLS and HMLS data into CityGML 3.0-compliant models. Their approach emphasizes curbside-specific semantics, classifying urban spaces into parking for different users, parking by time, parking entrances, garbage bin spaces, and terrace areas. The processed point clouds are translated into CityGML's Transportation module at LoD3 with the 'way' semantic granularity, in which individual objects (e.g. driving lanes, sidewalks) are modelled per surface function. Validation against OpenDRIVE models and visualization using 3DCityDB and Google Earth Pro yielded F1-scores above 0.8 and IoU above 0.78, demonstrating strong alignment with real-world geometry.

An example of a city-scale semantic modelling initiative is presented by Lehner et al., 2024 through the development of Vienna's Digital geoTwin. Based on CityGML 2.0, the authors defined a tailored profile and implemented an Application Domain Extension (ADE) to accommodate Vienna's urban data requirements. The project integrated various datasets, such as terrain, buildings, vegetation, and city furniture, into a coherent semantic environment, forming Vienna's digital twin. To maintain geometric continuity when extracting features like buildings or bridges from the terrain model, they introduce a new ADE class, *LandUseClosureSurface*, conceptually extending CityGML's *ClosureSurface* which is however only available for Building, Tunnel, Bridge and WaterBody classes. The digital twin supports multiple Levels of Detail (LoD), with special emphasis on building decomposability and modular reuse through template geometries and xLinks.

In addition to surface-based representations, CityGML 3.0 introduces enhanced support for linear geometries, such as LoDXMultiCurve, enabling the representation of road networks with navigable, graph-based semantics. Beil & Kolbe, 2024 provided a comprehensive review of applications of semantic 3D streetspace models, highlighting the potential of CityGML 3.0 to support linear representations for modelling traffic direction and network topology. They emphasized the benefits of these representations for applications that require fine-grained navigation data, such as real-time routing, multimodal transport simulations, and simulation platforms for autonomous driving.

Yarroudh et al., 2023 presented a systematic approach for generating standardized 3D road infrastructure models using CityGML 3.0 and its CityJSON encoding. Their methodology relies on semi-automatic extraction of linear features—such as curbs and road boundaries—from mobile mapping LiDAR point clouds, which are then used to create semantically segmented surfaces (e.g., sidewalk, roadway, green space) through a codification and classification system. The final models are textured and exported to CityJSON, stored in a MongoDB database via the Measur3D platform, and visualized in a web-based application that was extended to support inspection of semantic surfaces.

https://doi.org/10.5194/isprs-annals-X-4-W6-2025-201-2025

3. Method

Building upon recent approaches, this study presents a structured methodology to transform urban road elements, derived from urban point clouds, into CityGML 3.0-compliant representations. The proposed method addresses both the geometric and semantic modelling of previously segmented road features, including driving lanes, sidewalks, carriageways, and parking areas, in accordance with the hierarchical structure and object classes defined by the CityGML 3.0 Transportation module. The proposed methodology follows a stepwise process: polygon pre-processing, semantic mapping to CityGML 3.0 classes, and generation of XML-based CityGML 3.0 output files.

In our method, data is organized into three distinct levels of granularity. It takes inspiration from the revised Transportation module of CityGML 3.0 (Beil et al., 2023). A multi-level granularity system is proposed to support hierarchical representations or transportation infrastructure. We define our three granularity levels as "area", "way", and "lane", respectively (Figure 1).

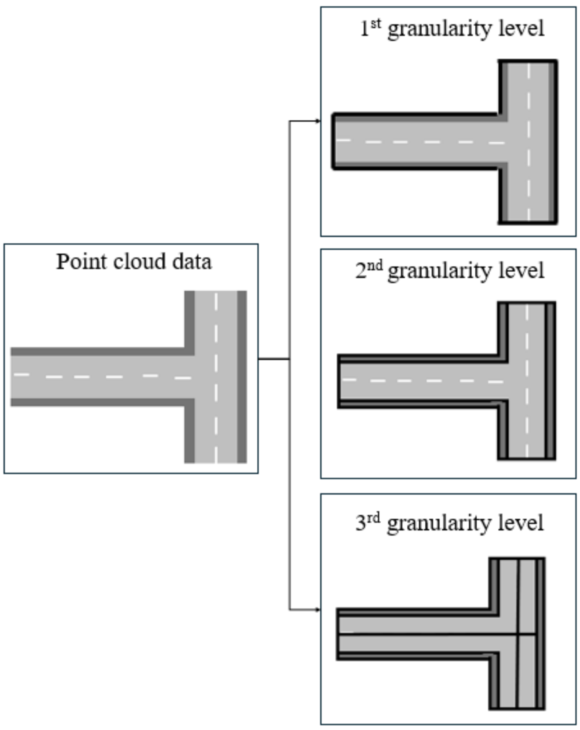

Figure 1. Conceptual representation of the 3 levels of granularity: 1st level (area), 2nd level (way), 3rd level (lane).

