

# PLASTIC SURGERY FOR 3D CITY MODELS: A PIPELINE FOR AUTOMATIC GEOMETRY REFINEMENT AND SEMANTIC ENRICHMENT

O. Wysocki<sup>1</sup>, B. Schwab<sup>2,3</sup>, L. Hoegner<sup>1</sup>, T. H. Kolbe<sup>2</sup>, U. Stilla<sup>1</sup>

<sup>1</sup>Chair of Photogrammetry and Remote Sensing, Technical University of Munich, 80290 Munich, Germany – (olaf.wysocki, ludwig.hoegner, stilla)@tum.de

<sup>2</sup>Chair of Geoinformatics, Technical University of Munich, 80290 Munich, Germany – thomas.kolbe@tum.de

<sup>3</sup>AUDI AG, 85045 Ingolstadt, Germany – benedikt.schwab@audi.de

## Commission IV

**KEY WORDS:** Geometry Refinement, Semantic 3D City Model, HD Map, Point Cloud, Mobile Mapping, Autonomous Driving

### ABSTRACT:

Nowadays, the number of connected devices providing unstructured data is rapidly rising. These devices acquire data with a temporal and spatial resolution at an unprecedented level creating an influx of geoinformation which, however, lacks semantic information. Simultaneously, structured datasets like semantic 3D city models are widely available and assure rich semantics and high global accuracy but are represented by rather coarse geometries. While the mentioned downsides curb the usability of these data types for nowadays' applications, the fusion of both shall maximize their potential. Since testing and developing automated driving functions stands at the forefront of the challenges, we propose a pipeline fusing structured (CityGML and HD Map datasets) and unstructured datasets (MLS point clouds) to maximize their advantages in the automatic 3D road space models reconstruction domain. The pipeline is a parameterized end-to-end solution that integrates segmentation, reconstruction, and modeling tasks while ensuring geometric and semantic validity of models. Firstly, the segmentation of point clouds is supported by the transfer of semantics from a structured to an unstructured dataset. The distinction between horizontal- and vertical-like point cloud subsets enforces a further segmentation or an immediate refinement while only adequately depicted models by point clouds are allowed. Then, based on the classified and filtered point clouds the input 3D model geometries are refined. Building upon the refinement, the semantic enrichment of the 3D models is presented. The deployment of a simulation engine for automated driving research and a city model database tool underlines the versatility of possible application areas.

## 1. INTRODUCTION

Currently, large municipalities around the world develop 3D city models. The wide availability of aerial images, Airborne Laser Scanning (ALS) point clouds, accurate cadastral records, and ultimately efficient algorithms leads to the creation of urban 3D models on an unprecedented scale. The models are often created in a CityGML-compliant manner enabling the managing of 3D semantic models. However, the automatic reconstruction methods have certain limitations resulting from the geospatial information acquisition technique (Haala and Kada, 2010). One of the pivotal downsides is the top-view looking acquisition that e.g., prevents capturing building façades and thus limits the achievable Level of Detail (LoD) of the reconstructed object. The recent interest in detailed road space modeling is driven by several factors. Thereby, the development of automated driving functions is a pivotal one. This trend reflects in an increased number of mobile mapping units scanning road environments. This, however, results in an influx of geodata like Mobile Laser Scanning (MLS) point clouds and High Definition (HD) Maps that depict the road network and its space supporting the navigation and simulation of automated vehicles. Nevertheless, HD Maps may be valid for several test categories of automated driving functions, but as soon as more complex physical sensor effects are demanded for testing, they are not sufficient anymore (Schwab and Kolbe, 2019). For that purpose, more detailed geometrical and semantical representations of real environments are needed. Moreover, the geodata flood is strengthened by the growth of connected devices equipped with LiDARs, cameras, and RGB-D sensors. Consequently, the question arises of how preexisting models can be geometrically refined and semantically enriched

using the increasing influx of unstructured data. Simultaneously,

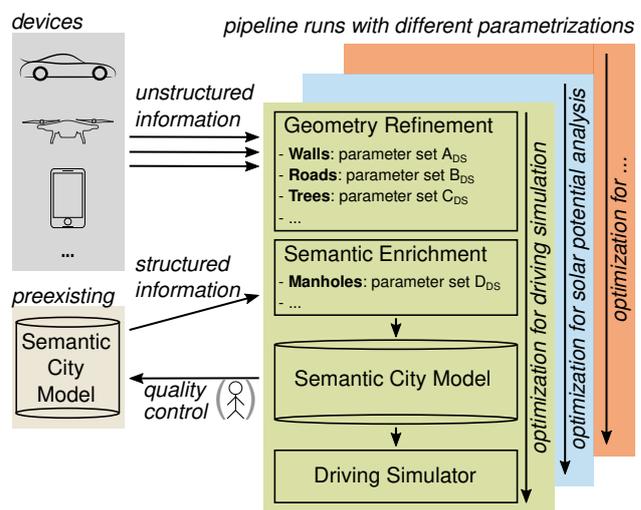


Figure 1. Overview of the pipeline concept, whereas DS denotes the application driving simulation

a broadening range of applications for different purposes is being developed. Depending on the task, each of these applications have different requirements and preferences for 3D models. For example, while maximizing the geometric accuracy of roof surfaces may improve the results of a solar potential analysis (Willenborg et al., 2018), the increased complexity could have a negative impact on the real-time capability of a driving simulation (Schwab and Kolbe, 2019). For the latter, it might be

tolerable that the geometric deviation increases quadratically with the distance to the road. Moreover, the geometric accuracy may be in conflict with the time required to conduct a citywide solar potential analysis. Fundamentally, this is a multi-objective optimization problem with conflicting objectives (e.g., application runtime, result accuracy, memory usage). Since the weighting of the objectives is application or application run specific, a Pareto efficient solution can be found at best. As application algorithms react differently to changing 3D model characteristics, the cost functions of the optimization problem are also application specific.