In particular, the *area* level consists of a set of objects, classified into sections or intersections and having each only one polygon-based geometry. Section and intersection objects can be then grouped to represent a whole road. In the 2nd level of granularity, *way*, individual objects are modelled per surface function including carriageway, sidewalk, and parking areas. Each object has its own polygon-based geometry and is obtained by splitting (wherever possible or necessary) the corresponding polygons of the previous level. Finally, in the 3rd level, *lane*, each individual lane is modelled separately, by further splitting the geometries of the previous level, wherever applicable. A graphical example is provided in Figure 2.

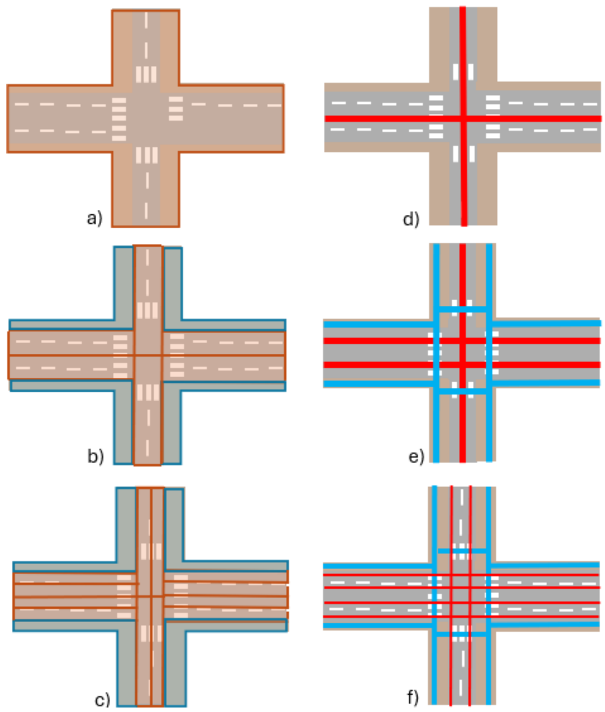

Figure 2. Three levels of granularity in surface and linear-based representations. a) *area*, b) *way*, c) *lane*, d) *area*, e) *way*, f) *lane*.

3.1 Geometrical pre-processing

This study builds upon a detailed segmentation methodology that transforms point cloud data into structured geometries of urban road space (Tsiranidou et al., 2025). This process begins with already classified point clouds (González-Collazo et al., 2024), in which the classes *road*, *sidewalk*, and *curb* are used to delineate functional urban areas. By detecting road markings, which serve as spatial subdivisions of road space, and using as inputs the road centrelines and the width of the road, surfaces such as driving lanes, parking areas, and zebra crossings are segmented into distinct functional areas (Tsiranidou et al., 2024) (Figure 3).

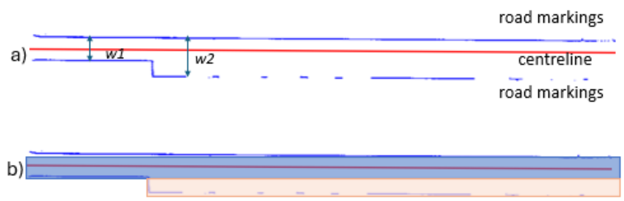

Figure 3. a) Input data (road width: *w1*, *w2*, centreline, road markings) to delineate road space. b) driving lane with blue colour, parking area with pink colour.

The resulting surface geometries derived from point cloud segmentation often contain noise, and topological inconsistencies, which complicate their direct use in semantically structured city models. Therefore, the initial step of this work involves geometric cleaning where small-scale noise and redundant vertices are removed (Tsiranidou et al., 2025).

At this stage, it should be noted that the geometries lack any of the semantic descriptors necessary for their integration into structured urban models. Thus, after geometric refinement, the dataset is prepared for semantic enrichment by ensuring compatibility with surface-based representations used in CityGML.

https://doi.org/10.5194/isprs-annals-X-4-W6-2025-201-2025

3.2 Semantic mapping to CityGML 3.0 classes

Once the geometries are pre-processed as described in the previous section, the next step involves their mapping to the structure defined by CityGML 3.0. In other words, the focus of this step is to interpret the geometric data by assigning it to appropriate classes within the Transportation module of the CityGML schema.

The hierarchical and semantical representation adopted in our method coincides very well with the one used by CityGML 3.0, in which class Road is used for both vehicles and pedestrians and is composed by Sections and Intersections. Each of these classes are specializations of class AbstractTransportationSpace, which can be further composed of TrafficSpace. In terms of geometry, a TrafficSpace object is meant to represent the volumetric characteristics of the space it models. However, TrafficSpace objects can be represented also by means of surface-based objects using class TrafficArea (Figure 4). Please note that, for the sake of readability, we only refer here to classes TrafficSpace and TrafficArea, but we actually mean also to the conceptually similar and structurally equivalent classes AuxiliaryTrafficSpace and AuxiliaryTrafficArea.

In general, the mapping between our levels of granularity and CityGML 3.0 works as follows:

- For each road, geometry-less objects from class Road are instantiated. They contain each geometry-less Section and Intersection objects, derived from the respective classes.
- For each Section/Intersection, we further create geometry-less TrafficSpace objects. Since the geometries resulting from our method belong to the boundary representation paradigm, we use then TrafficArea objects, as children of the respective TrafficSpace objects to store the geometries using multi-surfaces in different LoDs, depending on the level of granularity.
  In particular, we choose:
  - LoD1 for level *area*,
  - LoD2 for level *way*, and
  - LoD3 for level *lane*.

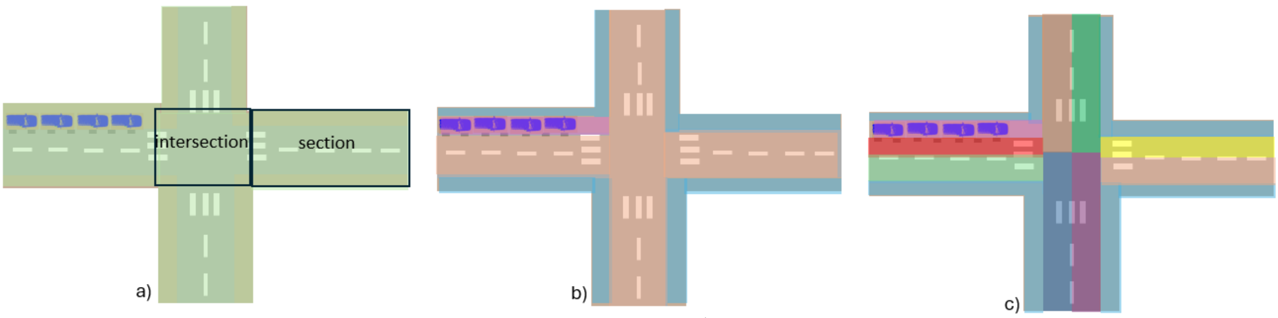

Figure 4. a) Road space in 1st level of granularity, b) TrafficAreas in 2nd level of granularity (blue colour for sidewalks, orange for carriageways and pink for parking areas), c) TrafficAreas in 3rd level of granularity (blue colour for sidewalks, pink for parking areas and driving lanes with distinct colour each one).

More in detail, a number of operations are still needed in order to enrich the geometrical data with the amount of additional information which is required for a proper mapping to CityGML. First, a globally unique identifier (*gml:id*) is assigned to each object in each level of granularity to ensure referential integrity and allow linking between components. In addition, each geometry is further annotated with descriptive attributes, including its segmentation origin, geometric classification and relevant contextual descriptors. These attributes are then used for semantic mapping, where each object is mapped to a specific class within the CityGML Transportation module.

For the 1st level of granularity, fields such as *name* (name of the street for sections and names of the intersected streets for intersections), *class* (indicating the object's type), and *object_class* (section or intersection) are defined for each section and intersection. Currently, data from the 1st level of granularity contains also attributes needed for mandatory top-level container objects in CityGML (i.e. Roads), such as the road *gml_id*, *class* and *function* attributes.

For the 2nd and 3rd level of granularity each object contains all attributes to extract both a TrafficSpace and the associated TrafficArea objects at once. In particular, attributes such as *class*, *function*, *usage, granularity* (indicating which modes of transportation can use it) and *traffic direction* are mapped to TrafficSpace attributes. At the same time, by using a prefix to avoid homonymous names in the input data, attributes such as *class*, *function*, *usage* and *surface material* are defined for TrafficSpace.

For all objects in all levels, information about their hierarchical order is provided by means of a *gml_parent_id* attribute. This field is later used, together with the *gml_id* attribute, to model the objects according to the CityGML 3.0 data model.