In order to maximize the potential of structured and unstructured datasets, we propose a customizable pipeline concept accommodating for application-specific requirements as depicted in Figure 1. To optimally configure the parameters for a specific application, a complete parameterization of the pipeline modules should therefore be possible before the execution is triggered. The pipeline should be considered as an end-to-end solution in which different modules for geometry refinement and semantic enrichment can be added. While there are various definitions of semantic enrichment (Xue et al., 2021), we define it as a process of joining semantic information to a semantic city model both as a geometric and non-geometric semantic for application-specific tasks following the definition of (Xue et al., 2021). Whereas the geometry refinement refers to a challenge of the resolution increase of existing geometries for application-specific tasks abstracting from defined LoDs (Gröger et al., 2012) while maintaining existing geometric semantics (Xue et al., 2021). Both concepts, however, are inline with 2.0 and 3.0 versions of CityGML modeling guidelines (Gröger et al., 2012; Kutzner et al., 2020). The pivotal strength of the proposed end-to-end pipeline is the integration of solutions from various domains like point cloud semantic segmentation, object reconstruction, and modeling while maintaining the geometric and semantic validity of processed objects. Moreover, the processing algorithms are supported by prior knowledge extracted from city models reducing the complexity of tasks. This underlines how existing semantic city models may help in tackling issues like semantic segmentation of unstructured datasets without the need for e.g., computationally expensive deep learning algorithms deployment. Moreover, our work proposes an automatic plausibility test for surface reconstruction based on point clouds coverage analysis as restrictions to data acquisition often occur (e.g., backyard) and limit reconstruction possibility (Xu and Stilla, 2021). Hence, we have placed *plastic surgery* in the title as the pipeline forces enhancements of only adequately covered city models. As a first feasibility test of the concept, an exemplary pipeline with modules for geometry refinement and semantic enrichment for the purpose of automated driving testing is presented. Moreover, parameterization tests are conducted and pipeline results are evaluated using reference building models in LoD2 and LoD3. Finally, the refined models are transferred to first applications, such as the Unreal Engine. The implementation is partly based on the Master's Thesis of (Wysocki, 2020).

## 2. RELATED WORK

Data or information can be distinguished w.r.t. its underlying structure. Thereby, structured data is organized in a predefined schema enabling efficient data processing and content navigation (Sint et al., 2009). In order to structure geometric, topological, appearance, and semantic information of cities and landscapes, the open standard CityGML is utilized internationally. CityGML is used for representing, storing, and exchanging semantic 3D city and landscape models. It provides a common definition of

basic entities, attributes as well as relations and is therefore applied in a variety of application domains (Biljecki et al., 2015). The standard is an application schema of the Geography Markup Language (GML) and version 2 was issued by the Open Geospatial Consortium (OGC) in 2012 (Gröger et al., 2012), with version 3 currently being finalized (Kutzner et al., 2020). To describe the logic of road networks including their lane topology, geometries, and traffic rules, the standard OpenDRIVE is widely adopted for driving and traffic simulation applications. OpenDRIVE is based on a linear referencing concept, whereby the lane geometries, road objects, and traffic rules are defined in a track coordinate system. The standard was developed for simulation and testing purposes but is also used to describe HD Maps by georeferencing the road network with a proj4 string. The current version 1.6 was published in 2020 by the Association for Standardization of Automation and Measuring Systems (2020).

To create semantically rich 3D models of the as-built environment, surveying campaigns are conducted, which yield unstructured data. E.g., point clouds acquired via Terrestrial Laser Scanning (TLS) are often used in 3D building modeling, development of digital surface models, and environment monitoring (Vosselman and Maas, 2010). Point clouds have been suggested as the most appropriate data source for the sake of 3D mapping in large-scale urban scenes because measured 3D points can provide directly spatial coordinates of measured surfaces. The method for generating building models from point clouds is split into several steps. First, the segmentation and classification of the point cloud into basic building elements like planes and cylindrical objects can either be data-driven or model-driven. Data-driven methods are based on point features (Habib et al., 2010) like intensity values or geometric features like i.e., normal direction from a local point neighborhood (Niemeyer et al., 2014). These neighborhoods can be fixed or adaptive with respect to the point density (Weinmann et al., 2015). Such neighborhood can also be replaced by a voxel structure where the feature description is then stored per voxel instead per point (Xu et al., 2018b). Based on features, points can be classified and similar points are connected to segments (Yang et al., 2016). These segmentation and classification approaches are based on methods like Markov Random Field (Lu and Rasmussen, 2012) or Random Forest (Chehata et al., 2009) classifiers or neural networks (Wang et al., 2017). In the next step, the extraction of primitives can be carried out on points or voxels. Some objects can be represented by fitting geometric primitives to point cloud segments like i.e., planes or cylindrical objects (Xu et al., 2018a). After reconstruction, the resulting geometric primitives, as well as voxels and points, are labeled with classes and handed over to further processing to fulfill necessary requirements for building or city models like CityGML.

The numerous works tackle the challenge of 3D reconstruction in contrast to the enrichment of existing 3D city models that gained little research attention (Xue et al., 2021). Nevertheless, adding geometric and non-geometric semantics is addressed e.g., by detecting and modeling windows on a façade based on the so-called *voyeur effect* (Tuttas and Stilla, 2013). Other approaches focus on the city models enriching utilizing building information models (BIM) (Stouffs et al., 2018). Also, the geometry refinement research niche is expressed by (Willenborg et al., 2018) and the linking of existing mesh models with superimposed semantic models is presented.

However, these approaches do not comprehensively leverage the information from already existing semantic 3D model datasets derived from prior surveying campaigns. They focus on linking existing meshes to semantic models, selectively apply prior information, or neglect the prior 3D models in the reconstruction

process. Furthermore, previous works have primarily focused on the reconstruction of single object types or groups. Thus, the question arises of how to integrate the variety of methods into one pipeline and how existing semantic models can support the reconstruction methods at subsequent processing steps while maintaining semantic and geometric validity w.r.t. to city model.

### 3. PROPOSED PIPELINE

In order to address the aforementioned challenges, we propose the method shown in Figure 2. The strategy assumes utilization of dense MLS point cloud data and HD Maps as OpenDRIVE converted to the CityGML standard using the converter r:trân (Schwab et al., 2020). The prerequisite for this method is a georeferenced MLS point cloud at the cm-grade global accuracy. As a supportive and optional dataset serves an ALS point cloud. All the steps of the workflow are implemented within the FME 2020 environment with integrated LASTools, MeshLabServer, and Python scripts presenting an end-to-end solution. The implementation is available within the project’s repository <sup>1</sup>.