3.3 Generation of CityGML 3.0 output

The mapping defined in the previous step has been implemented in FME Form 2024 in order to write an XML-based CityGML 3.0 file. As at the time of writing (spring 2025) FME Form provides a specific CityGML 3.0 Reader but does not provide a specific CityGML 3.0 Writer (unlike for CityGML 2.0), a generic GML Writer must be used instead. The XSD files of CityGML 3.0 are used by the GML Writer to generate the corresponding writer transformers (one for each feature class), following a similar logic to the CityGML 2.0 approach. There are however some minor differences in the way data is prepared before being sent to the corresponding feature writer. For example, for CityGML 2.0, attributes such as *citygml_feature_role,* together with *gml_id* and *gml_parent_id,* are used for features to define in FME their hierarchy within the resulting output CityGML file. In the case of CityGML 3.0 and the GML writer, the procedure is similar, but the attribute *gml_parent_property* is used instead of the *citygml_feature_role*. At geometry level, similar differences

https://doi.org/10.5194/isprs-annals-X-4-W6-2025-201-2025 | © Author(s) 2025. CC BY 4.0 License.

apply, too. In CityGML 2.0, the attribute as *citygml_lod_name* must be “injected” as geometry property/trait into the corresponding geometry. In the case of CityGML 3.0 and the GML writer, this is not the case anymore. The FME “standard” attribute *geometry_name* is injected instead, i.e. via a GeometryPropertySetter transformer.

In order to test the validity of the resulting file, the FME option to validate the file has been used throughout the process. Once the XML file has been written, additional tests have been carried out to test its usability in other software tools. For example, visualisation tests have been successfully carried out using both the FME Data Inspector 2024 and the KIT ModelViewer. Additionally, data has been successfully imported into the recently released 3DCityDB 5.0 (Yao et al., 2025), which supports CityGML 3.0, and successfully visualised in QGIS using a newly developed server-side part of the QGIS plugin for the 3DCityDB (Tsai et al., 2025).

4. Experiments and results

4.1 Case study

To validate the proposed methodology, a real-world case study was conducted using MLS point cloud data from a 2-kilometer urban street network, as described in (González-Collazo et al., 2024). Data acquisition was performed using a hybrid approach, combining handheld and vehicle-mounted Mobile Laser Scanners in a 2-kilometer urban street network. The captured point clouds underwent a semi-automated semantic labelling process, leveraging both heuristic-based rules and Deep Learning classifiers, resulting in eight distinct classes: *road*, *sidewalk*, *curb*, *buildings*, *vehicles*, *vegetation*, *poles*, and *others*. For this work, only the *road*, *sidewalk* and *curb* classes were used (Figure 5).

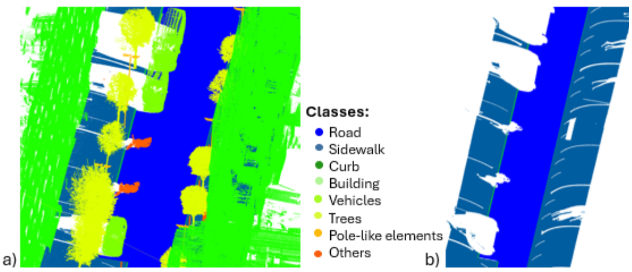

Figure 5. a) Point cloud data, b) Classes (*road*, *sidewalk*, *curb*) used in this work.

4.2 Geometrical pre-processing

In the initial phase of our workflow, raw MLS point cloud data was processed to extract polygonal geometries corresponding to urban elements. The input dataset covered approximately 2 km of a dense urban network (Figure 6).

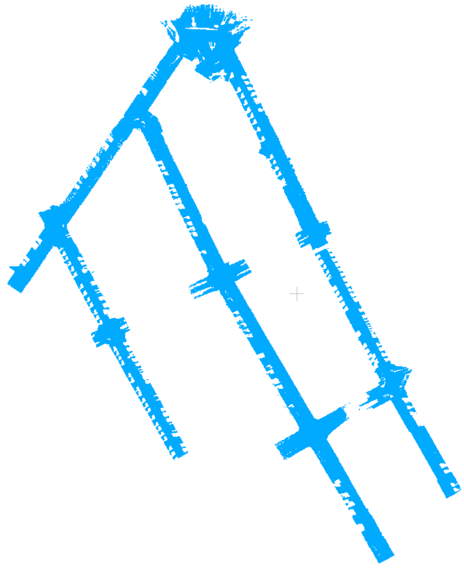

Figure 6. The input point cloud data (roads, curbs and sidewalk classes).

From the raw point clouds, a total of 19 polygons were automatically extracted for the 1st level of granularity, corresponding to sections and intersections. At the 2nd level, representing individual carriageways, sidewalks, and parking areas, 84 polygons were extracted, and the 3rd level yielded a total of 123 polygons corresponding to driving lanes, sidewalks, and parking areas (Figure 7).

4.3 Semantic mapping to CityGML 3.0 classes

The mapping process followed the granularity-driven strategy, as defined before, which aligns with the hierarchical structure of CityGML 3.0. Specifically, 7 Road objects were created composed of 11 Sections and 8 Intersections, each one modelled as instance of the AbstractTransportationSpace class. XLinks were used to model those intersections belonging to different roads. Each Section and Intersection were further enriched by instantiating TrafficSpace objects and surface geometries were stored using TrafficArea objects, for the LoD1, LoD2 and LoD3. In the 2nd level of detail, a total of 84 TrafficArea objects were created, including 51 sidewalks, 19 carriageways, and 14 parking areas. In the 3rd level of detail, a total of 123 TrafficArea objects were created, consisting of 58 driving lanes, 51 sidewalks, and 14 parking areas.

All geometries in each level of granularity were assigned a unique *gml_id* attribute. For the 1st level of granularity, the name of the street was provided for each section and intersection (via the *name* field), and a class attribute was added to indicate the type of section or intersection. All sections in the dataset were labelled as *road corridor*, while intersections were classified based on how many roads intersect: *4-way intersection* (4 roads intersect), or *T-type intersection* (3 roads intersect). Also, in the field *object_class* each object was characterized as *section* or *intersection*. Finally, to represent hierarchical relationships, the *gml_parent_id* field (storing an array of ids, in the case of intersections) was completed for the sections according to which Road they belong, and for the intersections based on which Roads intersect for each one. The *road_class* was specified as *road traffic* for all the objects and the *road_function* as *municipal road* for the sections and *junction* for the intersections, in accordance with their respective roles in the transportation network.

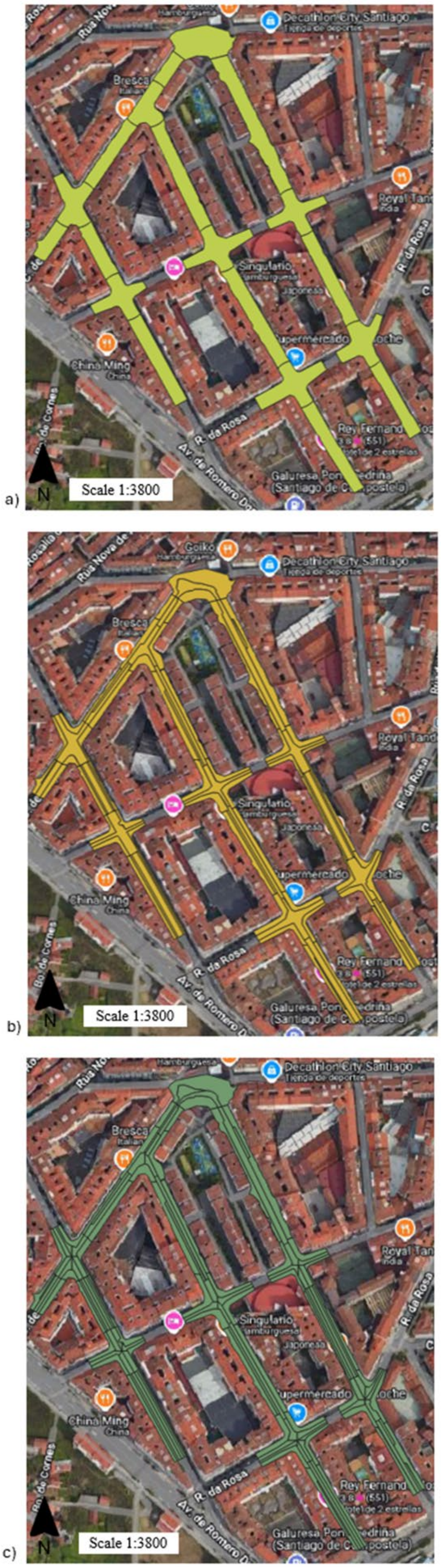

Figure 7. a) 1st level of granularity. b) 2nd level of granularity. c) 3rd level of granularity.