After clipping the point clouds according to the objects to be refined, the point cloud subsets representing a vertical-like object (e.g., walls) are directed to the segmentation processing step, while subsets representing a horizontal-like object (e.g., roads) are directly directed to the surface reconstruction step. The splitting into horizontal-like and vertical-like object representations is architecturally shown in Figure 2 and enables a faster execution of the pipeline. The suffix *-like* is added as neither horizontal nor vertical objects are represented by ideal plane surfaces in reality. For example, a single segment of a road is a horizontal-like object consisting of horizontal-like parts but a building’s wall is a vertical-like structure.

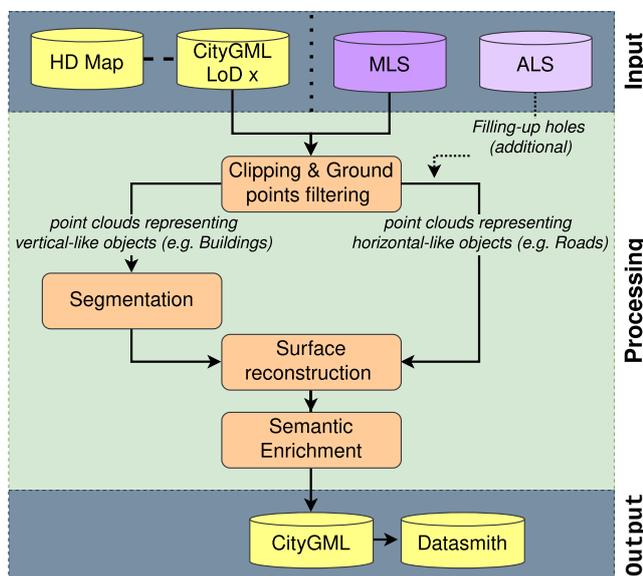


Figure 2. Methodology overview: Unstructured datasets marked with purple while structured datasets with yellow hue and the action boxes have orange color

Firstly, the raw semantic vector objects restrict the respective MLS point cloud dataset to the maximum possible extent depending on the input models as described in subsection 3.1. This operation reduces the input dataset from city to building scale, while simultaneously preserving inliers. Afterwards, the separation to vertical-like and horizontal-like objects is applied to

<sup>1</sup> <https://github.com/tum-gis/CityModelSurgery>

remove the majority of outliers from horizontal-like structures but not for vertical-like where large portions of outliers are still present (e.g., vegetation). However, the horizontal-like structures have gaps resulting from filtering objects occluding the depiction of a surface. This is overcome by adding the ALS point cloud to fill-in occluded areas in the dataset (see subsection 3.1). Then, horizontal-like objects are passed to the reconstruction part while vertical-like objects are further segmented, as described in subsection 3.2. The reconstruction part with pre-processed point clouds is controlled by four parameters – this process is described within subsection 3.3. The refined geometries are augmented by additional semantic information, as shown in subsection 3.4. Thereafter, the output models are stored as CityGML 2.0 and 3.0 datasets, whereas the validation is performed *inter alia* with the 3DCityDB suite. Moreover, a converter from the CityGML to the Datasmith format is created. This format is dedicated to utilization in Unreal Engine applications. Based on the translated data an interactive game is developed (see section 4.). This stands as a proof of concept for the pipeline utilization in automated driving applications and 3D cadastre among others.

#### 3.1 Clipping & ground points filtering

To enable efficient processing, the first step is to select only those points that represent the road space object to be geometrically refined. Since this operation depends on the absolute accuracy of the object, the clipping range is selected depending on the object’s LoD. Here, the recommendations of the CityGML standard are adopted as the preset, whereas this parameterization can also be defined by the pipeline operator beforehand. For example, a LoD1 building model leads to a buffer of 5 m, while a LoD2 geometry of a road may require a different buffer optimum depending on subsequent reconstruction methods.

Due to the calculations of Euclidean distances and the creation of sphere-like masks, standard geographic buffer operations in 3D can be computationally demanding. In order to avoid that, a mixture of a 2D buffer with an extrusion operation is proposed. First, a 2D buffer is created, then an extrusion is calculated in positive and negative direction in the third dimension. Therefore, the respective buffers operate in the X, Y, Z directions (Z positive and negative) in a Manhattan-like manner overestimating the buffer’s range. This approach prevents the removal of inliers while significantly reducing the number of outliers.

After clipping, the point cloud subsets still contain outliers. As shown in Figure 3, road objects may contain representations of vehicles, whereas wall objects can still contain trees. To separate horizontal- and vertical-like objects within the subsets, the lasground tool of the LASTools collection is applied with dedicated *non-airborne* and *urban environment* parameters. Due to inherited semantics from the input models, the algorithm can decide to mark horizontal-like points as positive or negative (e.g., roads or buildings respectively in horizontal-like subsets) and subsequently steer the subsets for further segmentation or directly to surface reconstruction. As shown in Figure 3, the segmentation is required for vertical-like subsets, since portions of point clouds depicting irrelevant extruded objects, such as trees, are still present. In the case of horizontal-like subsets, the fusion of ALS and MLS data is performed in order to accommodate for anticipated gaps resulting from filtering-out vertical structures, as shown in Figure 3. Alternatively, a Digital Elevation Model (DEM) can be used to compensate for areas where vertical occlusions constantly exist (like parking lots). ALS point clouds are not fused for vertical objects as the acquisition geometry results in very sparse coverage of vertical structures.