For the 2nd level of granularity, the *class* attribute was assigned based on the type of each object, with values including *carriageway*, *sidewalk*, and *parking*. The *function* attribute was then derived accordingly: objects classified as *carriageway* were assigned the function *driving_lane*, *sidewalk* objects were assigned *footpath*, and *parking* objects were assigned *car_park*. For the *usage* attribute, carriageway and parking objects were labelled with *car*, and sidewalk objects with *pedestrian*. The *surfaceMaterial* attribute was also defined as *asphalt* for carriageway and parking areas, and as *pavement* for sidewalks. Additionally, in the field *object_class* each object was characterized as *parking*, *carriageway*, or *sidewalk*. For the *granularity*, we assigned the label *way*, corresponding to its representation at LoD2. Following the same approach as in the 1st level, the *gml_parent_id* attribute was filled with numerical values showing hierarchical order with the 1st level of granularity.

In the 3rd level of granularity, the same values used in the 2nd level were assigned to the attributes *class*, *function*, *usage*, *surfaceMaterial*, and *object_class*. The *granularity* attribute was set to *lane* corresponding to its representation at LoD3. Lastly, hierarchical consistency was maintained through the *gml_parent_id*, which links each object to its corresponding parents in the 1st level. Figure 8 shows the resulting street model, written as a unique XML file containing each level of granularity, and visualized in the KIT ModelViewer. In particular, Figure 8c depicts the third level of granularity, where each driving lane, sidewalk, and parking area is represented as an individual polygon. The visual distinction between Figure 8b and Figure 8c may appear limited at first glance because the same colours are used across levels to represent consistent surface functions—for example, both carriageways and individual driving lanes are shown in pink.

A closer analysis of the results at the 3rd level of granularity (Figure 8c) shows that the method can effectively distinguish and generate individual surface functions, such as driving lanes, sidewalk, and parking areas as separate CityGML objects. In most road segments, the subdivision is consistent with the available spatial cues (e.g., road markings and centrelines), producing well-defined geometries that support downstream applications like routing or simulation. However, the method's accuracy is reduced in cases where the road markings are missing, worn, or occluded—common issues in dense urban environments. In such situations, the subdivision may become less reliable or produce ambiguous boundaries between functional areas. Nonetheless, with supplementary geometric cues such as road width and centrelines, the subdivision can still achieve satisfactory accuracy. These cues help guide the delineation of driving lanes and parking areas even in the absence of explicit markings, contributing to more stable results across varying data quality.

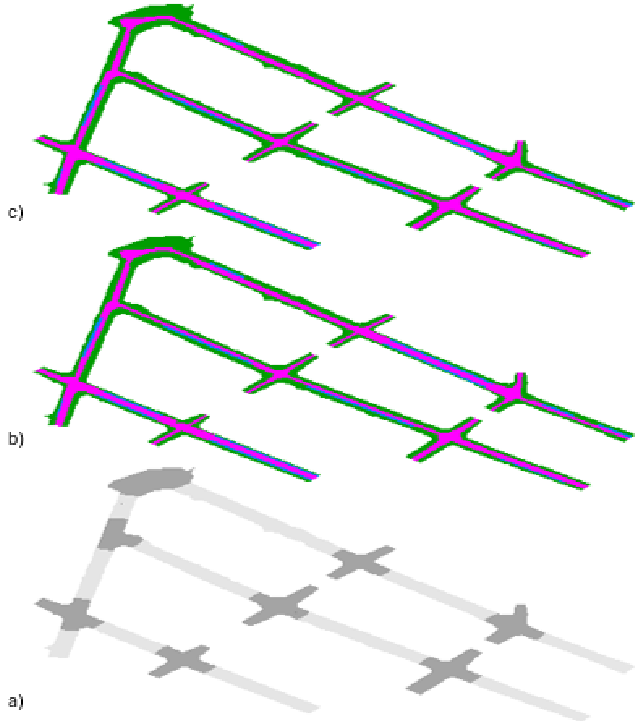

Figure 8. Visualization in KIT ModelViewer. a) 1st level. b) 2nd level. c) 3rd level.

5. Conclusions

This paper presents a structured methodology for integrating segmented road surface elements, derived from urban point clouds, into semantically rich and CityGML 3.0-compliant representations. By leveraging the structure of the CityGML 3.0 Transportation module, our approach systematically transforms geometric representations of road infrastructure, such as driving lanes, sidewalks, and parking areas, into valid XML-based outputs ready for integration into urban digital twins, planning tools, and simulation platforms.