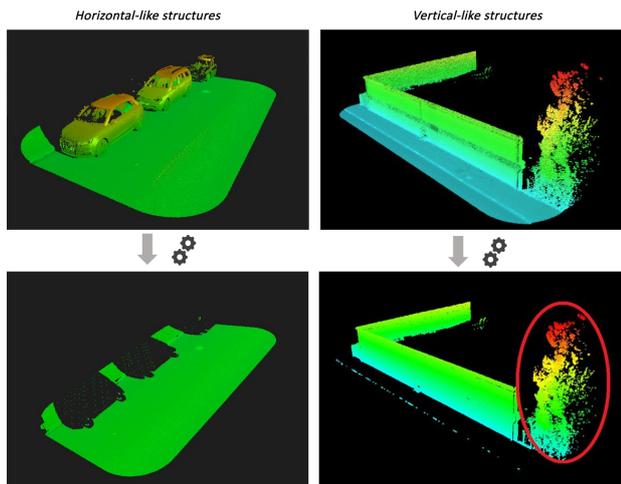


Figure 3. Application of the filtering to horizontal- and vertical-like structures within clipped area. The red circle marks outstanding outliers within vertical-like objects

### 3.2 Segmentation

Since the goal is to refine planar city features (e.g., fences, walls, traffic signs), all complex extruded objects like vehicles and vegetation should be treated as noise. Most often, the vertical-like structures consist of several vertical segments, such as a building that is composed of several walls. However, due to occlusions or objects not in the scanner's field of view (e.g., backyard), not all objects are adequately represented by the MLS point cloud. Such structures should be skipped in the further processing.

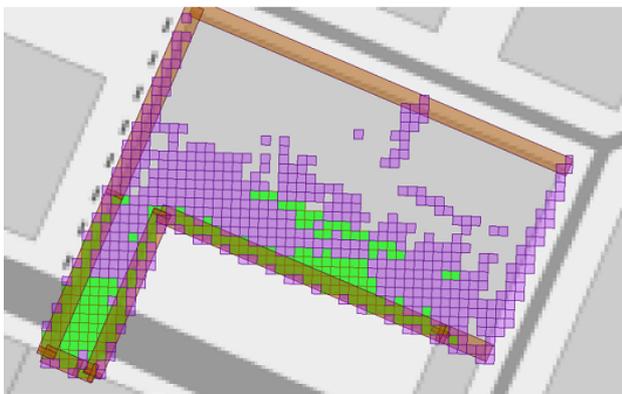


Figure 4. Coverage assessment of the walls of a building visualized from the top view: Qualified walls (brown), dense patches (green), and sparse patches (purple). The large glass fronts of the building resulted in substantial laser reflections inside the building footprint.

Hence, the coverage needs to be analyzed to assess which walls are suitable for refinement. The point cloud dataset is flattened to 2D and tiled to a  $2\text{ m} \times 2\text{ m}$  grid. Within each cell of the grid, a sum of points is calculated and a rejection threshold for numbers lower than the 80th percentile value is introduced. To avoid biases caused by too densely covered parts of a wall, a measure for the uniformity of the point distribution is proposed. First, a 2D buffer is created around a wall according to the object accuracy (as in subsection 3.1). The areas of patches and respective 2D buffers are calculated and the percentage ratio of those is obtained. The percentage ratio of 60% is utilized as a threshold for eligible walls for reconstruction. The process is visualized in Figure 4.

Since the analyzed structures are vertical-like planes, the RANdom SAmple Consensus (RANSAC) algorithm is utilized, which allows for certain deviations of the plane estimation. It enables outliers filtering, which, due to prior operations, is performed within a shrunken area, as shown in Figure 5. This makes the al-

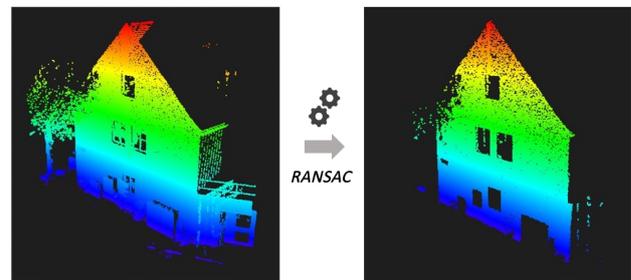


Figure 5. RANSAC applied to the extent shrunken using semantics of existing models

gorithm more robust by minimizing the possibility to fit a plane to an irrelevant object within a point cloud subset. Also, this assures consistency w.r.t to the input model. The parameters of RANSAC are designed to utilize a general plane model with observations as an unordered set of pre-processed points, with the topmost number of iterations set to 100, while the distance threshold is set to 0.1 m taking into account the high density of MLS point clouds.

### 3.3 Surface reconstruction

In order to use external implementations for point cloud processing within the pipeline, the MeshLabServer is controlled via FME and the parameterization is realized via automatically generated XML configurations. The reconstruction is performed as follows: First, computation of the normals for the input point clouds. Second, application of the Screened Poisson surface reconstruction algorithm (Kazhdan and Hoppe, 2013). Third, simplification with the Quadric Edge Collapse Decimation function (Corsini et al., 2012).

The reconstruction success is influenced by four main parameters. The parameter *adaptive octree depth* of the Screened Poisson algorithm controls the resolution of the reconstructed surface, where the value 10 is selected as default. Here, a larger number reflects a higher resolution of the reconstruction, but also a higher computational cost. The *target number of faces* and *percentage reduction* parameters of the Quadric Edge Collapse Decimation function control the ultimate number of faces of the algorithm. If there is an anticipated number of polygons, the *target number of faces* can simplify the complex mesh to a fixed number of faces. This parameter is prone to errors and an absolute number of faces is rarely known. Hence, the *percentage reduction* parameter is usually more suitable. The pipeline operator can estimate a rough anticipated representation of the refinement and, by typing-in a percentage, the mesh is simplified by this number. The *post-simplification cleaning* option enables the suppression of features that have unreferenced vertices, bad faces, and similar errors.

Due to the utilization of the Screened Poisson algorithm, a reconstruction of a continuous surface is enforced. This is an advantage in the case of unstructured datasets like MLS point clouds accommodating for gaps in the dataset. However, it also results in the overestimation of the end range. Thus, a mask of the raw model extent is applied to reduce the area and assure compatibility with the input model.

The semantics of the raw model is transferred to parts and groups in the cutting part to ensure compatibility with the whole input

city model. Additionally, generic attributes are added to distinguish the raw from the refined geometries. The *Timestamp* marks the refinement date in UTC format, the *FeatureNo* indicates the number of refined faces per single feature, and *HasGeoRefined* enables querying only reconstructed objects. Allowed GML geometries suitable for storing such refined objects are saved as *MultiSurface*. Depending on the class of the city model object, the reconstructed geometry can either be replaced with a raw geometry or be added as an additional feature. For example, to create a CityGML 2.0 compliant building representation, a class *WallSurface* can be utilized to store a raw wall geometry in LoD2 whereas the refined one in LoD3 – pointing to the same *Building*. However, this is not a feasible solution for the geometric refinement of a model that is already represented in the highest LoD.

### 3.4 Semantic enrichment

New challenges for city models are being addressed through the ongoing revision of state-of-the-art data models, as exemplified by CityGML 3.0 (Kutzner et al., 2020). This involves not only the revision of concepts, but also the introduction of new feature classes like *Hole* and *HoleSurface* placed within the CityGML 3.0 ecosystem (detailed relations with other city objects explained in (Kutzner et al., 2020)) to accommodate for emerging application areas. We present an automatic semantic enrichment method for water manhole covers defined by *Hole* and *HoleSurface* CityGML 3.0 classes. The method utilizes prior knowledge based on national norms, refined geometries as well as intensity values of MLS point clouds. The method's overview is illustrated in Figure 6. Manhole covers can be distinguished from

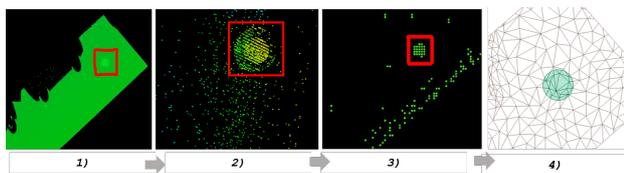


Figure 6. Step 1, 2 and 3 show the manhole detection with the red rectangles encompassing the approximate location. Step 4 shows the explicitly modeled geometry (green) of the manhole within the refined road segment

the surrounding road surface based on their structure, material and shape. Since these characteristics depend on the respective countries, the German national norm class *D 400* is utilized in our case. It is assumed that this approach is also applicable to manhole cover types in other countries by adjusting the intensity and geometry patterns of the respective national or international standards.

The selection of the region of interest (ROI) is obtained as described under subsection 3.1. Here, a road segment is delineating the ROI, as a manhole cover is assumed to be located within a road surface. Although the measured intensity values depend on instrumental effects, acquisition geometry, and environmental effects, the intensity distributions can provide clues to material properties. For example, the *stucco building* class is in the range of 28400 to 29200 intensity value (Kashani et al., 2015), which corresponds fairly well to the rough concrete surface used for the manholes in Germany. After a min-max normalization of the measured intensity values to the target range of 18000 to 32800, all points not matching the manhole cover filling can be filtered out. Due to the presence of noise and varying acquisition conditions, this processing step does not yet return an absolute position of the manhole cover, as shown in Figure 6 2). Therefore, in order

to find the location of a manhole, a density measure is pursued. The point cloud is transformed into an image with a pixel size of  $0.1\text{ m} \times 0.1\text{ m}$  storing the number of points as a band value. The pixels now serve as patches of the point cloud representing the corresponding density. To simplify processing, the pixels are coerced to vector points that contain an attribute indicating the total number of points in a patch. The patches are presented in Figure 6 3). Afterwards, the 10 densest points are chosen to reject the sparsest regions. The final search is decided based on an overlap check. The buffers of 0.2 m around each point (due to  $0.1\text{ m} \times 0.1\text{ m}$  pixel size of input points) are introduced. The densest buffered patches, overlapping at least five times, determine whether there is a manhole within that segment. If the test is positive, the most overlapping part is selected as an area within each center of a manhole is localized. In order to find the final center location, a gravity center is extracted from a polygon as a seed point. This seed point serves as a location for the search of the manhole's center point creating a new area of interest with the radius of: *Diameter of the manhole + diameter of the stucco part + introduced pixel size as possible deviations*. The dense patches found (with at least 10 points per patch) within this area serve as features to calculate the manhole's center point as the centroid of the patches.

The modeling of the manhole is performed as a cut around the center with a diameter of 0.785 m (based on the respective manhole class). Then, the manholes are stored in CityGML 3.0 as independent geometries of a road segment, as illustrated in Figure 6 4). The revised CityGML standard allows to explicitly represent manholes as a class *Hole* (holding semantics) and *HoleSurface* which is designed to represent the surface geometry of the manhole cover.

## 4. EXPERIMENTAL RESULTS

### 4.1 Datasets

The testing area has an extent of roughly  $0.5\text{ km} \times 0.5\text{ km}$  and is located within the city center of Ingolstadt, Bavaria, Germany. The urban location is typical for a central European city not exceeding 200 000 inhabitants and consists of historic buildings, urban roads, city furniture, and vegetation. The plethora of available datasets depicted in Figure 7 enhances the validation possibilities of the presented pipeline. Moreover, the utilized LoD3 buildings are published as open data,<sup>2</sup> enabling further investigations. In order to evaluate the method, buildings and roads served as vertical-like and horizontal-like structures, respectively. The buildings with the lowest available LoD1 were selected for more challenging testing as they have a lower accuracy and the least number of additional attributes. The same applies to roads where only drivable segments have been selected for testing. The LoD2 and LoD3 building models served for validation purposes. The MLS and ALS datasets consist of co-registered point clouds.