In our work point clouds are used as the primary data source. These datasets provide a high-resolution representation of real-world environments, capturing fine-grained details such as road markings, curbs and changes in elevation. Their dense spatial coverage and precise geometric information allow for the accurate segmentation of complex urban road networks, including sidewalks, parking areas, and driving lanes. It is important to note that point clouds enable data-driven modelling that reflects the actual physical conditions of road infrastructure, enhancing the realism and utility of resulting 3D models.

Our methodology pre-processes these geometries to remove noise but also maps them to CityGML 3.0 schema using a granularity-driven classification approach. The resulting data can be used or visualised in tools such as 3DCityDB, FME Data Inspector, KIT ModelViewer and QGIS. The hierarchical mapping between road elements—using attributes such as *gml:id* and *gml_parent_id*—ensures structural integrity and supports interoperability with existing GIS systems.

While this work focuses primarily on surface-based representations, a key approach for future research is the integration of linear representations through the *LoDXMultiCurve* geometry type introduced in CityGML 3.0. Linear representations are especially relevant for applications requiring a clear understanding of network topology, such as route planning, modelling of traffic direction, and simulation of multimodal transport networks. Their integration would allow the development of semantically enriched road graphs that complement the surface-based geometries currently used.

In summary, this work provides a framework for bridging the gap between raw point cloud data and semantically structured city models. It shows that, through a well-defined process of geometric refinement and semantic mapping, it is possible to produce interoperable, CityGML 3.0-compliant datasets that contribute to the development of smart cities and urban digital twins.

6. Discussion

This study presents a structured and semantically aligned pipeline for transforming segmented point clouds into CityGML 3.0-compliant models. The experiments confirm that our method can effectively generate LoD1, LoD2, and LoD3 representations, which can be imported and visualized in existing CityGML-compatible platforms such as 3DCityDB, KIT ModelViewer, and QGIS. However, several aspects warrant further discussion regarding generalisation, limitations, and real-world applicability.

Although the current case study covers a 2-kilometer urban street segment, the modular nature of our pipeline suggests it is scalable to larger datasets. In practice, extending the workflow to larger urban areas would primarily require higher computational resources during the semantic segmentation and polygon pre-processing stages. However, future experiments on full-city datasets are required to quantitatively assess runtime, storage, and processing bottlenecks in high-density urban scenarios.

In terms of robustness, the pipeline assumes a reliable input segmentation, but practical cases often involve noise, occlusions, or ambiguous features. The workflow includes geometric cleaning and consistency checks, yet certain edge cases may still benefit from user supervision or interactive validation steps. Future improvements could integrate uncertainty estimation or confidence levels in the segmentation stage, helping to identify and address ambiguous areas more effectively.

The detailed output produced by the method is particularly relevant for applications that require fine-grained spatial logic, such as routing, accessibility analysis, and traffic simulation. Nonetheless, a critical step in the process is the subdivision of carriageways into individual driving lanes and parking areas. This relies on the accurate detection of road markings and geometric features in the point cloud. When such features are missing, occluded, or worn, the segmentation may be less complete or ambiguous, which can affect the semantic detail and structural correctness of the final model. Enhancing this stage with complementary information—such as curb lines, prior road layouts, or HD maps—could further improve the robustness and completeness of the results.

Finally, the structured, hierarchical output aligns with CityGML 3.0's Transportation module and is suitable for integration into GIS platforms, digital twins, and urban analysis workflows. By supporting multiple levels of detail, the method enables different applications to access road infrastructure data at the appropriate resolution, balancing semantic richness with practical usability.

Acknowledgements

This work was partially supported by human resources grant RYC2020-029193-I funded by MCIN/AEI/10.13039/501100011033 and FSE 'El FSE invierte en tu futuro', by grant ED431F 2022/08 funded by Xunta de Galicia, Spain-GAIN, and by the projects PID2021-123475OA-I00 funded by MCIN/ AEI /10.13039/501100011033/ and by FEDER 'Una manera de hacer Europa' and CNS2022-135730 funded by MCIN/AEI /10.13039/501100011033 and by European Union NextGenerationEU/PRTR. The statements made herein are solely the responsibility of the authors.

References

Beil, C., & Kolbe, T. H. (2020). Combined modelling of multiple transportation infrastructure within 3D city models and its implementation in CityGML 3.0. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *6*(4/W1), 29–36. https://doi.org/10.5194/isprs-annals-VI-4-W1-2020-29-2020

Beil, C., & Kolbe, T. H. (2024). Applications for Semantic 3D Streetspace Models and Their Requirements—A Review and Look at the Road Ahead. In *ISPRS International Journal of Geo-Information* (Vol. 13, Issue 10). Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/ijgi13100363

Beil, C., Kutzner, T., Schwab, B., & Kolbe, T. H. (2023). *Road2CityGML3 (1.0)*. Zenodo.