### 4.2 Results evaluation

Besides testing the method itself, the evaluation provides insights into the influence of the introduced parameters on the final results. Within the evaluation process, the *percentage reduction* parameter was fixed to 0.01 % in order to compare the effects of the other parameters under constant conditions. According to the suggestions of Kazhdan et al., the parameter values 8, 10 and 12 were applied for the *adaptive octree level*. All experiments were conducted on a computer with following parameters: Intel

<sup>2</sup> <https://github.com/savenow/loD3-road-space-models>

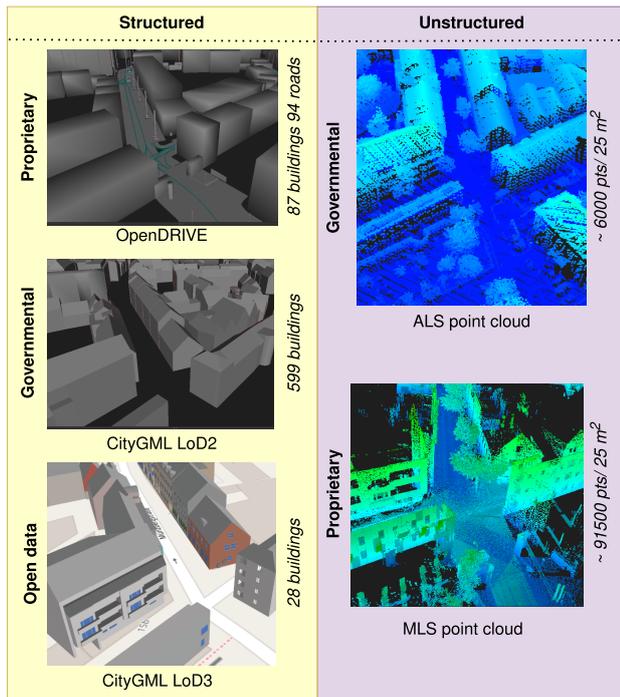


Figure 7. Overview of the Ingolstadt datasets

Core i7-8750H CPU @2.20 GHz as processor, 16 GB for memory (RAM) and Windows 10 as operating system.

### Accuracy assessment

The quantitative assessment of the refined structures is measured using the one-sided Hausdorff distance (Cignoni et al., 1998). The testing scenario is designed to compare the refined building and road structures at different octree levels (sampled surface) with the available city model (target surface). The horizontal-like objects that are represented by 94 road segments depict the surface within the input borders of the HD Map features. This ensures the input topology relation between adjacent objects. However, this approach prevents refinements of the feature's extent and thus the final refinement is highly dependent on the quality of the input vector dataset. The utilization of a supportive dataset (i.e., ALS point cloud) increases the stability of the surface reconstruction. As depicted in Figure 8, the comparison of raw

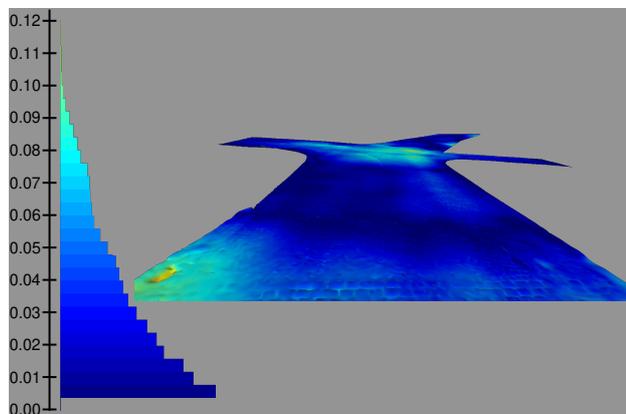


Figure 8. Deviations, obtained by Hausdorff metric given in meters, projected to the refined road surface at level 12 in comparison to the input road model

geometries enables verification or modeling of road's superelev-

ation. Additionally, depending on the accuracy, relatively small changes are modeled by this method. For example, cobblestone structures and potholes can be observed in Figure 8.

The vertical-like objects consist of 87 buildings in this test scenario. The coverage analysis (see subsection 3.2) has rejected 18 buildings from the reconstruction process. This accelerated the reconstruction process and avoided reconstruction errors. Furthermore, only those LoD1 walls were accepted for further reconstruction for which the corresponding LoD3 wall contained an *DataAvailable* attribute of *Sufficient* (except two on the periphery of the area). These attributes have been added by the creators of the LoD3 dataset and document the MLS point cloud coverage of the LoD3 buildings. Similar to the road segment experiment, the assumption of rigid boundaries has certain advantages and disadvantages that also apply to the buildings. For example, due to the rigid borders of the LoD1 input model, the modeling of walls of gable roof buildings present in LoD2 and 3 was prevented, as shown in Figure 9. On the other hand, an increased depiction of details on the building surface, such as windows and doors, can be observed. These are not present in LoD2 but LoD3 building models. Ultimately, the refined structure shows higher geometric details and captures even small deviations compared to the generalized geometries of the LoD3 building model, as shown in Figure 9.

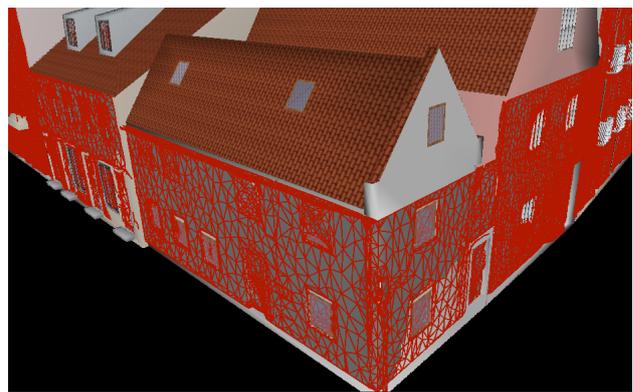


Figure 9. Comparison of a refined LoD1 buildings geometry (red edges) with LoD3 buildings

Moreover, the additional building features not present and significantly distant from the searched plane in the input dataset like balconies (in case of LoD1) are not reconstructed. Also, objects adjacent to buildings, such as tree branches, can be misclassified as buildings parts. This is due to the assumption that the RANSAC algorithm should find one portion of inliers per building feature. However, this only occurs if the object is located within the respective accuracy range and on the prolonged plane direction and within the plane margin introduced by the RANSAC fitting plane model. This can be extended by the introduction of another stopping criterion.

Since the walls of the LoD1 building models are the subject of the refinement, this comparison reflects the deviations between the raw buildings and the reconstructed surfaces that shall be perceived as a gain of the method. The validation, however, is performed using the building models in LoD2 and LoD3. As shown in Table 1, the validation against LoD3 confirms that the refined structures at the highest octree level 12 have the highest quality w.r.t. chosen measure. The discrepancies encountered when comparing to the LoD2 models are due to the different measurement techniques. The outliers present in the *max* column of Table 1 are caused by falsely segmented points or balconies, as shown in Figure 10, where the histogram indicates that most faces

Octree level	LoD1 (raw geom.)			LoD2 (ref. geom.)			LoD3 (ref. geom.)		
	max	$\mu$	RMS	max	$\mu$	RMS	max	$\mu$	RMS
8	6.85	0.18	0.41	6.80	0.41	0.69	4.78	0.18	0.43
10	3.93	0.16	0.31	6.70	0.37	0.60	4.40	0.13	0.31
12	2.94	0.16	0.31	5.88	0.39	0.63	3.59	0.11	0.27

Table 1. Comparison of refined LoD1 building geometries with the geometries of LoD1-3 building models, with all metrics given in meters

Octree level	LoD2 (raw geom.)		
	max	$\mu$	RMS
8	0.24	0.04	0.05
10	0.24	0.04	0.05
12	0.25	0.04	0.04

Table 2. Comparison of the refined road geometries with the raw geometry, with all metrics given in meters

deviate by about 0.2 m distance. The deviation measures for the road segments (see Table 2) show no significant gain when the octree level is increased. Nevertheless, the qualitative assessment indicates that more details can be extracted, as presented in Figure 8.

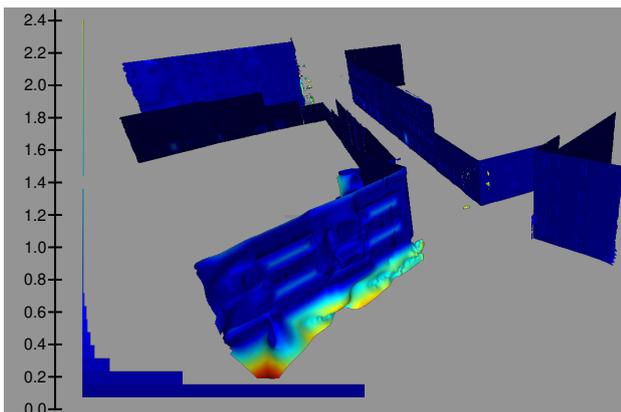


Figure 10. Deviations, obtained by Hausdorff metric given in meters, projected to refined structures at level 12 in comparison to LoD3 buildings

### Evaluation of geometric fidelity & its impact

For our test datasets, the octree level 10 was found to be a suitable compromise between processing time, exploration possibilities, and occupation of disk space. While the main benefit of octree level 12 is the high degree of detail, it is also necessary to take into account the large amount of memory required – 120 MB for 94 road segments, whereas level 10 requires only 25 MB in such a case. For buildings, this translates to 437 MB for 69 refined buildings, while level 10 requires only 138 MB. The octree depth parameter also has a high influence on the final processing time. It spans from roughly 25 min for 94 reconstructed road segments to almost 140 min at the level 8 and 12, respectively. The differences in computational time between horizontal and vertical-like objects emphasize the complexity of structures and of the dedicated algorithm. However, the selection of the parameter value should be guided by the final reconstruction requirements, whereby this parameter revealed to be the most influential.

### 4.3 Possible applications

The enriched models from the experimental results have been used to create an interactive game that is shared in the aforementioned GitHub repository<sup>1</sup> – the visualization is shown in Figure 11. This confirms that semantic models can be used in the



Figure 11. Refined models used in city models management tool (left) and automated driving simulator engine (right)

Unreal Engine software which is an engine of tools like CARLA that serves purposes of automated driving research. Besides, the models can be utilized in 3D GIS solutions like the 3DCityDB-Web-Map-Client, as shown in Figure 11, and serve the purposes of a 3D or 4D cadastre (Döner et al., 2011), as our concept also includes the time factor. Since semantic LoD3 road space models required for validation are currently only available in Ingolstadt to the best of our knowledge, the presented pipeline was tested with datasets from this area. The pipeline is expected to generate comparable results for mid-sized cities in Europe, but the transferability should be further examined for more architectural styles, such as skyscraper environments of megacities.

## 5. CONCLUSIONS & OUTLOOK

This work presented a first implementation of the proposed pipeline concept for automated geometry refinement and semantic enrichment of existing 3D city models using MLS point clouds. The solution proved that pre-existing knowledge from semantic city models can be incorporated to reduce the complexity of point clouds segmentation for refinement purposes. In order to generate suitable results for various application needs, the pipeline was implemented as an end-to-end solution with refinement modules that can be parameterized before the launch. Moreover, the effects of parameter variations were evaluated by comparing the refined geometries obtained from the pipeline with LoD2 and LoD3 building models that served as references. It was shown that the refinement can substantially reduce the geometric deviation to the LoD3 building models, whereby the resulting geometries required considerably more storage space and computational power. Furthermore, a method for semantic enrichment for manholes has been successfully integrated into the pipeline and already supports the export of CityGML 3.0 datasets, whereas a validation of this method is intended as one of the next steps. Since the RANSAC method currently estimates only one plane per wall surface, the next step is to investigate the enrichment of balconies, building installations, and also stairs. This applies not only to façade elements but also to street space objects in general, for which the position may already be known in the HD Map, such as trees, bushes, fences, and wall barriers.

Every set of parameters used for refinement and enrichment thereby leads to a result that represents a tradeoff between conflicting objectives (e.g., simulation accuracy vs. simulation runtime). Both the weighting of the objectives and the impact of the model characteristics on these objectives depend on the requirements and preferences of the applications and its user. Hence, the question arises of how to formalize these requirements and preferences for 3D models. Based on the formalizations, a

pipeline could find the set of parameters that leads to the result which is Pareto optimal for the particular application.

## 6. ACKNOWLEDGMENTS

This work is supported by the German Federal Ministry of Transport and Digital Infrastructure (BMVI) within the Automated and Connected Driving funding program under Grant No. 16AVF2145A (SAVe). Moreover, the work was carried out as part of the Leonhard Obermeyer Center at the Technical University of Munich.

## References

Association for Standardization of Automation and Measuring Systems, 2020. ASAM OpenDRIVE 1.6 Base Standard. Technical report. <https://www.asam.net/standards/detail/opensdrive/> (29 March 2020).

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S. and Çöltekin, A., 2015. Applications of 3D City Models: State of the Art Review. *ISPRS Int. J. of Geo-Inf.* 4(4), pp. 2842–2889.