Beil, C., Ruhdorfer, R., Coduro, T., & Kolbe, T. H. (2020). Detailed streetspace modelling for multiple applications: Discussions on the proposed CityGML 3.0 transportation model. *ISPRS International Journal of Geo-Information*, *9*(10). https://doi.org/10.3390/ijgi9100603

Davletshina, D., Reja, V. K., & Brilakis, I. (2024). Automating construction of road digital twin geometry using context and location aware segmentation. *Automation in Construction*, *168*. https://doi.org/10.1016/j.autcon.2024.105795

González-Collazo, S. M., Balado, J., Garrido, I., Grandío, J., Rashdi, R., Tsiranidou, E., del Río-Barral, P., Rúa, E., Puente, I., & Lorenzo, H. (2024). Santiago urban dataset SUD: combination of handheld and mobile laser scanning point clouds. *Expert Systems with Applications*, *238*. https://doi.org/10.1016/j.eswa.2023.121842

González-Collazo, S. M., Schwab, B., Beil, C., Kolbe, T. H., González, E., & Balado, J. (2025). Curbside management from MLS and HMLS point clouds to CityGML 3.0. *Geo-Spatial Information Science*. https://doi.org/10.1080/10095020.2025.2454522

Gröger, G., & Plümer, L. (2012). CityGML - Interoperable semantic 3D city models. In *ISPRS Journal of Photogrammetry and Remote Sensing* (Vol. 71, pp. 12–33). https://doi.org/10.1016/j.isprsjprs.2012.04.004

Kolbe, T. H., Kutzner, T., Smyth, C. S., Nagel, C., Roensdorf, C., & Heazel, C. (2021). OGC City Geography Markup Language (CityGML) Part 1: Conceptual Model Standard. *Open Geospatial Consortium*.

Kutzner, T., Chaturvedi, K., & Kolbe, T. H. (2020). CityGML 3.0: New Functions Open Up New Applications. *PFG - Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, *88*(1), 43–61. https://doi.org/10.1007/s41064-020-00095-z

Lehner, H., Kordasch, S. L., Glatz, C., & Agugiaro, G. (2024). Digital geoTwin: A CityGML-Based Data Model for the Virtual Replica of the City of Vienna. *Lecture Notes in Geoinformation and Cartography*, 517–541. https://doi.org/10.1007/978-3-031-43699-4_32

Schwab, B., Beil, C., & Kolbe, T. H. (2020). Spatio-semantic road space modeling for vehicle-pedestrian simulation to test automated driving systems. *Sustainability (Switzerland)*, *12*(9). https://doi.org/10.3390/su12093799

Tan, Y., Liang, Y., & Zhu, J. (2023). CityGML in the Integration of BIM and the GIS: Challenges and Opportunities. In *Buildings* (Vol. 13, Issue 7). Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/buildings13071758

Tsai, B.-S., Agugiaro, G., León-Sánchez, C., Nagel, C., & Yao, Z. (2025). Introducing server-side support for 3DCityDB 5.0 to the 3DCityDB-Tools plug-in for QGIS. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci. Proc. of the 20th 3DGeoInfo and 9th Smart Data Smart Cities* .

Tsiranidou, E., Brea Portals, A., Fernández Álvarez, A., & Díaz Vilariño, L. (2024). Semantic enrichment of urban space: A point cloud-based framework for road marking analysis. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *10*(4/W5-2024), 281–288. https://doi.org/10.5194/isprs-annals-X-4-W5-2024-281-2024

Tsiranidou, E., Fernández, A., & Díaz Vilariño, L. (2025). A Framework for Road Space Extraction from Point Clouds and Integration into 3d City Models. *Available at SSRN*. https://doi.org/10.2139/ssrn.5153890

Yao, Z., Nagel, C., Kendir, M., Willenborg, B., & Kolbe, T. H. (2025). The new 3D City Database 5.0 – Advancing 3D city data management based on CityGML 3.0. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci. Proc. of the 20th 3DGeoInfo and 9th Smart Data Smart Cities* .

Yarroudh, A., Nys, G. A., & Hajji, R. (2023). 3D modelling of road infrastructures according to CityGML 3.0 and its CityJSON encoding. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *48*(1/W2-2023), 63–70. https://doi.org/10.5194/isprs-archives-XLVIII-1-W2-2023-63-2023