Chehata, N., Guo, L. and Mallet, C., 2009. Airborne LIDAR feature selection for urban classification using random forests. *ISPRS - Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XXXVIII-3/W8, pp. 207–212.

Cignoni, P., Rocchini, C. and Scopigno, R., 1998. Metro: measuring error on simplified surfaces. In: *Computer Graphics Forum*, Vol. 17, Blackwell Publishers, pp. 167–174.

Corsini, M., Cignoni, P. and Scopigno, R., 2012. Efficient and flexible sampling with blue noise properties of triangular meshes. *IEEE Trans. Vis. Comput. Graph.* 18(6), pp. 914–924.

Döner, F., Thompson, R., Stoter, J., Lemmen, C., Ploeger, H., van Oosterom, P. and Zlatanova, S., 2011. Solutions for 4D cadastre—with a case study on utility networks. *Int. J. Geogr. Inf. Sci.* 25(7), pp. 1173–1189.

Gröger, G., Kolbe, T. H., Nagel, C. and Häfele, K.-H., 2012. OGC City Geography Markup Language CityGML Encoding Standard. Open Geospatial Consortium: Wayland, MA, USA, 2012.

Haala, N. and Kada, M., 2010. An update on automatic 3d building reconstruction. *ISPRS J. Photogramm. Remote Sens.* 65(6), pp. 570 – 580. ISPRS Centenary Celebration Issue.

Habib, A., Datchev, I. and Bang, K., 2010. A comparative analysis of two approaches for multiple-surface registration of irregular point clouds. *ISPRS - Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XXXVIII-1, pp. 61–66.

Kashani, A., Olsen, M., Parrish, C. and Wilson, N., 2015. A Review of LIDAR Radiometric Processing: From Ad Hoc Intensity Correction to Rigorous Radiometric Calibration. *Sensors* 15, pp. 28099–28128.

Kazhdan, M. and Hoppe, H., 2013. Screened poisson surface reconstruction. *ACM Trans. Graph.* 32, pp. 1–13.

Kazhdan, M., Bolitho, M. and Hoppe, H., 2006. Poisson surface reconstruction. In: *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, Eurographics Association, Goslar, DEU, p. 61–70.

Kutzner, T., Chaturvedi, K. and Kolbe, T. H., 2020. CityGML 3.0: New Functions Open Up New Applications. *PFG – J. Photogramm. Remote Sens. Geoinf. Sci.* pp. 1–19.

Lu, Y. and Rasmussen, C., 2012. Simplified Markov random fields for efficient semantic labeling of 3D point clouds. In: *2012 IEEE/RSJ IROS, 7 - 12 October 2012, Algarve, Portugal*, pp. 2690–2697.

Niemeyer, J., Rottensteiner, F. and Soergel, U., 2014. Contextual classification of lidar data and building object detection in urban areas. *ISPRS J. Photogramm. Remote Sens.* 87, pp. 152–165.

Schwab, B. and Kolbe, T. H., 2019. Requirement Analysis of 3D Road Space Models for Automated Driving. *ISPRS - Int. Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* IV-4/W8, pp. 99–106.

Schwab, B., Beil, C. and Kolbe, T. H., 2020. Spatio-Semantic Road Space Modeling for Vehicle–Pedestrian Simulation to Test Automated Driving Systems. *Sustainability* 12(9), pp. 3799.

Sint, R., Schaffert, S., Stroka, S. and Ferstl, R., 2009. Combining Unstructured, Fully Structured and Semi-Structured Information in Semantic Wikis. In: *CEUR Workshop Proceedings*, Vol. 464, pp. 73–87.

Stouffs, R., Tauscher, H. and Biljecki, F., 2018. Achieving complete and near-lossless conversion from ifc to citygml. *ISPRS Int. J. Geo-Inf.* 7(9), pp. 1–17.

Tuttas, S. and Stilla, U., 2013. Reconstruction of façades in point clouds from multi aspect oblique ALS. *ISPRS - Int. Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* II-3/W3, pp. 91–96.

Vosselman, G. and Maas, H.-G., 2010. Airborne and terrestrial laser scanning. CRC Press.

Wang, P.-S., Liu, Y., Guo, Y.-X., Sun, C.-Y. and Tong, X., 2017. O-CNN: Octree-based Convolutional Neural Networks for 3D Shape Analysis. *ACM Trans. Graph.* 36(4), pp. 72.

Weinmann, M., Schmidt, A., Mallet, C., Hinz, S., Rottensteiner, F. and Jutzi, B., 2015. Contextual classification of point cloud data by exploiting individual 3d neighbourhoods. *ISPRS - Int. Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* II-3/W4, pp. 271–278.

Willenborg, B., Pültz, M. and Kolbe, T. H., 2018. Integration of Semantic 3D City Models and 3D Mesh Models for Accuracy Improvements of Solar Potential Analyses. *ISPRS - Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XLII-4/W10, pp. 223–230.

Wysocki, O., 2020. Semantic-based Geometry Refinement of 3D City Models for Testing Automated Driving. Master's Thesis, Technical University of Munich.

Xu, Y. and Stilla, U., 2021. Towards building and civil infrastructure reconstruction from point clouds: A review on data and key techniques. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 14, pp. 2857–2885.

Xu, Y., Tuttas, S., Hoegner, L. and Stilla, U., 2018a. Reconstruction of scaffolds from a photogrammetric point cloud of construction sites using a novel 3D local feature descriptor. *Autom. Constr.* 85, pp. 76–95.

Xu, Y., Tuttas, S., Hoegner, L. and Stilla, U., 2018b. Voxel-based segmentation of 3D point clouds from construction sites using a probabilistic connectivity model. *Pattern Recognit. Lett.* 102, pp. 67–74.

Xue, F., Wu, L. and Lu, W., 2021. Semantic enrichment of building and city information models: A ten-year review. *Adv. Eng. Inform.* 47, pp. 101245.

Yang, B., Dong, Z., Liang, F. and Liu, Y., 2016. Automatic registration of large-scale urban scene point clouds based on semantic feature points. *ISPRS J. of Photogramm. Remote Sens.* 113, pp. 43–58.